Kalman Filter

Edited by Vedran Kordic

Kalman Filter

Edited by

Vedran Kordić

Kalman Filter
http://dx.doi.org/10.5772/233
Edited by Vedran Kordic

Notice

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,200+
Open access books available

116,000+
International authors and editors

125M+
Downloads

151
Countries delivered to

Our authors are among the

Top 1%
most cited scientists

12.2%
Contributors from top 500 universities

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Meet the editor

Vedran Kordic is the founder and Board member of IntechOpen. After obtaining his Master's degree in Mechanical Engineering he continued his education at the Vienna University of Technology where he obtained his PhD degree in 2004. He worked as a researcher at the Automation and Control Institute, Faculty of Electrical Engineering, Vienna University of Technology until 2008. His studies in robotics lead him not only to a PhD degree but also inspired him to co-found and build the International Journal of Advanced Robotic Systems - world's first Open Access journal in the field of robotics. Vedran's background and experience remain crucial for the company's development as he is continuously engaged with overseeing the IT department.

Preface

The seminal Kalman filter paper, entitled A new approach to linear filtering and prediction problems, and published in 1960, reformulated the Wiener problem and proposed a new solution based on state transition, avoiding the stationary limitations of the Wiener filter and giving a more suitable algorithm to be implemented in computers. This paper concludes with a prophetic sentence: "... The Wiener problem is shown to be closely connected to other problems in the theory of control. Much remains to be done to exploit these connections."

The Kalman filter has been successfully employed in diverse knowledge areas over the last 50 years and the chapters in the book review its recent applications. We hope the selected works will be useful for readers, contributing to future developments and improvements of this filtering technique.

The aim of this book is to provide an overview of recent developments in Kalman filter theory and their applications in engineering and science. The book is divided into 20 chapters corresponding to recent advances in the filed.

Editor

Vedran Kordić

Vienna University of Technology
Austria, European Union

Contents

Kalman Filter Applications
for Traffic Management

Constantinos Antoniou[1], Moshe Ben-Akiva[2] and Haris N. Koutsopoulos[3]
[1]National Technical University of Athens,
[2]Massachusetts Institute of Technology,
[3]Royal Institute of Technology,
[1]Greece
[2]U.S.A.
[3]Sweden

1. Introduction

Traffic congestion is a major problem in urban areas that has a significant adverse economic impact through deterioration of mobility, safety and air quality. As a result, the importance of better management of the road network to efficiently utilize existing capacity is increasing. To that end, many urban areas build and operate modern Traffic Management Centers (TMCs), which perform several functions, including collection and warehousing of real-time traffic data, as well as utilization of this data for various dynamic traffic control and route guidance applications. In order to be effective, these applications —which include Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS)— require traffic models that provide, in real-time, estimation and prediction of traffic conditions.

The complexity of transportation systems often dictates the use of detailed simulation-based Dynamic Traffic Assignment (DTA) models (Ben-Akiva et al., 1991, 2002, Mahmassani, 2001) for this purpose. Dynamic Traffic Assignment (DTA) systems support both planning and real–time applications. Planning applications may include the off–line evaluation of incident management strategies, the evaluation of alternative traffic signal and ramp meter operation strategies and the generation of evacuation and rescue plans for emergencies (e.g. natural disasters) that could affect the traffic network. Real–time applications make use of the traffic prediction capabilities of DTA systems and may include on–line evaluation of guidance and control strategies, real–time incident management and control, support of real–time emergency response efforts and optimization of the operation of TMCs through the provision of real–time predictions.

Real–time DTA systems typically comprise two main functions: traffic state estimation, and traffic prediction (Ben-Akiva et al., 2002). An overview of the state-of-the-art Dynamic Traffic Assignment framework is shown in Fig. 1. DTA functionality is supported by two main modules: a demand simulator and a supply simulator. The demand simulator fuses surveillance information with historical information for the estimation and prediction of the evolving demand patterns. This is achieved through a combination of aggregate predictive

models and disaggregate behavioral models (Antoniou et al., 1997). The supply simulator is usually based on high–level (mesoscopic or macroscopic) models that represent traffic dynamics using speed-density relationships, kinematic representation of traffic elements of queueing theory, etc. A detailed treatment of the demand-supply interactions within a state-of-the-art DTA system can be found in Ben-Akiva et al. (2002).
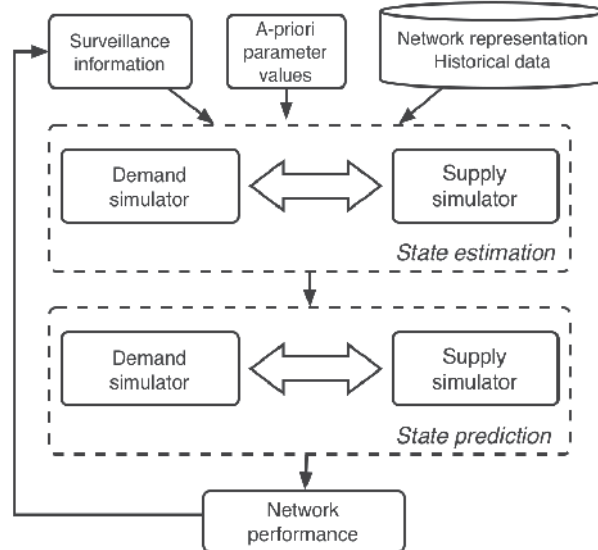
Fig. 1. Dynamic traffic assignment framework overview

In the current DTA framework, only the OD flows are calibrated on-line: one of the key components of dynamic traffic assignment is the Origin-Destination (OD) estimation and prediction process (Ashok & Ben-Akiva, 1993, Ashok, 1996, Ashok & Ben-Akiva, 2000, 2002). OD estimation combines historical and real-time information to obtain dynamic demand matrices. However, a number of other parameters are used by models in the demand simulator and the supply simulator. On the supply side parameters include speed–density relationship parameters and output capacities of network links and intersections. On the demand side, additional parameters (besides the OD flows) include behavioral model parameters.

In most cases, the approach to the problem of calibration of these parameters has been to perform off–line calibration of the simulation models using a database of historic information. The calibrated parameter values are then used in the on–line simulations. The calibrated model parameters therefore represent average conditions over the period represented in the data. Models that were calibrated this way may produce satisfactory results in off–line evaluation studies, which are concerned with the expected performance of various traffic management strategies.

However, this may not be the case in real-time applications, which are concerned with the system performance on the given day. If the model calibrated off-line is used without adjustment, the system is not sensitive to the variability of the traffic conditions between days, which are the result of variations in the parameters of the system, such as weather and surface conditions. Such variations may cause traffic conditions to differ significantly from the average values. Thus, the predictive power of the simulation model may be significantly reduced.

Speed–density relationships may depend on location–specific parameters, such as type of facility, number of lanes, lane width, slope, surroundings, as well as exhibit temporal variations, i.e. they may vary by season, day of the week, or even time of day, reflecting different driving behaviors (e.g. experienced drivers during commute periods). Off-line calibration could, in principle, deal with these situations, through the generation of a historical database of different speed-density relationships, categorized by the conditions. Based on the prevailing conditions, the "appropriate" relationship could then be retrieved and used. However, traffic dynamics also depend on factors that cannot always be anticipated or observed, such as weather conditions, incidents, unscheduled maintenance work, traffic mix. Even when these factors can be predicted, it would be impractical to calibrate traffic dynamics models for each possible scenario. Minor incidents (such as a car slowing down in the break–down lane) that are not reported or captured otherwise in the system may also impact the traffic dynamics.

The output capacity of the network links and intersections is another example. Average values could in general be obtained during an off-line calibration phase. However, capacities are affected by several phenomena (including weather and lighting conditions, traffic composition, etc.) and may therefore change as prevailing conditions change.

To overcome this problem, real-time data can be used to re-calibrate and adjust the model parameters on-line, so that prevailing traffic conditions can be captured more accurately. The wealth of information included in the off-line values can be incorporated into this process by using them as a priori estimates.

The remainder of this chapter is organized as follows. Section 2 presents a review of relevant literature, Section 3 presents a formulation of the on–line calibration problem as a state–space model, and Section 4 presents applicable solution approaches. Section 5 presents an application of the methodology to a network in Southampton, U.K., and Section 6 concludes the chapter with a summary and directions for further research.

2. Literature review

The topic of on-line calibration of traffic simulation models has received only limited attention in the literature. This section presents a review of prior on–line calibration research. System–level approaches are presented first, followed by research focused on individual components.

2.1 System–level approaches

Doan et al. (1999) outline a framework for periodic adjustments to a traffic management simulation model in order to maintain an internal representation of the traffic network consistent with that of the actual network. The authors categorize the error sources as demand estimation, path estimation, traffic propagation, internal traffic model structure, and on–line data observation and propose a system of on–line and off–line adjustment modules. A similar approach is proposed in Peeta and Bulusu (1999), where consistency is sought in terms of minimizing the deviations of the predicted time–dependent path flows from the corresponding actual flows. He et al. (1999) develop a combined off–line and on–line calibration process to adjust the analytical dynamic traffic model's output to be consistent with real–world traffic conditions by periodically detecting inconsistencies between model outputs andreal–world data, and actuating the correction model to correct the errors.

2.2 Supply parameters

van Arem and van der Vlist (1992) developed an on–line procedure for the estimation of current capacity at a motorway cross–section. The procedure is based on the combination of an on–line estimation of a "current" fundamental diagram with a maximum occupancy that may be achieved under free–flow conditions. The capacity is estimated by substituting the current maximum occupancy into the current fundamental diagram.

Tavana and Mahmassani (2000) use transfer function methods (bivariate time series models) to estimate dynamic speed–density relations from typical detector data. The parameters are estimated using the past history of speed–density data; no predetermined parameters or shape for the model are assumed. The method is based on time series analysis, using density as a leading indicator. Hyunh et al. (2000) extend the work of Tavana and Mahmassani (2000) by incorporating the transfer function model into a DTA simulation–based framework. Furthermore, the estimation of speeds using the transfer function model is implemented as an adaptive process, where the model parameters are updated on–line based on the prevailing traffic conditions. Qin and Mahmassani (2004) evaluate the same model with actual sensor data from several links of the Irvine, CA, network. In this chapter, determination of system input and output is derived from the higher–order continuum model. From the numerical results, the performance and the robustness of the transfer function model is in general found to be superior to the static model.

Van Lint et al. (2002) develop a state-space formulation of the travel time prediction problem and use it to derive a recurrent state-space neural network (SSNN) topology that captures the highly non-linear characteristics of the freeway travel time prediction problem. Van Lint et al. (2005) extend the model with preprocessing strategies based on imputation in order to achieve accuracy and robustness with respect to missing or corrupt data. Liu et al. (2006) present two distinct ways of using Extended Kalman Filters to address the problem of short-term urban arterial travel time prediction. Van Lint (2006) proposes a delayed EKF method for the online incremental training of a data driven travel time prediction model (a state-space neural network) for the prediction of travel times.

Antoniou et al. (2005) formulate the problem of on-line calibration of the speed-density relationship as a flexible state-space model and present applicable solution approaches. Three of the solution approaches [Extended Kalman Filter (EKF), Iterated EKF, and Unscented Kalman Filter (UKF)] are implemented and an application of the methodology with freeway sensor data from two networks in Europe and the U.S. is presented. The EKF provides the most straightforward solution to this problem, and indeed achieves considerable improvements in estimation and prediction accuracy. The benefits obtained from the —more computationally expensive— Iterated EKF algorithm are shown. An innovative solution technique (the UKF) is also presented.

Wang and Papageorgiou (2005) present a general approach to the real-time estimation of the complete traffic state in freeway stretches. They use a stochastic macroscopic traffic flow model, and formulate it as a state-space model, which they solve using an Extended Kalman Filter. The formulation allows dynamic tracking of time-varying model parameters by including them as state variables to be estimated. A random walk is used as the transition equations for the model parameters. A detailed case study of this methodology is presented in Wang et al. (2007).

Boel and Mihaylova (2006) present a stochastic model of freeway traffic suitable for on-line estimation. The model is estimated using a recursive filter based on Monte Carlo techniques (called also particle filters). Ben Aissa et al. (2006) use sequential Monte-Carlo or particle filter methods for the estimation and prediction of travel time.

2.3 Behavioral parameters

Peeta and Yu (2006) propose a behavior–based consistency–seeking modeling approach to bridge the functional gaps between route choice models and dynamic traffic assignment models vis–a–vis predicting the time-dependent network traffic flow patterns. The approach consistently addresses day–to–day learning and within–day dynamics using a single hybrid probabilistic–possibilistic behavioral model (Peeta & Yu, 2004, 2005) through intuitive if–then rules that are based on the findings of past studies in the literature. The approach avoids rigid assumptions on driver behavioral tendencies and a priori knowledge of driver behavior class fractions, and enables the classification of information characteristics and the consistent modeling of information effects. The proposed approach uses currently available data and achieves computational tractability by obviating a search procedure to predict the dynamically evolving traffic flow pattern.

2.4 Demand parameters

Ashok and Ben-Akiva (Ashok & Ben-Akiva, 1993, Ashok, 1996, Ashok & Ben-Akiva 2000, 2002) formulate the real–time OD estimation and prediction problem as a state–space model and solve it using a Kalman Filtering algorithm. One interesting characteristic of this approach is the use of deviations of OD flows (instead of the OD flows themselves) as variables. The use of deviations incorporates the wealth of structural information about spatial and temporal relationships between OD flows contained in the historical estimates into the OD estimation framework. The real–time OD estimation and prediction framework has been implemented in the DynaMIT DTA system (Antoniou et al., 1997, Ben-Akiva et al., 2002). An efficient solution algorithm for the OD estimation problem has been presented by Bierlaire and Crittin (2004).

Zhou and Mahmassani (2004) develop a similar Kalman-filter based adaptive OD estimation and prediction procedure using a polynomial trend filter to recursively capture demand deviations from a priori demand estimates.

2.5 Conclusion

The problem of on–line calibration of DTA systems has received some attention in the literature. Most existing methodologies, however, impose serious constraints and make restrictive assumptions. In particular, the components of a DTA system are considered in a sequential approach and iterative/heuristic approaches are proposed to estimate the appropriate parameters on–line.

Individual approaches for the on–line calibration of subsets of the parameters have also been developed. Such approaches update only a subset of the parameters in a DTA system. Therefore, all error or uncertainty is attributed to one source, which is unrealistic. Instead, an approach is needed that jointly estimates demand and supply parameters simultaneously and captures the complex demand and supply interactions (Ben-Akiva et al., 2002), thus ensuring consistency between the estimated parameters.

3. State–space formulation

A classical technique for dealing with dynamic systems is state–space modeling. In this section, the on–line calibration problem is formulated as a state–space model, comprising:

- Transition equations that capture the evolution of the state vector over time, and

- Measurement equations that capture the mapping of the state vector on the measurements.

Given that state–space models have been extensively studied and efficient algorithms have been developed to solve them, this formulation will lead us naturally to Section 4 where solution approaches are discussed.

The first step in developing a state–space model is to define the state vector. In this context, the parameters and inputs that need to be calibrated define the state. Measurement and transition equations are developed next, followed by a reformulation of the problem in terms of deviations.

3.1 State vector

The concept of the state (or state vector) is fundamental in the description of a state–space model. The state vector $\mathbf{x}_h$ is defined as the minimal set of data that is sufficient to uniquely describe the dynamic behavior of the system at time interval h (the assumption of a discrete, stochastic, dynamic system is made). The state vector includes the parameters $\boldsymbol{\pi}_h$ that need to be calibrated during time interval h. The main parameters for the on–line calibration problem for a DTA system include:

- OD flows,
- Behavioral model parameters, such as route, departure and mode choice model parameters,
- Speed–density relationship parameters, and
- Segment capacities.

It should be noted, however, that the approach is general and can easily incorporate a different set of parameters.

3.2 Measurement equations

Available information is associated with the unknown parameter values through measurement equations. A priori values of the model parameters provide direct measurements of the unknown parameters. Surveillance information, on the other hand, can be used to formulate indirect measurement equations, where the output of the simulator model S (when the unknown set of parameter values is used as input) would match the surveillance information.

By definition, a direct measurement provides a preliminary estimate of a parameter. Within the context of on–line calibration, preliminary estimates of the parameters are provided by the off–line calibration. Therefore, the vector of off–line calibrated parameter values $\boldsymbol{\pi}_h^{\alpha}$ can be used as an a priori estimate of the true parameter vector $\boldsymbol{\pi}_h$. The a priori values of the input parameters can be expressed as a function of the "true" parameters:

$$\pi_h^a = \pi_h + \mathbf{v}'_h \tag{1}$$

where $\mathbf{v}'_h$ is a vector of random error terms.

Direct measurements of some OD flows could also be available from advanced surveillance technologies, such as Automated Vehicle Identification (AVI) systems or probe vehicles. Such technologies allow the tracking of equipped vehicles as they move through the network, thus obtaining detailed surveillance information (based on a sample of the population). When the vehicles can be detected close to their origin and their destination, it

is possible to infer direct measurements of OD flows (Antoniou et al., 2004). Such information could easily be incorporated as additional direct measurements.

Practically any type of available traffic measurements can be used as indirect measurement equations, linking the observed traffic measurements with their simulated counterparts when a particular set of parameters is used as input. In the general case, modeled trips last longer than one interval. Therefore, simulated trajectories of vehicles are impacted by the traffic conditions during previous intervals (and consequently by the model parameters used during these intervals). The simulated traffic measurements during time interval h can therefore be represented as:

$$\mathbf{M}_h^s = S(\pi_h, \pi_{h-1}, ..., \pi_{h-p}) = S(\Pi_h) \tag{2}$$

where S is a mapping of the input parameters onto the measurements (representing the simulation model), p is the number of intervals required for the longest trip in the network, and $\Pi_h = \pi_h, \pi_{h-1}, ..., \pi_{h-p}$ is an augmented vector of parameters.

The relationship between the observed and the simulated measurements can then be written as follows:

$$\mathbf{M}_h^o = \mathbf{M}_h^s + \upsilon'_h \tag{3}$$

where $\upsilon'_h = \upsilon_h^f + \upsilon_h^s + \upsilon_h^m$ is a compound observation error comprising three error sources:

- υ_h^f captures structural errors (due to the inexactness of the simulation models),
- υ_h^s captures simulation errors (e.g. sampling and numerical errors), and
- υ_h^m captures measurement errors.

As it is not possible to distinguish between these three error components, however, they will be treated together. Furthermore, it is assumed that υ'_h is independent from the error vector $\mathbf{v}'_h$ introduced in Equation 1.

3.3 Transition equations

Transition equations capture the evolution of the state vector over time. A typical formulation for the transition equation relates the state during a given interval to a series of states from previous intervals. A general formulation of such a transition equation would be:

$$\pi_{h+1} = T(\pi_h, \pi_{h-1}, ..., \pi_{h-p}) + \eta'_h \tag{4}$$

where T is a function capturing the dependence of the parameter vector π_{h+1} during interval $h+1$ on the values of the parameter vector during the past several intervals, p is the number of past parameter vectors that are considered, and η'_h is a vector of random error terms.

A common approach to the representation of transition equations is the use of autoregressive processes. Expressed as an autoregressive function, the transition equation can be written in matrix form as follows:

$$\pi_{h+1} = \sum_{q=h-p}^{h} F_q^{h+1} \pi_q + \eta''_h \tag{5}$$

The three components of the state vector (OD flows, speed–density relationship parameters, capacities) represent distinct aspects of the transportation problem and have different

characteristics. Therefore, each of these may evolve over time according to a distinct autoregressive process. This can easily be handled by writing a separate transition equation like the one presented in Equation 5 for each such autoregressive process.

3.4 The idea of deviations

Suppose that the model parameters and inputs have been estimated from historical data for several previous days or months. These already estimated (demand and supply) parameters embody a wealth of information about the relationships that affect trip making and traffic dynamics, as well as their temporal and spatial evolution. It is desirable to incorporate as much historical information into the formulation as possible. The most straightforward way to achieve this is to use deviations of the model parameters from best available estimates instead of the actual parameters themselves as state variables. Thus, the model formulation would indirectly take into account all the available a priori structural information. The use of deviations has been proposed by Ashok and Ben-Akiva (1993) for the OD estimation and prediction problem.

Using deviations also has other benefits. Traffic flow variables have skewed distributions (unlike the normal distribution which is symmetric). On the other hand, the corresponding deviations of these variables from available estimates would have symmetric distributions and hence are more amenable to approximation by a normal distribution. A normal distribution for the model variables is a useful property for the available statistical tools such as the Kalman Filter extensions used in this research.

The state vector can therefore be expressed as deviations from best historical values: $\Delta\pi_h = \pi_h - \pi_h^H$. The transition equation can easily be reformulated with respect to the new state vector as:

$$\pi_{h+1} - \pi_{h+1}^H = \sum_{q=h-p}^{h} F_q^{h+1}(\pi_q - \pi_q^H) + \eta_h \Rightarrow$$

$$\Delta\pi_{h+1} = \sum_{q=h-p}^{h} F_q^{h+1} \cdot \Delta\pi_q + \eta_h \tag{6}$$

Similarly, the direct measurement equation can be written in deviations' form as:

$$\pi_h^a - \pi_h^H = \pi_h - \pi_h^H + \mathbf{v}_h \Rightarrow$$

$$\Delta\pi_h^a = \Delta\pi_h + \mathbf{v}_h \tag{7}$$

It should be noted that π_h^α and π_h^H capture essentially the same thing: an available estimate of the state vector. However, there are subtle differences and —in the interest of generality— a distinction is made. For example, the a priori parameters π_h^α may correspond to the parameters obtained from the off–line calibration, while the historical parameters π_h^H may refer to the latest available estimates (e.g. values obtained from the same interval the previous day).

Finally, the indirect measurement equation can be rewritten as:

$$\mathbf{M}_h - \mathbf{M}_h^H = S(\pi_h) - \mathbf{M}_h^H + \upsilon_h \Rightarrow$$

$$\Delta\mathbf{M}_h = S(\pi_h^H + \Delta\pi_h) - \mathbf{M}_h^H + \upsilon_h \tag{8}$$

where the historical traffic measurement vector $\mathbf{M}_h{}^H$ can be obtained from an evaluation of the historical parameters π_h:

$$\mathbf{M}_h^H = S(\pi_h) \tag{9}$$

3.5 The model at a glance

The on–line calibration algorithm has been expressed in deviations' form (where Equation 6 is the transition equation and Equations 7 and 8 are the measurement equations). The complete state–space model is shown below for clarity:

$$\Delta\pi_{h+1} = \sum_{q=h-p}^{h} F_q^{h+1} \cdot \Delta\pi_q + \eta_h$$

$$\Delta\pi_h^a = \Delta\pi_h + \mathbf{v}_h \tag{10}$$

$$\Delta\mathbf{M}_h = S(\pi_h^H + \Delta\pi_h) - \mathbf{M}_h^H + \upsilon_h$$

Before moving to the presentation of applicable solution approaches (Section 4), it is useful to express the model in the following form:

$$\mathbf{x}_{h+1} = \mathbf{f}(\mathbf{x}_h) + \mathbf{w}_h \tag{11}$$

$$\mathbf{y}_h = \mathbf{h}(\mathbf{x}_h) + \mathbf{u}_h \tag{12}$$

where Equation 11 is the transition equation and Equation 12 is the measurement equation. This form is obtained directly from Equations 10 if we denote

$$\mathbf{x}_h = \Delta\pi_h$$

$$\mathbf{y}_h = \begin{bmatrix} \Delta\pi_h^a \\ \Delta\mathbf{M}_h \end{bmatrix}$$

$$\mathbf{f}(\mathbf{x}_h) = \sum_{q=h-p}^{h} F_q^{h+1} \mathbf{x}_q$$

$$\mathbf{h}(\mathbf{x}_h) = \begin{bmatrix} \Delta\pi_h \\ S(\pi_h^H + \Delta\pi_h) - \mathbf{M}_h^H \end{bmatrix}$$

$$\mathbf{u}_h = \begin{bmatrix} \mathbf{v}_h \\ \upsilon_h \end{bmatrix}$$

Furthermore, the following assumptions are made on the error vectors $\mathbf{w}_h$ and $\mathbf{u}_h$:
1. $E[\mathbf{w}_h] = 0$

2. $E[\mathbf{w}_h \; \mathbf{w}'_m]=\mathbf{Q}_h\delta_{hm}$ where δ_{hm} is the Kronecker delta, i.e. $\delta_{hm}=1$ if $h=m$ and 0 otherwise $\forall h,m$, and $\mathbf{Q}_h$ is a variance–covariance matrix.
3. $E[\mathbf{u}_h]=0$
4. $E[\mathbf{u}_h \; \mathbf{u}'_m]=\mathbf{R}_h\delta_{hm}$ where δ_{hm} is the Kronecker delta, i.e. $\delta_{hm}=1$ if $h=m$ and 0 otherwise $\forall h,m$, and $\mathbf{R}_h$ is a variance–covariance matrix.
5. $E[\mathbf{u}_h \; \mathbf{w}'_m]=0$ $\forall h,m$, i.e. the errors of the transition and measurement equations are uncorrelated.

These assumptions allow for the derivation of the Kalman Filter–based solution approaches. The assumption of no serial correlation for the transition equation can be defended because the unobserved factors that could be correlated over time are captured by the historical matrix $\boldsymbol{\pi}^{\alpha}_h$ (or $\boldsymbol{\pi}^H_h$). In some situations (e.g. incidents), however, this assumption might break down. A violation of this assumption, however, can be easily taken care of by using a variant of the estimation algorithm. (An algorithm to handle correlated errors in the transition or measurement equations can be found, for example, in Chui and Chen, 1999). The assumption of no serial correlation for the measurement equation can be defended using a similar argument. However, this assumption might also break down if, for example, a specific detector consistently under–estimates or over–estimates a link volume on a particular day. Again, it is easy to relax this assumption and use a variant of the estimation algorithm.

3.6 An alternative formulation

The on-line calibration problem can be formulated as a minimization problem where the objective function aims to jointly minimize the following components:

-   ε^o_h: deviation of simulated traffic conditions $\mathbf{M}^s_h$ from the respective observed measurements $\mathbf{M}^o_h$, and
-   ε^{α}_h: deviation of a set of parameters and inputs π_h (over which the optimization is performed) from their a priori values $\boldsymbol{\pi}^{\alpha}_h$.

The objective function could then be expressed as:

$$\pi_h \min[N_1(\varepsilon^o_h) + N_2(\varepsilon^a_h)] \qquad (13)$$

where $N_i(.)$ are appropriate functions measuring the magnitude of the errors. For example, $N_i(.)$ may be the Euclidian norm.

Substituting the expressions for the error terms from Equations 1 and 3, the objective function can be restated as:

$$\pi_h \min[N_1(\mathbf{M}^o_h - \mathbf{M}^s_h) + N_2(\pi^a_h - \pi_h)] \qquad (14)$$

The above formulation can be made operational in a number of different ways, depending on the assumptions regarding the nature of the various error terms and the functional forms of $N_i(.)$. The various formulations may lead to different solution approaches with different convergence and computational properties. For example, if ε_{α} and ε_o are assumed to be normally distributed the formulation reduces to the following generalized least squares (GLS) problem:

$$\pi_h \min[(\mathbf{M}^o_h - \mathbf{M}^s_h)'\mathbf{W}^{-1}(\mathbf{M}^o_h - \mathbf{M}^s_h) +$$
$$+ (\pi^a_h - \pi_h)'\mathbf{V}^{-1}(\pi^a_h - \pi_h)] \qquad (15)$$

where $\mathbf{W}$ and $\mathbf{V}$ are the variance–covariance matrices of the measurements and a priori values, respectively. The solution π^*_h to this optimization problem can then be obtained from:

$$\pi^*_h = \pi_h \arg\min[(\mathbf{M}^o_h - \mathbf{M}^s_h)'\mathbf{W}^{-1}(\mathbf{M}^o_h - \mathbf{M}^s_h) + \\ + (\pi^a_h - \pi_h)'\mathbf{V}^{-1}(\pi^a_h - \pi_h)] \tag{16}$$

In an on–line application, however, this formulation would be impractical since the problem needs to be solved at every time interval, with all the information on previous time intervals (because of the temporal correlations between the errors). Stating this problem as a state–space model, which can then be solved efficiently using recursive methods such as Kalman Filtering techniques, overcomes this difficulty.

The on–line calibration approach can also be solved using other algorithms for non–linear systems of equations. A particularly suitable algorithm has been recently developed (Crittin, 2003, Crittin & Bierlaire, 2003) as a generalization of secant methods. The proposed class of methods calibrates a linear model based on several previous iterates. The difference with existing approaches is that the linear model to interpolate the function is not imposed. Instead, the linear model, which is as close as possible to the nonlinear function (in the least–squares sense), is identified.

4. Solution approaches

The Kalman Filter is the optimal minimum mean square error (MMSE) estimator for linear state–space models (Kalman, 1960). However, the on–line calibration approach is non–linear (due to the indirect measurement equation). Since many other interesting problems are non–linear, solutions for non–linear models have been sought, leading to the development of modified Kalman Filter methodologies. The most straightforward extension is the Extended Kalman Filter (EKF), in which optimal quantities are approximated via first order Taylor series expansion (linearization) of the appropriate equations (Kalman, 1960, Gelb, 1974). A special case of the EKF with very favorable computational properties is the Limiting Extended Kalman Filter (LimEKF; Limiting Kalman Filters are discussed e.g. in Chui & Chen, 1999). The Unscented Kalman Filter (UKF) (Julier et al., 1995; Julier & Uhlmann, 1997; Wan et al., 2000; Wan & van der Merwe, 2000; van der Merwe et al., 2000) is an alternative filter. The main difference between the EKF and UKF lies in the representation of the random variables for propagation through the system dynamics.

4.1 Extended Kalman filter
The Extended Kalman Filter employs a linearization of the non–linear relationship to approximate the measurement equation with a first–order Taylor expansion:

$$\mathbf{H}_h = \frac{\vartheta\mathbf{h}(\mathbf{x}^*)}{\vartheta\mathbf{x}^*}\bigg|_{\mathbf{x}^*=\mathbf{X}_{h|h-1}} \tag{17}$$

In words, the Extended Kalman Filter main steps are as follows. Suppose that a starting estimate of the state $\mathbf{X}_0$ is available (Equation 18), along with an estimate of the initial state variance–covariance matrix $\mathbf{P}_0$ (Equation 19). A time update phase makes a prediction of the state (Equation 20) and its covariance matrix (Equation 21) for the next time interval.

The measurement update phase incorporates the new information about the measurement vector $\mathbf{Y}_h$ and uses it to correct the prediction of the state made during the time update. The measurement matrix $\mathbf{H}_h$ is obtained through an intermediate linearization step (Equation 22). Instrumental in this process is the Kalman gain $\mathbf{G}_h$, which is computed as per Equation 23. The state can then be updated (corrected) using Equation 24. Similarly, the state covariance is updated using Equation 25.

Algorithm 1. Extended Kalman Filter
Initialization

$$\mathbf{X}_{0|0} = \mathbf{X}_0 \tag{18}$$

$$\mathbf{P}_{0|0} = \mathbf{P}_0 \tag{19}$$

for *h=1* to *N* **do**
    Time update

$$\mathbf{X}_{h|h-1} = \mathbf{F}_{h-1}\mathbf{X}_{h-1|h-1} \tag{20}$$

$$\mathbf{P}_{h|h-1} = \mathbf{F}_{h-1}\mathbf{P}_{h-1|h-1}\mathbf{F}_{h-1}^{T} + \mathbf{Q}_h \tag{21}$$

    Linearization

$$\mathbf{H}_h = \frac{\vartheta \mathbf{h}(\mathbf{x}^*)}{\vartheta \mathbf{x}^*}\Bigg|_{\mathbf{x}^* = \mathbf{X}_{h|h-1}} \tag{22}$$

    Measurement update

$$\mathbf{G}_h = \mathbf{P}_{h|h-1}\mathbf{H}_h^{T}\left(\mathbf{H}_h\mathbf{P}_{h|h-1}\mathbf{H}_h^{T} + \mathbf{R}_h\right)^{-1} \tag{23}$$

$$\mathbf{X}_{h|h} = \mathbf{X}_{h|h-1} + \mathbf{G}_h\left[\mathbf{Y}_h - \mathbf{h}\left(\mathbf{X}_{h|h-1}\right)\right] \tag{24}$$

$$\mathbf{P}_{h|h} = \mathbf{P}_{h|h-1} - \mathbf{G}_h\mathbf{H}_h\mathbf{P}_{h|h-1} \tag{25}$$

end for

Further information on the Extended Kalman Filter can be found in many texts, including for example Sorenson (1985) and Chui and Chen (1999).

The on–line calibration approach presented in previous sections does not — in general— have an analytical representation. Therefore, in order to perform the linearization step (Equation 22) it is necessary to use numerical derivatives. Assuming the use of central derivatives, it is necessary to evaluate the function *2n* times, where *n* is the dimension of the state vector. (If forward derivatives are used, then this number drops to *n+1* evaluations.) Each such evaluation implies one run of the simulator. Therefore, it becomes apparent that this process of linearization dominates the computational complexity of the algorithm.

4.2 Limiting extended Kalman filter

In this section, a special case of the Extended Kalman Filter is presented that significantly improves the computational performance of the algorithm. As mentioned in Section 4.1, the most computationally intensive step in the EKF algorithm is the linearization of the measurement equation (Equation 22), as it requires the use of numerical derivatives. Another costly operation is the inversion required for the computation of the Kalman gain (Equation 23). In real–time applications, it may be possible to replace the Kalman Gain matrix G_h by a constant gain matrix considerably decreasing the computation time. The limiting Kalman Filter will be defined by replacing G_h with its "limit" G, called the limiting (or stable) Kalman gain matrix (Chui & Chen, 1999). The main steps of the Limiting Extended Kalman Filter (LimEKF) algorithm are presented in Algorithm 2. The differences from the EKF algorithm are limited to the computation of the numerical derivative (which is not computed on–line in the LimEKF) and the use of the limiting Kalman gain G for every iteration (Equations 30 and 31).

The limiting Kalman gain matrix can be computed off–line. The simplest way would be to express the limiting Kalman gain matrix as the average of a number of available Kalman gain matrices:

$$\mathbf{G} = \frac{\sum_{m=1:M} \mathbf{G}_m}{M} \tag{32}$$

where $\mathbf{G}_m$ is the Kalman gain obtained from EKF during interval m and M is the total number of available Kalman gain matrices. Several strategies can be developed to improve the quality of the limiting Kalman gain. For example, the EKF could be run off–line, with each run producing a new Kalman gain matrix. These Kalman gain matrices could then be used to update the *limiting* Kalman gain matrix. Another strategy would be to consider only the last few Kalman gain matrices, i.e. use a type of moving average. Weighted averages (e.g. using lower weights for "older" gain matrices) can also be considered.

Algorithm 2. Limiting Extended Kalman Filter
Generation of *limiting* Kalman gain matrix $\mathbf{G}$ and $\mathbf{H}$
Initialization

$$\mathbf{X}_{0|0} = \mathbf{X}_0 \tag{26}$$

$$\mathbf{P}_{0|0} = \mathbf{P}_0 \tag{27}$$

for *h=1* to *N* **do**
　　Time update

$$\mathbf{X}_{h|h-1} = \mathbf{F}_{h-1}\mathbf{X}_{h-1|h-1} \tag{28}$$

$$\mathbf{P}_{h|h-1} = \mathbf{F}_{h-1}\mathbf{P}_{h-1|h-1}\mathbf{F}_{h-1}^{T} + \mathbf{Q}_h \tag{29}$$

　　Measurement update

$$\mathbf{X}_{h|h} = \mathbf{X}_{h|h-1} + \mathbf{G}\left[\mathbf{Y}_h - \mathbf{h}\left(\mathbf{X}_{h|h-1}\right)\right] \tag{30}$$

$$\mathbf{P}_{h|h} = \mathbf{P}_{h|h-1} - \mathbf{GHP}_{h|h-1} \qquad (31)$$

end for

The main component of the Kalman gain matrix is the derivative $\mathbf{H}_h$ of the measurement equation. This is directly required in Equation 31. Using the same principle as above, it is possible to replace the time–dependent matrix $\mathbf{H}_h$ with the average $\mathbf{H}$ of a number of available matrices:

$$\mathbf{H} = \frac{\sum_{m=1:M} \mathbf{H}_m}{M} \qquad (33)$$

where $\mathbf{H}_m$ is the matrix obtained from EKF during interval m and M is the total number of available matrices. The resulting matrix $\mathbf{H}$ can be then used to update the state covariance in Equation 31.

4.3 Unscented Kalman filter

The Unscented Kalman Filter (UKF) (Julier et al., 1995; Julier and Uhlmann, 1997; Wan et al., 2000; Wan and van der Merwe, 2000; van der Merwe et al., 2000) is an alternative filter for dynamic state–space models. The UKF uses a deterministic sampling approach (Unscented Transformation, UT) to represent a random variable using a number of deterministically selected sample points (often called *sigma points*). These points capture the mean and covariance of the random variable and, when propagated through the *true* nonlinear system, capture the posterior mean and covariance accurately to the second order (Taylor series expansion).

The Unscented Transformation is based on the intuitive expectation that "with a fixed number of parameters it should be easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function/transformation" (Julier & Uhlmann, 1996). Following this intuition, one would seek to find a parameterization that would capture the mean and covariance information while at the same time permitting the direct propagation of the information through an arbitrary set of nonlinear equations. This can be accomplished by generating a discrete distribution having the same first and second (and possibly higher) moments, where each point in the discrete approximation can be directly transformed. The mean and covariance of the transformed ensemble can then be computed as the estimate of the nonlinear transformation of the original distribution.

Given an n–dimensional distribution with covariance $\mathbf{P}$, it is possible to generate $O(n)$ points with the same sample covariance from the columns (or rows) of the matrices $\pm\sqrt{n\mathbf{P}}$ (the positive and negative roots). This set of points has a zero mean. However, simply adding the mean $\mathbf{X}$ of the original distribution to each of the points yields a symmetric set of $2n$ points with the desired mean and covariance. Because the set is symmetric its odd central moments are zero, so its first three moments match the original Gaussian distribution.

The main steps of the Unscented Transformation (UT) for calculating the statistics of a random variable that undergoes a nonlinear transformation (e.g. $\mathbf{y_h} = f(\mathbf{x_h})$) are presented in Algorithm 3 (Julier & Uhlmann, 1997). Let the n-dimensional random variable $\mathbf{x}_h$ with covariance matrix $\mathbf{P}_{x,h}$ denote the state for time interval h. Since this algorithm also considers the covariance of the measurement vector $\mathbf{P}_{y,h}$ during interval h and the covariance of the state and measurement vectors $\mathbf{P}_{xy,h}$, the covariance of the state vector will be denoted

$\mathbf{P}_{x,h} = \mathbf{P}_h$ in order to avoid confusion.

To calculate the statistics of $\mathbf{y}$, a matrix X is generated using $2n+1$ weighted *sigma points*. $\kappa \in R$ is a scaling parameter and $\left(\sqrt{(n+\kappa)\mathbf{P}_{x,h}} \right)_i$ is the ith row or column of the matrix square root of $(n+\kappa)$ $\mathbf{P}_{x,h}$. A Cholesky decomposition (Golub & van Loan, 1996) can be used for this step. The value of the scaling parameter κ has a direct effect on the scaling of the points and is an input to the algorithm. The constant α determines the spread of the sigma points around $\overline{\mathbf{X}}$ and is usually set to $0.0001 \leq \alpha \leq 1$. b is used to incorporate prior knowledge of the distribution of $\mathbf{X}$ (for Gaussian distributions, $b=2$ is optimal). The weights are not time–dependent and do not need to be recomputed for every time interval.

Algorithm 3. Unscented Transformation
Generation of *sigma points*

$$X_{0,h} = \mathbf{x_h} \tag{34}$$

for *i=1* to *n* **do**

$$X_{i,h} = \mathbf{x_h} + \left(\sqrt{(n+\kappa)\mathbf{P}_{x,h}} \right)_i \tag{35}$$

end for
for *i=n+1* to *2n* **do**

$$X_{i,h} = \mathbf{x_h} - \left(\sqrt{(n+\kappa)\mathbf{P}_{x,h}} \right)_i \tag{36}$$

end for
Generation of weights

$$W_0^m = \kappa / (n+\kappa) \tag{37}$$

$$W_0^c = \kappa / (n+\kappa) + (1 - a^2 + b) \tag{38}$$

for *i=1* to *2n* **do**

$$W_i^m = W_i^c = 1 / \left[2(n+\kappa) \right] \tag{39}$$

end for

The main steps of the UKF are presented in Algorithm 4 (van der Merwe et al., 2000). The initialization step uses the Unscented Transformation (Algorithm 3) to generate the $2n+1$ *sigma points* and appropriate weights for the mean and covariance computations. A time and measurement update step is repeated for each run of the algorithm.

The first step in the time update phase is the propagation of the *sigma points* through the transition equation (Equation 40). The prior estimate of the state vector is computed as a weighted sum of the propagated sigma points (Equation 41). A similar approach is used for the prior estimate of the state covariance (Equation 42). The true measurement equation is used to transform the sigma points into a vector of respective measurements (Equation 43). The measurement vector is computed as a weighted sum of the generated measurements (Equation 44).

The computation of the Kalman gain (and consequently the "correction" phase of the filtering) is based on the covariance of the measurement vector (Equation 45) and the covariance of the state and measurement vectors (Equation 46). These are computed using the weights (that were obtained from the Unscented Transformation during the initialization step) and the deviations of the sigma points from their means.

The Kalman gain is then computed from these covariance matrices (Equation 47). Equation 48 introduces the measurement vector $\mathbf{y}_h$ and uses the Kalman gain to correct the state estimate $\mathbf{x}_h$. The state covariance is updated using Equation 49.

Algorithm 4. Unscented Kalman Filter
for *h=1* to *N* **do**
Generate *sigma points* and weights using the Unscented Transformation (Algorithm 3)
Time update

$$X_{h|h-1} = \mathbf{f}(X_{h-1}) \tag{40}$$

$$\mathbf{x}_{h|h-1} = \sum_{i=0}^{2n} W_i^m X_{i,h|h-1} \tag{41}$$

$$\mathbf{P}_{x,h|h-1} = \sum_{i=0}^{2n} W_i^c \left(X_{i,h|h-1} - \mathbf{x}_{h|h-1} \right) \times \tag{42}$$

$$\times \left(X_{i,h|h-1} - \mathbf{x}_{h|h-1} \right)^T + \mathbf{Q}_h$$

$$Y_{i,h|h-1} = \mathbf{h}\left(X_{i,h|h-1} \right) \tag{43}$$

$$\mathbf{y}_{h|h-1} = \sum_{i=0}^{2n} W_i^m Y_{i,h|h-1} \tag{44}$$

Measurement update

$$\mathbf{P}_{y,h} = \sum_{i=0}^{2n} W_i^c \left(Y_{i,h|h-1} - \mathbf{y}_{h|h-1} \right) \times \tag{45}$$

$$\times \left(Y_{i,h|h-1} - \mathbf{y}_{h|h-1} \right)^T + \mathbf{R}_h$$

$$\mathbf{P}_{xy,h} = \sum_{i=0}^{2n} W_i^c \left(X_{i,h|h-1} - \mathbf{x}_{h|h-1} \right) \times \tag{46}$$

$$\times \left(Y_{i,h|h-1} - \mathbf{y}_{h|h-1} \right)^T$$

$$\mathbf{G}_h = \mathbf{P}_{xy,h} \mathbf{P}_{y,h}^{-1} \tag{47}$$

$$\mathbf{x}_h = \mathbf{x}_{h|h-1} + \mathbf{G}_h \left(\mathbf{y}_h - \mathbf{y}_{h|h-1} \right) \tag{48}$$

$$\mathbf{P}_{x,h} = \mathbf{P}_{x,h|h-1} - \mathbf{G}_h \mathbf{P}_{y,h} \mathbf{G}_h^T \tag{49}$$

end for

5. Application

The objective of this application is to demonstrate the performance of the joint on–line estimation of demand and supply parameters for a DTA system. The situation when only the demand parameters are calibrated on–line is used as the base. Since this reference problem is the usual OD estimation problem, GLS or Kalman Filter algorithms can be applied. When both the demand and supply parameters are jointly updated on–line, however, the problem cannot be represented analytically and the algorithms presented in Section 4 can be used. The on–line calibration has been implemented and demonstrated as it applies to the DynaMIT–R DTA system. Three algorithms have been implemented (EKF, LimEKF and UKF) and their performance for a freeway network in Southampton, UK, is presented.

DynaMIT–R is a state–of–the–art DTA system. The high–level framework of DynaMIT–R has been presented in Fig. 1. The key to the functionality of DynaMIT–R is its detailed network representation, coupled with models of traveler behavior. Through an effective integration of historical databases with real-time inputs, DynaMIT–R is designed to efficiently achieve:

- Real time estimation of network conditions.
- Rolling horizon predictions of network conditions in response to alternative traffic control measures and information dissemination strategies.
- Generation of traffic information and route guidance to steer drivers towards optimal decisions.

To sustain users' acceptance and achieve reliable predictions and credible guidance, DynaMIT–R incorporates unbiasedness and consistency into its core operations. Unbiasedness guarantees that the information provided to travelers is based on the best available knowledge of current and anticipated network conditions. Consistency ensures that DynaMIT–R's predictions of expected network conditions match what drivers would experience on the network. DynaMIT–R has the ability to trade-off level of detail (or resolution) and computational practicability, without compromising the integrity of its output. A more detailed description of DynaMIT-R can be found in Ben-Akiva et al. (2002).

The network includes a 35km long part of freeway (M27) from Southampton, U.K. The network starts to the west of the city of Southampton, then goes around it, and continues eastbound towards Portsmouth. The network also includes seven off–ramps and eight on–ramps. A schematic representation is shown in Fig. 2, which indicates the ten sensors, which provide traffic information (counts, speeds and occupancies). Traffic is loaded onto the network via twenty origin-destination pairs. The peak period for this direction is the afternoon/evening. Weekday data (speeds and densities) from the first two weeks of September 2001 have been used. Fig. 3 shows the speed and density distribution at a mainline sensor for these days.

Fig. 2. Schematic of the study network
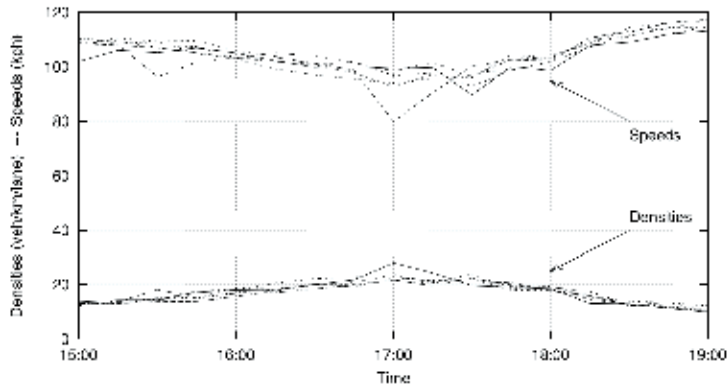
Fig. 3. Speeds/densities (days with dry weather conditions)

5.1 Methodology

Since an off–line calibration was not available for this network, the first step in this case study was to perform an off–line calibration. Data from five weekdays (without adverse weather or incidents) during the first two weeks of September 2001 were used and a sequential off–line calibration approach was followed. Supply parameters were first calibrated. Speed–density relationship parameters were obtained by fitting speed and density data to the appropriate functional form. Speed–density relationship parameters were obtained by using non–linear regression to fit speed and density data to the speed–density relationship used by DynaMIT–R:

$$u = u_f \left[1 - \left(\frac{\max(0, K - K_{min})}{K_{jam}} \right)^\beta \right]^\alpha \qquad (50)$$

where u denotes the speed, u_f is the free flow speed, K is the density, K_{min} is the minimum density, K_{jam} is the *jam* density and α and β are model parameters. The 45 segments of the network were split into three groups with homogeneous characteristics. The mainline segments were classified into two types, while the ramp segments were grouped together.

Segment capacities were estimated using Highway Capacity Manual guidelines (HCM, 2000). Capacity computations are usually based on appropriate guidelines (e.g. the Highway Capacity Manual, HCM, 2000, for the United States). Although the study network is in the United Kingdom, no equivalent national guidelines are available for the United Kingdom and therefore the Highway Capacity Manual guidelines were used. Analysis and comparison of the estimated capacities against observed counts were performed, to ensure

that the capacity values did not result in counterintuitive results (such as observed sensor counts exceeding the segment capacity).

Then, using the calibrated parameters as inputs, time–dependent OD flows were estimated. A sequential OD estimation approach (Balakrishna, 2002) was applied on five weekdays. A static seed matrix was used to initialize the process. For the first day, the estimated OD flows from each interval were used as historical estimates for the next interval. The estimated flows for each day were then used as historical flows for the next day. An ordinary least squares (OLS) approach was used for the first two days. At the end of the second day, measurement error covariances were estimates from the residuals of the fitted sensor counts and OD flows from their observed or historical values. This allowed for the use of a generalized least squares (GLS) approach for the remaining days. Estimated OD flows across time intervals were used to estimate autoregressive factors for the transition equation. The planning version of the DynaMIT system (DynaMIT–P) was used in this process. The Normalized Root Mean Square error (RMSN) statistic for the fit-to-counts was equal to 0.1232. The total fit of the speeds, as quantified by the Normalized Root Mean Square error (RMSN) statistic, was equal to 0.1102. The RMSN statistic was computed using the following formula (Ashok & Ben-Akiva, 2000, 2002):

$$RMSN = \frac{\sqrt{N \sum_N (y - \hat{y})^2}}{\sum_N y}$$

where N is the number of observations, y denotes an observation and $\hat{y}$ is the corresponding estimated (predicted) value.

One of the main outputs of DTA systems is traffic information and guidance, usually in the form of travel times. Speeds are the closest surveillance measurement and there are ways to compute travel times from speeds. Furthermore, given a properly calibrated traffic estimation and prediction system it is possible to obtain (simulated) travel times directly. On the other hand, the most ubiquitous traffic measurement is traffic counts. Therefore, the two first measures of effectiveness are based on fit of estimated (predicted) speeds and counts with observed values, quantified using the normalized root mean square error (RMSN).

The computational performance of the algorithms is another important consideration. In particular, given the *on–line* nature of the application, it is important to understand the computational complexity of each algorithm. Given a simulation–based DTA system, the function evaluations (required by the solution approaches) are by far the most computational intensive task, since each evaluation implies one run of the simulator. In the subsequent discussion, the number of function evaluations are used as a measure of effectiveness for each algorithm.

The state vector for the on–line calibration consists of OD flows, segment capacities and speed–density relationship parameters. (Route choice parameters are not included due to the nature of the network used for the case study.) The total dimension is 80, broken down in:

- 20 OD flows
- 45 segment capacities
- 15 speed–density relationship parameters: a speed–density relationship (and therefore 5 parameters: free flow speed, minimum and jam density and exponents α and β) has been defined for each of the three segment types.

The duration of the estimation and prediction intervals was set to fifteen minutes. OD estimation requires that the count measurements are aggregated to the duration of the interval (i.e. fifteen minutes in this case study). To maintain consistency between the various algorithms, this level of aggregation has been maintained for the counts in the on–line calibration approach. Furthermore, minute–by–minute speed and density surveillance information was used.

Therefore, the measurement vector for each fifteen–minute interval consisted of 390 elements:

- 15–minute count measurements: 10 count measurements in total
- Minute–by–minute speed measurements: 150 speed measurements
- Minute–by–minute density measurements: 150 density measurements
- *A priori* state vector: 80 elements comprising 20 OD flows, 45 capacities, and 15 speed–density relationship parameters (3 groups of 5 parameters each).

The transition fractions that were estimated during the off–line calibration were used for the OD flows. The degree of the autoregressive process for the OD flows was found to be equal to two. For the supply parameters, a degree of one was used for the autoregressive process. Furthermore, variance/covariance matrices were estimated based on the output of the off–line calibration (Ashok, 1996, Balakrishna et al., 2005).

The period of analysis comprises six intervals of fifteen minutes (i.e. from 16:15 to 17:45). A warm–up period of 75 minutes (from 15:00 to 16:15) is used to ensure that the network is adequately loaded.

5.2 Results

Tables 1 and 2 summarize the obtained fit for estimated and predicted speeds and counts. The *base* row provides the performance when only demand parameters are estimated on–line. The next three rows show the results obtained when demand and supply parameters are jointly estimated (each row corresponds to one of the considered algorithms). RMSN values are provided, as well as percent improvement over the base case; in particular, the results for the cases where demand and supply parameters are estimated jointly are shown as percent improvement over the *base* case (i.e. when only demand parameters are estimated on–line).

	Estimated		One-step pred.		Two-step pred.		Three-step pred.	
Algorithm	RMSN	(%)	RMSN	(%)	RMSN	(%)	RMSN	(%)
Base	0.1286	---	0.1540	---	0.1666	---	0.1905	---
EKF	0.1039	(19.2%)	0.1318	(14.4%)	0.1550	(7.0%)	0.2008	(-5.4%)
LimEKF	0.1091	(15.1%)	0.1321	(14.2%)	0.1702	(-2.2%)	0.2036	(-6.9%)
UKF	0.1293	(-0.6%)	0.1505	(2.3%)	0.1756	(-5.4%)	0.2221	(-16.6%)

Table 1. Case study results (counts)

These results indicate that the joint calibration of demand and supply parameters can improve the ability of the system to accurately estimate and predict the traffic conditions. The EKF algorithm exhibits the best performance with considerable improvements in estimation and prediction accuracy (except for three–step predicted counts).

	Estimated		One-step pred.		Two-step pred.		Three-step pred.	
Algorithm	RMSN	(%)	RMSN	(%)	RMSN	(%)	RMSN	(%)
Base	0.1266	---	0.1494	---	0.1448	---	0.1494	---
EKF	0.1107	(12.6%)	0.1209	(19.1%)	0.1303	(10.0%)	0.1331	(10.9%)
LimEKF	0.1121	(11.5%)	0.1249	(16.4%)	0.1346	(7.0%)	0.1362	(8.8%)
UKF	0.1156	(8.7%)	0.1249	(16.4%)	0.1315	(9.2%)	0.1346	(9.9%)

Table 2. Case study results (speeds)

A small decrease in performance (compared to the EKF) —but still an improvement— is obtained when the LimEKF algorithm is used. It is interesting to note that while the LimEKF algorithm has order(s) of magnitude lower computational complexity (than the EKF or UKF algorithms), it still provides a significant improvement over the base case. Improvements of more than 10% are obtained in estimated and one–step predicted speeds and counts. Furthermore, the LimEKF algorithm provides a 7% improvement in two–step predicted speeds and close to a 9% improvement in three–step predicted speeds. However, there is a small deterioration (-2.2%) in the two–step predicted counts and almost a 7% deterioration in three–step predicted counts.

The UKF algorithm seems to have the least desirable performance in this application. While in general this algorithm provides improvement over the base case, its two–step and three–step predicted counts deteriorate (-5.4% and -16.6). Furthermore, (with the exception of two–step and three–step predicted speeds) this algorithm is generally outperformed by the LimEKF, which has vastly better computation properties.

One observation that is applicable to all algorithms is that the approach shows a superior performance in prediction of speeds over prediction of counts. The source of this phenomenon may be traced to the surveillance data that have been used for this case study. In particular, sensor counts have been aggregated in 15-minute intervals, while minute–by–minute speeds and densities have been used (resulting in a larger number of observations and hence a larger "weight" on these measurements). Counts have been aggregated to 15–minute intervals in order to be as compatible and comparable as possible with the base case where only OD flows are estimated on–line. For more details on the OD estimation procedure used in DynaMIT, cf. Antoniou et al. (1997).

The third measure of effectiveness (besides the fit of speed and counts) is the computational complexity of each algorithm. As discussed above, the computational performance of the on–line calibration approach for a simulation–based DTA system is determined largely by the number of function evaluations required. The EKF and the UKF algorithms require *2n* and *2n+1* function evaluations (and thus runs of the simulator) respectively (where *n* is the dimension of the state vector). Therefore, the two algorithms have similar computational requirements (converging as the dimension of the problem increases).

The LimEKF algorithm, on the other hand, requires a single function evaluation irrespective of the dimension of the problem. Therefore, the computational performance of this algorithm is vastly superior to that of the other two algorithms (EKF and UKF). The *constant* computational requirements of the LimEKF algorithm (i.e. the fact that a single function evaluation is required irrespective of the application) makes it easy to obtain some further insight into the scalability of the approach to larger networks.

For example, assuming that an estimation interval of fifteen minutes is used, this approach can be used for any application (i.e. combination of DTA system and network) that allows for one function evaluation (for the on–line calibration) and another run of the simulator (for the prediction of the state using the on–line calibrated parameters within that time–frame). A related observation is that using the LimEKF and the on–line calibration approach to jointly estimate all inputs and model parameters is computationally equivalent to the base case (i.e. only using OD estimation to calibrate OD flows on–line).
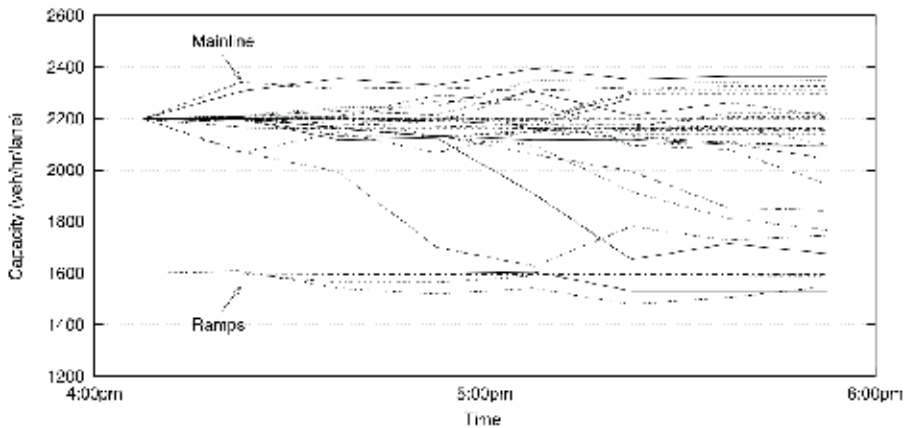
Fig. 4. Estimated capacities (EKF)

The impact of the on–line calibration on the parameters is discussed next, using the results from the EKF algorithm. Fig. 4 shows the estimated capacities for all segments over time. The capacity in several mainline segments has been increased (from the original 2200 vehicles per hour per lane), sometimes close to 2400 vehicles per hour per lane. A capacity of 2400 veh/hr/lane for a motorway such as M27 is reasonable. The capacity for a few mainline segments, which include weaving and/or merging, is reduced to less than 1800 vehicles per hour per lane. The location of these four segments with reduced on–line calibrated capacity is indicated in Fig. 5. The distribution of the other segment capacities is fairly uniform around the off–line calibrated mean capacity (obtained through the general guidelines of the HCM, which could not capture the subtle –geometrical or other– variations among the segments). On–line calibration, on the other hand, is able to estimate individual capacities for each segment, based on the traffic dynamics of these segments, thus capturing these variations. Ramp capacity is generally stable, with the exception of three (out of sixteen in total) ramp segments, in which small decreases (of less than 125 vehicles per hour per lane) are observed.
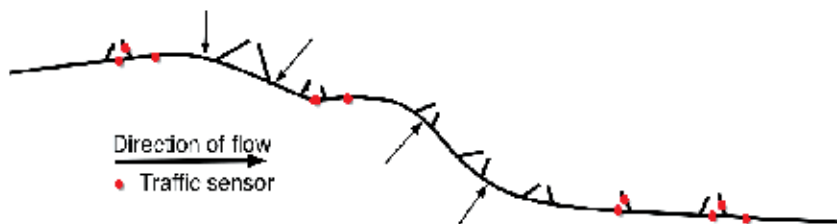
Fig. 5. Mainline segments with lower capacities (due to weaving and merging)

Fig. 6 and 7 show the evolution of the on–line calibrated speed–density relationships over time for mainline segments and ramp segments respectively. Each curve in these figures corresponds to a time interval, for which the on–line calibration was performed. One general observation is that on the day, for which the on–line calibration took place, the observed speeds and densities were higher than those observed during the off-line calibration (as indicated by the off–line calibrated speed–density relationship, also shown in these figures). The on–line calibrated speed–density relationships have changed in a way that captures this behavioral shift, indicating that the on–line calibration performs as intended. This shift is clearer for ramp segments (Fig. 7) and the second mainline segment group (Fig. 6(b)), while smaller changes (still in the right direction) are obtained for the other group of mainline segments (Fig. 6(a)).

(a) Group A (b) Group B
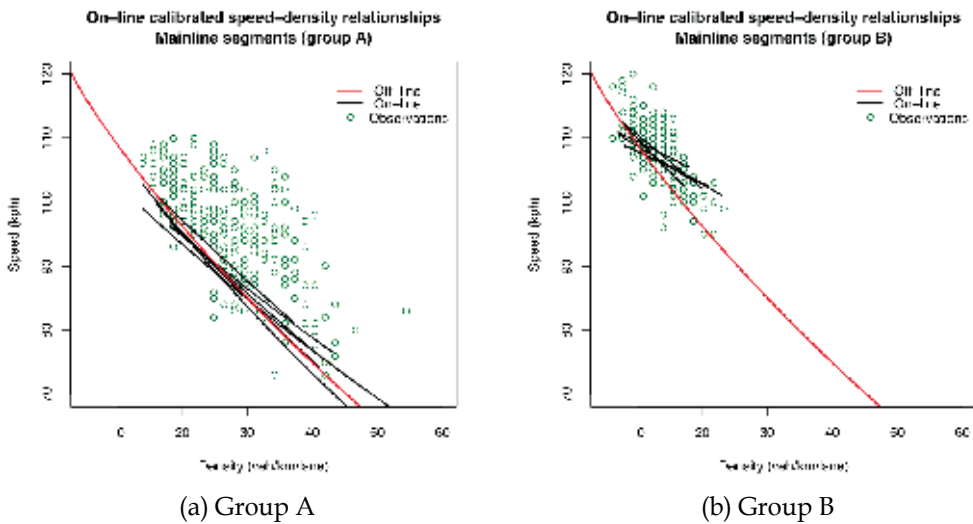
Fig. 6. On–line calibrated speed–density relationships (Mainline segments – EKF)
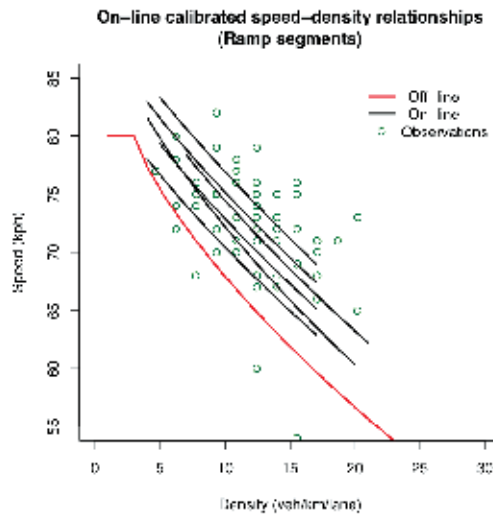
Fig. 7. On–line calibrated speed–density relationships (Ramp segments – EKF)

6. Conclusion

An on–line calibration approach for dynamic traffic assignment systems has been developed. The approach is general and flexible and makes no assumptions on the type of the DTA system, the models or the data that it can handle. Therefore, it is applicable to a wide variety of tools including simulation–based and analytical, as well as microscopic and macroscopic models.

The objective of the on–line calibration approach is to introduce a systematic procedure that will use the available data to *steer* the model parameters to values closer to the *realized* ones. The output of the on–line calibration is therefore a set of parameter values that —when used as input for traffic estimation and prediction— minimizes the discrepancy between the simulated (estimated and predicted) and the observed traffic conditions. The scope of the on–line calibration is neither to duplicate nor to substitute for the off–line calibration process. Instead, the two processes are complementary and synergistic in nature.

The on–line calibration problem is formulated as a state–space model. State–space models have been extensively studied and efficient algorithms have been developed, such as the Kalman Filter for linear models. Because of the non–linear nature of the on–line calibration formulation, modified Kalman Filter methodologies have been presented. The most straightforward extension is the Extended Kalman Filter (EKF), in which optimal quantities are approximated via first order Taylor series expansion (linearization) of the appropriate equations. The Limiting EKF is a variation of the EKF that eliminates the need to perform the most computationally intensive steps of the algorithm on–line. The use of the Limiting EKF provides dramatic improvements in terms of computational performance. The Unscented Kalman Filter (UKF) is an alternative filter that uses a deterministic sampling approach. The computational complexity of the UKF is of the same order as that of the EKF.

Empirical results suggest that joint on–line calibration of demand and supply parameters can improve estimation and prediction accuracy of a DTA system. While the results obtained from this real network application are promising, they should be validated in further empirical studies. In particular, the scalability of the approach to larger, more complex networks needs to be investigated.

The results also suggest that —in this application— the EKF has more desirable properties than the UKF (which may be expected to have superior performance over the EKF), while the UKF seems to perform better in terms of speeds than in terms of counts. Other researchers have also encountered situations where the UKF does not outperform the EKF, e.g. LaViola, J. J., Jr. (2003) and van Rhijn et al. (2005).

The Limiting EKF provides accuracy comparable to that of the best algorithm (EKF), while providing order(s) of magnitude improvement in computational performance. Furthermore, the LimEKF algorithm is that it requires a single function evaluation irrespective of the dimension of the state vector (while the computational complexity of the EKF and UKF algorithms increases proportionally with the state dimension). This property makes this an attractive algorithm for large–scale applications.

7. References

Antoniou, C., Ben-Akiva, M., Bierlaire, M., and Mishalani, R. (1997). Demand simulation for dynamic traffic assignment. In Proceedings of the 8th IFAC Symposium on Transportation Systems, Chania, Greece.

Antoniou, C., Ben-Akiva, M., and Koutsopoulos, H. N. (2004). Incorporating automated vehicle identification into origin–destination estimation. Transportation Research Record: Journal of the Transportation Research Board, 1882:37–44. Washington, D.C.

Antoniou, C., Ben-Akiva, M., and Koutsopoulos, H. N. (2005). On–line calibration of traffic prediction models. Transportation Research Record: Journal of the Transportation Research Board, 1934:235–245.

Ashok, K. (1996). Estimation and Prediction of time-dependent Origin-Destination Flows. PhD thesis, Massachusetts Institute of Technology.

Ashok, K. and Ben-Akiva, M. (1993). Transportation and Traffic Theory, chapter Daganzo, C., editor, Dynamic O-D matrix estimation and prediction for real-time traffic management systems, pages 465–484. Elsevier Science Publishing.

Ashok, K. and Ben-Akiva, M. (2000). Alternative approaches for real-time estimation and prediction of time-dependent origin–destination flows. Transportation Science, 34(1):21–36.

Ashok, K. and Ben-Akiva, M. (2002). Estimation and prediction of time-dependent origin-destination flows with a stochastic mapping to path flows and link flows. Transportation Science, 36(2):184–198.

Balakrishna, R. (2002). Calibration of the demand simulator in a dynamic traffic assignment system. Master's thesis, Massachusetts Institute of Technology.

Balakrishna, R., Koutsopoulos, H. N., and Ben-Akiva, M. (2005). H. S. Mahmassani, ed., 16th International Symposium on Transportation and Traffic Theory, chapter Calibration and validation of Dynamic Traffic Assignment systems, pages 407–426. University of Maryland, College Park. ISBN: 0-08-044680-9.

Ben Aissa, A., Sau, K., El Faouzi, N.-E., and de Moyzon, O. (2006). Sequential monte-carlo traffic estimation for intelligent transportation systems: Motorway travel time prediction application. In Proceedings of 2nd ISTS Symposium. Lausanne, Switzerland.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. N., and Mishalani, R. (2002). Gendreau, M. and Marcotte, P., editors, Transportation and network analysis: current trends, chapter Real-time simulation of traffic demand-supply interactions within DynaMIT, pages 19–36. Kluwer Academic Publishers. Miscellenea in honor of Michael Florian.

Ben-Akiva, M., DePalma, A., and Kaysi, I. (1991). Dynamic network models and driver information systems. Transportation Research A, 25(5):251–266.

Bierlaire, M. and Crittin, F. (2004). An efficient algorithm for real–time estimation and prediction of dynamic OD tables. Operations Research, 52(1):116–127.

Boel, R. and Mihaylova, L. (2006). A compositional stochastic model for real time freeway traffic simulation. Transportation Research Part B, 40:319–334.

Chui, C. K. and Chen, G. (1999). Kalman Filtering with Real-Time Applications. Springer-Verlag.

Crittin, F. (2003). New algorithmic methods for real–time transportation problems. PhD thesis, Ecole Polytechnique Federale de Lausanne.

Crittin, F. and Bierlaire, M. (2003). A generalization of secant methods for solving non-linear systems of equations. In Proceeding of the 3rd Swiss Transportation Research Conference, Ascona, Switzerland. http://www.strc.ch/Paper/crttin 6.pdf.

Doan, D. L., Ziliaskopoulos, A., and Mahmassani, H. (1999). On-line monitoring system for real-time traffic management applications. Transportation Research Record, 1678:142–149.

Gelb, A., editor (1974). Applied Optimal Estimation. M.I.T. Press.

Golub, G. H. and van Loan, C. F. (1996). Matrix Computations. Johns Hopkins Series in the Mathematical Sciences. Johns Hopkins University Press, 3rd edition.

HCM (2000). Highway Capacity Manual. Transportation Research Board, Washington, D.C.

He, R., Miaou, S., Ran, B., and Lan, C. (1999). Developing an on-line calibration process for an analytical dynamic traffic assignment model. In Proceedings of the 78th Annual Meeting of the Transportation Research Board, Washington, D.C.

Huynh, N., Mahmassani, H., and Tavana, H. (2002). Adaptive speed estimation using transfer function models for real-time dynamic traffic assignment operation. In Proceedings of the 81st Annual Meeting of the Transportation Research Board.

Julier, S. and Uhlmann, J. (1997). A new extension of the Kalman filter to nonlinear systems. In Proceedings of Aerosense: The 11th International Symposium of Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL., volume Multi Sensor Fusion, Tracking and Resource Management II.

Julier, S. and Uhlmann, J. (1996). A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, OX1 3PJ United Kingdom.

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. (1995). A new approach for filtering nonlinear systems. In Proceedings of the American Control Conference, pages 1628–1632.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering (ASME), 82D:35–45.

LaViola, J. J., Jr. (2003). A comparison of unscented and extended Kalman filtering for estimating quaternion motion. In Proceedings of the 2003 American Control Conference, pages 2435–2440.

Liu, H., van Lint, H., van Zuylen, H., and Zhang, K. (2006). Two distinct ways of using Kalman filters to predict urban arterial travel time. In Proceedings of the IEEE Intelligent Transport Systems Conference (ITSC), pages 845–850, Toronto, Canada.

Mahmassani, H. S. (2001). Dynamic network traffic assignment and simulation methodology for advanced system management applications. Networks and Spatial Economics, 1(3):267–292.

Peeta, S. and Bulusu, S. (1999). Generalized singular value decomposition approach for consistent on–line dynamic traffic assignment. Transportation Research Record, 1667:77–87.

Peeta, S. and Yu, J. W. (2004). Adaptability of a hybrid route choice model to incorporating driver behavior dynamics under information provision. IEEE Transactions on Systems, Man and Cybernetics — Part A: Systems and Humans, 34(2):243–256.

Peeta, S. and Yu, J. W. (2005). A hybrid model for driver route choice incorporating en–route attributes and real–time information effects. Networks and Spatial Economics, 5(1):21–40.

Peeta, S. and Yu, J. W. (2006). Behavior–based consistency–seeking models as deployment alternatives to dynamic traffic assignment models. Transportation Research Part C, 14:114–138.

Qin, X. and Mahmassani, H. (2004). Adaptive calibration of dynamic speed-density relations for online network traffic estimation and prediction applications. In Proceedings of the 83rd Annual Meeting of the Transportation Research Board, Washington, D.C.

Sorenson, H. W., editor (1985). Kalman Filtering: Theory and Application. IEEE Press, New York.

Tavana, H. and Mahmassani, H. (2000). Estimation and application of dynamic speed-density relations by using transfer function models. Transportation Research Record, 1710:47–57.

van Arem, B. and van der Vlist, M. J. M. (1992). An on–line procedure for estimating current capacity. Technical Report INRO-VVG 1991-17, TNO Institute of Spatial Organization, Delft.

van der Merwe, R., Doucet, A., de Freitas, N., and Wan, E. (2000). The unscented particle filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department.

van Lint, H., Hoogendoorn, S. P., and van Zuylen, H. J. (2002). State space neural networks for freeway travel time prediction. Artificial Neural networks - ICANN 2002 Lecture Notes in Computer Science, 2415:1043–1048.

van Lint, J., Hoogendoorn, S. P., and van Zuylen, H. J. (2005). Accurate freeway travel time prediction with state–space neural networks under missing data. Transportation Research Part C–Emerging Technologies, 13(5–6):347–369.

van Lint, J. W. C. (2006). Incremental and online learning through extended Kalman filtering with constraint weights for freeway travel time prediction. In Proceedings of the IEEE Intelligent Transport Systems Conference (ITSC), pages 1041–1046, Toronto, Canada.

van Rhijn, A., van Liere, R., and Mulder, J. (2005). An analysis of orientation prediction and filtering methods for VR/AR. In Proceedings of the IEEE Virtual Reality Conference 2005, pages 67–74.

Wan, E. and van der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In Proceedings of IEEE Symposium (AS-SPCC). Lake Louise, Alberta, Canada.

Wan, E., van der Merwe, R., and Nelson, A. T. (2000). Dual estimation and the unscented transformation. In Solla, S. A., Leen, T. K., and Mueller, K.-R., editors, Advances in Neural Information Processing Systems 12, pages 666–672. MIT Press.

Wang, Y. and Papageorgiou, M. (2005). Real–time freeway traffic state estimation based on extended Kalman filter: A general approach. Transportation Research Part B, 39:141–167.

Wang, Y., Papageorgiou, M., and Messmer, A. (2007). Real–time freeway traffic state estimation based on extended Kalman filter: A case study. Transportation Science, 41:167–181.

Zhou, X. and Mahmassani, H. S. (2004). A structural state space model for real–time origin–destination demand estimation and prediction in a day–to–day updating framework. In Proceedings of the 83rd Annual Meeting of the Transportation Research Board, Washington, D.C.

Kalman Filter in Control and Modeling

Mouna Ghanai[1] and Kheireddine Chafaa[2]
[1]University of Batna, Electronics Department, 05000, Batna
[2]University of M'sila, Electronics Department, 28000, M'sila,
Algeria

1. Introduction

Originaly, a filter is a physical device for removing unwanted components of mixtures (gas, liquid, solid). In the area of telecommunications, signals are mixtures of different frequencies, and the term of filter is used to describe the attenuation of the unwanted frequencies. Since 1940, the concept of a filter was extended to the separation of signals from noise. With Kalman filter, the meaning of filter is well beyond the notion of separation. It also includes the solution of an inversion problem, in which one knows how to represent the measurable variables as functions of variables of principle interest.

Least squares method proposed by Carl Friedrich Gauss in 1795 was the first method for forming an optimal estimate from noisy data, and it provides an important connection between the experimental and theoretical sciences.

Before Kalman, Norbert Wiener proposed his famous filter called Wiener filter which was restricted only to stationary scalar signals and noises, the solution obtained by this filter is not recursive and needs the storing of the entire pas observed data.

Kalman filter is a generalization of Wiener filter. The significance of this filter is in its ability to accommodate vector signals and noises which may be non stationary. The solution is recursive in that each update estimate of the state is computed from the previous estimate and the new input data, so, contrary to Wiener filter, only the previous estimate requires storage, so Kalman filter eliminate the need for storing the entire pas observed data.

In this chapter, we present two important applications of Kalman filter. In the first one we show how this filter can be used as an adaptive controller system (Chafaa et al., 2006). Studies proposed in this part illustrate a structure for the control of a positional system towards a mobile target in a three dimensional space (see Fig.1). In the presence of a random disturbances (white noise) or when few system parameters change, the use of an adaptive and optimal controller turns out necessary (Mudi & Nikhil, 1999; Zdzislaw, 2005). In this case we are choosing to use Kalman filter as a controller. This technique is based on the theory of Kalman's filtering (Kalman, 1960; Eubank, 2006), it transforms Kalman's filter into a Kalman controller.

In the second application we give the use of such filter in estimating the membership functions of fuzzy sets in order to obtain a fuzzy model (Chafaa et al., 2007). Fuzzy modelling is an effective tool for the approximation of nonlinear systems. Takagi-Sugeno (TS) model is widely used fuzzy modeling technique (Takagi & Sugeno, 1986; Angelov & Filev, 2004). The TS model utilizes the idea of linearization in a fuzzily defined region of the state space. Due to the fuzzy regions, the nonlinear system is decomposed into a multi-

model structure consisting of linear models that are not necessarily independent (Johansen & Babuska, 2003). A TS fuzzy model is usually constructed in two steps: Step 1: Determine the membership functions of the antecedents; Step 2: Estimate the parameters of the consequent functions.

One of the most techniques used to release the first step is the fuzzy clustering in the Cartesian product-space of the inputs and outputs (Babuska & Verbruggen, 1995; Babuska & Verbruggen, 1997; Bezdek & Dunn, 1975). As the consequent functions are usually chosen to be linear in their parameters, the second step is done by standard linear least-squares methods (Babuska & Verbruggen, 1997; Babuska et al., 1998).

Many clustering algorithms can be found in the literature, they are based on the optimization of fuzzy C-means functional (Nascimento et al., 2003). Some of them utilize an Euclidian distance norm (Bezdek et al., 1987; Hathaway & Bezdek, 1991) in which the detected clusters have an hyperspherical shapes, i.e., clusters whose surfaces of constant membership are hyperspheres. Others extend the Euclidian distance norm to an adaptive distance norm (Bezdek & Dunn, 1975; Gustafson & Kessel, 1997; Gath & Geva, 1998) in order to detect clusters of different geometrical shapes in one data set.
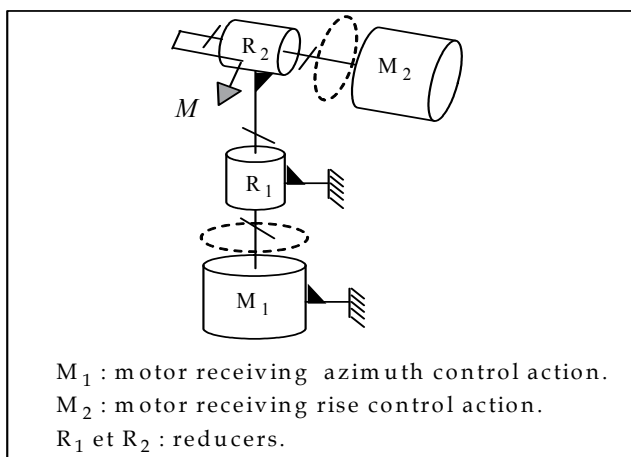
Fig. 1. Positioning system

Fuzzy clustering in the Cartesian product-space of the inputs and the outputs has been extensively used to obtain the antecedent membership functions (Babuska & Verbruggen, 1997; Babuska, 1998; Sugeno & Yasukawa, 1993). Attractive features of this approach are the simultaneous identification of the antecedent membership functions along with the consequent local linear models and the implicit regularization (Johansen& Babuska, 2003).

By clustering in the product-space, multidimensional fuzzy sets are initially obtained, which are either used in the model directly or after projection onto the individual antecedent variables (regressors). As it is generally difficult to interpret multidimensional fuzzy sets, projected one-dimensional fuzzy sets are usually preferred.

Babuska and Verbruggen (Babuska & Verbruggen, 1997) proposed a fuzzy modeling scheme based on Gustafson-Kessel clustering algorithm (GKCA) to estimate the premise membership functions and on least-squares method to estimate the parameters of the consequence functions. Abony et al (Abonyi et al., 2002) proposed to use the Gath-Geva

(GG) clustering algorithm instead of GKCA method, because with GG method, the parameters of the univariate membership functions can directly be derived from the parameters of the clusters.

In this part, a fuzzy modeling algorithm combining GKCA and Kalman filter (KF) is proposed (Chafaa et al., 2007). We use GKCA in order to detect clusters of different geometrical shapes in the data set and to obtain the point-wise membership functions of the premise. After that a Kalman filter is introduced to estimate the parameters of the premise membership functions and those of the consequence functions. In the premise part, the membership functions are triangular functions, then Kalman filter will estimate the parameters of a straight line functions by using the data corresponding to the premise membership functions defined point-wise, but in the consequence part, Kalman filter will be used as a linear regression to estimate the parameters of the TS fuzzy model using the input-output data set.

2. Kalman controller for target tracking system

2.1 Target tracking system

A target tracking system is a system for which inputs are the azimuth and rise, and outputs are the control actions for locating the motors. The target moves through azimuth-rise space. Two dc-motors adjust the platform position constantly towards the target (Chafaa et al., 2006; Brookner, 1998). The platform can be any directional system which can turn up exactly towards the target; such system can be a Laser, a video camera or an antenna. We suppose that we have a potentiometric system which can discover the direction of the platform towards the target (Ogata, 1970). The Radar sends azimuth and rise coordinates to the target tracking system in the end of every time interval, we calculate the current error and its variation in the platform position. Then, a Kalman controller determines the control actions for dc-motors, one action for azimuth motor and the other one for rise motor. These actions are going to reposition the platform as shown in Fig. 2. We can control independently the azimuth and rise positions by applying the same algorithm twice, it facilitates us calculations.
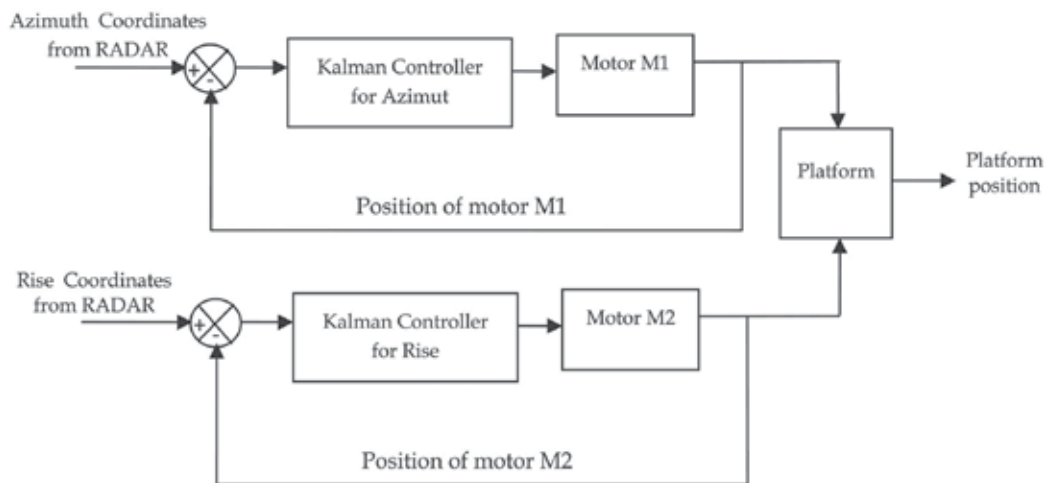
Fig. 2. Control system

2.2 Kalman controller

The Kalman filter is used for estimating or predicting the next stage of a system based on a moving average of measurements driven by white noise, which is completely unpredictable. It needs a model of the relationship between inputs and outputs to provide feedback signals but it can follow changes in noise statistics quite well. The Kalman filter is an optimum estimator that estimates the state of a linear system developing dynamically through time. An optimum estimator can be defined as an algorithm that processes all the available data to yield an estimate of the "state" of a system whilst at the same time estimating some predefined optimality criterion.

In this section we will conceive another type of controllers called "Kalman Controller" or "Kalman Filter controller". This technique consists to achieve a one-dimensional Kalman Filter acting as an alternative controller, i.e., it can provides the control actions to the dc-motor in addition to its filtering function (Kosko, 1992).

In the discrete state space formulation, the state and measurement equations for the controllers are given by:

$$x_{k+1} = Gx_k + Hc_k + w_k$$
$$z_k = Cx_k + v_k$$

$$(1)$$

Our proposed control structure contains two Kalman controllers, one for azimuth (Azimuth controller) and another for rise (rise controller). Since the two controllers act independently, so we can assume them to have one-dimensional models such that :

$$G = H = C = 1 \qquad (2)$$

Since the state is a control action, so we can take the input c_k to be :

$$c_k = e_k + \dot{e}_k \qquad (3)$$

Let x_{k+1} denotes the control action necessary at the moment k to exactly lock onto the target at the moment $k+1$. Then, the controller output at the moment k will be considered equal the prediction $u_k = \hat{x}_{k+1/k}$

Let us note that:

$$e_k = x_k - \hat{x}_{k/k-1}$$
$$\dot{e}_k = e_k - e_{k-1}$$

$$(4)$$

By substitution of 2 and 3 in 1 we obtain the new state equation:

$$x_{k+1} = x_k + e_k + \dot{e}_k + w_k \qquad (5)$$

where w_k represents a white noise that models target acceleration or other unmodeled effects. The new equation of measurements is

$$z_k = x_k + v_k$$
$$= \hat{x}_{k/k-1} + e_k + v_k$$
$$= \hat{x}_{k/k-1} + v'_k$$

$$(6)$$
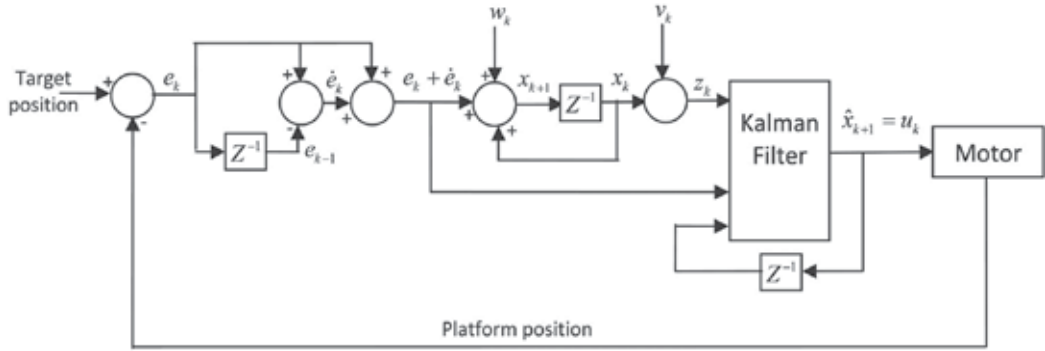
Fig. 3. Control system

Since we assume that e_k and v_k are uncorrelated, the variance of v'_k is:

$$
\begin{aligned}
R'_k &= E\left[v'^2_k\right] \\
&= E\left[\left(e_k + v_k\right)^2\right] \\
&= E\left[e^2_k\right] + E\left[v^2_k\right] \\
&= P_{k/k-1} + R_k
\end{aligned}
\tag{7}
$$

The recursive equations of Kalman Filter take the following general form:

$$
\begin{aligned}
\hat{x}_{k+1/k} &= G\hat{x}_{k/k} + HU_k \\
P_{k/k-1} &= GP_{k-1/k-1}G^T + Q_{k-1} \\
K_k &= P_{k/k-1}C^T\left[CP_{k/k-1}C^T + R_k\right]^{-1} \\
\hat{x}_{k/k} &= \hat{x}_{k/k-1} + K_k\left[z_k - C\hat{x}_{k/k-1}\right] \\
P_{k/k} &= P_{k/k-1} - K_k CP_{k/k-1}
\end{aligned}
\tag{8}
$$

To obtain the one–dimensional Kalman controller, we substitute 2, 3, 6 and 7 in 8, and then we obtain it under the following form:

$$
\begin{aligned}
u_k &= \hat{x}_{k/k} + e_k + \dot{e}_k \\
P_{k/k-1} &= P_{k-1/k-1} + Q_{k-1} \\
K_k &= \frac{P_{k/k-1}}{R'_k} \\
\hat{x}_{k/k} &= u_{k-1} + K_k v'_k \\
P_{k/k} &= \left[1 - K_k\right]P_{k/k-1}
\end{aligned}
\tag{9}
$$

Figure 3 illustrates the detailed structure of Kalman controller. The simulation results of this controller are presented in figures 4 and 5, where we see in figure 4 that the load disturbance is rejected and in figure 5 the tracking is carried out.
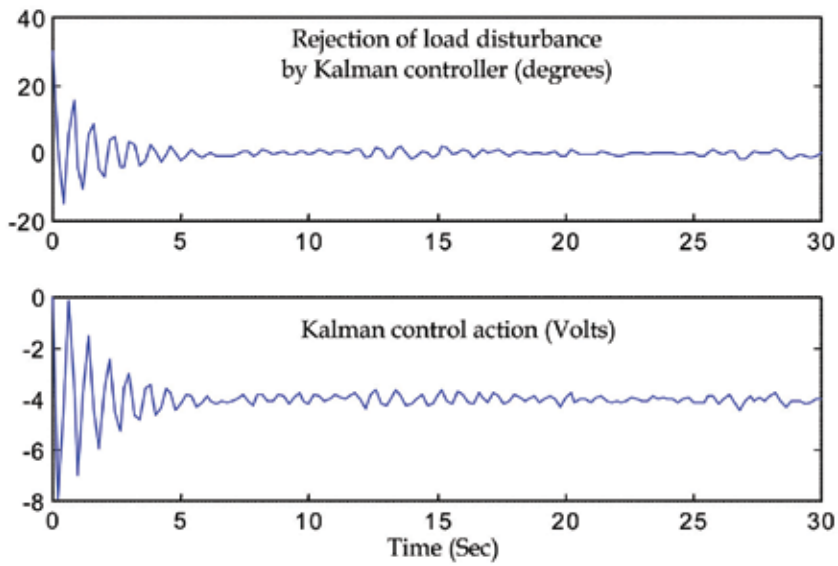
Fig. 4. System response and Kalman control action to a strong load disturbance
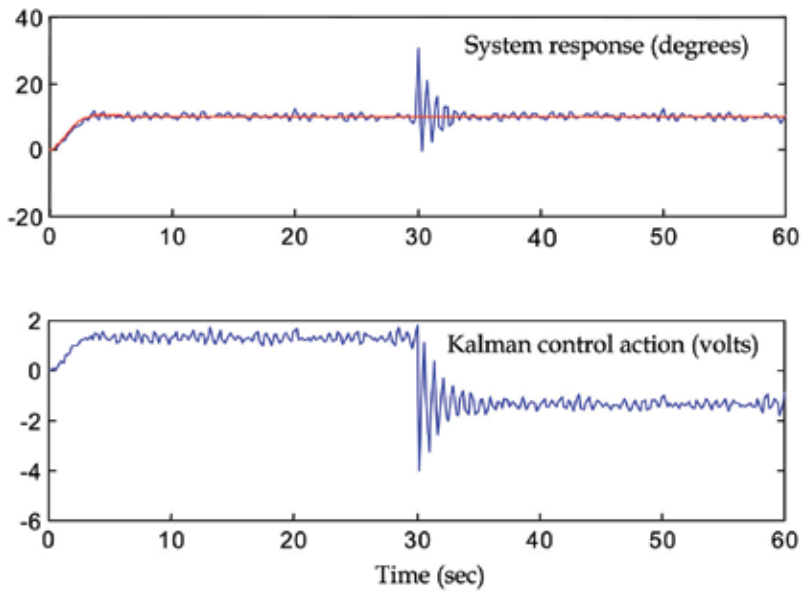
Fig. 5. System response and Kalman control action to a step input and a strong load disturbance

3. Fuzzy modeling and identification

3.1 Takagi-Sugeno fuzzy models

The TS fuzzy model can represent or model any unknown nonlinear system $y = f(\boldsymbol{x})$, based on some available input-output data $\boldsymbol{x}_k = \begin{bmatrix} x_{1k}, x_{2k}, ..., x_{nk} \end{bmatrix}^T$ and y_k. The index k denotes the individual data samples and n the number of regressors.

In the TS fuzzy model, the rule consequents are crisp functions of the model inputs:

$$R_i : IF \ \boldsymbol{x} \ is \ A_i(\boldsymbol{x}) \ THEN \ y_i = \boldsymbol{a}_i^T \boldsymbol{x} + b_i \qquad i = 1, 2, \dots c \tag{10}$$

Where $\boldsymbol{x}$ is $n \times 1$ input variable, $y_i \in R$ is the output variable. $n \times 1$ vector $\boldsymbol{a}_i$ and $b_i \in R$ are the TS parameters. R_i denotes the ith rule and c is the number of rules in the rule base. A_i is the premise multivariable membership function of the ith rule. $\boldsymbol{\theta}_i = \begin{bmatrix} \boldsymbol{a}_i & b_i \end{bmatrix}^T$ is the parameter vector of the ith rule.

The premise proposition " $\boldsymbol{x} \ is \ A_i(\boldsymbol{x})$ " can be expressed as a logical combination of propositions with univariate fuzzy sets defined for the individual components of $\boldsymbol{x}$, usually in the following conjunctive form (Kukolj & Levi, 2004) :

$$R_i : IF \ x_1 \ is \ A_{i1}(x_1) \ AND \dots\dots AND \ x_n \ is \ A_{in}(x_n) \ THEN \ y_i = \boldsymbol{a}_i^T \boldsymbol{x} + b_i \qquad i = 1, 2, \dots c \tag{11}$$

the degree of fulfilment of the rule is calculated as the product of the individual membership degrees :

$$\beta_i(\boldsymbol{x}) = \prod_{j=1}^{n} \mu_{A_{ij}}(\boldsymbol{x}) \tag{12}$$

where $\mu_{A_{ij}}(\boldsymbol{x})$ is the membership function of the fuzzy set A_{ij} .

The inference is reduced to the fuzzy-mean defuzzification formula (Takagi & Sugeno, 1986 ; Kukolj & Levi, 2004) :

$$y = \frac{\sum_{i=1}^{c} \beta_i(\boldsymbol{x}) \left(\boldsymbol{a}_i^T \boldsymbol{x} + b_i \right)}{\sum_{i=1}^{c} \beta_i(\boldsymbol{x})} \tag{13}$$

From (11) and (13), it is noted that TS fuzzy model approximates a nonlinear system with a combination of several linear systems by decomposing fuzzily the whole input space into several partial spaces and representing each input-output space with each linear equation.

3.2 New fuzzy modeling algorithm

The structure of the proposed algorithm is presented in Fig. 6. The identification algorithm proceeds in three steps:

1. from the input-output sequences $\{(x_k, y_k)\}_{k=1}^{N}$, partition the data into a set of local linear submodels by using GKCA in the product space $\boldsymbol{X} \times \boldsymbol{Y}$

2. Obtain the membership functions for the premise variables by using cluster projections and Kalman filtering.

3. Estimate the consequent parameters by Kalman filter algorithm.

The three procedures are repeated to find the appropriate number of clusters c as shown in Fig. 6 in which the performance index used is the mean squared error (MSE), so when MSE $\leq \varepsilon$ the loop is stopped and the optimal c is obtained.
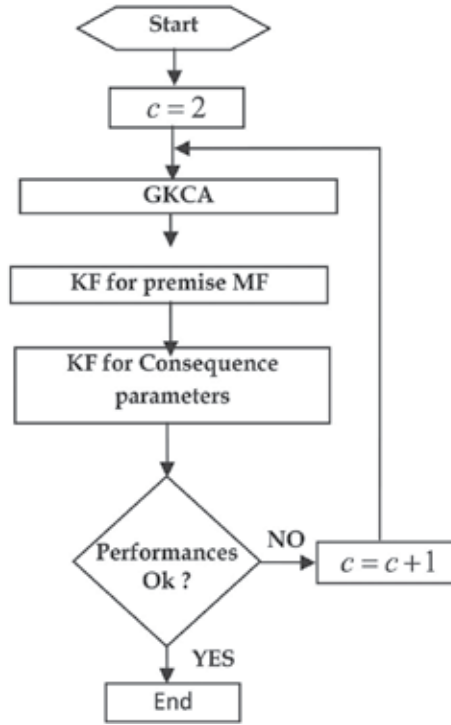
Fig. 6. Structure of the proposed fuzzy modeling

A. Fuzzy clustering
Clustering of numerical data forms the basis of many classification and system modeling algorithms (Bezdek & Dunn, 1975; Babuska et al., 1998). The purpose of clustering is to distil natural grouping of data from a large data set, producing a concise representation of a system's behavior. In particular, The GKCA has been widely studied and applied by many researchers (Bezdek et al., 1987; Hathaway & Bezdek, 1991). The GKCA is an iterative optimization algorithm that minimizes the cost function:

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ik}^{m} (z_k - v_i)^T M_i (z_k - v_i) \qquad (14)$$

where N is the number of data points, c is the number of clusters, z_k is the k'th data point, v_i is the i'th cluster center, μ_{ik} is the degree of membership of the k'th data in the i'th cluster, M_i is the norm-inducing matrix of the i'th cluster and m is a weighting exponent which determines the fuzziness of the resulting clusters (typically $m = 2$).
In this work, the GKCA is applied in order to obtain the fuzzy partition matrix $U = \left[\mu_{ik} \right]_{c \times N}$, with $\mu_{ik} \in \begin{bmatrix} 0 & 1 \end{bmatrix}$ is a membership degree.

B. Premise membership functions
The premise membership functions can be obtained from the results of fuzzy clustering by projecting the fuzzy sets defined point-wise in the partition matrix onto the premise variables x_j, $1 \le j \le n$. The TS rules are then expressed in the conjunctive form as in (11).

In order to obtain membership functions for the premise fuzzy sets $A_{ij}, 1 \leq i \leq c, 1 \leq j \leq n$, the multidimensional fuzzy set defined point-wise in the ith row of the partition matrix is projected onto the regressors x_j. Note that the resulting membership functions are defined point-wise, for the identification data only and may be nonconvex, which is caused by the probabilistic constraint in most fuzzy clustering algorithms and by the noise in the data (Babuska et al., 1998).

In order to obtain a prediction model or a model suitable for control purposes, the premise membership functions must be expressed in a form that allows computation of the membership degrees, also for input data not contained in the data set. To achieve this step, we propose to use Kalman filter to approximate the point-wise defined membership functions by some suitable straight line functions (triangular functions) as depicted in Fig. 7.

The Kalman filtering process is a recursive minimum mean square estimation procedure (Kalman, 1960; Mohinder & Angus, 2001). Each update estimate of the parameter vector corresponding to a straight line equation is computed from the previous estimate and the new input data (here the input data are the point-wise values of the membership functions). In this sense we propose to use Kalman filter as a linear regression as follows : Consider $2cn$ sets, each set represent the linear part of the point-wise set of a certain premise membership function. The linear part is obtained by taking the α-cut of the considered membership function. So we obtain $2cn$ parameter vectors (one for each set). In each set we will have N_j data (samples), where j denotes the jth set.

Then each set can be modeled by the following measurement equation:

$$
\begin{aligned}
y_{k_j}^j &= a^j x_{k_j} + b^j + v_{k_j} \qquad j = 1, 2, ..., 2cn \\
&= \begin{bmatrix} x_{k_j} & 1 \end{bmatrix} \begin{bmatrix} a^j \\ b^j \end{bmatrix} + v_{k_j} \qquad k_j = 1, 2, ..., N_j \\
&= C_{k_j}^j \theta_{k_j}^j + v_{k_j}
\end{aligned} \qquad (15)
$$

where $C_{k_j}^j$ is the observation vector at the moment k_j , $\theta_{k_j}^j = \begin{bmatrix} a^j & b^j \end{bmatrix}^T$ is the parameter vector, v_{k_j} is the measurement noise, N_j is the number of data (samples) in the jth set and the superscript j denotes the jth straight line regression. For simplicity, we will denote k_j by k . From equation (15), θ_k^j will be considered as a state variable, so the state equation will be

$$
\theta_k^j = A^j \theta_{k-1}^j + w_{k-1}^j \qquad j = 1, 2, ..., 2cn \qquad (16)
$$

where A^j is an 2×2 state transition matrix, and w_k^j is the state noise and θ_k^j is the value of the state variable at the moment k .

The state noise and the measurement noise are assumed to be statistically independent (Haykin, 2001) and can be modeled as zero mean, white noise processes whose covariances are given as
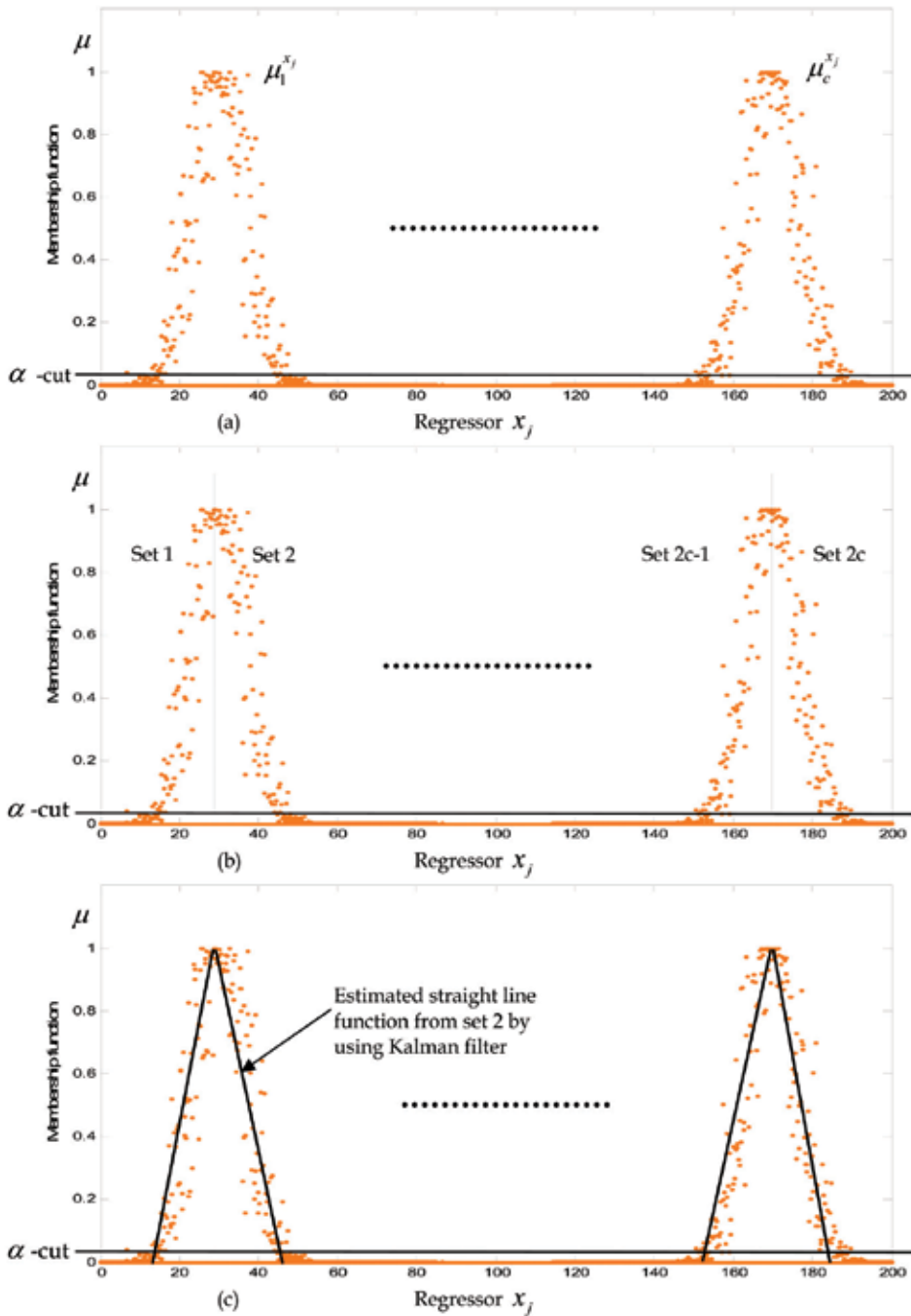
Fig. 7. Determination of premise membership functions :
(a) Premiss MF defined point-wise for the regressors x_j, j=1,2,...n
(b) Division of each membership function into 2 sets (2c sets are obtained for each x_j and 2cn sets are   obtained for all regressors)
(c) Approximation of each set by a straight line function by using Kalman filter

$$E\left[w_i^T w_j \right] = \begin{cases} Q & i = j \\ 0 & i \neq j \end{cases}$$

$$E\left[v_i v_j \right] = \begin{cases} r & i = j \\ 0 & i \neq j \end{cases} \qquad (17)$$

$$E\left[w_i v_j \right] = 0 \qquad \forall i, j$$

By recurrence proceeding, the update state equation and the predicted measure will be given by the following equations:

$$\hat{\boldsymbol{\theta}}_{k/k-1}^{j} = A \hat{\boldsymbol{\theta}}_{k-1/k-1}^{j} \qquad (18)$$

$$\hat{y}_k^j = \boldsymbol{C}_k^j \hat{\boldsymbol{\theta}}_{k/k-1}^{j} \qquad (19)$$

Now that the model representation of the parameter vectors is complete, the training of the parameters via Kalman filter technique is in order. The update of the parameters is according to the following recursion:

$$\hat{\boldsymbol{\theta}}_{k/k}^{j} = \hat{\boldsymbol{\theta}}_{k/k-1}^{j} + \boldsymbol{K}_k^j \left(y_k^j - \boldsymbol{C}_k^j \hat{\boldsymbol{\theta}}_{k/k-1}^{j} \right) \qquad (20)$$

where $\boldsymbol{K}_k$ is the computed Kalman gain. The computed Kalman gain can be viewed as an adaptive learning rate (Tzeng et al., 1994) ant its computation is according to the following steps :

$$\boldsymbol{K}_k^j = \boldsymbol{P}_{k/k-1}^{j} \boldsymbol{C}_k^{jT} \left(\boldsymbol{C}_k^j \boldsymbol{P}_{k/k-1}^{j} \boldsymbol{C}_k^{jT} + r \right)^{-1} \qquad (21)$$

$$\boldsymbol{P}_{k/k-1}^{j} = A^j \boldsymbol{P}_{k-1/k-1}^{j} A^{jT} + Q \qquad (22)$$

$$\boldsymbol{P}_{k/k}^{j} = \boldsymbol{P}_{k/k-1}^{j} - \boldsymbol{K}_k^j \boldsymbol{C}_k^j \boldsymbol{P}_{k/k-1}^{j} \qquad (23)$$

where $\boldsymbol{P}_{k/k-1}^{j} = E\left[\left(\boldsymbol{\theta}_k^j - \hat{\boldsymbol{\theta}}_{k/k-1}^{j} \right) \left(\boldsymbol{\theta}_k^j - \hat{\boldsymbol{\theta}}_{k/k-1}^{j} \right)^T \right]$ and $\boldsymbol{P}_{k/k}^{j} = E\left[\left(\boldsymbol{\theta}_k^j - \hat{\boldsymbol{\theta}}_{k/k}^{j} \right) \left(\boldsymbol{\theta}_k^j - \hat{\boldsymbol{\theta}}_{k/k}^{j} \right)^T \right]$

are the one step predicted and filter estimate error covariance matrices, respectively.
To simplify the implementation of the Kalman filtering technique, we assume that $A^j = \boldsymbol{I}$ where $\boldsymbol{I}$ is a unit matrix ; Q and r are an assigned variances of the process noise and measurement noise, respectively. The initial parameters values are set to be random numbers.

C. Estimating consequent parameters
There are several methods to obtain the consequent parameters (Angelov & Filev, 2004; Babuska & Verbruggen, 1997; Abonyi et al., 2002). In part we propose an algorithm based also on KF that can compute directly the consequent parameters from the data set and the estimated premise membership functions.
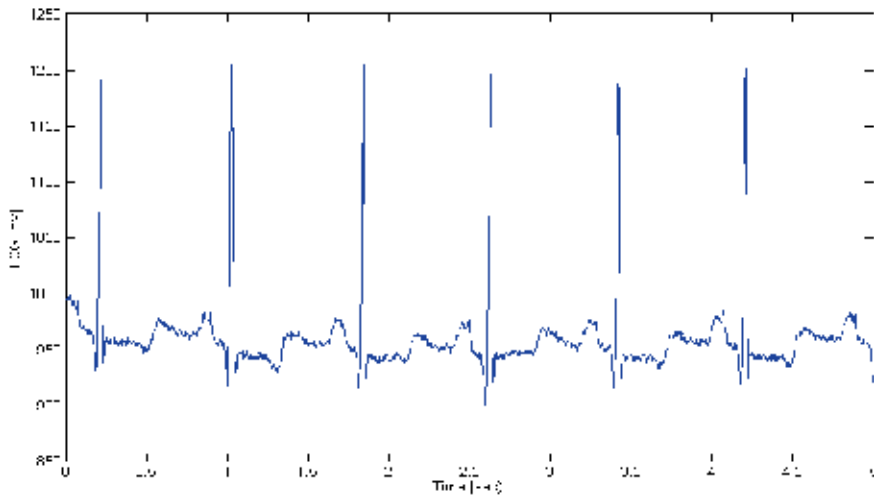
Fig. 8. ECG signal
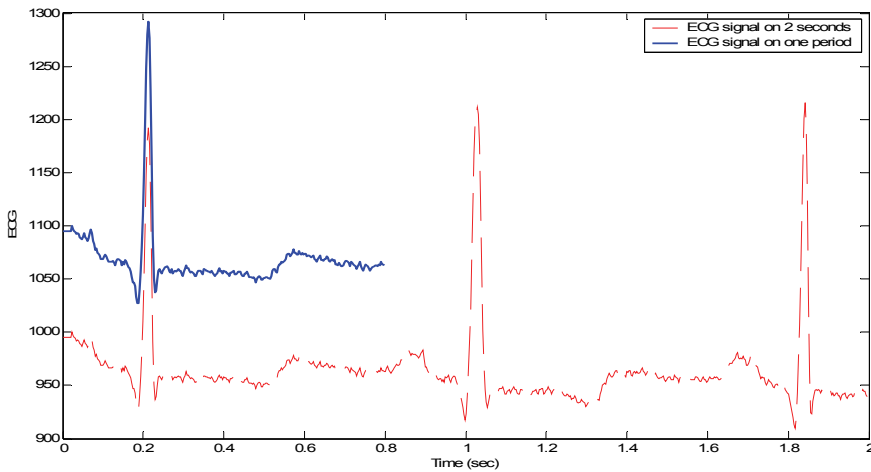
Fig. 9. Extraction of one period ECG signal

From equation (13) we have

$$y = \sum_{i=1}^{c} \varphi_i(\boldsymbol{x})\left(\boldsymbol{a}_i^T \boldsymbol{x} + b_i\right) \tag{24}$$

where $\varphi_i(\boldsymbol{x}) = \dfrac{\beta_i(\boldsymbol{x})}{\displaystyle\sum_{i=1}^{c}\beta_i(\boldsymbol{x})}$ is the normalized activation value of the ith rule. The

development of (24) gives

$$y = \begin{bmatrix} \varphi_1(\boldsymbol{x})[\boldsymbol{x} & 1] & \varphi_2(\boldsymbol{x})[\boldsymbol{x} & 1] & .... & \varphi_c(\boldsymbol{x})[\boldsymbol{x} & 1] \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ \vdots \\ a_c \\ b_c \end{bmatrix} \tag{25}$$

Let $\boldsymbol{\Theta} = \begin{bmatrix} a_1 & b_1 & ... & a_c & b_c \end{bmatrix}^T$ the $c(n+1) \times 1$ TS parameters vector and let the extended vector $\boldsymbol{x}_e = [\boldsymbol{x} \quad 1]$ with dimension $1 \times (n+1)$, also if we put $\boldsymbol{C} = \begin{bmatrix} \varphi_1 \boldsymbol{x}_e & \varphi_2 \boldsymbol{x}_e & .... & \varphi_c \boldsymbol{x}_e \end{bmatrix}$, then equation (25) can be rewritten as follows :

$$y = \boldsymbol{C}\boldsymbol{\Theta} \tag{26}$$

with $\boldsymbol{C}$ is an $1 \times (n+1)c$ vector.

To apply Kalman filter, we must introduce the measurement noise v_k, so the measurement equation corresponding to (27) at the moment k will take the following form :

$$y_k = \boldsymbol{C}_k \boldsymbol{\Theta}_k + v_k \tag{27}$$

then, we can consider that the state variable is $\boldsymbol{\Theta}_k$, so the state equation will take the following expression :

$$\boldsymbol{\Theta}_k = \boldsymbol{A}\boldsymbol{\Theta}_{k-1} + \boldsymbol{w}_{k-1} \tag{28}$$

where $\boldsymbol{A}$ is an $c(n+1) \times c(n+1)$ transition matrix and $\boldsymbol{w}_k$ is a state noise. v_k and $\boldsymbol{w}_k$ must satisfy some conditions as cited in the previous subsection.

Now we can apply Kalman filter to estimate the TS parameter vector $\boldsymbol{\Theta}_k$ as follows:

$$\hat{\boldsymbol{\Theta}}_{k/k-1} = \boldsymbol{A}\hat{\boldsymbol{\Theta}}_{k-1/k-1} \tag{29}$$

$$\boldsymbol{P}_{k/k-1} = \boldsymbol{A}\ \boldsymbol{P}_{k-1/k-1}\boldsymbol{A}^T + \boldsymbol{Q} \tag{30}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k/k-1}\boldsymbol{C}_k^T \left(\boldsymbol{C}_k \boldsymbol{P}_{k/k-1}\boldsymbol{C}_k^T + r \right)^{-1} \tag{31}$$

$$\hat{\boldsymbol{\Theta}}_{k/k} = \hat{\boldsymbol{\theta}}_{k/k-1} + \boldsymbol{K}_k \left(y_k - \boldsymbol{C}_k \hat{\boldsymbol{\Theta}}_{k/k-1} \right) \tag{32}$$

$$\boldsymbol{P}_{k/k} = \boldsymbol{P}_{k/k-1} - \boldsymbol{K}_k \boldsymbol{C}_k \boldsymbol{P}_{k/k-1} \tag{33}$$

where $\hat{\boldsymbol{\Theta}}_k$ is the estimated value of $\hat{\boldsymbol{\Theta}}_k$ and $\boldsymbol{K}_k$ is the computed Kalman gain, $\boldsymbol{P}_{k/k-1}$ and $\boldsymbol{P}_{k/k}$ are the one step predicted and filter estimate error covariance matrices, respectively. Also, for simplicity we will take $\boldsymbol{A} = \boldsymbol{I}$.

3.3 Application

In order to illustrate the effectiveness of the proposed method, we consider the problem of approximating the electrocardiogram (ECG) signal. The ECG is the graphical representation
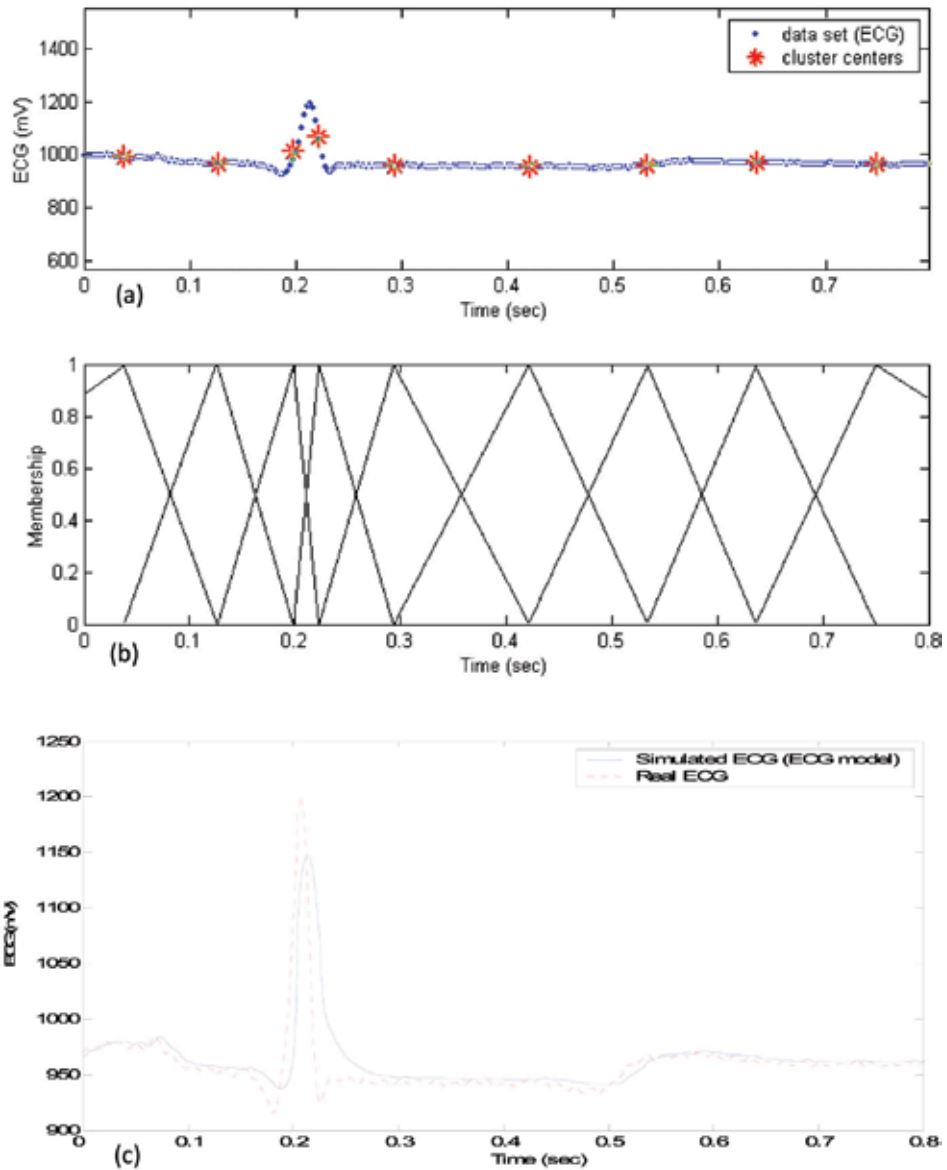
Fig. 10. (a) Clustering result for the ECG signal
          (b) Obtained membership function
          (c) Performance of our model for the ECG signal.

of the electrical activity generated by the heart. This activity shows dynamical behavior which is neither periodic nor deterministically chaotic.

To avoid the confusion between the two proposed Kalman filters, we will denote KF1 the filter used for premise membership functions and KF2 the filter used for the consequence parameters.

The considered ECG signal is taken from a publically available database of MIT (see figure 8). Before structure identification, extraction of one period ECG is done (see figure 9). In the

structure identification of the proposed method, 9 clusters are detected as shown in Fig. 10(a). By taking the projection of U on x, taking an $\alpha - cut = 0.1$ and by applying Kalman filter KF1, the premise membership functions are obtained (see figure 10(b)). The resulting model using our strategy is as follows:

$$R_i : If \quad x \quad is \quad A_i \,, \quad Then \quad y = a_i x + b_i \qquad i = 1, 2, ..., 9 \tag{34}$$

where A_i are the obtained premise membership functions, and a_i, b_i are the TS parameters to be estimated. After applying Kalman filter KF2, the TS parameters of the fuzzy model are obtained and presented in the antecedent membership function. The ECG signal and the simulated ECG signal (ECG model) are shown in Figure 10(c).

4. Conclusion

Investigations presented in this chapter were divided into two parts. In the first part, Kalman filter was used as an alternative controller. The main idea of this technique is to transform the Kalman filter from a state estimator to a control action estimator. We developed a Kalman controller system for real-time target tracking. According to our simulation results, we can say that this type of controller is very robust to load and stochastic disturbances.

In the second part, a fuzzy modelling algorithm is proposed and its validity is verified through computer simulations. This new algorithm has an excellent capacity to describe a given system. We have showed that Kalman filter can be used with fuzzy clustering to obtain a useful method to fuzzy modeling. The proposed algorithm is composed of three steps: 1) fuzzy clustering; 2) determination of premise membership functions; 3) estimation of the TS parameters. In the first step, the GKCA algorithm was used in order to detect clusters of different shapes. In the second step, a Kalman filter has been used in order to estimate the parameter values of the premise membership functions by considering the point-wise defined membership functions as a training sets. In the third step, Kalman filter is also used as a linear regression to efficiently choose the parameter values of the consequent part (TS parameters) of the fuzzy model from the input output data of the identified system. Consequently, the hybrid clustering and Kalman filter method can be efficiently constructed. The performances of the proposed modeling technique was demonstrated on modeling of ECG signal.

5. References

Abonyi, J., Babuska, R. & Szeifert, F. (2002). Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models. *IEEE Transactions on Systems, Man and Cybernetics. Part B*, Vol. 32, No. 5, (October 2002) 612-621

Angelov, P. P. & Filev, D. P. (2004). An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics*. Part B: Vol. 34, No. 1, (February 2004) 484-498

Babuska, R. (1998). *Fuzzy modeling for control*. Nowell, MA: Kluwer

Babuska, R.; Roubos, J. A. & Verbruggen, H. B. (1998). Identification of MIMO Systems by Input-Output TS Fuzzy Models, *IEEE International Conference on Fuzzy Fystems*, Vol. 1, pp. 657-662, Anchorage, 1998

Babuska, R. & Verbruggen, H. B. (1995). Identification of composite linear models via fuzzy clustering, *Proceedings of European Control Conference*, pp. 1207-1212, Rome, Italy, 1995

Babuska, R. & Verbruggen, H. B. (1997). Constructing fuzzy models by product space clustering, *In Fuzzy Model Identification, Selected Approaches,* Hellendoorn, H.& Driankov, D., (Ed.Berlin, Germany : Springer-Verlag) 53-90

Bezdek, J. C. & Dunn, J. C. (1975). Optimal Fuzzy Partitions : A Heuristic for Estimating the Parameters in a Mixture of Normal Distribution. *IEEE Transactions on Computers C-24*, (1975) 835-838

Bezdek, J. C.; Hathaway, R. J.; Howard, R. E.; Wilson, C. A. & Windham, M. P. (1987). Local convergence analysis of a grouped version of coordinate descent. *Journal of Optimization Theory and Application,* Vol. 3, No. 54, (1987) 471-477

Brookner, E. (1998). *Tracking and kalman filtering made easy*, john wiley & sons

Chafaa, K.; Ghanai, M. & Benmahammed, K. (2005). Control structures of target tracking system. *Association for the advancement of modelling and simulation techniques in entreprises' A.M.S.E, Advances in Modelling and Analysis C*, Vol. 60, No. 4, (2005) 59-73, ISSN 1240-4535.

Chafaa, K.; Ghanai, M. & Benmahammed, K. (2007). Fuzzy modeling using Kalman filter. *IET Control theory and applications*, Vol. 1, No. 1, (January 2007) 58-64

Eubank, R. L. (2006). *A Kalman filter primer*, Taylor & Francis Group

Gath, I. & Geva, A. B. (1998). Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Anal. Machine Intell.*, Vol. 7, (1998) 773-781

Gustafson, D. E. & Kessel, W. C. (1997). Fuzzy clustering with a fuzzy covariance matrix, *Proceedings IEEE CDC*, pp. 761-766, 1979.

Hathaway, R. J. & Bezdek, J. C. (1991). Grouped coordinate minimization using newton's method for Inexact minimization in one vector coordinate. *Journal of Optimization Theory and Applications*, Vol. 3, No. 71, (1991) 503-516

Haykin, S. (2001). *Kalman Filtering and Neural Networks*, John wiley & Sons

Johansen, T. A. & Babuska, R. (2003). Multiobjective identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, Vol. 11, No. 6, (December 2003) 847-860

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME, journal of the basic engineering*, (1960)

Kosko, B. (1992). Neural networks and Fuzzy systems, Prentice Hall

Kukolj, D.& Levi, E. (2004). Identification of complex systems based on neural and Takagi-Sugeno fuzzy model. *IEEE Transactions on Systems, Man and Cybernetics. Part B*, Vol. 34, No 1, (February 2004) 272-282

Mohinder, S. G. & Angus, A. P. (2001). Kalman Filtering: Theory and Practice Using Matlab, *John Wiley & Sons*

Mudi Rajani, K. & Nikhil Pal, R. (1999). A robust self-tuning scheme for PI and PD type fuzzy controllers. *IEEE transactions on fuzzy systems*, Vol. 7, No. 1, (February 1999) 2-16

Nascimento, S.; Mirkin, B. & Pives, F. M. (2003). Modeling proportional membership in fuzzy clustering. *IEEE Transactions on Fuzzy Systems.* Vol. 11, No. 2, (April 2003) 173-186

Ogata, K. (1970). *Modern control engineering*, Prentice Hall

Sugeno, M. & Yasukawa, T. (1993). A Fuzzy-logic based approach to qualitative modelling. *IEEE Transactions on Fuzzy Systems*, Vol. 1, (January 1993) 7-31

Takagi, T. & Sugeno, M. (1986). Fuzzy Identification of Systems and its Applications to Modeling and Control. *IEEE Transactions on. Systems Man and Cybernetics*. Vol. SMC-15, (Jan./Feb, 1986) 116-132

Tzeng, Y. C.; Chen, K. S.; Kao, W. L. & Fung, A. K. (1994). A dynamical learning neural network for remote sensing applicationsn. *IEEE Transactions on Geoscience, Remote Sensing*, Vol. 32, No. 5, (September 1994) 1096-1102

Zdzislaw, B. (2005). *Modern control theory*, Springer-Verlag Berlin Heidelberg

Complex Extended Kalman Filters for Training Recurrent Neural Network Channel Equalizers

Coelho Pedro H G and Biondi Neto Luiz
State University of Rio de Janeiro (UERJ)-DETEL
Brazil

1. Introduction

The Kalman filter was named after Rudolph E. Kalman published in 1960 his famous paper (Kalman, 1960) describing a recursive solution to the discrete-data linear filtering problem. There are several tutorial papers and books dealing with the subject for a great variety of applications in many areas from engineering to finance (Grewal & Andrews, 2001; Sorenson, 1970; Haykin, 2001; Bar-Shalom & Li, 1993). All applications involve, in some way, stochastic estimation from noisy sensor measurements. This book chapter deals with applications of Complex Valued Extended Kalman Filters for training Recurrent Neural Networks particularly RTRL (Real Time Recurrent Learning) neural networks. Gradient-based learning techniques are usually used in back-propagation and Real-Time Recurrent Learning algorithms for training feed forward Neural Networks and Recurrent Neural Network Equalizers. Known disadvantages of gradient-based methods are slow convergence rates and long training symbols necessary for suitable performance of equalizers. In order to overcome such problems Kalman filter trained neural networks has been considered in the literature. The applications are related to mobile channel equalizers using realistic channel responses based on WSSUS (Wide-Sense Stationary Uncorrelated Scattering) models. The chapter begins with a detailed description showing the application of Extended Kalman Filters to RTRL (Real Time Recurrent Learning) neural networks. The main equations are derived in a state space framework in connection to RTRL training. Then applications are envisioned for mobile channel equalizers where WSSUS models are adequate for handling equalization in presence of time-varying channels. This chapter proposes a fully recurrent neural network trained by an extended Kalman filtering including covariance matrices adjusted for better filter tuning in training the recurrent neural network equalizer. Several structures for the Extended Kalman Filter trained equalizer are described in detail, and simulation results are shown comparing the proposed equalizers with traditional equalizers and other recurrent neural networks structures. Conclusions are drawn in the end of the chapter and future work is also discussed.

2. Training a complex RTRL neural network using EKF

This chapter deals with the training of Recurrent Neural Networks that are characterized by one or more feedback loops. These feedback loops enable those neural networks to acquire

state representations making them appropriate devices for several applications in engineering such as adaptive equalization of communication channels, speech processing and plant control. In many real time applications fast training is required in order to make the application successful. This chapter extends the EKF (Extended Kalman Filter) learning strategy considered by Haykin (Haykin, 2001) for recurrent neural networks to the one using Real Time Recurrent Learning (RTRL) training algorithm for complex valued inputs and outputs. For instance, in the adaptive channel equalization problem for modulated signals, complex envelope signals are used, so a complex RTRL recurrent neural network could be useful in such equalization application. Rao, (Rao et. al., 2000) used EKF techniques for training a complex backpropagation neural network for adaptive equalization. The complex RTRL neural network training was also considered by Kechriotis and Manolakos (Kechriotis & Manolakos, 1994) and their training algorithm is also revisited in section 3 of this chapter with the use of a state space representation. Results indicate the feasibility of the proposed complex EKF trained RTRL neural network for tracking slow time varying signals but also shows the proposed structure does not suit scenarios where fast time varying signals are concerned. So, better time tracking mechanisms are needed in the proposed neural network structure. The authors are currently pursuing enhanced mechanisms in the complex RTRL neural network so to incorporate more information in the RTRL neural network in order improve fast time tracking. Next sections show details on how the EKF training is performed for a complex RTRL neural network. First the structure of a recurrent neural network is described then how is usually trained.

3. Recurrent neural networks

The structure of the neural network considered in this chapter is that of a fully connected recurrent network as depicted in figure 1. The usual training algorithm for that neural network is known as RTRL and was derived by Williams and Zipser (Williams & Zipser, 1989). For complex valued signals the corresponding training algorithm is called Complex EKF-RTRL or EKF-CRTRL in this chapter. Usually CRTRL training algorithms use gradient techniques for updating the weights such as the training scheme proposed by and Kechriotis and Manolakos (Kechriotis & Manolakos, 1994). Their training algorithm can be rewritten in terms of a state space representation extending Haykin's analysis (Haykin, 1999) for complex signals. So, in the noise free case, the dynamic behavior of the recurrent neural network in figure 1 can be described by the nonlinear equations.

$$\underline{x}\,(\,n+1\,) = \underline{\varphi}^{C}\,(\,W_a\,\underline{x}\,(\,n\,) + W_b\,\underline{u}\,(\,n\,)\,) =$$

$$= \underline{\varphi}\,(\,\text{real}\,(W_a\,\underline{x}\,(\,n\,) + W_b\,\underline{u}\,(\,n\,)\,) + i\,\varphi\,(\,\text{imag}(W_a\,\underline{x}\,(\,n\,) + W_b\,\underline{u}\,(\,n\,)\,)\,) =$$

$$= \underline{x}^{R}\,(\,n+1\,) + i\,\,\underline{x}^{I}\,(\,n+1\,) = \tag{1}$$

$$\underline{y}\,(\,n\,) = C\,\underline{x}\,(\,n\,) \tag{2}$$

where W_a is a q-by-q matrix, W_b is a q-by-m matrix, C is a p-by-q matrix and $\underline{\varphi}: \Re^q \rightarrow \Re^q$ is a diagonal map described by

$$
\underline{\varphi}: \begin{vmatrix} x_1 \text{ R or I} \\ x_2 \text{ R or I} \\ \vdots \\ \vdots \\ x_q \text{ R or I} \end{vmatrix} \rightarrow \begin{vmatrix} \varphi (x_1 \text{ R or I}) \\ \varphi (x_2 \text{ R or I}) \\ \vdots \\ \vdots \\ \varphi (x_q \text{ R or I}) \end{vmatrix} \tag{3}
$$

for some memoryless component-wise nonlinearity $\varphi^c: C \rightarrow C$. The spaces C^m, C^q, and C^p are named the input space, state space, and output space, respectively. It can be said that q, that represents the dimensionality of the state space, is the order of the system. So the state space model of the neural network depicted in figure 1 is an m-input, p-output recurrent model of order q. Equation (1) is the process equation and equation (2) is the measurement equation. Moreover, $\mathbf{W_a}$ contains the synaptic weights of the q processing neurons that are connected to the feedback nodes in the input layer. Besides, $\mathbf{W_b}$ contains the synaptic weights of each one of the q neurons that are connected to the input neurons, and matrix $\mathbf{C}$ defines the combination of neurons that will characterize the output. The nonlinear function φ^c (.) represents the sigmoid activation function of each one of the q neurons supposed to have the form
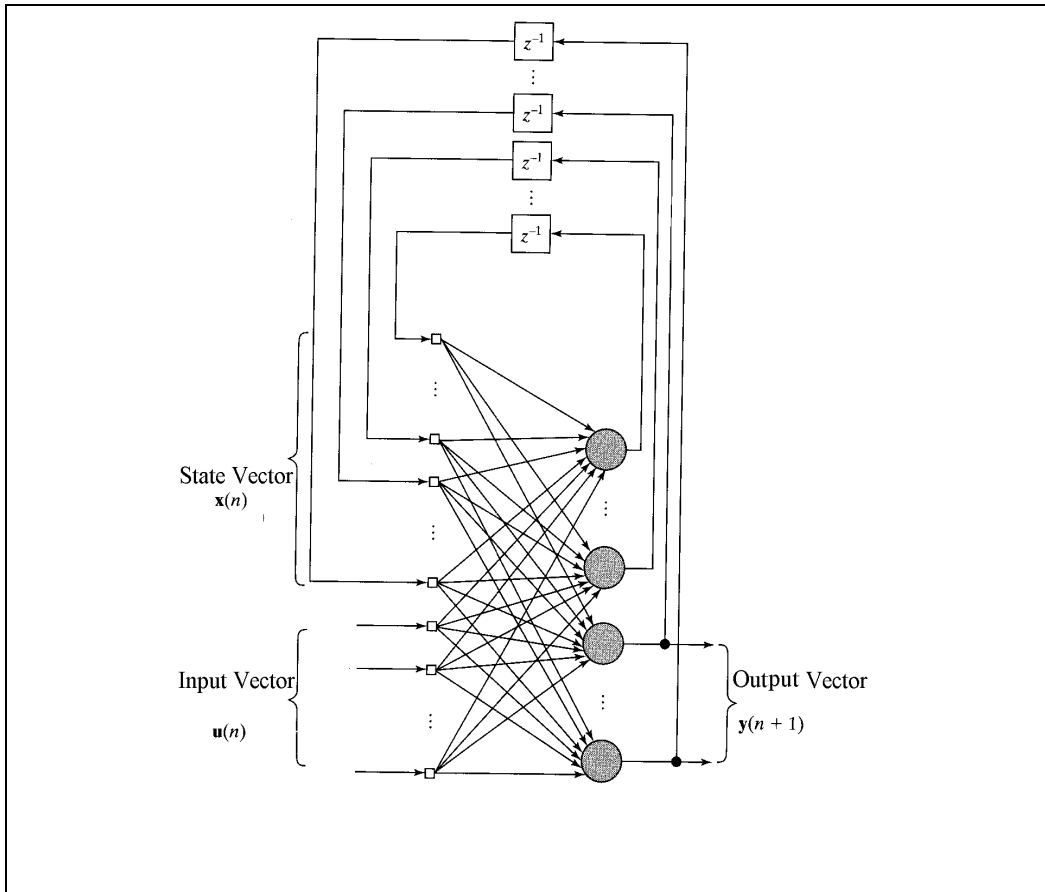

Fig. 1. Recurrent Neural Network Structure

$$\varphi^C = \varphi\,(\,\text{real}\,(\,x\,)\,) + i\,\varphi(\,(\text{imag}\,(\,x\,)\,) \tag{4}$$

where

$$\varphi\,(\,x\,) = \tanh\,(\,x\,) = (\,1 - e^{-2X}\,) / (\,1 + e^{-2X}\,) \tag{5}$$

It should be noted that the function φ^C defined by equation (4) is scalar and is obviously different from the vector function $\underline{\varphi}^C$ defined by equation (1).

4. CRTRL learning representation using a state space model

This section derives the CRTRL learning algorithm in terms of a state space model presented in section 3. The process equation (1) can be written in an expanded form as

$$\underline{x}\,(\,n+1\,) = [\,\varphi^C\,(\,\underline{w}_1{}^H\,\underline{\xi}\,(\,n\,)\,)\,\ldots\,\varphi^C\,(\,\underline{w}_q{}^H\,\underline{\xi}\,(\,n\,)\,)\,]^T =$$

$$= [\,\underline{\varphi}\,(\,\text{real}\,(\,\underline{w}_1{}^H\,\underline{\xi}\,(\,n\,)\,)\,)\,\ldots\,\underline{\varphi}\,(\,\text{real}\,(\,\underline{w}_q{}^H\,\underline{\xi}\,(\,n\,)\,)\,)\,]^T$$

$$+\,i\,[\,\underline{\varphi}\,(\,\text{imag}(\,\underline{w}_1{}^H\,\underline{\xi}\,(\,n\,)\,)\,)\,\ldots\,\underline{\varphi}\,(\,\text{imag}\,(\,\underline{w}_q{}^H\,\underline{\xi}\,(\,n\,)\,)\,)\,]^T \tag{6}$$

where it is supposed that all q neurons have the same activation function given by (4) and H is the Hermitian operator. The (q+m)-by-1 vector w_j is defined as the synaptic weight vector of neuron j in the recurrent neural network, so that

$$\underline{w}_j = \left| \begin{array}{c} \underline{w}_{a,j} \\[1em] \underline{w}_{b,j} \end{array} \right| , \qquad j = 1,2, ..., q \tag{7}$$

where $\underline{w}_{a,j}$ e $\underline{w}_{b,j}$ are the j th columns of the transposed weight matrices $\mathbf{W_a}^T$ e $\mathbf{W_b}^T$ respectively. The (q+m)-by-1 vector $\underline{\xi}\mathbf{(n)}$ is defined by

$$\underline{\xi}\,(\,n\,) = \left| \begin{array}{c} \underline{x}\,(\,n\,) \\[1em] \underline{u}\,(\,n\,) \end{array} \right| \tag{8}$$

where $\underline{x}(n)$ is the q-by-1 state vector and $\underline{u}(n)$ is the m-by-1 input vector.
Before deriving the CRTRL learning algorithm some new matrices are defined, where the indexes A and B indicate real or imaginary parts.

$$\Lambda_j{}^{AB}\,(\,n\,) = \frac{\partial \underline{\mathbf{x}}^A\,(\,n\,)}{/\,\partial \underline{\mathbf{w}}_j{}^B} \left| \begin{array}{cccc} \partial x_1{}^A/\partial w_{j1}{}^B & \partial x_1{}^A/\partial w_{j2}{}^B & \cdots & \partial x_1{}^A/\partial w_{jq+m}{}^B \\ \partial x_2{}^A/\partial w_{j1}{}^B & \partial x_2{}^A/\partial w_{j2}{}^B & \cdots & \partial x_2{}^A/\partial w_{jq+m}{}^B \\ \vdots & \vdots & \ddots & \vdots \\ \partial x_q{}^A/\partial w_{j1}{}^B & \partial x_q{}^A/\partial w_{j2}{}^B & \cdots & \partial x_q{}^A/\partial w_{jq+m}{}^B \end{array} \right| \tag{9}$$

$$
U_jA = \begin{vmatrix} 0T \\ \xi A\ T\ (n) \\ 0T \end{vmatrix} \qquad \leftarrow \text{j-th row} \tag{10}
$$

$$
\xi^A = \begin{vmatrix} \underline{x}^A \\ \\ \underline{u}^A \end{vmatrix} \tag{11}
$$

$$
\phi_R\ (\ n\ ) = \text{diag} [\ \varphi\ '(\ \text{real}\ (\ \underline{w}_1{}^H\ \underline{\xi}\ (\ n\ )\ )\ )\ \ldots\ \varphi'\ (\ \text{real}\ (\ \underline{w}_q{}^H\ \underline{\xi}\ (\ n\ )\ )\ )\ ]
$$

$$
\phi_I\ (\ n\ ) = \text{diag}[\varphi\ '(\ \text{imag}(\ \underline{w}_1{}^H\ \underline{\xi}\ (\ n\ )\ )\ )\ \ldots\ \varphi'\ (\ \text{imag}\ (\ \underline{w}_q{}^H\ \underline{\xi}\ (\ n\ )\ )\ )\ ] \tag{12}
$$

Updating equations for the matrices $\Lambda_j{}^{A\ B}\ (\ n\ )$ is needed for the CRTRL training algorithm. There are four such matrices and they all can be obtained using their formal definitions. For instance:

$$
\Lambda_j{}^{R\ R}\ (\ n\ ) = \frac{\partial \underline{x}^R\ (\ n\ )}{\partial \underline{w}_j{}^R} \tag{13}
$$

and so

$$
\Lambda_j{}^{R\ R}\ (\ n\ ) = \frac{\partial \underline{\boldsymbol{\varphi}}\ (\ \text{real}\ (W_a\ \underline{x}\ (n) + W_b\ \underline{u}\ (n)\ )\ )}{\partial \underline{w}_j{}^R} = \frac{\partial \underline{\boldsymbol{\varphi}}\ (\ \underline{s}^R)}{\partial \underline{w}_j{}^R} = \frac{\partial \underline{\boldsymbol{\varphi}}\ (\ \underline{s}^R)\ \partial \underline{s}^{\ R}}{\partial \underline{s}^{\ R}\ \partial \underline{w}_j{}^R} \tag{14}
$$

However,

$$
\frac{\partial \underline{\boldsymbol{\varphi}}\ (\ \underline{s}^R\ )}{\partial \underline{s}^{\ R}} = \text{diag}\ [\ \varphi\ '(\ \underline{s}_1{}^R\ (\ n\ )\ )\ \ldots\ \varphi'\ (\ \text{real}\ ('(\ \underline{s}_q{}^R\ (\ n\ )\ )\ ] = \phi_R\ (\ n\ ) \tag{15}
$$

and

$$
\frac{\partial \underline{\boldsymbol{\varphi}}\ (\ \underline{s}^R\ )}{\partial \underline{w}_j{}^R} = W_a{}^R\ \Lambda_j{}^{R\ R}\ (\ n\ ) - W_a{}^I\ \Lambda_j{}^{I\ R}\ (\ n\ ) + U_j{}^R\ (\ n\ ) \tag{16}
$$

where

$$U_j^R = \begin{vmatrix} \underline{0}^T \\ \underline{\xi}^{RT}(n) \end{vmatrix} \quad , \; \underline{\xi}^{RT}(n) = [\; \underline{x}^{RT} \quad \underline{u}^{RT}] \tag{17}$$

So

$$\Lambda_j^{RR}(n) = \phi_R(n) \; [\; W_a^R \Lambda_j^{RR}(n) - W_a^I \Lambda_j^{IR}(n) + U_j^R(n) \;] \tag{18}$$

The other ones can be obtained in a similar way. The four matrices can be written in a compact form as

$$\begin{vmatrix} \Lambda_j^{RR} & \Lambda_j^{RI} \\ \Lambda_j^{IR} & \Lambda_j^{II} \end{vmatrix}(n+1) \begin{vmatrix} \phi_R & \underline{0} \\ \underline{0} & \phi_I \end{vmatrix}(n+1) \begin{vmatrix} W_a^R & W_a^I \\ W_a^I & W_a^R \end{vmatrix} \begin{vmatrix} \Lambda_j^{RR} & \Lambda_j^{RI} \\ \Lambda_j^{IR} & \Lambda_j^{II} \end{vmatrix}(n) \; + \; \begin{vmatrix} U_j^R & -U_j^I \\ U_j^I & U_j^R \end{vmatrix}\tag{19}$$

The weights updating equations are obtained by minimizing the error

$$\varepsilon(n) = \tfrac{1}{2} e^H(n) \, e(n) = \tfrac{1}{2} [\underline{e}_R^{\,T}(n) \, \underline{e}_R(n) + \underline{e}_I^{\,T}(n) \, \underline{e}_I(n) \;] \tag{20}$$

The error gradient is defined as

$$\nabla_{\mathbf{wj}} \, \varepsilon(n) = \frac{\partial \varepsilon(n)}{\partial \underline{w}_j^{\,R}} + i \, \frac{\partial \varepsilon(n)}{\partial \underline{\mathbf{w}}_j^{\,I}} \tag{21}$$

where

$$\frac{\partial \varepsilon(n)}{\partial \underline{\mathbf{w}}_j^{\,R}} = \; -\Lambda_j^{RR}(n)^T \, C^T \, \underline{e}_R(n) \; - \Lambda_j^{IR}(n)^T \, C^T \, \underline{e}_I(n) \tag{22}$$

and

$$\frac{\partial \varepsilon(n)}{\partial \underline{\mathbf{w}}_j^{\,I}} = \; -\Lambda_j^{RI}(n)^T \, C^T \, \underline{e}_R(n) \; - \Lambda_j^{II}(n)^T \, C^T \, \underline{e}_I(n) \tag{23}$$

The weights updating equation uses the error gradient and is written as

$$\underline{w}_j(n+1) = \; \underline{w}_j(n) - \eta \, \nabla_{\underline{\mathbf{wj}}} \, \varepsilon(n) \tag{24}$$

So the weights adjusting equations can be written as

$$\Delta\underline{w}_j(n) \quad = \quad \Delta\underline{w}_j^{\,R}(n) + i\Delta\underline{w}_j^{\,I}(n) \quad = \eta \quad \{[\underline{e}_R^T(n)C \; \underline{e}_R^T(n)C] \begin{vmatrix} \Lambda_j^{RR} & \Lambda_j^{RI} \\ \Lambda_j^{IR} & \Lambda_j^{II} \end{vmatrix}(n)\} \begin{vmatrix} 1 \\ i \end{vmatrix} \tag{25}$$

The above training algorithm uses gradient estimates and convergence is known to be slow (Haykin, 2001) This motivates the use of faster training algorithms such as the one using EKF techniques which can be found in (Haykin, 2001) for real valued signals. Next section shows the application of EKF techniques in the CRTRL training.

5. EKF-CRTRL learning

This section derives the EKF-CRTRL learning algorithm. For that, the supervised training of the fully recurrent neural network in figure 1 can be viewed as an optimal filtering problem, the solution of which, recursively utilizes information contained in the trained data in a manner going back to the first iteration of the learning process. This is the essence of Kalman filtering (Kalman, 1960). The state-space equations for the network may be modeled as

$$\underline{w}_j(n+1) = \underline{w}_j(n) + \underline{\omega}_j(n) \quad j=1,\ldots,q$$

$$\underline{x}(n) = \underline{\varphi}^C(W_a\underline{x}(n-1) + W_b\underline{u}(n-1)) + \underline{\upsilon}(n) \quad = \begin{vmatrix} \varphi^C(\underline{w}_1{}^H\underline{\xi}(n-1)) \\ \vdots \\ \varphi^C(\underline{w}_j{}^H\underline{\xi}(n-1)) \\ \vdots \\ \vdots \\ \varphi^C(\underline{w}_q{}^H\underline{\xi}(n-1)) \end{vmatrix} + \underline{\upsilon}(n) \qquad (26)$$

where $\underline{\omega}_j(n)$ is the process noise vector , $\underline{\upsilon}(n)$ is the measurement noise vector, both considered to be white and zero mean having diagonal covariance matrices Q and R respectively and now the weight vectors $\mathbf{w}_j$ (j= 1, q) play the role of state. It is also supposed that all q neurons have the same activation function given by (4). It is important to stress that when applying the extended Kalman filter to a fully recurrent neural network one can see two different contexts where the term state is used (Haykin, 1999). First, in the evolution of the system through adaptive filtering which appears in the changes to the recurrent network's weights by the training process. That is taken care by the vectors $\mathbf{w}_j$ (j= 1, q). Second, in the operation of the recurrent network itself that can be observed by the recurrent nodes activities. That is taken care by the vector $\mathbf{x(n)}$. In order to pave the way for the application of Kalman filtering to the state-space model given by equations (6), it is necessary to linearize the second equation in (6) and rewrite it in the form

$$\underline{x}(n) = [\Lambda_1(n-1) \ldots \Lambda_j(n-1) \ldots \Lambda_q(n-1)] \begin{vmatrix} \underline{w}_1 \\ \vdots \\ \underline{w}_j \\ \vdots \\ \underline{w}_q \end{vmatrix} \qquad (27)$$

$$= \sum_{j=1}^{q} \Lambda_j(n-1)\underline{w}_j + \underline{\upsilon}(n)$$

The synaptic weights were divided in q groups for the application of the decoupled extended Kalman filter (DEKF) (Haykin, 1999). The framework is now set for the application of the Kalman filtering algorithm (Haykin, 1999) which is summarized in Table 1. Equations in Table 1 are extensions from real to complex values. The expression involving Λ_j can be evaluated through the definition

$$\Lambda_j(n) = \partial\underline{x}(n) / \partial w_j{}^R - i\,\partial\underline{x}(n) / \partial w_j{}^I \qquad (28)$$

The training procedure was improved in the EKF training by the use of heuristic fine-tuning techniques for the Kalman filtering. The tuning incorporated in the filter algorithm in Table

1 is based on the following. It is known that initial values of both the observation and the process noise covariance matrices affect the filter transient duration. These covariances not only account for actual noises and disturbances in the physical system, but also are a means of declaring how suitably the assumed model represents the real world system (Maybeck, 1979). Increasing process noise covariance would indicate either stronger noises driving the dynamics or increased uncertainty in the adequacy of the model itself to depict the true dynamics accurately. In a similar way, increased observation noise would indicate the measurements are subjected to a stronger corruptive noise, and so should be weighted less by the filter. That analysis indicate that a large degree of uncertainty is expected in the initial phase of the Kalman filter trained neural network so that it seems reasonable to have large initial covariances for the process and observation noises. Therefore the authors suggest a heuristic mechanism, to be included in the extended Kalman filter training for the recurrent neural network, that keeps those covariances large in the beginning of the training and then decreases during filter operation. In order to achieve that behavior, a diagonal matrix is added both to the process and to the observation noise covariance matrices individually. Each diagonal matrix is composed by an identity matrix times a complex valued parameter which decreases at each step exponentially. The initial value of this parameter is set by means of simulation trials. Simulation results indicated the success of such heuristic method.

Initialization:
1. Set the synaptic weights of the recurrent network to small values selected from a complex uniform distribution.
2. Set $K_j(0) = (\delta_R + i\,\delta_I)\,I$ where δ_R e δ_I are small positive constants.
3. Set $R(0) = (\gamma_R + i\,\gamma_I)\,I$ where (γ_R and γ_I are large positive constants, typically $10^2 - 10^3$.

Heuristic Filter Tuning:
1. Set $R(n) = R(n) + \alpha I$, where α decreases exponentially in time.
2. Set $Q(n) = Q(n) + \beta I$, where β decreases exponentially in time.

Compute for n = 1, 2, ...

$$\Gamma(n) = [\sum_{j=1}^{q} \Lambda_j(n) K_j(n-1) \Lambda_j^H(n) + R(n)]^{-1}$$

$$G_j(n) = K_j(n) \Lambda_j^H(n) \Gamma(n)$$

$$\underline{\alpha}(n) = \underline{d}(n) - C \sum_{j=1}^{q} \Lambda_j(n-1) \underline{w}_j$$

$$\underline{w}_j(n+1) = \underline{w}_j(n) + G_j(n)\underline{\alpha}(n)$$

$$K_j(n+1) = K_j(n) - G_j(n)\Lambda_j(n)K_j(n) + Q_j(n)$$

$$\Lambda_j(n) = \Lambda_j^{RR}(n) + \Lambda_j^{II}(n) + i(\Lambda_j^{IR}(n) - \Lambda_j^{RI}(n))$$

$\underline{d}(n)$ is the desired output at instant n

Table 1. Recurrent Neural Network Training Via Decoupled Extended Kalman Filter DEKF Algorithm Complex (**D**ecoupled **E**xtended **K**alman **F**ilter)

The authors applied the EKF-CRTL training in channel equalization problems for mobile and cell communications scenarios and representative papers are (Coelho & Biondi , 2006), (Coelho, 2002) and, (Coelho & Biondi , 2006).

6. Results and conclusion

Numerical results were obtained for the EKF-CRTRL neural network derived in the previous section using the adaptive complex channel equalization application. The objective of such an equalizer is to reconstruct the transmitted sequence using the noisy measurements of the output of the channel (Proakis, 1989).

A WSS-US (Wide Sense Stationary-Uncorrelated Scattering) channel model was used which is suitable for modeling mobile channels (Hoeher, 1992). It was assumed a 3-ray multipath intensity profile with variances (0.5, 0.3, 0.2). The scattering function of the simulated channel is typically that depicted in figure 2. This function assumes that the Doppler spectrum has the shape of the Jakes spectrum (Jakes, 1969). The input sequence was considered complex, QPSK whose real and imaginary parts assumed the values +1 and – 1.The SNR was 40 dB and the EKF-CRTRL equalizer had 15 input neurons and 1 processing neuron. It was used a Doppler frequency of zero. The inputs comprised the current and previous 14 channel noisy measurements. Figure 3 shows the square error in the output vs. number of iterations.
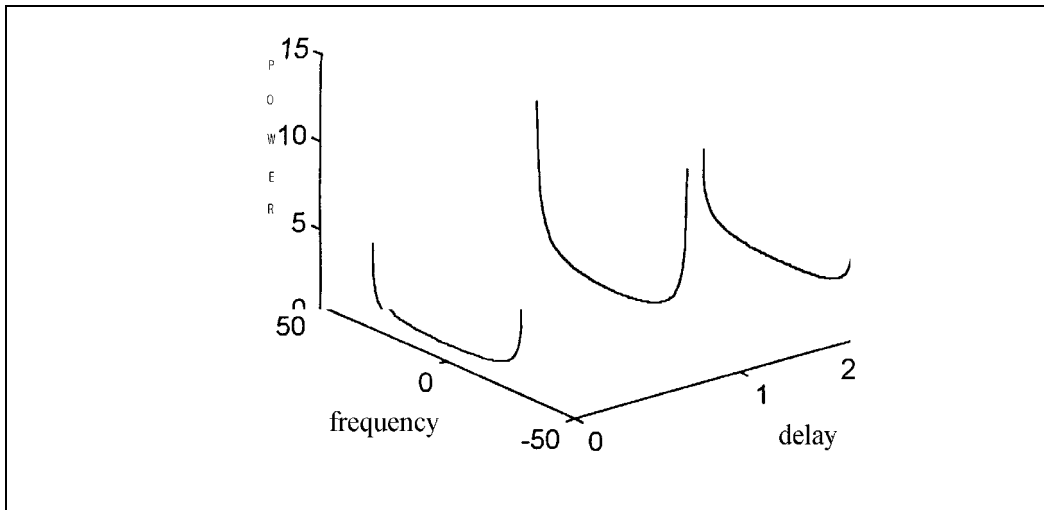
Fig. 2. Scattering Function of the Simulated Mobile Channel

Figure 3 shows a situation where the mobile receiving the signal is static, e.g. Doppler frequency zero Hz. Figure 4 shows a scenario where the mobile is moving slowly, e.g. Doppler frequency 10 Hz. To assess the benefits in the EKF-CRTRL training algorithm one can compare the square error in its output with that in the output of the CRTRL algorithm that uses gradient techniques as described in section 3. Figure 5 shows the square error in the output of the CRTRL equalizer for a 0 Hz Doppler frequency. It can be noted that convergence is slower than the EKF-CRTRL algorithm and that was obtained consistently with all simulations performed.

Fig. 3. Square error in the output of the EKF-CRTRL Equalizer (m=15,q=1, SNR=40 dB, 6 symbol delay and Doppler Frequency Zero Hz) vs. number of iterations
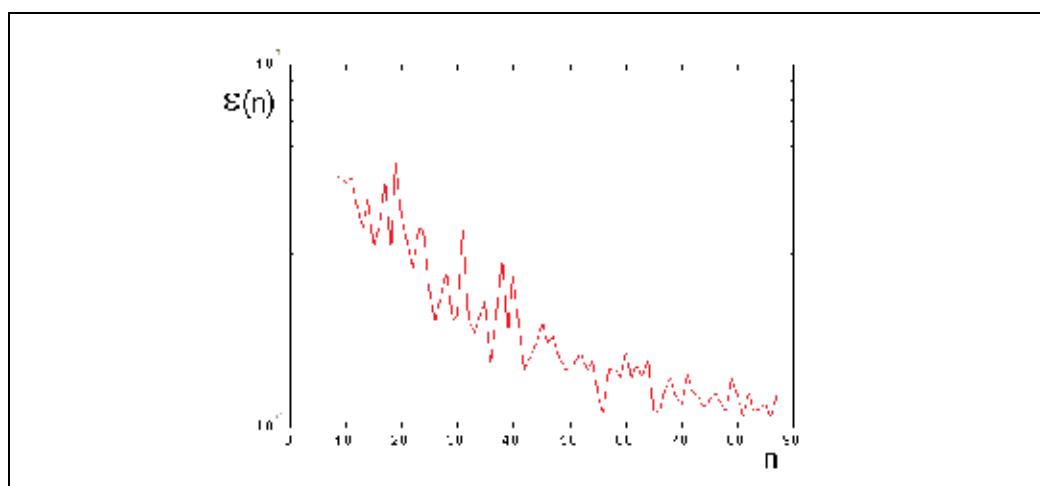
Fig. 4. Square error in the output of the EKF-CRTRL Equalizer(m=15,q=1, SNR=40 dB, 6 symbol delay and Doppler Frequency 10 Hz) vs. number of iterations

The results achieved with the EKF-CRTRL equalizer were superior to those of (Kechriotis et. al, 1994). Their derivation of CRTRL uses gradient techniques for training the recurrent neural network as the revisited CRTRL algorithm described in section 3 of this chapter. Faster training techniques are useful particularly in mobile channel applications where the number of training symbols should be small, typically about 30 or 40. The EKF-CRTRL training algorithm led to a faster training for fully recurrent neural networks. The EKF-CRTRL would be useful in all real time engineering applications where fast convergence is needed.
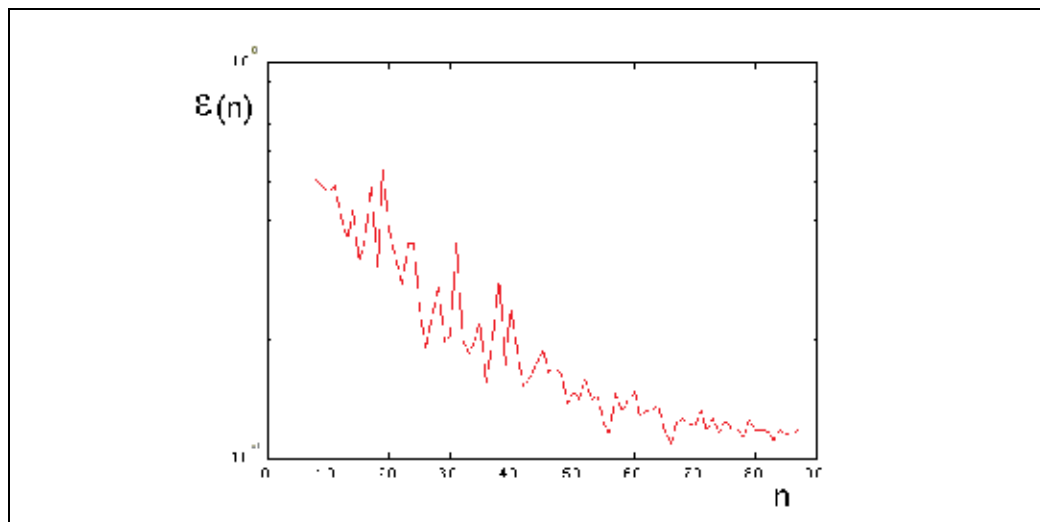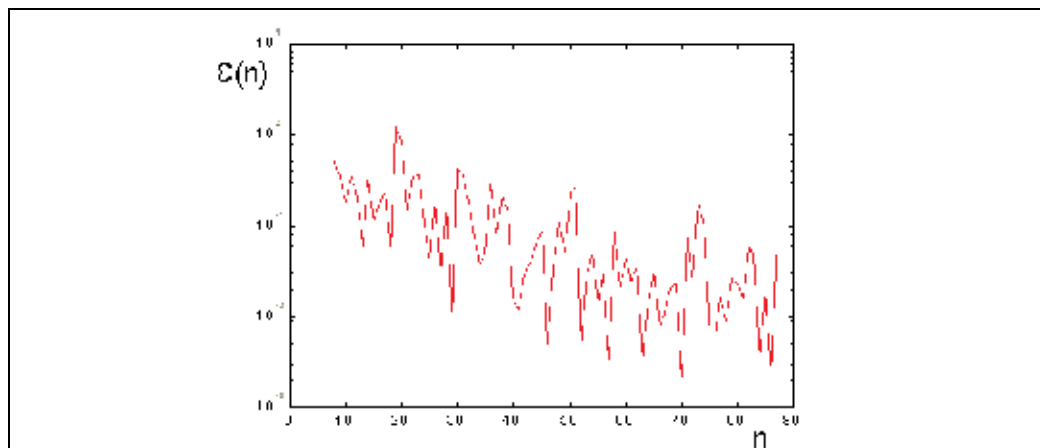
Fig. 5. Square error in the output of the CRTRL Equalizer(m=12,q=1, SNR=40 dB, 6 symbol delay and Doppler Frequency Zero Hz) vs. number of iterations
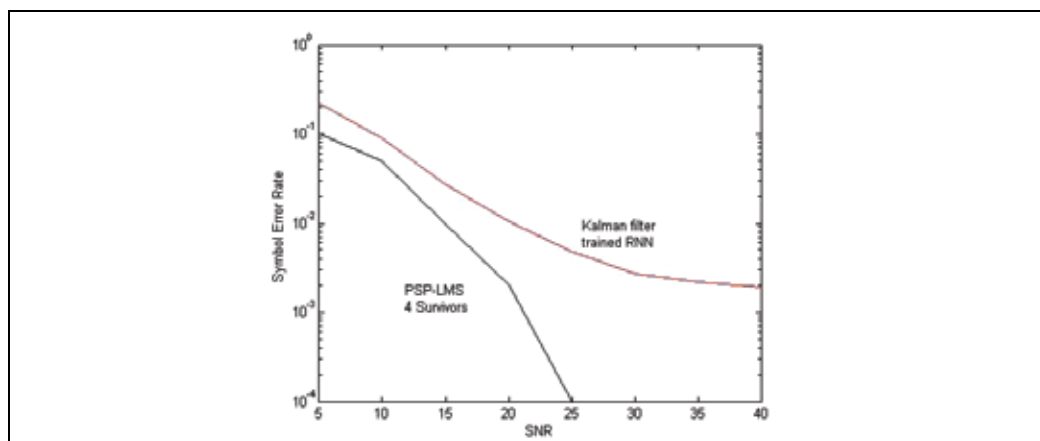

Fig. 6. Performance of the EKF-CRTRL equalizer and the PSP-LMS equalizer for f_D =0 Hz
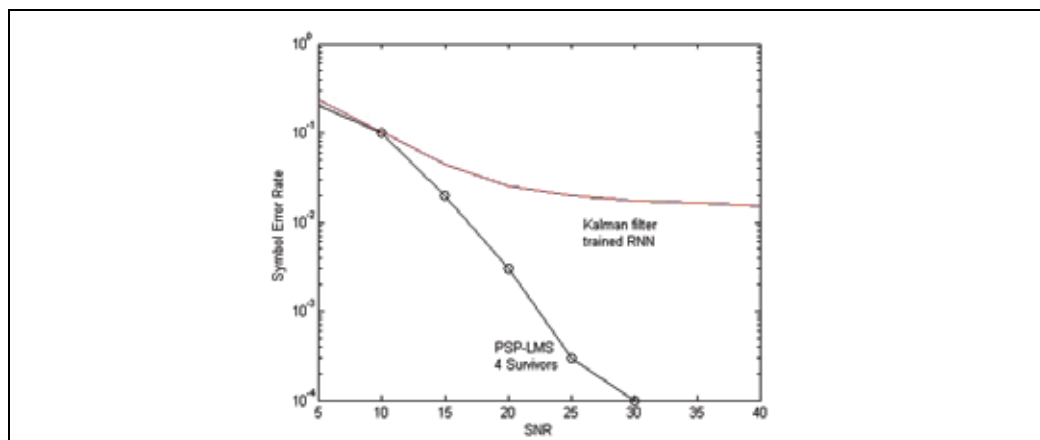
Fig. 7. Performance of the EKF-CRTRL equalizer and the PSP-LMS equalizer for f_D =10 Hz

However, the proposed equalizer is outperformed by the class of equalizers known in the literature as (PSP (Per Surviving Processing) equalizers which are a great deal more computational complex than the proposed equalizer. Figure 6 and 7 show comparisons involving the recurrent neural network equalizer regarding symbol error rate performances where one can see the superiority of the PSP-LMS equalizer. Details of such class of equalizers can be found in (Galdino & Pinto, 1998) and are not included here because is beyond the scope of the present chapter. In order to assess the EKF-CRTRL equalizer performance compared with traditional equalizers figure 8 shows symbol error rates for the equalizer presented in this chapter and the traditional Decision feedback equalizer (Proakis,
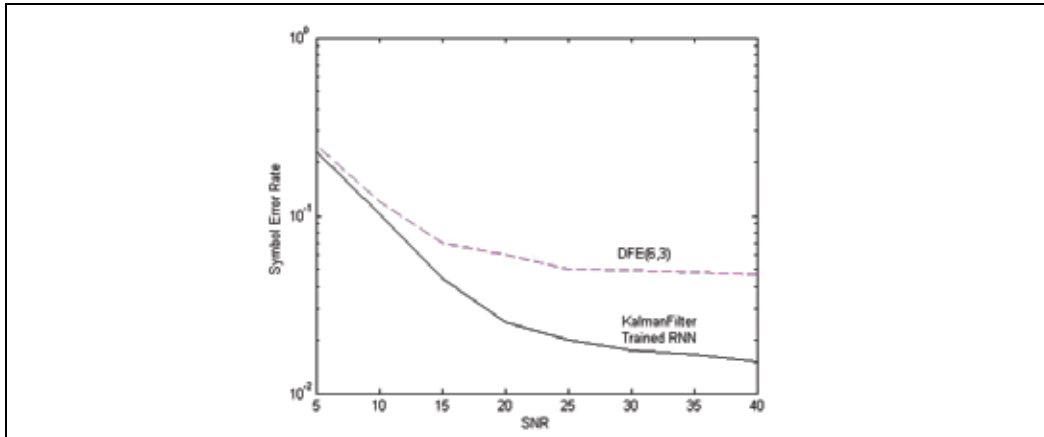
Fig. 8. Symbol Error Rate (SER) x SNR for f_D =10
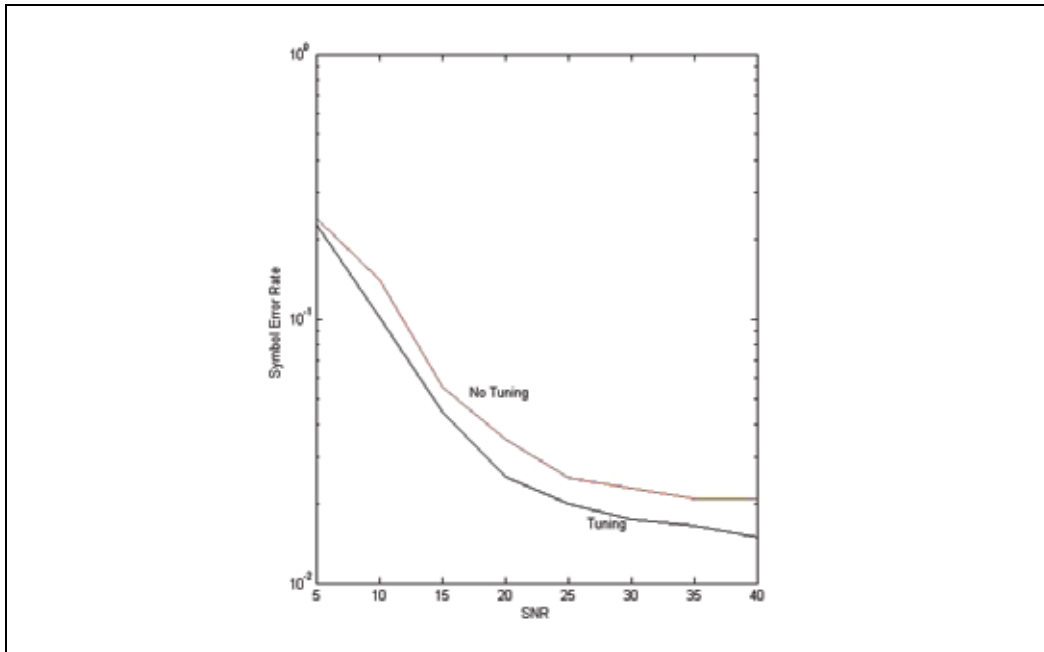
Fig. 9. Performance of the Kalman filter trained equalizer with Tuning for f_D = 0

1989). One can see the superiority of the EKF-CRTRL equalizer in the figure. Such results suggest for future work to include in the recurrent network a mechanism to enhance temporal tracking for fast time varying scenarios in high mobility speeds, typically above 50 km/h, e.g. Doppler frequencies above 40 Hz. The authors are currently working on that. Finally, figure 9 show the efficiency of the heuristic tuning mechanism proposed in this chapter in connection with the EKF-CRTRL equalizer. One can see the superior results in terms of error rate for equalizers with such tuning algorithm.

7. References

Grewal, M., S., & Andrews, A., P. (2001). *Kalman Filtering Theory and Practice Using MATLAB (Second ed.)*, John Wiley & Sons, Inc., ISBN 0-471-39254-5, New York, NY, USA

Sorenson, H. W. (1970). Least-Squares estimation: from Gauss to Kalman. *IEEE Spectrum*, (July 2001), pages 63-68, ISSN 0018-9235

Haykin, S., Ed. (2001). *Kalman Filtering and Neural Networks,* John Wiley & Sons, Inc., ISBN 0-471-36998-5, New York, NY, USA

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME – Journal of Basic Engineering*, Vol . 82 (Series D), pages 35-45

Grewal, M., S., & Andrews, A., P. (2001). *Kalman Filtering Theory and Practice Using MATLAB (2nd Edition)*, John Wiley & Sons, Inc., ISBN 0-471-39254-5, New York, NY, USA

Sorenson, H. W. (1970). Least-Squares estimation: from Gauss to Kalman. *IEEE Spectrum*, (July 2001), pages 63-68, ISSN 0018-9235

Haykin, S., Ed. (2001). *Kalman Filtering and Neural Networks,* John Wiley & Sons, Inc., ISBN 0-471-36998-5, New York, NY, USA

Bar-Shalom, Y., & Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, Inc.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation (2nd Edition)*, Prentice Hall, ISBN-13: 978-0132733502, USA

Kechriotis, G., and Manolakos, E. S.(1994). Training Fully Recurrent Neural Networks with Complex Weights, *IEEE Trans. Circuits Syst. II*, (March 1994), pages 235-238

Rao, D. K., Swamy, M. N. S. and Plotkin, E. I. (2000). Complex EKF Neural Network for Adaptive Equalization. ISCAS 2000, *Proceedings of the International Symposium on Circuits and Systems*, pp. II-349- II-352, Geneva May 28-31, 2000, Switzerland

Williams, R. J. and Zipper. D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, Vol.1, 1989, pages 270-280

Maybeck, P. S.(1979). *Stochastic Models, Estimation, and Control,* Vol. 1, Academic Press, ISBN-0-12-480701-1, NY, USA

Coelho, P.H.G. & Biondi Neto L. (2006). Complex Kalman Filter Trained Recurrent Neural Network Based Equalizer for Mobile Channels, *Proceedings of the 2006 International Joint Conference on Neural Networks,* pp. 2349-2353, Sheraton Vancouver Wall Centre Hotel, July 16-21, 2006, Vancouver, BC, Canada

Coelho, P. H. G. and Biondi Neto L., (2005). Further Results on The EKF-CRTRL Equalizer for Fast Fading and Frequency Selective Channels, *Proceedings of the 2005 International Joint Conference on Neural Networks*, pp. 2367-2371, July 31-August 4 2005, Montreal, Quebec, Canada

Coelho, P.H.G. (2002). Adaptive Channel Equalization Using EKF-CRTL Neural Networks, *Proceedings of the 2002 International Joint Conference on Neural Networks,* pp. 1195-1199, , May 12-17, 2002, Honolulu, Hawaii, USA

Coelho, P. H. G. & Biondi Neto L. (2006). Complex Kalman Filter Trained Recurrent Neural Network Based Equalizer for Mobile Channels, *Proceedings of the 2006 International Joint Conference on Neural Networks,* pp. 2349-2353, Sheraton Vancouver Wall Centre Hotel, July 16-21, 2006, Vancouver, BC, Canada

Proakis, J. G. (1989). *Digital Communications (2nd Edition)*, McGraw-Hill, ISBN 0-07-100269-3, USA

Hoeher, P. (1992). Statistical Discrete- Time Model for the WSSUS Multipath Channel, *IEEE Transactions on Vehicular Technology*, Nov. 1992, Vol 41. Number 4, pp. 461-468

Jakes, W. C. Jr. (1969). *Microwave Mobile Communications,* Wiley, USA

Kechriotis, G., Zervas, E. and Manolakos, E. S. (1994). Using Recurrent Neural Networks for Adaptive Communication Channel Equalization, *IEEE Transactions on Neural Networks*, March 1994, Vol. 5, Number 2, pages 267-278

Galdino J. F. and Pinto E. L., (1998). A New MLSE-PSP Scheme Over Fast Frequency-Selective Fading Channels, *International Symposium on Information Theory and Its Applications*, Mexico City, Mexico, 14-16 October 1998.

4

Kalman Filtering for Sensor Fusion in a Human Tracking System

Juan A. Corrales, Francisco A. Candelas and Fernando Torres
Automatics, Robotics and Computer Vision Group, University of Alicante
Spain

1. Introduction

Robotic systems need to be context-aware in order to adapt their tasks to the different states of their environment. This context-awareness does not only imply the detection of the objects which are near the robot but it also includes the tracking of people who collaborate with it. Thus, human-robot interaction tasks become more natural and unobtrusive because robots are able to change their behaviour depending on this context information.

In industrial environments, these context-aware systems should also guarantee the safety of human operators who interact with industrial robots. Therefore, a precise localization of all the limbs of the body of the operator has to be determined. In this chapter, the use of an inertial motion capture system for tracking full-body movements of the operator is described. It is composed of 18 IMUs (Inertial Measurement Units) attached to the body of the operator which determine the rotation angle of each joint. It has several advantages over other motion capture technologies: easy installation, self-containment, occlusions-free and precise rotational measurements. However, it accumulates a small error (drift) in the estimation of the global translation of the human operator in the environment which becomes considerable after several movements of the operator. Therefore, an additional location system based on UWB (Ultra-Wide Band) signals has been added to correct this drift accumulation.

The features of both tracking systems are complementary. The inertial motion capture system registers accurate joint rotation angles at a high rate while the UWB location system estimates global translation in the environment at a low rate. The combination of these systems will reduce the drawbacks of each one with the advantages of the other one. On one hand, the global translation measurements of the UWB system will correct the accumulated drift of the motion capture system. On the other hand, the high rate measurements of the motion capture system will complete the periods of time when there are not any measurements from the UWB system.

Firstly, a simple fusion algorithm of both tracking systems is presented. This first fusion algorithm transforms measurements from the two systems in the same coordinate system by recalculating the transformation matrix each time a new measurement from the UWB system is received. This approach relies heavily on the accuracy of the measurements from the UWB system because the transformation matrix recalculation assumes that the last UWB measurement is completely correct. Thus, errors in UWB measurements are not considered and only the translational errors of the motion capture system are corrected. Furthermore,

when there is a considerable distance between the last measurement from the motion capture system and the last measurement from the UWB system, significant gaps appear in the final trajectory returned by the fusion algorithm. Another fusion algorithm which takes into account UWB errors has been developed in order to overcome these drawbacks and obtain more continuous trajectories. This second fusion algorithm is based on a Kalman filter.

This chapter is organized as follows. First of all, the two tracking systems are described in detail in section 2. In section 3, the fusion algorithms developed to combine the measurements of these systems are explained and compared with previous research. In section 4, several experiments are presented in order to compare the accuracy of both fusion algorithms. The Kalman filter algorithm is applied in a human-robot interaction task. Finally, the conclusions of this chapter and future research are presented in section 5.

2. Overview of the tracking systems

2.1 Inertial motion capture system

A system based on inertial sensors has been selected because it has several advantages over other motion capture sensor technologies (Welch & Foxlin, 2002). It is comfortable for the user because it does not limit his/her movements like mechanical motion capture systems. Its measurements are not negatively influenced by magnetic distortions like magnetic systems. Finally, it does not suffer from occlusion problems like optical systems.

The inertial motion capture system used in the present research (Animazoo, 2008) is composed of 18 small IMUs (Inertial Measurement Units) attached to a lycra suit (Fig. 1a) which is worn by a human operator. Each IMU (Fig. 1b) estimates the orientation (roll, pitch and yaw) of the operator's limb to which it is attached by combining the measurements from three miniaturized gyroscopes, three accelerometers and three magnetometers
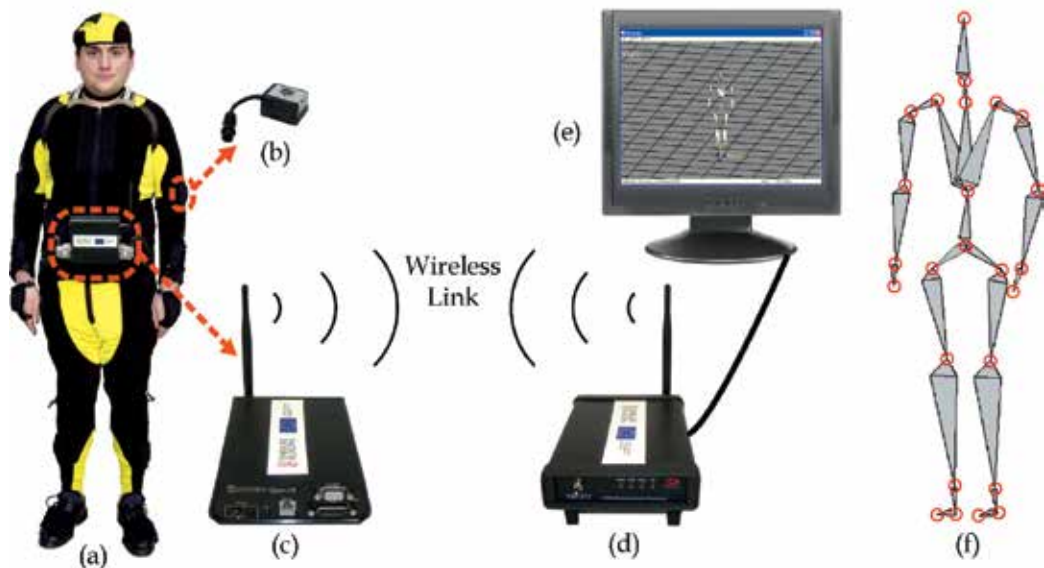

Fig. 1. Components of the inertial motion capture system: (a) suit, (b) IMU, (c) MPU, (d) wireless modem, (e) controller PC and (f) skeleton structure.

(Foxlin *et al.*, 1998). All the IMUs are connected to a MPU (Main Processing Unit, Fig. 1c) which recovers orientation measurements and sends them wirelessly to a controller PC (Fig. 1d and Fig. 1e). This PC calculates the global translation of the user in the environment with a software algorithm with determines the length of the user's steps. Limbs' rotations registered by the IMUs are applied over a 3D skeleton (Fig. 1f) which represents the basic structure of the human operator.

Orientation measurements returned by these IMUs have a resolution of $0.1°$ and an accuracy of $1°$ in yaw and $0.25°$ in roll and pitch. These rotational errors are small enough for most industrial applications. Nevertheless, the accuracy of the global translation measurements estimated by the footstep extrapolation algorithm is not specified by the manufacturer. A set of experiments has been developed in order to quantify the translational error of the system. In these experiments, a person who is wearing the motion capture suit walks along a linear path of different lengths (200, 300 and 400cm). The error values obtained by comparing translation measurements from the motion capture system and actual distances of the experiments are shown in Table 1. These errors are very high for industrial purposes and an additional localization system is needed in order to correct them. The following section describes the UWB location system which has been used in this chapter.

Distance (cm)	Minimum error	Maximum error	Mean error	Standard Deviation
200	16.70	66.04	40.10	17.92
300	15.33	69.54	37.92	20.97
400	35.43	64.23	51.09	10.67

Table 1. Global translation error statistics (in cm) in the motion capture system.

2.2 UWB location system

A location system based on UWB pulses has been used because it has some advantages over other wireless indoor location systems (Liu et al., 2007). The small temporal duration of UWB pulses makes them less susceptible to multipath fading and interferences than other radio-frequency technologies. In addition, the infrastructure that has to be installed in the workspace is smaller than other technologies (e.g. ultrasound) because sensors have a bigger operating range.

The UWB system used in this chapter (Ubisense, 2008) consists of two kinds of hardware devices: sensors (Fig. 2a and Fig. 2b) and tags (Fig. 2c). Sensors are situated at fixed positions in the localization area. Tags are small devices, of similar size to a credit card, which are carried by the user. The tag sends UWB pulses to the sensors, which use a combination of TDOA (Time-Difference of Arrival) and AOA (Angle of Arrival) techniques to estimate the 3D location of the user who is carrying the tag (Ubisense, 2007).

The UWB sensors are connected to an Ethernet switch (Fig. 2d) and send the location information to a controller PC (Fig. 2e) which estimates the global position of the tag in the coordinate system of the UWB system. Slave sensors are also connected to a master sensor for synchronization in the TDOA algorithm.
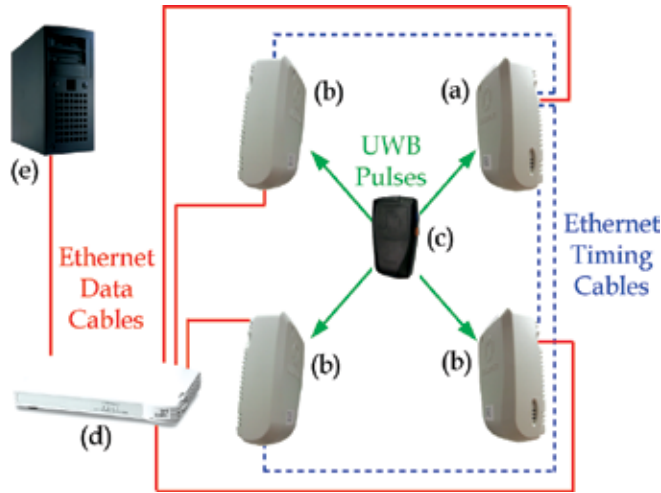
Fig. 2. Components of the UWB location system: (a) master sensor, (b) slave sensors, (c) tag, (d) Ethernet switch and (e) controller PC.

3. Sensor fusion algorithms

The experiments described in Table 1 have shown that translational measurements from the motion capture system have high error values (larger than 60cm in many cases). The UWB localization system obtains more accurate position measurements with errors smaller than 15cm. Nevertheless, the sampling rate of the UWB system (5-9Hz) is not high enough to track quick human movements in industrial environments. On the other hand, the inertial motion capture system supplies high data rates (30-120Hz).

Since both tracking systems have complementary features, their combination will make the most of their advantages. UWB measurements will be used to correct the accumulated drift in the position estimated by the motion capture system. Position measurements from the motion capture system which are obtained between each pair of UWB measurements will be used to reduce the latency of the UWB system. Thereby, the resulting system from the fusion of both trackers will have a higher sampling rate and a better accuracy than each system separately. Rotational measurements for each joint (obtained directly from the IMUs) will remain unchanged because they are accurate relative rotation transformations in the skeleton structure of the motion capture system.

In the following sections, two algorithms for combining the measurements of both tracking systems are explained in detail. The first algorithm is a simple approach where the measurements of both systems are transformed to the same coordinate system and each UWB measurement is used to correct the following motion capture measurements. The second algorithm is based on a Kalman filter which has been modified in order to incorporate the measurements of the two tracking systems.

3.1 Transformation recalculation algorithm

The first step to combine global position measurements of both tracking systems is to represent them in the same coordinate system. The frame U of the UWB system is a fixed coordinate system in the workspace because it is established according to the static positions

where the UWB sensors are installed. However, the frame *I* of the inertial motion capture system is a dynamic coordinate system because it is established in the place where the user is standing every time the system is initialized. The frame *U* of the UWB system has been selected as the reference coordinate system because it is able to compare the position of the human operator with the position of static objects (like machinery) in the environment.

XY planes of the *U* and *I* frames are parallel to the plane of the floor in the environment. Therefore, between the motion capture frame and the UWB frame there is only a translation and a rotation about the *Z* axis. Equation (1) is used to transform a point *p* from frame *I* to frame *U*:

$$p^U = {}^U T_I \cdot p^{\,I} = Trans\left(t_x, t_y, t_z\right) \cdot Rot\left(z^U, \alpha\right) \cdot p^{\,I} \tag{1}$$

The development of equation (1) results in the following equation:

$$\begin{bmatrix} x^U \\ y^U \\ z^U \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & t_x \\ \sin(\alpha) & \cos(\alpha) & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x^I \\ y^I \\ z^I \\ 1 \end{bmatrix} \tag{2}$$

The parameter α is a known constant value which represents the angle between the *Y* axis of the frames *U* and *I*. Therefore, the only unknown of the transformation matrix ${}^U T_I$ are the three coordinates of the translation vector between frame *U* and frame *I*. They can be calculated from equation (2) by substituting two corresponding measurements of both systems:

$$t_x = x^U - x^I \cos(\alpha) + y^I \sin(\alpha) \tag{3}$$

$$t_y = y^U - x^I \sin(\alpha) - y^I \cos(\alpha) \tag{4}$$

$$t_z = z^U - z^I \tag{5}$$

After obtaining the transformation matrix ${}^U T_I$, all the translational measurements from the motion capture system will be transformed to the UWB frame by applying equation (1).

However, if the transformation matrix ${}^U T_I$ is calculated only when the system is initialized (with the first two measurements), the motion capture system will accumulate translational errors through time. The measurements from the UWB system have to be used in order to correct these errors. Therefore, this transformation matrix is estimated each time a new UWB measurement is received (p^U_{uwb}). The following measurements from the inertial motion capture system ($p_{inertial}$) are transformed from the *I* frame ($p^I_{inertial}$) to the *U* frame ($p^U_{inertial}$) by applying this new transformation matrix. These transformed measurements are used as estimates of the global position of the operator in the environment. The transformation recalculation corrects the accumulated translation errors from the motion capture system while the high rate of its measurements is maintained. The complete fusion algorithm is summarized in Fig. 3.
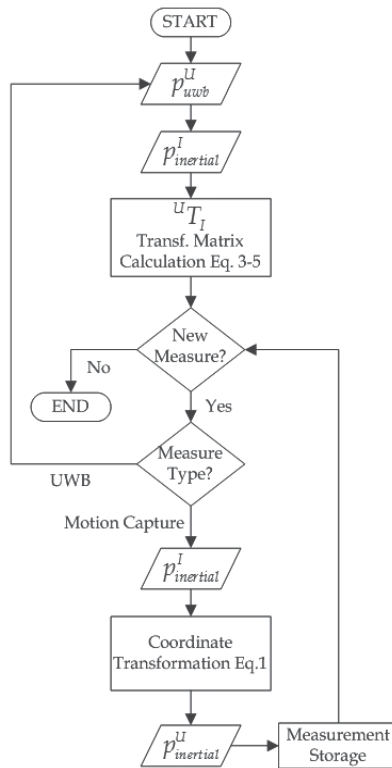
Fig. 3. Diagram of the transformation recalculation fusion algorithm.

3.2 Kalman filter algorithm
3.2.1 Previous work on Kalman filtering sensor fusion

Pose (position and orientation) estimation by inertial sensors is a well-studied field with applications in: vehicle navigation, virtual reality (VR), augmented reality (AR), robotics and human motion capture. Positions and orientations are calculated by integrating accelerations and angular rates respectively returned by accelerometers and gyroscopes. This dead reckoning process introduces a small error (drift) which is accumulated through time and becomes considerable in a few seconds. Location systems based on inertial sensors usually include additional sensors (e.g. GPS, ultrasound, magnetic, cameras, UWB and WiFi) which return absolute pose measurements in order to correct the inertial drift. In these hybrid location systems, Bayes filtering techniques (Fox et al., 2003) are generally used to estimate probabilistically the system's pose (state) from the noisy measurements of the sensors (observations). Kalman filters are the most commonly used technique to implement Bayesian filters. Different adaptations of the Kalman filter (Simon, 2001) have been proposed in previous work in order to combine measurements from several sensors. The two adaptations more commonly used are: the complementary Kalman filter and the definition of several channels (measurement models) in the correction step of the filter.

A complementary Kalman filter is an easy way to integrate several sensors measurements in a Kalman filter because the internal structure of the filter is not changed. Complementary Kalman filters estimate sensors errors instead of direct measurements. They receive as input

the differences between the sensors measurements. (Foxlin, 1996) develops an IMU based on a complementary Kalman filter for head-tracking in virtual environments. It is composed by three orthogonal angular rate gyroscopes, a two-axis inclinometer and a two-axis compass. This system estimates errors in orientation (from the inclinometer and the compass) and angular rate (from the gyros). The complementary Kalman filter has also been used by (Foxlin, 2005) in a navigation system which tracks the location of a pedestrian from two IMUs mounted on his/her shoes and a GPS receiver. (Roetenberg et al., 2007) presents a human motion capture system which uses the differences between an inertial tracking system and a magnetic tracker in position and orientation as the measurement updates for a complementary Kalman filter.

(Caron et al., 2006) extends the definition of the Kalman filter to include measurements from multiple sensors (an IMU and a GPS receiver) in the correction step. This Kalman filter has one measurement model for each sensor type which is weighted according to fuzzy context variables. These fuzzy variables represent sensors data reliability and are used to reject bad measurements. (You & Neumann, 2001) develops a similar solution where an extended Kalman filter with two independent correction channels combines position measurements from a camera and orientation measurements from three orthogonal rate gyroscopes.

3.2.2 Description of the proposed algorithm

The generic Kalman filter algorithm (Simon, 2001) has been adapted to incorporate sensor measurements from the two tracking systems. The state of the Kalman filter is modelled by two parameters at each step t: the state estimate $\hat{p}_t$ and the error covariance matrix P_t. The state p_t is composed by the coordinates $p_t = (x_t, y_t, z_t)$ of the global position of the human operator in the environment. These two parameters are calculated by the Kalman filter in two steps: prediction and correction. The prediction step uses a state evolution model where the a-priori estimate of the state $\hat{p}_t^-$ is obtained from the current measurement of the inertial motion capture system $p_{inertial}^t$ plus a Gaussian noise $w_{inertial}^t$:

$$\hat{p}_t^- = p_{inertial}^t + w_{inertial}^t \tag{6}$$

This model is implemented in the Kalman filter prediction step using the following two equations:

$$\hat{p}_t^- = A p_{inertial}^t \tag{7}$$

$$P_t^- = A P_{t-1} A^T + Q \tag{8}$$

The matrix A represents the state transition model of (6) and thus it is a 3x3 identity matrix which uses the last measurement from the motion capture system as the current state estimate. In the correction step, measurements from the UWB system p_{uwb}^t are modelled from the last state estimate of the prediction step $\hat{p}_t^-$ plus a Gaussian noise w_{uwb}^t:

$$p_{uwb}^t = \hat{p}_t^- + w_{uwb}^t \tag{9}$$

This measurement model is implemented in the Kalman filter correction step using the following three equations:

$$K_t = P_t^- H^T \left(H P_t^- H^T + R \right)^{-1} \tag{10}$$

$$\hat{p}_t = \hat{p}_t^- + K_t \left(p_{uwb}^t - H \hat{p}_t^- \right) \tag{11}$$

$$P_t = \left(I - K_t H \right) P_t^- \tag{12}$$

Position measurements from the UWB system are used as observations p_{uwb}^t in this step in order to correct the predicted position $\hat{p}_t^-$ calculated from the motion capture system. The matrix H represents the measurement model of (9) and it is a 3x3 identity matrix because UWB measurements have the same dimension as the state.

Error covariance matrices (Q and R) of the Kalman filter are 3x3 diagonal matrices because error components are not correlated. The diagonal terms of both matrices represent the variance of error of each tracking system (motion capture and UWB, respectively) and have been adjusted experimentally in order to reduce the errors in the final state estimates.

The implemented algorithm is depicted in Fig. 4. The prediction step (equations (7) and (8)) is executed when a measurement from the inertial motion capture system is received while the correction step (equations (10), (11) and (12)) is executed when UWB measurements are received. The main advantage of this algorithm over previous sensor fusion techniques
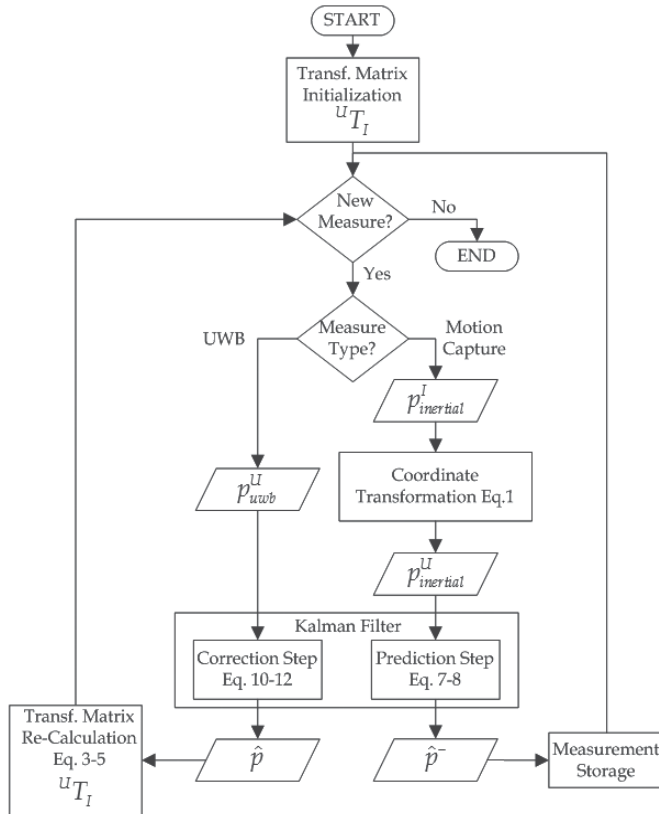

Fig. 4. Diagram of the Kalman filter fusion algorithm.

based on Kalman filtering (see section 3.2.1) is that only one step of the algorithm (prediction or correction) is executed each time a measurement is obtained. Thereby, the execution time of the algorithm is smaller. In addition, the prediction step of the developed algorithm adjusts better to changes in the movements of the operator because it uses real position measurements from the inertial motion capture system. Previous Kalman filter algorithms implement theoretical kinematic models which suppose that the velocity (You & Neumann, 2001) or the acceleration (Caron et al., 2006; Roetenberg et al., 2007) of the operator can be represented as a constant value plus a Gaussian noise.

Although this approach implies executing the correction step with a lower rate than the prediction step, this fact does not imply a dramatic reduction in the accuracy of the estimated states. The error accumulated by the motion capture system between each pair of UWB measurements is not significant and it is not accumulated for future motion capture measurements after a new UWB measurement is received. This is because the corrected state estimate obtained from the last UWB measurement is used to recalculate the transformation matrix $^{U}T_{I}$ between the frames of both tracking systems. This new transformation matrix is applied to the subsequent measurements from the motion capture system and thus the error of the previous measurements is removed.

4. Experimental results

4.1 Comparison between the fusion algorithms

Some experiments have been performed to compare the accuracy of the two fusion algorithms presented in the previous sections. A human operator wearing the motion capture suit and a UWB tag has walked along two different pre-established paths: a linear trajectory (3m) and a rectangular trajectory (8m). Each trajectory has been repeated fifteen times. These measurements are processed in two Matlab functions which implement the two fusion algorithms.

In Fig. 5a, original measurements from the UWB system and the motion capture (MoCap) system in an experiment of the linear trajectory are represented in the same coordinate system. This plot shows the advantages and disadvantages of both tracking systems. On one hand, motion capture measurements have a higher sampling rate but they accumulate an error which increases through time. On the other hand, UWB measurements have a smaller sampling rate but do not accumulate errors through time.

Fig. 5b and Fig. 5c depict the position estimates obtained from the fusion algorithms. The trajectories obtained by the Kalman filter algorithm (Fig. 5c) are more continuous than the trajectories from the transformation recalculation algorithm (Fig. 5b). In addition, the position estimates of the Kalman filter have smaller errors than those of the transformation recalculation algorithm (Fig. 5d).

In Fig. 6a, Fig. 6b and Fig. 6c, an experiment of the rectangular trajectory is represented and similar results are obtained. The estimates obtained from the Kalman filter are more precise than those from the transformation recalculation algorithm (Fig. 6d). The higher accuracy of the Kalman filter estimates is due to the fact that this algorithm takes into account the errors of both tracking systems while the transformation recalculation algorithm only corrects the errors of the motion capture system. The transformation recalculation algorithm uses the

UWB measurements as correct positions in order to calculate the transformation matrix and thus correct the motion capture errors. However, the errors of the UWB system are not corrected and cause discontinuities in the trajectories (Fig. 5b and Fig. 6b). In conclusion, the Kalman filter fusion algorithm obtains more accurate position estimates and continuous trajectories and it will be used to implement human-robot interaction tasks where the human operator has to be localized precisely.

(a)

(b)

(c)

(d)
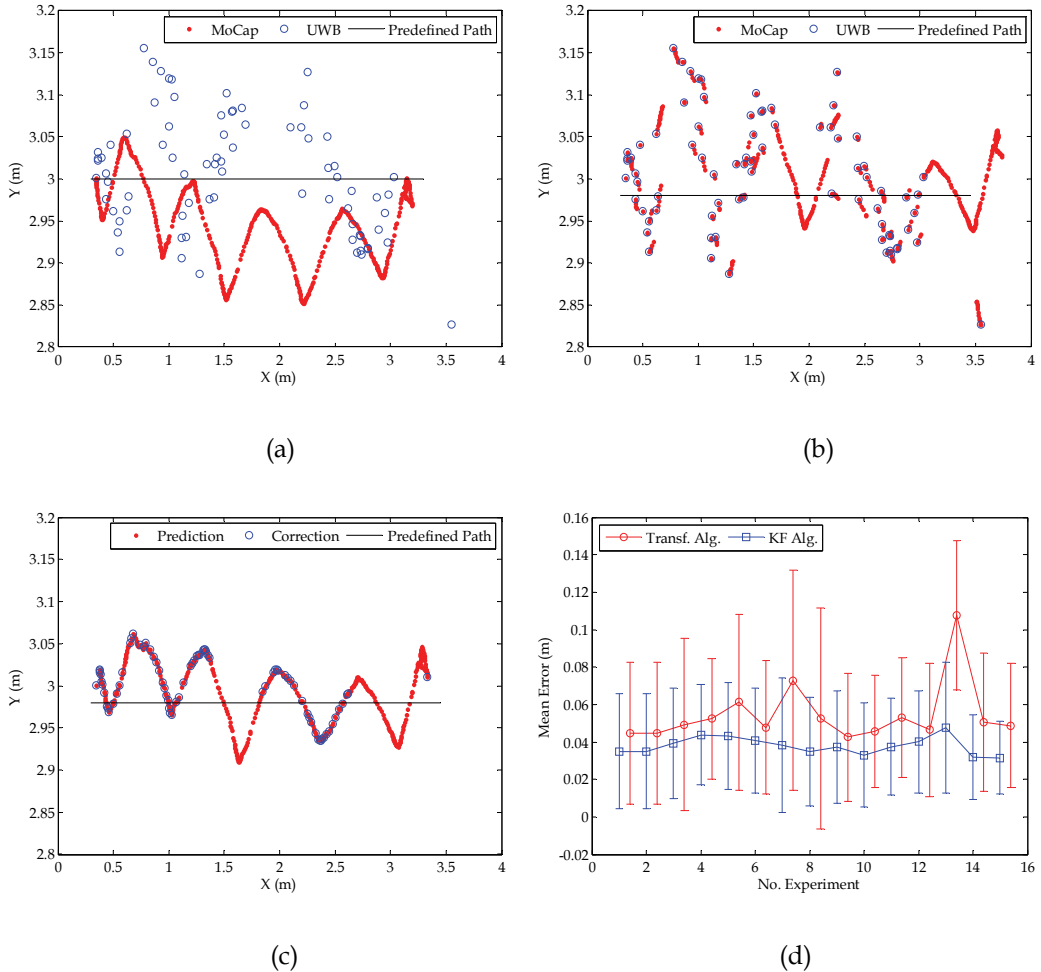
Fig. 5. Linear trajectory experiments: (a) original position measurements from both tracking systems in experiment no. 14; (b) position measurements obtained from the transformation recalculation algorithm in experiment no. 14; (c) position estimates obtained from the prediction and correction steps of the Kalman filter algorithm in experiment no. 14; (d) mean error and standard deviation of the position estimates from the two fusion algorithms in the 15 linear path experiments.
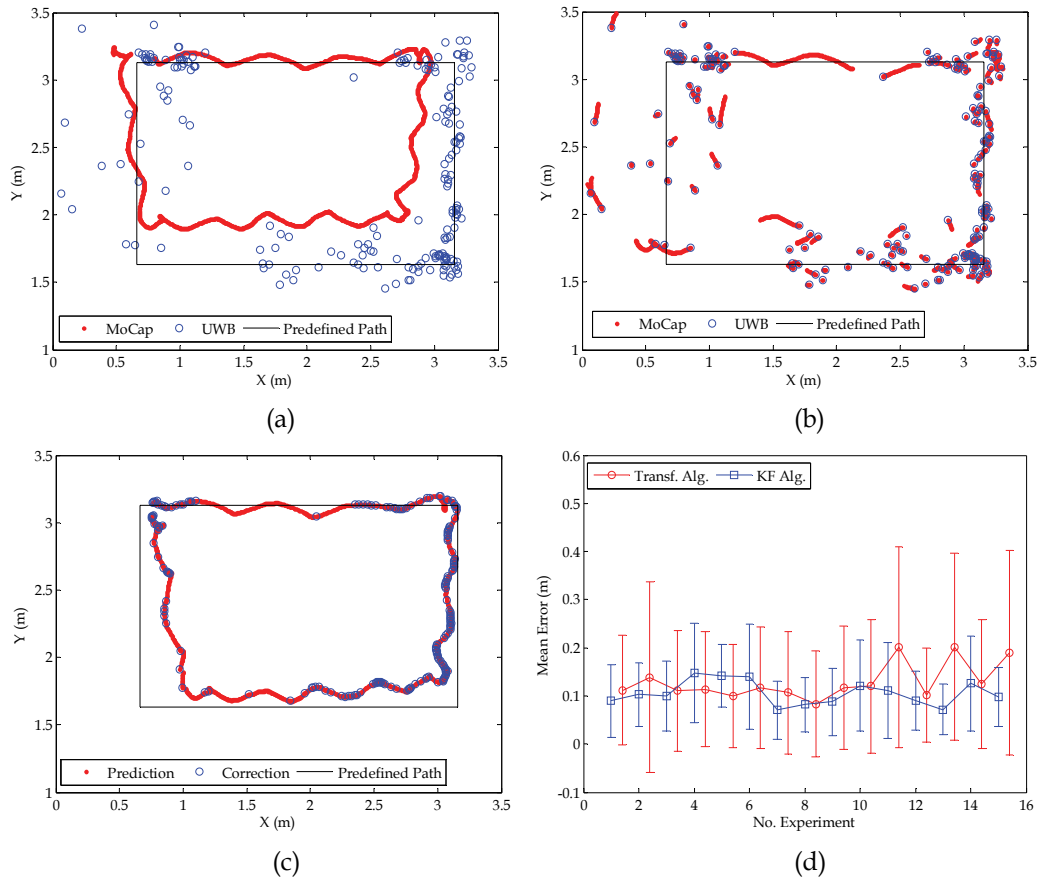
Fig. 6. Rectangular trajectory experiments: (a) original position measurements from both tracking systems in experiment no. 3; (b) position measurements obtained from the transformation recalculation algorithm in experiment no. 3; (c) position estimates obtained from the prediction and correction steps of the Kalman filter algorithm in experiment no. 3; (d) mean error and standard deviation of the position estimates from the two fusion algorithms in the 15 rectangular path experiments.

4.2 Application in a human-robot interaction task

The global position measurements obtained from the Kalman filter fusion algorithm and the joint rotations registered by the motion capture system are applied to the skeleton model of the human operator in order to perform human-robot interaction tasks. In Fig. 7, a collaboration task between a robotic manipulator (Mitsubishi PA-10) and a human operator is shown. The main frames of the task are depicted with the corresponding models of the human operator and the robot. In this task, the robotic manipulator has to remove two connectors from a metallic structure in order to leave them in a storage box. However, the collaboration of a human operator is needed because the unscrewing of the connectors is a difficult action to be performed by one robot.

Firstly, the operator unscrews the first connector (Fig. 7a and Fig. 7b). When the operator has finished the unscrewing process, the robotic manipulator begins to move towards the

structure by using a time-independent visual sevoing technique (Garcia et al., 2007) in order to grab the unscrewed connector (Fig. 7c). The end of the unscrewing process is determined by the tracking of the human operator. In particular, the robot does not begin its task until the distance between the end-effector of the robot and the operator is greater than a safety threshold (1m). While the robot is removing the first connector from the structure, the human operator unscrews the other connector (Fig. 7d). When the human operator has finished unscrewing the second connector, he leaves the robotic workspace (Fig. 7e). Finally, the manipulator places the first connector inside the storage box (Fig. 7f). The removing of the second connector has not been depicted in Fig.7 for the sake of clarity but it is very similar to the processing of the first connector.

| (a) | (b) | (c) |
| (d) | (e) | (f) |
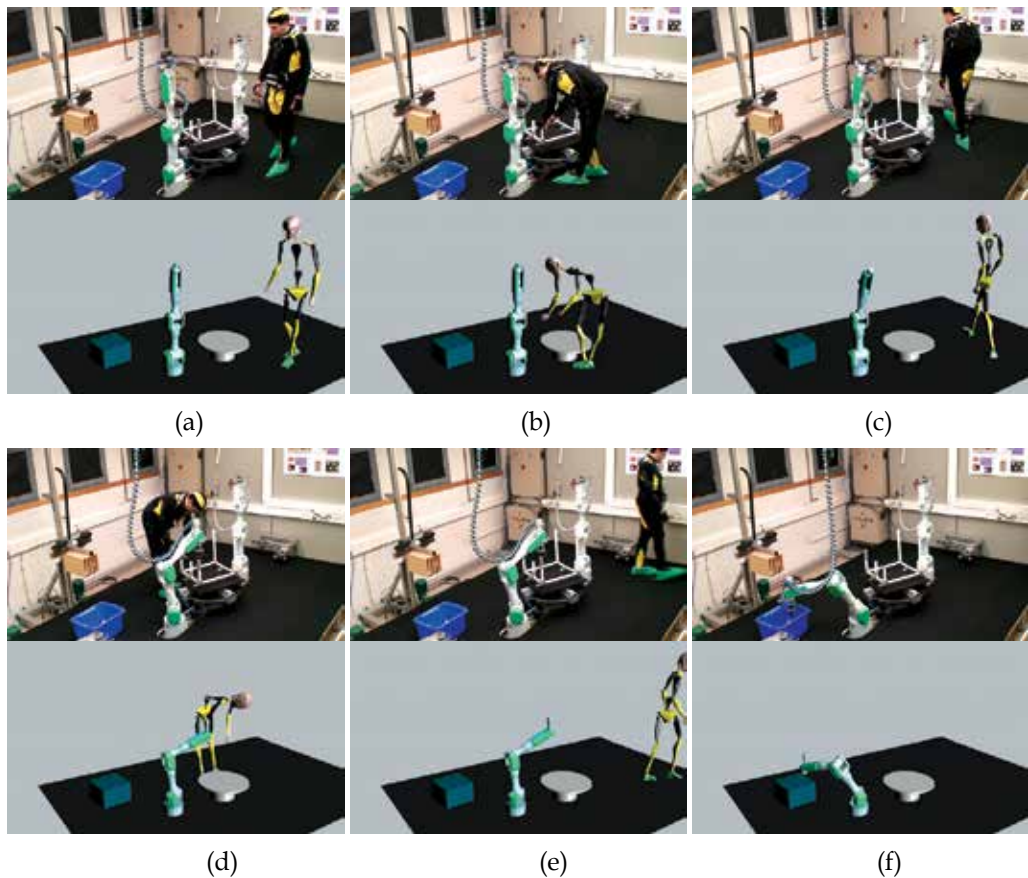
Fig. 7. Sequence of 6 frames of a human-robot interaction task. The skeleton of the human obtained from the tracking systems by using the presented Kalman filter algorithm and the model of the robot are depicted below each frame.

5. Conclusions

In this chapter, two different algorithms have been developed to combine position measurements from two human tracking systems: an inertial motion capture system and a UWB location system. The first algorithm recalculates the transformation matrix between

the coordinate systems of both trackers each time a UWB measurement is received. The error accumulated by the inertial motion capture system is therefore removed because the following measurements are transformed to the UWB frame with this new transformation matrix. However, this approach considers the UWB measurements as completely correct data and ignores their errors. The errors of the UWB system are not corrected and they are transferred to the measurements obtained from the fusion algorithm.

The second fusion algorithm is based on a Kalman filter and solves the drawbacks of the first algorithm. The Kalman filter algorithm models the errors of both tracking systems and takes them into account to estimate the position of the human operator. Several experiments have been developed to verify that this algorithm based on a Kalman filter obtains more accurate position estimates and more continuous trajectories than the first algorithm. In this algorithm, the prediction step of the Kalman filter is executed when measurements from the inertial motion capture system are received and the correction step is only executed when measurements from the UWB system are obtained. Thereby, this Kalman filter algorithm has a better computational cost than previous Kalman filter fusion algorithms which complete both steps of the filter (prediction and correction) each time a new measurement is received.

Finally, the position estimates obtained from the Kalman filter algorithm and the rotations from the motion capture system have been applied to a human skeleton model in order to develop a real collaborative task between a human operator and a robotic manipulator. In this task, the real-time tracking of the human operator does not only guarantee the safety of the operator but also determines the behaviour of the robot. Thus, the robot does not start its trajectory until the human has finished the previous task and is at a safety distance from the robot.

In future work, the movements of the skeleton should be interpreted in order to understand behaviours of the human operator and thus develop more complex collaboration tasks between robots and humans.

6. Acknowledgments

7. References

Animazoo (2008). GypsyGyro-18 Motion Capture System. http://www.animazoo.com

Caron, F., Duflos, E., Pomorski, D. & Vanheeghe, P. (2006). GPS/IMU Data Fusion Using Multisensor Kalman Filtering: Introduction of Contextual Aspects. *Information Fusion,* 7, 2, 221-230.

Fox, V., Hightower, J., Lin, L., Schulz, D. & Borriello, G. (2003). Bayesian filtering for location estimation. *IEEE Pervasive Computing,* 2, 3, 24-33.

Foxlin, E. (1996). Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter, *Proceedings of IEEE Virtual Reality Annual International Symposium*, pp. 185-194, Santa Clara, California.

Foxlin, E. (2005). Pedestrian Tracking with Shoe-Mounted Inertial Sensors. *IEEE Computer Graphics and Applications,* 25, 6, 38-46.

Foxlin, E., Harrington, M. & Altshuler, Y. (1998). Miniature 6-DOF Inertial System for Tracking HMDs, *Proceedings of Helmet and Head-Mounted Displays III*, pp. 214-228, Orlando, FL, USA.

Garcia, G. J., Pomares, J. & Torres, F. (2007). A New Time-Independent Image Path Tracker to Guide Robots using Visual Servoing, *Proceedings of 12th IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 957-964, Patras, Greece.

Liu, H., Darabi, H., Banerjee, P. & Liu, J. (2007). Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews,* 37, 6, 1067-1080.

Roetenberg, D., Slycke, P. J. & Veltink, P. H. (2007). Ambulatory Position and Orientation Tracking Fusing Magnetic and Inertial Sensing. *IEEE Transactions on Biomedical Engineering,* 54, 5, 883-890.

Simon, D. (2001). Kalman Filtering. *Embedded Systems Programming,* 14, 6, 72-79.

Ubisense (2007). *System Overview White Paper*, Ubisense AG, Chesterton, UK.

Ubisense (2008). Ubisense Real-Time Localization System. http://www.ubisense.net

Welch, G. & Foxlin, E. (2002). Motion Tracking: No Silver Bullet but a Respectable Arsenal. *IEEE Computer Graphics and Applications*, 22, 6, 24-38.

You, S. & Neumann, U. (2001). Fusion of Vision and Gyro Tracking for Robust Augmented Reality Registration, *Proceedings of IEEE Virtual Reality 2001 Conference*, pp. 71-78, Yokohama, Japan.

Robust Kalman Filter with Application to State Estimation of a Nuclear Reactor

Zhe Dong

Institute of Nuclear and New Energy Technology (INET), Tsinghua University
People's Republic of China

1. Introduction

State estimation algorithm deals with recovering some desired state variables of a dynamic system from available noisy measurements, and estimation of the state variables is one of the fundamental and significant problems in control and signal processing areas, and many significant progresses have been made in this area. In 1940s, Wiener, the founder of the modern statistical estimation theory, established the Wiener filtering theory which solves the minimum variance estimation problem for stationary stochastic processes. It was not until late 1950s and early 1960s that Kalman filtering theory, a novel recursive filtering algorithm, was developed by Kalman and Bucy which did not require the stationarity assumption (Kalman, 1960; Kalman & Bucy, 1961). Since Kalman filter theory is only applicable for linear systems and almost all practical dynamic systems are nonlinear, Bucy and some other researchers were engaged in extending Kalman filtering theory to nonlinear systems in the following 10 years (Bucy & Senne, 1971). The most celebrated and widely used nonlinear filtering algorithm is the extended Kalman filter (EKF), which is essentially a suboptimal nonlinear filter. The key idea of the EKF is using the linearized dynamic model to calculate the covariance and gain matrices of the filter. The Kalman filter (KF) and the EKF are all widely used in many engineering areas, such as aerospace, chemical and mechanical engineering.

However, it is well known that both the KF and EKF are not robust against modelling uncertainties and disturbances. This has motivated many studies on extending the Kalman filtering theory to its robust version, which may yield a suboptimal state estimator with respect to the nominal system model, but will guarantee an upper bound to the estimation error variance in spite of large disturbances and modelling uncertainties. In recent years, several results have been made on the design of such robust state estimators. Based on the fuzzy dynamic programming technique, a bounding estimator (Jian, 1975) for uncertain nonlinear systems was developed, which gave an upper bound to the error for any allowed system parameter variation. Petersen & McFarlane (1991, 1994) converted a robust Kalman filter design problem to a H_∞ controller design problem of a certain linear continuous time-invariant system based on the concept of quadratic stability of linear continuous time-invariant systems. Xie & Soh (1994) converted the existence of robust Kalman filters to the existence of two differential Riccati equations, and then established a design approach of robust estimators for linear continuous time systems with uncertainties in the state and the

output matrices. Yang & Wang (2001) presented a robust Kalman filter design approach for uncertain linear continuous time-invariant systems with norm-bounded uncertainties not only in the state and the output matrices bus also in the estimator's gain matrix, and this design approach were given in terms of solutions to algebraic Riccati equations. The continuous-time case was fully discussed in Shaked & de Souza (1995) where both the finite and infinite-horizon filtering problems were addressed, and necessary and sufficient conditions for the existence of robust Kalman filters with an optimized upper bound of the error variance were established.

In most engineering applications, both the KF and the EKF are implemented as program codes in computers. Therefore, the design of robust estimators for uncertain discrete-time systems is more important. Recently, there have been some promising results in the design of discrete-time robust Kalman filters. Xie, Petersen and Shaked extended their results on the design of continuous-time Kaman filters to that of discrete-time Kalman filters (Xie et al., 1994; Petersen & McFarlane, 1996; Theodor & Shaked, 1996). Fu, de Souza and Luo (2000) presented a design approach for finite-horizon robust Kalman filters, and the scalar parameters of the robust Kalman filter can be obtained by solving a semidefinite program. Shaked, Xie and Soh (2001) established a new robust Kalman filter design approach through which all the parameters of the designed robust Kalman filter are given by solving a semidefinite program subjected to a set of linear matrix inequalities (LMIs). Zhu, Soh and Xie (2002) gave necessary and sufficient conditions to the design properties of the robust Kalman filters over finite and infinite horizon in terms of a pair of parameterized difference Riccati equation. Garcia, Tabouriech and Peres (2003) showed that if the original systems satisfied some particular structural conditions, such as minimum-phase, and if the uncertainty had a specific structure, then the formula of the robust Kalman filter only involved the original system matrices. The discrete-time systems discussed in the aforementioned papers are all linear discrete-time systems with uncertainty only in the state and the output matrices. Very recently, results on some new types of linear uncertain discrete-time systems have also been given. Yang, Wang and Hung (2002) presented a design approach of a robust Kalman filter for linear discrete time-varying systems with multiplicative noises. Since the covariance matrices of the noises cannot be known precisely, Dong and You (2006) derived a finite-horizon robust Kalman filter for linear time-varying systems with norm-bounded uncertainties in the state matrix, the output matrix and the covariance matrices of noises. Based on the techniques presented in Zhu, Soh and Xie (2002), Lu, Xie and *et al.* (2007) gave a robust Kalman filter design approach for the linear discrete-time systems with measurement delay and norm-bounded uncertainty in the state matrix. Hounkpevi and Yaz (2007) proposed a robust Kalman filter for linear discrete-time systems with sensor failures and norm-bounded uncertainty in the state matrix. Though there have been many types of promising robust Kalman filters, new engineering demand can stimulate the birth of novel robust Kalman filters.

From the viewpoint of the robust state estimator design based on the dynamical model for a nuclear reactor, there is not only uncertainty in the state and the output matrices but also nonlinear perturbation and uncertainty in the input and the direct output matrices. The robust Kalman filters presented in the aforementioned papers cannot deal with this case. In this chapter, a finite-horizon robust Kalman filter for discrete-time systems with nonlinear perturbation and norm-bounded uncertainties in the state, the output, the input and the direct output matrices is presented. Moreover, the robust Kalman filter is applied to solve

the state-estimation problem of a low temperature pressurized water reactor (LTPWR), and a numerical experiment with a contrast to the EKF is done. Simulation results show that the state estimation performance provided by the robust Kalman filter is higher than that provided by the EKF.

This chapter is organized as follows: In Section 2, the model of the discrete-time system with nonlinear perturbation and norm bounded uncertainties in all the system matrices is presented, and the corresponding robust Kalman filtering problem is formulated. This problem is solved, for the finite-horizon case, in Section 3. In Section 4, the dynamic model of the aforementioned LTPWR is introduced, and then numerical simulation results are given to demonstrate the feasibility of applying the new robust Kalman filter to the state estimation for the LTPWR and the high performance of this estimator.

2. Problem formulation

Consider the following class of uncertain discrete time-varying systems defined on $k = 0, 1, \ldots, N$:

$$
\begin{aligned}
x_{k+1} &= \left(A_k + \Delta A_k\right)x_k + \left(B_k + \Delta B_k\right)w_k + f_k\left(x_k\right) \\
y_k &= \left(C_k + \Delta C_k\right)x_k + \left(D_k + \Delta D_k\right)v_k
\end{aligned}
\tag{1}
$$

where $x_k \in \mathbb{R}^n$ is the state vector, $y_k \in \mathbb{R}^p$ is the system output, $w_k \in \mathbb{R}^m$ is the process noise, $v_k \in \mathbb{R}^q$ is the measurement noise, A_k, B_k, C_k and D_k are given real time-varying matrices with proper dimensions, the matrices ΔA_k, ΔB_k, ΔC_k and ΔD_k correspond to the uncertainties in the matrices A_k, B_k, C_k and D_k respectively, and $f_k(x_k)$ denotes the nonlinear perturbation.

Before formulating the robust Kalman filter design problem, we firstly make the following three assumptions to system (1).

Assumption 1 Suppose the uncertainty matrices ΔA_k, ΔB_k, ΔC_k and ΔD_k satisfying the following condition

$$
\begin{bmatrix} \Delta A_k & \Delta B_k \\ \Delta C_k & \Delta D_k \end{bmatrix} = \begin{bmatrix} H_{1,k} \\ H_{2,k} \end{bmatrix} F_k \begin{bmatrix} E_{1,k} & E_{2,k} \end{bmatrix}
\tag{2}
$$

where $H_{1,k} \in \mathbb{R}^{n \times r}$, $H_{2,k} \in \mathbb{R}^{p \times r}$, $E_{1,k} \in \mathbb{R}^{s \times n}$ and $E_{2,k} \in \mathbb{R}^{s \times m}$ are given matrices, and $F_k \in \mathbb{R}^{r \times s}$ denotes the norm-bounded time-varying uncertainty satisfying

$$
F_k^{\mathrm{T}} F_k \leq I .
\tag{3}
$$

Assumption 2 Suppose the nonlinear perturbation $f_k(x_k)$ satisfying

$$
f_k(O) = O .
\tag{4}
$$

Therefore, from equation (4), the stochastic vector $f_k(x_k)$ can be expressed as

$$
f_k(x_k) = P_k(x_k)x_k .
\tag{5}
$$

Moreover, we suppose that the stochastic matrix $P_k(x_k)$ satisfies

$$-P \le P_k\left(x_k\right) \le P \,, \tag{6}$$

where the matrix P is a given positive definite symmetric matrix.

Assumption 3 Suppose the process noise w_k, the measurement noise v_k and the initial state vector x_0 have the following statistical properties:

$$\begin{cases} \mathrm{E}\left\{\begin{bmatrix} w_k \\ v_k \\ x_0 \end{bmatrix}\right\} = \begin{bmatrix} O \\ O \\ \bar{x}_0 \end{bmatrix}, \\ E\left\{\begin{bmatrix} w_k \\ v_k \\ x_0 \end{bmatrix}\begin{bmatrix} w_j \\ v_j \\ x_0 \end{bmatrix}^{\mathrm{T}}\right\} = \begin{bmatrix} W_k\delta_{kj} & O & O \\ O & V_k\delta_{kj} & O \\ O & O & X_0 \end{bmatrix}. \end{cases} \tag{7}$$

where $\mathrm{E}\{\bullet\}$ stands for the mathematical expectation operator, W_k, V_k and X_0 represents the covariance matrices of the process noise, the measurement noise and the initial state vector respectively, δ_{kj} is the Kronecker function which is equal to unity for $k=j$ and zero for the other case.

Moreover, let the state estimator of system (1) take the form

$$\hat{x}_{k+1} = A_{ok}\hat{x}_k + K_{ok}\left(y_k - C_k\hat{x}_k\right) \tag{8}$$

where $\hat{x}_k \in \mathbb{R}^n$ is the state estimation with $\hat{x}_0 = O$, A_{ok} and K_{ok} are respectively the state and the gain matrix. In the following part, we shall design a finite-horizon Kalman filter for structure (8), such that for all allowed nonlinear perturbation and model uncertainty, there exists a sequence of symmetric positive-definite matrices $\bar{\Sigma}_k$ satisfying

$$\mathrm{E}\left[\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^{\mathrm{T}}\right] \le \bar{\Sigma}_k \,. \tag{9}$$

Since

$$\mathrm{E}\left[\left(x_k - \hat{x}_k\right)^{\mathrm{T}}\left(x_k - \hat{x}_k\right)\right] \le \mathrm{tr}\left(\bar{\Sigma}_k\right), \tag{10}$$

the robust Kalman filter design problem to be solved in the next section is

$$A_{ok}, K_{ok} = \arg \min_{A_{ok}, K_{ok}} \mathrm{tr}\left(\bar{\Sigma}_k\right)$$
$$\text{s.t. (1), (2), (5), (6), (7), (8) and (9)}. \tag{11}$$

3. Robust Kalman filter design

In this section the robust Kalman filter for the uncertain system (1) is presented. Firstly, the state space model of the augmented system consisting of system (1) and estimator (8) will be established.

Define the state vector for the augmented system as

$$\tilde{x}_k = \begin{bmatrix} x_k^{\mathrm{T}} & \hat{x}_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$

then the state-space model of the augmented system can be represented as

$$\tilde{x}_{k+1} = \left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)\tilde{x}_k + \left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k}\right)\tilde{w}_k + \tilde{f}_k\left(\tilde{x}_k\right) \tag{12}$$

where $\quad \tilde{A}_k = \begin{bmatrix} A_k & O \\ K_{ok}C_k & A_{ok} - K_{ok}C_k \end{bmatrix}, \quad \tilde{B}_k = \begin{bmatrix} B_k & O \\ O & K_{ok}D_k \end{bmatrix}, \quad \tilde{H}_{1,k} = \begin{bmatrix} H_{1,k} \\ K_{ok}H_{2,k} \end{bmatrix}, \quad \tilde{E}_{1,k} = \begin{bmatrix} E_{1,k} & O \end{bmatrix},$

$\tilde{H}_{2,k} = \begin{bmatrix} H_{1,k} & O \\ O & K_{ok}H_{2,k} \end{bmatrix}, \quad \tilde{F}_k = \begin{bmatrix} F_k & O \\ O & F_k \end{bmatrix}, \quad \tilde{w}_k = \begin{bmatrix} w_k \\ v_k \end{bmatrix}, \tilde{f}_k\left(\tilde{x}_k\right) = \begin{bmatrix} f_k\left(x_k\right) \\ O \end{bmatrix}.$

Let the covariance matrix of the augmented system (12) to be $\tilde{\Sigma}_k = \mathrm{E}\left[\tilde{x}_k\tilde{x}_k^{\mathrm{T}}\right]$, then we can derive that

$$\tilde{\Sigma}_{k+1} = \left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)\tilde{\Sigma}_k\left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)^{\mathrm{T}} + \left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k}\right)\tilde{W}_k\left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k}\right)^{\mathrm{T}} +$$

$$\left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)\mathrm{E}\left[\tilde{x}_k\tilde{f}_k^{\mathrm{T}}\left(\tilde{x}_k\right)\right] + \mathrm{E}\left[\tilde{f}_k\left(\tilde{x}_k\right)\tilde{x}_k^{\mathrm{T}}\right]\left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)^{\mathrm{T}} + \mathrm{E}\left[\tilde{f}_k\left(\tilde{x}_k\right)\tilde{f}_k^{\mathrm{T}}\left(\tilde{x}_k\right)\right] \tag{13}$$

where

$$\tilde{W}_k = \begin{bmatrix} W_k & O \\ O & V_k \end{bmatrix}.$$

From Assumption 2,

$$\tilde{\Sigma}_{k+1} = \left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)\tilde{\Sigma}_k\left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)^{\mathrm{T}} + \left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k}\right)\tilde{W}_k\left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k}\right)^{\mathrm{T}} +$$

$$\left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)\tilde{\Sigma}_k\tilde{G}_k^{\mathrm{T}}\tilde{P}^{\mathrm{T}} + \tilde{P}\tilde{G}_k\tilde{\Sigma}_k\left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k}\right)^{\mathrm{T}} + \tilde{P}\tilde{G}_k\tilde{\Sigma}_k\tilde{G}_k^{\mathrm{T}}\tilde{P}^{\mathrm{T}}$$

$$= \left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k} + \tilde{P}\tilde{G}_k\right)\tilde{\Sigma}_k\left(\tilde{A}_k + \tilde{H}_{1,k} F_k \tilde{E}_{1,k} + \tilde{P}\tilde{G}_k\right)^{\mathrm{T}} +$$

$$\left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k}\right)\tilde{W}_k\left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k}\right)^{\mathrm{T}}, \tag{14}$$

where

$$\tilde{P} = \begin{bmatrix} P & O \\ O & O \end{bmatrix},$$

$$\tilde{G}_k = \begin{bmatrix} G_k & O \\ O & O \end{bmatrix},$$

and the matrix G_k satisfies $G_k^{\mathrm{T}}G_k \le I$. Moreover, if we define

$$\breve{H}_{1,k} = \begin{bmatrix} H_{1,k} & P \\ K_{ok}H_{2,k} & O \end{bmatrix},$$

$$\breve{F}_k = \begin{bmatrix} F_k & O \\ O & G_k \end{bmatrix},$$

and

$$\breve{E}_{1,k} = \begin{bmatrix} E_{1,k} & O \\ I & O \end{bmatrix},$$

then $\tilde{\Sigma}_{k+1}$ can be rewritten as

$$\tilde{\Sigma}_{k+1} = \left(\tilde{A}_k + \breve{H}_{1,k}\breve{F}_k\breve{E}_{1,k} \right)\tilde{\Sigma}_k\left(\tilde{A}_k + \breve{H}_{1,k}\breve{F}_k\breve{E}_{1,k} \right)^{\mathrm{T}} + \left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k} \right)\tilde{W}_k\left(\tilde{B}_k + \tilde{H}_{2,k}\tilde{F}_k\tilde{E}_{2,k} \right)^{\mathrm{T}} \quad (15)$$

For the convenience of the following discussion, we now recall some useful lemmas.

Lemma 1 (Xie & de Souza, 1993) For given matrices A, H, E and F with compatible dimensions such that $F^{\mathrm{T}}F \leq I$, let X be a symmetric positive-definite matrix and $\alpha > 0$ be an arbitrary positive constant such that $\alpha^{-1}I - EXE^{\mathrm{T}} > O$, then the following matrix inequality holds:

$$(A + HFE)X(A + HFE)^{\mathrm{T}} \leq AXA^{\mathrm{T}} + AXE^{\mathrm{T}}\left(\alpha^{-1}I - EXE^{\mathrm{T}} \right)^{-1}EXA^{\mathrm{T}} + \alpha^{-1}HH^{\mathrm{T}}.$$

Lemma 2 (Theodor & Shaked, 1996) Let $f_k(\cdot): \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$, $k = 0, 1, \ldots, N$, be a sequence of matrix functions so that

$$f_k(A) = f_k^{\mathrm{T}}(A), \forall A = A^{\mathrm{T}} > 0$$

and

$$f_k(B) \geq f_k(A), \forall B = B^{\mathrm{T}} > A = A^{\mathrm{T}}$$

and let $g_k(\cdot): \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$, $k = 0, 1, \ldots, N$, be a sequence of matrix functions so that

$$g_k(A) = g_k^{\mathrm{T}}(A) \geq f_k(A), \forall A = A^{\mathrm{T}} > 0.$$

Then, the solutions $\{A_k\}_{k=0,1,\cdots,N}$ and $\{B_k\}_{k=0,1,\cdots,N}$ to the difference equations

$$A_{k+1} = f_k(A_k), B_{k+1} = g_k(B_k), A_0 = B_0 > 0$$

satisfy $A_k \leq B_k$ for $k = 0, 1, \ldots, N$.

Lemma 3 (Matrix Inversion Lemma) (Anderson & Moore, 1979) For given matrices A, B, C and D with compatible dimensions where $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{m \times m}$ are all nonsingular submatrices, then

$$(A \pm BCD)^{-1} = A^{-1} \mp A^{-1}B\left(C^{-1} \pm DA^{-1}B \right)^{-1}DA^{-1}.$$

Lemma 4 (Matrix Inversion Lemma for Matrices in Block Form) (Chui & Chen, 1999) Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where $A_{11} \in \mathbb{R}^{n \times n}$ and $A_{22} \in \mathbb{R}^{m \times m}$ are all nonsingular submatrices, such that $A_{11} - A_{12}A_{22}^{-1}A_{21}$ and $A_{22} - A_{21}A_{11}^{-1}A_{12}$ are also nonsingular. Then A is a nonsingular with

$$A^{-1} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^{-1} \\ -\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^{-1}A_{21}A_{11}^{-1} & \left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} \left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right)^{-1} & -\left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right)^{-1}A_{12}A_{22}^{-1} \\ -A_{22}^{-1}A_{21}\left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right)^{-1} & A_{22}^{-1} + A_{22}^{-1}A_{21}\left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right)^{-1}A_{12}A_{22}^{-1} \end{bmatrix}.$$

*Lemma 5 (Schur Complement) (*Boyd, Ghaoui & *et al.*, 1994) For a given symmetric matrix

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{12}^{\mathrm{T}} & S_{22} \end{bmatrix},$$

where $S \in \mathbb{R}^{(n+m)\times(n+m)}$, $S_{11} \in \mathbb{R}^{n\times n}$ and $S_{22} \in \mathbb{R}^{m\times m}$, the following statements are equivalent:

i. $S < O$

ii. $\begin{cases} S_{11} < O \\ S_{22} - S_{12}^{\mathrm{T}}S_{11}^{-1}S_{12} < O \end{cases}$

iii. $\begin{cases} S_{22} < O \\ S_{11} - S_{12}S_{22}^{-1}S_{12}^{\mathrm{T}} < O \end{cases}$

Based on the state-space model of the augmented system, Lemma 1 and Lemma 2, we can get the following Theorem 1.

Theorem 1 If there exist a symmetric positive-definite matrix sequence Σ_k, and two positive scalar sequences α_k and β_k such that

$$\alpha_k^{-1}I - \breve{E}_{1,k}\tilde{\Sigma}_k\breve{E}_{1,k}^{\mathrm{T}} > O , \tag{16}$$

$$\beta_k^{-1}I - \tilde{E}_{2,k}\tilde{W}_k\tilde{E}_{2,k}^{\mathrm{T}} > O , \tag{17}$$

$$\begin{aligned}\Sigma_{k+1} = &\tilde{A}_k\Sigma_k\tilde{A}_k^{\mathrm{T}} + \tilde{A}_k\Sigma_k\breve{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I - \breve{E}_{1,k}\Sigma_k\breve{E}_{1,k}^{\mathrm{T}}\right)^{-1}\breve{E}_{1,k}\Sigma_k\tilde{A}_k^{\mathrm{T}} + \alpha_k^{-1}\breve{H}_{1,k}\breve{H}_{1,k}^{\mathrm{T}} + \beta_k^{-1}\tilde{H}_{2,k}\tilde{H}_{2,k}^{\mathrm{T}} + \\ &\tilde{B}_k\tilde{W}_k\tilde{B}_k^{\mathrm{T}} + \tilde{B}_k\tilde{W}\tilde{E}_{2,k}^{\mathrm{T}}\left(\beta_k^{-1}I - \tilde{E}_{2,k}\tilde{W}_k\tilde{E}_{2,k}^{\mathrm{T}}\right)^{-1}\tilde{E}_{2,k}\tilde{W}\tilde{B}_k^{\mathrm{T}} ,\end{aligned} \tag{18}$$

and

$$\Sigma_0 = \tilde{\Sigma}_0 = \begin{bmatrix} X_0 & O \\ O & O \end{bmatrix}, \tag{19}$$

then

$$\tilde{\Sigma}_k \le \Sigma_k , k = 0,1,\cdots,N , \tag{20}$$

where $\tilde{\Sigma}_k$ satisfies equation (15).

Proof: If matrix inequalities (16) and (17) hold, then from Lemma 1

$$\begin{aligned}\tilde{\Sigma}_{k+1} \le &\left(\tilde{A}_k + \breve{H}_{1,k}\breve{F}_k\breve{E}_{1,k}\right)\left[\tilde{\Sigma}_k + \tilde{\Sigma}_k\breve{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I - \breve{E}_{1,k}\tilde{\Sigma}_k\breve{E}_{1,k}^{\mathrm{T}}\right)^{-1}\breve{E}_{1,k}\tilde{\Sigma}_k\right]\left(\tilde{A}_k + \breve{H}_{1,k}\breve{F}_k\breve{E}_{1,k}\right)^{\mathrm{T}} + \\ &\tilde{B}_k\left[\tilde{W}_k + \tilde{W}_k\tilde{E}_{2,k}^{\mathrm{T}}\left(\beta_k^{-1}I - \tilde{E}_{2,k}\tilde{W}_k\tilde{E}_{2,k}^{\mathrm{T}}\right)^{-1}\tilde{E}_{2,k}\tilde{W}_k\right]\tilde{B}_k^{\mathrm{T}} + \alpha_k^{-1}\breve{H}_{1,k}\breve{H}_{1,k}^{\mathrm{T}} + \beta_k^{-1}\tilde{H}_{2,k}\hat{H}_{2,k}^{\mathrm{T}} .\end{aligned}$$

Moreover, from the initial condition (19) and Lemma 2, $\tilde{\Sigma}_k \le \Sigma_k, k = 0,1,\cdots,N$, where $\Sigma_k, k = 0,1,\cdots,N$, satisfies equation (18). Thus the theorem has been proved.   □

Suppose

$$\bar{\Sigma}_k = \begin{bmatrix} I & -I \end{bmatrix} \Sigma_k \begin{bmatrix} I & -I \end{bmatrix}^{\mathrm{T}}, k = 0,1,\cdots,N.$$

It is clear that the matrix sequence $\bar{\Sigma}_k (k = 0,1,\cdots,N)$ satisfies matrix inequalities (9) and (10). In the following, we shall show how to determine the parameters A_{ok} , K_{ok} , α_k and β_k in order to minimize $\text{tr}(\bar{\Sigma}_k)$. The following Theorem 2 solves the problem of how to select the state matrix A_{ok} and the gain matrix K_{ok} of the robust Kalman filter taking the form (8) when the positive scalar sequences α_k and β_k are given.

Theorem 2 For given positive scalar sequences α_k and β_k which satisfy inequalities (16) and (17) respectively, if

$$A_{ok} = A_k + \left(A_k - K_{ok} C_k \right) \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \tag{21}$$

and

$$K_{ok} = \left(A_k S_k C_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{2,k}^{\mathrm{T}} \right) \left[C_k S_k C_k^{\mathrm{T}} + D_k T_k D_k^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1} \right) H_{2,k} H_{2,k}^{\mathrm{T}} \right]^{-1}, \tag{22}$$

then

$$\Sigma_n = \begin{bmatrix} \Sigma_{1,n} & \Sigma_{2,n} \\ \Sigma_{2,n} & \Sigma_{2,n} \end{bmatrix}, n = 0,1,\cdots,N \tag{23}$$

and $\text{tr}(\bar{\Sigma}_k)$ is minimal, where

$$\bar{H}_{1,k} = \begin{bmatrix} H_{1,k} & P \end{bmatrix},$$

$$\bar{H}_{2,k} = \begin{bmatrix} H_{2,k} & O \end{bmatrix},$$

$$\bar{E}_{1,k} = \begin{bmatrix} E_{1,k} \\ I \end{bmatrix},$$

$$S_k = \bar{\Sigma}_k + \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \bar{\Sigma}_k ,$$

and

$$T_k = V_k + V_k E_{2,k}^{\mathrm{T}} \left(\beta_k^{-1} I - E_{2,k} V_k E_{2,k}^{\mathrm{T}} \right)^{-1} E_{2,k} V_k .$$

Moreover, the covariance matrices of the state and the estimation error can be represented respectively as

$$
\begin{aligned}
\Sigma_{1,k+1} = {}& A_k \Sigma_{1,k} A_k^{\mathrm{T}} + A_k \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \Sigma_{1,k} A_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{1,k}^{\mathrm{T}} + \\
& B_k U_k B_k^{\mathrm{T}} + \beta_k^{-1} H_{1,k} H_{1,k}^{\mathrm{T}},
\end{aligned}
\tag{24}
$$

and

$$
\begin{aligned}
\bar{\Sigma}_{k+1} = {}& A_k \bar{\Sigma}_k A_k^{\mathrm{T}} - \left(A_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \hat{H}_{2,k}^{\mathrm{T}} \right) \left(\hat{C}_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \hat{R}_k \right)^{-1} \left(A_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \hat{H}_{2,k}^{\mathrm{T}} \right)^{\mathrm{T}} + \\
& + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{1,k}^{\mathrm{T}} + B_k U_k B_k^{\mathrm{T}} + \beta_k^{-1} H_{1,k} H_{1,k}^{\mathrm{T}},
\end{aligned}
\tag{25}
$$

where

$$
U_{k+1} = W_k + W_k E_{2,k}^{\mathrm{T}} \left(\beta_k^{-1} I - E_{2,k} W_k E_{2,k}^{\mathrm{T}} \right)^{-1} E_{2,k} W_k ,
$$

$$
\hat{C}_k = \begin{bmatrix} \bar{E}_{1,k} \\ C_k \end{bmatrix} ,
$$

$$
\hat{H}_{2,k} = \begin{bmatrix} O \\ \bar{H}_{2,k} \end{bmatrix} ,
$$

and

$$
\hat{R}_k = \begin{bmatrix} -\alpha_k^{-1} I & O \\ O & D_k T_k D_k^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1} \right) H_{2,k} H_{2,k}^{\mathrm{T}} \end{bmatrix} .
$$

Proof: In the following, the theorem will be proved based on the mathematical induction. From equation (19), it is clear that equation (23) holds when $n = 0$. Suppose (23) holds when $n = k$. In the following, we shall prove that equation (23) also holds when $n = k+1$. From equation (18),

$$
\Sigma_{k+1} = \begin{bmatrix} \Sigma_{1,k+1} & \Sigma_{12,k+1} \\ \Sigma_{12,k+1}^{\mathrm{T}} & \Sigma_{2,k+1} \end{bmatrix} ,
$$

where $\Sigma_{1,k+1}$ satisfies equation (24),

$$
\begin{aligned}
\Sigma_{12,k+1} = {}& A_k \left[\Sigma_{1,k} + \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \Sigma_{1,k} \right] C_k^{\mathrm{T}} K_{\mathrm{o}k}^{\mathrm{T}} + \\
& A_k \left[I + \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \right] \Sigma_{2,k} \left(A_{\mathrm{o}k} - K_{\mathrm{o}k} C_k \right)^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{2,k}^{\mathrm{T}} K_{\mathrm{o}k}^{\mathrm{T}} ,
\end{aligned}
\tag{26}
$$

and

$$
\begin{aligned}
\Sigma_{2,k+1} = {}& A_{\mathrm{o}k} \Sigma_{2,k} A_{\mathrm{o}k}^{\mathrm{T}} + K_{\mathrm{o}k} C_k \bar{\Sigma}_k C_k^{\mathrm{T}} K_{\mathrm{o}k}^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1} \right) K_{\mathrm{o}k} H_{2,k} H_{2,k}^{\mathrm{T}} K_{\mathrm{o}k}^{\mathrm{T}} + K_{\mathrm{o}k} D_k T_k D_k^{\mathrm{T}} K_{\mathrm{o}k}^{\mathrm{T}} + \\
& \left(A_{\mathrm{o}k} \Sigma_{2,k} + K_{\mathrm{o}k} C_k \bar{\Sigma}_k \right) \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \left(A_{\mathrm{o}k} \Sigma_{2,k} + K_{\mathrm{o}k} C_k \bar{\Sigma}_k \right)^{\mathrm{T}} .
\end{aligned}
\tag{27}
$$

Then, from equations (26) and (27), we can derive that

$$
\begin{aligned}
&\bar{\Sigma}_{k+1}\\
&=\begin{bmatrix} I & -I \end{bmatrix}\Sigma_{k+1}\begin{bmatrix} I & -I \end{bmatrix}^{\mathrm{T}}\\
&=A_k\left[\Sigma_{1,k}+\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\Sigma_{1,k}\right]A_k^{\mathrm{T}}+\alpha_k^{-1}\bar{H}_{1,k}\bar{H}_{1,k}^{\mathrm{T}}+\beta_k^{-1}H_{1,k}H_{1,k}^{\mathrm{T}}+\\
&\quad B_kU_kB_k^{\mathrm{T}}+A_{\mathrm{o}k}\Sigma_{2,k}A_{\mathrm{o}k}^{\mathrm{T}}+K_{\mathrm{o}k}C_k\bar{\Sigma}_kC_k^{\mathrm{T}}K_{\mathrm{o}k}^{\mathrm{T}}+\left(\alpha_k^{-1}+\beta_k^{-1}\right)K_{\mathrm{o}k}H_{2,k}H_{2,k}^{\mathrm{T}}K_{\mathrm{o}k}^{\mathrm{T}}+K_{\mathrm{o}k}D_kT_kD_k^{\mathrm{T}}K_{\mathrm{o}k}^{\mathrm{T}}+\\
&\quad \left(A_{\mathrm{o}k}\Sigma_{2,k}+K_{\mathrm{o}k}C_k\bar{\Sigma}_k\right)\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\left(A_{\mathrm{o}k}\Sigma_{2,k}+K_{\mathrm{o}k}C_k\bar{\Sigma}_k\right)^{\mathrm{T}}-\\
&\quad A_k\left[\Sigma_{1,k}+\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\Sigma_{1,k}\right]C_k^{\mathrm{T}}K_{\mathrm{o}k}^{\mathrm{T}}-\\
&\quad K_{\mathrm{o}k}C_k\left[\Sigma_{1,k}+\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\Sigma_{1,k}\right]A_k-\\
&\quad A_k\left[I+\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\right]\Sigma_{2,k}\left(A_{\mathrm{o}k}-K_{\mathrm{o}k}C_k\right)^{\mathrm{T}}-\\
&\quad \left(A_{\mathrm{o}k}-K_{\mathrm{o}k}C_k\right)\Sigma_{2,k}\left[I+\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\Sigma_{1,k}\right]A_k^{\mathrm{T}}-\\
&\quad \alpha_k^{-1}\left(\bar{H}_{1,k}\bar{H}_{2,k}^{\mathrm{T}}K_{\mathrm{o}k}^{\mathrm{T}}+K_{\mathrm{o}k}\bar{H}_{2,k}\bar{H}_{1,k}^{\mathrm{T}}\right)\\
&=\left(A_k-K_{\mathrm{o}k}C_k\right)\bar{\Sigma}_k\left(A_k-K_{\mathrm{o}k}C_k\right)^{\mathrm{T}}+\left(A_k-A_{\mathrm{o}k}\right)\Sigma_{2,k}\left(A_k-A_{\mathrm{o}k}\right)^{\mathrm{T}}+\\
&\quad \left(A_{\mathrm{o}k}\Sigma_{2,k}+K_{\mathrm{o}k}C_k\bar{\Sigma}_k-A_k\Sigma_{1,k}\right)\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\left(A_{\mathrm{o}k}\Sigma_{2,k}+K_{\mathrm{o}k}C_k\bar{\Sigma}_k-A_k\Sigma_{1,k}\right)^{\mathrm{T}}+\\
&\quad \alpha_k^{-1}\left(\bar{H}_{1,k}-K_{\mathrm{o}k}\bar{H}_{2,k}\right)\left(\bar{H}_{1,k}-K_{\mathrm{o}k}\bar{H}_{2,k}\right)^{\mathrm{T}}+\beta_k^{-1}K_{\mathrm{o}k}H_{2,k}H_{2,k}^{\mathrm{T}}K_{\mathrm{o}k}^{\mathrm{T}}+K_{\mathrm{o}k}D_kT_kD_k^{\mathrm{T}}K_{\mathrm{o}k}^{\mathrm{T}}+\\
&\quad B_kU_kB_k^{\mathrm{T}}+\beta_k^{-1}H_{1,k}H_{1,k}^{\mathrm{T}}. \tag{28}
\end{aligned}
$$

Since

$$
\frac{1}{2}\frac{\partial^2 \operatorname{tr}\left(\bar{\Sigma}_k\right)}{\partial A_{\mathrm{o}k}{}^2}=\Sigma_{2,k}+\Sigma_{2,k}\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\Sigma_{2,k}>O\,,
$$

and

$$
\frac{1}{2}\frac{\partial^2 \operatorname{tr}\left(\bar{\Sigma}_k\right)}{\partial K_{\mathrm{o}k}{}^2}=C_k\left[\bar{\Sigma}_k+\bar{\Sigma}_k\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\bar{\Sigma}_k\right]C_k^{\mathrm{T}}+\left(\alpha_k^{-1}+\beta_k^{-1}\right)H_{2,k}H_{2,k}^{\mathrm{T}}+D_kT_kD_k^{\mathrm{T}}>O\,,
$$

$\operatorname{tr}\left(\bar{\Sigma}_k\right)$ is minimal for the given α_k and β_k if and only if $\dfrac{\partial \operatorname{tr}\left(\bar{\Sigma}_k\right)}{\partial A_{\mathrm{o}k}}=O$ and $\dfrac{\partial \operatorname{tr}\left(\bar{\Sigma}_k\right)}{\partial K_{\mathrm{o}k}}=O$.

Moreover, we can derive that

$$
\begin{aligned}
\frac{1}{2}\frac{\partial \operatorname{tr}\left(\bar{\Sigma}_k\right)}{\partial A_{\mathrm{o}k}}=&\left(A_{\mathrm{o}k}-A_k\right)\left[I+\Sigma_{2,k}\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k}\right]\Sigma_{2,k}+\\
&\left(A_k-K_{\mathrm{o}k}C_k\right)\bar{\Sigma}_k\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I-\bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k},
\end{aligned}
$$

and

$$\frac{1}{2}\frac{\partial \mathrm{tr}\left(\bar{\boldsymbol{\Sigma}}_k\right)}{\partial \boldsymbol{K}_{\mathrm{o}k}} = \boldsymbol{K}_{\mathrm{o}k}\left\{\boldsymbol{C}_k\bar{\boldsymbol{\Sigma}}_k\boldsymbol{C}_k^{\mathrm{T}} + \boldsymbol{C}_k\bar{\boldsymbol{\Sigma}}_k\left[\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{2,k}\right]\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\cdot$$

$$\bar{\boldsymbol{E}}_{1,k}\bar{\boldsymbol{\Sigma}}_k\boldsymbol{C}_k^{\mathrm{T}} + \left(\alpha_k^{-1}+\beta_k^{-1}\right)\boldsymbol{H}_{2,k}\boldsymbol{H}_{2,k}^{\mathrm{T}} + \boldsymbol{D}_k\boldsymbol{T}_k\boldsymbol{D}_k^{\mathrm{T}}\Big\} - \boldsymbol{A}_k\Big\{\boldsymbol{I} + \bar{\boldsymbol{\Sigma}}_k\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\cdot$$

$$\left[\boldsymbol{I} - \boldsymbol{\Sigma}_{2,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{\boldsymbol{E}}_{1,k}\right]\Big\}\bar{\boldsymbol{\Sigma}}_k\boldsymbol{C}_k^{\mathrm{T}} - \alpha_k^{-1}\bar{\boldsymbol{H}}_{1,k}\bar{\boldsymbol{H}}_{2,k}^{\mathrm{T}}.$$

Therefore, from $\dfrac{\partial \mathrm{tr}\left(\bar{\boldsymbol{\Sigma}}_k\right)}{\partial \boldsymbol{A}_{\mathrm{o}k}} = \boldsymbol{O}$,

$$\boldsymbol{A}_{\mathrm{o}k} = \boldsymbol{A}_k + \left(\boldsymbol{A}_k - \boldsymbol{K}_{\mathrm{o}k}\right)\bar{\boldsymbol{\Sigma}}_k\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{\boldsymbol{E}}_{1,k}\left[\boldsymbol{I} + \boldsymbol{\Sigma}_{2,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{\boldsymbol{E}}_{1,k}\right]^{-1}.$$

From Lemma 3, we can obtain that $\boldsymbol{A}_{\mathrm{o}k}$ satisfies equation (21). Furthermore, from Lemma 3 and $\dfrac{\partial \mathrm{tr}\left(\bar{\boldsymbol{\Sigma}}_k\right)}{\partial \boldsymbol{K}_{\mathrm{o}k}} = \boldsymbol{O}$, we can see that $\boldsymbol{K}_{\mathrm{o}k}$ satisfies equation (22).

Substitute equations (21) and (22) to the expression of $\boldsymbol{\Sigma}_{12,k+1}$ and $\boldsymbol{\Sigma}_{2,k+1}$, i.e. (26) and (27),

$$\begin{aligned}\boldsymbol{\Sigma}_{12,k+1} &= \boldsymbol{A}_k\boldsymbol{M}_{1,k}\boldsymbol{\Sigma}_{1,k}\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{K}_{\mathrm{o}k}^{\mathrm{T}} + \boldsymbol{A}_k\boldsymbol{M}_{1,k}\boldsymbol{\Sigma}_{2,k}\bar{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{S}_k\left(\boldsymbol{A}_k - \boldsymbol{K}_{\mathrm{o}k}\boldsymbol{C}_k\right)^{\mathrm{T}} + \alpha_k^{-1}\bar{\boldsymbol{H}}_{1,k}\bar{\boldsymbol{H}}_{2,k}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{o}k}^{\mathrm{T}} \\ &= \left(\boldsymbol{A}_k\boldsymbol{S}_k\boldsymbol{C}_k^{\mathrm{T}}+\alpha_k^{-1}\bar{\boldsymbol{H}}_{1,k}\bar{\boldsymbol{H}}_{2,k}^{\mathrm{T}}\right)\left[\boldsymbol{C}_k\boldsymbol{S}_k\boldsymbol{C}_k^{\mathrm{T}} + \boldsymbol{D}_k\boldsymbol{T}_k\boldsymbol{D}_k^{\mathrm{T}} + \left(\alpha_k^{-1}+\beta_k^{-1}\right)\boldsymbol{H}_{2,k}\boldsymbol{H}_{2,k}^{\mathrm{T}}\right]^{-1}\cdot \\ &\quad \left(\boldsymbol{A}_k\boldsymbol{S}_k\boldsymbol{C}_k^{\mathrm{T}}+\alpha_k^{-1}\bar{\boldsymbol{H}}_{1,k}\bar{\boldsymbol{H}}_{2,k}^{\mathrm{T}}\right) + \boldsymbol{A}_k\boldsymbol{M}_{1,k}\boldsymbol{M}_{2,k}^{-1}\boldsymbol{\Sigma}_{2,k}\boldsymbol{M}_{1,k}^{\mathrm{T}}\boldsymbol{A}_k^{\mathrm{T}},\end{aligned} \qquad (29)$$

and

$$\begin{aligned}\boldsymbol{\Sigma}_{2,k+1} &= \boldsymbol{K}_{\mathrm{o}k}\boldsymbol{C}_k\left(\boldsymbol{M}_{1,k} - \boldsymbol{M}_{2,k}\right)\boldsymbol{M}_{2,k}^{-1}\bar{\boldsymbol{\Sigma}}_k\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{K}_{\mathrm{o}k}^{\mathrm{T}} + \boldsymbol{A}_k\boldsymbol{M}_{1,k}\boldsymbol{M}_{2,k}^{-1}\boldsymbol{\Sigma}_{2,k}\boldsymbol{M}_{1,k}^{\mathrm{T}}\boldsymbol{A}_k^{\mathrm{T}} + \\ &\quad \boldsymbol{K}_{\mathrm{o}k}\boldsymbol{C}_k\bar{\boldsymbol{\Sigma}}_k\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{K}_{\mathrm{o}k}^{\mathrm{T}} + \left(\alpha_k^{-1}+\beta_k^{-1}\right)\boldsymbol{H}_{2,k}\boldsymbol{H}_{2,k}^{\mathrm{T}} + \boldsymbol{K}_{\mathrm{o}k}\boldsymbol{D}_k\boldsymbol{T}_k\boldsymbol{D}_k^{\mathrm{T}}\boldsymbol{K}_{\mathrm{o}k}^{\mathrm{T}} \\ &= \left(\boldsymbol{A}_k\boldsymbol{S}_k\boldsymbol{C}_k^{\mathrm{T}}+\alpha_k^{-1}\bar{\boldsymbol{H}}_{1,k}\bar{\boldsymbol{H}}_{2,k}^{\mathrm{T}}\right)\left[\boldsymbol{C}_k\boldsymbol{S}_k\boldsymbol{C}_k^{\mathrm{T}} + \boldsymbol{D}_k\boldsymbol{T}_k\boldsymbol{D}_k^{\mathrm{T}} + \left(\alpha_k^{-1}+\beta_k^{-1}\right)\boldsymbol{H}_{2,k}\boldsymbol{H}_{2,k}^{\mathrm{T}}\right]^{-1}\cdot \\ &\quad \left(\boldsymbol{A}_k\boldsymbol{S}_k\boldsymbol{C}_k^{\mathrm{T}}+\alpha_k^{-1}\bar{\boldsymbol{H}}_{1,k}\bar{\boldsymbol{H}}_{2,k}^{\mathrm{T}}\right) + \boldsymbol{A}_k\boldsymbol{M}_{1,k}\boldsymbol{M}_{2,k}^{-1}\boldsymbol{\Sigma}_{2,k}\boldsymbol{M}_{1,k}^{\mathrm{T}}\boldsymbol{A}_k^{\mathrm{T}},\end{aligned} \qquad (30)$$

where

$$\boldsymbol{M}_{1,k} = \boldsymbol{I} + \boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{\boldsymbol{E}}_{1,k},$$

$$\boldsymbol{M}_{2,k} = \boldsymbol{I} + \boldsymbol{\Sigma}_{2,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\boldsymbol{\Sigma}_{1,k}\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{\boldsymbol{E}}_{1,k},$$

and

$$\boldsymbol{S}_k = \boldsymbol{M}_{1,k}\boldsymbol{M}_{2,k}^{-1}\bar{\boldsymbol{\Sigma}}_k = \bar{\boldsymbol{\Sigma}}_k + \bar{\boldsymbol{\Sigma}}_k\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}\boldsymbol{I} - \bar{\boldsymbol{E}}_{1,k}\bar{\boldsymbol{\Sigma}}_k\bar{\boldsymbol{E}}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{\boldsymbol{E}}_{1,k}\bar{\boldsymbol{\Sigma}}_k.$$

Thus $\Sigma_{12,k+1} = \Sigma_{2,k+1}$, and equation (23) holds when $n = k+1$.

In the following, we shall derive the iterative expression for the covariance matrix of the state estimation error. Based on Lemma 4 and equations (24) and (30), we can derive that

$$
\begin{aligned}
\bar{\Sigma}_{k+1} &= A_k S_k A_k^{\mathrm{T}} - \left(A_k S_k C_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{2,k}^{\mathrm{T}} \right) \left[C_k S_k C_k^{\mathrm{T}} + D_k T_k D_k^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1} \right) H_{2,k} H_{2,k}^{\mathrm{T}} \right]^{-1} \cdot \\
&\quad \left(A_k S_k C_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{2,k}^{\mathrm{T}} \right)^{\mathrm{T}} + B_k U_k B_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{1,k}^{\mathrm{T}} + \beta_k^{-1} H_{1,k} H_{1,k}^{\mathrm{T}} \\
&= A_k \bar{\Sigma}_k A_k^{\mathrm{T}} - \left[A_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \quad A_k \bar{\Sigma}_k C_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{2,k}^{\mathrm{T}} \right] \cdot \\
&\quad \left[\begin{matrix} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \left\{ -I + \bar{E}_{1,k} \bar{\Sigma}_k C_k^{\mathrm{T}} \left[C_k S_k C_k^{\mathrm{T}} + D_k T_k D_k^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1} \right) H_{2,k} H_{2,k}^{\mathrm{T}} \right]^{-1} \cdot \right. \\ \left. C_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \right\} \\ \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \bar{\Sigma}_k C_k^{\mathrm{T}} \left[C_k S_k C_k^{\mathrm{T}} + D_k T_k D_k^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1} \right) H_{2,k} H_{2,k}^{\mathrm{T}} + \right. \\ \left. C_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \bar{\Sigma}_k C_k^{\mathrm{T}} \right]^{-1} \\ \left[C_k S_k C_k^{\mathrm{T}} + D_k T_k D_k^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1} \right) H_{2,k} H_{2,k}^{\mathrm{T}} + C_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \bar{\Sigma}_k C_k^{\mathrm{T}} \right]^{-1} \cdot \\ C_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \\ \left[C_k S_k C_k^{\mathrm{T}} + D_k T_k D_k^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1} \right) H_{2,k} H_{2,k}^{\mathrm{T}} + C_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \left(\alpha_k^{-1} I - \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} \right)^{-1} \bar{E}_{1,k} \bar{\Sigma}_k C_k^{\mathrm{T}} \right]^{-1} \end{matrix} \right] \cdot \\
&\quad \left[A_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \quad A_k \bar{\Sigma}_k C_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{2,k}^{\mathrm{T}} \right]^{\mathrm{T}} + B_k U_k B_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{1,k}^{\mathrm{T}} + \beta_k^{-1} H_{1,k} H_{1,k}^{\mathrm{T}} \\
&= A_k \bar{\Sigma}_k A_k^{\mathrm{T}} - \left[A_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \quad A_k \bar{\Sigma}_k C_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{2,k}^{\mathrm{T}} \right] \left[\begin{matrix} -\alpha_k^{-1} I + \bar{E}_{1,k} \Sigma_{1,k} \bar{E}_{1,k}^{\mathrm{T}} & \bar{E}_{1,k} \Sigma_{1,k} C_k^{\mathrm{T}} \\ C_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} & \begin{pmatrix} C_k \bar{\Sigma}_k C_k^{\mathrm{T}} + D_k T_k D_k^{\mathrm{T}} + \\ \left(\alpha_k^{-1} + \beta_k^{-1} \right) H_{2,k} H_{2,k}^{\mathrm{T}} \end{pmatrix} \end{matrix} \right]^{-1} \cdot \\
&\quad \left[A_k \bar{\Sigma}_k \bar{E}_{1,k}^{\mathrm{T}} \quad A_k \bar{\Sigma}_k C_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{2,k}^{\mathrm{T}} \right]^{\mathrm{T}} + B_k U_k B_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{1,k}^{\mathrm{T}} + \beta_k^{-1} H_{1,k} H_{1,k}^{\mathrm{T}} \\
&= A_k \bar{\Sigma}_k A_k^{\mathrm{T}} - \left(A_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \hat{H}_{2,k}^{\mathrm{T}} \right) \left(\hat{C}_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \hat{R}_k \right)^{-1} \left(A_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \alpha_k^{-1} \bar{H}_{1,k} \hat{H}_{2,k}^{\mathrm{T}} \right)^{\mathrm{T}} + \\
&\quad + \alpha_k^{-1} \bar{H}_{1,k} \bar{H}_{1,k}^{\mathrm{T}} + B_k U_k B_k^{\mathrm{T}} + \beta_k^{-1} H_{1,k} H_{1,k}^{\mathrm{T}}
\end{aligned}
\tag{31}
$$

This completes the proof the Theorem 2.   □

From equations (21) and (22), both A_{ok} and K_{ok} have very close relationships of the positive scalar sequences α_k and β_k . The following Theorem 3 gives the approach of determining α_k and β_k through solving a semidefinite program subject to LMIs.

Theorem 3 Suppose that the covariance matrices $\bar{\Sigma}_k$ and $\Sigma_{1,k}$ are given, and the direct output matrix D_k is nonsingular. Then, the positive scalar sequences α_k and β_k can be determined by solving the following semidefinite program:

$$
\min_{\alpha_k, \beta_k} \mathrm{tr}(X)
\tag{32}
$$

s.t.

$$\begin{bmatrix} -X & A_k & \bar{H}_{1,k} & O & A_k\bar{E}_{1,k}^{\mathrm{T}} & A_kC_k^{\mathrm{T}}+\bar{H}_{1,k}\bar{H}_{2,k}^{\mathrm{T}} & B_k & H_{1,k} \\ A_k^{\mathrm{T}} & -\bar{\Sigma}_k^{-1} & O & O & O & O & O & O \\ \bar{H}_{1,k}^{\mathrm{T}} & O & -\alpha_kI & O & O & O & O & O \\ O & O & O & -\beta_kI & O & O & O & O \\ \bar{E}_{1,k}A_k^{\mathrm{T}} & O & O & O & -\alpha_kI & O & O & O \\ C_kA_k^{\mathrm{T}}+\bar{H}_{2,k}\bar{H}_{1,k}^{\mathrm{T}} & O & O & O & O & \begin{pmatrix}\beta_k\left(E_{2,k}D_k^{-1}\right)^{\mathrm{T}}\left(E_{2,k}D_k^{-1}\right)- \\ \left(D_kV_kD_k^{\mathrm{T}}\right)^{-1}\end{pmatrix} & O & O \\ B_k^{\mathrm{T}} & O & O & O & O & O & \beta_kE_{2,k}^{\mathrm{T}}E_{2,k}-V_k^{-1} & O \\ H_{1,k}^{\mathrm{T}} & O & O & O & O & O & O & -\beta_kI \end{bmatrix} < O, \tag{33}$$

$$\begin{bmatrix} -I & \alpha_k\bar{E}_{1,k} \\ \alpha_k\bar{E}_{1,k}^{\mathrm{T}} & \alpha_k\Sigma_{1,k}^{-1} \end{bmatrix} < O, \tag{34}$$

$$\begin{bmatrix} -I & \beta_kE_{2,k} \\ \beta_kE_{2,k}^{\mathrm{T}} & \alpha_kW_k^{-1} \end{bmatrix} < O \tag{35}$$

and

$$\begin{bmatrix} -I & \beta_kE_{2,k} \\ \beta_kE_{2,k}^{\mathrm{T}} & \alpha_kV_k^{-1} \end{bmatrix} < O \tag{36}$$

where $X = X^{\mathrm{T}} > O$.

Proof: It is obviously that α_k and β_k can be determined by the semidefinite program given as follows:

$$\min_{\alpha_k,\beta_k} \mathrm{tr}(X)$$

s.t.

$$\bar{\Sigma}_{k+1} < X, \tag{37}$$

$$\alpha_k^{-1}I - \bar{E}_{1,k}\Sigma_{1,k}\bar{E}_{1,k}^{\mathrm{T}} > O, \tag{38}$$

$$\beta_k^{-1}I - E_{2,k}W_kE_{2,k}^{\mathrm{T}} > O, \tag{39}$$

and

$$\beta_k^{-1}I - E_{2,k}V_kE_{2,k}^{\mathrm{T}} > O. \tag{40}$$

From equation (25), matrix inequality (37) is equivalent to

$$\begin{bmatrix} \bar{A}_k & \bar{A}_k\bar{C}_k^{\mathrm{T}} & B_k \end{bmatrix}\begin{bmatrix} \bar{Q}_k^{-1} & O & O \\ O & \bar{R}_k^{-1} & O \\ O & O & \bar{U}_k^{-1} \end{bmatrix}\begin{bmatrix} \bar{A}_k^{\mathrm{T}} \\ \bar{C}_k\bar{A}_k^{\mathrm{T}} \\ \bar{B}_k^{\mathrm{T}} \end{bmatrix}-X < O, \tag{41}$$

where

$$\bar{A}_k = \begin{bmatrix} A_k & \bar{H}_{1,k} & O \end{bmatrix},$$

$$\bar{C}_k = \begin{bmatrix} \hat{C}_k & \hat{H}_{2,k} & \hat{H}_{2,k} \end{bmatrix},$$

$$\bar{B}_k = \begin{bmatrix} B_k & H_{1,k} \end{bmatrix},$$

$$\bar{Q}_k = \begin{bmatrix} \bar{\Sigma}_k & O & O \\ O & \alpha_k^{-1}I & O \\ O & O & \beta_k^{-1}I \end{bmatrix},$$

$$\bar{U}_k = \begin{bmatrix} U_k & O \\ O & \beta_k^{-1}I \end{bmatrix}$$

and

$$\bar{R}_k = \begin{bmatrix} \alpha_k^{-1}I & O \\ O & D_k T_k D_k^{\mathrm{T}} \end{bmatrix}.$$

From Lemma 5, matrix inequality (41) is equivalent to the following LMI

$$\begin{bmatrix} -X & \bar{A}_k & \bar{A}_k C_k^{\mathrm{T}} & \bar{B}_k \\ \bar{A}_k^{\mathrm{T}} & -\bar{Q}_k^{-1} & O & O \\ C_k \bar{A}_k^{\mathrm{T}} & O & -\bar{R}_k^{-1} & O \\ \bar{B}_k^{\mathrm{T}} & O & O & -\bar{U}_k^{-1} \end{bmatrix} < O, \tag{42}$$

which is just (33). Also from Lemma 5, matrix inequalities (38), (39) and (40) are equivalent to LMIs (34), (35) and (36) respectively. This completes the proof of this theorem.   □

Now, the new robust Kalman filtering algorithm is summarized as follows:

$$\hat{x}_{k+1} = A_{ok}\hat{x}_k + K_{ok}\left(y_k - C_k\hat{x}_k\right),$$

$$A_{ok} = A_k + \left(A_k - K_{ok}C_k\right)\bar{\Sigma}_k\bar{E}_{1,k}^{\mathrm{T}}\left(\alpha_k^{-1}I - \bar{E}_{1,k}\bar{\Sigma}_k\bar{E}_{1,k}^{\mathrm{T}}\right)^{-1}\bar{E}_{1,k},$$

$$K_{ok} = \left(A_k S_k C_k^{\mathrm{T}} + \alpha_k^{-1}\bar{H}_{1,k}\bar{H}_{2,k}^{\mathrm{T}}\right)\left[C_k S_k C_k^{\mathrm{T}} + D_k T_k D_k^{\mathrm{T}} + \left(\alpha_k^{-1} + \beta_k^{-1}\right)H_{2,k}H_{2,k}^{\mathrm{T}}\right]^{-1},$$

$$\bar{\Sigma}_{k+1} = A_k \bar{\Sigma}_k A_k^{\mathrm{T}} - \left(A_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \alpha_k^{-1}\bar{H}_{1,k}\hat{H}_{2,k}^{\mathrm{T}}\right)\left(\hat{C}_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \hat{R}_k\right)^{-1}\left(A_k \bar{\Sigma}_k \hat{C}_k^{\mathrm{T}} + \alpha_k^{-1}\bar{H}_{1,k}\hat{H}_{2,k}^{\mathrm{T}}\right)^{\mathrm{T}} +$$
$$+ \alpha_k^{-1}\bar{H}_{1,k}\bar{H}_{1,k}^{\mathrm{T}} + B_k U_k B_k^{\mathrm{T}} + \beta_k^{-1}H_{1,k}H_{1,k}^{\mathrm{T}},$$

$$\bar{\Sigma}_0 = X_0,$$

where α_k and β_k are determined by the semidefinite program (32).

3. Application to state estimation of a low temperature pressurized water nuclear reactor

Nowadays, nuclear fission reaction provides more and more energy required for generating electrical power in the world. Safety demand and economic feasibility of a nuclear power

plant requires smooth and uninterrupted plant operation in the face of varying electrical power level. However, some state variables associated with the dynamics of a nuclear reactor are not available for measurement. If it is necessary to implement a state-feedback control or a fault detection scheme for the reactor, then these state variables could be required and consequently some observation structure should be utilized to reconstruct them. Moreover, the model nonlinearity, the variation of dynamical model parameters and the noisy character of measurement signals requires a state estimator which is robust to model uncertainty and nonlinearity, and, at the same time, must be sensitive to changes in the state variables. Therefore, the new robust Kalman filter presented in Section 2 can be applied to solve the state estimation problem for nuclear reactors. In this section, the dynamical model for the robust state estimator design of the LTPWR, which is a small research reactor designed by INET, is established, and then the robust Kalman filter derived in the last section is applied to estimate the state variables of the LTPWR.

3.1 Dynamical model
The LTPWR is a small research nuclear reactor, and Fig. 1 is the schematic drawing of the LTPWR.
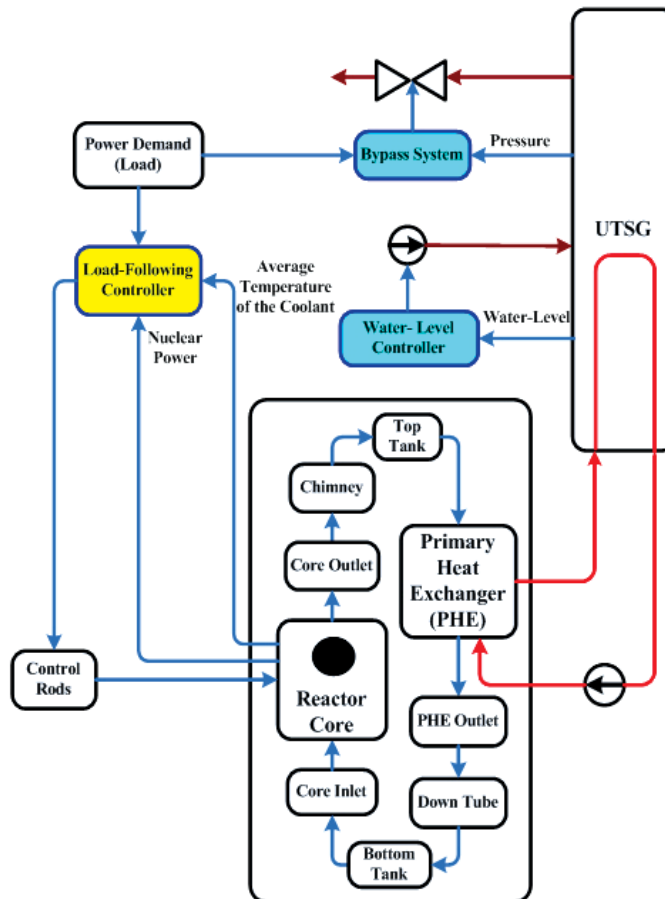
Fig. 1. Schematic drawing of the LTPWR designed by INET

The dynamical model of the nuclear reactor for designing the robust Kalman filter in this chapter is the point kinetics with one delayed neutron group and temperature feedback from the fuel and the coolant, which is summarized as follows:

$$\begin{cases} \dfrac{\mathrm{d}n_r}{\mathrm{d}t} = \dfrac{\delta\rho_r - \beta}{\Lambda}n_r + \dfrac{\beta}{\Lambda}c_r + \dfrac{\alpha_f n_r}{\Lambda}\left(T_f - T_{f0}\right) + \dfrac{\alpha_c n_r}{\Lambda}\left(T_{cav} - T_{cav0}\right) \\[2mm] \dfrac{\mathrm{d}c_r}{\mathrm{d}t} = \lambda n_r - \lambda c_r \\[2mm] \dfrac{\mathrm{d}T_f}{\mathrm{d}t} = -\dfrac{\Omega}{\mu_f}T_f + \dfrac{\Omega}{\mu_f}T_{cav} + \dfrac{P_0}{\mu_f}n_r \\[2mm] \dfrac{\mathrm{d}T_{cav}}{\mathrm{d}t} = -\dfrac{2M+\Omega}{\mu_c}T_{cav} + \dfrac{\Omega}{\mu_c}T_f + \dfrac{2M}{\mu_c}T_{cin} \\[2mm] \dfrac{\mathrm{d}\delta\rho_r}{\mathrm{d}t} = G_r z_r \end{cases} \tag{43}$$

where n_r is the neutron density relative to density at rated condition, c_r is the precursor density relative to density at rated condition, β is the fraction of delayed fission neutrons, Λ is the effective prompt neutron lifetime, λ is the effective precursor radioactive decay constant, α_f and α_c are respectively the average fuel and coolant temperature reactivity coefficient, T_f is the average reactor fuel temperature, T_{f0} is the equilibrium average reactor fuel temperature, T_{cav} and T_{cin} are respectively the average temperature of the coolant inside the reactor core and temperature of the coolant entering the reactor, T_{cav0} is the equilibrium average temperature of the coolant, Ω is the heat transfer coefficient between fuel and coolant, M is the mass flow rate times heat capacity of the coolant, P_0 is the rated power level, $\delta\rho_r$ is the reactivity due to the control rods, G_r is the total reactivity worth of the rod, and z_r is the control input, i.e. control rod speed. The parameters of the reactor at the initial of the fuel cycle in 100% power are given in Table 1.

Symbol	Quantity	Symbol	Quantity
β	0.0069	μ_c	25151(kW/℃)
Λ	1.0367e-4(s)	a_f	-3.85e-4(1/℃)
n_{r0}	1	a_c	-2.3e-5(1/℃)
λ	0.08(1/s)	G_r	0.0048
P_0	10(MW)	M	304.89(kW/℃)
μ_f	588.544(kW/℃)	Ω	437.15(kW/℃)

Table 1. Parameters of the LTPWR at the Initial of the Fuel Cycle in 100% Power

Let

$$\begin{cases} \delta n_r = n_r - n_{r0}, \\ \delta c_r = c_r - c_{r0}, \\ \delta T_f = T_f - T_{f0}, \\ \delta T_{cav} = T_{cav} - T_{cav0}, \\ \delta T_{cin} = T_{cin} - T_{cin0}, \end{cases} \tag{44}$$

where the symbol δ indicates the deviation of a variable from an equilibrium value, and n_{r0}, c_{r0}, T_{f0}, T_{cav0} and T_{cin0} correspond to the values of n_r, c_r, T_f, T_{cav} and T_{cin} at the given equilibrium condition respectively. Choose the state vector as

$$x = \begin{bmatrix} \delta n_r & \delta c_r & \delta T_f & \delta T_{cav} & \delta \rho_r \end{bmatrix}^T, \tag{45}$$

and then the state-space model, which is utilized to design the robust Kalman filter, is given as follows:

$$\begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} = A(t)x(t) + B\big[u(t)+w\big] + f(x), \\ \quad y = Cx(t) + Dv, \end{cases} \tag{46}$$

where

$$A(t) = \begin{bmatrix} -\dfrac{\beta}{\Lambda} & \dfrac{\beta}{\Lambda} & \dfrac{\alpha_f n_{r0}}{\Lambda} & \dfrac{\alpha_c n_{r0}}{\Lambda} & \dfrac{n_{r0}}{\Lambda} \\ \lambda & -\lambda & 0 & 0 & 0 \\ \dfrac{P_0}{\mu_f} & 0 & -\dfrac{\Omega}{\mu_f} & \dfrac{\Omega}{\mu_f} & 0 \\ 0 & 0 & \dfrac{\Omega}{\mu_c} & -\dfrac{2M+\Omega}{\mu_c} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & G_r \end{bmatrix}^T,$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$D = 1,$$

$$f(x) = \begin{bmatrix} \dfrac{\alpha_f}{\Lambda}\delta n_r \delta T_f + \dfrac{\alpha_c}{\Lambda}\delta n_r \delta T_{cav} + \dfrac{1}{\Lambda}\delta n_r \delta \rho_r \\ 0 \\ 0 \\ \dfrac{2M}{\mu_c}\delta T_{cin} \\ 0 \end{bmatrix},$$

and w and v are the process noise and the measurement noise respectively, which are all zero-mean Gaussian white noises.

As we have stated as above, the parameters of the reactor vary with the power level. Since the state-space model (46) is established near a specific working point, the sensors and the actuators are also not precise indeed, there must be uncertainties in $A(t)$, B, C and D. Let $\Delta A(t)$, $\Delta B(t)$, $\Delta C(t)$ and $\Delta D(t)$ denote the norm-bounded uncertaintes of $A(t)$, B, C and D respectively, and suppose

$$\begin{bmatrix} \Delta A(t) & \Delta B(t) \\ \Delta C(t) & \Delta D(t) \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} F(t) \begin{bmatrix} E_1 & E_2 \end{bmatrix} \tag{47}$$

where

$$H_1 = \begin{bmatrix} 0 & 0 & \dfrac{\alpha_f}{\varLambda} & \dfrac{\alpha_c}{\varLambda} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\dfrac{\varOmega}{\mu_f} & \dfrac{\varOmega}{\mu_f} & 0 \\ 0 & 0 & \dfrac{\varOmega}{\mu_c} & -\dfrac{2M+\varOmega}{\mu_c} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$E_1 = \mathrm{diag}\left(0.1, 0.1, 0.1, 0.1, 0\right),$$

$$H_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}},$$

$$E_2 = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0.01 \end{bmatrix},$$

$$F(t) = \mathrm{diag}\left(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5\right),$$

and the real scalars $\varepsilon_i, i = 1, 2, \cdots, 5$, satisfy $|\varepsilon_i| \le 1, i = 1, 2, \cdots, 5$ respectively. Therefore, the state-space model containing model uncertainties and nonlinear perturbation can be written as follows:

$$\begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} = \left[A(t) + \Delta A(t)\right]x(t) + \left[B + \Delta B(t)\right]\left[u(t) + w\right] + f(x), \\ y = \left[C + \Delta C(t)\right]x(t) + \left[D + \Delta D(t)\right]v. \end{cases} \quad (48)$$

Furthermore, it is very easy to obtain the discrete-time state-space model corresponding to system (48), which takes the form as (1).

3.2 Numerical simulation

Both the robust Kalman filter (8) and the EKF are applied to the situation that the power level of the LTPWR rises from 50% to 100% rated power. The feedback control law for nuclear power and coolant temperature control is a static output feedback control taking the following form

$$u(t) = -\left(F_n \delta n_r + F_T \delta T_{\mathrm{cav}}\right) \quad (49)$$

where F_n and F_T are both positive scalars. The parameter values for the numerical simulation are given in the following Table 2, where T_s is the sampling period to obtain the discrete-time state-space model (1) from the continuous-time state-space model (48).

Symbol	Quantity
F_n	0.05
F_T	0.005
W	1e-6
V	1e-6
X_0	1e-8I_5
P	1e-2 I_5
T_s (s)	0.1

Table 2. Parameter values for the numerical simulation
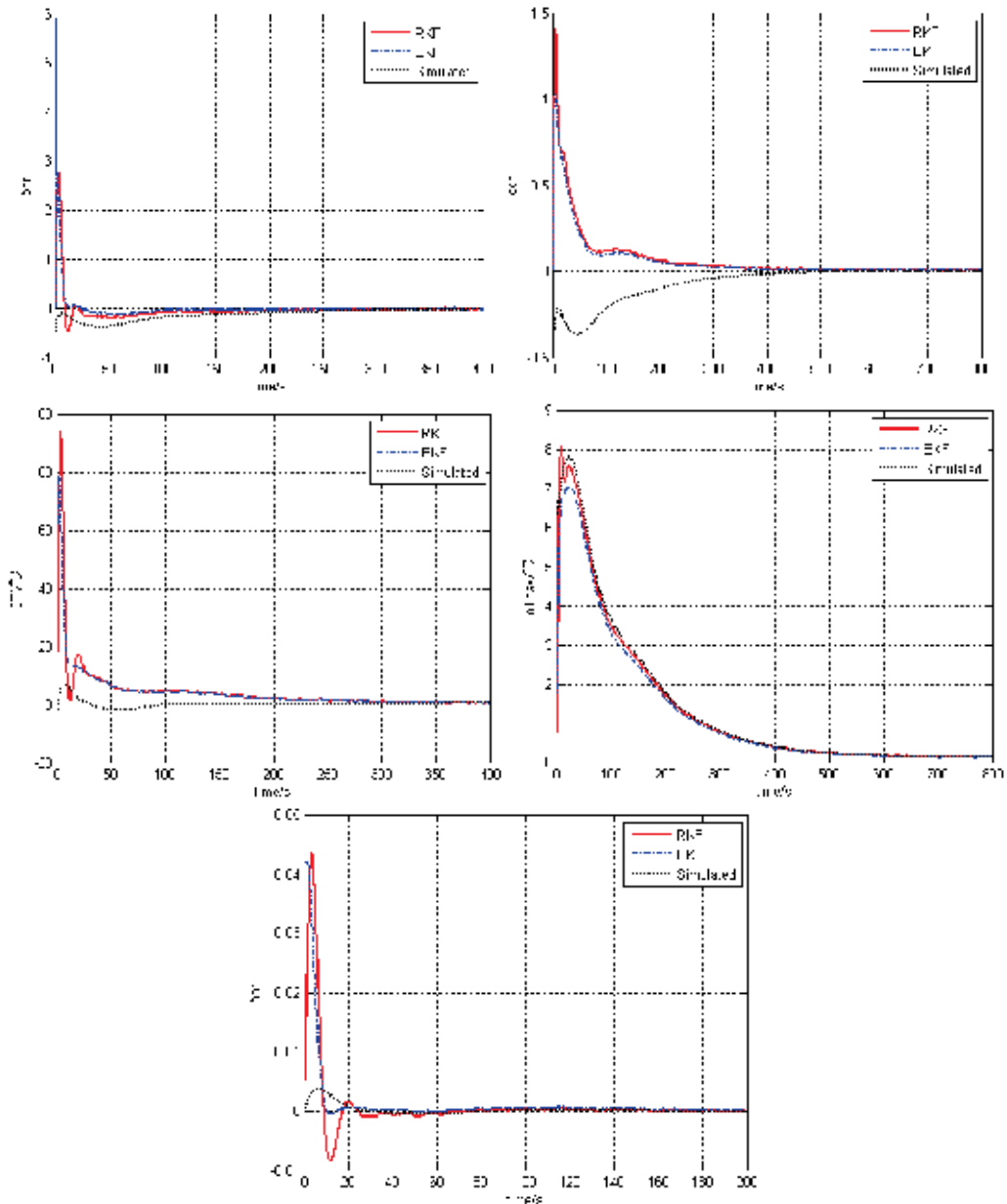
Fig. 1. Simulated and estimated value of the state variables

Moreover, the simulation model of the LTPWR utilized in this numerical simulation, which was developed by INET based on MATLAB /SIMULINK®, contains point kinetics model with six delayed neutron groups (7th order), 2nd order heat exchange dynamics of the reactor core, 2nd order dynamics of primary heat exchanger, 5th order dynamics of a U-tube steam generator (UTSG) with a water-level controller and bypass system, 2nd order dynamics of the pump of UTSG, and 6th order dynamics of other heat transmission pipe or volume cells. The mathematical model of the LTPWR is introduced briefly in Section 5 (Appendix).

Simulation results are illustrated as the follows. The estimation values of the state variables, i.e. δn_r, δc_r, δT_f, δT_{cav} and $\delta \rho_r$, provided by the robust Kalman filter (RKF) and the EKF, and the simulated values of these state variables provided by the simulation model are all given in Fig. 1. The values of the estimation error variance of the EKF and the upper-bound of the estimation error variance of the robust Kalman filter are given in Fig. 2. The mean squared errors (MSE) of the EKF and the RKF are illustrated in Fig. 3, and the MSE is defined as:

$$\begin{cases} MSE = MSE_1 + MSE_2 \\ MSE_1 = \left(\delta n_r - \delta \hat{n}_r\right)^2 + \left(\delta T_{cav} - \delta \hat{T}_{cav}\right)^2 + \left(\delta \rho_r - \delta \hat{\rho}_r\right)^2 \\ MSE_2 = \left(\delta c_r - \delta \hat{c}_r\right)^2 + \left(\delta T_f - \delta \hat{T}_f\right)^2 \end{cases} . \tag{50}$$

The values of the positive scalar sequences α_k and β_k are given in Fig. 4.
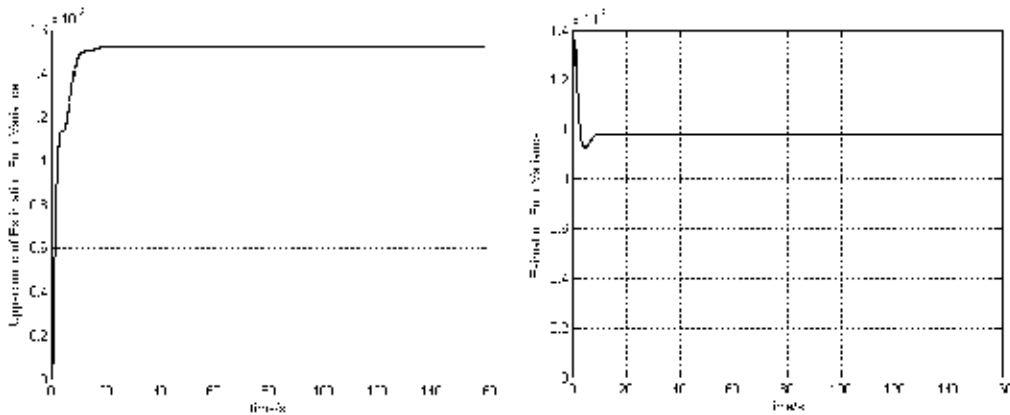
Fig. 2. Values of the estimation error variances
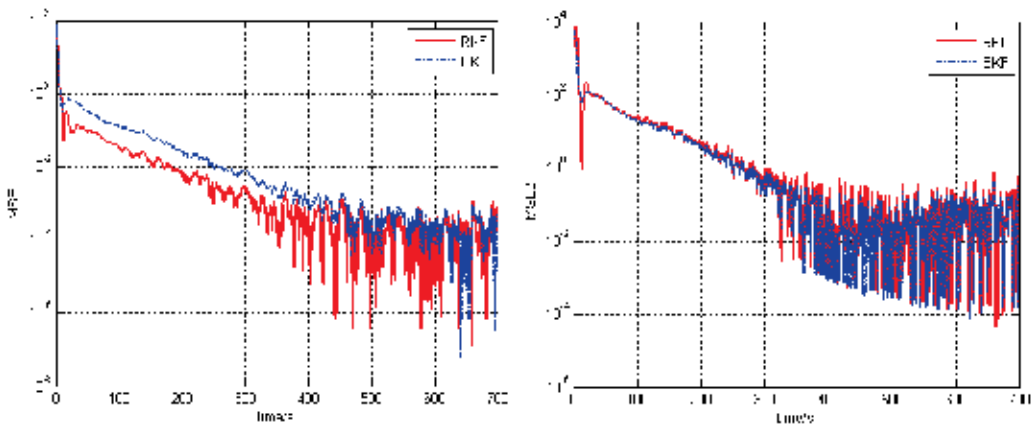
Fig. 3. MSEs of the RKF and the EKF
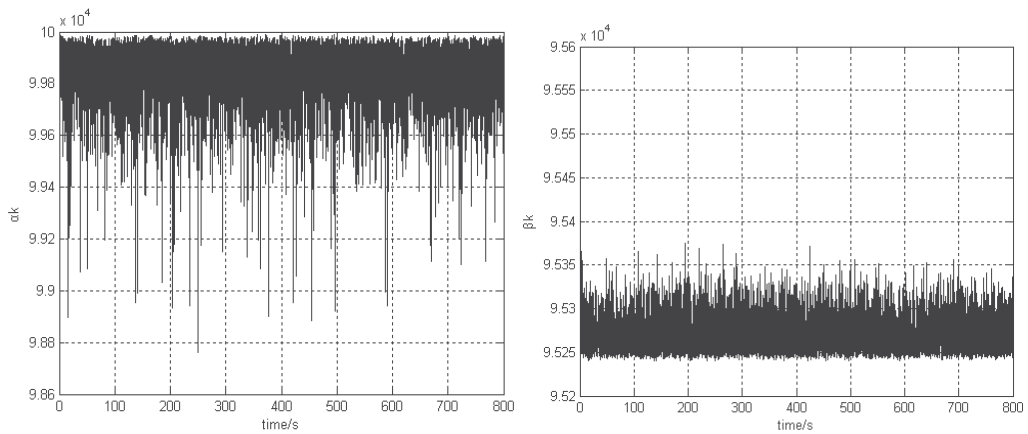
Fig. 4. Values of α_k and β_k

3.3 Discussion

As we can see from Fig. 2., the upper-bound of the state estimation error variance provided by the RKF is larger than the state estimation error variance provided by the EKF. However, this does not show that the estimation performance of the EKF is higher than that of the RKF because the state estimation error variance of the EKF is obtained from the nominal linearized dynamic model of the LTPWR. Moreover from Fig. 1., the estimation values of the state variables with the RKF track the variation of the corresponding simulated values faster than that with the EKF do, and this also shows the high performance of the RKF derived in this chapter. Fig. 3 shows that the MSE of the EKF is larger than that of the RKF, though they are nearly equal to each other when the power level has been lifted to 100%. The reason of the phenomenon illustrated in Figs. 1, 2 and 3 is: During the procedure of power lift from 50% to 100%, there must be model uncertainties and nonlinear perturbations due to variations of the parameter values of the LTPWR. Therefore, the MSE of the RKF can be smaller than that of the EKF to the system model with nonlinear perturbation and uncertain parameter values.

4. Conclusions

Motivated by the robust state estimation of nuclear reactors, a new finite-horizon robust Kalman filter for discrete-time systems with nonlinear perturbation and norm-bounded uncertainties in the state, the output, the input and the direct output matrices was presented in this chapter. After deriving the mathematical expressions of the robust Kalman filter, the newly presented filter was then applied to the state estimation for a low temperature pressurized water reactor designed by INET. Simulation results show that the performance of the robust Kalman filter is higher than that of the celebrated EKF. The future research work lies in two aspects. The first one is to extend the robust Kalman filter in this chapter to time-delayed systems since there are many time-delays in a nuclear reactor system, and the second one is to estimate the density of poisons, such as xenon 135 and samarium 149, for a nuclear reactor.

5. Appendix: brief Introduction of the mathematical model for the LTPWR

The mathematical model of the aforementioned LTPWR will be introduced briefly as follows. The model is constructed upon the fundamental conservation of mass, energy and momentum. Here, only the dynamical equations are given, and the derivation procedures are all omitted.

5.1 Neutron kinetics

Point kinetics model with six delayed neutron groups is considered for the LTPWR, and the dynamical equations are given as follows.

$$
\begin{cases}
\dfrac{\mathrm{d}n_r}{\mathrm{d}t} = \dfrac{\delta\rho_r - \beta}{\Lambda}n_r + \displaystyle\sum_{i=1}^{6}\dfrac{\beta_i}{\Lambda}c_{ri} + \dfrac{\alpha_f n_r}{\Lambda}\left(T_f - T_{f0}\right) + \\[2mm]
\qquad \dfrac{\alpha_c n_r}{\Lambda}\left(T_{cav} - T_{cav0}\right) \\[2mm]
\dfrac{\mathrm{d}c_{ri}}{\mathrm{d}t} = \lambda_i\left(n_r - c_{ri}\right)(i = 1, 2, \cdots, 6) \\[2mm]
\dfrac{\mathrm{d}\delta\rho_r}{\mathrm{d}t} = G_r z_r
\end{cases}
\tag{51}
$$

where c_{ri} is the precursor density relative to density of the ith group delayed fission neutrons at rated condition, β_i is the fraction of the ith group delayed fission neutrons, and λ_i is the effective precursor radioactive decay constant of the ith group delayed fission neutrons.

5.2 Reactor and primary loop

The primary loop is divided into seven volume units and two transport units. As illustrated in Fig. 1, the seven volumes are the reactor core, the reactor outlet, the top tank, the Primary Heat-Exchanger (PHE), the PHE outlet, the bottom tank, and the core inlet, and the two transport units are the chimney and the down tube.

(1) Reactor Core

Some assumptions are given to provide a simple model: (i) the fuel element is homogeneous, so one-dimensional thermo-dynamic equations can be achieved; (ii) the thermal power is produced by the fission products and radiation process such as β-radiation and γ-radiation, but here the heat produced by the radiation process is neglected; (iii) the temperatures of the fuel element and the coolant inside the reactor core are assumed even. Thus a lumped parameter approach can be given as follows:

$$
\begin{cases}
\dfrac{\mathrm{d}T_f}{\mathrm{d}t} = -\dfrac{\Omega}{\mu_f}T_f + \dfrac{\Omega}{\mu_f}T_{cav} + \dfrac{P_0}{\mu_f}n_r, \\[2mm]
\dfrac{\mathrm{d}T_{cav}}{\mathrm{d}t} = -\dfrac{2M + \Omega}{\mu_c}T_{cav} + \dfrac{\Omega}{\mu_c}T_f + \dfrac{2M}{\mu_c}T_{cin}.
\end{cases}
\tag{52}
$$

(2) Volume Units

Assume that the coolant in these units is well mixed. Energy balance equations for each unit are given as follows.
Core outlet:

$$\rho_{21}V_{21}\frac{dT_{21}}{dt} = W_1\left(T_{cout} - T_{21}\right), \tag{53}$$

Top tank:

$$\rho_{23}V_{23}\frac{dT_{e1pin}}{dt} = W_1\left(T_{22} - T_{e1pin}\right), \tag{54}$$

PHE outlet:

$$\rho_{41}V_{41}\frac{dT_{41}}{dt} = W_1\left(T_{e1pout} - T_{41}\right), \tag{55}$$

Bottom tank:

$$\rho_{43}V_{43}\frac{dT_{43}}{dt} = W_1\left(T_{42} - T_{43}\right), \tag{56}$$

Core inlet:

$$\rho_{43}V_{44}\frac{dT_{cin}}{dt} = W_1\left(T_{43} - T_{cin}\right). \tag{57}$$

Here the subscripts 21, 22, 23, 41, 42, 43 and 44 denote the core outlet, chimney, top tank, PHE outlet, down tube, bottom tank, and core inlet respectively. T_{21}, T_{22}, T_{23}, T_{41}, T_{42} and T_{43} are the outlet temperatures of these units. ρ_{21}, ρ_{22}, ρ_{23}, ρ_{41}, ρ_{42}, ρ_{43} denote the densities of the coolant in these units. T_{e1pin} and T_{e1pout} denote the inlet and outlet temperatures of the PHE. W_1 is the natural circulation flow-rate of the coolant.

(3) Transport Units
It is assumed that there is no mixing within the chimney and the down tube. Therefore, the coolant flowing through these two units will lead to delay in the temperature variations. The equations are:
Chimney:

$$T_{22}\left(t\right) = T_{21}\left(t - \frac{\rho_{22}V_{22}}{W_1}\right), \tag{58}$$

Down tube:

$$T_{42}\left(t\right) = T_{41}\left(t - \frac{\rho_{42}V_{42}}{W_1}\right). \tag{59}$$

(4) Natural Circulation Flux
The primary coolant flow of the LTPWR is natural circulation which is driven by the density difference between the hot and cold coolant. During the modeling procedure, the gravity pressure drop in the loop, the frictional pressure drop in the core and PHE, and the entrance pressure drop at the core inlet are considered, while all the resistance pressure drops in the other units as well as the acceleration pressure drop in the loop are neglected. Then the natural circulation flow-rate of the coolant W_1 is the determined by the following algebraic equation:

$$\rho_3 g h_3 + \rho_{42} g h_{42} - \rho_{22} g h_{22} - \rho_1 g h_1 = \frac{1}{2} k_{e1p} \rho_{41} \left(\frac{W_1}{\rho_{41} A_{e1p}} \right)^2 + \frac{1}{2} \left(k_{cin} + k_c \right) \rho_{41} \left(\frac{W_1}{\rho_1 A_c} \right)^2 \qquad (60)$$

where ρ_1 and ρ_3 are the coolant density in the reactor core and the PHE, h_1, h_{22}, h_3 and h_{42} are the lengths of the reactor core, the chimney, the PHE and the down tube respectively, k_{e1p}, k_{cin} and k_c are the resistance pressure coefficient in the primary side of the PHE, the core inlet and the core, A_c and A_{e1p} are the flow cross section of the core and the primary side of the PHE, and g is the gravitational acceleration.

(5) Primary Heat Exchanger

In order to obtain an accurate yet simple model, two approximations are introduced. The first approximation is to lump the tube metal thermal inertia with the primary side coolant, and the second one is the PHE outlet temperatures of both sides are calculated assuming a single node and perfect mixing. The dynamic equations are given as follows.

Primary side:

$$\begin{cases} \dfrac{dT_{e1p}}{dt} = \dfrac{W_1 C_3 \left(T_{e1pin} - T_{e1pout} \right) - K_{e1} A_{e1} \Delta T_{e1}}{M_{e1m} C_{e1m} + \rho_3 V_3 C_3} \\[3mm] T_{e1p} = \dfrac{1}{2} \left(T_{e1pin} + T_{e1pout} \right) \\[3mm] \Delta T_{e1} = \dfrac{\left(T_{e1pout} - T_{e1sin} \right) - \left(T_{e1pin} - T_{e1sout} \right)}{\ln \left(\dfrac{T_{e1pout} - T_{e1sin}}{T_{e1pin} - T_{e1sout}} \right)} \end{cases} \qquad (61)$$

Secondary side:

$$\begin{cases} \dfrac{dT_{e1s}}{dt} = \dfrac{K_{e1} A_{e1} \Delta T_{e1} + W_2 C_{e1s} \left(T_{e1sin} - T_{e1sout} \right)}{M_{e1s} C_{e1s}} \\[3mm] T_{e1s} = \dfrac{1}{2} \left(T_{e1sin} + T_{e1sout} \right) \end{cases} \qquad (62)$$

where M_{e1m} is the mass of the heat exchange tube, K_{e1} denotes the overall heat transfer coefficient, A_{e1} is the heat exchange area, ρ_3 and V_3 are respectively the density and volume of the coolant in the PHE, C_{e1m} and C_3 are the specific heat of the coolant in the primary side and the metal of the heat exchange tube.

5.3 Steam generator

The steam generator of the LTPWR is a U-tube steam generator (UTSG). The dynamical equations are established based on mass balance, energy balance and momentum balance. The structure of the UTSG discussed in this paper is given in Fig. 8. Two assumptions are made to derive the mathematical model. The first one is a linear dependence on the steam pressure p for the saturate liquid and steam, and the second is that the flow quality of the homogenous two-phase flow is a linear function of the length from the top of the U-tubes to the outlet of the riser.

(1) Mass Balances

Steam Dome:

$$V_{st} \frac{d\rho_g}{dp} \frac{dp}{dt} = x_M W_{sep} - W_{st} ,$$ (63)

Downcomer:

$$\rho_d A_d \frac{dL_d}{dt} = W_{fw} + (1 - x_M) W_{sep} - W_d ,$$ (64)

Riser:

$$\frac{k_1 V_s}{v_s^2} \left(v_{fg} \frac{dx_M}{dt} + x_M \frac{dv_{fg}}{dp} \frac{dp}{dt} \right) = W_{sep} - W_d .$$ (65)
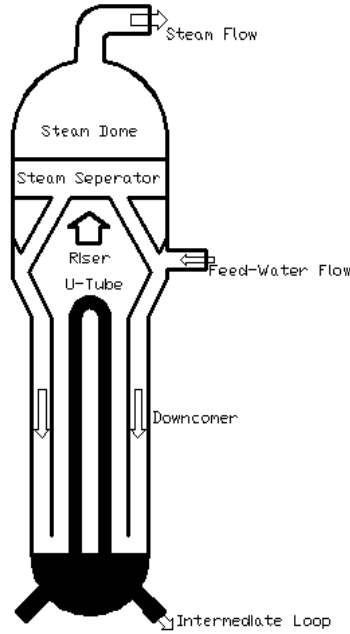
Fig. 8. Structure of UTSG

Here V_{st} is the volume of the steam dome, ρ_g is the density of the saturated steam inside the steam dome, p is the steam pressure, x_M is the steam quality of the homogeneous two phase flow that reaches the steam separator, W_{sep} is the total mass flow leaving the riser region and entering the steam separator region, W_{st} is the steam mass flow rate leaving the steam dome, ρ_d is the downcomer water density, A_d is the downcomer flow area, L_d is the downcomer water level, W_{fw} is the flow rate of feed water, W_d is the downcomer flow rate, V_s is the volume of the riser, v_s is the average specific volume of the two-phase flow inside the riser, $k_1 = 1 - \frac{L_1}{2L_2}$, L_1 is the height of the U-tube bundle, L_2 is the length of the riser, $v_{fg} = v_g - v_f$, v_f and v_g are the specific volume of the liquid and steam inside the riser respectively.

(2) Energy Balances

Downcomer:

$$\rho_d A_d \left(h_d \frac{\mathrm{d}L_d}{\mathrm{d}t} + L_d \frac{\mathrm{d}h_d}{\mathrm{d}t} \right) = h_{fw} W_{fw} + \left(1 - x_M\right) h_f W_{sep} - h_d W_d \, , \tag{66}$$

Riser :

$$\frac{V_s k_1 h_s}{v_s} \left[\left(\frac{h_{fg}}{h_s} - \frac{v_{fg}}{v_s} \right) \frac{\mathrm{d}x_M}{\mathrm{d}t} - \frac{x_M}{v_s} \frac{\mathrm{d}v_{fg}}{\mathrm{d}p} \frac{\mathrm{d}p}{\mathrm{d}t} \right] = -h_{sep} W_{sep} + h_d W_d + Q \, , \tag{67}$$

Here h_f is the enthalpy of the liquid inside the riser, h_{fg} is the latent heat of vaporization, h_s is the average enthalpy of the two-phase flow inside the riser, v_s is the average specific volume of the two-phase flow inside the riser, h_{sep} is the enthalpy of two-phase flow leaving the riser region and entering the steam separator region, h_d is the enthalpy of the liquid inside the downcomer, and Q is the total heat input to the UTSG.

(3) Momentum Balance for the Downcomer Flow

Assume that the mass flow rate W_d from the downcomer to the riser is simply due to the static head between water level L_w inside the riser and the water level L_d inside the downcomer, i.e.

$$\begin{cases} W_d = k_d \sqrt{\dfrac{L_d}{v_d} - \dfrac{L_w}{v_f}}, \\[2mm] L_w = \left(1 - k_1 x_M\right) \dfrac{v_f A_1}{v_s A_w} L_2. \end{cases} \tag{68}$$

where k_d is a given constant, v_d is the specific volume of the water inside the downcomer, v_f is the specific volume of the liquid inside the riser, A_1 is the cross section area of the steam separator, and A_w is the cross section area of the riser.

6. References

Anderson, B. D. O. & Moore, J. B. (1979). *Optimal filtering*, Prentice-Hall Inc., 0-13-638122-7, Englewood Cliffs, N. J., USA

Boyd, S.; Ghaoui, L. E.; Feron, E. & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*, Society for Industrial and Applied Mathematics, 0-89-871485-0, Philadelphia, USA

Bucy, R. S. & Senne, K. D. (1971). Digital synthesis of nonlinear filters, *Automatica*, Vol. 7, No. 3, pp. 287-289, 0005-1098

Chui, C. K. & Chen, G. (1999). *Kalman filtering with real-time applications (third edition)*, Springer, 3-540-64611-6, Germany

Dong, Z. & You, Z. (2006). Finite-horizon robust Kalman filtering for discrete time-varying systems with uncertain-covariance white noises, *IEEE Signal Processing Letters*, Vol. 13, No. 8, pp. 493-496, 1070-9908

Fu, M.; de Souza, C. E. & Luo, Z.-Q. (2001). Finite-horizon robust Kalman filter design. *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, pp. 2103-2112, 0018-9286

Garcia, G.; Tarbouriech S. & Peres, P. L. D. (2003). Robust Kalman filtering for uncertain discrete-time linear systems, *International Journal of Robust and Nonlinear Control*, Vol. 13, pp. 1225-1238, 1099-1239

Hounkpevi, F. O. & Yaz, E. E. (2007). Robust minimum variance linear state estimators for multiple sensors with different failure rates, *Automatica*, Vol. 43, pp. 1274-1280, 0005-1098

Jian, B. N. (1975). Guaranteedestimation in uncertain systems, *IEEE Transactions on Automatic Control*, Vol. 23, pp. 230-232, 0018-9286

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *Journal of Basic Engineering Transactions of the ASME, Series D*, Vol. 82, No. 1, pp. 35-45, 0021-9223

Kalman, R. E. & Bucy R. S. (1961). New results in linear filtering and prediction problems, *Journal of Basic Engineering Transactions of the ASME, Series D*, Vol. 83, No. 3, pp. 95-108, 0021-9223

Lu, X.; Xie, L; Zhang, H. & Wang, W. (2007). Robust Kalman filtering for discrete-time systems with measurement delay, *IEEE Transactions on Circuits and Systems---II: Express Briefs*, Vol. 54, No. 6, pp.522-526, 1549-7747

Petersen, I. R. & McFarlane, D. C. (1991). Robust state estimation for uncertain systems, Proceedings of the 30th Conference on Decision and Control, pp. 1630-1631, Brighton, England, December 1991, IEEE

Petersen, I. R. & McFarlane, D. C. (1994). Optimal guaranteed cost control and filtering for uncertain linear systems, *IEEE Transactions on Automatic Control*, Vol. 39, No. 9, pp. 1971-1977, 0018-9286

Petersen, I. R. & McFarlane, D. C. (1996). Optimal guaranteed cost filtering for uncertain discrete-time systems, *International Journal of Robust and Nonlinear Control*, Vol. 6, pp. 267-280, 1099-1239

Shaked, U. & de Souza, C. E. (1995). Robust Minimum Variance Filtering, *IEEE Transactions on Signal Processing*, Vol. 43, No. 11, pp. 2474-2483, 1053-587X

Shaked, U.; Xie, L. & Soh, Y. C. (2001). New approaches to robust minimum variance filter design. *IEEE Transactions on Automatic Control*, Vol. 49, No. 11, pp. 2620-2629, 0018-9286

Theodor, Y. & Shaked, U. (1996). Robust discrete-time minimum-variance filtering. *IEEE Transactions on Signal Processing*, Vol. 44, No. 2, pp. 181-189, 1053-587X

Xie, L. & de Souza, C. E. (1993). H_∞ state estimation for linear periodic systems. *IEEE Transactions on Automatic Control*, Vol. 38, pp. 1704-1707, 0018-9286

Xie, L. & Soh, Y. C. (1994). Robust Kalman filtering for uncertain systems. *Systems and Control Letters*, Vol. 22, pp. 123-129, 0167-6911

Xie, L.; Soh, Y. C. & de Souza, C. E. (1994). Robust Kalman filtering for uncertain discrete-time systems. *IEEE Transactions on Automatic Control*, Vol. 39, No. 6, pp. 1310-1314, 0018-9286

Yang, G.-H. & Wang, J. L. (2001). Robust nonfragile Kalman filtering for uncertain linear systems with estimator gain uncertainty, *IEEE Transactions on Automatic Control*, Vol. 46, No. 2, pp. 343-348, 0018-9286

Yang, F.; Wang, Z. & Hung, Y. S. (2002). Robust Kalman filtering for discrete time-varying uncertain systems with multiplicative noises. *IEEE Transactions on Automatic Control*, Vol. 47, No. 7, pp.1179-1183, 0018-9286

Zhu, X.; Soh, Y. C. & Xie, L. Design and analysis of discete-time robust Kalman filters. *Automatica*, Vol. 38, pp. 1069-1077, 0005-1098

The Kalman Filter in Power Quality – Theory and Applications

Mario González and Víctor Cárdenas
University of San Luis Potosí, UASLP
Mexico

1. Introduction

Power quality is a topic in the area of engineering that emerged by the 1970 decade. It is a field that has attracted special interest recently due to the continuous industrial growth, to the raise of power demands, and to the proliferation of "polluting" electrical loads. One of the objectives of power quality is to meet a clean sinusoidal, stable, reliable, regulated, and uninterrupted supply voltage in electrical systems, in order to feed the so-called *critical loads*. A critical load is such an equipment that, if it fails or works inadequately, it can cause very high losses, being economical, or of any class; and a low quality supply can cause this situation. For example, it can cause the loss of vital information, interruption of an expensive industrial process, poor quality or damage to products; interruption of important communications as air traffic control, security units, and financial information; permanent damage to equipments, and even put lives in danger at hospitals.

The great increment in critical processes has led to the requirement of assuring a high quality and safe power supply in many medical, communications, and industrial procedures, with the aim of feeding machinery and automatic systems that perform diverse important tasks. A safe protection of the operations is the objective.

Problems and failures in electrical loads can be caused by some of the disturbances that exist in electrical systems. Among them are: *harmonic distortion*, *unbalances*, *non characteristic harmonics*, *sags*, *swells*, short and long *interruptions*, *flicker*, and *short circuits*. The disturbances that more affect critical loads, specially the industrial kind, are the sags, the swells, and the interruptions. The sag is defined as a 10% to 90% decrement of the nominal value of voltage, which can last from a half cycle to one minute. The swell is defined in a similar way, but represented by the 10% to 80% increase of the nominal value (IEEE Std 1159-1995, 1995), (IEEE Std 1159.3-2003, 2004).

Sags have been identified as the most severe disturbance, and as the one that more causes damages and problems to facilities and equipments. It is considered that sags, together with momentary interruptions, are responsible of the 92% of power quality problems that face typical industrial consumers (Bhadkamkar et al., 2003). Sags can be caused by atmospheric discharges, short circuits, energizing of motors and high power loads, operation of soldering machines, and arc furnaces, to mention some. The swells can be caused by disconnections of high power loads from the grid. As can be noted, the disturbances can be generated by natural cause, by neighbor installations, or by accident. It is always desirable their

elimination at the origin point. In the cases where this is not possible, the installation of a power equipment that compensates the disturbances in real time becomes viable.

To evaluate power quality problems and propose a solution adequately, first a monitoring of voltage parameters as magnitude, phase, and frequency of the fundamental and harmonic components is required. In addition, before voltage variations can be compensated, a fast detection of them must be performed by the compensation system. Mathematical real-time estimation algorithms are crucial for both the mentioned purposes.

The Kalman filter (KF) is a mathematical method widely extended in many areas of science, and has important applications in the power quality field of electrical engineering. Some of the most important are discussed in this chapter. First, some power quality compensators and the usual requirements of the supply voltage in electrical systems are addressed, to have a clear view of the applications. Then, the applications of the KF as detector of disturbances, and as estimator of parameters used in monitoring are discussed.

2. Active power compensators for improving power quality

Numerous power electronics topologies have been proposed for the compensation of disturbances in electrical systems (Bollen, 2000), (Dugan et al., 2003), (Martínez, 1992), (Sannino et al., 2000), (Zamora, 1997). Among the most important to compensate sags, swells, and interruptions are: the uninterruptible power supply (UPS), static transfer switch (STS), dynamic voltage restorer (DVR), and the transformer with changeable taps. A brief operative description of the aforementioned equipments is given next, in order to relate the applicability of the KF. A detailed description of the compensators, as well as more kinds that are focused in correcting other disturbances can be consulted in the above references.

Uninterruptible power supply (UPS) – it is considered one of the best options of power compensators. It consists of an ac/dc converter (controlled rectifier, uncontrolled rectifier or active front-end rectifier), an energy storage bank (frequently based on batteries), and a dc/ac converter (inverter). The UPS is very common and is very extended worldwide in low, medium, and high power levels. It keeps a constant and regulated voltage at the load, even under eventualities as sags and interruptions (with backups going from a few cycles to several minutes). A basic scheme of the UPS is shown in Figure 1. This scheme in particular can be operated in online or offline mode. The online mode consists in feeding the load directly from the inverter (when the ac mains is present it works with the rectifier, and when it is not it works with the batteries), and if it fails, the static switch changes the connection to the source. The offline mode consists on maintaining the load connected to the source permanently, and when a disturbance appears, the switch changes the operation to the inverter. An algorithm to detect the presence of a sag or an interruption in real time is needed to manage the operation of the equipment and protect the load effectively, particularly in offline systems. The speed of detection and the corrective action together have to be smaller than half cycle of the sinusoidal voltage wave, since the majority of critical loads can not tolerate a disturbance during a greater time.

Static transfer switch (STS) – under an eventuality, this equipment based on static switches disconnects the load from the disturbed source and transfers it to another source with good conditions, or to a small generation plant. The approach is applicable and effective for low and medium voltage at distribution level. The two sources must be really independent and synchronized correctly. It requires fast detection of disturbances too, to coordinate the transfer. Figure 2 shows a general scheme of the STS.
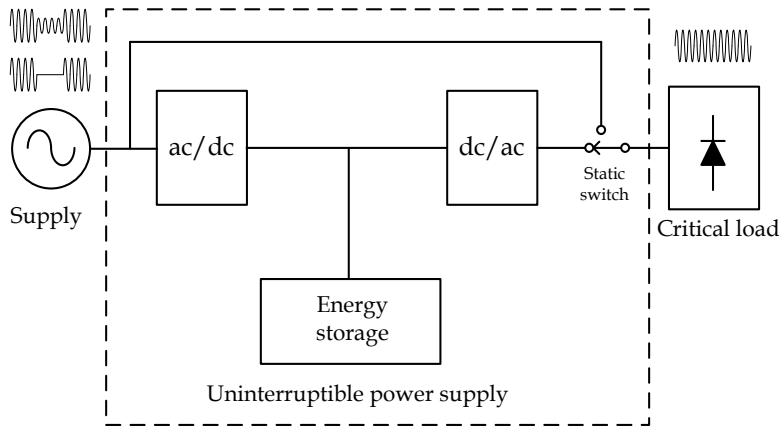
Fig. 1. A basic scheme of the uninterruptible power supply (UPS).
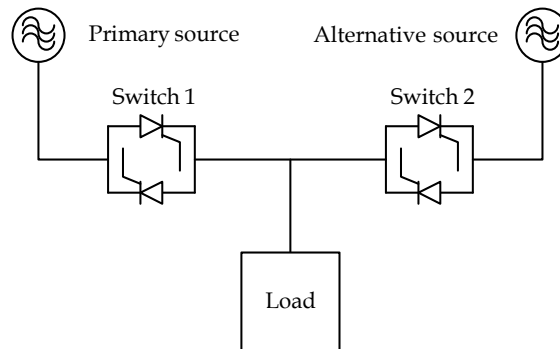
Fig. 2. Static transfer switch (STS).

Dynamic voltage restorer (DVR) *–* this compensator has received special attention recently. Depending on the topology, it can provide protection against sags and swells in low, medium, and high voltage levels (Affolter & Connell, 2003), (Joos et al., 2004), (Nielsen et al., 2004), (Wunderlin et al., 1998). This solution has already been used at high power levels in countries such as the United States, Israel, Japan, United Kingdom, and Singapore for the protection of complete industrial facilities. Figure 3 shows a basic scheme of the DVR. It consists of a power inverter connected in series between the source and the load through a coupling transformer, having also an energy storage element. When a disturbance appears, the equipment injects the missing or exceeding voltage in series in such a way that the load always has a constant and regulated voltage. The DVR also requires a disturbance detection algorithm with the aim of coordinating the change in the reference to inject.

Transformer with electronic tap changers *–* it uses static switches to change the turns ratio of a transformer dedicated to a single load or of a distribution transformer, under voltage variations (Baitch & Barr, 1985), (Hingorani & Gyugyi, 2000), (Fletcher & Stadlin, 1983), (Sannino, 2000). It is a solution that maintains the voltage regulation within a tolerance band, generally between 90% and 110%. The secondary coil is divided into various sections, which are connected or disconnected by the switches. It also requires a method that monitors the thresholds of voltage in a fast way. Figure 4 illustrates the device.

It is evident that the compensation equipments abovementioned require a fast method to monitor and detect voltage variation disturbances. A fast detection is crucial for the equipment to be effective, and to accomplish its operation of compensating the disturbance as fast as possible. The KF, among other methods, is a valuable tool for this purpose. But, to design a correct algorithm, it is necessary to know the limits on the requirements of the supply voltage for critical loads, a topic discussed in the following section.
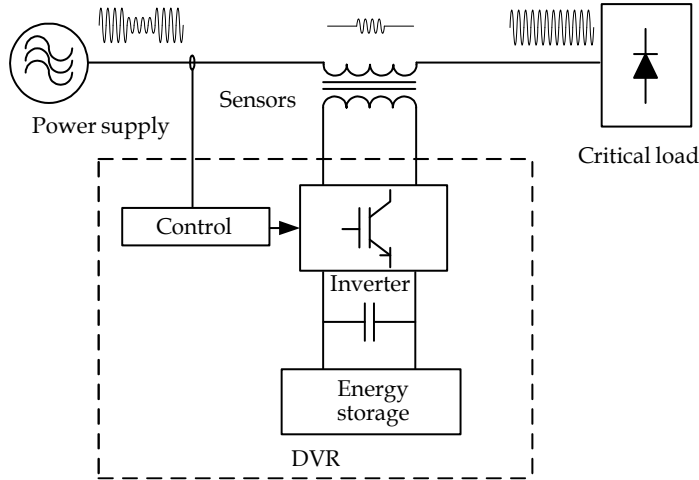
Fig. 3. Dynamic voltage restorer (DVR).
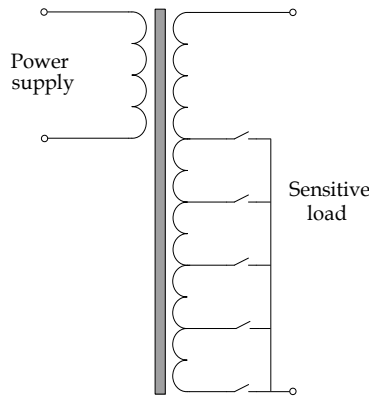
Fig. 4. Transformer with electronic tap changers.

3. Requirements of the supply voltage for critical loads

Requirements of the supply voltage for specific critical loads are shown in Table 1 (Martínez, 1992). The majority of low critical loads allow slow and steady state voltage variations within ±10% of the nominal voltage. Nonetheless, some highly critical loads only allow variations within the +5% and the -8%, or even within ±5% . Similar requirements are established in (IEEE Std 1100-2005, 2006). In general, the tolerance in magnitude is greater for fast voltage variations, during short lapses of time. As with the steady state case, the

requirements for transient events are stricter for highly critical loads. These are general outlooks, but specifically, each load has its own susceptibility profile of failure regarding magnitude of voltage variations vs. duration.

Nowadays, there is a standard susceptibility graphic for equipments of information technology, computer equipment, copy and fax machines, and other electronic equipment. This is the CBEMA curve (IEEE Std 1100-2005, 2006). It reflects the magnitude tolerances of voltage variations vs. duration for the aforementioned equipments. The first version of this curve was proposed in 1977 by the Computer and Business Equipment Manufacturers Associations (CBEMA). Since then, the curve has been widely published and used in the literature. It was updated in 1996 and revised in 2000 by the Information Technology Industry Council (ITI), and is now an essential part of the standard 1100 of the Institute of Electrical and Electronics Engineers (IEEE).

The curve has been improved to the point that today it is a representative standard for modern computer equipment, and it is frequently used in the power quality field as a performance guide for loads in general, although not all the equipment manufacturers make their designs to meet the tolerance shown in it, and even though this curve does not reflect an absolute profile for all kind of loads, but only for the informatics and computing equipment. Figure 5 shows the CBEMA graphic. If the coordinates of magnitude vs. duration of a voltage variation fall outside the tolerance zone, the event could cause problems. For example, the graphic allows a +200% variation for an event lasting $1ms$; and permits a limit up to +500%, but for short impulses below $100\mu s$.

Characteristic	Requirements of little critical loads	Requirements of very critical loads
Stability of voltage at steady state	+10% -10%	+5% -8%
Stability of voltage at transient state	+20% -30% ($\leq 40ms$)	+20% -30% ($\leq 4ms$)
Short outages of less than 10ms	No disconnection but can generate errors	Disconnection exists and generate errors

Table 1. Requirements of the supply voltage for critical loads.

On the basis of this graphic, an algorithm that determines the existence of an electrical disturbance when the voltage surpasses a tolerance band (for example of $\pm6\%$) can be proposed, with the aim of controlling the corrective action of power quality compensators.

4. Detection of electrical disturbances with the Kalman filter and the digital RMS algorithm

4.1 Previous works on detection of electrical disturbances

It is evident that to perform correcting actions in an electrical system, an algorithm that detects the start of disturbances is necessary, as this action governs when a compensating device begins operating. Several methods for these purposes have been published, with different advantages and drawbacks. Among the most used, and relevant detection and estimation techniques are: space vector PWM (SVPWM) (Wang et al., 2004), fast Fourier transform (FFT), DQ transformation (Montero & Enjeti, 2005), symmetrical components
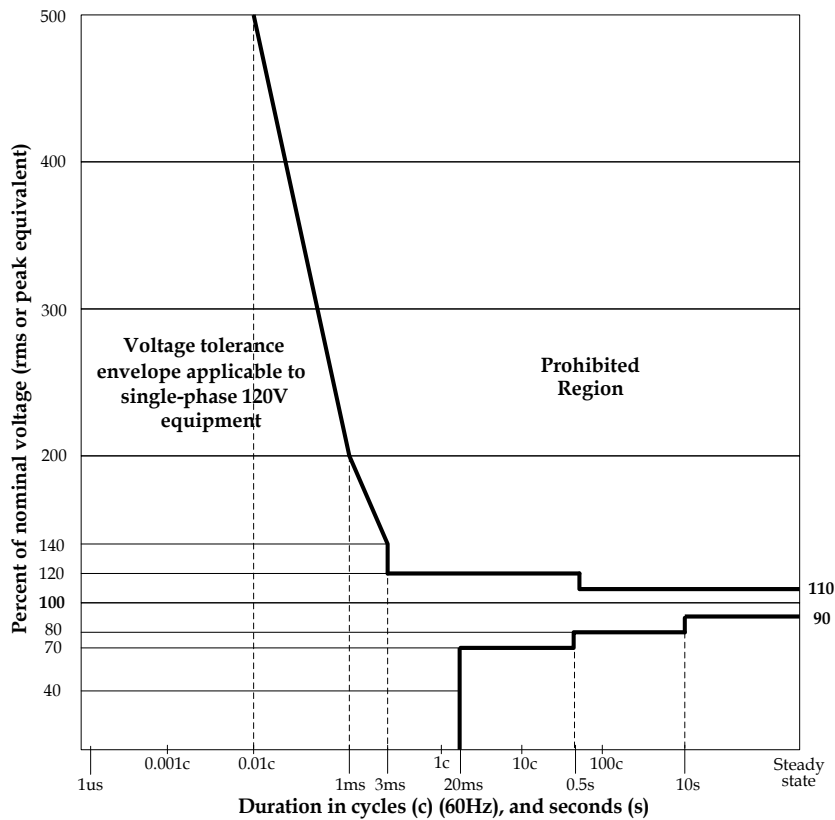
Fig. 5. ITI (CBEMA) curve (2000). Profile graphic of voltage variation vs. admissible time in the supply of modern electronic and computer equipment.

method, Kalman filter (Ma & Girgis, 1996), (Vilathgamuwa et al., 2003), (Yu et al., 2005), (Schwartzenberg et al., 1994), rms of the error vector (Nielsen et al., 2004), peaks detection (Fitzer et al., 2004), numerical matrix method (Fitzer et al., 2004), digital RMS calculation (Bollen, 2000), (Deckmann & Ferreira, 2002), and wavelet transform (Saleh & Rahman, 2004), (Dash et al., 2001).

The FFT process has slow performance due to the one cycle window requirement. The peaks detection method can take up to half cycle to detect a disturbance and the noise can detriment the performance. The numerical matrix method has response times around 7ms and the operation is sensible for high sampling rates, conducting to errors.

The method in (Dash et al., 2001) uses spline wavelets to detect the beginning and end of disturbances, and KF at steady state condition to classify and give information of the disturbance, such as frequency, depth, and phase (the KF is not used for detection). In addition, the dependence of the speed of the wavelets method on the starting phase angles of the disturbance is not considered. In fact, slow results can be obtained at some of them. Besides, the method does not take into account the influence of voltage harmonics. Their presence can guide to slower and less reliable detections.

One advantage of using the DQ transformation as in (Montero & Enjeti, 2005) and (Nielsen et al., 2004) is that only one algorithm is needed to monitor a complete three phase system.

However, the loss of information due to the three to one mapping gives more disadvantages. The method can become too slow, having a wide time range of detection, depending on the depth of the voltage variations, the simultaneousness of them on the three phases, and the electrical angles at which they appear. Moreover, if the system has harmonics, the algorithm must be configured to perform slower in order to be reliable, having then high probabilities of very slow detections.

Two of the fastest, with more advantages, and most used methods are the KF and the digital online RMS algorithm. The KF as applied in (Schwartzenberg et al., 1994) is used only for prediction of the future behavior of the signal at *N* samples in the future, to use it as part of a control system. It is not used for detection of disturbances. The signal is modeled taking *n* harmonics into account, but the main disadvantage is the high processing time consumption due to the big extension of the model.

In fact, the KF is applicable to help some active power compensators, by playing the roll of online detector of electrical disturbances, mainly of sags, swells, and interruptions. The detections are required to be as fast as possible, in order to give the compensator the command that will start the corrective action. The KF is a good candidate since it can be fast. However, it suffers slowness in estimations in some specific cases. In reality, as will be realized, the KF can be combined with another method to eliminate its drawbacks.

A fast detection algorithm for sags, swells, and interruptions based on the combination of the digital RMS technique and the KF, called the KF-RMS method, is addressed in section 4.4 (González et al., 2006), (González et al., 2008). The slow detection speeds the KF has at some regions are eliminated by the RMS algorithm, and the fast speeds it has at other regions are availed. A complete study and a range of detection times are derived, assuring high probabilities that detections of disturbances are very fast. The KF-RMS algorithm is faster than many algorithms and considers all possible starting angles of disturbances, all depths, voltage increases, and a voltage polluted by harmonics; thus considering all the cases, and not only particular approaches. Experimental tests are also discussed in section 4.6.

4.2 The Kalman filter as voltage estimator and detector of disturbances
The ideal case
In order to make an estimation of the rms value of a voltage with the KF, the following development can be made. If only the fundamental component of the voltage is considered to derive a model for the KF, it can be started from the following digitalized expression:

$$v[kT] = b_1 \sin[\omega kT + \phi] \tag{1}$$

where b_1 is the nominal peak value of the voltage, k is the sample number, and T is the sampling period. It can be decomposed as:

$$v[kT] = b_1 \cos\phi \sin[\omega kT] + b_1 \sin\phi \cos[\omega kT] \tag{2}$$

If the states are chosen to be $x_1 = \dfrac{b_1}{\sqrt{2}}\cos\phi$ and $x_2 = \dfrac{b_1}{\sqrt{2}}\sin\phi$, then the state space model can be expressed as:

$$\mathbf{X_k} = \mathbf{A X_{k-1}} + \mathbf{W_{k-1}} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k-1} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_{k-1} \tag{3}$$

where w_1 and w_2 allow the state variables to randomly move in time. It should be noted that at normal conditions (no phase-shift), the state x_1 has the maximum value $b_1/\sqrt{2}$ (rms value) whereas x_2 is zero. Then, the measurement matrix can be defined as:

$$\mathbf{H_k} = \left[\sqrt{2}\sin[\omega kT] \quad \sqrt{2}\cos[\omega kT] \right] \tag{4}$$

which must be synchronized with the source voltage. Next, the following well-known KF matrix equations are computed recursively at every sample:

$$\mathbf{X_k} = \mathbf{A}\mathbf{X_{k-1}} \tag{5}$$

$$\mathbf{P_k^-} = \mathbf{A}\mathbf{P_{k-1}^+}\mathbf{A^T} + \mathbf{Q_k} \tag{6}$$

$$\mathbf{K_k} = \mathbf{P_k^-}\mathbf{H^T}\left(\mathbf{H}\mathbf{P_k^-}\mathbf{H^T} + \mathbf{R}\right)^{-1} \tag{7}$$

where $\mathbf{P_k^-}$ is the prior estimate of the error covariance matrix, and $\mathbf{K_k}$ is the optimal Kalman gain matrix, which improves the estimation at every sample. Then a measurement is taken:

$$y_k = \mathbf{H}\mathbf{X_k} + v_k \tag{8}$$

where y_k is the sample and v_k is the noise of the measurement.

In KF, a prior estimate of the system at sample k, denoted by $\mathbf{X_k^-}$, is used to derive an updated estimate $\mathbf{X_k^+}$ based on the measurement y_k. This estimate is computed with:

$$\mathbf{X_k^+} = \mathbf{X_k^-} + \mathbf{K_k}(y_k - \mathbf{H}\mathbf{X_k^-}) \tag{9}$$

And finally, the updated estimate of the error covariance matrix, based on its prior estimate is calculated:

$$\mathbf{P_k^+} = \left(\mathbf{I} - \mathbf{K_k}\mathbf{H}\right)\mathbf{P_k^-} \tag{10}$$

The covariance matrix $\mathbf{Q}$ of w is given by:

$$E\left[w_i w_j^T\right] = \begin{cases} Q_i & i = j \\ 0 & i \neq j \end{cases}, \text{ or } \mathbf{Q} = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \tag{11}$$

and the covariance of v is $\mathbf{R} = R$. The signs + and – at all expressions are used to denote the times immediately before and immediately after a sample is taken.

Figure 6 shows the KF estimation of the rms value of an ideal sinusoidal voltage (without harmonics neither noise) with nominal value $b_1 = 179.6V$ ($127 Vrms$), in which a $0.4 p.u.$ sag (remaining $0.6 p.u.$ in the voltage) occurs at $t = 1s$.

As with many other algorithms, the KF has a trade-off: the speed can be improved but the cost is the insertion of high overshoots during transient response; or the estimation can be performed softly and without overshoots, but having a slower response in consequence. The values of the state variances in this case are chosen as $Q_1 = Q_2 = 0.004$, to obtain a reasonable fast speed in the estimation without having transient overshoots. Realize that the estimation of interest is only the one given by x_1, and that x_2 has no useful information. x_1 has a fast and flat response.

It takes about 9*ms* to get the new valid value at steady state. Nevertheless, together with the KF estimation, thresholds of +6% and -6% of the reference value $b_1/\sqrt{2}$ can be used to make the detection of the disturbance in a much shorter time. When the estimation x_1 gets outside this range, then the sag is detected. Detection time in this way is only 1.7*ms* (very fast) and a pulse shows the moment of detection in the third graph (zoom). The thresholds are set to $\pm 6\%$ in order meet the CBEMA guidelines of voltage supply, which establishes the limit at $\pm 10\%$, allowing normal voltage variations and having shorter times of detection. Note that there is also a trade-off in the selection of this tolerance band. The smaller it is, the more susceptibility for false detections, and as greater it is, less susceptibility.
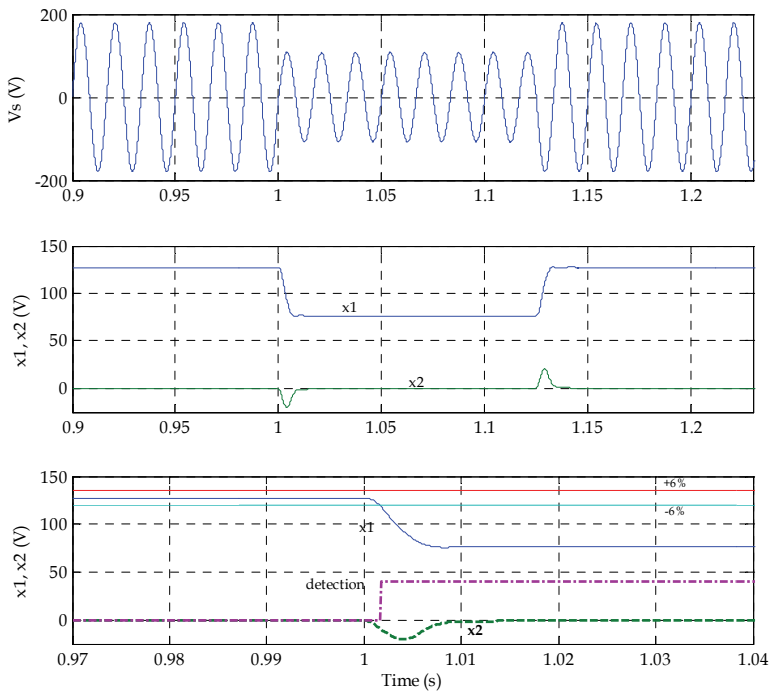
Fig. 6. Estimation with the KF of the rms value of an ideal supply voltage with nominal value 127*Vrms*. A 0.4*p.u.* sag occurs at *t*=1*s*, and its detection is shown.

The case with harmonics

The previous example of voltage is impossible to find in practice. These results of detection speed and flatness will never be exactly reproduced in the estimation of a real voltage. Voltage supply, even at residential facilities has the presence of harmonic distortion, and the situation could be much marked at industrial environments. Hence, a more practical and useful algorithm for detection of disturbances must consider a non clean voltage.

Numerous authors have proposed models of the KF to make estimations with harmonic distortion. Some examples are shown in (Girgis et al., 1991), (Schwartzenberg et al., 1994), (Ma & Girgis, 1996), (Yu et al., 2005). This can be made considering the harmonic terms in the signal:

$$v[kT] = b_1 \sin[\omega kT + \phi_1] + b_3 \sin[3\omega kT + \phi_3] + b_5 \sin[5\omega kT + \phi_5] + b_7 \sin[7\omega kT + \phi_7] + \ldots , \quad (12)$$

and deriving the KF model with all the components. The great advantage of doing this is that the magnitudes of each harmonic component can be calculated online separately, and can be used for monitoring. However, this makes the model (3) very big, with two states for each harmonic that is considered. E.g. if four harmonics were established, the dimension of the model would be 10. So the matrix operations involved in the KF become bulky, and the algorithm becomes impractical. The computational load itself can take a lot of time.

An approach to overcome this is by using the original model that considers only the fundamental component in (3) and (4), but managing it in such a way that it can work reasonable well in the presence of harmonics. The KF model based on only the information of the fundamental component has the benefit of high speed estimation, but it becomes oscillatory if the system has harmonics. The method can be manipulated to give higher speed, and also to give less oscillation on the estimation of x_1, by giving more weight to the state variance Q_2 of x_2 than to the state variance Q_1 of x_1. In this way a greater part of the oscillation (caused by harmonics) of the estimation is "sent" to x_2.

Figure 7 shows an example of the estimation with KF of the rms value of a voltage that is disturbed with a 0.4$p.u.$ rectangular sag at t=1s, and that has the following magnitudes of harmonics ($THD = 11.84\%$): $b_3 = 0.020b_1$, $b_5 = 0.100b_1$, $b_7 = 0.050b_1$, $b_{11} = 0.030b_1$, and $b_{13} = 0.015b_1$; with the fundamental component as $b_1 = 179.6V$ (127$Vrms$). These magnitudes of harmonics are selected to take above the worst THD case allowed in the standards IEC 61000-2-2 for low voltage, and IEC 61000-3-6 for medium and high voltage power supply systems (which are set to 8%); and above the recommended limits in the IEEE Std 519-1992 (which are set to 5% for low voltage, 2.5% for medium voltage, and 1.5% for high voltage systems). This represents a though and highly distorted condition to test the algorithm, really giving it high usefulness for real practical systems. The state variances are now set to $Q_1 = 0.0009$ and $Q_2 = 0.0625$. These values of Q give an adequate speed with low oscillation in the x_1 estimation.
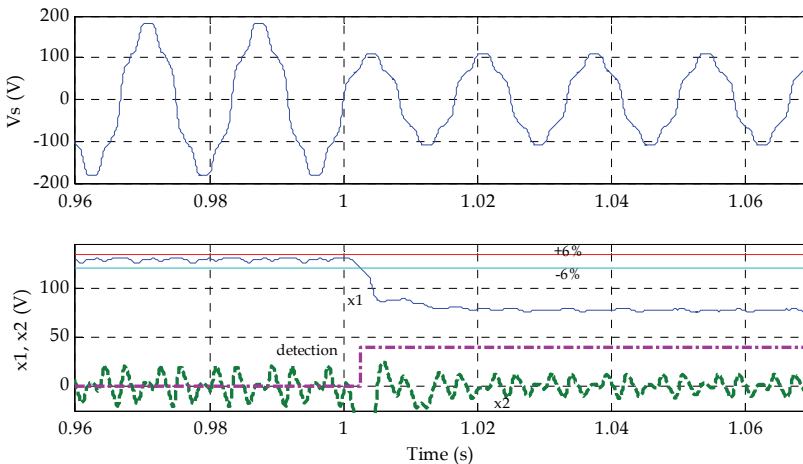
Fig. 7. Estimation of the rms value of a distorted supply voltage with the KF. The detection of a 0.4$p.u.$ sag is also shown.

Although there is a little oscillation on x_1, it can be noted that a greater part is sent to x_2. In this case, x_1 takes about 14ms to get the new valid value. As with the ideal case, the same thresholds of +6% and -6% of the reference value $b_1/\sqrt{2}$ are used to detect the disturbance. The little oscillation is allowed by this tolerance band satisfactorily. Detection time in this case is 2.6ms and a pulse shows the moment of detection. The estimation speed has to be slightly decreased than in the ideal case, to have a more reliable estimation that involves the harmonic distortion of the measured supply voltage, allowing a correct disturbance detection. This behavior is good enough to have an useful algorithm with low processing work, fast, and with easy implementation.

Consideration of the starting angle of the disturbance in the Kalman filter algorithm

The previous examples showed the beginning of the event at $t = 1s$ in a 60Hz system, i.e. at an angle of 0° of the signal. The dependence of the estimation speed of the KF with the electrical angle at which the disturbance appears, and the presence of the harmonics listed in the previous example, are considered in the following analysis.

Figure 8 shows a graphic of detection time vs. starting angle of a rectangular sag of 0.4$p.u.$, considering the harmonics. It can be observed that at some regions of phase the KF is very fast (less than 1ms), but at others, slow detections are experienced. For instance note that at 100° the detection time is around 6ms, whereas at 70° it is much less than 1ms. The range of detection for sags of 0.4$p.u.$ is $0.18ms \le t_d \le 6.10ms$. It is important to consider all the cases in an algorithm, to evaluate if it is really going to be fast at all circumstances. Numerous authors have reported fast speeds with other algorithms or even with the KF, but actually show results at favourable angles that give fast estimations. This is evident in Figure 8.

The detection times that can be obtained, considering the previously mentioned levels of harmonics and the starting angle of the disturbance, result in a wide range of values. The digital RMS algorithm is other of the methods that give very good performance with relatively high speed. This method can be used to help obtaining faster estimations at the regions the KF shows slowness. The performance of the RMS method is addressed next, and then how it can help the KF, in a unified method, in the following section.
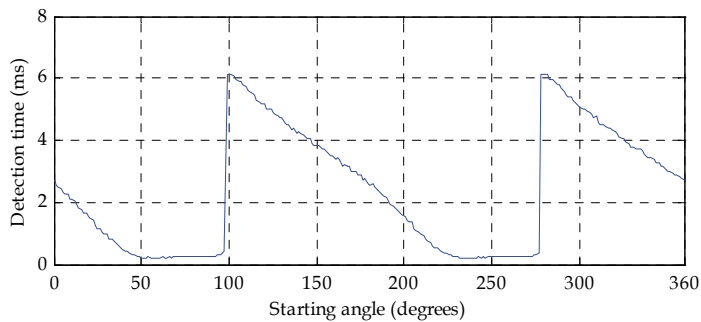
Fig. 8. Detection time vs. starting angle of a 0.4$p.u.$ rectangular sag using the KF.

4.3 The digital recursive RMS calculation method as detector of disturbances

The expression for the well-known online digital RMS calculation is given by:

$$V_{RMS}[k] = \sqrt{\frac{1}{N} \sum_{i=k-N+1}^{k} v^2[i]} \qquad (13)$$

where N is the sliding window size for the computation, k is the number of sample, and v_i are the samples. There are two kinds of application, the one cycle window, and the half cycle window. The method needs N to be an integer multiple of one cycle or of a half cycle, in order to avoid oscillations on the estimation. A recursive alternative to spend less processing time can be found in (González et al., 2006). It provides a significant processing time saving when N is large. The RMS method has the advantage of providing a very flat estimation, even if there are harmonics in the voltage, which translates into reliability.

The one cycle RMS technique takes one cycle in giving a valid result for a change in the voltage, and the half cycle RMS method takes a half cycle. However, if inferior and superior thresholds are also defined around a reference voltage, both window sizes can be used for a faster detection of disturbances when the estimated voltage value gets outside them. As the half cycle RMS takes less time to get to a new value on transitions, it is clear that it is also faster to detect disturbances when using thresholds. Thus, it is used in the following, also with the thresholds of ±6% around the reference voltage to meet the CBEMA guidelines, and to allow faster detections, as selected for the KF.

Figure 9 shows the calculation of the rms value of the voltage of the same example shown for the KF (0.4$p.u.$ sag, with the same harmonics: $b_3 = 0.020b_1$, $b_5 = 0.100b_1$, $b_7 = 0.050b_1$, $b_{11} = 0.030b_1$, and $b_{13} = 0.015b_1$, with $b_1 = 179.6V$ ($127Vrms$)), but now using the half cycle RMS method. The estimation takes half cycle to get to the new valid value, but detection time is 2.8ms, a very similar result than the one of the KF. A pulse shows the moment of detection. Note that the estimation is soft and flat, even with the considered harmonics.

Consideration of the starting angle of the disturbance in the RMS algorithm

The speed of this method is found to be dependant on the starting angle of the disturbance too. Figure 10 shows the graphic of detection time vs. starting phase angle of a 0.4$p.u.$ rectangular sag (the harmonics of the previous examples are considered too). At some regions the method is slow and at others is fast, but not with marked behavior as the KF. The detection range for a 0.4$p.u.$ sag is $0.79ms \le t_d \le 3.96ms$.
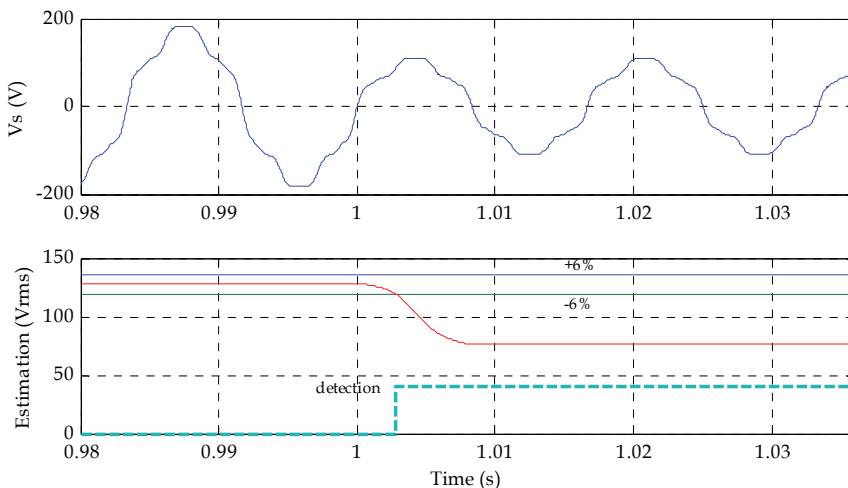
Fig. 9. Estimation of the rms value of a distorted voltage with the digital RMS algorithm. The detection of the occurrence of a 0.4$p.u.$ sag is also shown.
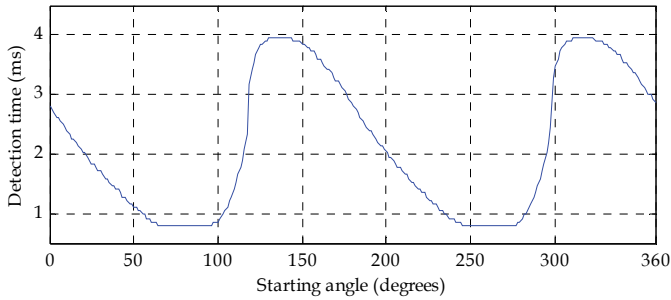
Fig. 10. Detection time vs. starting angle of a 0.4*p.u.* sag, using the digital RMS algorithm.

4.4 Method for detection of disturbances based on the combination of the KF and RMS algorithms: the KF-RMS method (González et al. 2006), (González et al. 2008)

The graphics speed of detection vs. starting angle of a 0.4*p.u.* sag of the KF and RMS methods are shown together in Figure 11. Observe that in some cases, depending on the starting angle of the disturbance, the KF is faster than the RMS, whereas in other cases the RMS is faster. This graph leads to the proposal of the KF-RMS method, based on the combination of the KF and the digital half cycle RMS algorithm. The central idea can be noticed by observing the advantages and drawbacks of speed of each one of them. The method consists on monitoring with the online execution of both algorithms simultaneously, and taking the decision of the one that first detects the disturbance (logic OR of detecting signals). Figure 12 shows the resultant graphic for the KF-RMS method.
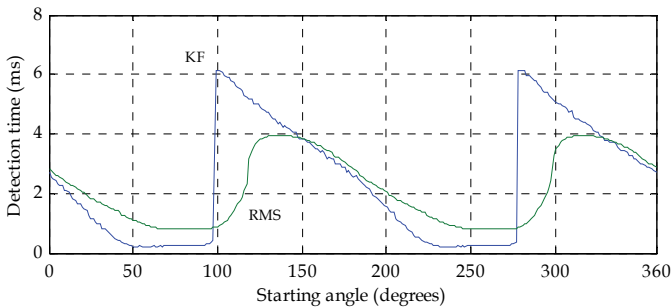
Fig. 11. Detection time vs. starting angle of a 0.4*p.u.* rectangular sag, occurring in a signal distorted by harmonics. Both KF and digital RMS algorithms are shown.
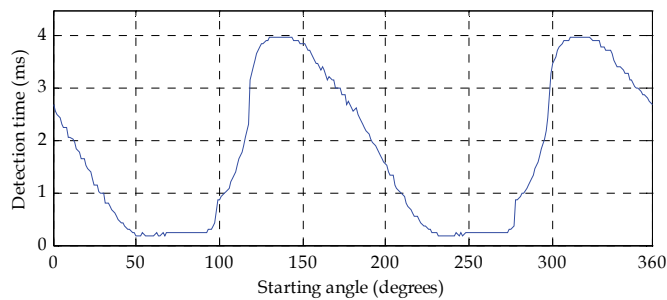
Fig. 12. Detection time vs. starting angle of 0.4*p.u.* rectangular sag occurring in a distorted voltage signal, with the KF-RMS method.

In this way, very fast detections are obtained with the KF, and the regions at which it is slow are rejected by the RMS method. Detection time is now between 0.18*ms* and 3.96*ms*, which clearly falls in less than a quarter of cycle of 60Hz systems. However, this is only valid for 0.4*p.u.* rectangular sags, since all depths are not considered.

A very few authors have considered the dependence of speed of the KF and the RMS methods on the depth and on the starting angle of the disturbance together, and the small amount that consider it, do not consider a voltage distorted by harmonics. Considering all these aspects is crucial as a validation of one reliable method at all possible events can be obtained, not only for particular cases. Hence, to make a generalization, 3D graphics are developed next. The detection time, the angle at which the disturbance begins, and the sag depth are considered. The high levels of voltage harmonics addressed in the previous sections are also considered.

Figure 13 shows the corresponding 3D graphic for the KF method. The analysis considers depths from 0.1*p.u.* to 1.0*p.u.* (complete interruption). The same behavior of Figure 8 can be noted, but it can be seen that as the depth increases, the detection becomes faster. At all depths the method is fast at some regions of angles, while at others it is slow, as with the 0.4*p.u.* sag case. Note that the maximum detection time is 9*ms*, which is a high value of time (more than half cycle). If a sag occurs near these regions the method performs slowly. Considering all points, the KF method has a range of detection times of $0.061ms \le t_d \le 9ms$.
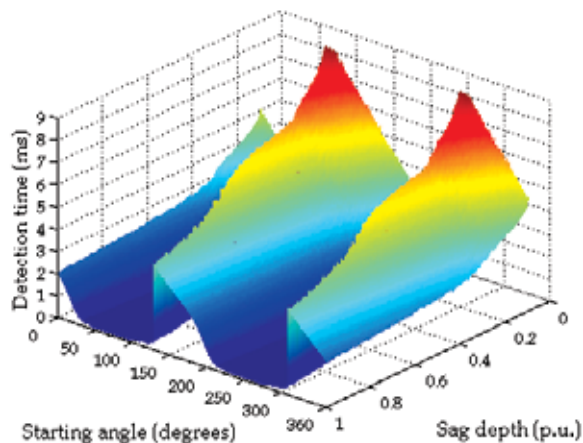
Fig. 13. Detection time vs. starting angle vs. sag depth with the KF algorithm.

Figure 14 shows the 3D graphic for the RMS algorithm. The same behavior as in Figure 10 is observed at all depths too, and the detection speed also increases with depth. The method has a detection time range of $0.48ms \le t_d \le 6.9ms$.

If the KF-RMS algorithm is executed, the 3D graphic of Figure 15 results. Observe that the advantage of fast detections in some regions of KF is availed, and the regions at which KF is slow are rejected by the RMS method. The range for detection in general is now $0.061ms \le t_d \le 6.9ms$. Figure 16 shows the same graphic but with a plane that represents the quarter of cycle (4.16*ms* at 60*Hz*). It can be seen that the majority of the events are below this time. Extracting probabilities from all the points, it can be established that, for rectangular sags between 0.1*p.u.* and 1.0*p.u.*, and supposing that all the points come out with the same
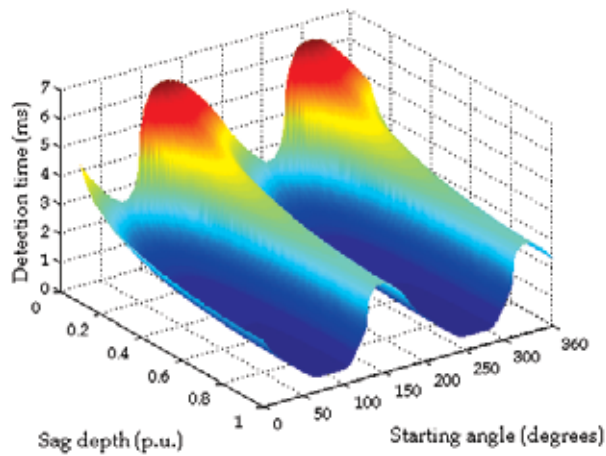
Fig. 14. Detection time vs. starting angle vs. sag depth with the digital RMS algorithm.
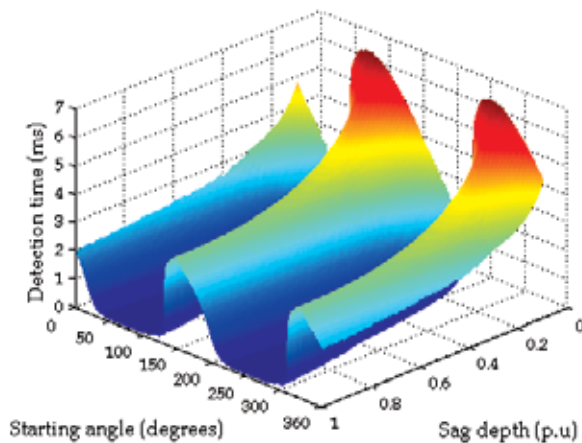
Fig. 15. Detection time vs. starting angle vs. sag depth with the KF-RMS algorithm.

recurrence (in practical systems short sag depths are more probable), the KF-RMS algorithm has the following properties:

• Range of detection: $0.061ms \leq t_d \leq 6.9ms$. In a distorted environment: $THD = 11.84\%$

• $prob(t_d > 4.16ms) = 9.04\%$

• $prob(t_d < 4.16ms) = 90.9\%$

• $prob(t_d < 2.00ms) = 56.9\%$

The method has a very high probability (90.9%) that sags and interruptions are detected in less than a quarter of cycle, and more than 50% of probability (56.9%) that they are detected in less than 2ms, which is very fast. Times a little greater than a quarter of cycle may be obtained with sags of small depth, but only at some specific regions of phase. In fact, detection times will never be equal or greater than half cycle.

All the previous analyses are made for swells. A better result, with much more favorable behavior is obtained for swells with the KF-RMS method. Figure 17 shows the corresponding 3D graphic. It can be seen that most of the events are below 2*ms*. The following properties are obtained for swells from 1.1*p.u.* to 2.0*p.u.*:

- Range of detection: $0ms \leq t_d \leq 6.3ms$. In a distorted environment: $THD = 11.84\%$

- $prob(t_d > 4.16ms) = 3.95\%$

- $prob(t_d < 4.16ms) = 96.04\%$

- $prob(t_d < 2.00ms) = 78.87\%$

Detection times of practically 0*ms* are obtained for several swell cases, leaving all the dependence to the processing time, and the probabilities are much better than for sags and interruptions.
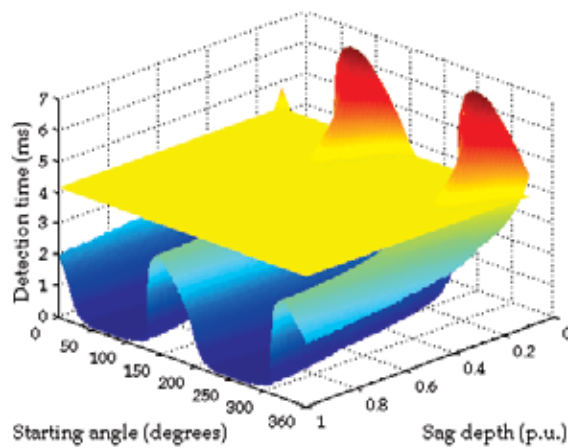
Fig. 16. Probability that sags and interruptions are detected in less than a quarter of cycle, with the KF-RMS algorithm.
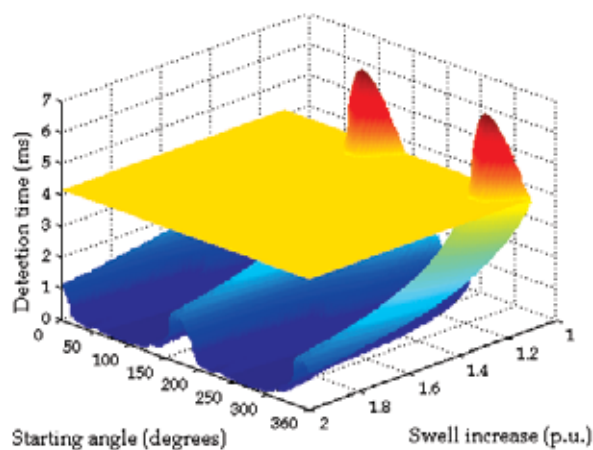
Fig. 17. Detection time vs. starting angle vs. swell increase with the KF-RMS algorithm.

The KF-RMS algorithm represents a better option for detection of voltage variation disturbances. It is important to note that the angle in which the disturbance occurs and its depth are considered in the analysis, besides of an amount of harmonic pollution that is superior to the allowed by the most important worldwide standards on voltage harmonic levels in practical electrical systems. The advantage of a very fast detection at some regions of KF is availed, and the regions at which it is slow are rejected by the RMS method.

4.5 Dependence of the KF-RMS algorithm on the sampling frequency

The previous results were derived with a sampling frequency equal to $16.384 kHz$ and with a $60Hz$ supply frequency. The performance of the method also shows dependence on the sampling frequency selection. Figure 18 shows how the speed of detection is affected in both KF and RMS methods when the sampling frequency is changed, for a $0.4p.u.$ sag starting at $0°$ (including also the harmonics that have been considered through all the examples). It can be noted predominantly that the KF sees its detection speed negatively affected if the sampling frequency is reduced, whereas the RMS algorithm maintains it practically at all sample frequencies.
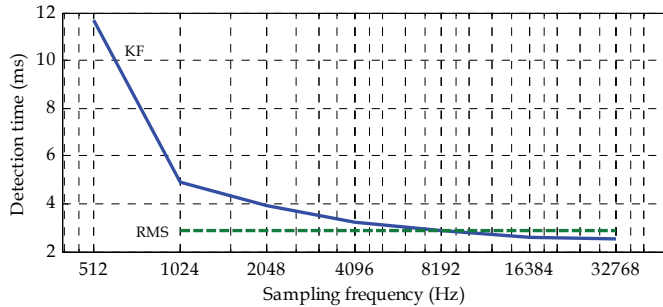
Fig. 18. Dependence of the detection time of KF and RMS methods on the variation of the sampling frequency, for a $0.4p.u.$ rectangular sag starting at electrical $0°$.

Then, the global response of the KF-RMS method also depends on the sampling frequency. When decreased, the regions at which this method avails the high speeds of KF are lost, but the rejection of slow speeds with RMS is kept. In other words, if the sampling frequency is reduced, the behavior of the KF-RMS method will get close to the RMS one operating alone, since the global response will depend only on it. An important care must be taken when varying the sampling frequency: the RMS method must keep the number of samples of its calculation window an integer multiple of half cycle. Also, care must be taken in always keeping the Nyquist criterion satisfied on both methods (and its limit far enough) when the sampling frequency is reduced; in order to consider from the fundamental to a specific harmonic component, so aliasing does not appear that would guide to erroneous estimations.

The method is found to show its main benefits and work better with sampling frequencies greater than $8.192 kHz$. This statement is valid if the constants of the covariance matrix **Q** of the KF are kept fixed. Indeed, the optimal constants for one sampling frequency are not the most appropriate for others. In order to maintain the advantages of the KF-RMS method at all sampling frequencies, an adequate selection of the Q_1 and Q_2 constants is needed at each case. In this way the dependence of the KF on the sampling frequency is eliminated. Figure 19 shows that the dependence on sampling frequency is eliminated by selecting adequate Q constants. The selections are summarized in Table 2.
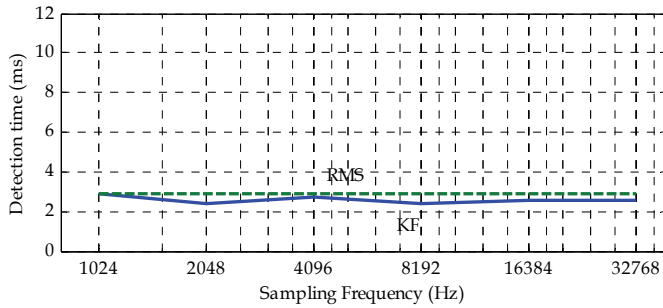
Fig. 19. Detection time of the KF and RMS methods, for a 0.4*p.u.* rectangular sag starting at 0°, and using adequate values Q_1 and Q_2 for different sampling frequencies.

Sampling Frequency	Q_1	Q_2
1024	1.0000	25.0000
2048	0.1296	6.2500
4096	0.0784	6.2500
8192	0.0400	3.2400
16384	0.0009	0.0625
32768	0.0169	1.0000

Table 2. Selection of covariance matrix constants of KF for different sampling frequencies.

4.6 Experimental tests of the KF-RMS algorithm

The KF-RMS algorithm is implemented in the Texas Instruments TMS320F2812 DSP platform. A single phase voltage signal is sampled, and scaled to 0-3*V* in order to use the internal ADC of the DSP. Then sags and swells of different magnitudes are emulated at the sensor. The experimental results of the KF-RMS method agree significantly with the analyses, and with the detection time graphics of Figures 16 and 17. This is proved with several generated events. Detection cases are shown in Figure 20 for sags, and in Figure 21 for swells. Very fast detection times are obtained in cases 20(a), 20(c), 20(d), 21(a), and 21(d) due to the region at which the disturbance occurs. A pulse is generated in one port pin of the DSP to show the moment of detection. The *THD* of the sampled signal is around 5%.

(a)

(b)

(c)   (d)
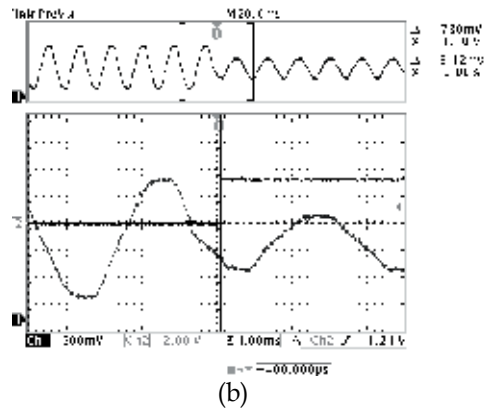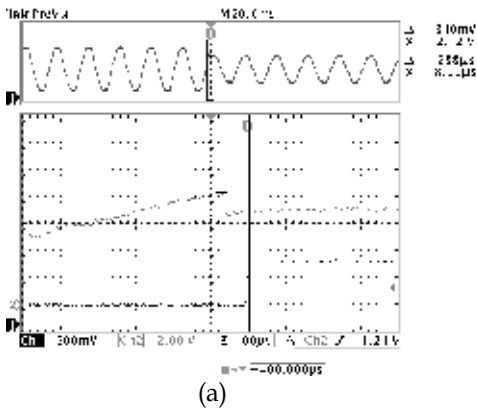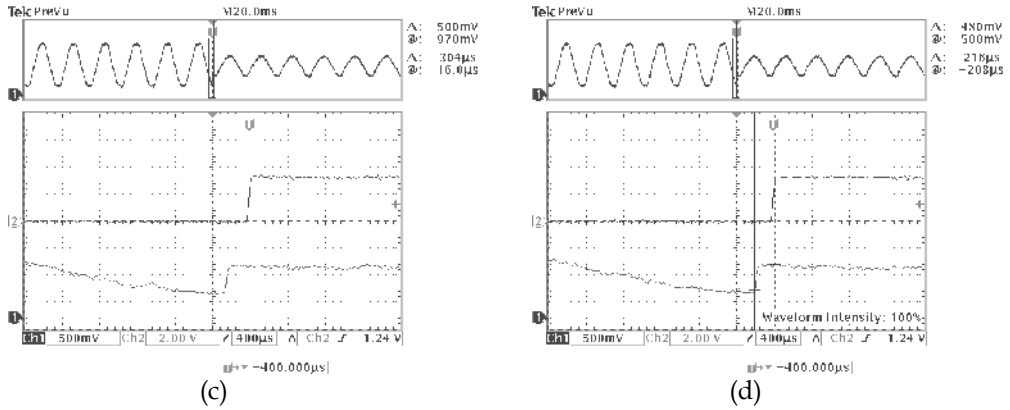
Fig. 20. Experimental detection times of the KF-RMS algorithm for sags. a) 0.4*p.u.* sag at 80°, $t_d = 288\mu s$. b) 0.5*p.u.* sag at 170°, $t_d = 3.12ms$. c) 0.5*p.u.* sag at 260°, $t_d = 304\mu s$. d) 0.5*p.u.* sag at 270°, $t_d = 216\mu s$.

(a)   (b)

(c)   (d)
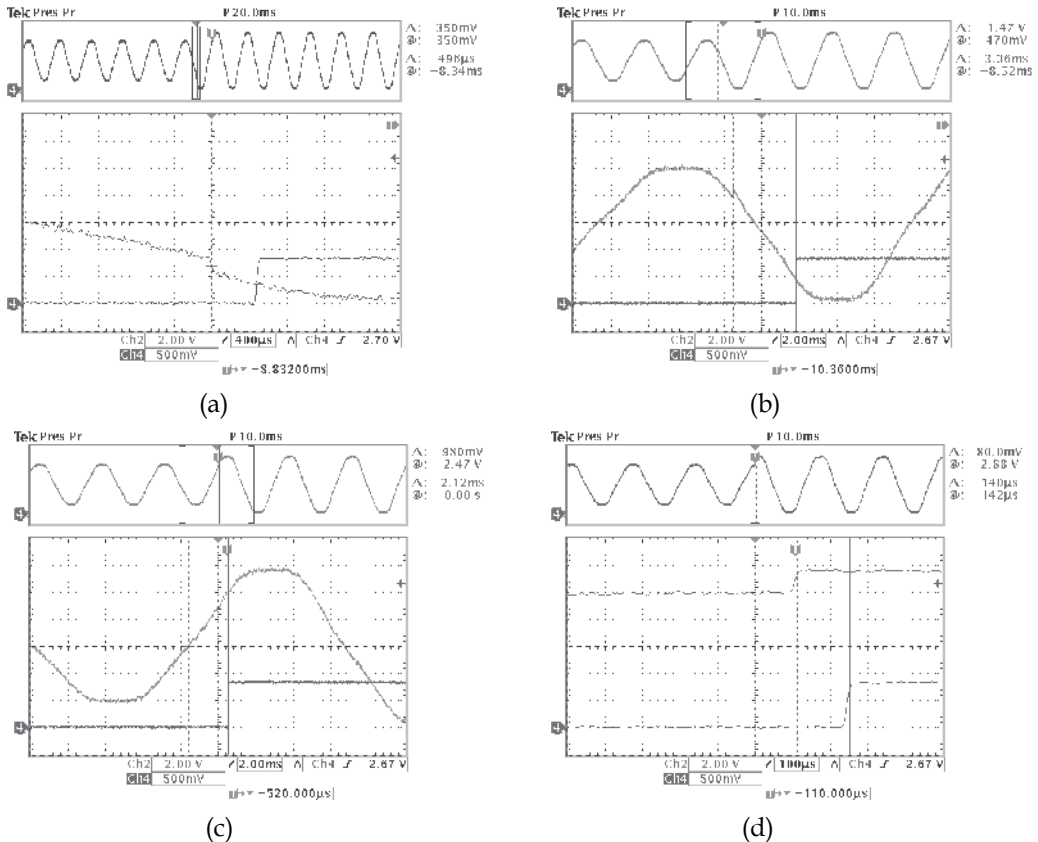
Fig. 21. Experimental detection times of the KF-RMS algorithm for swells. a) 0.4*p.u.* swell at 224°, $t_d = 496\mu s$. b) 0.4*p.u.* swell at 153°, $t_d = 3.36ms$. c) 0.4*p.u.* swell at 0°, $t_d = 2.12ms$. d) 0.4*p.u.* swell at 80°, $t_d = 140\mu s$.

5. The Kalman Filter applied for estimation of parameters in power quality monitoring of electrical systems

Another application of the KF in power quality is as estimator of parameters for monitoring purposes. Its fast real-time estimator characteristics make itself a good candidate.

Monitoring consists on obtaining information of one or more points of an electrical system, such as the behavior of a voltage or a current within a lapse of time, which can be from a few seconds to several days. This information can be recorded and used to generate the statistics of disturbances in that system, to evaluate power quality problems, and propose solutions. The information collected by voltage or current monitoring can be parameters such as magnitude, phase, and frequency of the fundamental and/or harmonic components at steady state. The number of transient disturbances within a range of time, their classifications, and durations can also be considered and recorded.

Generally, monitoring performed by most of the devices consists of four or five steps, depending of the application, and are mainly: the sampling of the signal of voltage or current of interest, the real time estimation of the magnitude, detection of disturbances, classification and characterization of the event, and compact storage of the collected data (Deckmann & Ferreira, 2002), (IEEE Std 1159.3-2003, 2004).

The voltage magnitude can be determined in a variety of ways. Many actual monitors obtain the steady state magnitude and also those of disturbances such as sags, swells, and interruptions from the digital RMS calculation (see the above bibliography and (Bollen, 2000)). Although the half cycle choice is faster, the one cycle window is frequently used. This algorithm has important advantages, such as the very flat estimation, even with the presence of harmonics, and that no matter how deep and how the shape of a disturbance is, it takes exactly a half cycle or one cycle (depending on the selected window) to reach the new valid value. This method is very useful for steady state classification of disturbances, but it only gives information related with the magnitude. Other parameters such as the fundamental frequency, the phase, and harmonic magnitudes and phases can not be interpreted. Moreover, the RMS method calculates the rms value of the signal as a whole, that is, including the fundamental and harmonic components together in the result. So, the estimation is not exactly the value of the real fundamental magnitude.

 Regarding this, the KF has great advantages, as it can make the aforementioned parameter estimations simultaneously. Another method that can perform magnitude, frequency and phase estimations of all harmonic components is the Fast Fourier transform (FFT), but requires one cycle to update results. A combination of various methods can be used to obtain fast, reliable, and accurate monitoring with fast transient response. In the following, only the KF applications are assessed. Some cases of monitoring are shown, evidencing its usefulness and advantages in monitoring.

5.1 Monitoring of the fundamental component in a distorted voltage

As stated, the KF can be used for rms or peak value magnitude estimation. To classify this information in monitoring, a flat and accurate estimation is needed. At first intuition, the more complicated model with harmonics could be taken for this aim, but in fact the model that only considers the fundamental component in (3) and (4) can be used effectively, using much less processing work. In this case, as the KF is not focused for detection, but for monitoring, a different tuning of the Q matrix can be used. For instance, $Q_1 = 0.08$ and $Q_2 = 80$ are adequate values for this purpose.

Figure 22 shows the estimation of the fundamental component in a voltage distorted by harmonics (the same values that were used in the previous section are used: $b_3 = 0.020b_1$, $b_5 = 0.100b_1$, $b_7 = 0.050b_1$, $b_{11} = 0.030b_1$, and $b_{13} = 0.015b_1$, with $b_1 = 179.6V$ ($127Vrms$)), with the KF and also with the RMS method to compare their monitoring performances. A $0.25p.u.$ swell occurs at $t = 1s$ and ends at $t = 1.2s$ to show the transient responses. Due to the mentioned tuning of the KF, a flat estimation is achieved, similar to the one obtained with the RMS method. Note also that a very abrupt estimation change to the final value is observed, which seems to reveal the KF estimation is extremely fast. However, this is only in appearance, as the estimation actually takes about a quarter of cycle to respond from the time the disturbance occurred at $t = 1s$.

Even with this delay, the KF indeed is faster in getting the final value in this case (around a quarter cycle vs. the half cycle of the RMS), and the opposite occurs in the examples of Figures 7 and 9, i.e. the RMS gets the steady state final value first. However the goal of those cases is the detection of the disturbance. If the KF were used for detection with the tuning of this example, its detections would actually be slower than the RMS. The tuning of those examples is adequate to have a faster estimation (although with tolerable oscillations), and hence allowing faster detections.

Thus, although a very flat estimation is obtained with the KF in the application of this example, it is not as useful for fast detection of disturbances, but better for transient and steady state monitoring.
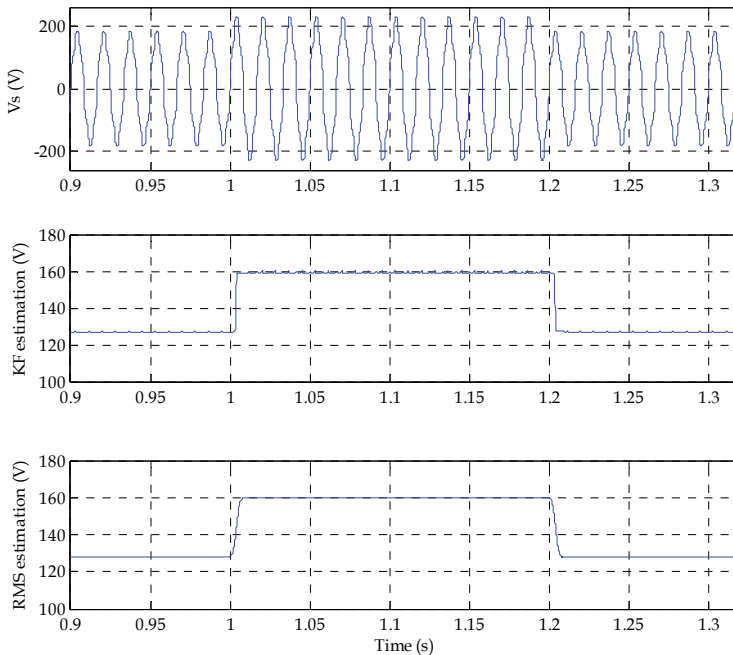
Fig. 22. RMS value estimation of the fundamental component of a distorted voltage, with the KF and RMS algorithms. A 0.25*p.u.* swell occurs at *t*=1*s* and ends at *t*=1.2*s*.

5.2 Monitoring of the magnitude and phase of the fundamental and harmonic components

If the information of the fundamental and harmonic components is needed to monitor in an application, an extended model of the KF considering harmonics can be used. Although the computational load becomes high, the great advantage is that the magnitudes and phases of all present harmonics in a current or voltage signal can all be estimated simultaneously. The application of estimating the magnitude of harmonic components has been widely published in literature throughout many years. Some works can be seen in (Girgis et al., 1991), (Schwartzenberg et al., 1994), (Ma & Girgis, 1996), (Yu et al., 2005). An approach to estimate also the phases simultaneously is adressed next.

The KF model can be defined considering the signal with its harmonic components:

$$v[kT] = b_1 \sin[\omega kT + \phi_1] + b_2 \sin[2\omega kT + \phi_2] + \dots\dots\dots + b_n \sin[n\omega kT + \phi_n] \tag{14}$$

which can be decomposed as:

$$v[kT] = b_1 \cos\phi_1 \sin[\omega kT] + b_1 \sin\phi_1 \cos[\omega kT] + b_2 \cos\phi_2 \sin[2\omega kT] + b_2 \sin\phi_2 \cos[2\omega kT] + \dots$$
$$\dots\dots + b_n \cos\phi_n \sin[n\omega kT] + b_n \sin\phi_n \cos[n\omega kT] \tag{15}$$

As with the states definition of the first example in (2), two states are now needed for each harmonic component. These can be defined as: $x_{11} = \dfrac{b_1}{\sqrt{2}}\cos\phi_1$, $x_{12} = \dfrac{b_1}{\sqrt{2}}\sin\phi_1$, $x_{21} = \dfrac{b_2}{\sqrt{2}}\cos\phi_2$, $x_{22} = \dfrac{b_2}{\sqrt{2}}\sin\phi_2$,……., $x_{n1} = \dfrac{b_n}{\sqrt{2}}\cos\phi_n$, $x_{n2} = \dfrac{b_n}{\sqrt{2}}\sin\phi_n$. Thus, the general measurement matrix can be written as:

$$\mathbf{H_k} = \begin{bmatrix} \sqrt{2}\sin[\omega kT] & \sqrt{2}\cos[\omega kT] & \sqrt{2}\sin[2\omega kT] & \sqrt{2}\cos[2\omega kT] \\ & \dots\dots\dots & \sqrt{2}\sin[n\omega kT] & \sqrt{2}\cos[n\omega kT] \end{bmatrix} \tag{16}$$

Then, execution of recursive equations (5) to (10) of the KF is performed to obtain the online estimation of all the involved components. Observe that the two associated states of each harmonic are indeed the in-phase and the quadrature components of each one. The measurement matrix is synchronized with the fundamental component, but the harmonic components are rarely in phase with it in practice. Hence, the phase-shifts with respect to 0° make that both states of each harmonic share information of their actual magnitude and actual phase. One state alone can not reveal the complete information of each component. Therefore, the magnitude and phase of each harmonic can be determined with the following expressions (also online at every sample):

$$Mag_n[kT] = \sqrt{\left(x_{n1}[kT]\right)^2 + \left(x_{n2}[kT]\right)^2} \tag{17}$$

$$Phase_n[kT] = \sin^{-1}\left[\frac{x_{n2}[kT]}{\sqrt{\left(x_{n1}[kT]\right)^2 + \left(x_{n2}[kT]\right)^2}}\right] \tag{18}$$

which are accurate since the KF model that considers harmonics has the advantage that the estimation of both states of each harmonic stabilize with no oscillations. The only care in (18)

is to identify the quadrant at which the states form their vector, and add the corresponding adjustment angle. To show an example, the simultaneous estimation of the magnitude and phase of the fundamental and harmonic components of the same distorted signal of the previous example, but with different phases on the harmonics ($\phi_3 = 15°$, $\phi_5 = 150°$, $\phi_7 = 120°$, $\phi_{11} = -60°$ and $\phi_{13} = -45°$), is shown in Figures 23, 24, and 25. Since the fundamental and 3rd, 5th, 7th, 11th, and 13th harmonics are considered, the dimension of the model is 12. Figure 23 shows the original signal and the estimation of the fundamental magnitude. Figure 24 shows the rest of the harmonic amplitude estimations, all in rms values. The estimation of the phases of all components is shown in Figure 25. Observe that all states take around $20ms$ to reach the final steady state values. The speed results are not as fast as in the previous example, due to the interaction and dependence of all components with each other in the matrix operations. However these estimations are very useful for steady state monitoring, since harmonics do not have fast changes in electrical systems. It should be noted that all the parameters are estimated online and simultaneously, and that all the delivered estimations are flat and accurate. The monitoring of harmonics in a point of an electrical system is a very common practice in power quality, and can be used to solve a variety of problems they cause.
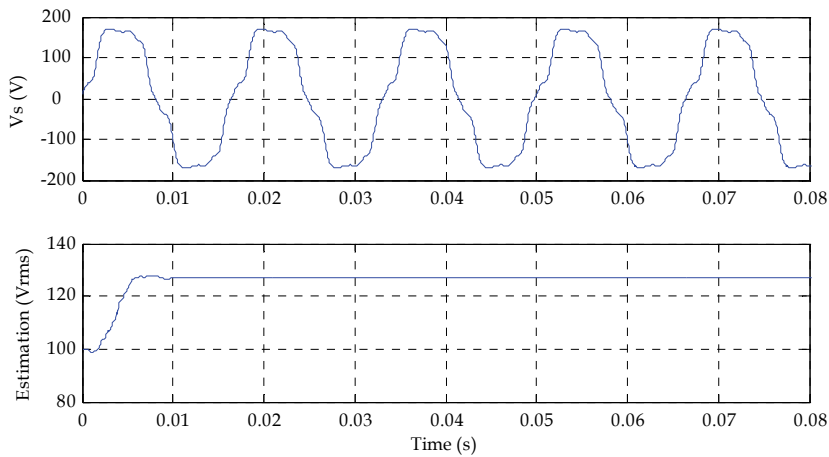
Fig. 23. Estimation of the fundamental component of the voltage signal with harmonics. The complete model that considers the fundamental and harmonic components is used.
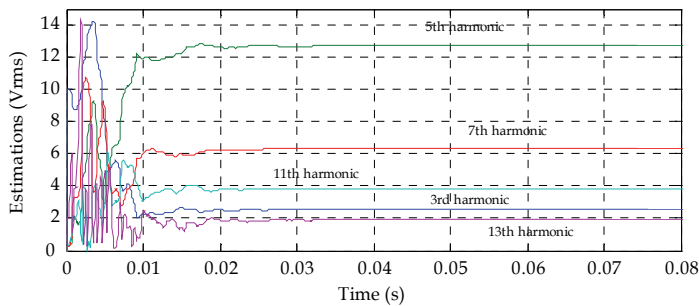
Fig. 24. Magnitude estimation of the harmonic components of the distorted voltage.
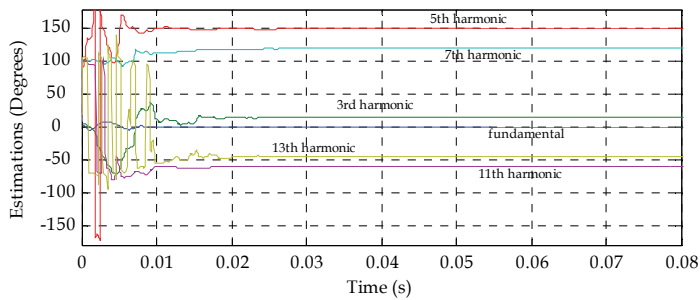
Fig. 25. Phase estimation of the harmonic components of the distorted voltage.

Other parameter that can be estimated with the KF is the frequency of a signal. One approach that estimates the magnitude, phase, and frequency of a signal simultaneously is discussed in [Dash et al., 2001], using an extended complex Kalman filter (ECKF). In fact, the estimation of all the mentioned parameters can be made simultaneously, another different parameter can be estimated, or any other combination of parameters can be extracted with the KF. The main problem is the model definition, which must be proposed adequately to have the desired estimations.

6. Conclusions

The main applications of the Kalman filter in the power quality field of electrical engineering are discussed in this chapter.

Due to its high-quality characteristics as estimator, the KF can be applied as an algorithm that is part of the control stage of some active power compensators to fast detect in real time voltage disturbances, mainly sags, swells, and interruptions. Detection of disturbances on an ideal voltage and on a more realistic case with harmonics are evaluated and compared. A simple low order KF model that considers the influence of harmonics in the signal is discussed and validated. The dependence of the speed of detection of the KF on the starting angle of the disturbances is also addressed. One drawback is that the KF has some regions of slowness.

To overcome this, a fast detection method for sags, swells, and interruptions based on the combination of the KF and the digital RMS algorithm can be used. When combined, the advantage of high speed of the KF at some regions is availed, and the slowness it has at other regions is rejected with the digital RMS method. The KF-RMS algorithm considers the dependence of the time of detection on all possible electrical angles at which a disturbance could appear, sag depths from $0.1 p.u.$ to the complete interruption, swell rises from $1.1 p.u.$ to $2.0 p.u.$, and an amount of harmonics that is higher than the allowed by the most important worldwide standards on voltage harmonic levels in practical electrical systems, leaving apart in this way the ideal and non practical case. The method has a very high probability that all events are detected in less than a quarter of cycle, and more than 50% of probability they are detected in less than $2 ms$, which can be considered very fast. A small zone of angles and depths at which the method takes more than a quarter of cycle to detect disturbances exists, but indeed, detection times will never be equal or greater than half cycle. The performance of the KF-RMS method is also affected by the sampling frequency. However, this dependence can be eliminated by selecting an adequate **Q** matrix for each sampling frequency. Experimental tests of the method validate and strengthen the contribution in detection of disturbances.

The application of the KF as estimator of parameters for power quality monitoring is also addressed. It can be applied for the estimation of the magnitude, frequency, and phase of the fundamental and harmonic components of a signal, in an independent way, or simultaneously using any combination of them. The collected information can be recorded and used to obtain statistics on the behavior of a voltage or current within a lapse of time, such as the steady state parameters and the number voltage variations, with their classification and characterization. This monitoring information can be used to propose solutions for power quality improvement. The correct definition of the model is crucial to have the desired estimations. An actual fact is that the KF proves to be a very useful and advantageous algorithm to help improving power quality in a variety of ways.

7. References

Affolter, R. & Connell, B. (2003). Experience with a dynamic voltage restorer for a critical manufacturing facility, *Proceedings of the IEEE PES Transmission and Distribution Conference and Exposition*, Vol. 3, pp. 937 – 939, ISBN 0-7803-8110-6, Sept. 2003.

Baitch A. & Barr R.A. (1985). A tapping range and voltage level analysis chart for tap changing transformers, IEEE *Transactions on Power Apparatus and Systems*, Vol. PAS-104, Issue 11, Nov 1985, pp. 3269-3277, ISSN 0018-9510.

Bhadkamkar A.; Bendre A.; Schneider R.; Kranz W. & Divan D. (2003). Application of zig-zag transformers in a three-wire three-phase dynamic sag corrector system, *Proceedings of the 34th Annual IEEE Power Electronics Specialist Conference 2003 PESC 03*, Vol. 3, pp. 1260-1265, 15-19, ISBN 0-7803-7754-0, June 2003.

Bollen M.H.J. (2000). *Understanding Power Quality Problems: Voltage Sags and Interruptions*, IEEE Press, ISBN 0-7803-4713-7, New York, USA.

Dash, P.K.; Sahoo, D.K.; Panigrahi, B.K. & Panda, G. (2001). Integrated spline wavelet and Kalman filtering approach for power quality monitoring in a power network, *Proceedings of the 4th IEEE International Conference on Power Electronics and Drive Systems*, Vol. 2, pp. 858–863, ISBN 0-7803-7233-6, Oct. 2001.

Deckmann, S.M. & Ferreira, A.A. (2002). About voltage sags and swells analysis, *Proceedings of the 10th International Conference on Harmonics and Quality of Power*, Vol. 1, pp. 144-148, ISBN 0-7803-7671-4, Oct. 2002.

Dugan R.C.; McGranaghan M.F.; Santoso S. & Beaty H.W. (2003). *Electrical Power Systems Quality*, McGraw Hill, ISBN 0-07-138622-X, USA.

Fitzer C.; Barnes M. & Green P. (2004). Voltage sag detection technique for a dynamic voltage restorer, IEEE *Transactions on Industry Applications*, Vol. 40, Issue 1, Jan-Feb 2004, pp. 203-212, ISSN 0093-9994.

Fletcher D.L. & Stadlin W.O. (1983). Transformer tap position estimation, IEEE *Transactions on Power Apparatus and Systems*, Vol. PAS-102, Issue 11, Nov 1983, pp. 3680-3686, ISSN 0018-9510.

Girgis A.A.; Chang W.B. & Makram E.B. (1991). A digital recursive measurement scheme for on-line tracking of power system harmonics, *IEEE Transactions on Power Delivery*, Vol. 6, No. 3, July 1991, pp. 1153-1160, ISSN 0885-8977.

González M.; Cárdenas V. & Álvarez R. (2006). Detection of sags, swells and interruptions using the digital RMS method and Kalman filter with fast response, *Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society IECON2006*, pp. 2249-2254, ISBN 1-4244-0391-X, Paris, France, November 2006.

González M.; Cárdenas V. & Álvarez R. (2008). A fast detection algorithm for sags, swells, and interruptions based on digital RMS calculation and Kalman filtering, *Submitted to IEEE Transactions on Aerospace and Electronic Systems*.

Hingorani N.G. & Gyugyi L. (2000). *Understanding FACTS: Concepts and Technology of Flexible AC Transmission Systems*, IEEE Press, ISBN 0-7803-3455-8, New York, USA.

IEEE Std 519-1992 (1993). *IEEE Recommended Practices and Requirements for harmonic Control in Electrical Power Systems*, IEEE inc., ISBN 1-55937-239-7, New York, USA.

IEEE Std 1100-2005 (2006). *IEEE Emerald Book, Recommended Practice for Powering and Grounding Electronic Equipment*, IEEE inc., ISBN 0-7381-4979-9, New York, USA.

IEEE Std 1159-1995 (1995). *IEEE Recommended Practice for Monitoring Electric Power Quality*, IEEE inc., ISBN 1-55937-549-3, New York, USA.

IEEE Std 1159.3-2003 (2004). *IEEE Recommended Practice for the Transfer of Power Quality Data*, IEEE inc., ISBN 0-7381-3578-X, New York, USA.

Joos, G.; Su Chen & Lopes, L. (2004). Closed-loop state variable control of dynamic voltage restorers with fast compensation characteristics, *Proceedings of the IEEE 39th IAS Annual Meeting Industry Applications Conference*, Vol. 4, pp. 2252 – 2258, ISBN 0-7803-8486-5, Oct. 2004.

Ma H. & Girgis A.A. (1996). Identification and tracking of harmonic sources in a power system using Kalman filter, *IEEE Transactions on Power Delivery*, Vol. 11, No. 3, July 1996, pp. 1659-1665, ISSN 0885-8977.

Martínez G. S. (1992). *Alimentación de equipos informáticos y otras cargas críticas (In Spanish)*, Mc-Graw-Hill, ISBN 84-7615-920-X, Spain.

Montero-Hernández O.C. & Enjeti P.N. (2005). A fast detection algorithm suitable for mitigation of numerous power quality disturbances, *IEEE Transactions on Industry Applications*, vol. 41, Issue 6, Nov.-Dec. 2005, pp. 2661-2666, ISSN 0093-9994.

Nielsen J.G.; Newman M.; Nielsen H. & Blaabjerg, F. (2004). Control and testing of a dynamic voltage restorer (DVR) at medium voltage level, *IEEE Transactions on Power Electronics*, Vol. 19, Issue 3, May 2004, pp. 806 – 813, ISSN 0885-8993.

Saleh S.A. & Rahman M.A. (2004). Wavelet-based dynamic voltage restorer for power quality improvement, *Proceedings of the IEEE 35th Annual Power Electronics Specialists Conference PESC04*, Vol. 4, pp. 3152-3156, ISBN 0-7803-8399-0, June 2004.

Sannino A.; Miller M.G. & Bollen M.H.J. (2000). Overview of Voltage Sag Mitigation, *Proceedings of the IEEE Power Engineering Society Winter Meeting*, Vol. 4, pp. 2872-2878, ISBN 0-7803-5935-6, January 2000.

Schwartzenberg, J.W.; Nwankpa, C.O.; Fischl, R. & Sundaram, A. (1994). Prediction of distribution system disturbances, *Proceedings of the IEEE 25th Annual Power Electronics Specialists Conference PESC94*, Vol. 2, pp. 1077-1082, ISBN 0-7803-1859-5, June 1994.

Vilathgamuwa D.M., Perera A.A.D. & Choi S.S. (2003). Voltage sag compensation with energy optimized dynamic voltage restorer, *IEEE Transactions on Power Delivery*, Vol. 18, July 2003, pp. 928 – 936, ISSN 0885-8977.

Wang K., Zhuo F., Li Y., Yang X. & Wang Z. (2004). Three-phase four-wire dynamic voltage restorer based on a new SVPWM algorithm, *Proceedings of the IEEE 35th Annual Power Electronics Specialists Conference PESC 04*, Vol. 5, pp. 3877 – 3882, ISBN 0-7803-8399-0, June 2004.

Wunderlin, T.; Amhof, O.; Dahler, P. & Guning, H. (1998). Power supply quality improvement with a dynamic voltage restorer (DVR), *Proceedings of the International Conference on Energy Management and Power Delivery EMPD98*, Vol. 2, pp. 518 – 525, ISBN 0-7803-4495-2, March 1998.

Yu K.K.C.; Watson N.R. & Arrillaga J. (2005). An adaptive Kalman filter for dynamic harmonic state estimation and harmonic injection tracking, *IEEE Transactions on Power Delivery*, Vol. 20, No. 2, April 2005, pp. 1577-1584, ISSN 0885-8977.

Zamora B. M. & Macho S. V. (1997). *Distorsión armónica producida por convertidores estáticos (In Spanish)*, Iberdrolla, ISBN 84-921269-1-9, Spain.

Application of Kalman Filter to Bad-Data Detection in Power System

Chien-Hung Huang, Kuang-Kong Shih,
Chien-Hsing Lee and Yaw-Juen Wang
Taiwan Power Company, National Formosa University of Science and Technology,
National Cheng Kung University& National Yunlin University of Science and Technology
Taiwan

1. Introduction

Bad-data detection in pre-estimation can help to improve state estimation [Teeuwsen & Erlich, 2006]. Since transient and abnormal conditions may occur in a power system, measurements may be polluted by bad data to cause estimated errors. Hence, it is necessary to accurately detect the bad data. To ensure the reliability of the measured data, a practical state estimator should have an ability to detect and identify the bad data as well as to eliminate their effects on the estimation [Zhang & Lo, 1991].

Generally, bad-date detection is important to guarantee the reliability of the measured data. If one or more errors occur in power system measurements, the states of the estimated system may be biased and the safety of power supply may be potentially dangerous. To avoid this situation, several bad-data detection and identification schemes have been presented. For example, WLS (weighted least squares) was proposed in 1989 [EI-Keib et al., 1989]. The weighted sum of squares of the measurement residuals was chosen as the objective function to be minimized. But, WLS-based state estimators were only developed by using a linearized measurement function [Huang & Lin, 2003] with complicated computations. Then, linear programming (LP) was proposed to improve the identification method [Peterson & Girgis, 1988]. However, the LP estimator may fail to reject the bad data and it can be attributed to the existence of leverage in the power system model [Ali Abur, 1990]. Thus, Ali Abur had proposed hypothesis testing identification (HTI) to extend the case of the LP estimator. Nevertheless, it had caused computational burdens with taking the special properties of the LP estimation equations into account. Huang [Huang & Lin, 2003] proposed a changeable weighting matrix to identify the bad data but it only can apply for static state estimations. Nevertheless, Zhang had proposed recursive measurement error estimation identification (RMEEI) for bad-data identification [Zhang et al., 1992]. State variables, residuals and their parameters can be updated after removing a measurement from the suspected data set to the remaining data set by using a set of linear recursive equations. With splitting the raw measurements into some parts, a set of residual equations used by the traditional methods can only apply to linear systems and it may result the operation of calculation burden and complexity because each part consists of some measurements [Zhang et al., 1992].

Thus, an artificial neural network (ANN) technique was proposed to overcome this problem. Moreover, data projection technique [Souza, et al,. 1998] is proposed, the normalized innovations are used as input variables to construct ANN based on the Group Method of Data Handling (GMDH) for bad data identification. Pattern analysis techniques [Alves da Silva, et al., 1992] is developed which based on a probabilistic approach with implementing ANN to correct the bad data in critical measurements. The ANN can find a nonlinear function mapping between input and output through the trained weights; meanwhile, it can be considered as an estimator to filter abnormal variations from input not the output. This means noise or large variations can be eliminated or depressed if they occur at the ANN input. Furthermore, the ANN can process raw estimated measurements by the trained ANN to diagnose bad data with a threshold value [Salehfar & Zhao, 1995] or gap-statistic-algorithm [Teeuwsen & Erlich, 2006; Huang & Lin, 2004]. Its advantage is that most measured errors can be identified and responded quickly but the pre-requisite is under a well training to construct the nonlinear function between input and output. In literature [Teeuwsen & Erlich, 2006; Salehfar & Zhao, 1995; Huang & Lin, 2002], the back-propagation algorithm has usually been applied to train the ANN. However, it has several drawbacks as follows:

1. The complex quantities measured in a power system need to separate into two parts since the standard ANN only uses real numbers as variables. This increases the size of the neural model to result poor convergence behavior. Although Salehfar and Zhao [Salehfar & Zhao, 1995] had proposed a divided technique to solve this problem, it will increase the number of iterations in the learning stage. Thus, the divided technique is labor-intensive and inconvenient.
2. The nonlinear mapping function used in power system measurements is complex and cannot be implemented with the standard ANN. Thus, the filtering performance is degraded. This means the standard ANN may not be able to remove noise and abnormal variations well at the input.
3. The training method used in the standard ANN is based on the back-propagation algorithm. However, the back-propagation algorithm cannot validly keep immunity from noise on the training data. Thus, the nonlinear function will be affected by the training algorithm in bad mapping.
4. The learning rate is not sure to be suitably applied because of the heuristic choice. It may incur the problem of stability and suffer from much slower convergence if an improper learning rate is chosen. Moreover, the learning rate is usually adjusted at the training stage for different systems. This is inconvenient for applications.

To overcome those drawbacks, in this chapter we proposes an extended complex Kalman filter artificial neural network (ECKF-CANN). It can perform well on bad-data identification with fast computational speed in two stages. The first stage is the learning process. State variables consist of the weighting that can be learned using extended complex Kalman filter (ECKF) to achieve the purpose of adjusting the learning of artificial neural network (ANN) constantly. As the training has been finished, similarly, a polluted value with several times of the standard deviation of the measurements was added into the measurements. The second stage uses the complex ANN (CANN) with the trained weighting to estimate the measurements. The rule of bad-data decision is to minimize the square difference between the measured and estimated values.

2. Complex artificial neural network

This chapter uses the ECKF algorithm to train the weightings of the ANN since the ECKF can estimate continually to modify the weightings in order to reach the learning purpose of the ANN without the learning parameter at the learning stage. Then, the trained CANN can be used for bad data detection. Moreover, the proposed ECKF-CANN method can increase the magnitude of the squared errors to enhance the efficiencies of bad data detection as bad data have been occurred because the nonlinear mapping function of the CANN and the complex measurements are coordinated well and the ECKF has a pre-filtering characteristic. A framework of the proposed ECKF-CANN method is shown in Fig. 1. As seen in Fig. 1, this method uses two algorithms. One is the artificial neural network (ANN) with complex-type variables; the other is the extended complex Kalman filter (ECKF) to train the link weightings of the complex ANN (CANN).
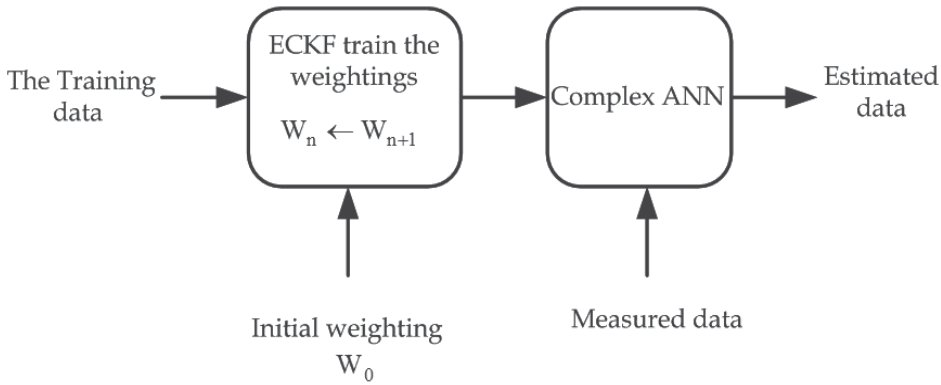
Fig. 1. Framework of the ECKF-CANN method.

The structure in the nth neuron of the L layer is shown in Fig. 2. The input complex signal can be separated into real and imaginary parts. The output is a complex-type by operating with the activation function $f(\bullet)$ to suppress the varying range of the input signal. The relation of input and output of the neuron is written to be

$$O_n^L = f(S_n^L) = f(S_{n,R}^L) + jf(S_{n,I}^L) \tag{1}$$

Note that $O_{n,R}^L$ and $O_{n,I}^L$ as shown in Fig. 1 are the real and imaginary parts of the output neuron O_n^L, respectively.
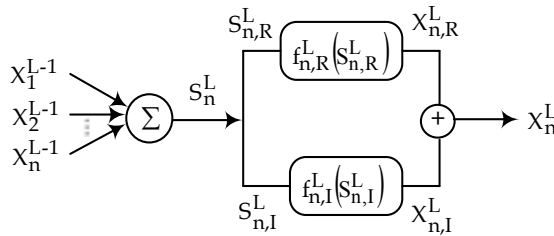
Fig. 2. Configuration of complex neurons.

Similarly, $S_{n,R}^{L}$ and $S_{n,I}^{L}$ are the real and imaginary parts of the input neuron $S_{n,}^{L}$, respectively. S_{n}^{L} is a linear combination of the output of the prior layer and is represented by the following equation.

$$S_{n}^{L} = S_{n,R}^{L} + jS_{n,I}^{L} = \sum_{m=1}^{n} O_{n}^{L-1} * W_{nm}^{L-1} \tag{2}$$

In (2), W_{nm}^{L-1} denoted the weighting of the prior layer is also a complex-type. Its operating structure is shown in Fig. 3 and the relation of the input data and the weighting is written to be

$$\begin{aligned} O * W &= (O_R + jO_I)(W_R + jW_I) \\ &= (O_R * W_R - O_I * W_I) + j(O_R * W_I + O_I * W_R) \end{aligned} \tag{3}$$
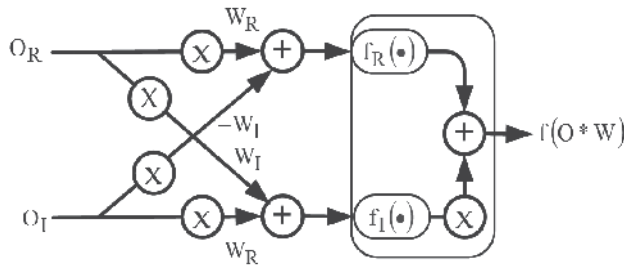
Fig. 3. Configuration of data resolution.

Substituting (3) into the activation function $f(\bullet)$, one can obtain

$$f(O * W) = f(O_R * W_R - O_I * W_I) + jf(O_R * W_I + O_R * W_I) \tag{4}$$

To satisfy the conditions of the activation function, a complex activation function is obtained by conforming to a special property which it is analytic and bounded everywhere in the complex plane [Taehwan Kim & Adali, 2000]. This paper selects the function of $\tanh(z)$ as an activation function, where z is a complex value.

3. Extended complex Kalman filter

The ECKF-CANN method is proposed to use the innovation vector to minimize the difference between input and output. The unknown link weighting w of the ECKF-CANN method [Deergha Rao, 1996; Deergha Rao et al., 2000; Taehwan Kim & Adali, 2000] from the first layer to the M layer can be considered as state variables of the ECKF for estimations as below:

$$w = [(w^1)^T, (w^2)^T, (w^3)^T, ..., (w^{M-1})^T]^T, (L \times 1) \tag{5}$$

where superscript T is the matrix transpose operation, L indicates the total number of the sum of link weighting. If N_n is the number of the neuron in the n layer, the link weighting at i^{th} node of the n layer w_i^n is written to be

$$w_i^n = [w_{i,1}^n, w_{i,2}^n, w_{i,3}^n, ..., w_{i,N_n}^n]^T, (N_n \times 1) \tag{6}$$

The link weighting w^n of the n layer is expressed to be

$$w^n = [(w_1^n)^T, (w_2^n)^T, (w_3^n)^T, ..., (w_{N_{n+1}-1}^n)^T]^T,$$

$$(N_n(N_{n+1}-1) \times 1) \tag{7}$$

and the total number of the link weighting of the CANN is given to be

$$L = \sum_{n=1}^{M-1} N_n(N_{n+1}-1). \tag{8}$$

Assuming the output $O^n(t)$ of the nodes at n^{th} layer can written to be

$$O^n(t) = [O_1^n(t), O_2^n(t), ..., O_{N_n}^n(t)]^T, (N_n \times 1) \tag{9}$$

and the desired output $d(t)$ is given to be

$$d(t) = [d_1(t), ..., d_{N_M}(t)]^T, (N_M \times 1) \tag{10}$$

As a result, the model of multilayered neural network can then be expressed by nonlinear equations as below:

$$w(t+1) = w(t) \tag{11}$$

$$d(t) = O^M(t) + v(t) \tag{12}$$

where $O^M(t)$ is the output layer of the CANN at an instant time t, M is the output of the last layer, $v(t)$ is a random noise with covariance $R_v(t)$. Thus, the learning algorithm of the ECKF-CANN is summarized to be

$$\hat{w}(t) = \hat{w}(t-1) + K(t)[d(t) - \hat{O}^M(t)], (L \times 1) \tag{13}$$

$$K(t) = P(t-1)H(t)^H[H(t)P(t-1)H(t)^H + R_v(t)]^{-1},$$

$$(L \times N_M) \tag{14}$$

$$P(t) = P(t-1) - K(t)H(t)P(t-1), (L \times L) \tag{15}$$

where K(t) is the Kalman gain, H(t) is the measurement matrix, H(t)[H] is the Hermitian matrix of H(t) called the Jacobian matrix, $\hat{w}$ is the estimated value of w, and P(t) means the expectation values of the residual of the $\hat{w}(t)$ and $\hat{w}(t-1)$. One can further write the P(t) and H(t) to be as below:

$$P(t) = E\{(\hat{w}(t) - \hat{w}(t-1))(\hat{w}(t) - \hat{w}(t-1))^H\} \tag{16}$$

$$H(t) = (\frac{\partial O^M(t)}{\partial w})_{w=\hat{w}(t-1)}$$

$$= \frac{\partial(O_R^M(t) + jO_I^M(t))}{\partial(w_R(t) + jw_I(t))}|_{w=\hat{w}(t-1)}$$

$$= \left(\frac{\partial(O_R^M(t) + jO_I^M(t))}{\partial(w_R(t) + jw_R(t))}\right)_{w=\hat{w}(t-1)} + \left(\frac{\partial(O_R^M(t) + jO_I^M(t))}{\partial(w_I(t) + jw_I(t))}\right)_{w=\hat{w}(t-1)}$$

$$= [H_1^1(t),\ldots,H_{N_2-1}^1(t),H_1^2(t),\ldots H_{N_3-1}^2(t),\ldots,H_1^{M-1}(t),\ldots H_{N_{M-1}}^{M-1}(t)],$$

$$(N_M \times L) \tag{17}$$

and

$$H_i^n(t) = \left(\frac{\partial O^M(t)}{\partial w_i^n}\right)_{w=\hat{w}(t-1)}$$

$$= \left(\frac{\partial O_R^M(t)}{\partial w_{iR}^n}\right)_{w=\hat{w}(t-1)} + j\left(\frac{\partial O_I^M(t)}{\partial w_{iR}^n}\right)_{w=\hat{w}(t-1)}$$

$$+ \left(\frac{\partial O_R^M(t)}{\partial w_{iI}^n}\right)_{w=\hat{w}(t-1)} + j\left(\frac{\partial O_I^M(t)}{\partial w_{iI}^n}\right)_{w=\hat{w}(t-1)}, \quad N_M \times N_n \tag{18}$$

$$H_i^n(t) = \Delta_{iRR}^n(t)|_{w=\hat{w}(t-1)} \hat{O}_R^n(t)^T + j\Delta_{iIR}^n(t)|_{w=\hat{w}(t-1)} \hat{O}_I^n(t)^T$$
$$+ \Delta_{iRI}^n(t)|_{w=\hat{w}(t-1)} \hat{O}_R^n(t)^T + j\Delta_{iII}^n(t)|_{w=\hat{w}(t-1)} \hat{O}_I^n(t)^T$$

$$= H_{iR}^n(t) + H_{iI}^n(t) \tag{19}$$

where

$$H_{iR}^n = \hat{\Delta}_{iRR}^n(t)|_{w=\hat{w}(t-1)} \hat{O}_R^n(t)^T + \hat{\Delta}_{iRI}^n(t)|_{w=\hat{w}(t-1)} \hat{O}_R^n(t)^T \tag{20}$$

$$H_{iI}^n = j\hat{\Delta}_{iIR}^n(t)|_{w=\hat{w}(t-1)} \hat{O}_I^n(t)^T + j\hat{\Delta}_{iII}^n(t)|_{w=\hat{w}(t-1)} \hat{O}_I^n(t)^T \tag{21}$$

$$\Delta_{iRR}^n(t) = (1 - (\hat{O}_{iR}^{n+1}(t))^2) \sum_{l=1}^{N_{n+2}-1} \hat{w}_{l,iR}^{n+1}(t-1)\hat{\Delta}_{lRR}^{n+1}(t) \tag{22}$$

$$\Delta_{iIR}^n(t) = (1 - (\hat{O}_{iI}^{n+1}(t))^2) \sum_{l=1}^{N_{n+2}-1} \hat{w}_{l,iI}^{n+1}(t-1)\hat{\Delta}_{lIR}^{n+1}(t) \tag{23}$$

$$\Delta_{iRI}^{n}(t) = (1 - (\hat{O}_{iR}^{n+1}(t))^2) \sum_{l=1}^{N_{n+2}-1} \hat{w}_{l,iR}^{n+1}(t-1)\hat{\Delta}_{lRI}^{n+1}(t) \tag{24}$$

$$\Delta_{iII}^{n}(t) = (1 - (\hat{O}_{iI}^{n+1}(t))^2) \sum_{l=1}^{N_{n+2}-1} \hat{w}_{l,iI}^{n+1}(t-1)\hat{\Delta}_{lII}^{n+1}(t) \tag{25}$$

Note that the noise of the training data can be depressed since the ECKF used to train the weightings of the CANN does not consider the learning rate with heuristics.

4. Bad data detection

For bad-data detection, complex-type data obtained from power flow calculations are used in this chapter. The data learned by the ECKF will be used to train the variety of the weightings and the architecture of Complex ANN filter is applied to estimate the polluted measurements.

Since the difference between the measured value X_i and the estimated value O_i at a particular measurement point is larger than a pre-specified detection threshold in the ECKF-CANN, the decision rule based on CANN is given by

$$(X_i - O_i)^2 > r_i^2 \ , \ i = 1,...,n \tag{26}$$

where the parameters i and r represent the measurement index and the threshold, respectively. An appropriate threshold is important for bad-data detection. Many trials have to be made in order to determine the best value. Generally, the square of 10 times standard deviation is chosen for each measurement index [Salehfar & Zhao, 1995]. Bad data can be flagged as the square of the residual between the measured and estimated values is larger than the corresponding threshold. Thus, the bad data can be detected. The procedure of bad-data detection is described as follows:

Step 1.   Inputting normal measurement from a telemeter instrument at a control point such as voltages or power flows in a power system.

Step 2.   Performing the learning phase of ECKF to estimate the link weighting.

$$S_n^{(L)} = \sum_{m=0}^{N_{L-1}} W_{nm}^{(L)} O_m^{(L-1)} \tag{27}$$

Step 3.   Completing the learning as the residual is smaller than the accepted range during constant training of ECKF by the past historical data.

$$E = \sum_{n=1}^{N} ((D_{nR} - O_{nR}) + (D_{nI} - O_{nI}))^2 \tag{28}$$

where D_{nR} and D_{nI} are the real and imaginary parts of the designed value, respectively. O_{nR} and O_{nI} are also the real and imaginary parts of the output of ECKF, respectively.

Step 4.   Inputting the polluted measurement and executing the CANN algorithm.

Step 5. Determining bad data by squaring the difference between the estimated and measured values with the decision rule as shown in (26).

Step 6. Using the estimator to directly estimate with no bad-data existing. If bad data have occurred, the original measured value can be replaced by the estimated value and state estimation can be executed again.

The whole process of estimations as mentioned above may continue until the measurement index is beyond the time point obtained from the raw measurement.

5. Simulation

Two test systems including a 6-bus and the IEEE 30-bus power systems are used in this study. ANN configurations include input, hidden and output layers. The 6-bus system as shown in Fig. 4 consists of 6-bus voltage magnitude $|V_i|$, 3 pairs of active and reactive generations, 3 pairs of active and reactive loads Pi, Qi, and 22 pairs of active and reactive line flows Pi-j, Pj-i, Qi-j and Qj-i. Total of 62 data are measured in this system. Since the active and reactive power flows can be represented with a complex-type, the total measured data is then reduced to 34. As for the 30-buses system, it consists of 24 load buses, 5 generator buses and 1 reference (swing) bus as shown in Fig. 5. As a result, total of 142 data are needed to measure with the complex-type representation.
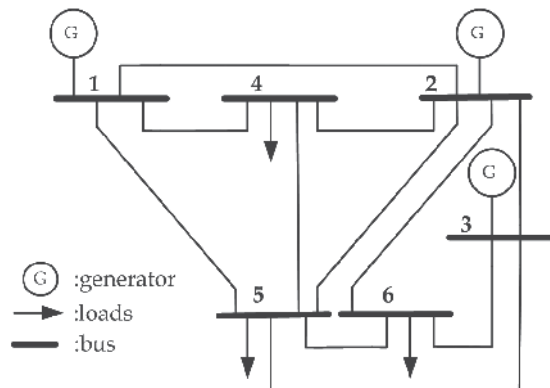
Fig. 4. Measurement configuration of 6-bus system.

Three methods will be used to detect the bad data, including ECKF-CANN, real back-propagation artificial neural network (RBP-ANN) and complex back-propagation artificial neural network (CBP-ANN). Moreover, abilities on convergences and noises rejection of the three methods are performed to assess their efficiency on bad-data detection. Comparison of the convergent behavior for detecting the 6-bus system using the three methods is shown in Fig. 6. As seen from Fig. 6, the squared error is 0.60726 at the 2nd number using the ECKF-CANN. However, the squared errors reach 0.94414 and 0.93272 at the 70th and 40th training number for the RBP-ANN and CBP-ANN, respectively.

To avoid interfering bad-data detection, the ability of noises injection is tested by applying above three methods and the results are shown in Fig. 7. As seen from Fig. 7 the squared error of the RBP-ANN reaches 0.45 at the noise of 18 dB. However, the squared errors of the CBP-ANN and ECKF-CANN reach only 0.0014461 and 0.00016034 at the noise of 20dB, respectively. Thus, performance on noise injection using the ECKF-CANN is the best among the three methods.
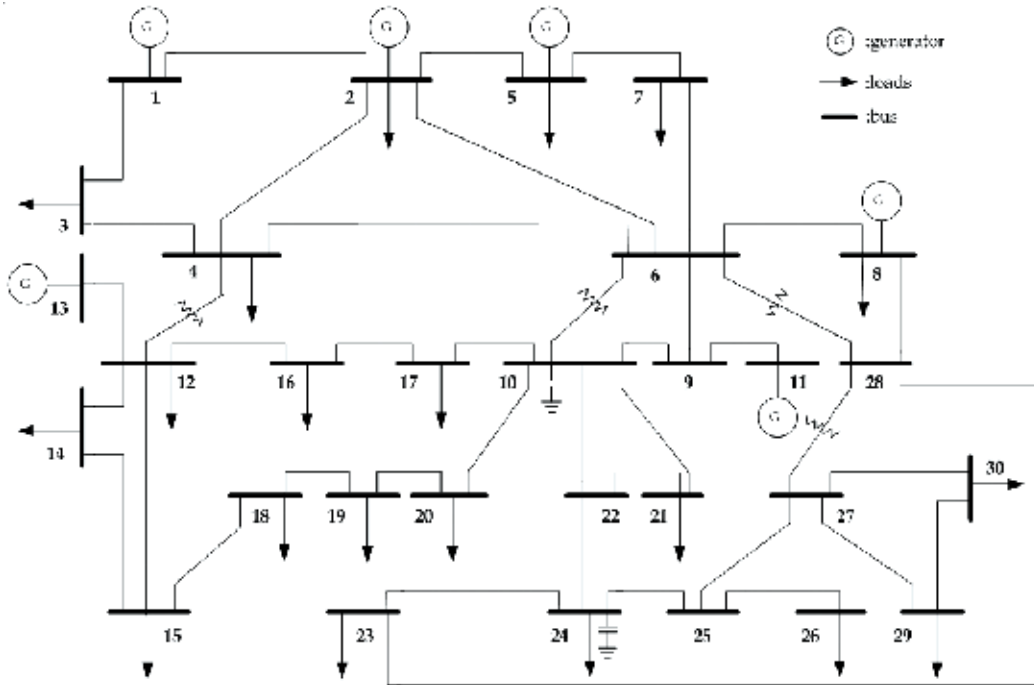
Fig. 5. Measurement configuration of IEEE 30-bus system.
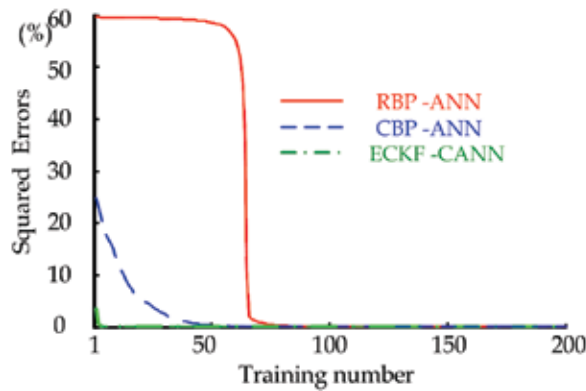

Fig. 6. Comparison of the convergent behavior for three methods.

For the 6-bus test system, the data of 150 time points can be obtained from the original power flow calculation and the first 100 time points are used as the training data of the neural network. For convenient observations, the last 20 time points of the data will be used for bad-data detection. The standard deviations of 0.01 and 0.02 for the bus voltages and the rest of measured data at the last 20 time points will be used for evaluating bad-data detection of ANN during the time duration of the last 20 time points. Similarly, the data measured at the power system of IEEE standard 30-buses will also use the last 20 time points.

Normally, the standard deviation obtained from the measured values at the first 100 time points of the data is used to generate the bad data. However, 20 to 100 times of the standard

deviation was used as the error to add into the measurement in order to generate the bad data [Salehfar & Zhao, 1995]. However, this paper uses 20 times of the standard deviation of the measured values to pollute the measurements. Three of the polluted data will be used for bad-data detection and they are described as below.
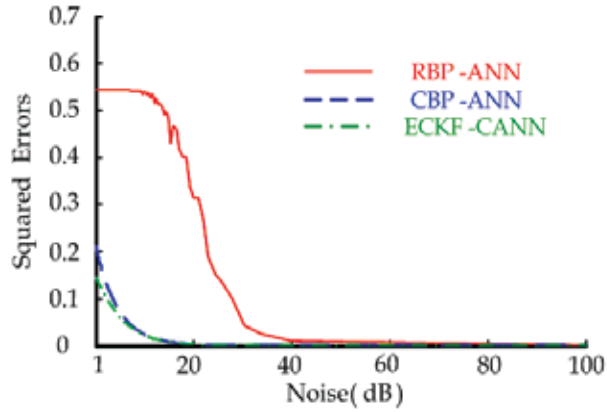

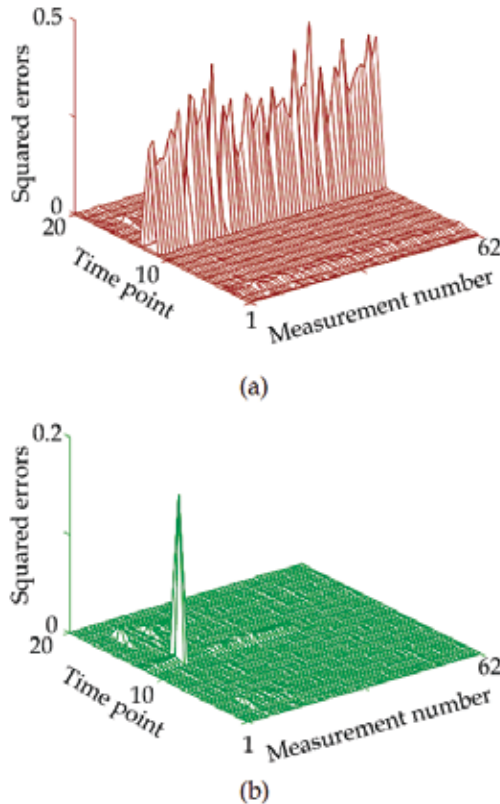Fig. 7. Comparison of the capacity of noise rejection for three methods.

(a)

(b)

Fig. 8. Comparison of the squared errors for bad data detection in P_4 using two different methods on a 6-bus system, (a) RBP-ANN, (b) ECKF-CANN.

Case 1: Single bad data
Situation A: 6-bus power system
For the 6-bus system shown in Fig. 4, a measurement at bus 4 (i.e., P_4) numbered as 10 is assumed to have a variation of 20 times standard deviation of the measurement at 12th time point. The squared errors between the measured and estimated values are shown in Fig. 8. As seen from Fig. 8(b), the ECKF-CANN method can effectively detect the bad data of power signals since it only has a pillar spiking out at 12th time point for the measurement number 10. However, the squared errors for each measurement at 12th time point using the RBP-ANN method as seen from Fig. 8(a) are all beyond the threshold value. This means the RBP-ANN method cannot detect the bad data effectively in this case.

Moreover, the estimated measurement at 12th time point using the RBP-ANN and ECKF-CANN methods as estimators are shown in Fig. 9. As seen from Fig. 9(a), the gap between the measured and estimated values at each measurement number is large except the measurement number 10. For example, the estimated and measured values at the measurement number 10 are -0.0464 and -0.144, respectively. Thus, the gap is only 0.0976 with its square error of 0.00953 which is smaller than the gaps of other measurements. Thus, those gaps result in the squared error of some measurements using the RBP-ANN method are beyond the threshold value except the measurement number 10. As seen from Fig. 9(b), the gap between measured and estimated values is large only as the measurement number
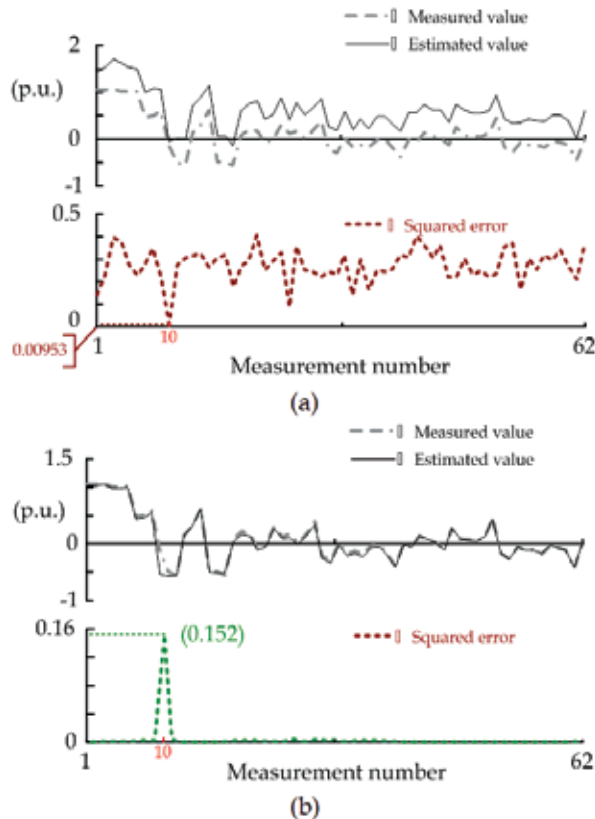

Fig. 9. Estimated results of the estimator at 12th time point using different methods on a 6-bus system, (a) RBP-ANN, (b) ECKF-CANN.

of bad data has occurred. For the measurement number 10, the estimated and measured values are -0.5335 and -0.144, respectively. Thus, the gap will reach 0.3894 with its square error of 0.152. This means the proposed ECKF-CANN method is surely more effective for single bad data detection than the Real-ANN method.

In addition, this chapter uses robust statistics [Mili ,et al,.1996; Pires ,et al,.1999] to obtain robust distances of the measured data for bad data detection under the same condition mentioned above. This method uses a chi-square distribution $(\chi^2_{v,0.025})^{1/2}$ with v degrees of freedom (i.e., v = n-1, where n is the number of the bus) as a cutoff value to flag the bad data. The detected result of the measurement at 12th time point for the measurement number 10 based on robust statistics is shown in Fig. 10. As can be seen from Fig. 10, the magnitude of the robust distance at the measurement number 10 is about 29.4077 and is greater than the cutoff value 3.582. Thus, the bad data in P_4 can be detected.
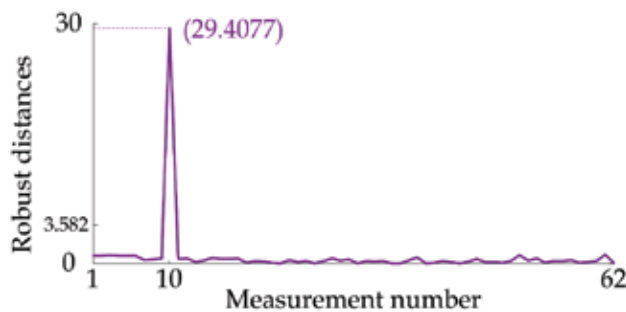

Fig. 10. Estimated results at 12th time point using robust statistics on a 6-bus system.

Situation B: IEEE 30-bus power system

This case assumes the bad data occurs at 12th time point of the P_{14} (i.e., P_{14} is the real power at bus 14) for the measurement number 12. The square errors with only the front 50 measurements obtained by the two methods are shown in Fig. 11. As seen from Fig. 11(b), the squared error reaches 0.406 at 12th time point which is largely greater than the squared errors of other measurements. This means the bad data occurred in the P_{14} can easily be detected. Unlike Fig. 11(b), there are many squared errors obtained by the RBP-ANN method at some measurement numbers as shown in Fig. 11(a). For example, two pillars are spiking out at the measurement number 13 for different time point and the squared error even reaches 0.0304. Nevertheless, the squared error is only 0.00018 at the measurement number 12 at 12th time point. This situation will result a difficulty on detecting the bad data if the threshold value is chosen.

In addition, a bad-data is assumed to occur at the 12th time point of the P_{1-3} (i.e., P_{1-3} is the real power of the transmission line flow from bus 1 to bus 3) for the measurement number 10. The threshold used to identify the bad data is predetermined to be 0.0009874 and the standard deviation is computed to be 0.0031423. The test result using the ECKF-CANN method is shown in Fig. 12. As seen from Fig. 12, the squared error is near 0.02 at the 12th time point of the first area and it is greater than the threshold. Thus, it can be used to detect the bad data occurred in the P_{1-3}.
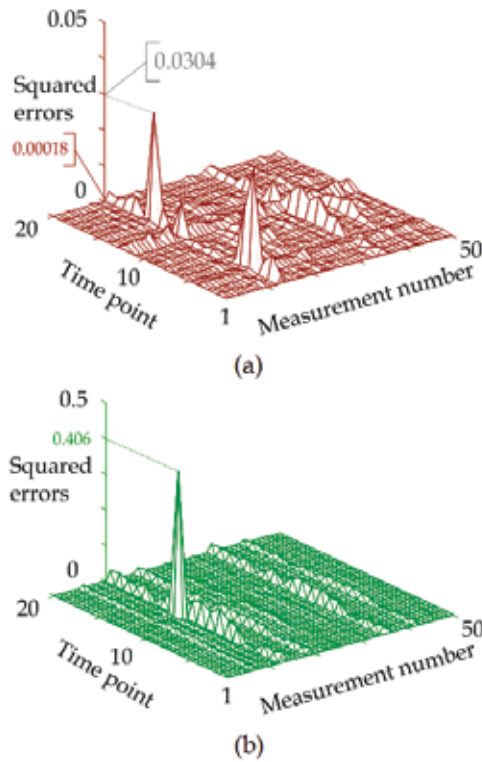
(a)

(b)

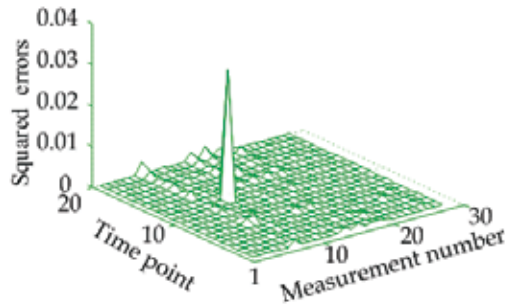Fig. 11. Squared errors of P_{14} for the IEEE 30-bus system, (a) RBP-ANN, (b) ECKF-CANN.

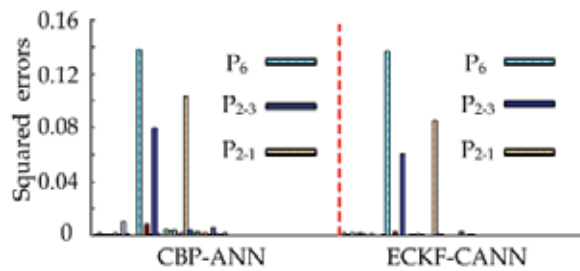Fig. 12. Squared errors of P_{1-3} for the 30-bus system using the ECKF-CANN.

Fig. 13. Comparison of the squared error for the 6-bus system using the CPB-ANN and ECKF-CANN methods.

Case 2: Multiple bad data
Situation A: 6-bus power system
Bad data are assumed to occur in the P_6, P_{2-3} and P_{2-1} of the measurement value at the 8th time point. The numbers 12, 22 and 30 are real-type measurements and the numbers 12, 16 and 24 are complex-type measurements. The threshold used to identify the bad data is predetermined to be 0.013938 and the standard deviation is computed to be 0.011806 in the P_{2-3} measurement value. Comparison of the results using the ECKF-CANN and CBP-ANN methods is shown in Fig. 13. As seen from Fig. 13, the squared errors in the P_6, P_{2-3} and P_{2-1} of both methods are all greater than the threshold. However, the results using the CBP-ANN method for detecting other measurements are beyond the threshold. Thus, the detected results were interfered. As seen from the right side in Fig. 13, the squared error of other measurements is smaller than the threshold. This means the ECKF-CANN method is more useful for bad-data detection in power signals.

Similarly, detection of the imaginary part of complex state variable with multiple bad data is tested. The bad data are assumed to occur in the Q_6, Q_{1-4} and Q_{4-1} of the measurement value at the 12th time point. As seen from Table 1, the squared errors for detecting Q_6, Q_{1-4} and Q_{4-1} are all beyond the threshold value. However, the squared errors in the Q_{1-4} and Q_{4-1} using the ECKF-CANN method are greater than those of using the CBP-ANN method. As a result, bad-data detection using the ECKF-CANN method is better than the CBP-ANN method since it has higher squared error.

Methods	Q_6	Q_{1-4}	Q_{4-1}
CBP-ANN	0.99662	0.035799	0.027097
ECKF-CANN	0.60883	0.081422	0.069158

Table 1. The squared error of Q_6, Q_{1-4} and Q_{4-1} for the 6-bus system.

Situation B: IEEE 30-bus power system
Bad data are assumed to occur in the measurements of P_2, P_{5-7} and P_{7-6} at the 10th time point of the 2nd area. The numbers 6, 19 and 31 are real-type measurements and the numbers 6, 14 and 26 are complex-type measurements. The test results are shown in Fig. 14. As seen from Fig. 14, the RBP-ANN method cannot effectively detect the bad data because its squared errors are all larger than the threshold value of each measurement. However, the CBP-ANN and ECKF-CANN methods can detect the bad data in the P_2, P_{5-7} and P_{7-6} at the 10th time point since their squared errors are all beyond the threshold value, as seen from Fig. 14(a) and 14(c).

Case 3: Interacting bad data
Situation A: 6-bus power system
Bad data are assumed to occur in the P_1 and P_4 with excessive errors and in the P_{1-2} and P_{3-2} with reverse signs of measurement values at the 10th time point. The numbers 7, 10, 19 and 34 are real-type measurements and the numbers 7, 10, 13 and 28 are complex-type measurements. Comparison of the results using the CBP-ANN and ECKF-CANN methods is summarized in Table 2. As seen from Table 2, the squared error in the P_{1-2} at the 10th time point using the ECKF-CANN method is greater than that of using the CBP-ANN method. Moreover the square error using the ECKF-CANN method is increased to raise the rate of distinguishing the bad data. This means the ECKF-CANN method is better for detecting the combination of different types of bad data than the CBP-ANN method.
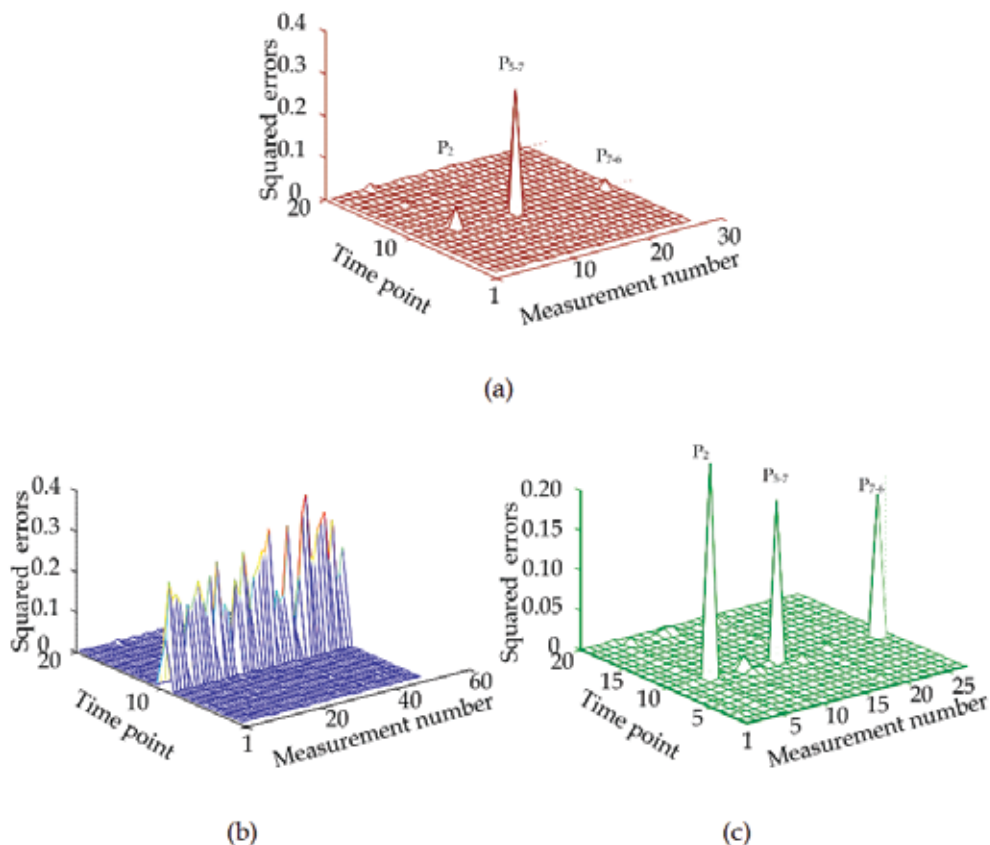
(a)

(b)

(c)

Fig. 14. Comparison of the squared errors for bad-data detection in P_2, P_{5-7} and P_{7-6} using three different methods for the 30-bus system, (a) CBP-ANN, (b) RBP-ANN, (c) ECKF-CANN.

Methods	Measured value	Estimated value	Squared error
CBP-ANN	-0.051971	0.056848	0.011842
ECKF-CANN	-0.051971	0.076853	0.016596

Note: The threshold is 0.014329 and the standard deviation is 0.01197.

Table 2. Results for detecting different types of bad data occurring in P_{1-2}.

Situation B: IEEE 30-bus power system
Bad data are assumed to occur in the P_{21} and P_{9-10} with an excessive error and in the P_{22-21} and P_{28-8} with a reverse sign of the measurement values at the 10th time point of the 4th area. The results are shown in Fig. 15. As seen from Fig.15, the ECKF-CANN method can detect the bad data in the measurements of P_{21}, P_{9-10}, P_{22-21} and P_{28-8} obviously. But, the squared errors of each measurements of the RBP-ANN at the 10th time point are all beyond the threshold value. Thus, the RBP-ANN cannot judge whether there are bad data polluted in the measurement.
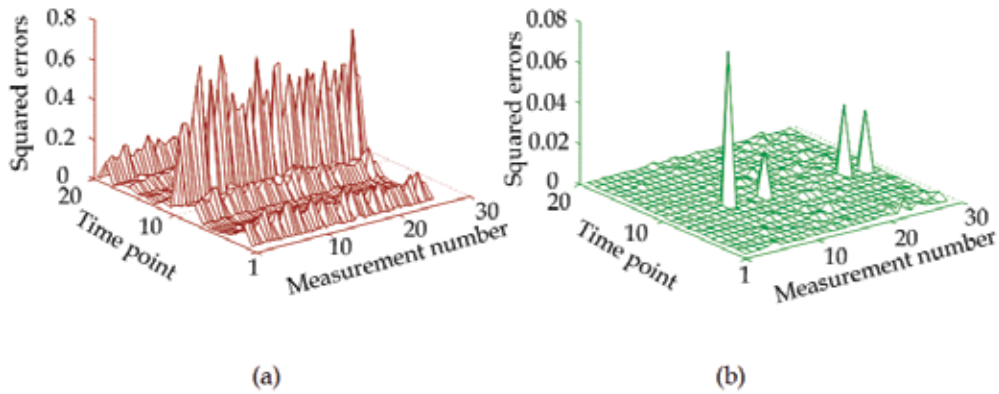
Fig. 15. Comparison of the squared errors for bad-data detection in P_{21}, P_{9-10}, P_{22-21} and P_{28-8} using three different methods for the 30-bus system, (a) RBP-ANN, (b) ECKF-CANN.

6. Conclusion

The method with the ECKF learning algorithm based CANN has been developed in this paper to identify the bad-data occurred in a power system. Complex state variables were applied as a link weighting. The proposed method not only can largely reduce node numbers of neurons, but also can search out the suitable and available training variables. Moreover, the ECKF-CANN method converges faster than the traditional algorithms and its capacity of noise rejection is better than the traditional algorithms.

7. References

Teeuwsen, S. P. & Erlich, I. (2006).Neural network based multi-dimensional feature forecasting for bad data detection and feature restoration in power system, *Proceedings of* the 2006 *IEEE General Meeting on Power Engineering Society*, pp. 1-6, ISBN:1-4244-0493-2, June,2006

Zhang, B. M. & Lo, K. L. (1991). A recursive measurement error estimation identification method for bad data analysis in power system state estimation, *IEEE Transactions on power systems*, vol. 6, no. 1, (February 1991)pp. 191-198, ISSN:1558-0679

EI-Keib, A. A.; EI-Sayed, S. M. L. & Brown, H. E. (1989). A model decoupled state estimator with effective bad data identification, *Proceedings of the 1989 IEEE Proceedings Conference on Energy and Information Technologies in the Southeast*, vol. 2, pp. 826-831, Columbia, SC, April 1989

Huang, S. -J. & Lin, J. -M. (2003). A changeable weighting matrix applied for state estimation of electric power, *Proceedings of PEDS the Fifth International Conference on Power Electronics and Drive Systems*, vol. 2, no. 17-20, pp. 873-876, ISBN: 0-7803-7885-7, November 2003

Peterson ,W. L. & Girgis, A. A. (1988). Multiple bad data detection in power system state estimation using linear programming, *Proceedings of the Twentieth Southeastern Symposium on System Theory*, pp. 405-409, ISBN: 0-8186-0847-1, Charlotte, NC, March 1988

Abur, Ali (1990). A bad data identification method for linear programming state estimation, *IEEE Transactions on Power Systems*, vol. 5, no. 3, (August 1990) pp. 894-901, , ISSN: 1558-0679

Zhang, B. M. ;Wang, S. Y. & Xiang, N. D. (1992). A linear recursive bad data identification method with real-time application to power system state estimation, *IEEE Transactions on Power Systems*, vol. 7, no. 3, (August 1992)pp. 1378-1385, ISSN: 1558-0679

Huang, S. -J. & Lin, J. -M. (2002). Enhancement of power system data debugging using GSA-based data mining technique, *IEEE Transactions on power systems*, vol. 17, no. 4, (November 2002)pp. 1022-1029, ISSN: 1558-0679

Huang S. -J. & Lin, J. -M. (2004). Enhancement of anomalous data mining in power system predicting-aided state estimation, *IEEE Transactions on power systems*, vol. 19, no. 1, , (February 2004) pp. 610-619, ISSN: 1558-0679

Souza, J. C. S.; Leite da Silva A. M. & Alves da Silva, A. P. (1998). Online topology determination and bad data suppression in power system operation using artificial neural networks," *IEEE Transactions on Power Systems*, vol. 13, no. 3, (August 1998) pp. 796-803, ISSN: 1558-0679

Alves da Silva ,A. P.; Quintana , V. H. & Pang, G. K. H. (1992). Associative memory models for data processing, *International Journal of Electrical Power & Energy Systems*, vol. 14, no. 1, (February 1992) pp. 23-32, SSDI: 0142-0615

Deergha Rao, K. (1996). Narrowband direction finding using complex EKF trained multilayered neural Networks, *Proceedings of the 1996 IEEE 3rd International Conference on Signal Processing,* vol. 2, pp. 1377-1380, ISBN: 0-7803-2912-0, Beijing, October 1996

Deergha Rao K.; Swamy,M. N. S. & Plotkin, E. I. (2000). Complex EKF neural network for adaptive equalization, *Proceedings of the 2000 IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 349-352, ISBN: 0-7803-5482-6, Geneva, May 2000

Taehwan Kim & Adali, T. (2000). Fully complex backpropagation for constant envelope signal processing, *Proceedings of the 2000 IEEE Signal Processing Society Workshop*, vol.1, pp. 231-240, ISBN: 0-7803-6278-0, Sydney NSW, December 2000

Salehfar, H. & Zhao, R. (1995). A neural network preestimation filter for bad-data detection and identification in power system state estimation, *Electric Power System Research*, vol. 34, no. 2, (February 1995) pp.127-134, SSDI: 0378-7796

Mili, L.; Cheniae, M. G.; Vichare, N. S. & Rousseeuw, P. J. (1996). Robust state estimation based on projections statistics, *IEEE Transactions on Power Systems*, vol. 11, no. 2, (May 1996)pp. 1118-1127, ISSN: 1558-0679

Pires, R. C.; Costa, A. J. A. S. & Mili, L.(1999).Iteratively reweighted least-squares state estimation through givens rotations, *IEEE Transactions on Power Systems*, vol. 14, no. 4, (November 1999)pp. 1499-1507, ISSN: 1558-0679

Extended Kalman Filter Based Speed Sensorless PMSM Control with Load Reconstruction

Dariusz Janiszewski
Poznan University of Technology
Poland

1. Introduction

There is increasing demand for dynamical systems to become more realizable and more cost-effective. These requirements extend new method of control and operation. In the robotic world important is rapidity and precision as well. These two criteria can be connected in low-cost sensorless robot arm system.

Permanent Magnet Synchronous Motors (PMSM) are used in the most modern motion control application. The PMSM products have been a new generation of electric motor of low energy, environmental protection, high efficiency that can be widely applied in many fields. The motor described simplicity of construction, high torque and small moment of inertia due motor size. One weakness can be noticed rotor position is necessary to full performance control. The Field Oriented Control (Vas, 1999) strategy permits one to fast response to load and speed changes. The purpose of this chapter is to obtain a fully PMSM drive control algorithm used for robot arm drive with load torque recognition without using any mechanical sensor. A few steps shows how to use an Extended Kalman Filter (EKF) instead of sensors for mechanical quantities.

Standard approaches defined state vector of observer restricted to sensorless control for motor only, like speed and position (Bolognani et al., 2003; Dhaouadi et al., 1991). There are often situation in which we want to obtain other types of state estimates, which can be helpful to whole robot system controllers like load torque (Janiszewski, 2006; Terorde & Belmans, 2002; Zhu et al., 2000;). In the most practical tasks information about forces acted in particular arm axis should be necessary to know. The unavoidable environment, robot modelling errors and uncertainties may cause rising of contact forces ultimately leading to the unstable behaviour during the interaction. Managing the interaction of the robot with the environment can be motion strategy.

The Kalman filter (Kalman, 1960; Gelb, 1974; Grewal & Andrews, 2001) is often applied during dissolving state estimation of dynamical system. Extended Kalman Filter is generalized algorithm, which can be used for non-linear systems such as PMSM. The estimation is done upon undisturbed input signals from overriding controller and disturbed output signals of a real non-linear plant, which are measured.

2. Permanent magnet synchronous motor drives

2.1 Permanent magnet synchronous motor

The Permanent Magnet Synchronous Motor is a rotating electric machine where the stator is a classic three phase coils like that of an induction motor and the permanent magnets are located on the rotor surface. A PMSM provides rotation at a fixed speed in synchronization with the frequency of the power source, regardless of the fluctuation of the load or line voltage. The motor runs at a fixed speed synchronous with mains frequency, at any torque up to the motor's operating limit. The PMSM consist stator windings and rotor permanent magnets sinusoidally distributed so Field Oriented Control can be used (Vas, 1990). From a control point of view, FOC is transfer and extension of DC motor control theory into PMSM. The basic concept is control by a excitation field and armature field-current (Vas, 1990).

This type of machines are extensively used in servo drives for low power machine tool, e.g. robots, positioning devices etc. They are receiving increased attention by possibility to use in the region of larger power e.g. electricity generation. The following requirements for servo drives must be served:

- high possible power to weight ratio,
- large torque to inertia ratio – high acceleration possible,
- smooth torque in wide speed – small pulsation of speed,
- full torque at zero speed – stand still working,
- high speed operation,
- compact design and small size.

New types of Permanent Magnet materials offer the ability to design electromagnetic energy converters with complicated shapes. Permanent magnets can be sticked or inserted to the small rotor. Rare-earth magnets are mostly used in modern drive. It can be obtained air-gap flux density in range of 1 T.

PMSM are used in high-accuracy direct-drive applications mainly due to their advantages. Compared to conventional DC motors, they have no brushes or mechanical commutators, which eliminates the problems due to mechanical wear of the moving parts. In addition, the better heat dissipation characteristic and ability to operate at high speed render them superior to the PMSM drives.
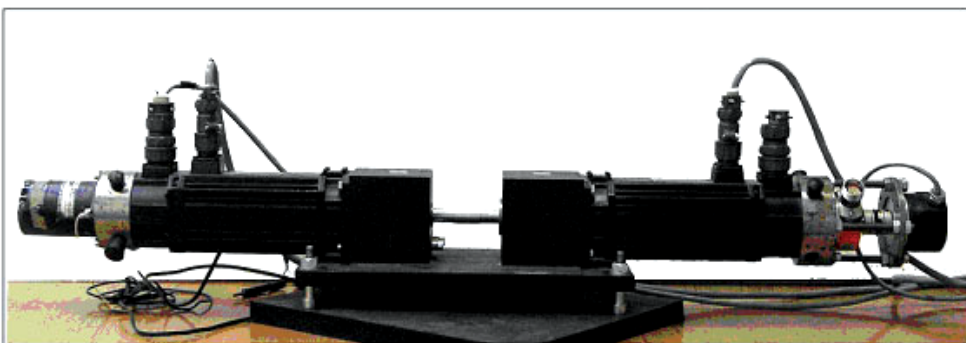

Fig. 1. Two coupled PMSM – laboratory setup.

2.2 Vector control scheme of PMSM

The control scheme of PMSM is relative simple, proposed scheme is presented on Figure 2.
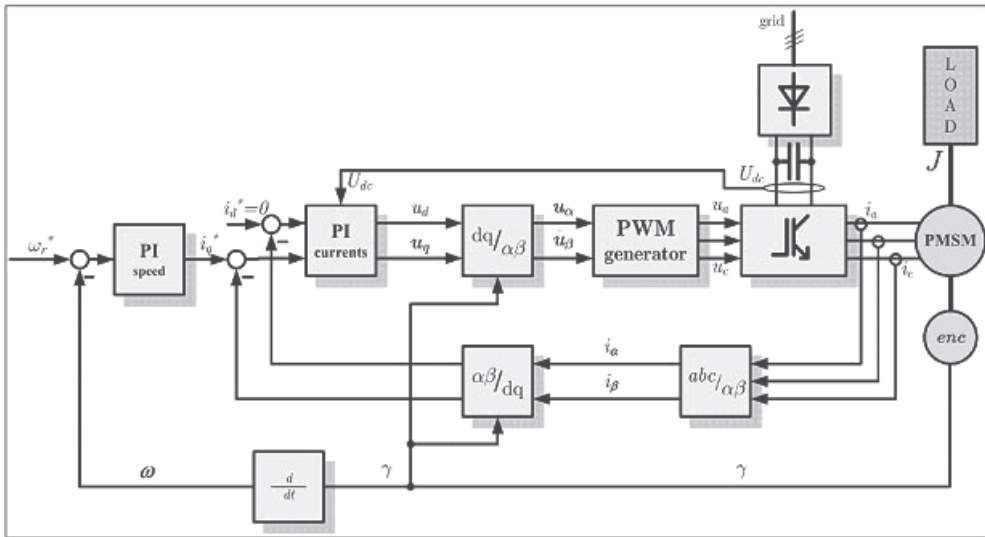
Fig. 2. Structure of control strategy of the PMSM sensor control

The field-oriented controller is based on a current-controlled voltage source inverter structure (Vas, 1990; Janiszewski 2004; Terorde, 2002). The current control loops are arranged in the 2-phase synchronously rotating rotor reference frame *d-q* aligned with rotor flux (also rotor position *g*), while the rotor position and speed detection operates in the 2-phase stationary reference frame *αβ*.
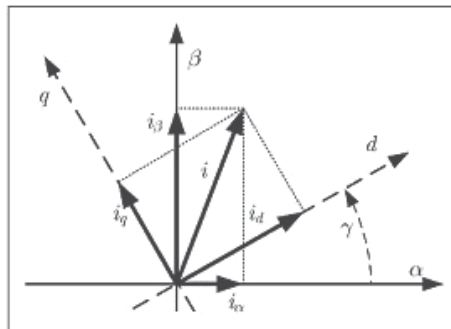

Fig. 3. *dq* and *α-β* reference frame.

The *dq*-phase arrangement PMSM machine is similar to DC machine in operation. The excitation flux (Ψf) is frozen to the direct axis of the rotor and is associated with *d* axis. Equivalent of armature current in DC machine is associated with current in *q* axis. There is also assumed that the effect of any magnetic saturation is neglected during working and thus for modelling purposes the permanent magnets can be considered as constants.

Based on above assumptions the control scheme is similar to cascaded DC motor control. The PI speed controller feeds the reference value for the torque - exciting current i_q. Due to the permanent magnets at the rotor, the reference value for the field exciting current i_d is kept to zero. Motor operating does not require the field weakening, as assumed.

Current feedback is obtained by measuring the 3-phase currents and transformations to the stator components [abc/ab] and next to the rotor components [ab/dq]. The rotor current *dq*-

axis components are needed for current regulation. Standard PI controllers with anti wind-up limitation are used for all regulators.

The output of the current controllers represents the reference voltages (u_d and u_q) in the rotor coordinates. These values are transformed [dq/ab], knowing rotor position g, into the stator coordinates ($u\alpha$ and $u\beta$) in order to calculate the desired polar voltage vector.

Using space vector Pulse Width Modulation (PWM), the polar voltage vector is converted to three phase currents by PWM inverter. Simplified electrical structure of inverter is presented on Figure 4.
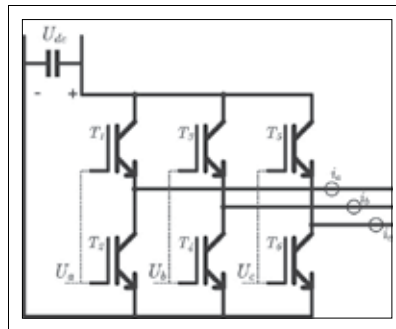

Fig. 4. Idea of 3-phase PWM inverter

The inverter (Fig 4. and [PWM] block on fig. 2) consists of three half-bridge units where the upper and lower switches are controlled complimentary. That is meaning when the upper one is turned on, the lower one must be turned off, and vice-versa. The output voltage is created by a PWM technique by modulating the high-side (T_1, T_3, T_5 on Fig. 4) and low-side (T_2, T_4, T_6) switches of the power inverter. (Vas, 1990)

The power devices for this applications are IGBT (Insulated Gates Bipolar Transistor). The main advantages of IGBT are small conducting resistance, small voltage drop and it is designed to rapidly turn on and off switching, so it is suitable using for synthesize high frequency PWM switches.

Mechanical part of the system (see Fig 1) consist only still shaft and dynamical load generator. To simplify considerations and laboratory tests to be make a assumption that shaft is extremely still, and only one parameter describable – concentrated moment of inertia J. This parameter is amount of all moments of inertia in mechanical system. Dynamical load T_{load} is done by second similar motor (right side of Fig 1) which is supplied from independent converter and it is used only for generation variable load torque.

In presented method position of rotor was measured. In fact most cases are used position encoder only, but speed is computed derivatively.

2.3 Sensorless control

The scheme of sensorless control of PMSM is very similar to sensor control (Barut, et al. 2005; Bolognani et al., 2003; Dhaouadi et al., 1991; Janiszewski, 2004, 2005). Proposed scheme is presented on Figure 5. There are one significant different, system have no any mechanical sensors. It can be possibility to replaced physical sensors by mathematical algorithm.

Sensorless control diagram consists of PI current controllers subordinated to speed controller and adequate frame coordinate transformations. There are no differences in this part. The main state of presented system is an estimator, realized by Kalman Filter theory operation.

The PI speed controller feeds current i_d* in q axis in order to keep Field Oriented Control (Vas, 1999). The demanded current is computing by using the difference between requested speed (ω_r*) and speed ($\hat{\omega}_r$) estimated by Kalman filter. Motor operating does not require the field weakening, as assumed. Therefore desired current i_d* in d axis is maintained to zero. These signals are inputs of PI current controllers, which provides desired voltages in dq reference frame. Basing on estimating shaft position $\hat{\gamma}$, voltages are converted into the stationary two axis frame (αβ) and send to control Pulse Width Modulation inverter.
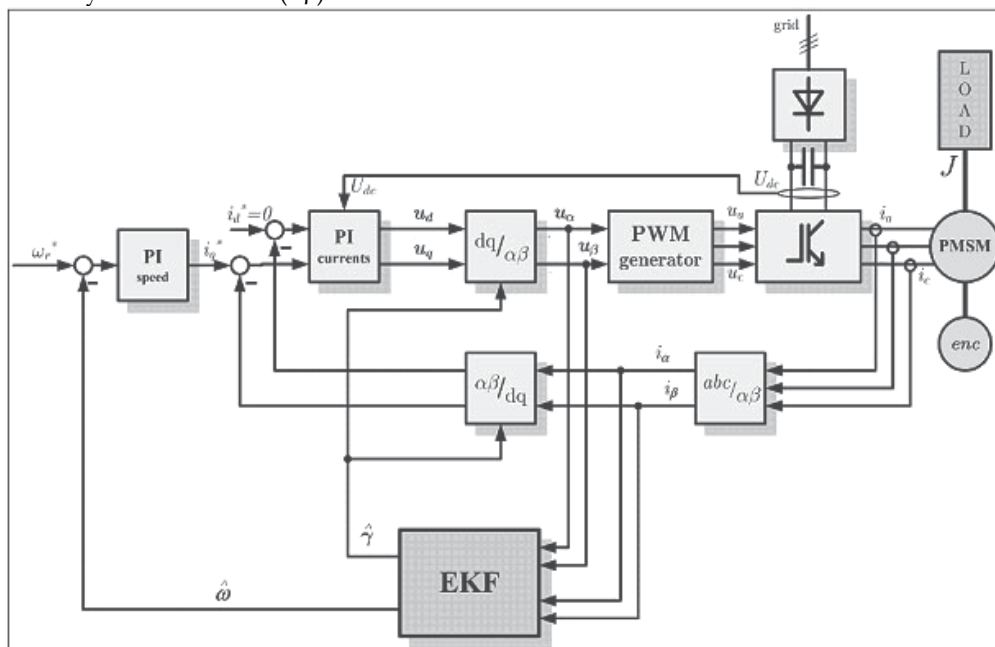

Fig. 5. Structure of control strategy of the PMSM sensorless control

3. Modelling of the system

As known, a motor model is required for the implementation of observer based on Kalman filter approach. There are no general methods that can be used to get a complete model. Each object has its own characteristics, several papers concerning modelling of PMSM (Bolognani et al., 2003; Janiszewski, 2004; Pillay & Krishnan, 1988; Vas, 1999). Some general guidelines can be given but mathematical model building often has to be combined with experiments.

The mathematical model of object is a major task during building up an observer. The proper model can simplify a solution of estimation approach. Choosing a rotor reference *dq* frame causes simplification sinusoidally distributed inductances and causes model similar to the DC machines. As is well known, the transformation of the synchronous machine equations from the *abc* phase variables to the *dq* variables forces all sinusoidally varying inductances in the *abc* frame to become constant in the *d, q*.

The following assumptions are made in the derivation:

- saturation is neglected although it can be taken into
- the back emf is sinusoidal;

- eddy currents and hysteresis losses are negligible.

The electrical properties of the motor in continuous time are completely described by two voltage network equations of the stator in rotating quadrature dq frame, as follows:

$$u_d = R_s \cdot i_d + L_d \frac{di_d}{dt} - \omega_r \cdot L_q \cdot i_q \tag{1}$$

$$u_q = R_s \cdot i_q + L_q \frac{di_q}{dt} + \omega_r \cdot L_d \cdot i_d + \omega_r \cdot \Psi_f \tag{2}$$

where: u_d, u_q are stator voltages, i_d, i_q – stator currents, R_s is stator phase resistance, L_d, L_q –dq axis stator inductances, Ψ_f – rotor flux, ω_r – mechanical speed in electrical rad/s.

The electromagnetic torque of the PMSM with surface mounted magnets and with symmetrical stator winding can be achieved very simply, and can be expressed similarly do the DC machine, as a product of i_q axis current and magnetic field. In case of interior permanent magnets the additional reluctance torque can be exploited:

$$T_{el} = \frac{3}{2} \cdot p \cdot i_q \left(\Psi_f - \left(L_q - L_d \right) i_d \right), \tag{3}$$

where: p is a number of pair of poles.

The simple mechanical system can be consider with one Inertia (J) and one acting load torque (T_{load}). Mechanical dynamics may be described as the acceleration equation:

$$\frac{d\omega_r}{dt} = \frac{1}{J} \left(T_{el} - T_{load} \right). \tag{4}$$

It is possible to define general movement equation as

$$\frac{d\omega_r}{dt} = \frac{p}{J} \cdot i_q \left(\Psi_f - \left(L_q - L_d \right) i_d \right) - \frac{T_{load}}{J}. \tag{5}$$

Many speed observers described in literature do not recognize load torque (Bolognani et al., 2003; Dhaouadi et al., 1991). It is assumed, the velocity is treated like constant in short period of time. These solutions avoid load torque which is treated like unknown disturbance.

The derivative of angular shaft position is defined by:

$$\frac{d\gamma}{dt} = p\omega_r. \tag{6}$$

The state-space representation of the model is useful for observer construction. The main problem when making a mathematical model is to find the states of the system. The states variables definitively describe storage of energy and mass in the system. Typical variables that are chosen as states are voltages and currents for electrical systems, position, speed and torque for mechanical. Basing on described balanced equations (1,2,5,6), we can write down similar to (Barut, 2004; Janiszewski, 2005, 2006; Terorde & Belmans, 2002) as the state-space vector:

$$x_k = \begin{bmatrix} i_d & i_q & \omega_r & \gamma & T_{load} \end{bmatrix} \tag{7}$$

Discrete-time of presented continuous system is described by the following state equations:

$$x_{k+1} = \mathbf{A}_k\left(x_k\right) \cdot x_k + \mathbf{B}_k\left(x_k\right) \cdot u_k + w_k \tag{8}$$

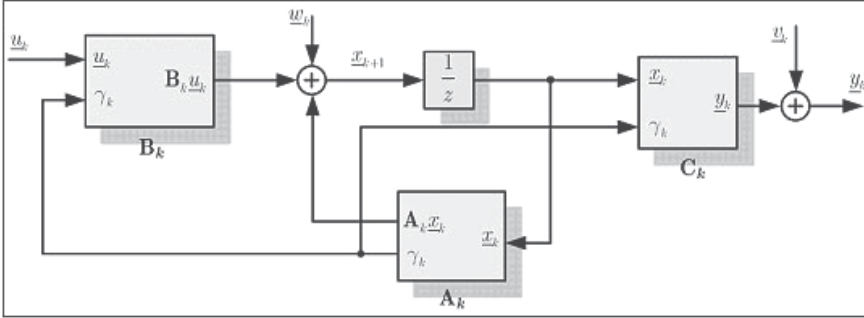$$y_k = \mathbf{C}_k\left(x_k\right) \cdot x_k + v_k. \tag{9}$$

Fig. 6. The PMSM discrete model

In this case all variables are available. The system has a electrical input – stator voltages ($\alpha\beta$ reference frame):

$$u_k = \begin{bmatrix} u_\alpha & u_\beta \end{bmatrix} \tag{10}$$

and electrical output – stator currents:

$$y_k = \begin{bmatrix} i_\alpha & i_\beta \end{bmatrix} \tag{11}$$

In the original *abc* reference frame the windings are displaced at 120 electrical degrees in space, and voltages and currents equations are dependent of each other and time. The transformation to two perpendicular phases ($\alpha\beta$) can avoid this relation. The relation between the two-axis stator currents components and the corresponding three-phase measured current can be obtained by:

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{12}$$

The next very important step is avoid the rotating vector. It can be possible to observing the current in the reference frame rotation with the same speed as the current state vector. In this consideration d-axis of the rotating frame is aligned with the rotor flux (Ψ_f). The following transformation can change reference frame from $\alpha\beta$ into *dq*:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}.$$ (13)

The inverse transformation is usually used for the calculation of the reference voltages being the input of the PWM module (fig. 2, 5):

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} u_d \\ u_q \end{bmatrix}.$$ (14)

The value of external torque T_{load} is treated as state variable (8) and part of disturbance vector w_k (9) (Barut et al., 2005; Janiszewski, 2005, 2006; Terorde & Belmans, 2002; Zhu et al., 2000). This operation allows to estimate parameters of differential equations of model. It is also assumed, that this estimated variable is treated like white noise and it is constant in small interval (sampling time T_s):

$$\frac{dT_{load}}{dt} \approx 0 .$$ (15)

It is noticed, that the state-space equations are non-linear. Some elements of the $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k$ matrices depend on an instant x_k vector (values of angular position γ, shaft velocity ω_r and current i_d). In order to build estimator basing on EKF, the non-linear model of PMSM is a priority.
There are matrices:

$$\mathbf{A}_k = \begin{bmatrix} 1 - T_s\dfrac{R_s}{L_d} & T_s\omega_r\dfrac{L_q}{L_d} & 0 & 0 & 0 \\[2ex] T_s\omega_r\dfrac{L_d}{L_q} & 1 \quad T_s\dfrac{R_s}{L_q} & T_s\dfrac{\Psi_f}{L_q} & 0 & 0 \\[2ex] 0 & T_s\dfrac{3}{2}\dfrac{p}{J}\left[\Psi_f - \left(L_q - L_d\right)i_d\right] & 1 & 0 & -T_s \\[2ex] 0 & 0 & T_s & 1 & 0 \\[2ex] 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$ (16)

$$\mathbf{B}_k = \begin{bmatrix} T_s\dfrac{1}{L_d}\cos\gamma & T_s\dfrac{1}{L_d}\sin\gamma \\[2ex] -T_s\dfrac{1}{L_q}\sin\gamma & T_s\dfrac{1}{L_q}\cos\gamma \\[2ex] 0 & 0 \\[1ex] 0 & 0 \\[1ex] 0 & 0 \end{bmatrix},$$ (17)

$$\mathbf{C}_k = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 & 0 \end{bmatrix}. \tag{18}$$

2. EKF state estimation

The Kalman filter is often applied during dissolving state estimation of dynamical system, disturbed by the known signals (Gelb, 1974, Grewal & Andrews, 2001; Kalman, 1960). Kalman filter algorithm is used for estimating the parameters of linear system, but the PMSM model is non-linear, so we can not use that filter in this case. Extended Kalman Filter is generalized algorithm, which can be used for non-linear systems (Gelb, 1974, Grewal & Andrews, 2001). The estimation is done upon undisturbed input signals (u_α, u_β for presented PMSM drive) and disturbed output signals (measured stator currents i_α, i_β) of a real non-linear plant. As assumed, state space estimation is carried out in few steps at each computation cycle.

The EKF is an optimal estimator in the least-square sense for estimating the states of dynamic non-linear system (Kalman, 1960). An linearisation is build on assuming, that the state variables (7) are constant in one step of computation. We obtain new linearised matrices $\mathbf{A}_k(\hat{\underline{x}}_k)$, $\mathbf{B}_k(\hat{\underline{x}}_k)$ and $\mathbf{C}_k(\hat{\underline{x}}_k)$ valid only in one step. After this linearisation the inner equation (8) gets a new form:

$$\frac{\partial\hat{\boldsymbol{\Phi}}}{\partial\hat{\underline{x}}} - \frac{\partial(\mathbf{A}_k(\hat{\underline{x}})\cdot\hat{\underline{x}} + \mathbf{B}_k(\hat{\underline{x}})\cdot\underline{u})}{\partial\hat{\underline{x}}} \tag{19}$$

and output equation (9) the form:

$$\frac{\partial\hat{\mathbf{h}}}{\partial\hat{\underline{x}}} = \frac{\partial(\mathbf{C}_k(\hat{\underline{x}})\cdot\hat{\underline{x}})}{\partial\hat{\underline{x}}}. \tag{20}$$

EKF algorithm consists of two main parts: measurement and time actualisation. Measurement actualisation equation:

$$\hat{\underline{x}}_{k|k} = \hat{\underline{x}}_{k|k-1} + \mathbf{K}_k\left(\underline{y}_k - \mathbf{C}_k\hat{\underline{x}}_{k|k-1}\right) \tag{21}$$

predicts $\hat{x}_{k|k}$ state for instant time t_k. The correction proceeds on previous predicted state $\hat{x}_{k|k-1}$, using filter coefficient $\mathbf{K}_k$ (gain matrix) from t_{k-1} state and measured actual output vector y_k (currents). An error covariance is done by the following recursive relation:

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\frac{\partial\mathbf{h}}{\partial\hat{\underline{x}}}\bigg|_{\hat{\underline{x}}=\hat{\underline{x}}_{k-1}}\mathbf{P}_{k|k-1}. \tag{22}$$

The object linearisation (19) is attached to this group as well.

The second step – time actualisation, is described as the prediction of previous $\hat{x}_{k|k}$ state into a new state $\hat{x}_{k+1|k}$ used (8) in the form:

$$\hat{\underline{x}}_{k+1|k} = \mathbf{A_k}\cdot\hat{\underline{x}}_{k|k} + \mathbf{B_k}\cdot\underline{u}_k. \tag{23}$$

The filter gain matrix is defined by:

$$\mathbf{K}_k = \mathbf{P}_{k\,k-1} \left.\frac{\partial \mathbf{h}^T}{\partial \hat{\underline{x}}}\right|_{\hat{x}=\hat{x}_{k-1}} \left(\left.\frac{\partial \mathbf{h}}{\partial \hat{\underline{x}}}\right|_{\hat{x}=\hat{x}_{k-1}} \mathbf{P}_{k|k-1} \left.\frac{\partial \mathbf{h}^T}{\partial \hat{\underline{x}}}\right|_{\hat{x}=\hat{x}_{k-1}} + \mathbf{R}\right)^{-1} . \tag{24}$$

The error covariance matrix is predicted by a relation:

$$\mathbf{P}_{k+1|k} = \left.\frac{\partial \hat{\Phi}}{\partial \hat{\underline{x}}}\right|_{\underline{x}=\underline{x}_{k|k}} \mathbf{P}_{k\,k} \left.\frac{\partial \hat{\Phi}^T}{\partial \hat{\underline{x}}}\right|_{\underline{x}=\underline{x}_{k|k}} + \mathbf{Q} , \tag{25}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are respectively: the covariance matrices of the system and measurement. The whole algorithm is recursive by

$$\begin{aligned}\hat{\underline{x}}_{k|k-1} &\longrightarrow \hat{\underline{x}}_{k+1|k} \\ \mathbf{P}_{k|k-1} &\longrightarrow \mathbf{P}_{k+1|k} \end{aligned} . \tag{26}$$

An open question, one critical step towards the implementation of EKF is the choice of the values of the matrices $\mathbf{Q}$ and $\mathbf{R}$ (Barut et al., 2004; Bolognani et al., 2003; Dhaouadi et al., 1991; Gelb, 1974, Grewal & Andrews, 2001; Kalman, 1960; Janiszewski, 2005, 2006; Terorde & Belmans, 2002; Zhu et al., 2000). They have to be set based on stochastic properties of the corresponding noises (Kalman, 1960). The change of covariance matrices effects both the dynamic and steady-state. The discussed matrices are simplified to diagonal form in order to eliminate state vector elements co-disturbance. Increasing particular values of $\mathbf{Q}$, it corresponds to stronger system noises and finally an instability in result of the strongest correction. Decreasing values of $\mathbf{Q}$, it corresponds to weaker correction and estimation state errors. Matrix $\mathbf{R}$ matches measurement noises, can be measured easily in advance. Measuring is generally possible because the current measurement is needed anyway while operating the filter. Some tests, sample measurements are taken in order to determine the variance of measurement error. Laboratory particular tuning EKF shows necessary to increase to large values near 10. It helps eliminate strong noises.

In case of system covariance, the calculation is less deterministic. Considering system equation and one knowing inaccuracy (current measurements) it can be possible to estimate system noises in narrow range. The first step was appoint current measurement covariance $\mathbf{Q}(1,1)$, $\mathbf{Q}(2,2)$ in range $2.3 \cdot 10^{-8}$, and next based on mathematical model of PMSM estimate other covariances. That theoretical analysis of this consideration expanded to real plant by introducing some scaling coefficients.

Basing on described rules they were chosen and adjusted experimentally:

$$\mathbf{R} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \tag{27}$$

$$\mathbf{Q} = \begin{vmatrix} 2.3 \cdot 10^{-3} & 0 & 0 & 0 & 0 \\ 0 & 2.3 \cdot 10^{-3} & 0 & 0 & 0 \\ 0 & 0 & 0.1138 & 0 & 0 \\ 0 & 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 10^{1} \end{vmatrix} \qquad (28)$$

It should be noticed, that choosing of process covariances is proportional to the square of the sample time (T_s).

The presented algorithm does not require a precise choice of the initial values for the state vector $\hat{x}_{0|0}$ and the error covariance $\mathbf{P}_{0|0}$ matrix. It was proved during the experimental investigation, that $\mathbf{P}_{0|0}$ did not have a significant influence in the behaviour of EKF. The algorithm does not need the initial rotor position (non zero part of real state values) but it is possible to start the motor from the standstill place even if initial state vector of estimator is null.

The entire control algorithm with EKF was implemented on a DSP. The implemented algorithm consist several states, one of them is EKF. The most of EKF algorithm was written in C language and it is presented in section Code 1.

3. Experimental results

3.1 Laboratory setup

For experimental verification of the proposed estimation method, a laboratory setup has been constructed (see figure 1). It consists of the surface mounted magnets synchronous motor, supplied from the three phase power IGBT inverter. This motor is coupled via stiff shaft with the second twin motor supplied from industrial controller (DIGITAX made by Control Techniques).

During mechanical construction wont to avoid any complex mechanical solution. There was choose short stiff shaft between twin trade motors. This shaft assure small additional moment of inertia, any additional rotating masses. Every motor is equipped with position sensor, an 3600 pulses per revolution optical encoder.

A practical compact Intelligent Power Module with Insulated Gate Bipolar Transistor, that aims to combine economic reasonable cost with a high level of functionality was used. Figure 4 shows a simplified block diagram of PWM converter. The IGBT, Insulated Gate Bipolar Transistor, is a switching transistor freely controlled by voltage applied to gate terminal.

The experiments presented in this section have been carried out with DSP board that was developed in Institute of Automation and Information Engineering of Poznan University of Technology. This board was specially constructed to fit the needs and flexibility mainly for research projects. The mainboard consist powerful SHARC DSP from *Analog Devices* ADSP21060, set of memories: RAM, boot EPROM, pair of serial channel and parallel interface for extension boards. The flexibility is done by changing extension boards or mainboard freely. The extension board consists additional hardware. Motion coprocessor *Analog Devices* ADMC201 is used for PWM generator handling Eight channel 12-bit simultaneously sampled AD Converter is used: 3 channel feeds by LEM transducers for *3*-phase currents (i_a, i_b, i_c) measure, one channel feed by voltage LEM transducer for DC link (U_{dc}) measure. *Alter*a FPGA Flex 6000 was used for incremental encoder counting. Precise

measurement of position is obtained by optical encoder that is used for verify the estimated position and during identification process.

```
// assumes that the system has:
//  n states
//  m inputs
//  r outputs
// and variables be defined:
// Ak n by n matrix
// Bk n by m matrix
// Ck r by n matrix
// xhat is n size vector
// yk is r size vector
// uk is m size vector

void ekf(void)
{
        yk[0] = i_alfa;                          //measurement values
        yk[1] = i_beta;
        // measurement actualisation
        Ck = ...                                 //compute new Ck matrix based on xhat
        matmul((float*)Cx,(float*)Ck,(float*)xhat);             //Cx
        matsub((float*)yCx,(float*)yk,(float*)Cx);              //(y-Cx)
        matmul((float*)KyCxhat,(float*)Kk,(float*)yCx);         //K(y-Cx)
        matadd((float*)xhat,(float*)xhat,(float*)KyCxhat);      //xhat=xhat+K(y-Cx)

        h = ...                                  //compute new h matrix based on xhat
        transpmf((float*)(ht),(float*)(h));                     //ht = h'
        matmul((float*)Pht,(float*)Pk,(float*)ht);             //Pk*h'
        matmul((float*)hPht,(float*)h,(float*)Pht);            //h*Pk*h'
        matadd((float*)hPhtR,(float*)hPht,(float*));           //(h*Pk*h'+R)
        matinvf((float*)hPhtRinv,(float*)hPhtR);               //hPhtRinv = (hPhtR)^-1
        matmul((float*)Kk,(float*)Pht,(float*)hPhtRinv);       //Kk = Pk*h'*(hPhtR)^-1

        matmul((float*)hP,(float*)h,(float*)Pk);               //h*Pk
        matmul((float*)KhP,(float*)Kk,(float*)hP);             //kk*h*Pk
        matsub((float*)Pk,(float*)Pk,(float*)KhP);             //Pk = Pk - Kk*h*Pk

        //time actualisation
        Ak = ...                                 //compute new Ak, Bk matrces based on xhat
        Bk = ...
        uk[0] = u_alfa*UDC;
        uk[1] = u_beta*UDC;
        matmul((float*)Axhat,(float*)Ak,(float*)xhat);         //Ax
        matmul((float*)Bu,(float*)Bk,(float*)uk);              //Bu
        matadd((float*)xhat,(float*)Axhat,(float*)Bu);         //(Ax+Bu)
}
```

Code 1. Main code of EKF algorithm

The System contains also one four-channel DA Converter, that was built in for auxiliary outputs, helpful in state visualisation.

The mainboard allows for high performance algorithm implementation. The architecture of board is specially optimized for motion control works in hard real-time single task system. After initial stage main loop is repeated recursively based on external low level interrupting each $T_s = 200\ \mu s$. Observer and control code has been mainly written in C language, however except low-level procedures written directly in Assembler.

3.2 Obtained results

Experiment were performed and examined with regards to the following tasks: possibility speed changing and torque acting. At the first part of investigation, it was focused on controlled system behaviour by reference speed (ω_r*) excitation without additional load torque. This type of reference signal has such significant stages for estimator behaviour like: zero signal with no initial values in estimation vector, step signal and reversal. The maximum module of reference speed is *1000 rev/min* (*1/3* of maximum speed). The second part consist in acting external load torque and observing of behaviour control system. Investigations were carried out on a drive with parameters given in the Appendix. The position error is counted as difference between estimated value and real position.

Results of working due speed changing are presented on figure 7 and zoomed parts: start on fig. 8, reverse on fig 9. For these reason the speed settling time is limited by maximum load

torque and mechanical moment of inertia (see equation 5). These figures show some kind of error that occurs when speed is suddenly changed.
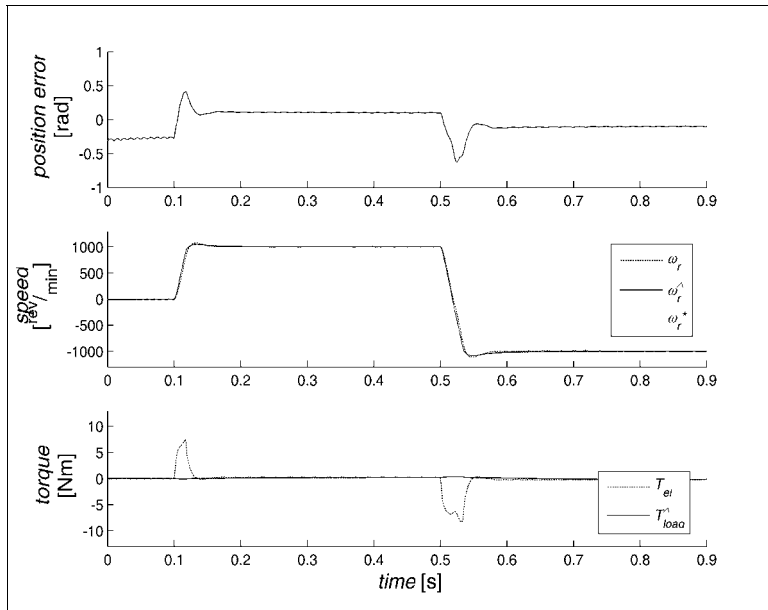

Fig. 7. Waveforms of position error, speed, torque estimations during reference speed changing

Figure 8 presents first part of the test. It should be noticed that observer did not have information about initial speed. The observer do not required any special initiation, they can find initial position after supply motor. This case happen when motor is started. Position error is cancelled and hold in small error range. As poorly shown the load torque is on level near *0,5* [Nm].

Very interesting situation is presented on figure 9 when the motor reversing. During this investigation the demanded and real speeds are near zero values. There is difficult stage of observer. When the speed is near zero, the extorter voltage values are also near zero, what causes zero activate the observer. As shown presented solution of EKF did not lost state vector, rises errors of estimation but system is still stable. After process of reversing the errors are naturally reduced.

As shown there are small estimations errors during the steady state. Some errors appears during dynamical acting. The source of errors stays in the differences between motor model and real plant, particularly at current limitation in current controller.

Figure 10 presents the main investigation, the transient response of the PMSM on step change of load (50% rated torque) at a motor speed of 1000 *rev/min*. Before load acting friction and other parasite torques appears. In steady state, the estimated load equals to electromechanical torque. Erroneous load torque dynamical reconstruction declined fast. Source of appears dynamical errors disclose in assumption that differentiation of load torque should be equal to zero (15). In practical consideration we may assumed dynamics limitation. This influence is small compared to a potential wide range of load variation.

In the author opinion the construction and physical asymmetry of PMSM rotor issues the apparent systematic position error. Verification short tests with other types of motor shows that this systematic error can be cancelled.

Due to the physical rotor asymmetry, the presented drive is also suitable for position control.
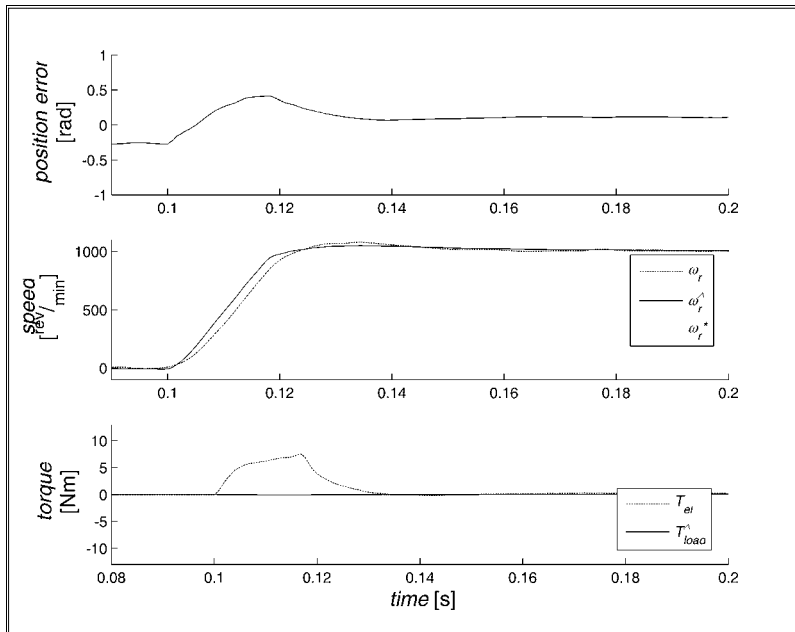

Fig. 8. Waveforms of position error, speed, torque estimations during reference speed changing – starting part
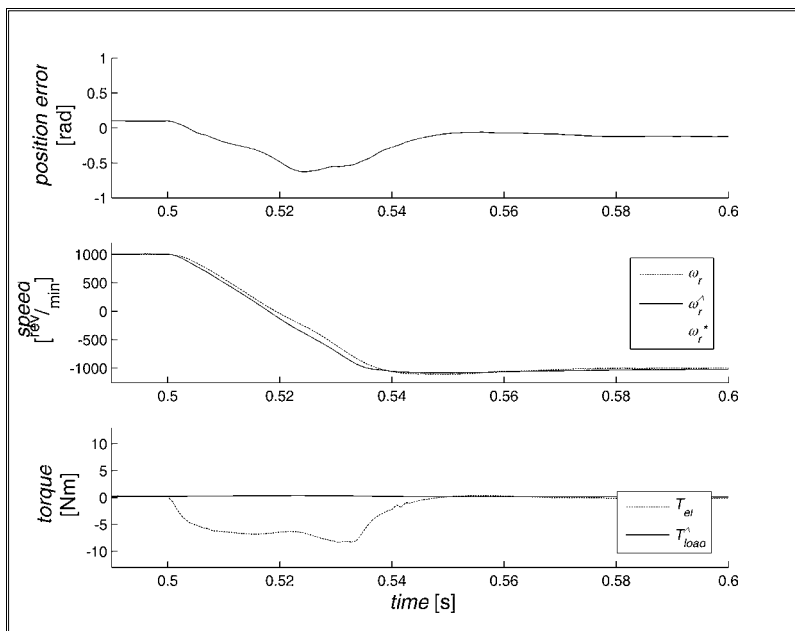

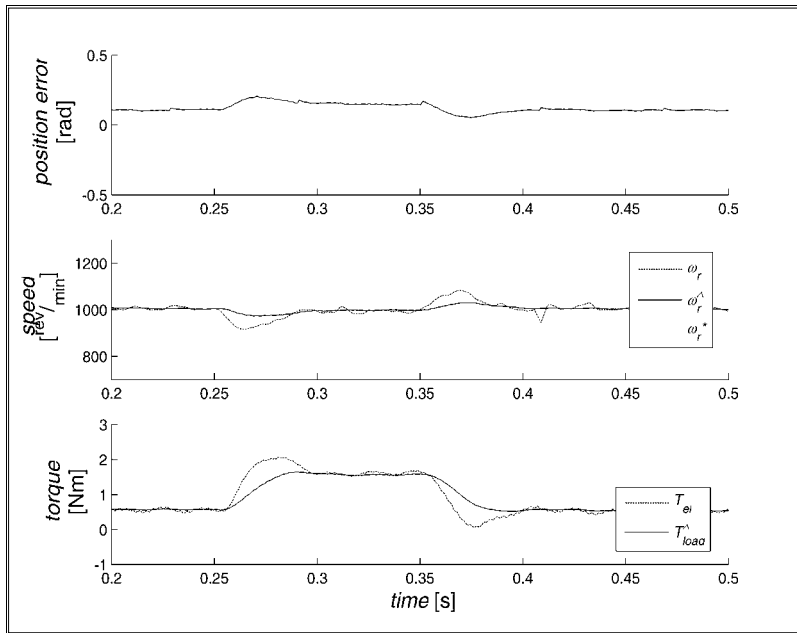Fig. 9. Waveforms of position error, speed, torque estimations during reference speed changing – reverse part

Fig. 10. Waveforms of position error, speed, torque estimations after load acting

4. Conclusions

The intent of this chapter was to show the utility of the EKF as a fundamental method for solving range of problems in sensorless control of PMSM. There was presented the design and implementation of high-performance servo motor drive with speed, position and torque estimation. The described control system is a solution without any mechanical sensor for a wide range of applications where good steady-state and dynamical properties are required. The FOC based speed controlled PMSM requires no shaft sensors for rotor position and speed. Due to rotor asymmetry, the presented drive is also suitable for position control. The presented sensorless control scheme and EKF algorithm are self-starting.

The most application required tension pick-up for torque measurement can be cheaper and rugged by load torque estimation. The described estimation of torque which was accented, can be helpful in a lot of robotic applications instead of expensive stress sensors. The consideration of the load torque as a constant term in the observation algorithm aims to capture other uncertainties besides the load torque than at rush changes. The results obtained through experiments under various challenging test demonstrate the good performance of the observer.

For the experimental study presented above, the system performance was observed to be quiet good under step and reversals desired speed and the load torque changes. The practical success in sensorless control of nonlinear object suggests that the combination of EKF and classical PI controllers can provide powerful rapidity and precision drive.

Finally, the chapter shows successful way of design of the EKF based observer of mechanical quantities for sensorless control of PMSM drive. The application of the algorithm demonstrates its potential in real-word robotic context.

5. References

Barut M., Bogosyan S. & Gokasan M. (2005), Speed sensorless direct torque control of IMs with rotor resistance estimation. *Science Direct – Energy Conversion and Management*, 2005, Vol. 46, No. 3, pp. 335-349, ISSN: 0196-8904

Bolognani S., Tubiana L., Zigliotto M. (2003), Extended Kalman filter tuning in sensorless PMSM drives. *IEEE Transactions on Industry Applications*, Nov.- Dec. 2003, Vol. 39, pp. 1741–1747, ISSN: 0093-9994

Dhaouadi R., Mohan N., Norum L. (1991), Design and implementation of an extended Kalman filter for the state estimation of a permanent magnet synchronous motor. *IEEE Transactions on Power Electronics*, Jul 1991, Vol. 6, pp. 491–497, ISSN: 0885-8993

Gelb A. (1974), *Applied Optimal Estimation*. MIT Press, Cambridge, ISBN: 978-0262570480

Grewal M., Andrews A. (2001), *Kalman Filtering Theory and Practice Using MATLAB*, Wiley-Interscience; 2 edition , 16January 2001, New York, ISBN: 978-0471392545

Kalman R.E. (1960), New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, No. 82 (Series D), pp. 374-382, ISSN: 0097-6822

Janiszewski D. (2004), Extended Kalman Filter Estimation of Mechanical State Variables of a Drive with Permanent Magnet Synchronous Motor, *Studies in Automation and Information Technology* Vol. 28/29, 2004, pp. 79-90, ISSN: 0867-3977

Janiszewski D. (2005), EKF estimation of mechanical quantities for drive with PM Synchronous Motor, *Proceedings of Power Electronics and Inteligent Control for Energy Conversation (PELINCEC)*, CDROM, Warsaw, Poland, 17-19 October 2005

Janiszewski D. (2006), Extended Kalman Filter Based Speed Sensorless PMSM Control with Load Reconstruction. *Procdings of IECON 2006 - 32nd IEEE Annual Conference on Industrial Electronic*, Nov. 2006, pp. 1465–1468, ISSN: 1553-572X, ISBN: 1-4244-0391-X

Pillay P., Krishnan R. (1988), Modeling of permanent magnet motor drives, *IEEE Transactions on Industrial Electronics*, Nov 1988, Vol. 35, pp. 537–541, ISSN: 0278-0046

Terorde G. & Belmans R. (2002), High Performance Induction Motor Drive with Speed, Flux and Torque Estimation. Procdings of *15th International Conference on Electrical Machines (ICEM)*, CDROM, Brugge, Belgium, 26-28 August 2002.

Vas P. (1990), *Vector Control of AC Machines*. Clarendon Press, Oxford, ISBN: 978-0198593706

Zhu G., Dessaint L. A., Akhrif O., Kaddouri A. (2000), Speed Tracking Control of Permanent-Magnet Synchronous Motor with State and Load Torque Observer, *IEEE Transactions on Industrial Electronics*, vol. 47, no. 2, pp. 346-355, ISSN: 0278-0046

APPENDIX

Data of investigated motor:

R_s = 5.9 [&]

$L_d = L_q$ = 32 ·10-3 [H]

J = 2 ·15 ·10-3 [kg m²] (two coupled motors)

Ψ_f = 1.56 [Vs]

$\omega_{r\,|\,nom}$ = 3000 [rpm]

$T_{load\,|\,nom}$ = 2.8 [Nm]

$T_{load\,|\,max}$ = 15 [Nm]

Application of Receding Horizon Kalman Filter to Underwater Navigation Systems

Gyungnam Jo and Hang S. Choi
Department of naval architecture and ocean engineering, Seoul National University
Korea

1. Introduction

In this chapter, the receding horizon Kalman filter is applied to underwater navigation systems. The ocean covers about two-thirds of the earth and has a great effect on human beings. However, the ocean is overlooked while we focus our attention on land and atmospheric issues; we have not been able to explore the full depths of the ocean, its abundant livings and non-living resources. For example, only recently we have discovered, by using manned submersibles, that a large amount of methane and carbon dioxide comes from the seafloor and extraordinary groups of organisms live in hydrothermal vent areas. However, a number of complex issues due to the unstructured and hazardous undersea environment make it difficult to survey in the ocean even though today's technologies have allowed humans to land on the moon and robots to travel to Mars.

Unmanned underwater vehicles (UUVs) can help us better understand marine and other environmental issues, protect the ocean resources of the earth from pollution, and efficiently utilize them for human welfare. The UUV is a platform for a variety of sensors: acoustic, magnetic, gravimetric and chemical ones. Most commercial UUVs are tethered and remotely operated, referred to as remotely operated vehicles (ROVs). Extensive use of manned submersibles and ROVs are currently limited to a few applications because of very high operational costs, operator fatigue and safety issues. The demand for advanced underwater vehicle technologies is growing and will eventually lead to fully autonomous and reliable underwater vehicles. Autonomous underwater vehicles (AUVs) were initially developed to perform missions that were not easy for ROVs and manned underwater vehicles. Since the autonomy allows AUVs to be used for risky missions such as a mine countermeasure (MCM) or under-ice operations, AUVs are replacing ROVs towed vehicles as well as manned underwater vehicles (Whitcomb, 2000). For detailed ocean surveys, an AUV acts as a more stable platform for precision sensors than ROVs or towed vehicles because an AUV is not subject to physical disturbances transmitted along the cable to the surface vessel. This absence of physical attachment also allows AUVs to measure ocean characteristics at specific depths and perform bottom-following missions as owing to its autonomy. In short, An AUV provides marine researchers with a new form of access to deeper ocean.

For an AUV to successfully complete a typical survey mission, it must follow a path specified by the operator as closely as possible and arrive at a precise location for collecting data. When an AUV is not able to follow the path accurately during the mission, critical

features may not be recorded and the position of any features during the mission will be uncertain. When the final position of an AUV is not accurate, the AUV may even be unrecoverable. Since AUVs normally carry out missions in wide, unstructured and highly dynamic environments, an accurate and 'reliable' navigation system is required. Thus the most important part of an AUV is the navigation system.

There are many navigation systems for underwater vehicles. Global positioning system (GPS) based navigation systems offer good navigation solutions due to the absolute positioning capability (Farrell & Barth, 1999). However, the drawback of the GPS based navigation systems is that the AUVs need to rise to shallow water depths in order to communicate with satellites, which is a time and energy consuming task (Marco & Healey, 2001). An alternative to the GPS based devices is an acoustic based positioning system (Larsen, 2000, Lee et al., 2004). An acoustic system uses external sound emitting beacons in order to triangulate its position. While precise, navigation based on this type of device incurs a high cost and limits the mission space to the area covered by the beacons. Dead reckoning systems, such as the inertial navigation system (INS) or the Doppler velocity log (DVL), estimate the position of an AUV with respect to the initial position by measuring linear and angular velocities and/or accelerations. Using this type of sensor offers a practical and inexpensive navigation method. Nevertheless, these sensors accumulate drifts over time because position errors tend to increase with any new measurement.

Unlike ground and aerial vehicles, an AUV presents a uniquely challenging navigational problem because it operates autonomously in a highly unstructured environment where satellite based navigations such as the GPS are not directly available. In this respect, many AUVs basically navigate with the help of underwater navigation systems which are comprised of optimal filters and sensors such as an inertial measurement unit (IMU), a DVL, a current meter and a magnetic compass.

The primary challenge of navigation systems for AUVs is maintaining the accuracy of the position of the AUVs over the course of a long mission. An initially accurate position may quickly become uncertain through variations in the motion of the AUVs. While this effect can be reduced by using accurate acceleration, heading and velocity sensors, but these sensors cannot be made as accurate as required. During long missions, these inaccuracies become significant. Temporal and strong currents or other underwater disturbances that affect the motion of the AUV cannot be precisely modelled in advance and these may deteriorate the accuracy further. If the position of an AUV is not externally referenced, the position accuracy will inevitably degrade over the course of the mission. The absence of an easily observable, external reference makes the navigation of AUVs very difficult. External references must be used for any AUV navigation systems that yield to an accurate navigation over long missions.

As stated above, the navigation system may employ multiple measuring devices to enhance the accuracy and reliability of the system. This practice becomes important because existing systems can be upgraded by supplementing more accurate and economic navigation devices. This implies the necessity of finding a new way of sharing and merging data obtained from the various devices. Since the data collected by each device represent only partial information of the phenomenon under survey, a process of 'sensor data fusion' is required in order to acquire complete information. With measurement fusion, some measured data for both steady and unsteady systems are passed directly to the fusion centre for centralized Kalman filter. Because of its relatively lower state estimation errors, the measurement fusion is a widely used method (Titterton & Weston, 1997).

However, it is well known that an AUV often experiences position errors caused by environmental disturbances as mentioned above. This is particularly the case for an inertial navigation system with velocity measurement and when the standard Kalman filter is influenced by the action of temporary currents (Jo et al., 2006). This is the reason why robust filters for navigation systems have been studied (Yu et al., 2004, Seo et al., 2006). This article focuses on the receding horizon Kalman filter (RHKF), which is a kind of robust filter.

The estimation of the state of a dynamic system by using the measurement obtained from the most recent time is defined in various ways, including fixed memory, receding horizon or sliding window estimation (Bierman, 1975). These estimation methods were originally reported in the work of Jazwinski (1968). Since then, the estimation method has been widely used in many application areas, where measurement uncertainties hinder the proper use of the Kalman filter.

Finite impulse response (FIR) filters utilize finite measurements over the most recent time interval. FIR structured filters have actually become a standard filter commonly used because they are more robust against numerical errors and temporary uncertainties than infinite impulse response (IIR) structured filters, which utilize all measurements on the infinite interval.

Kwon et al. (1989) and Kwon et al. (1999) suggested the use of optimal FIR filters and the receding horizon Kalman FIR filter for discrete linear time invariant systems for state estimation. When using these filters, error covariance is minimized in discrete-time stochastic state-space models. The optimal FIR filter has several advantages over existing optimal filters such as the cerebrated Kalman filter, which is a kind of an optimal IIR filter. Since the optimal FIR filter utilizes finite measurement over the most recent time interval, it is robust against temporary modelling and measurement uncertainties that may cause a divergence of the estimated state. Divergence was a problem for the optimal IIR structured filter. An optimal FIR filter can be effectively derived by modifying the Kalman filter because the Kalman filter is easy to use and widely applied in many engineering problems. In this article, the standard Kalman filter is combined with the receding horizon strategy which has already been adopted in many optimal control and estimation methods. We denote it as the RHKF for time-varying systems.

The velocity-aided inertial navigation system is relatively underdeveloped for underwater applications, although it has been proven to be a good method for land applications (Marco & Healey, 2001). The DVL is a kind of measuring device used for velocity information. However, the DVL can give reliable velocity information to AUVs only when the AUVs cruise close to the bottom or the surface. Current meters are used more often for an AUV than the DVL devices because current meters measure the relative speed between the AUV and the surrounding fluid. The devices have no limit of relative distances to surface or bottom. The disadvantage of current meters is that their accuracy may be deteriorated by ocean current. In general, the ocean current often occurs temporarily and its magnitude can exceed the speed of the vehicle. In this case, the velocity-aided navigation system with the Kalman filter is deteriorated by temporary disturbances. In the worst case, the position error of the system may even 'blow up'. However, as it will be shown, the underwater navigation system based on the RHKF is robust against temporary disturbances. Therefore the current meters can be used for an accurate the velocity-aided navigation system for AUVs with help of the RHKF.

The statistical process and measurement noises are often unknown a priori in most practical situations. In such a situation, the Kalman filter may fail to accurately estimate state

variables or may even cause system failure due to divergence (Fitzgerald, 1971, Sangsuk-Iam & Bullock, 1990). With these uncertainties, it is difficult to develop an accurate navigation system with the Kalman filter. The difficulty can be overcome by using a number of estimation methods of noise covariance and some have been proposed. The problem is also important to make the navigation systems accurate. But it must be kept in mind that it remains herein. More information can be found in Mehra (1970), Belanger (1974), Um et al. (2000) and Jo et al. (submitted).

This chapter contains the following information: In section II, the error equations for a velocity-aided underwater navigation system are introduced. In section III, the RHKF for time-varying systems is derived in an iterative form. In section IV, simulations for a linear time-varying system and the navigation system are explained. Finally, a short summary is given.

2. Dynamic model for navigation errors

2.1 Coordinate systems

Coordinate systems used in this chapter are the inertial frame (i-frame), the Earth-fixed frame (e-frame), the navigational frame (n-frame) and the body frame (b-frame). The i-frame is fixed with the centre of the Earth as the origin. The e-frame coincides with the i-frame at the origin but rotates with the Earth rate. The n-frame is a local level frame in which the vertical axis is parallel with the gravity vector. Fig. 1 shows the reference frames.

In the strapdown inertial navigation system (SDINS), the determination of the transformation matrix from the b-frame to the n-frame is very important and error generated during the computation of the transformation matrix becomes one of the main error sources of the system.

Various mathematical representations are used to define the attitude of a body with respect to a coordinate reference frame. In three-dimensional space, spatial rotations can be pararmeterized using both Euler angles and unit quaternions. The parameters associated with each method may be stored in computer and updated as the vehicle rotates, using the measurements of turn rate by the SDINS. Theorems derived from the Euler angles state that any given sequence of rotations can be represented as a single rotation about a single fixed axis. In the Euler angle representation, the attitude is defined by roll (Φ), pitch (θ) and yaw (ψ) angles, which constitute a direction cosine matrix (DCM) (Siouris, 1993). The yaw angle is equivalent to the heading angle while the roll and pitch angles represent the levelling angles.

However, among the popular methods of attitude computations, quaternions are known to be the most effective one due to their simplicity and easy normalization procedure, in modern navigation systems (Siouris, 1993). The representation of rotation as a quaternion is more compact than the representation of the DCM or the Euler angles. Furthermore, for a given axis and angle, one can easily construct the corresponding quaternion and, conversely, for a given quaternion one can easily read off the axis and the angle. When composing several rotations on a computer, round off errors inevitably accumulate. A quaternion that is slightly off still represents a rotation after being normalised. The quaternion also avoids the phenomenon called gimbal lock which can result when, for example in the Euler angle represensation, the pitch angle is rotated by 90° up or down, so that yaw and roll correspond to the same motion, and a degree of freedom of roation is lost. In the SDINS, for instance, this could lead to a disastrous result if the vehicle is in a steep

dive or ascent. For more rigorous and mathematical analysis, refer to Altmann (1986). In this chapter, the dynamic equation for attitude is mainly described by quaternions.
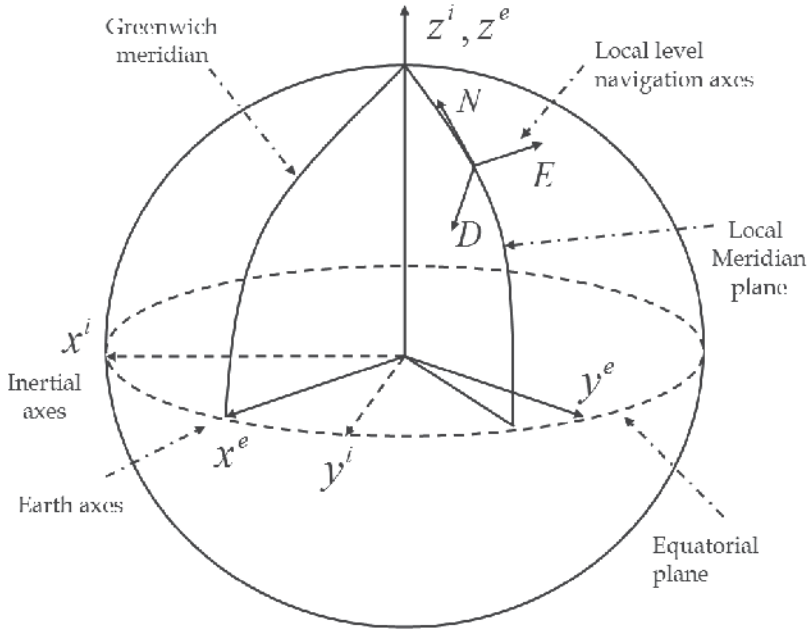

Fig. 1. Reference frames

2.2 Error model for SDINS

The SDINS error model plays an important role in implementing an optimal filter for alignment or for velocity-aided navigation algorithms. The quaternion error model with the attitude and the velocity error state for SDINS was first developed in the i-frame by Friedland (1978), and later in the n-frame by Shibata (1986). The axes of the n-frame point in the directions of north, east and downward in that order. In the n-frame, the rate vector is represented as follows:

$$\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n$$
$$\omega_{ie}^n = \begin{bmatrix} \Omega_N & 0 & \Omega_D \end{bmatrix}^T = \begin{bmatrix} \Omega\cos L & 0 & -\Omega\sin L \end{bmatrix}^T$$
$$\omega_{en}^n = \begin{bmatrix} \rho_N & \rho_E & \rho_D \end{bmatrix}^T = \begin{bmatrix} \dot{l}\cos L & -\dot{L} & -\dot{l}\sin L \end{bmatrix}^T$$

where ω_{in}^n is the rate of the n-frame relative to the i-frame expressed in the n-frame, ω_{ie}^n is the Earth rate, ω_{en}^n is the rate of the n-frame relative to the e-frame, L denotes the latitude, l represents the longitude and Ω is the magnitude of the Earth rate. The superscripts n and b denote the n-frame and the b-frame, respectively.

The dynamic equations of the position, velocity and attitude of SDINS in the n-frame are given as follows (Titterton & Weston, 1997):

$$\dot{L} = \frac{V_N}{R_m + h} \ , \ \dot{l} = \frac{V_E}{(R_t + h)\cos L} \ , \ \dot{h} = -V_D \tag{1}$$

$$\dot{v}^n = C_b^n f^b - \left(2\omega_{ie}^n + \omega_{en}^n\right) \times v^n + g^n \tag{2}$$

$$\dot{C}_b^n = C_b^n \Omega_{nb}^b \tag{3}$$

with $R_m = R_0(1 - 2e + 3e\sin^2 L)$ and $R_t = R_0(1 + e\sin^2 L)$, where R_0 is the radius of the Earth at the equator, e denotes the major eccentricity of the Earth, h is the height, g^n is the gravitational force represented in the n-frame, $f^b \left(= [f_x \quad f_y \quad f_z]^T\right)$ denotes the specific force measured at the accelerometer, $v^n \left(= [V_N \quad V_E \quad V_D]^T\right)$ is the velocity of the vehicle represented in the n-frame, $\omega_{nb}^b \left(= [\omega_x \quad \omega_y \quad \omega_z]^T\right)$ is the rate of the b-frame relative to the n-frame, and Ω_{xy}^z is the skew-symmetric matrix for the rate ω_{xy}^z.

The relationships among the quaternion, Euler angles and DCM are given by

$$
\begin{aligned}
C_b^n &= \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \cos\psi\sin\phi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \\
&= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}
\end{aligned} \tag{4}
$$

The attitude equation by the quaternion is

$$\dot{q} = 0.5 q * \omega_{nb}^b = 0.5 q * \left(\omega_{ib}^b - C_n^b \left(\omega_{ie}^n + \omega_{en}^n\right)\right) \tag{5}$$

where $q = q_0 + \hat{i}q_1 + \hat{j}q_2 + \hat{k}q_3$ is a quaternion, ω_{ib}^b denotes the measurement of gyroscopes and * is the quaternion product which is defined as

$$
\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} * \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{6}
$$

The error model of SDINS can be obtained by the perturbation method under several assumptions (Titterton & Weston, 1997)

$$\dot{\delta L} = \frac{\rho_E R_{mm}\delta L + \rho_E \delta h + \delta v_N}{R_m + h} \tag{7}$$

$$
\begin{aligned}
\dot{\delta l} = &\ \rho_N \sec L \left(\tan L - \frac{R_{tt}}{R_t + h}\right)\delta L \\
&- \frac{\rho_N \sec L \delta h - \sec L \delta v_E}{R_t + h}
\end{aligned} \tag{8}
$$

$$\delta \dot{h} = -\delta V_D \tag{9}$$

$$\delta \dot{v} = [C_b^n f^b] \times \varepsilon - (2\omega_{ie}^n + \omega_{en}^n) \times \delta v^n + C_b^n \delta f^b$$
$$+ v^n \times (\delta 2\omega_{ie}^n + \delta \omega_{en}^n) \tag{10}$$

$$\dot{\varphi} = -\omega_{in}^n \times \varphi - C_b^n \delta \omega_{ib}^b + \delta \omega_{in}^n \tag{11}$$

with $R_{mm} = \partial R_m / \partial L = 6R_0 e \sin L \cos L$ and $R_{tt} = \partial R_t / \partial L = 2R_0 e \sin L \cos L$, where δL, δl and δh are errors of latitude, longitude and altitude, respectively, and $\delta v^n = [\delta v_N \quad \delta v_E \quad \delta v_D]^T$ is a velocity error in the n-frame.

$\varphi = [\varphi_N \quad \varphi_E \quad \varphi_D]^T$ is a tilt angle that is approximately equal to the Euler angle error under a small angle assumption and $\delta \omega_{ie}^n$, $\delta \omega_{en}^n$ and $\delta \omega_{in}^n$ are defined as

$$\delta \omega_{ie}^n = [-\Omega \sin L \delta L \quad 0 \quad -\Omega \cos L \delta L]^T \tag{12}$$

$$\delta \omega_{en}^n \approx \begin{bmatrix} -\dfrac{\rho_N}{R_t + h}\delta h + \dfrac{1}{R_t + h}\delta v_E \\[2mm] -\dfrac{\rho_E}{R_m + h}\delta h - \dfrac{1}{R_m + h}\delta v_N \\[2mm] -\rho_N \sec^2 L \delta L - \dfrac{\rho_D}{R_t + h}\delta h + \dfrac{\rho_D}{v_E}\delta v_E \end{bmatrix} \tag{13}$$

$$\delta \omega_{in}^n = \delta \omega_{ie}^n + \delta \omega_{en}^n \tag{14}$$

In (10) and (11), δf^b is an accelerometer error vector and $\delta \omega_{ib}^b$ is a gyro error vector. These errors of inertial sensors may be simply modelled as a sum of random constant and white Gaussian noise.

$$\delta f^b = \nabla_a + w_a(t), \quad w_a(t) \sim N(0, Q_a)$$
$$\dot{\nabla}_a = 0, \quad \nabla_a = [\nabla_x \quad \nabla_y \quad \nabla_z]^T \tag{15}$$

$$\delta \omega_{ib}^b = \varepsilon_g + w_g(t), \quad w_g(t) \sim N(0, Q_g)$$
$$\dot{\varepsilon}_g = 0, \quad \varepsilon_g = [\varepsilon_x \quad \varepsilon_y \quad \varepsilon_z]^T \tag{16}$$

Herein, the biases of the accelerometers and gyros, ∇_a and ε_g, are assumed to be random constant even though they generally vary very slowly.

2.3 Error model of measurements

Some auxiliary sensors should be used to compensate for the navigation errors of the SDINS. A pressure sensor, current meters and the magnetic compass can be used as auxiliary sensor in the navigation system for AUVs. A surface navigation system has been successfully developed by integrating position fixing systems such as GPS. As stated above, introducing GPS to the navigation system for an AUV is limited to the case of shallow water vehicles repeatedly surfacing to update the position information. If available, long-base line

(LBL) and short-base line (SBL) systems can be generally used for positioning underwater vehicles. Furthermore, near surface navigation systems can be utilized when a vehicle operates near the water surface or intentionally approaches the surface. The navigation systems for long-range cruising-type AUVs are structured by dead-reckoning. However, the inevitable growth of errors within the navigation system motivates the need for on-line calibration methods such as GPS-aided or Loran-C-aided navigation. Larsen (2000) and Lee et al. (2004) proposed hybrid navigation systems based on the IMU combined with acoustic velocity sensors.

We simply assumed such a situation for a numerical example. The modelled vehicle is assumed to be equipped with an IMU, a 3-axis magnetic-type current meter, a 3-axis magnetic compass and a pressure sensor to obtain depth information. They are also modeled to be the sum of random constant and white Gaussian noise.

The equations of the sensors can be formulated by

$$\hat{h}_{INS} - h_m = h_{true} + \delta h - (h_{true} + h_{bias}) = \delta h - h_{bias} \tag{17}$$

$$\hat{v}_{INS}^n - v_m = \hat{v}_{true}^n + \delta v^n - C_b^n (v_{true}^n + v_{bias})$$
$$\simeq \delta v^n - v^n \times \phi - C_b^n v_{bias} \tag{18}$$

$$\begin{bmatrix} \hat{\phi}_{INS} \\ \hat{\theta}_{INS} \\ \hat{\psi}_{INS} \end{bmatrix} - \begin{bmatrix} \phi_m \\ \theta_m \\ \psi_m \end{bmatrix} = - \begin{bmatrix} \phi_N \\ \phi_E \\ \phi_D \end{bmatrix} - \begin{bmatrix} \phi_{bias} \\ \theta_{bias} \\ \psi_{bias} \end{bmatrix} \tag{19}$$

where $\wedge$ denotes the estimated value by SDINS and $[\phi \quad \theta \quad \psi]^T$ denotes the roll, pitch and yaw angle of the vehicle, respectively. The subscript m, INS and bias denotes the measured value of the sensors, the estimated value of optimal filters and sensor bias. Optimal filters for the navigation system can be updated with these equations.

Combining (1)-(19) with the differential equations for sensor errors yields the following error equation for the velocity-aided navigation system.

$$\dot{x}(t) = F(t)x(t) + w(t), \quad w \sim N(0, Q(t)) \tag{20a}$$

where

$$x = \begin{bmatrix} x_{INS}^T & x_{bias}^T \end{bmatrix}^T$$

$$x_{INS} = \begin{bmatrix} \delta P^T & (\delta v^n)^T & \varphi^T & \nabla_a^T & \varepsilon_g^T \end{bmatrix}^T$$

$$x_{bias} = \begin{bmatrix} h_{bias} & v_{bias}^T & \phi_{bias} & \theta_{bias} & \psi_{bias} \end{bmatrix}^T$$

$$w = \begin{bmatrix} 0_{1\times9} & w_a^T & w_g^T & 0_{1\times7} \end{bmatrix}^T$$

The time-varying system matrix $F(t)$, which is a differential equation of SDINS can be used to estimate the position, velocity, attitude and the biases of the sensors (See Appendix for

details). The state variable $x(t)$ has 22 error states: δP denotes the position error in the latitude, longitude and altitude. δv^n denotes the velocity errors in the n-frame, while φ denotes the vector of roll, pitch and yaw angles.

The innovation (also called measurement residual) of the optimal filter is the difference between estimated values by SDINS and measurements of pressure, velocity and attitude. The innovation for a velocity-aided underwater navigation system at t_k may be expressed in terms of the error state variables as follows:

$$y(t_k) = \begin{bmatrix} \hat{h}_{INS} \\ \hat{v}^n_{INS} \\ \hat{\phi}_{INS} \\ \hat{\theta}_{INS} \\ \hat{\psi}_{INS} \end{bmatrix} - \begin{bmatrix} h_m \\ v^n_m \\ \phi_m \\ \theta_m \\ \psi_m \end{bmatrix}$$

$$= \begin{bmatrix} 0_{1\times2} & 1 & 0_{1\times12} & -1 & 0_{1\times3} & 0_{1\times3} \\ 0_{3\times3} & I_{3\times3} & -V^n\times & 0_{3\times7} & -C^n_b & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -I_{3\times3} & 0_{3\times7} & 0_{3\times3} & -I_{3\times3} \end{bmatrix} x(t_k) \qquad (20b)$$

$$+ \begin{bmatrix} v_h(t_k) \\ v_{cur}(t_k) \\ v_{com}(t_k) \end{bmatrix}$$

where v_h, v_{cur} and v_{com} are the measurement noise of the pressure sensor, current meters and compass, respectively.
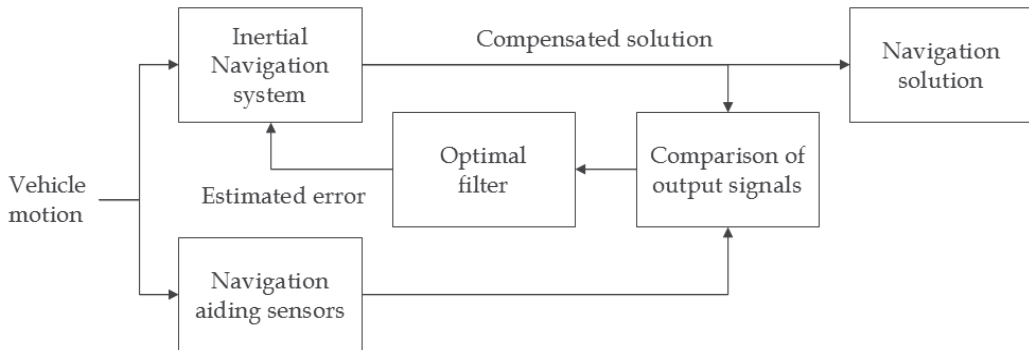

Fig. 2. Indirect feedback structure

In general, while the inertial navigation system is nonlinear, the error of the system can be assumed to be linear. This is the reason why the indirect feedback method is used for the navigation system. Fig.2 shows the block diagram of the indirect feedback method. In the figure, it can be seen that the navigation system predicts the position, velocity, attitude and biases of the sensors with the output of the SDINS and then compensates the error states for the system by using an optimal filter. The navigation solutions are updated indirectly each time when the depth, velocity and attitude are measured by the sensors. In the indirect scheme, the navigation solutions are already given as the predetermined nominal points,

while the Kalman filter estimates the navigation solution. The word 'indirect' means that the filter estimates not the navigation solutions but the error of the solutions. Incorporated with the SDINS equations, the indirect method becomes efficient. If the navigation system uses indirect feedback states, the error system consists of a linearized system with compensated states.

3. Receding horizon Kalman filter for underwater navigation systems

In this section, the receding horizon Kalman filter is introduced for the underwater navigation system. Consider a linear discrete time-varying state space model with control input

$$\begin{aligned} x_{k+1} &= F_k x_k + B_k u_k + G w_k \\ y_k &= H_k x_k + v_k \end{aligned}$$ (21)

where $x_k \in \Re^n$ is the state vector and $u_k \in \Re^l$ and $y_k \in \Re^q$ are the input vector and the measured output vector, respectively.

The initial state x_{k_0} is assumed to be random with a certain mean $\bar{x}_{k_0}$ and a certain covariance Σ_{k_0}. The process and measurement noises are assumed to be zero-mean white Gaussian and are not correlated with each other. The covariances of w_k and v_k are denoted by Q and R, respectively. It is assumed that these noises are not correlated with the initial state x_{k_0}.

The following Kalman filter for a time-varying system provides a state estimate $\hat{x}_k$ called, the one-step predicted estimate of the system state x_k, with control input

$$\hat{x}_{k+1} = F_k \hat{x}_k + K_k (y_k - H_k \hat{x}_k) + B_k u_k$$ (22)

$$K_k = F_k P_k H_k^T (R + H_k P_k H_k^T)^{-1}$$ (23)

$$P_{k+1} = F_k P_k F_k^T + G Q G^T - F_k P_k H_k^T (H_k P_k H_k^T + R)^{-1} H_k P_k F_k^T$$ (24)

with $\hat{x}_{k_0} = \bar{x}_{k_0}$, where P_k is the error covariance of the estimate $\hat{x}_k$ with the initial value $P_{k_0} = \Sigma_{k_0}$.

In order to derive the RHKF for the stochastic systems, input and output information on the horizon $[k-N, k]$ are utilized together with information about the state at the starting point $k-N$. We write $k_N \equiv k-N$ for convenience. We refer to this state at t_k as the horizon initial state. It is logical to assume that the horizon initial state cannot be measured and thus is unknown. An information form of the Kalman filter is adopted as follows:

At first, let us define the new variables

$$\Omega_k \equiv P_k^{-1}, \quad \bar{\Omega}_k \equiv P_k^{-1} + H_k^T R^{-1} H_k$$

The estimation error covariance (24) can then be rewritten as

$$\Omega_{k+1} = [I + F_k^{-T} \bar{\Omega}_k F_k^{-1} G Q G^T]^{-1} F_k^{-T} \bar{\Omega}_k F_k^{-1}$$ (25)

with $\Omega_{k_0} = \Sigma_{k_0}^{-1}$

The information form of filter (22) becomes

$$\hat{x}_{k+1} = F_k \bar{\Omega}_k^{-1} \left(\Omega_k \hat{x}_k + H_k^T R^{-1} y_k \right) + B_k u_k . \tag{26}$$

The information form (26) of the Kalman filter uses all measurements starting from the initial time k_0 to provide the one-step predicted state estimate at the present time. By introducing the receding horizon strategy to filter (26), the RHKF at the present time k uses only the finite measurements on the horizon $[k_N, k]$ and discards the past measurements outside the horizon. We re-derived filter (26) on the horizon covering from the horizon initial time k_N to the present time k. The filter at time $k_N + i$ on the horizon $[k_N, k]$ is denoted as $\hat{x}_{k_N+i|k}$ with $0 \le i < N$. In (24), the error covariance of the estimate $\hat{x}_k$, system F_{k_N+i} and measurement equations H_{k_N+i} are not correlated with the measurements for linear time-varying systems. The horizon initial condition is denoted as Ω_{k_N}. Since the measurements and control inputs are not estimated variables, these can be written as $y_{k_N+i|k} = y_{k_N+i}$ and $u_{k_N+i|k} = u_{k_N+i}$.
The filter can now be re-written by

$$\hat{x}_{k_N+i+1|k} = F_{k_N+i} \bar{\Omega}_{k_N+i}^{-1} \left(\Omega_{k_N+i} \hat{x}_{k_N+i|k} + H_{k_N+i}^T R^{-1} y_{k_N+i} \right) + B_{k_N+i} u_{k_N+i} \tag{27}$$

In (25), $\Omega_{k_N+i} > 0$ for all $i \ge n$, if $\{F_k, H_k\}$ is uniformly observable (Kwon & Pearson, 1978). In the above equation, n denotes the dimension of the system. This guarantees that a converged result is globally optimal to Kalman filter based FIR filters. Since it is difficult to know the horizon initial state $x_{k_N|k}$, it is assumed to be unknown. The horizon initial state must have an arbitrary mean and infinite covariance as $\Omega_{k_N} = 0$.
Theorem 1 RHKF for time-varying systems (Jo & Choi, 2006)
Assume that $\{F_k, H_k\}$ is uniformly observable and $N \ge n$. If the horizon initial state $x_{k_N|k}$ is assumed to be unknown on the horizon $[k_N, k]$, the RHKF for the state x_k is given by

$$\hat{x}_{k_N+i+1|k} = F_{k_N+i} \bar{\Omega}_{k_N+i}^{-1} \left(\Omega_{k_N+i} \hat{x}_{k_N+i|k} + H_{k_N+i}^T R^{-1} y_{k_N+i} \right) + B_{k_N+i} u_{k_N+i} \tag{28}$$

where

$$\Omega_{k_N+i+1} = [I + F_{k_N+i}^{-T} \bar{\Omega}_{k_N+i} F_{k_N+i}^{-1} G Q G^T]^{-1} F_{k_N+i}^{-T} \bar{\Omega}_{k_N+i} F_{k_N+i}^{-1} \tag{29}$$

Jo & Choi (2006) proved that the derived filter (28) has two important properties – it is unbiased and deadbeat and that the RHKF becomes a deadbeat observer when the filter is applied to the following noise-free system:

$$\begin{aligned} x_{k+1} &= F_k x_k + B_k u_k \\ y_k &= H_k x_k \end{aligned} \tag{30}$$

If (30) is contaminated by noises similar to (21), the RHKF optimally compensates for the noises. When the system has no noise, the RHKF can yield an exact estimate of the state. This deadbeat property indicates the finite convergence time and the fast tracking ability of the RHKF. Thus, we can expect that the suggested RHKF is appropriate for quick estimation and detection of AUV tracking even when occurrence of noises is not known.

4. Simulation

In this section, the rate at which the RHKF converges against temporary unknown disturbances is determined and its robustness is illustrated. At first, the RHKF is applied to a linear time-varying system. The RHKF is then applied to the velocity-aided underwater navigation system.

Consider the following linear system:

$$\dot{x}(t) = A(t)x(t) + Gw(t)$$
$$y(t) = H(t)x(t) + v(t)$$

(31)

with

$$A(t) = \begin{bmatrix} -a + (a-b)\sin^2 \omega t + \delta(t) & \omega + (a-b)\sin \omega t \cos \omega t \\ -\omega + (a-b)\sin \omega t \cos \omega t & -a + (a-b)\cos^2 \omega t + \delta(t) \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \delta(t) = \begin{cases} 0.05 & 100 \le t < 200 \\ 0 & \text{otherwise} \end{cases}$$

where $x \left(= [x_1 \quad x_2]^T \right)$ is the state vector, $w \left(= [w_1 \quad w_2]^T \right)$ denotes the process noise vector, y is the output and v is the measurement noise. When $a = 0.003$, $b = 0.007$ and $\omega = 0.01$ are chosen, and the variance of the noises is assumed to be 0.05, the estimated error of x_1 is as shown in Fig. 3. The RHKF estimates the present state x_k with information on the prescribed horizon interval. The RHKF has some advantages (robustness against temporary disturbances and fast tracking ability), if the horizon interval is properly chosen. However, it takes a considerable amount of time to estimate the present state because the filter recedes from the horizon initial state each time. This is the reason why the RHKF takes more computing time (of about N times) than the Kalman filter, which has an IIR structure. A proper horizon interval must be chosen carefully for a real-time system.

In Fig. 3, as the horizon interval increases, the estimates obtained by the RHKF more closely approach those provided by the Kalman filter and it becomes less robust against the temporary disturbance. It can be also seen that, if the horizon interval is small, the estimates obtained by the RHKF fluctuate around the real value instead of converging to the real one. However, if the interval is large, it approaches closer to the estimate obtained from the Kalman filter. The horizon interval is important in the optimal estimation. Unfortunately, it is impossible to find an optimal horizon interval for general non-linear systems. Thus, a reasonable interval N must be chosen based on experience. The rule of thumb is to take it slightly larger than the limit of the index to converge to the covariance of the standard Kalman filter.

Let us consider that an AUV is equipped with a low-degree IMU, a pressure sensor, a 3-axis current meter, and a 3-axis magnetic compass. The sampling rate of the IMU is 100 Hz and that of the pressure sensor, 3-axis current meter, and the 3-axis magnetic compass is all 1 Hz. The characteristics of the sensors are listed in Table 1.

In this simulation, the following assumptions are made: The vehicle moves directly north at a rate of 2m/s and downward at a rate of 0.05m/s without any change in attitude for 1200 seconds. It is known that SDINS has poor observability during straight forward running without any change in attitude because the attitude error model depends on the angular rate of the vehicle (Lee et al., 1993).
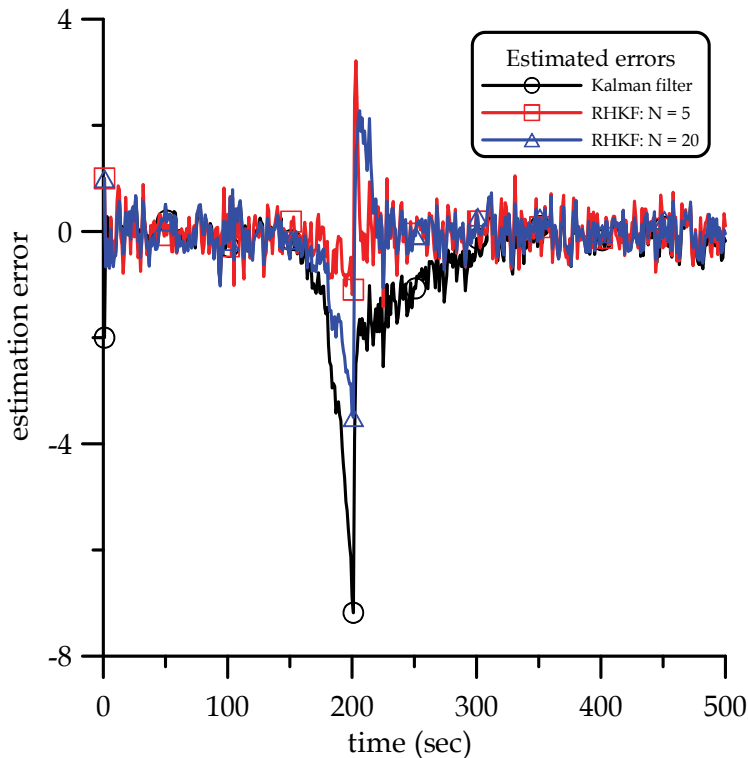
Fig. 3. Comparison of estimated error for the linear system

		Bias (1σ)	Noise (1σ)
IMU	Acceleromters	9.8 mm/s²	0.49 mm/ s²
	Gyros	1deg/hr	0.35 deg/hr
Pressure sensor		0.1 m	0.1 m
Current meter		0.01 m/s	0.01 m/s
Compass		1 deg	2 deg

Table 1. Characteristics of Sample Sensors

Upwelling current of 0.5 m/s is introduced as a temporal disturbance. Fig. 4 shows the assumed velocity profile of the upwelling current in the NED coordinate. As shown in the figure, the upwelling current occurred temporarily during a time span between 385 sec and 655 sec. Fig. 5 and Fig. 6 show the reference trajectory of the vehicle under the action of the disturbance and Fig. 7 shows the reference velocity profile of the vehicle. It was assumed that the vehicle reacted against the current in 5 sec when it encountered with the current as shown in Fig. 7.

The trajectory and velocity of the vehicle under the upwelling current were estimated using the navigation filters, the standard Kalman filter and the RHKF. Figs. 8 shows the estimated velocity obtained by the Kalman filter. By comparing Fig. 7 with Fig. 8, it can be seen that the esimation error of the Kalman filter incearses after the vehicle encounters with the current.

The choice of the horizon interval N affects the convergence speed of the RHKF. It is very difficult, as stated above, to find the optimal horizon interval N for general nonlinear systems. However, if a system is uniformly observable, the horizon interval N can be chosen as larger than the dimension of the system, because $\Omega_{k_N+i|k} > 0$ for all $i \geq n$, if $\{F_{k_N+i|k}, H_{k_N+i|k}\}$ is uniformly observable. Hereby n denotes the dimension of the system. It may be recommended with care that one may choose the interval to be larger than the dimension of the system, but less than 4 times. In this simulation, we chose $N = 30$.

As stated above, the RHKF is robust against temporary modelling and measuring uncertainties because it utilizes only the finite measurements on the most recent horizon. The estimated velocity obtained by the RHKF is shown in Fig. 9. In Fig. 10, it is seen that the navigation system based on the RHKF does not lost its position but the vehicle that was navigated by the Kalman filter has lost its position after the current occurs. While the pressure sensor helps the navigation system to compensate altitude errors, the deviation of estimated altitude obtained by the Kalman filter is larger than that estimated by the RHKF as shown in Fig. 11.

This simulation shows that, while the estimated navigation solution based on the RHKF may be somewhat noisy, the estimated errors do not blow up, which indicates the fast convergent rate and robustness of the RHKF under the temporary disturbances, as given in Fig. 4. Also, the estimation error of the RHKF is considerably smaller than that of the Kalman filter in the interval of measurement uncertainties. Therefore, it is concluded that the suggested RHKF for time-varying navigation systems is more robust than the standard Kalman filter when there are measuring uncertainties.
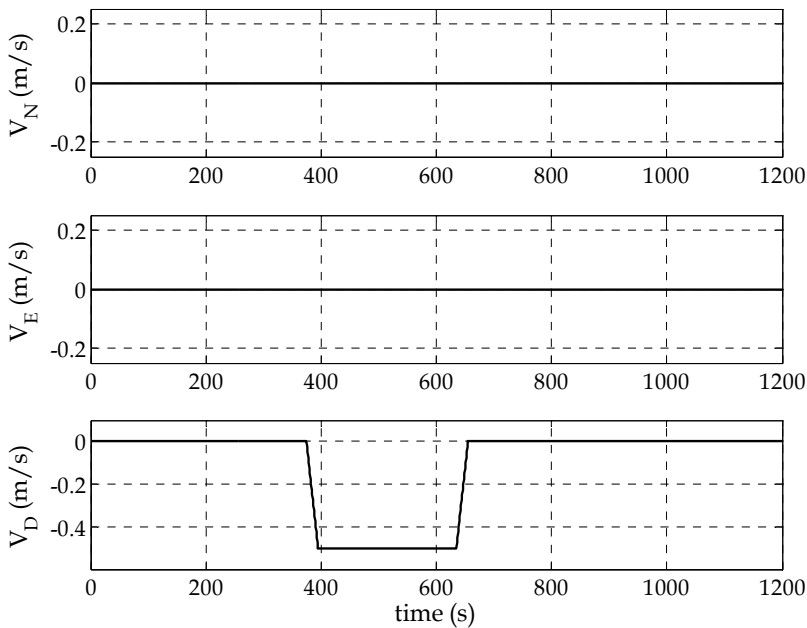

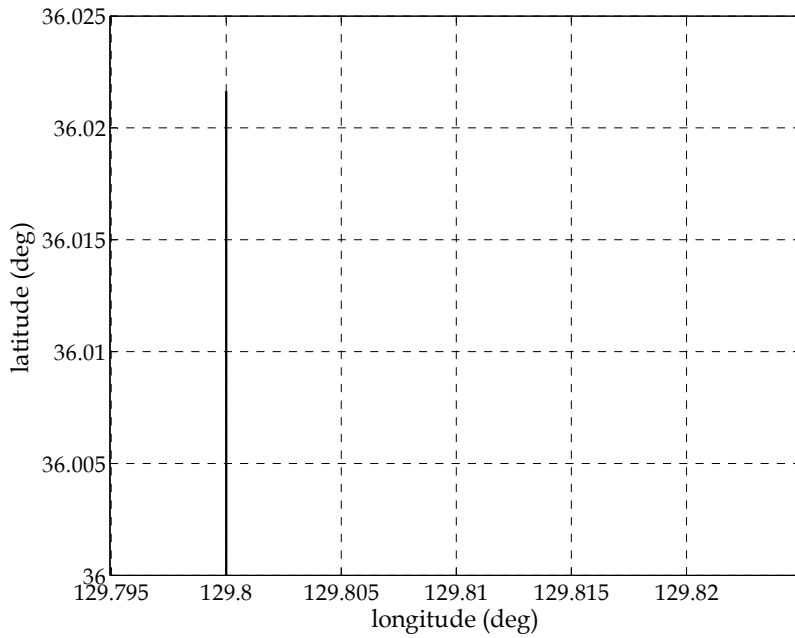Fig. 4. Velocity profile of the upwelling current in the n-frame

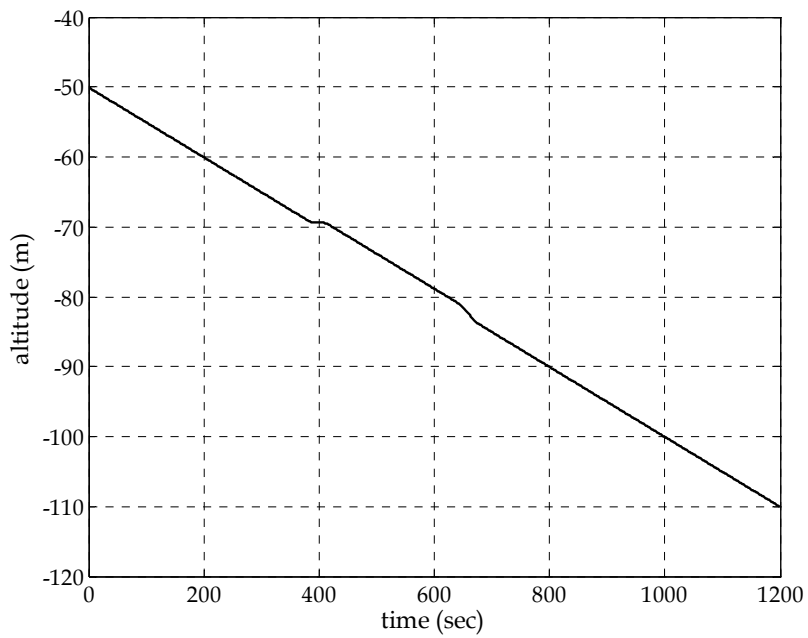Fig. 5. Reference trajectory of the vehicle in the n-frame

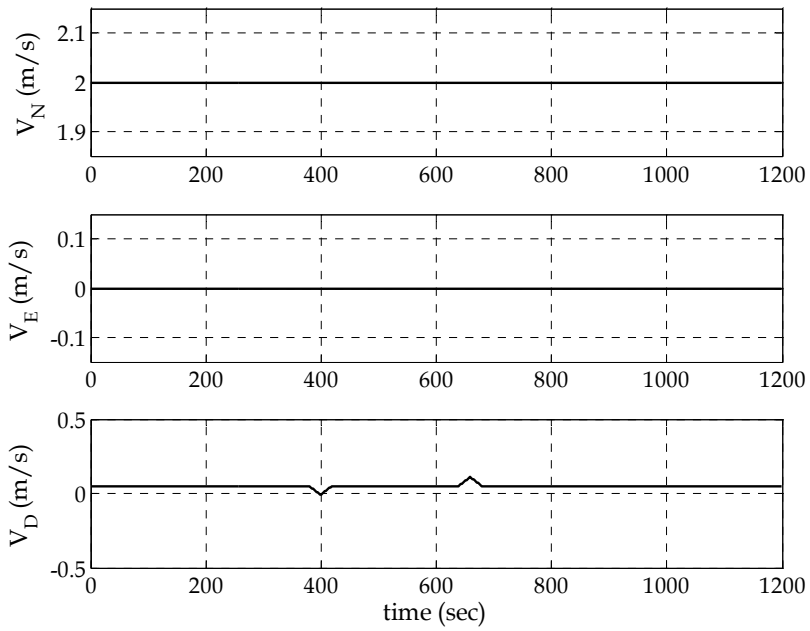Fig. 6. Reference altitude of the vehicle in the n-frame

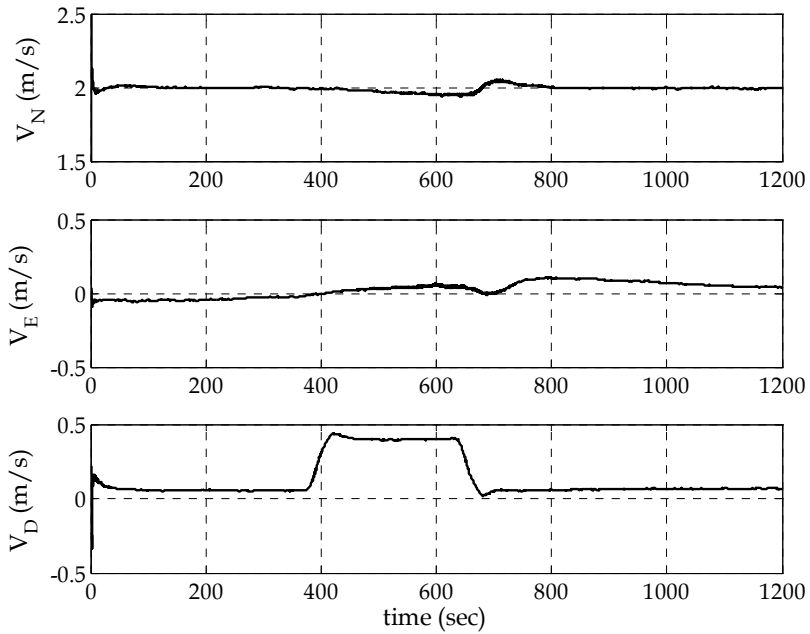Fig. 7. Reference velocity of the vehicle in the n-frame

Fig. 8. Estimated velocity of the vehicle by the Kalman filter in the n-frame
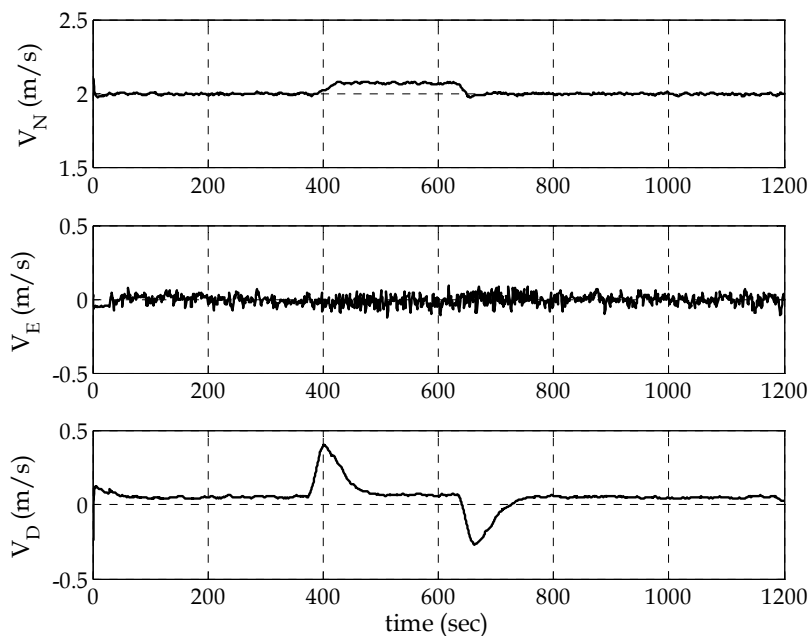
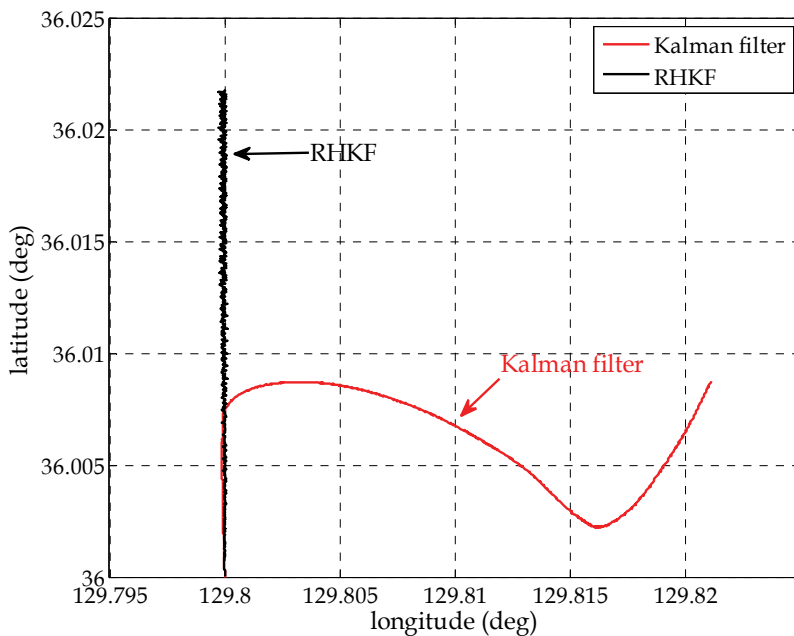Fig. 9. Estimated velocity of the vehicle by the RHKF in the n-frame

Fig. 10. Estimated trajectory of the vehicle by the Kalman filter in the n-frame
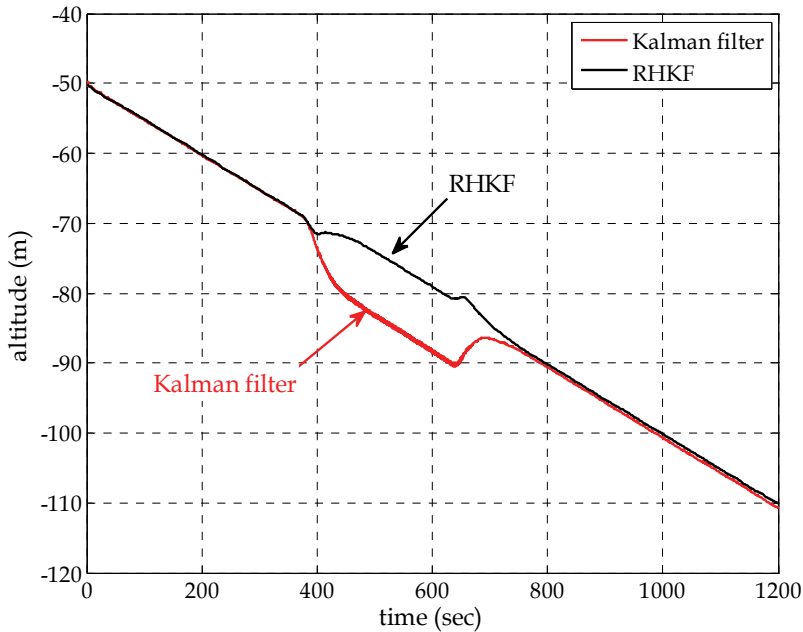
Fig. 11. Comparison of estimated altitudes in the n-frame

6. Summary

In this chapter, the RHKF and its application for a velocity-aided underwater navigation system are discussed. Firstly, the limitations of the standard Kalman filter in underwater navigation systems are described. And then it is shown how the RHKF can be replaced in order to improve the position accuracy of the navigation system when the vehicle is to be operated under uncertain environments. It is shown that the RHKF is robust against environmental uncertainties by tracking only the most recent finite measurement. When a navigation system is completely observable, the RHKF is exact for noise-free systems. This deadbeat property indicates the finite convergence time and fast tracking ability of the filter. Thus the filter is appropriate for fast estimation under temporary disturbances. Based on simulations for the velocity-aided navigation system, it is then demonstrated that the RHKF guides a better and faster transient performance of the underwater navigation system compared to the standard Kalman filter.

7. Appendix

The system matrix $F(t)$ of the SDINS system error model equation is as follows:

$$F = \begin{bmatrix} F_{11} & F_{12} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ F_{21} & F_{22} & F_{23} & C_b^n & 0_{3\times3} & 0_{3\times3} \\ F_{31} & F_{32} & F_{33} & 0_{3\times3} & -C_b^n & 0_{3\times3} \\ & & & 0_{13\times22} & & \end{bmatrix}$$

where

$$
F_{11} = \begin{bmatrix} \dfrac{\rho_E R_{mm}}{R_m + h} & \dfrac{\rho_E R_{mm}}{R_m + h} & \dfrac{\rho_E}{R_m + h} \\[4mm] \rho_N \sec L \left(\tan L - \dfrac{R_{tt}}{R_t + h} \right) & 0 & -\dfrac{\rho_N \sec L}{R_t + h} \\[4mm] 0 & 0 & 0 \end{bmatrix}
$$

$$
F_{12} = \begin{bmatrix} \dfrac{1}{R_m + h} & 0 & 0 \\[4mm] 0 & \dfrac{\sec L}{R_t + h} & 0 \\[4mm] 0 & 0 & -1 \end{bmatrix}
$$

$$
F_{21} = \begin{bmatrix} -(2\Omega_N + \rho_N \sec^2 L + \dfrac{\rho_D R_{tt}}{R_t + h}) V_E + \dfrac{\rho_E R_{mm}}{R_m + h} V_D & 0 & -\dfrac{\rho_D V_E}{R_t + h} + \dfrac{\rho_E V_D}{R_m + h} \\[4mm] (2\Omega_N + \rho_N \sec^2 L + \dfrac{\rho_D R_{tt}}{R_t + h}) V_N + (2\Omega_D + \dfrac{\rho_N R_{tt}}{R_t + h}) V_D & 0 & \dfrac{\rho_D V_N}{R_t + h} - \dfrac{\rho_N V_D}{R_t + h} \\[4mm] -\dfrac{\rho_E R_{mm}}{R_m + h} V_N - (2\Omega_D + \dfrac{\rho_N R_{tt}}{R_t + h}) V_E & 0 & -\dfrac{\rho_E V_N}{R_m + h} + \dfrac{\rho_N V_E}{R_t + h} \end{bmatrix}
$$

$$
F_{22} = \begin{bmatrix} \dfrac{V_D}{R_m + h} & 2\Omega_D + 2\rho_D & -\rho_E \\[4mm] -2\Omega_D - \rho_D & \dfrac{V_N \tan L + V_D}{R_t + h} & 2\Omega_N + \rho_N \\[4mm] 2\rho_E & -2\Omega_N - 2\rho_N & 0 \end{bmatrix}
$$

$$
F_{23} = \begin{bmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{bmatrix}
$$

$$
F_{31} = \begin{bmatrix} \Omega_D - \dfrac{\rho_N R_{tt}}{R_t + h} & 0 & -\dfrac{\rho_N}{R_t + h} \\[4mm] -\dfrac{\rho_E R_{mm}}{R_m + h} & 0 & -\dfrac{\rho_E}{R_m + h} \\[4mm] \Omega_N - \rho_N \sec^2 L - \dfrac{\rho_D R_{tt}}{R_t + h} & 0 & -\dfrac{\rho_D}{R_t + h} \end{bmatrix}
$$

$$F_{32} = \begin{bmatrix} 0 & \dfrac{1}{R_t + h} & 0 \\[2mm] -\dfrac{1}{R_m + h} & 0 & 0 \\[2mm] 0 & -\dfrac{\tan L}{R_t + h} & 0 \end{bmatrix}$$

$$F_{33} = -\begin{bmatrix} \Omega_N + \rho_N \\ \rho_E \\ \Omega_D + \rho_D \end{bmatrix} \times$$

8. References

Altmann, S.L. (1986). *Rotations, Quaternions, and Double Groups,* Clarendon press, ISBN 0-19-855372-2, New York

Belanger, P.R. (1974). Estimation of noise covariance matrices for a linear time varying stochastic process, *Automatica*, Vol. 10, pp.267-275, ISSN 0005-1098

Bierman, G.J. (1975). Fixed-memory least squares filtering. *IEEE transactions on Information Theory*, Vol. IT-21, No. 6, pp. 690-692, ISSN 0018-9448

Farrell, J.A & Barth, M. (1999). *The Grobal Positioning System & Inertial Navigation,* McGraw-Hill, ISBN 0-07-116269-0, New York

Fitzgerald, R.J. (1971). Divergence of the Kalman Filter, *IEEE transactions on Automatic Control*, Vol. 16, No. 6, ISSN 0018-9286

Friedland, B. (1978). Analysis strapdown navigation using quaternions. *IEEE transactions on Aerospace and Electronic Systems,* Vol. AES-14, No.5, pp. 764-768, ISSN 0018-9251/78/0900-0764

Jazwinski, A.H. (1968). Limited memory optimal filtering. *IEEE transactions on Automatic Control*, Vol. AC-13, No. 5, pp. 558-563, ISSN 0018-9286

Jo, G. & Choi, H.S. (2006). Velocity-Aided Underwater Navigation System Using Receding Horizon Kalman Filter. *IEEE Journal of Oceanic Engineering*, Vol. 31, No. 3, pp.565-573, ISSN 0364-9059

Jo, G.; Seo, D.C. & Choi, H.S. (submitted-a). An Adaptive Receding Horizon Kalman Filter for underwater navigation systems – part I: estimation method of noise covariance. *IEEE Journal of Oceanic Engineering*, ISSN 0364-9059

Jo, G.; Seo, D.C. & Choi, H.S. (submitted-b). An Adaptive Receding Horizon Kalman Filter for underwater navigation systems – part II: ARHKF and applications. *IEEE Journal of Oceanic Engineering*, ISSN 0364-9059

Kwon, O.K.; Kwon, W.H. & Lee, K.S. (1989). FIR filters and recursive forms for discrete-time state-space models. *Automatica*, Vol. 25, No. 5, pp. 715-728, ISSN 0005-1098

Kwon, W.H. & Pearson, A.E. (1978). On feedback stabilization of time varying discrete linear system. *IEEE transactions on Automatic Control*, Vol. 23, No. 3, pp. 479-481, ISSN 0018-9286

Kwon, W.H.; Kim, P.S. & Park, P. (1999). A Receding Horizon Kalman FIR Filter for Discrete Time-Invariant Systems. *IEEE transactions on Automatic Control*, Vol. 44, No. 9, pp. 1787-1791, ISSN 0018-9286

Larsen, M.B. (2000). Synthetic Long Baseline Navigation of Underwater Vehicles, *Proceedings of Oceans 2000*, pp. 2043-2050, ISBN 0-7803-6551-8/00, Rhode Island USA, Sep., 2000, IEEE

Lee, J.G.; Park, C.G & Park, H.W. (1993). Multiposition Alignment of Strapdown Ineritial Navigation System. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 4, pp.1323-1328, ISSN 0018-9251

Lee, P.M.; Jeon, B.H., Choi, H.T., Lee, C.M., Aoki, T. & Hyakudome, T. (2004). An Integrated Navigation System for Autonomous Underwater Vehicles with Two Rage Sonars, Inertial Sensors and Doppler Velocity Log, *Proceedings of Oceans Techno-Oceans 2004*, pp. 1586-1593, ISBN 0-7803-8669-8/04, Tokyo, Nov., 2004, IEEE

Marco, D.B. & Healey, A.J. (2001). Command, control and navigation experimental results with the NPS ARIES AUV. *IEEE Journal of Oceanic Engineering*, Vol. 26, No. 4, pp. 466-476, ISSN 0364-9059

Mehra, R.K. (1970). On the Identification of Variances and Adaptive Kalman Filtering, *IEEE transactions on Automatic Control*, Vol. 15, No. 2, pp.175-184, ISSN 0018-9286

Sangsuk-Iam, S. & Bullock, T.E. (1990). Analysis of Discrete-time Kalman filtering under Incorrect Noise covariance, *IEEE transactions on Automatic Control*, Vol. 35, No. 12, pp.1304-1309, ISSN 0018-9286

Seo, J.W.; Yu, M.J., Park, C.G. & Lee, J.G. (2006). An extended robust h-infinite filter for nonlinear constrained uncertain systems. *IEEE transactions on Signal Processing*, Vol.54, No.11, pp. 4471-4475, ISSN 1053-587X

Shibata, M. (1986). Error analysis strapdown inertial navigation using quaternions. *IEEE Journal of Guidance and Control*, Vol.9, No.3, pp. 379-381, ISSN 0018-9251

Siouris, G.M. (1993). *Aerospace Avionics Systems: A Modern Synthesis*, Academic Press, ISBN 0-12-646890-7, New York

Titterton, D.H. & Weston, J.L. (1997). *Strapdown Inertial Navigation Technology*, Peter Peregrinus Ltd., ISBN 0-86341-260-2, London

Um, T.Y.; Lee, J.G., Park, S.T. & Park, C.G. (2000). Noise covariances estimation for systems with bias states, *IEEE transactions on Aerospace and Electronic Systems*, Vol. 36, No. 1, pp.226-233 ISSN 0018-9251

Whitcomb, L.L. (2000). Underwater robotics: Out of the research laboratory and into the field, *Proceedings of IEEE International Conf. on Robotics and Automation 2000*, pp. 709-716, ISBN 0-7803-5886-4/00, San Francisco, Apr., 2000, IEEE

Yu, M.J.; Lee, J.G & Park, C.G. (2004). Nonlinear robust observer design for strapdown INS in-flight alignment. *IEEE transactions on Aerospace and Electronic Systems*, Vol. 40, No. 3, pp. 797-807, ISSN 0018-9251

Tracking Relative Errors in Internet Coordinate Systems by a Kalman Filter

M.A. Kaafar[1], L. Mathy[2], K. Salamatian[2], C. Barakat[3],
T. Turletti[3] and W. Dabbous[3]

[1]University of Liege,
[2]Lancaster University,
[3]INRIA Sophia-Antipolis,
[1]Belgium
[2]UK
[3]France

1. Introduction

Internet Coordinate Systems, shortly ICS, (e.g. [9] [8]) have been proposed to allow for distance (Round-Trip Time, shortly RTT) estimation between nodes, in order to reduce the measurement overhead of many applications and overlay networks. Indeed, by embedding the Internet delay space into a metric space – an operation that only requires each node in the system to measure delays to a small set of other nodes (its neighbors), nodes are attributed coordinates that can then be used to estimate the RTT between any two nodes, without further measurements, simply by applying the distance function associated with the chosen metric space to the nodes' coordinates.

Recent works have shown how coordinate-embedding services could be vulnerable to malicious attacks, providing a potentially attractive fertile ground for the disruption or collapse of the many applications and overlays that would use these services [2]. There are actually two obvious ways to disrupt the operation of a coordinate based system. First when requested to give its coordinate for a distance estimation at the application-level, a malicious node could simply and blatantly lie. Second, a malicious node, or even a colluding group, may aim at disrupting the embedding process itself. This latter strategy is very insidious and effective as it can result in important distortions of the coordinate space which then spoils the coordinate computations of many nodes (malicious and honest alike) [2]. This chapter focuses on developing and studying generic Kalman filter-based methods to secure the coordinate embedding process. More precisely, the embedding process, regardless of the actual coordinate-based positioning system, works on the premise that nodes adjust their coordinate based on some comparison between measured and estimated distances to some other nodes. Malicious nodes can interfere with this embedding process by, amongst other things, lying about their real coordinate and/or tampering with measurement probes, to create a discrepancy between measured and estimated latencies, so that unsuspecting nodes would wrongly adjust their own coordinate in a bid to reduce the difference [1]. Because the load on the network naturally varies in time, so does latency between pair of nodes, and as a

result, the embedding process must be run periodically by all nodes to track changes in network conditions. This "continuous" adjustment of nodes' coordinates can not only result in a drift of the coordinate space [10] but also gives plenty of scope and opportunities for malicious activity. We therefore seek to equip (honest) nodes with a means to detect, with low overhead, malicious activities they may encounter during embedding.

Noting that, in the absence of malicious nodes, a node's coordinate depends on the combination of network conditions and the specificities of the embedding process itself (e.g. which coordinate protocol is in use, the chosen dimensionality of the geometric space, etc), we therefore introduce the concept of *Surveyor nodes* (or *Surveyors* in short). Surveyors form a group of trusted (honest) nodes, scattered across the network, which use each other *exclusively* to position themselves in the coordinate space. Of course, Surveyors do assist other nodes in their positioning (as prescribed by the embedding protocol), but we stress that Surveyors never rely on non-Surveyor nodes to compute their own coordinate. This strategy thus allows Surveyors to experience and learn the natural evolution of the coordinate space, as observed by the evolution of their own coordinate, in the absence of malicious activities. In essence, Surveyor nodes are thus vintage points guaranteed to be immune from malicious activities. The idea is that Surveyors can then share a "representation" of normal behavior in the system with other nodes to enable them to detect and filter out abnormal behavior.

We postulate and verify that, in the absence of malicious activity, a node's coordinate can be viewed as a stochastic process with linear dependencies whose evolution can be tracked by a Kalman filter [4]. Each Surveyor then computes and calibrates the parameters of a linear state space model and shares the parameters of this model with other nodes. These nodes can then use these parameters, to run locally and in a "standalone" fashion a Kalman filter tracking the coordinate adjustments. These nodes can then use the Kalman filter output (the innovation process), to compare their observed coordinate adjustments with the one predicted by the Kalman filter, and flag as "suspicious" embedding steps where the difference would be too high.

In section 2, we present a general model of coordinate embedding, in the absence of malicious nodes, that naturally leads to the Kalman filter framework. In section 3, we validate the model, with both simulations and PlanetLab experiments, in the case of both Vivaldi [9] and NPS [8]. This section also studies the viability of the idea of using Surveyor nodes in secure coordinate embedding. We then describe and evaluate, in sections 4 and 5, how Surveyors can effectively be used for malicious node detection in the specific embedding process of Vivaldi and NPS.

2. Coordinate embedding model

The goal of embedding systems, regardless of the embedding method and geometric space used, is to assign a coordinate to every node in the system so that, at any time, the distance between any two points in the geometric space should provide a good estimate of the network distance, measured as an RTT (Round Trip Time), between the corresponding nodes. Obviously, because at any instant in time, the RTT that can be measured between two nodes depends on the state of the network (e.g. traffic load, state of queues in routers, etc) as well as the state of the operating system in nodes (e.g. scheduling state generating measurement noise, etc), the exact value of the RTT varies continuously. However, it has been shown that RTT values in the Internet exhibit some stability in a statistical sense [14],

with the statistical properties of RTTs exhibiting no significant change at timescales of several minutes. It is that property that embedding systems exploit to provide good distance estimates while only needing to have nodes adjust (recalculate) their coordinate on a periodic basis. Consequently, the coordinate of a node can be viewed as a discrete stochastic process, and we will use X_i^n to represent the coordinate of node i at "discrete time" n.

Without loss of generality, consider that a node (called the embedding node) computes its coordinate through a series of embedding steps, where each embedding step represents a coordinate adjustment based on a one-to-one interaction with another node, called a peer node (e.g. peer nodes are called neighbors in Vivaldi, and landmarks or reference points in NPS). Note that when the embedding protocol requires that a node uses several peer nodes simultaneously for repositioning, for the purpose of our modelization, we simply consider that each peer node corresponds to a distinct embedding step, each taking place at "successive" discrete times.

At every embedding step, the "fitness" (or "correctness") of the embedding node coordinate is assessed by computing the deviation between the measured RTT towards the corresponding peer node and the one estimated in the coordinate system. More precisely, suppose that at its nth embedding step, embedding node i has current coordinate X_i^n and uses peer node j with current coordinate X_j^n. Suppose that the RTT between these nodes, measured during this embedding step, is RTT_{ij}^n. The fitness of the embedding node coordinate can then be computed as the *measured relative error* $D_n = \frac{|\,||X_i^n - X_j^n|| - RTT_{ij}^n\,|}{RTT_{ij}^n}$. The goal of *any* embedding system, regardless of the embedding method proposed and/or the geometric space structure, is to minimize a "cost" indicator (e.g. mean square error) that captures the measured relative error that could be observed between any node and any other node in the system, at any time.

As the measured relative errors are fundamental performance indicators to all embedding systems, it seems natural to develop a model that captures their dynamic characteristics, although we note that relative errors often have complex behavior (and may thus not be a natural choice from a modeling perspective).

Measured relative errors are subject to fluctuations of the RTT for the reasons mentioned above, namely transient network congestion and operating system scheduling issues. To isolate the impact of these RTT fluctuations on anomaly detection, we introduce Δ_n, the *nominal relative error* that our node under consideration would have obtained at its nth embedding step if the RTTs in the network had not fluctuated. An anomaly becomes simply a large deviation of measured relative error D_n from its nominal value defined by Δ_n.

Because many sources contribute to the deviation of D_n from its nominal value (RTT measurement error, RTT fluctuations, errors in node coordinate), it is reasonable to suppose that they relate to each other as follows,

$$D_n = \Delta_n + U_n \qquad (1)$$

where U_n is a Gaussian random variable with mean zero and variance v_U. We now focus on the dynamics of the system in its nominal regime where RTTs do not fluctuate. In the absence of complete and accurate knowledge of the system, nodes keep on adapting the nominal relative error on a pairwise basis with their peer nodes, aiming to optimize the cost indicator. This adaptation is subject to an error caused by the other nodes in the system adapting their coordinate (and corresponding relative error) in a completely distributed

way. We thus define the *system error* W_n which represents the impact of other nodes on the positioning of a node at embedding step n. Since the system error at a node results from many contributing sources, it is also reasonable to assume that it is a white gaussian process (with mean $\bar{w}$ and variance v_W)[1].

Because of the nature of large-scale embedding processes, the nominal relative error Δ_n can be deemed to follow a stochastic process that converges to some stationary regime characterized by a positive average. As a first approximation, the process Δ_n could be modeled as a first order Auto Regressive (AR) model:

$$\Delta_{n+1} = \beta\Delta_n + W_n. \tag{2}$$

where β is a constant factor strictly less than one otherwise the relative error does not converge to a stationary regime independently of the initial condition. This equation captures the dynamic evolution of the nominal relative error of a node through successive embedding steps.

Equations 2 and 1 define a linear state space model for the relative error of a node. Our goal is to devise a way to obtain relative error predictions from this model. Because of the linear properties of the model, a Kalman filter can be used to track the evolution of the nominal relative error and obtain a *predicted relative error* $\hat{\Delta}_{n|n-1}$ (see section 2.1). However, it is important to notice at this level that tracking relative errors using the Kaman filter would be relevant, only if we consider the embedding systems at their permanent regime. A permanent regime is typically reached when the average relative errors of nodes involved in the system stabilize enough, so that coordinates are considered to be usable by nodes.

The idea behind the strategy we mention above, is that if the stochastic space model, and especially its associated Kalman filter, are calibrated within a clean embedding system, then a simple hypothesis test can be used to assess whether the deviation between the measured relative error and the predicted relative error, observed at a given embedding step, is normal or is the fruit of anomalous or malicious activity from the peer node. From this perspective, even if the state space model considered is crude, its quality should be evaluated based on the final outcome in terms of probability of detection and false positive rate. We will see in the evaluation section (section 5) that this model achieves very good performance.

2.1 Kalman filter equations

The Kalman filter is used here to estimate Δ_n given the set of previously measured relative errors $\mathcal{D}_0^n = \{D_0, \ldots, D_n\}$. Under the hypothesis of a gaussian noise process in the underlying state space model, the Kalman filter gives the Least Mean Squared estimates of Δ_n, $\hat{\Delta}_n$. Moreover, it gives the quality of these estimates through an evaluation of the mean squared error i.e., $E[(\hat{\Delta}_n - \Delta_n)^2]$. This last value could be used to detect anomalies through large deviations of the measured relative error from its mean.

We will assume here that all the parameters of the space model given in Eq. (1) and Eq. (2) are known and given. In the next section we will describe how to derive these parameters.

Let us denote by $\hat{\Delta}_{i|i-1}$ the estimation of Φ_i knowing the observations of network delay up to time $i-1$, and $\hat{\Delta}_{i|i}$ the estimate after the measurement D_i is done. Similarly, let $P_{i|i-1}$ be the

---

[1] The value $\bar{w}$ accounts for the drift that has been observed in positioning systems [10].

estimated *a posteriori* error variance at time i knowing the observations up to time $i-1$, and let $P_{i|i}$ be the estimation of the *a posteriori* error variance after D_i is known. The Kalman Filter is composed of two steps that are iterated. The first step is called the prediction step and the second one the update step.

In the prediction step, the value of $\hat{\Delta}_{i|i-1}$ is calculated based on $\hat{\Delta}_{i-1|i-1}$ as:

$$\hat{\Delta}_{i|i-1} = \beta \hat{\Delta}_{i-1|i-1} + \bar{w}.$$

The *a posteriori* error variance of this estimate is:

$$P_{i|i-1} = \beta^2 P_{i-1|i-1} + v_W.$$

In the update step, $\hat{\Delta}_{i|i-1}$ is updated to integrate the observed measurement D_i:

$$\hat{\Delta}_{i|i} = \hat{\Delta}_{i|i-1} + K_i(D_i - \hat{\Delta}_{i|i-1})$$

where K_i denotes the updated gain and is obtained as:

$$K_i = \frac{P_{i|i-1}}{P_{i|i-1} + v_U}.$$

The *a posteriori* error variance of this estimate is :

$$P_{i|i} = \frac{v_U}{P_{i|i-1} + v_U} P_{i|i-1}.$$

The value $\eta_i = D_i - \hat{\Delta}_{i|i-1}$ is called the innovation process and is the main process to observe for anomalous behavior detection (see section 4.1). The innovation process is a white (meaning that it is an independent process) gaussian process with a mean 0 and a variance equal to $v_{\eta,i} = v_U + P_{i|i-1}$. Abnormality simply amounts to a significant deviation from the nominal values of the innovation process characterized by the Kalman filter.

To run the Kalman estimation, we need as initial values the system state value w_0 and the *a priori* state variance $P_{0|0} = p_0$. These two values are estimated during the parameters calibration step.

2.2 Calibration of the Kalman filter

Before running the estimation using the Kalman filter, the values of the filter parameters $\theta = (\beta, v_W, v_U, w_0, p_0)$ have to be computed. For this purpose we need to calibrate these parameters over coordinate measurements collected during a stationary and cheater-free period. The calibration can be done using a maximum likelihood criteria (choosing parameter values such that the likelihood of observing the measurements is maximized) by applying the Expectation Maximization (EM) method. We follow the approach presented in [15] for the EM derivation.

In the following, we give a brief description of the maximum likelihood estimation criterion and the EM method, we are using in this section to calibrate our filter.

The maximum likelihood estimation criterion Maximum likelihood estimation (MLE) is a statistical method used to make inferences about parameters of the underlying probability distribution from a given data set [16]. Basically, this method allows us to infer the parameters of a distribution given a sample of data $X = X_1, \ldots, X_n$. Commonly, one assumes

the data are independent, identically distributed (i.i.d) drawn from a particular distribution with unknown parameters and uses the MLE technique to create estimators for these unknown parameters.

Let us consider a family of distributions P_θ indexed by a parameter (which could be a vector of parameters) θ that belongs to a set Θ. Let $f(X|\theta)$ be either a probability function (in case of discrete distribution) or a probability density function (continuous case) of the distribution P_θ. Given our i.i.d. sample $X_1, \ldots, X_n$ with unknown distribution P_θ from this family, i.e. parameter θ is unknown. A likelihood function is defined by:

$$L(\theta|X) = f(X_1|\theta) \times \ldots \times f(X_n|\theta).$$

If our distributions are discrete then the probability function $f(x|\theta) = P_\theta(X = x)$ is the probability to observe a point x. $L(\theta) = f(X_1|\theta) \times \ldots \times f(X_n|\theta) = P_\theta(X_1) \times \ldots P_\theta(X_n) = P_\theta(X_1, \ldots, X_n)$ is the probability to observe the sample $X_1, \ldots, X_n$ when the parameters of the distribution are equal to θ.

Suppose that there exists a parameter $\hat{\theta}$ that maximizes the likelihood function $L(\theta)$ on the set of possible parameters, i.e.

$$L(\hat{\theta}) = \max_{\theta \in \Theta} L(\theta)$$

Then $\hat{\theta}$ is called the Maximum Likelihood Estimator (MLE) of θ.

When finding the MLE, it is sometimes easier to maximize the log-likelihood function since

$$L(\theta) \to maximize \Leftrightarrow log(L(\theta)) \to maximize$$

maximizing $L(\theta)$ is equivalent maximizing $\log L(\theta)$. Log-Likelihood function can be written as $\log L(\theta) = \sum_{i=1}^{n} (\log f(X_i|\theta))$.

To summarize, one needs to recall that the MLE criterion chooses the parameter $\hat{\theta}$ that maximizes the probability of seeing the observed data given that their distribution follows these parameters. The log of the likelihood is often used instead of true likelihood because it leads to easier formulas, but still attains its maximum at the same point as the likelihood.

The Expectation maximization (EM) algorithm is a valuable approach for maximum likelihood parameter estimation. In the next paragraph, we use this method to calibrate our Kalman filter parameters, i.e . find the values for the filter parameters that maximize the probability of a sequence of observations.

Calibration by EM method Let's assume that $\mathcal{D}_0^N$ is the set of all measured prediction errors, $\mathcal{D}_0^N = \{D_0, \ldots, D_N\}$ and let $\mathbf{\Delta}_0^N = \{\Delta_0, \ldots, \Delta_N\}$ be the set of nominal relative errors.

As all the noise processes are assumed to be gaussian, $\mathcal{D}_0^N$ and $\mathbf{\Delta}_0^N$ will jointly follow a gaussian distribution. The log-likelihood of $\mathcal{D}_0^N$ and $\mathbf{\Delta}_0^N$, based on equations 2 and 1, can therefore be written as follows:

$$\log \mathrm{Prob}\{\mathcal{D}_0^N, \mathbf{\Delta}_0^N\} = -\sum_{i=0}^{N} \frac{(D_i - \Delta_i)^2}{2v_U}$$

$$-\frac{N+1}{2} \log v_U - \sum_{i=1}^{N} \frac{(\Delta_i - \beta\Delta_{i-1} - \bar{w})^2}{2v_W}$$

$$-\frac{N}{2} \log v_W - \frac{(\Delta_0 - w_0)^2}{2p_0} - \frac{1}{2} \log p_0$$

$$-(N+1) \log 2\pi.$$

In a Maximum likelihood setting, we wish to find the values for the parameters that will maximize the above log-likelihood assuming that the sequence $\mathcal{D}_0^N$ has been observed. However as the sequence of system state $\boldsymbol{\Delta}_0^N$ has not been observed, this maximization is not tractable directly and we have to apply the Expectation Maximization method [17]. This method transforms the maximization of the above likelihood function with unobserved system state sequence $\boldsymbol{\Delta}_0^N$ to an iteration of successive steps where the system state sequence is assumed to be known and the parameters can be obtained through maximization of the likelihood function.

Each iteration of the EM method consists therefore of two steps. In the first step, we compute the expectation (over all values of the sequence of states $\boldsymbol{\Delta}_0^N$) of the loglikelihood, given the observed values of D_n and assume that the parameter values are equal to $\theta^{(k)}$. In a second step, the parameters $\theta^{(k+1)}$ are chosen so as to maximize the previously obtained likelihood expectation. Next we explain these two operations with some further details.

Let the superscript (k) indicate the value of any parameter at the kth step of the EM algorithm. As explained before, in the EM method, we need to estimate the value of the unobserved system states to be able to calculate the overall likelihood to maximize. The variables $\hat{\delta}_i^{(k)}$ are in fact those estimates at iteration k and $\hat{\pi}_i^{(k)}$ and $\hat{\pi}_{i,i-1}^{(k)}$ are the estimation error variances of this sequence of states:

$$\hat{\delta}_i^{(k)} = E[\Delta_i | \mathcal{D}_0^N, \theta^{(k)}], \ \hat{\pi}_i^{(k)} = E[\Delta_i^2 | \mathcal{D}_0^N, \theta^{(k)}],$$

$$\hat{\pi}_{i,i-1}^{(k)} = E[\Delta_i \Delta_{i-1} | \mathcal{D}_0^N, \theta^{(k)}].$$

Expectation step The expected value of log-likelihood knowing the set of measured values $\mathcal{D}_0^N$ and the parameter $\theta^{(k)}$ is given by:

$$
\begin{aligned}
\bar{L}(\theta, \theta^{(k)}) &= E[\log \operatorname{Prob}\{\mathcal{D}_0^N, \boldsymbol{\Delta}_0^N\} | \mathcal{D}_0^N, \theta^{(k)}] \\
&= -\sum_{i=0}^N \frac{D_i^2 - 2D_i \hat{\delta}_i^{(k)} + \hat{\pi}_i^{(k)}}{2v_U} \\
&\quad - \frac{N+1}{2} \log v_U - \sum_{i=1}^N \frac{\hat{\pi}_i^{(k)} + \beta^2 \hat{\pi}_{i-1}^{(k)} + \bar{w}^2}{2v_W} \\
&\quad + \sum_{i=1}^N \frac{\beta \hat{\pi}_{i,i-1}^{(k)} + \hat{\delta}_i^{(k)} \bar{w} - \beta \hat{\delta}_{i-1}^{(k)} \bar{w}}{v_W} \\
&\quad - \frac{N}{2} \log v_W - \frac{\pi_0 - 2\hat{\delta}_0^{(k)} w_0 + w_0^2}{2p_0} \\
&\quad - \frac{1}{2} \log p_0 - (N+1) \log 2\pi.
\end{aligned}
$$

By replacing θ by its value at the kth step of the EM algorithm, we obtain $\hat{\delta}_i^{(k)}$, $\hat{\pi}_i^{(k)}$ and $\hat{\pi}_{i,i-1}^{(k)}$, which gives the expected log-likelihood at the $(k+1)^{th}$ step. Next, we describe how to compute these values.

Calculating the parameters $\hat{\delta}_i$, $\hat{\pi}_i$, $\hat{\pi}_{i,i-1}$ As explained in section 2.2, the sequence of system states $\boldsymbol{\Delta}_0^N$ is not observable. However, we need to give an estimate of this sequence to be able to obtain the likelihood. The sequence $\hat{\delta}_i$, $i = 1 \ldots, N$, is the sequence of system state estimates and $\hat{\pi}_i$, and $\hat{\pi}_{i,i-1}$ is the error variance of these estimates assuming that the sequence $\mathcal{D}_0^N$ has been observed. We resort to the solution in [15] for the calculation of these estimates using the overall measurements set.

The value $\hat{\delta}_i$, $\hat{\pi}_i$, and $\hat{\pi}_{i,i-1}$ are estimated using a Kalman filter, assuming that the system parameters are set as in $\theta_{(k)}$. However, there is a subtle difference with Kalman filter case described in section 2.1; here the estimates $\hat{\delta}_i$, $\hat{\pi}_i$ and $\hat{\pi}_{i,i-1}$ do not depend only on observations up to time i, but on future observations up to time $N \geq i$. The solution to deal with this is to implement a forward-backward approach similar to Baum- Welch filter used for finite EM algorithm [20].

For each value of the parameter set $\theta^{(k)}$, we first do a forward step following the relations given in section 2.1. The application of this forward step results in the values $\hat{\Delta}_{i|i}$, $i = 1, \ldots, N$, $P_{i|i}$, $i = 1, \ldots, N$ and $P_{i|i-1}$, $i = 1, \ldots, N$.

To add the future measurements in the Kalman filter, a backward recursion step is also added. This step consists of the following equations:

$$\begin{cases} \hat{\delta}_{i-1} = \hat{\Delta}_{i-1|i-1} + J_{i-1}(\hat{\delta}_i - \beta\hat{\Delta}_{i-1|i-1}) \\ \hat{\pi}_{i-1} = P_{i-1|i-1} + J_{i-1}^2(\hat{\pi}_i - P_{i|i-1}) \\ J_{i-1} = \beta\frac{P_{i-1|i-1}}{P_{i|i-1}} \end{cases}$$

These equations give recursively the values $\hat{\delta}_i$ and $\hat{\pi}_i$. It still remains to obtain $\hat{\pi}_{i,i-1}$. This last value could be obtained using the relation:

$$\hat{\pi}_{i,i-1} = v_{i,i-1}^N + \hat{\delta}_i\hat{\delta}_{i-1}$$

where $v_{i,i-1}^N$ can be obtained through the backward recursion

$$v_{i,i-1}^N = P_{i-1|i-1}J_{i-1} + J_i(v_{i+1,i}^N - \beta P_{i|i})J_{i-1},$$

that is initialized by setting

$$v_{N,N-1}^N = (1 - K_N)\beta P_{N-1|N-1}.$$

Maximization step In this step, the parameter vector at step $(k + 1)$ is chosen to maximize the expected log-likelihood. This is done by solving the equation

$$\frac{\partial \bar{L}(\theta, \theta^{(k)})}{\partial \theta} = 0.$$

This results in the following set of equations:

$$
\begin{cases}
w_0^{(k+1)} = \hat{\delta}_0^{(k)} \\
p_0^{(k+1)} = \hat{\pi}_0^{(k)} - (\hat{\delta}_0^{(k)})^2 \\
v_U^{(k+1)} = \frac{1}{N+1} \sum_{i=0}^N D_i^2 - 2D_i \hat{\delta}_i^{(k)} + \hat{\pi}_i^{(k)} \\
\bar{w}^{(k+1)} = \sum_{i=1}^N \hat{\delta}_i^{(k)} - \beta^{(k+1)} \sum_{i=1}^N \hat{\delta}_{i-1}^{(k)} \\
\beta^{(k+1)} = \frac{\sum_{i=1}^N \hat{\pi}_{i,i-1}^{(k)} - \bar{w}^{(k+1)} \sum_{i=1}^N \hat{\delta}_{i-1}^{(k)}}{\sum_{i=1}^N \hat{\pi}_{i-1}^{(k)}} \\
v_W^{(k+1)} = \frac{1}{N} \sum_{i=1}^N \hat{\pi}_i^{(k)} + (\beta^{(k+1)})^2 \hat{\pi}_{i-1}^{(k)} + (\bar{w}^{(k+1)})^2 \\
\quad - 2\beta^{(k+1)} \hat{\pi}_{i,i-1}^{(k)} - 2\hat{\delta}_i^{(k)} \bar{w}^{(k+1)} + 2\beta^{(k+1)} \hat{\delta}_{i-1}^{(k)} \bar{w}^{(k+1)}.
\end{cases}
$$

By solving this set of equations we can obtain the vector $\theta^{(k+1)}$, then we iterate with the expectation calculation as described above.

We note that the complexity of the approach lies in the linear state space modeling phase by EM algorithm that incurs a number of iterations over N dimensional vectors, which is well within the capability of modern computers. We will see later that this phase has to be run on a subset of nodes (the Surveyors). On the other hand, predicting relative errors using the Kalman filter(section 2.1), which occurs on every node, only implies a few simple scalar operations and is negligible in terms of required computing power.

Finally, because we expect each of the innovation observation η_n to be inside a confidence interval of $\pm 2\sqrt{v_{\eta,n}}$ (where $v_{\eta,n}$ is the variance of the innovation process at time n) with a probability higher than 95%, when a Kalman filter yields 10 consecutive innovation observations outside such confidence interval, the filter is re-calibrated by re-applying the calibration procedure described in this section. Re-calibration is likely to occur following a significant change in the corresponding node's coordinate, caused by changes in network conditions.

3. Validation

To validate our model, we conducted simulations and PlanetLab experiments for both Vivaldi [9] and NPS [8]. We consider Vivaldi as a prominent representative of purely peer-to-peer-based (i.e. without infrastructure support) positioning systems, while NPS is typical of infrastructure-based systems, where a hierarchy of landmarks and reference points govern the positioning of nodes.

As the goal of this section is to assess the fitness of the proposed model to represent the normal behavior of the embedding processes, all results presented were acquired in a clean environment with no malicious node. While the goal of the simulation studies is to assess our results for large scale coordinate-based systems, the PlanetLab experiments aim to show their applicability in real-world conditions. The PlanetLab experiments were conducted over a set of 280 PlanetLab nodes spread world-wide. We discuss a representative set of experimental results conducted over several days in December 2006. The simulations were driven by a matrix of inter-host Internet RTTs (the "King" dataset) to model latencies based on real world measurements. This dataset contains the pair-wise RTTs between 1740 Internet DNS servers collected using the King method [13] and was used to generate a topology with 1740 overlay nodes.

In the case of Vivaldi, each node had 64 neighbors (i.e. was attached to 64 springs), 32 of which being chosen to be closer than 50 ms. The constant fraction C_c for the adaptive

timestep is set to 0.25, as proposed in [9]. A 2-dimensional coordinate space augmented with a height vector was used.

For NPS, we considered an 8-dimensional Euclidean space for the embedding. We used an NPS positioning hierarchy with 4 layers. The top layer had a set of 20 well separated permanent landmarks. Each subsequent layer then had 20% of nodes randomly chosen as reference points. The security mechanism already proposed in NPS, shown to be too primitive in [2], was turned on and its sensitivity constant C was set to 4, as advised by authors in [8].

When needed, Surveyor nodes were chosen at random[2]. Again, because we seek to validate the kalman filter model, during the permanent regime of the embedding systems, all our experiments were are performed when nodes coordinates are stabilized enough, i.e. the relative errors vary at most by 0.02.

3.1 Assumption validation

In section 2, the assumption that the system error W_n follows a gaussian distribution was made. This is fundamental to the applicability of the Kalman filter framework. For the purpose of validation, every node calibrated its own Kalman filter based on the observation of its own embedding, and we checked this assumption by applying the Lilliefors test [18], a robust version of the well known kolmogoroff-Smirnov goodnessof- fit test, to whitened filter inputs. We observed that the Lilliefors test leads to only 14 gaussian fitting rejections in simulations (over 1720 samples) and 5 rejections in PlanetLab (over 260 samples). This test allows us to conclude that the hypothesis we took for the Kalmanmodel is valid. In addition, we plot in figure 1 the Quantile-Quantile (QQ) plots of 2 innovation processes (for both Vivaldi and NPS) taken on PlanetLab nodes running their own Kalman filter. These plots, typical of observations on all nodes, show that each of these distributions indeed closely follows a Gaussian distribution.
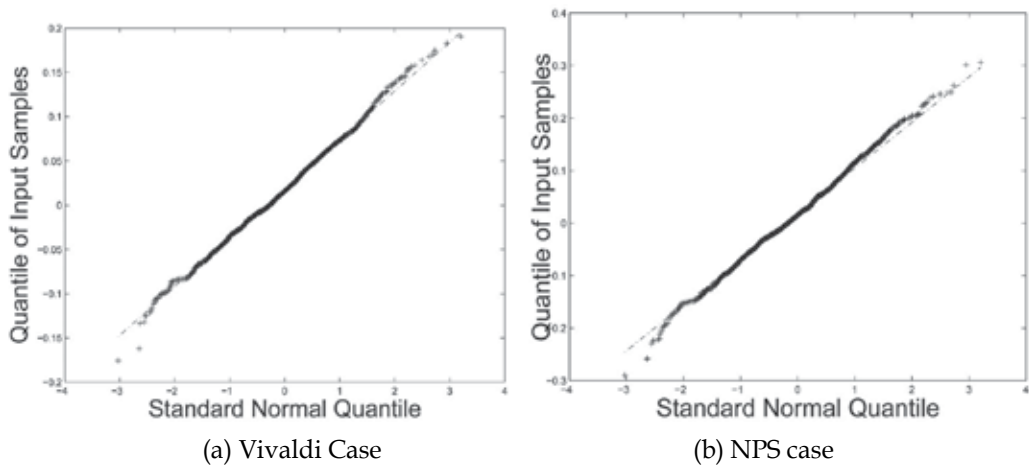

(a) Vivaldi Case   (b) NPS case

Fig. 1. Quantile-Quantile plot of 2 innovation processes.

---

[2] Note that in NPS, all permanent landmarks also act as Surveyors.

3.2 Effective behavior representation

From section 2, it is clear that the Kalman filter model attempts to represent the behavior of the embedding process by capturing the dynamics of the system through its convergence behavior (by tracking of relative errors over time). In this section, we therefore assess the representational power of this approach by having each node calibrate its own Kalman filter from the measurements it observed during the embedding of its own coordinate, in a cheat-free regime. Then, once the model has converged at every node (i.e. the EM method has converged and the variations of all the θ components become smaller than 0.02), a new embedding process is started (i.e. the nodes forget their coordinates and rejoin the system). During this second embedding process, the prediction error, that is the absolute value between the error predicted by the node's Kalman filter and the measured actual error, is computed.

Figure 2 shows a typical evolution of actual (measured) relative errors and predicted errors for a node on PlanetLab (for Vivaldi, but similar behavior was observed for NPS). The two curves of the top graph of the figure are so similar that they are almost indistinguishable. The bottom graph of the figure represents the prediction error which is the difference between these two curves (note the different scale used for this graph). We see that the prediction errors are small which shows that a node's calibrated Kalman filter can capture effectively the node's behavior "in the wild".
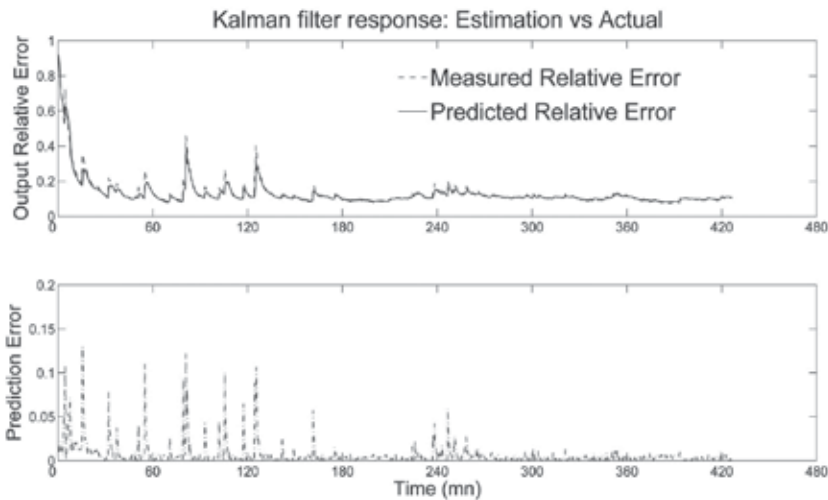
Fig. 2. Prediction errors (PlanetLab node).

Figure 3 shows the cumulative distribution function (CDF) of all the prediction errors observed across all nodes in the system. This confirms that the vast majority of predictions are indeed excellent. This demonstrates the power and generality of the model in capturing the dynamics of the system and its adaptability to current system conditions.

However, there are a few "outlier" predictions with large errors. To better understand their nature, we show in table 1 the repartition of prediction errors in error intervals. The table shows the number of nodes with prediction errors in this interval, the number of occurrences of the smallest prediction error observed in the interval, and the number of occurrences of the largest prediction errors observed in the interval (e.g. Number of node(s)/number of observed min error/number of observed max error).We see that only a

few nodes contribute (sometimes very many) large prediction errors. Looking further, we identified 3 nodes, all located in India, who contributed consistently to the "tail" of the CDF in figure 3. It is interesting to note that these nodes exhibited large (> 0.75) average measured relative errors during embedding, and were clearly having trouble with the embedding process itself, due to adverse network conditions.

Error Interval	(Vivaldi)	(NPS)
0.0-0.05	257/830/922	232/854/943
0.05-0.1	32/201/995	44/180/941
0.1-0.15	5/3/992	18/5/229
0.15-0.2	1/997/997	2/12/884
0.4-0.45	4/3/56	3/5/12
0.45-0.5	1/12/12	2/1/17
0.5-0.55	1/985/985	2/32/40
0.6-0.65	-	1/851/851

Table 1. Prediction Error Histogram
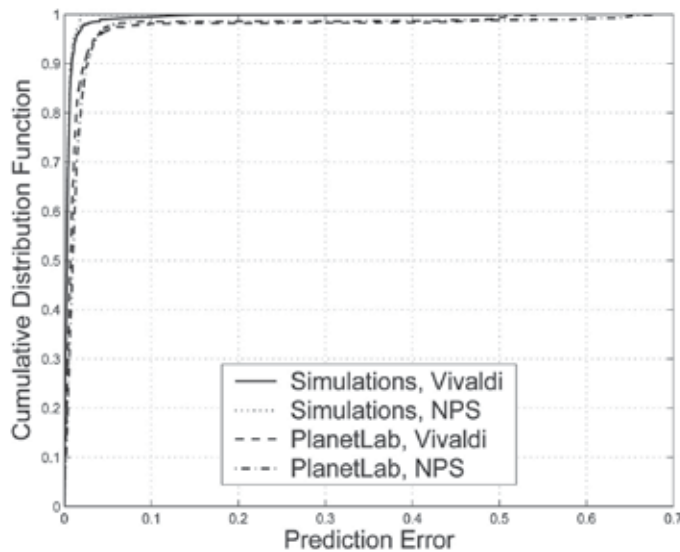
Fig. 3. CDF of prediction errors.

3.3 Representativeness of surveyors

If a subset of nodes in the system (called Surveyor nodes) are trusted and use each other exclusively to compute their own coordinates, they will be immune to the effects of malicious behavior during embedding. The premise of our work is that the "clean" (normal) system behavior thus learnt can then be shared with other nodes and used by these nodes to detect malicious behavior they may be subjected to by other nodes in the system. This obviously assumes that the behavior of the system as observed by Surveyors can approximate or represent well enough the normal behavior of the system as observed by

other nodes in the absence of malicious behavior. In the following validation of this assumption, Surveyors are chosen at random in the node population.

Note that a random choice will give an upper bound on the number of Surveyors needed. Indeed, intuitively, Surveyors should be roughly uniformly distributed in the system to be representative of most other nodes. However, choosing nodes at random in the system does not yield a uniform distribution of Surveyors (i.e. "Surveyor clusters" appear due to the cluster structure of nodes themselves) and therefore not every new Surveyor increases representativeness. Consequently, more Surveyors must be chosen in order to achieve a good coverage of the system, than if they were placed more strategically. Nevertheless, the random choice method does provide general results, without the need to address the question of optimal Surveyor deployment strategy.

One of the first questions to answer is how many Surveyors are needed to be representative of the rest of the population. Noting that our model is based on measured relative errors and that each node in the system observes a series (i.e. distribution) of such errors, we characterize the system-wide relative error behavior as the CDF of the 95th percentiles of the relative errors observed at each (normal) node (i.e. the distribution is made up of the 95th percentile value observed at every node). We then compare this CDF with those of the 95th percentiles of the relative errors observed across a varying population of Surveyors. The choice of the 95th percentile is so that outliers, as observed in section 3.2, do not skew the results, while preserving a high degree of generality. Figure 4, obtained by simulations of Vivaldi, indicates that a population of Surveyors of about 8% of the overall population is closely representative of this overall population (because the CDF for these populations in the figure are similar).
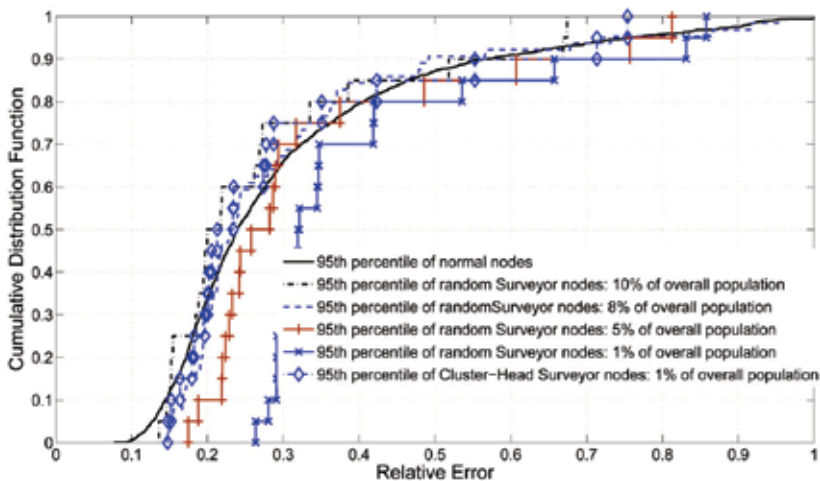

Fig. 4. Impact of Surveyor population size on repretentativeness.

To generalize this result, we then repeated the experiment, using both simulations and PlanetLab measurements, on a Vivaldi system with 8% of Surveyors. We again chose to represent the system by the distribution of the 95th percentile of the measured relative errors (figure 5).

These experiments confirm that less than 10% of randomly chosen Surveyor nodes is enough to gain a good representation of the system behavior. Similar results were observed for NPS. We note that 8% to 10% of the overall node population is a very stringent

requirement for most practical purposes and can represent a huge number of nodes. However, as already pointed out above, random Surveyor deployment is not optimal and this value is an upper bound on the number of Surveyors needed.
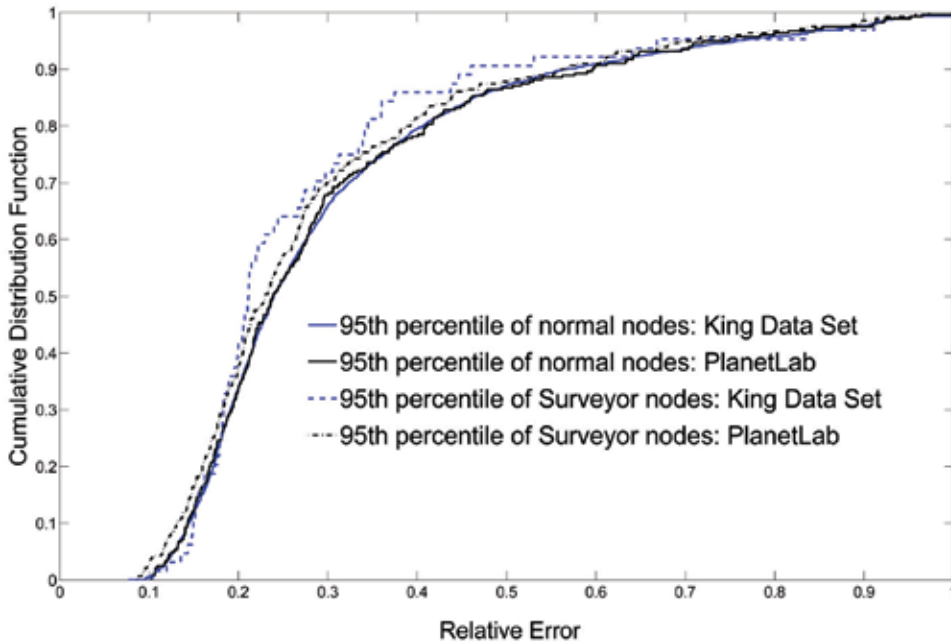

Fig. 5. Representativeness with 8% Surveyor nodes.

To gain further insight into how conservative this upper bound may be, we tried a simple k-means clustering algorithm for Surveyor deployment. Figure 4 shows that when taking cluster heads as Surveyors, good representativeness can be achieved with roughly 1% of Surveyors. Although this does not give much indication as to what the lower bound on the number of Surveyors needed is in the case of optimal Surveyor deployment, it nevertheless shows that simple deployment methods can reduce requirements considerably and that the upper bound yielded by random deployment is indeed very conservative.

Having shown that a population of Surveyors can represent the overall system, the next question is how well the behavior of the system as captured by the Kalman filter calibrated by a Surveyor, can represent the behavior of a single (normal) node. To answer this question, we carried out an experiment where a population of nodes took part in a Vivaldi embedding on PlanetLab. Each node used the Kalman filter of every Surveyor and generated multiple prediction errors (one per Surveyor) at every embedding step.

Figure 6 shows the maximum prediction error yielded by each Surveyor, for each normal node in the system, observed during this experiment. What we observe is that although each normal node can find at least one Surveyor node whose Kalman filter yields very low prediction errors, not every Surveyor is a good representative for any given normal node. The Surveyor chosen as a representative by a normal node is therefore important to achieve good prediction performance (and thus good malicious behavior detection).

Figure 7 plots the prediction accuracy (measured as an average prediction error) against the distance (measured as an RTT) between a node and the corresponding Surveyor, as

observed during the PlanetLab experiment. It is clear that better locality between a node and its Surveyor yields more accurate predictions. This property seems intuitive, as a Surveyor closer in terms of RTT will also be closer in the geometric space, and will thus be more likely to experience dynamics of the coordinate system similar to that of the local area where the node resides. This is confirmed in figure 8 which shows the maximum prediction error, observed for Vivaldi on PlanetLab, when nodes use the closest Surveyor as their representative. Similar results were observed for PlanetLab experiments with NPS.
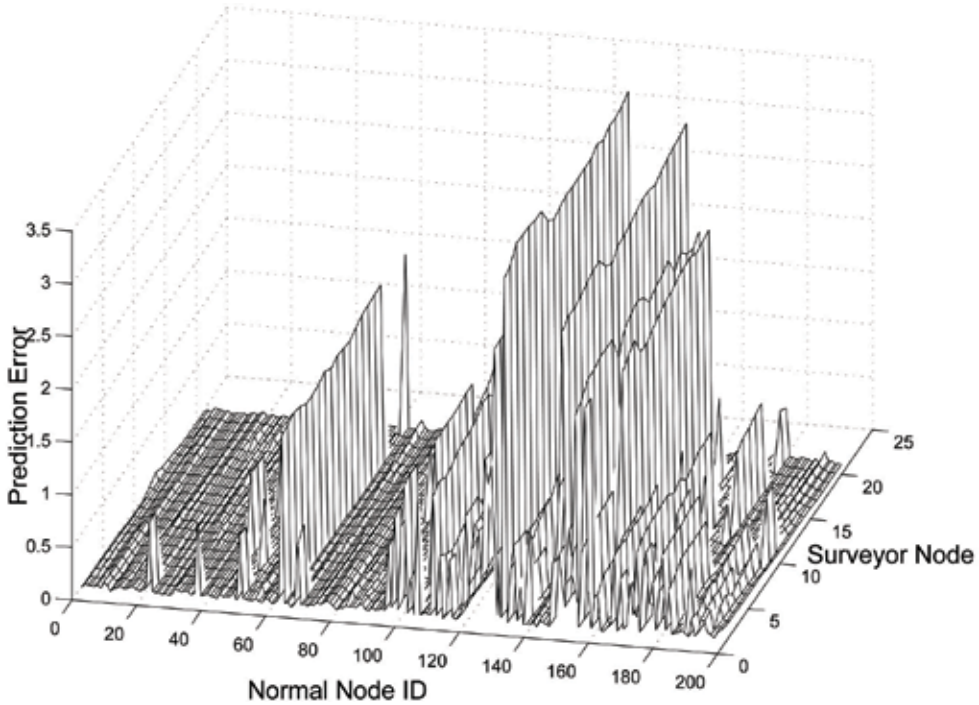
Fig. 6. Maximum prediction errors with Surveyor filter parameters (PlanetLab).
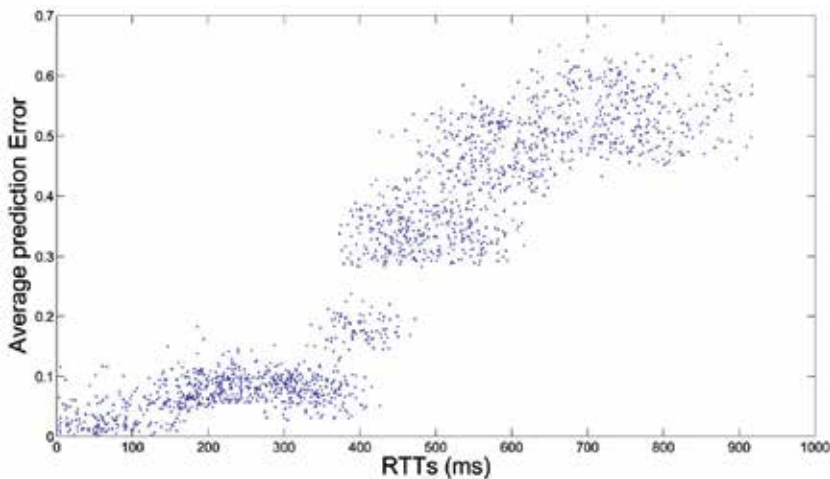
Fig. 7. Correlation between 'Node-Surveyor' RTTs and estimation accuracy (PlanetLab).
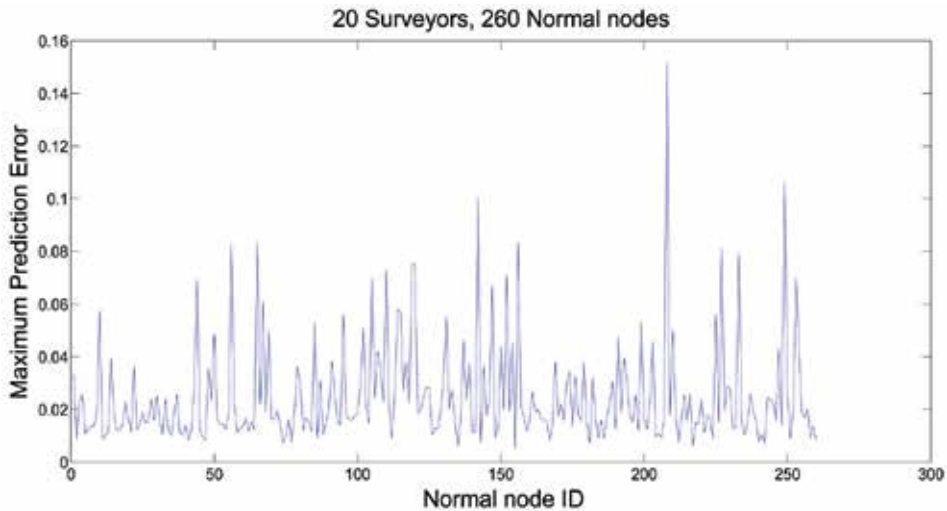
Fig. 8. Maximum prediction errors with closest Surveyor.

Finally, it is again important to note that all the results in this section were obtained with randomly deployed Surveyors. Strategically placing Surveyors to ensure a better coverage of the network and coordinate space, would simply improve the prediction accuracy, while reducing the number of Surveyors required.

4. Malicious behavior detection

The previous section has shown that normal node behavior can be modeled by a Kalman filter. More importantly, it has also been shown that this technique is powerful and robust enough that the normal behavior model captured on one node is readily and effectively applicable on other nearby nodes. This property leads to the idea of Surveyors.

Surveyors are a set of nodes in the coordinate space that exclusively use each other to compute their own coordinates. In other words, in Vivaldi, Surveyors only use other Surveyors as neighbors, while in NPS, they only use other Surveyors as reference points (note that in NPS, all landmarks also act as Surveyors, although not all reference points will be Surveyors). Of course, Surveyors can, and will be chosen as neighbors or reference points by other (non-Surveyor) nodes in the system, but the point is that a Surveyor adjusts its coordinate solely in response to embedding steps (i.e. measurements) with other Surveyors. If Surveyors run a clean version of the coordinate embedding software and they are carefully kept clean of malicious software, such as viruses or worms, that could implement malicious modifications to the embedding, then they can be considered as clean, honest nodes. Because Surveyors only interact with each other during their own embedding, they are therefore immune to malicious or anomalous behavior in the system, and they therefore observe the behavior of the system in clean, normal conditions. The idea is then to use the thus obtained normal behavior model as a basis for anomalous behavior detection at other nodes of the system. To do so, nodes use the parameters of the Kalman filter calibrated at a nearby Surveyor.

It is important to note that the proposed method is entirely distributed as each node has its own filter. Indeed, Surveyors calibrate and recalibrate their own filter as needed, depending

on varying network conditions, and share the resulting filter parameters with other nodes, but they take no further active part in anomalous behavior detection at other nodes. When a node's filter needs re-calibrating (e.g. because it starts giving too many detection alarms), the node simply obtains fresh filter parameters from a Surveyor.

4.1 Anomalous behavior detection method

At each embedding step, a node computes a measured relative error D_n towards a peer node. Recall from section 2 that the Kalman filter at the node can provide $\hat{\Delta}_{n|n-1}$, the predicted relative error from the previously measured relative errors. The innovation process of the Kalman filter yields the deviation between the measured and predicted relative errors, $\eta_n = (D_n - \hat{\Delta}_{n|n-1})$, which, in a system without malicious node, follows a zero-mean gaussian distribution with variance $v_{\eta,n} = v_U + P_{n|n-1}$ (also yielded by the filter).

This allows us to detect malicious behavior as a simple hypothesis test. Let H_0 be the hypothesis that the peer node has a normal behavior (i.e. it is honest). The hypothesis testing simply consists of assessing whether the deviation between the measured and predicted relative errors is normal enough under expected system behavior. Given a "significance level" α, which determines the "aggressivity" or "strictness" of the test, the problem is to find the threshold value t_n such that

$$P(|D_n - \hat{\Delta}_{n|n-1}| \ge t_n \mid H_0) = \alpha. \tag{3}$$

But since, under hypothesis H_0, $(D_n - \hat{\Delta}_{n|n-1})$ follows a zero-mean normal distribution with variance $v_{\eta,n}$, we also have that

$$P(|D_n - \hat{\Delta}_{n|n-1}| \ge t_n \mid H_0) = 2Q(t_n/\sqrt{v_{\eta,n}}), \tag{4}$$

where $Q(x) = 1 - \Phi(x)$, with $\Phi(x)$ being the CDF of a zero-mean, unit-variance normal distribution.

From equations 3 and 4, we therefore have

$$t_n = \sqrt{v_{\eta,n}} Q^{-1}(\alpha/2). \tag{5}$$

If the observed deviation exceeds the threshold given by equation 5, then the hypothesis is rejected, the peer node is flagged as suspicious, the embedding step is aborted and the measured relative error D_n is discarded (i.e. it is not used to update the state of the filter).

Note that a suspicious node, as detected by this test, is not necessarily associated with malicious intent, but could be caused by changing network conditions. Honest nodes classified as suspicious represent false positives and have little impact on the system as long as their occurrence is low. The trade-off between aggressivity and strictness of the test is represented by the so called ROC (Receiver Operation Characteristic) curves [19]. These curves plot the true positive rate versus the false positive rate, *i.e.* the probability of correctly detecting a malicious node versus the probability of labelling an honest node as malicious. In practice trying to increase the true positive rate (the probability of malicious node detection) comes at the cost of increasing the false positive rate.

4.2 Generic detection protocol

In general, on identifying a peer node as suspicious, a node will replace it, that is choose a new neighbor in Vivaldi or a new reference point in NPS.

The only exception to this rule is when the node was embedding against the peer node for the very first time. In this particular case, the node uses its local error e_l [3] as an indicator of the confidence it has in its own coordinate, to carry out a second hypothesis test identical to that presented in the previous section, but this time with a confidence level of $e_l\alpha$. If the test is accepted, then the peer node gets a reprieve and is not replaced, so that a second embedding against this peer node will be attempted at a later time.

The main idea behind this potential reprieve for first-time peer nodes is that a node whose coordinate has already converged towards its true value can afford a few aborted embedding steps with very little impact on the accuracy of its coordinate. On the other hand, a new peer node which is in the process of joining the network may trigger the abortion of an embedding step, simply because its coordinate has not converged yet (as opposed to because it displays a malicious behavior). In this case, the reprieve simply gives time to the new (joining) peer node to converge before being identified as malicious. Of course an embedding node which is not confident in its coordinate must strive to reduce the number of aborted embedding steps so as not to compromise its convergence in the system, and will therefore grant fewer reprieves (because its e_l is higher) than a node that has already converged.

Finally, we use a simple mechanism for the selection of the Surveyor from which a node obtains its calibrated Kalman filter. All Surveyor nodes register with an infrastructure server (e.g. the membership server in NPS can act as Surveyor registrar, while in Vivaldi such server must either be introduced or at least integrated inside an existing bootstrap infrastructure). On joining the coordinate system, a node interrogates this server to obtain the identity of several (randomly chosen) Surveyors. The node then measures its distance to these Surveyors and selects the closest one as representative. From there on, the node fully complies to the embedding protocol rules, except that it will use our detection method to accept or reject embedding steps.

However, when the node has rejected half of its current peer nodes during a same embedding round, it will seek to acquire a new filter as the high rejection rate may indicate that the filter parameters in use may have become stale (i.e. the filter needs "recalibrating"). The node then gets from its current Surveyor (or, as a fallback, any other Surveyors it knows, or the infrastructure server) the list of all the Surveyors it knows. After acquiring the current coordinates of these Surveyors, the node selects the closest one (in term of estimated distance) and obtains its Kalman filter parameters. Note that in the experiments we have carried out, which are described below, we observed very few "recalibrations", so this very simple Surveyor selection mechanism was appropriate. However, more sophisticated approaches can be considered if need be.

5. Evaluation

We evaluate the effectiveness of the simple anomalous/malicious behavior detection method in securing both Vivaldi and NPS. For each of these embedding protocols, we chose

---

[3] e_l is the exponential moving weighted average of the measured relative errors of all previously completed embedding steps.

the most potent attack described in [2] and experimented with various populations of malicious nodes within the experimental set-up described in section 3. On PlanetLab, all these experiments were run concurrently so as to experience the same network conditions. In line with the results of section 3.3, the population of Surveyors was set to 8% of the overall population. Surveyors and malicious nodes were chosen at random.

5.1 Performance metrics

To characterize the performance of our detection test, we use the classical false/true positives/negatives indicators. Specifically, a *negative* is a normal embedding step which should therefore be accepted by the test and completed. A *positive* is a malicious embedding step (i.e. where either, or both, the distance estimation and distance measurement between the node and its peer node have been tampered with) which should therefore be rejected by the test and aborted. The number of negatives (resp. positives) in the population comprising all the embedding steps is $\mathcal{P}_N$ (resp. $\mathcal{P}_P$).

A *false negative* is a malicious embedding step that has been wrongly classified by the test as negative, and has therefore been wrongly completed. A *false positive* is a normal embedding step that has been wrongly rejected by the test and therefore wrongly aborted. *True positives (resp. true negatives)* are positives (resp. negatives) that have been correctly reported by the test and therefore have been rightly aborted (resp. completed). The number of false negatives (resp. false positives, true negatives and true positives) reported by the test is $\mathcal{T}_{FN}$ (resp. $\mathcal{T}_{FP}$, $\mathcal{T}_{TN}$ and $\mathcal{T}_{TP}$).

We use the notion of *false negative rate* (FNR) which is the proportion of all the malicious embedding steps that have been wrongly reported as normal by the test, and FNR = $\mathcal{T}_{FN}/\mathcal{P}_P$. The *false positive rate* (FPR) is the proportion of all the normal embedding steps that have been wrongly reported as positive by the test, so FPR = $\mathcal{T}_{FP}/\mathcal{P}_N$. Similarly, the *true positive rate* (TPR) is the proportion of malicious embedding steps that have been rightly reported as malicious by the test, and we have TPR = $\mathcal{T}_{TP}/\mathcal{P}_P$.

The *true positive test fraction* (TPTF) is the proportion of positive tests that correctly identified malicious embedding steps (TPTF = $\mathcal{T}_{TP}/(\mathcal{T}_{TP}+\mathcal{T}_{FP})$).

5.2 Securing Vivaldi

We experimented with our detection scheme on a Vivaldi system subjected to a colluding isolation attack as described in [2]. In this scenario, malicious nodes are trying to isolate a target node, by repulsing all other nodes away from it. The malicious nodes agree on a large "exclusion" zone around the target node and randomly set their own coordinates outside this zone to try and attract honest nodes out of the exclusion zone. Note that an attacker always uses the same coordinate when lying to a given honest node.

Detection Method Performance To evaluate the efficiency of the test, we first plot in figure 9, ROC (Receiver Operation Characteristics) curves observed for different significance levels (α) and several intensities of attacks. These plots show, for each significance level[4], the point corresponding to the false positive rate along the x-axis and to the true positive rate along the y-axis, with one curve per malicious group size (line x=y is plotted as a reference). Obviously, the closer to the upper left corner of the graph a curve is, the better, since such

---

[4] Significance level values α always increase as a ROC curve is "followed" from the origin. In our experiments, we used values of 1%, 3%, 5% and 10% for α.
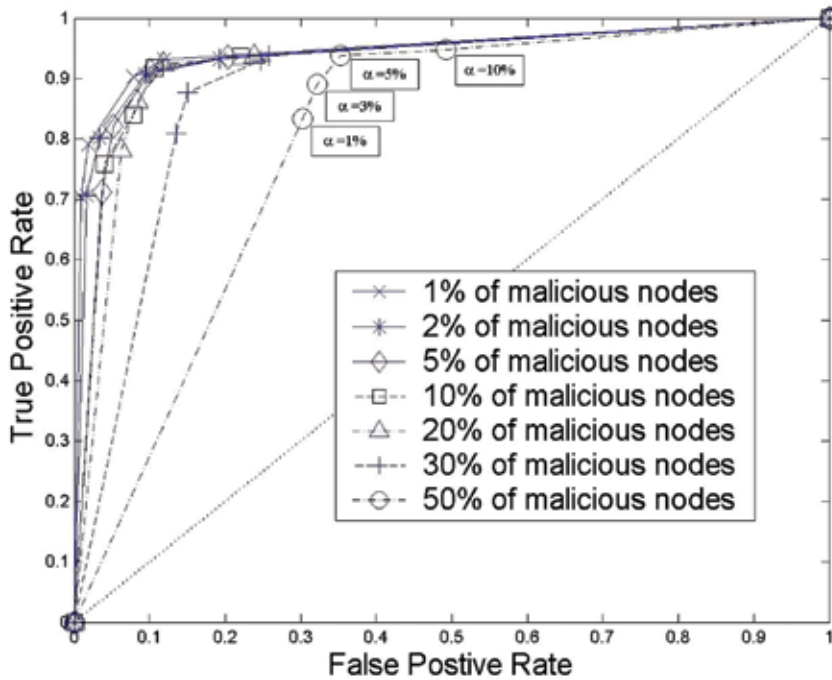
Fig. 9. ROC curves. Each tick on the plots corresponds to a different value of the test's significance level (α).

points correspond to high true positive rates (i.e. a high proportion of positives being reported as such by the test) for low false positive rates (i.e. a small proportion of negatives incorrectly reported as positives). We observe that from this perspective, the detection method can be considered to be excellent for 20% of malicious nodes or less, and still performs well even under heavy attack of up to about 30% of malicious nodes, while the power of the detection method naturally decreases as the malicious population becomes more significant. Another interesting properties of ROC curves is that they show the optimal range for the significance level. Indeed, as the slope of the ROC curve flattens, the increase in true positive rate is proportionally smaller than the corresponding increase in false positives. In other words, a higher significance level, although it always increases the true positive rate of the test, is not always productive as it eventually does more bad than good through increased false positive rates (i.e. the proportion of normal embedding steps that are aborted increases). This means that the significance level of the test should be set to a value that yields a point in the "elbow" of the ROC curve. Based on figure 9, we can deduce that a significance level of 5% seems to be a good compromise.

Figure 10 shows the true positive test fraction of the detection method for various test significance levels under various intensity of attacks. We see that the proportion of positive tests that are true positives is constantly high, regardless of the significance level chosen, for moderate to quite significant proportions of malicious nodes in the population (up to 20% of malicious nodes). However, thereafter the proportion of correct positive tests starts to decrease, although the rate of decrease is inversely proportional to the significance level used. This is because a higher significance level produces more positive tests, catching most

malicious embedding steps, and so many more false positives are needed to make up a significant proportion of these. In light of this, a significance level of 5% offers a good compromise.
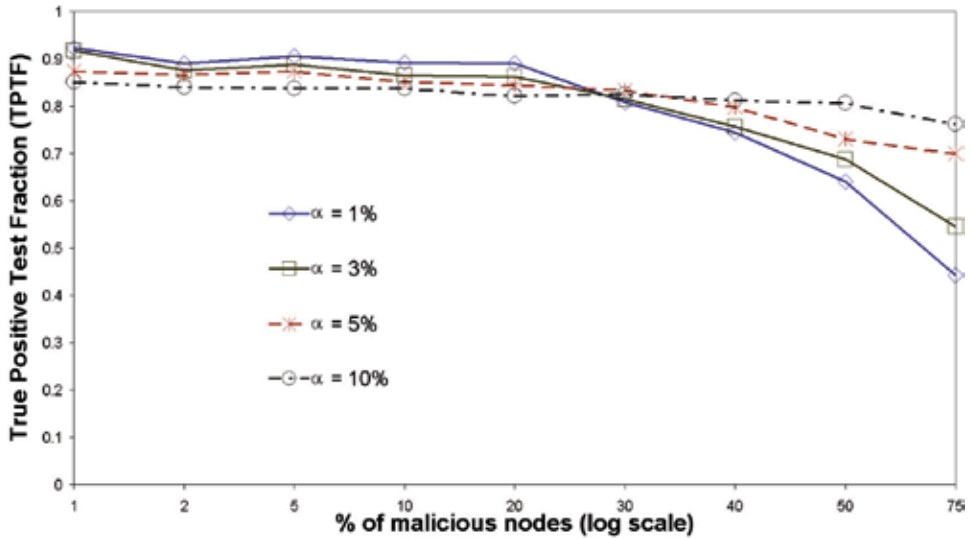

Fig. 10. True positive test fraction.

Figures 11 and 12 show the false positive and negative rates respectively. As expected, a higher significance level results in a more aggressive test that incorrectly classifies a larger portion of normal embedding steps (figure 11) as malicious, while a more lenient test (lower significance level) wrongly reports a higher proportion of malicious embedding steps as normal (figure 12).
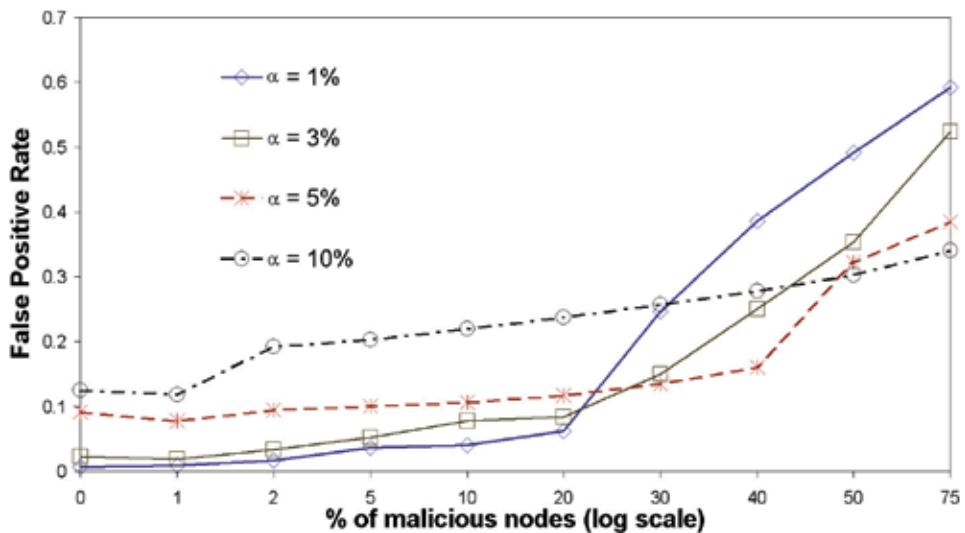
Fig. 11. False Positive Rate.
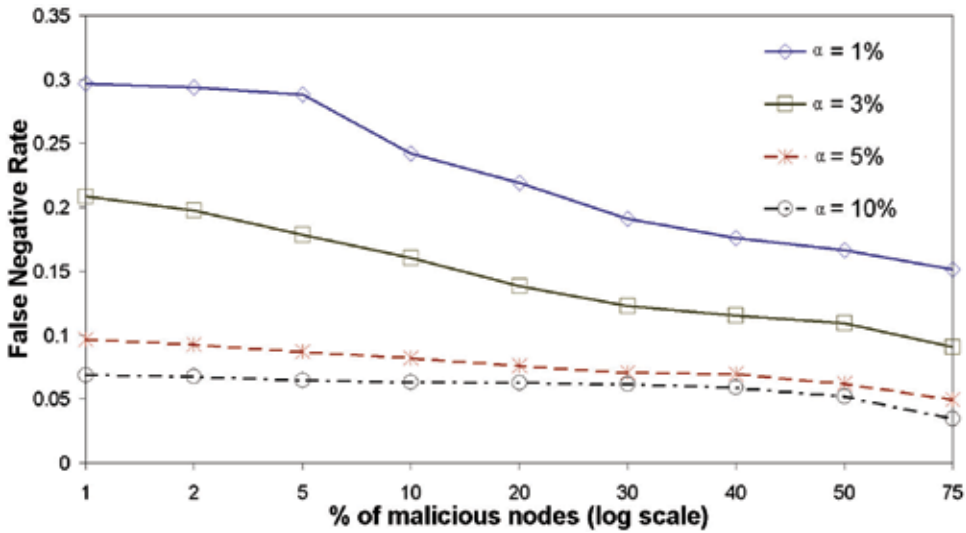
Fig. 12. False negative rate.

Incorrect test results do have a negative impact on the embedding system: false positives artificially reduce the size of available normal nodes that can be used for normal embedding; false negatives give malicious nodes opportunities to corrupt and distort the coordinate space, which can propagate through the system and result in a greater proportion of normal nodes being identified as malicious (false positives) because of mis-positioning. This is exemplified in figure 11, where the false positive rate increases faster, as the population of malicious nodes increases, for lower values of the significance level of the test. Also, despite the fact that the false negative rate curves (figure 12) clearly exhibit negative slopes, one should note that these rates decrease much slower than the increase in malicious population. That is to say that as the number of malicious nodes in the system increases, the number of false negatives does increase, and more damage is incurred in the coordinate space. Although the accuracy of coordinate systems increases with the number of participating nodes, false negatives can therefore have a greater impact on the system than false positives and should therefore be thwarted in priority. As the false negative rates exhibited by tests with significance levels of 5% and 10% are roughly similar, while the more aggressive test yields proportionally a higher false positive rate, the significance level of 5% is a good compromise.

Embedding System Performance From section 5.2, it should be clear that a significance level of 5% gives the overall best test performance. We therefore set the significance level to this value and assess the resistance of a Vivaldi system under various intensity of attacks.

The cumulative distribution function of the measured relative errors, across all normal nodes, after convergence (in the sense of error convergence as defined in section 2) is shown in figure 13. We see that the detection mechanism renders the system practically immune to the attack, when the proportion of malicious nodes is 30%, or less, of the overall node population. Although the system does indeed show degraded performance for higher intensities of malicious attacks, the steeper slope of the CDF with detection, compared to the corresponding curve without (e.g. curves for 50% of malicious nodes), shows that the detection mechanism is not completely overwhelmed and still offers good protection by significantly reducing the impact of the attack.
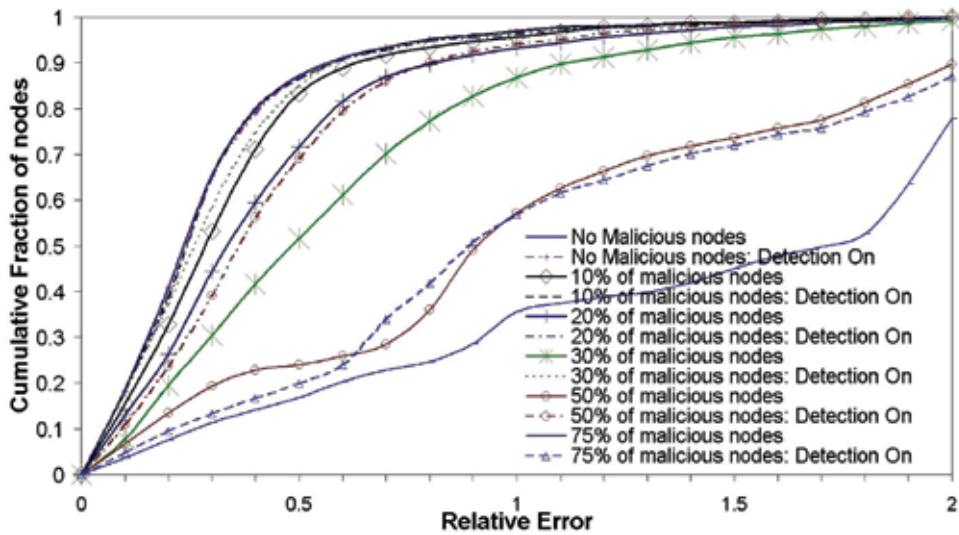
Fig. 13. Distribution of measured relative errors.

5.3 Securing NPS

To test our proposed detection method in the context of the NPS coordinate system, we chose to study the effects of colluding isolation attack as described in [2]. The malicious nodes cooperate with each other and behave in a correct and honest way until enough of them become reference points at each layer. As soon as a minimum number of malicious reference points has been reached (in our experiments this number is set to 5) in a layer, these attackers identify a common set of victims (50% of the normal nodes they know from the layer directly below). When involved in the positioning of their victims, the malicious nodes agree to pretend they are all clustered into a remote (far away) part of the coordinate space and try and push the victims into a remote location at the "opposite" of where the attackers pretend to be, in order to isolate the victims from the other nodes in the coordinate space. In order to evade detection, including the basic detection method proposed in NPS and which is *always* turned on in our experiments, the malicious nodes use the sophisticated anti-detection method proposed in [2] during their attacks.

Detection Method Performance Figure 14 shows the ROC curves for the detection test in NPS. These curves show characteristics similar to those observed in the Vivaldi system (see section 5.2), albeit slightly better. In particular, these curves show that the detection method withstands heavier attacks better in NPS than in Vivaldi.

There are several reasons for this. First, the basic detection method in NPS works in concert with our own, providing greater opportunities to identify malicious behavior. Also, by its very nature, the embedding method in NPS is less prone to mis-positioning error propagation amongst normal nodes, as nodes in the lower layer do not take part in the embedding of other nodes. And finally, by design, the attack considered in this section makes fewer victims than that studied in section 5.2 (i.e. 50% of normal nodes as victims vs 100% in Vivaldi).

The same observation is also true for the false positive and false negative rates (not shown) with again, overall, a significance level of 5% seemingly offering the best compromise between "catching" malicious embedding steps while not being overly cautious and over-reacting to normal variations in network conditions.
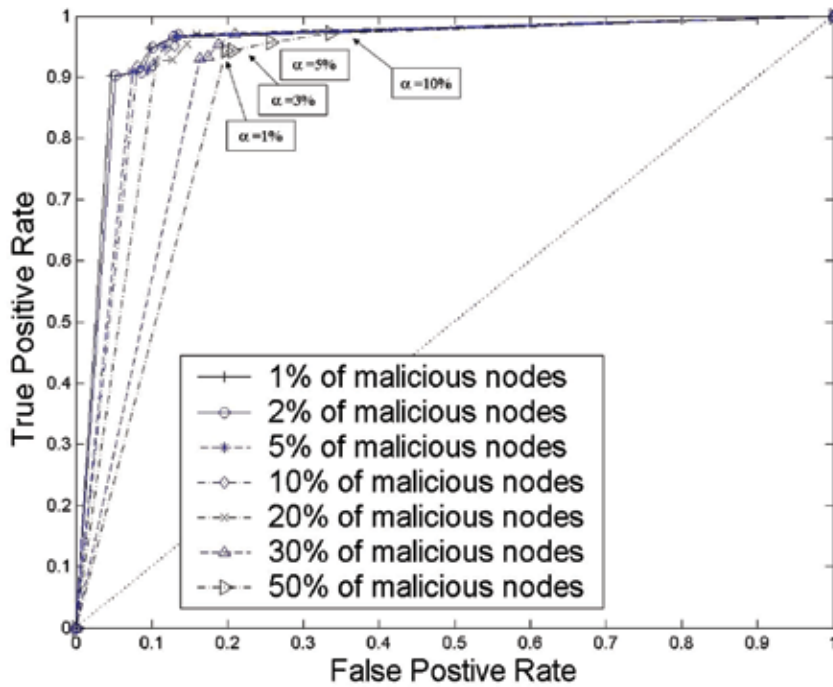
Fig. 14. ROC curves.

The similarities between the test performance under NPS and Vivaldi, despite the different nature of the attacks under consideration and even differences in coordinate "structure" (two-dimensional with height for Vivaldi versus eight-dimensional for NPS), illustrates the generality of the proposed detection method. This is because our detection test is based on the modeling of a dimension-less quantity (the relative error) which is at the very core of *any* coordinates embedding system.

Embedding System Performance We study the performance of the NPS embedding system when subject to increasing intensity of attacks, while being protected by our detection scheme. Note that in this section, "detection off" really means that our proposed detection mechanism is not used, but the basic NPS detection mechanism is still "on".

Figure 15 shows the cumulative distribution function of relative errors in the system. We note again similarities with the dynamic behavior of similar Vivaldi systems, except that the tail of the CDF for 50% malicious nodes with detection is heavier than the corresponding curve in the Vivaldi case. Keeping in mind that in NPS not all nodes are victims and that not all normal nodes will propagate mis-positioning errors, this indicates that the attack is still quite effective against its victims, albeit "dampened" by the detection mechanism. This effect is compounded by the fact that, with our simple detection protocol, malicious nodes that have found their way into the layer hierarchy of NPS and act as Reference Points, do stay in place throughout the experiment, despite numerous detections of their corrupt embedding steps. Nevertheless, the detection method proposed affords near immunity to the system up to rather severe attack conditions (e.g. about 30% of malicious nodes in the system).
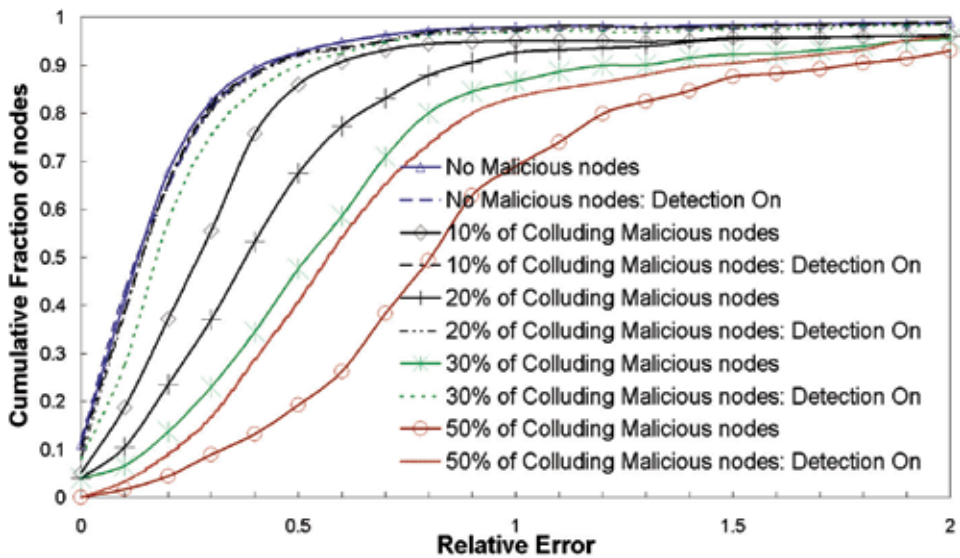
Fig. 15. Distribution of measured relative errors.

6. Conclusions

We have presented a method for malicious behavior detection to secure the embedding phase of Internet coordinate systems. Our method does not rely on the geometric properties of the coordinate space, and is therefore unaffected by potential triangular inequality violations which often occur in the Internet [11, 12]. Instead, our detection test is based on the modeling of the dynamic relative errors observed in a clean system. The relative error is a dimension-less quantity which is at the very core of *any* embedding method, leading us to believe that our proposed detection test can effectively identify malicious behavior in very many embedding protocols and coordinate space structures that are under a potential very large range of attacks. The experiments presented in this chapter do show that the performance of the detection test is effectively the same in two different scenarios involving different embedding protocols and different attacks. As far as we know, this is the first such general detection test, capable of surviving sophisticated attacks. Also, we consider exclusively attacks aimed at distorting the coordinate space, carried out by nodes inside the embedding system. Our method thus succeeds where more obvious methods based on authentication would fail.

7. References

[1] M. A. Kaafar, L.Mathy, T. Turletti andW. Dabbous, *Real attacks on virtual networks: Vivaldi out of tune*, In Proceedings of the SIGCOMM workshop on Large Scale Attack Defense LSAD 2006,p129-146, PISA, September 2006.
[2] M. A. Kaafar, L. Mathy, T. Turletti, and W. Dabbous, *Virtual Networks under Attack: Disrupting Internet Coordinate Systems,* In Proceedings of CoNext 2006, Lisboa, December, 2006.

[3] M. A. Kaafar, L. Mathy, C. Barakat, K. Salamatian, T. Turletti and W. Dabbous, *Securing Internet Coordinate Embedding Systems*, In proceedings of ACMSIGCOMM, Kyoto, Japan, August 2007.

[4] R.E. Kalman, and R.S. Bucy, *New Results in Linear Filtering and Prediction Theory*, In Transactions of the ASME - Journal of Basic Engineering Vol. 83: pp. 95-107, 1961.

[5] T. E. Ng, and H. Zhang, *Predicting internet network distance with coordinates-based approaches*, In Proceedings of the IEEE INFOCOM, New York, June 2002.

[6] M. Pias, J. Crowcroft, S. Wilbur, S. Bhatti, and T. Harris, *Lighthouses for Scalable Distributed Location*, In Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS), Berkeley, February 2003.

[7] M. Costa, M. Castro, A. Rowstron, and P. Key, *Practical Internet coordinates for distance estimation*, In Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), Tokyo, March 2004.

[8] T. E. Ng and H. Zhang, *A Network Positioning System for the Internet*, In Proceedings of the USENIX annual technical conference, Boston, June 2004.

[9] F. Dabek, R. Cox, F. Kaashoek and R. Morris, *Vivaldi: A decentralized network coordinate system*, In Proceedings of the ACM SIGCOMM, Portland, Oregon, August 2004.

[10] J. Ledlie, P. Gardner, and M. Seltzer, *Network Coordinates in the Wild*, In Proceedings of NSDI, Cambridge, April 2007. Available as Harvard University Computer Science Technical Report TR-20-06, October 2006.

[11] E. K. Lua, T. griffin, M. Pias, H. Zheng, and J. Crowcroft, *On the accuracy of Embeddings for Internet Coordinate Systems*, In Proceedings of InternetMeasurement Conference (IMC), Berkeley, October 2005.

[12] H. Zheng, E. K. Lua,M. Pias, and T. Griffin, *Internet Routing Policies and Round-Trip Times*, In Proceedings of the Passive Active Measurement (PAM), Boston, March 2005.

[13] K. P. Gummadi, S. Saroiu, and S. D. Gribble, *King: Estimating Latency between Arbitrary Internet End Hosts*, In Proceedings of SIGCOMMInternetMesasurementWorkshop (IMW), Pittsburgh November 2002.

[14] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, *On the Constancy of Internet Path Properties*, In Proceedings of ACMSIGCOMMInternetMeasurementWorkshop, San Francisco, November 2001.

[15] Z. Ghahramani, G. Hinton, *Parameter Estimation for Linear Dynamical Systems*, University of Toronto, Technical Report CRG-TR-96-2.

[16] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice Hall, Ch. 7, 1993.

[17] A. Dempster, N. Laird, and D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, Series B, 39(1):1-38, 1977.

[18] H. Lilliefors, *On the Kolmogorov-Smirnov test for normality with mean and variance unknown*, Journal of the American Statistical Association, Vol. 62. pp. 399-402, June, 1967.

[19] A. Soule, K. Salamatian, and N. Taft, *Combining Filtering and Statistical Methods for Anomaly Detection*, In Proceedings of Internet Measurement Conference (IMC), Berkeley, October, 2005.

[20] J. Bilmes, *A gentle tutorial on the EM algorithm including gaussian mixtures and baumwelch*, Technical Report TR-97-021, International Computer Science Institute, Berkeley, CA, 1997.

Variants of Kalman Filter for the Synchronization of Chaotic Systems

Sadasivan Puthusserypady[1] and Ajeesh P. Kurian[2]
[1]Department of Electrical and Computer Engineering, National University of Singapore
[2]Department of Electrical and Computer Engineering, University of Calgary
[1]Singapore
[2]Canada

1. Introduction

Perhaps the most important lesson to be drawn from the study of nonlinear dynamical systems over the past few decades is that even simple dynamical systems can give rise to complex behaviour which is statistically indistinguishable from that produced by a complex random process. Chaotic systems are nonlinear systems which exhibit such complex behaviour. In such systems, the state variables move in a bounded, aperiodic, random-like fashion. A distinct property of chaotic dynamics is its long-term unpredictability. In such systems, initial states which are very close to each other produce markedly different trajectories. When nearby points evolve to result in uncorrelated trajectories, while forming the same attractor, the dynamical system is said to possess sensitive dependence to initial conditions (Devaney, 1985). Due to these desirable properties, application of chaotic systems are explored for many engineering applications such as secure communications, data encryption, digital water marking, pseudo random number generation etc (Kennedy et. al., 2000). In most of these applications, it is essential to synchronize the chaotic systems at two different locations. In this chapter, we explore the Kalman filter based chaotic synchronization.

2. Synchronization of chaotic systems

Related works of synchronization dates back to the research carried out by Fujisaka and Yamada in 1983 (Fujisaka & Yamada, 1983). Pecora and Carroll suggested a drive-response system for synchronization of chaotic systems. They showed that if all the transversal Lyapunov exponents of the response system are negative, the systems synchronize asymptotically (Pecora & Carrol, 1990). Later a plethora of research work was reported on synchronization of chaotic systems (Nijmeijer & Mareels, 1997). One of the well researched approaches is the coupled synchronization, where a proper coupling is introduced between the transmitter and the receiver. The chaotic systems synchronize asymptotically when the coupling strength is above a certain threshold, which is determined by the local Lyapunov exponents (Suchichik et. al., 1997).
Let the transmitter and the receiver states of the chaotic systems be given by

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mu) \tag{1.a}$$

$$\hat{\mathbf{x}}_{k+1} = \mathbf{f}(\hat{\mathbf{x}}_k, \hat{\mu}) \tag{1.b}$$

where $\mathbf{x}_k = \left[x_k^1, \ldots, x_k^n \right]^T$ and $\hat{\mathbf{x}}_k = \left[\hat{x}_k^1, \ldots, \hat{x}_k^n \right]^T$ are the n-dimensional state vectors of the transmitter and the receiver systems, respectively. μ and $\hat{\mu}$ are the transmitter and the receiver system parameters and $\mathbf{f} = [f^1(.), \ldots, f^n(.)]^T$ is a smooth nonlinear vector valued function. Normally, we have a noisy observation at the receiver which is given by

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{2}$$

where $\mathbf{v}_k$ is the channel noise and $\mathbf{h}(.)$ is the measurement function. These two systems are said to be synchronized if

$$\lim_{k \to \infty} \left\| \mathbf{x}_k - \hat{\mathbf{x}}_k \right\| = 0. \tag{3}$$

2.1 Coupled synchronization

In coupled synchronization, a coupling is introduced between the transmitter and the receiver as:

$$\hat{\mathbf{x}}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1}) + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \tag{4}$$

where $\mathbf{K}_k$ is the appropriate coupling coefficient matrix. In conventional coupled synchronization, $\mathbf{K}_k$ is set to be a constant value such that the global and local Lyapunov exponents (The Lyapunov exponents of a dynamic system are the quantities that characterize the rate of divergence/convergence of the trajectories generated by infinitesimally close initial conditions under the dynamics) are negative. This makes the receiver to synchronize with the transmitter asymptotically. The schematic of the coupled synchronization is shown in Fig. 1. This method of synchronization can be treated as a predictor corrector filter approach. In general, a predictive filter predicts the subsequent states and corrects it with additional information available from the observation. Due to the measurement and the channel noises in a communication system, stochastic techniques have to be applied for synchronization. Instead of keeping $\mathbf{K}_k$ as a constant value if it is determined adaptively, the coupled synchronization has a similarity with the predictive filtering techniques.

3. Chaos synchronization: a stochastic estimation view point

In stochastic state estimation methods, one would like to estimate the state variable $\mathbf{x}_k$ based on the set of all available (noisy) measurement $\mathbf{y}_{1:k} = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ with certain degree of confidence. This is done by constructing the conditional probability density function (pdf), $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ (i.e. the probability of $\mathbf{x}_k$ given the observations, $\mathbf{y}_{1:k}$), known as the posterior probability. It is assumed that $p(\mathbf{x}_0 | \mathbf{y}_0)$ is available. In predictor corrector filtering methods, $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ is obtained recursively by a prediction step which is estimated without the knowledge of current measurement, $\mathbf{y}_k$ followed by a correction step where the knowledge of $\mathbf{y}_k$ is used to make the correction to the predicted values.
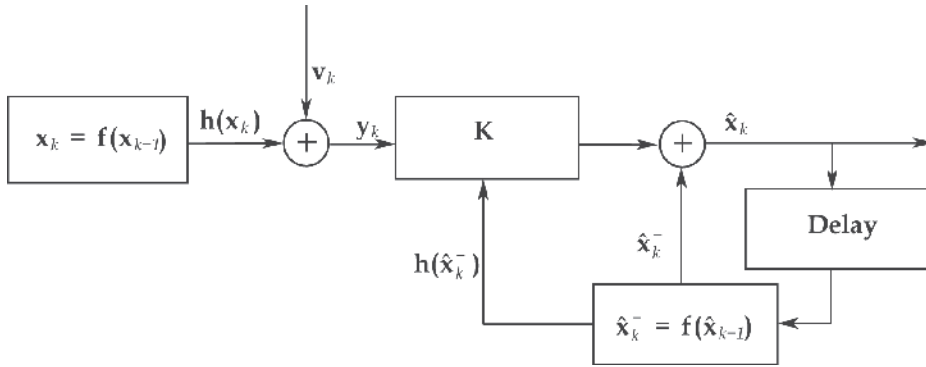
Fig. 1. Schematic of the coupled synchronization scheme.

In the recursive computation of $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$, it is assumed that at time, $k-1$, $p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})$ is available. Using the Chapman-Kolmogorov equation (Arulampalam, et. al. 2001), the prediction is estimated as

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} \tag{5}$$

where the state transition is assumed to be a Markov process of order one and $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$. To make the correction, one needs to make use of the information available in the current observation, $\mathbf{y}_k$. Using Bayes' rule,

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \frac{p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) p(\mathbf{y}_k \mid \mathbf{x}_k)}{p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})} \tag{6}$$

where the normalizing constant

$$p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) d\mathbf{x}_k . \tag{7}$$

Though closed form solutions of the above equations exist for a linear system with Gaussian noise (e.g. Kalman filter), in general, for a nonlinear system, they are not available. However, one of the suboptimal filtering methods, the extended Kalman filter (EKF) is found to be useful in many nonlinear filtering applications.

3.1 EKF for chaos synchronization
The Kalman filter is an optimal recursive estimation algorithm for linear systems with Gaussian noise. A distinctive feature of this filter is that its mathematical formulation is described in terms of the state-space. The EKF is an extension of the Kalman filtering algorithm to nonlinear systems (Grewal & Andrews, 2001). The system is linearized using first order Taylor series approximation. To this approximated system, the Kalman filter is applied to obtain the state estimates. Consider a generic dynamic system governed by

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_k) . \tag{8.a}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \tag{8.b}$$

where the process noise, $\mathbf{w}_k$, and observation (measurement) noise, $\mathbf{v}_k$, are zero mean Gaussian processes with covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. In minimum mean square estimation (MMSE), the receiver computes $\hat{\mathbf{x}}_k$ which is an estimate of $\mathbf{x}_k$ from the available observations $\mathbf{y}_{1:k} = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ such that the mean square error (MSE), $\mathbb{E}\left[\mathbf{e}_k^T \mathbf{e}_k\right]$ (where $\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$), is minimized. The EKF algorithm for the state estimation is given by,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, 0), \tag{9.a}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{W}_k\mathbf{Q}_k\mathbf{W}_k^T \tag{9.b}$$

In the above equations, the notation $k|k-1$ denotes an operation performed at time instant, k, using the information available till $k-1$. At time instant k, $\hat{\mathbf{x}}_{k|k-1}$ is the *a priori* estimate of the state vector $\mathbf{x}_k$, $\mathbf{P}_{k|k-1}$ is the *a priori* error covariance matrix, $\mathbf{F}_{k-1}$ is the Jacobian of $\mathbf{f}(.)$ with respect to the state vector $\mathbf{x}_{k-1}$ and $\mathbf{W}_k$ is the Jacobian of $\mathbf{f}(.)$ with respect to the noise vector $\mathbf{w}_k$. The EKF update equations are:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T \left\{\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{V}_k\mathbf{R}_k\mathbf{V}_k^T\right\}^{-1} \tag{10.a}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right) \tag{10.b}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{10.c}$$

where $\mathbf{K}_k$ is the Kalman gain, $\mathbf{H}_k$ is the Jacobian of $\mathbf{h}(.)$ with respect to $\hat{\mathbf{x}}_{k|k-1}$, $\hat{\mathbf{x}}_k$ is the *a posteriori* estimate of the state vector, $\mathbf{V}_k$ is the Jacobian of $\mathbf{h}(.)$ with respect to the noise vector $\mathbf{v}_k$, and $\mathbf{P}_k$ is the *a posterior* error covariance matrix. When EKF is used for synchronization of chaotic maps, $\mathbf{K}_k$ acts as the coupling strength which is updated iteratively. Schematic of EKF based synchronization is shown in Fig. 2.
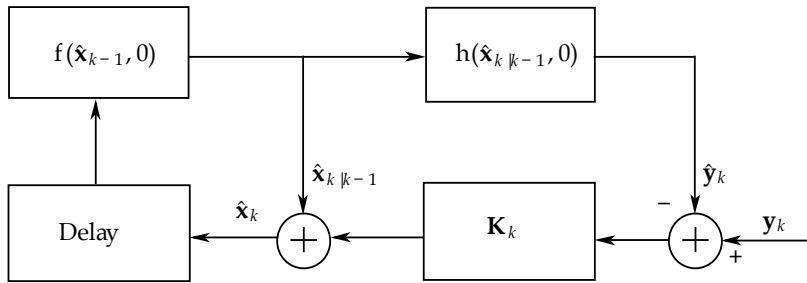

Fig. 2. Schematic of EKF based chaos synchronization

3.2.1 Convergence analysis

Convergence analysis of $\mathbf{K}_k$ can be carried out by studying the convergence of $\mathbf{P}_{k|k-1}$. At any time instant, k, according to the *matrix fraction propagation* of $\mathbf{P}_{k|k-1}$, it can be shown that (Grewal & Andrews, 2001),

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k\mathbf{B}_k^{-1}, \tag{11}$$

where $\mathbf{A}_k$ and $\mathbf{B}_k^{-1}$ are factors of $\mathbf{P}_{k|k-1}$. If $\mathbf{F}_k$ is nonsingular (i.e. the map is invertible), then $\mathbf{A}_{k+1}$ and $\mathbf{B}_{k+1}$ are given by the recursive equation as

$$\begin{bmatrix} \mathbf{A}_{k+1} \\ \mathbf{B}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_k + \mathbf{W}_k\mathbf{F}_k^{-T}\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k & \mathbf{W}_k\mathbf{F}_k^{-T} \\ \mathbf{F}_k^{-T}\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k & \mathbf{F}_k^{-T} \end{bmatrix}\begin{bmatrix} \mathbf{A}_k \\ \mathbf{B}_k \end{bmatrix}. \tag{12}$$

From the above expression, it can be shown that, when there is no process noise (i.e. $\mathbf{W}_k = \mathbf{0}$ and $\mathbf{F}_k$ is *contractive* (i.e. the magnitudes of its eigenvalues are less than one), $\mathbf{P}_{k|k-1}$ will converge in time. However, inside the chaotic attractor, the behaviour of $\mathbf{P}_{k|k-1}$ is aperiodic if $\mathbf{F}_k$ is time varying. This behaviour has dramatic influence on the convergence of Kalman filter based synchronization system (Kurian, 2006) for systems with hyperbolic tangencies (HTs).

3.3 Unscented Kalman filter

The approximation error introduced by the EKF together with the expansions of this error at the HTs makes the system unstable and diverging trajectories are generated at the receiver. One way to mitigate this problem is to use nonlinear filters with better approximation capabilities. Unscented Kalman filter (UKF) has shown to possess these capabilities (Julier & Uhlman, 2004). It is essentially an approximation method to solve Eq. (5). UKF works based on the principle of unscented transform (UT) (Julier & Uhlman, 1997).
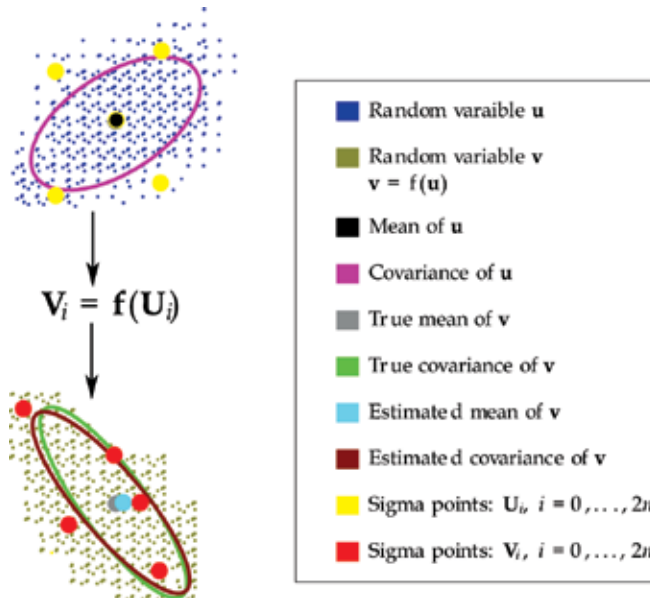

Fig. 3. Illustration of unscented transform (UT).

In Fig. 3, the UT of a random variable, $\mathbf{u}$, which undergoes a nonlinear transformation ($\mathbf{f}(\mathbf{u})$) to result in another random variable, $\mathbf{v}$ is shown. To calculate the statistics of $\mathbf{v}$, the ideal solution is to get posterior density, $p(\mathbf{v})$, analytically from the prior density $p(\mathbf{u})$. The mean and covariance of $\mathbf{v}$ can also be computed analytically. However, this is highly impractical in most of the situations because of the nonlinearity. UT is a method for

approximating the statistics of a random variable which undergoes a nonlinear transformation. It uses carefully selected vectors ($\mathcal{U}_i$), known as *sigma points*, to approximate the statistics of the posterior density. Each sigma point is associated with a weight W_i. The number of sigma points is $2n+1$ where n is the dimension of the state vector. With the knowledge of the mean ($\hat{\mathbf{u}}$) and covariance ($\mathbf{P_u}$) of the prior density, these sigma points are constructed as

$$
\begin{aligned}
(\mathcal{U}_0, W_0) &= \left(\hat{\mathbf{u}}, \frac{\kappa}{n+\kappa} \right); \quad i = 0 \\
(\mathcal{U}_i, W_i) &= \left(\hat{\mathbf{u}} + \left(\sqrt{(n+\kappa)\mathbf{P_u}} \right)_i, \frac{1}{2(n+\kappa)} \right); \quad i = 1,\ldots,n \\
(\mathcal{U}_i, W_i) &= \left(\hat{\mathbf{u}} - \left(\sqrt{(n+\kappa)\mathbf{P_u}} \right)_i, \frac{1}{2(n+\kappa)} \right); \quad i = n+1,\ldots,2n
\end{aligned}
\tag{13}
$$

where κ is a scaling parameter and $\left(\sqrt{(n+\kappa)\mathbf{P_u}} \right)_i$ is the ith row or column of the square root of the matrix, $(n+\kappa)\mathbf{P_u}$. These sigma points are propagated through the nonlinearity $\mathbf{f}(.)$ to obtain

$$
\mathcal{V}_i = \mathbf{f}(\mathcal{U}_i) \quad \text{for } i = 0,1,\ldots,2n.
\tag{14}
$$

Using the set of $\mathcal{V}_i$, the mean ($\hat{\mathbf{v}}$) and covariance ($\mathbf{P_v}$) of the posterior density is estimated as

$$
\hat{\mathbf{v}} = \sum_{i=0}^{2n} W_i \mathcal{V}_i
\tag{15.a}
$$

$$
\mathbf{P_v} = \sum_{i=0}^{2n} W_i \left(\mathcal{V}_i - \hat{\mathbf{v}} \right) \left(\mathcal{V}_i - \hat{\mathbf{v}} \right)^T.
\tag{15.b}
$$

It is shown that the UKF based approximation is equivalent to a third order Taylor series approximation if the Gaussian prior is assumed (Julier & Uhlman, 2004). Another advantage of UT is that it does not require the calculation of the Jacobian or Hessian.

3.3.1 Scaled unscented transform

The scaled unscented transform (SUT) is a generalization of the UT. It is a method that scales an arbitrary set of sigma points yet capture the mean and covariance correctly. The new transform is given by

$$
\mathcal{U}_i' = \mathcal{U}_0 + \alpha \left(\mathcal{U}_i - \mathcal{U}_0 \right) \quad \text{for } i = 0,\ldots,2n
\tag{16}
$$

where α is a positive scaling parameter. By this the distribution of the sigma points can be controlled without affecting the positive definitive nature of the matrix, $(n+\kappa)\mathbf{P_u}$. A set of sigma points, $\{\mathbf{U} = [\mathcal{U}_0,\ldots,\mathcal{U}_{2n}], \mathbf{W} = [W_0,\ldots,W_{2n}]\}$, is first calculated using Eq.(13) and then transformed into scaled sigma points, $\{\mathbf{U}' = [\mathcal{U}_0',\ldots,\mathcal{U}_{2n}'], \mathbf{W}' = [W_0',\ldots,W_{2n}']\}$ by

$$\mathcal{U}_i^{'} = \mathcal{U}_0 + \alpha\left(\mathcal{U}_i - \mathcal{U}_0\right) \quad \text{for } i = 0,1,\ldots,2n \tag{17.a}$$

$$W_i^{'} = \begin{cases} \dfrac{W_0}{\alpha^2} + (1 - \dfrac{1}{\alpha^2}) & i = 0 \\[2ex] \dfrac{W_i}{\alpha^2} & i \neq 0 \end{cases}. \tag{17.b}$$

The sigma point selection and scaling can be combined to a single step by setting

$$\lambda = \alpha^2(n + \kappa) - n \tag{18}$$

and

$$\mathcal{U}_0^{'} = \hat{\mathbf{u}} \tag{19.a}$$

$$\mathcal{U}_i^0 = \hat{\mathbf{u}} + \left(\sqrt{(n + \lambda)\mathbf{P}_{\mathbf{u}}}\right)_i \quad i = 1,\ldots,n \tag{19.b}$$

$$\mathcal{U}_i^{'} = \hat{\mathbf{u}} - \left(\sqrt{(n + \lambda)\mathbf{P}_{\mathbf{u}}}\right)_i \quad i = n+1,\ldots,2n \tag{19.c}$$

$$W_0^{(m)} = \frac{\lambda}{n + \lambda} \tag{19.d}$$

$$W_0^{(c)} = \frac{\lambda}{(n + \lambda)} + (1 - \alpha^2 + \beta) \tag{19.e}$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(\lambda + n)} \quad \text{for } i = 1,2,\ldots,2n. \tag{19.f}$$

Parameter β above is another control parameter which affects the weighting of the zeroth sigma point for the calculation of the covariance. Using SUT, the mean and the covariance can be estimated as

$$\hat{\mathbf{v}} = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{V}_i^{'} \tag{20.a}$$

$$\mathbf{P}_{\mathbf{v}} = \sum_{i=0}^{2n} W_i^{(c)} \left(\mathcal{V}_i^{'} - \hat{\mathbf{v}}\right)\left(\mathcal{V}_i^{'} - \hat{\mathbf{v}}\right)^T \tag{20.b}$$

where $\mathcal{V}_i' = \mathbf{f}(\mathcal{U}_i')$.

Selection of κ is such that it should result in positive semi definiteness of the covariance matrix. $\kappa \geq 0$ guarantees this condition and a good choice is $\kappa = 0$. Choose $0 \leq \alpha \leq 1$ and $\beta \geq 0$. For Gaussian prior density, $\beta = 2$ is an optimal choice. Since, α controls the spread of the sigma points, it is selected such that it should not capture the non-local effects when nonlinearities are strong.

3.3.2 Unscented Kalman filter

UKF is an application of the SUT. It implements the minimum mean square estimates as follows. The objective is to estimate the states $\mathbf{x}_k$, given the observations, $\mathbf{y}_{1:k}$. For this, the state variable is redefined as the concatenation of the original state and noise variables (i.e. $\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_k^T & \mathbf{w}_k^T & \mathbf{v}_k^T \end{bmatrix}^T$ with dimension n_a). The steps involved in UKF are listed below. First, we initialize the parameters

$$\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0] \tag{21.a}$$

$$\mathbf{P}_0 = \mathbb{E}\left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T\right] \tag{21.b}$$

$$\hat{\mathbf{x}}_0^a = \begin{bmatrix} \hat{\mathbf{x}}_0^T & \mathbf{0} & \mathbf{0} \end{bmatrix}^T \tag{21.c}$$

$$\mathbf{P}_0^a = \begin{bmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix} \tag{21.d}$$

For $k = 1, 2, \ldots$ calculate the sigma points:

$$\mathcal{X}_{k-1}^a = \begin{bmatrix} \hat{\mathbf{x}}_{k-1}^a & \hat{\mathbf{x}}_{k-1}^a \pm \sqrt{(n_a + \lambda)\mathbf{P}_{k-1}^a} \end{bmatrix} \tag{22}$$

Time update:

$$\mathcal{X}_{k|k-1} = \mathbf{f}\left(\mathcal{X}_{k-1}^x, \mathcal{X}_{k-1}^w\right) \tag{23.a}$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n_a} W_i^{(m)} \mathcal{X}_{i,k|k-1}^x \tag{23.b}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n_a} W_i^{(c)} \left[\mathcal{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_{k|k-1}\right]\left[\mathcal{X}_{i,k|k-1}^x - \hat{\mathbf{x}}_{k|k-1}\right]^T \tag{23.c}$$

$$\mathcal{Y}_{k|k-1} = \mathbf{h}\left(\mathcal{X}_{k|k-1}^x, \mathcal{X}_{k|k-1}^v\right) \tag{23.d}$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2n_a} W_i^{(m)} \mathcal{Y}_{i,k|k-1} \tag{23.e}$$

Measurement update:

$$\mathbf{P}_{\hat{\mathbf{y}}_k \hat{\mathbf{y}}_k} = \sum_{i=0}^{2n_a} W_i^{(c)} \left[\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}\right]\left[\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_{k|k-1}\right]^T \tag{24.a}$$

$$\mathbf{P}_{\hat{\mathbf{x}}_k \hat{\mathbf{y}}_k} = \sum_{i=0}^{2n_a} W_i^{(c)} \left[\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]\left[\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right]^T \tag{24.b}$$

$$\mathbf{K}_k = \mathbf{P}_{\hat{\mathbf{x}}_k \hat{\mathbf{y}}_k} \mathbf{P}_{\hat{\mathbf{y}}_k \hat{\mathbf{y}}_k}^{-1} \tag{24.c}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1} \right) \tag{24.d}$$

$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{\hat{\mathbf{y}}_k \hat{\mathbf{y}}_k} \mathbf{K}_k^T \tag{24.e}$$

It is shown in (Julier & Uhlman, 2004) that the approximation introduced by the UKF has more number of Taylor series terms. The effect of the approximation errors is different for different nonlinear systems. In some cases, if the nonlinearity is quadratic, approximation error will not have any strong influence.

4. Results and discussion

To assess the performance of the EKF and UKF based synchronization schemes, simulation studies are carried out on three different chaotic systems/maps[1]: (*i*) Ikeda map (IM), (*ii*) Lorenz system, and (*iii*) Mackey-Glass (MG) system. The Lorenz system is a three dimensional vector field, $\phi(x,y,z): R^3 \to R^3$, representing the interrelation of temperature variation and convective motion. The set of coupled differential equations representing the Lorenz system is given by

$$\dot{x}(t) = \sigma(y(t) - x(t)) \tag{25.a}$$

$$\dot{y}(t) = -x(t)z(t) + rx(t) - y(t) \tag{25.b}$$

$$\dot{z}(t) = x(t)y(t) - cz(t) \tag{25.c}$$

where $\sigma = 10$, $r = 28$ and $c = 8/3$ are used to obtain the Lorenz attractor. The three states (x, y and z) are randomly initialized and the system of differential equation is solved with the fourth order Runge-Kutta method. The Lorenz attractor is shown in Fig. 4.
Ikeda map represent a discrete dynamic system of pumped laser beam around a lossy ring cavity and is defined as

$$z_{k+1} = p + B z_k \exp\left[\sqrt{-1}\left(\phi - \frac{\omega}{1 + |z_k|^2} \right) \right] \tag{26}$$

where z_k is a complex-valued state variable with $z_k = x_k^R + \sqrt{-1} x_k^I$. Here, x_k^R is $\Re\{z_k\}$ and x_k^I is $\Im\{z_k\}$. $\Re\{.\}$ and $\Im\{.\}$ give the real and imaginary parts of a complex variable, respectively. For the set of parameters $p = 0.92$, $B = 0.9$, $\phi = 0.4$, and $\omega = 6$, the attractor of this map is shown in Fig. 5.
The Mackey-Glass system was originally proposed as a first order nonlinear delay differential equation to describe physiological control systems. It is given by

---

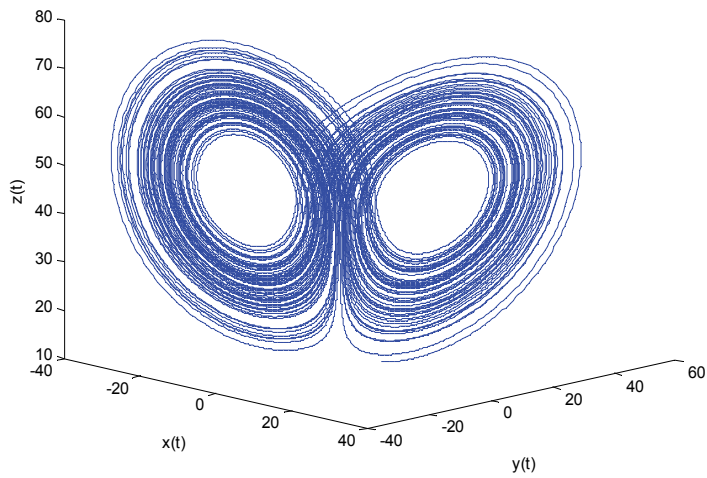[1] Map is used to represent discrete dynamic systems. i.e. $f : X \to X$

Fig. 4. Lorenz attractor.

$$\dot{x}(t) = -ax(t) + \frac{bx(t-\tau)}{1 + x(t-\tau)^{10}} \tag{27}$$

This system is chaotic for values of $a = 0.1$ and $\tau \geq 17$. Figure 6 shows the Mackey-Glass attractor with $\tau = 17$.
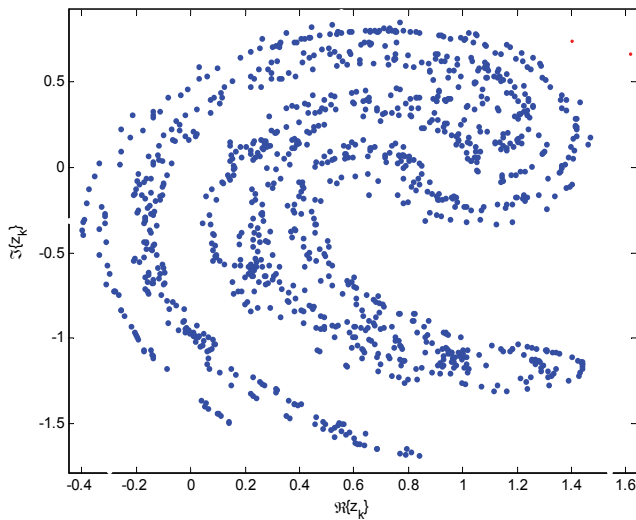

Fig. 5. Ikeda map attractor.

These systems/maps have distinct dynamical properties and well suited for our analysis. Lorenz system is one of the archetypical chaotic systems commonly studied for chaotic

synchronization. Ikeda map, on the other hand, has higher order derivatives and is an appropriate candidate for studying the effect of high nonlinearity in filter based synchronization. An interesting feature of Mackey-Glass system is that its complexity (i.e. the correlation dimension) increases as τ increases.
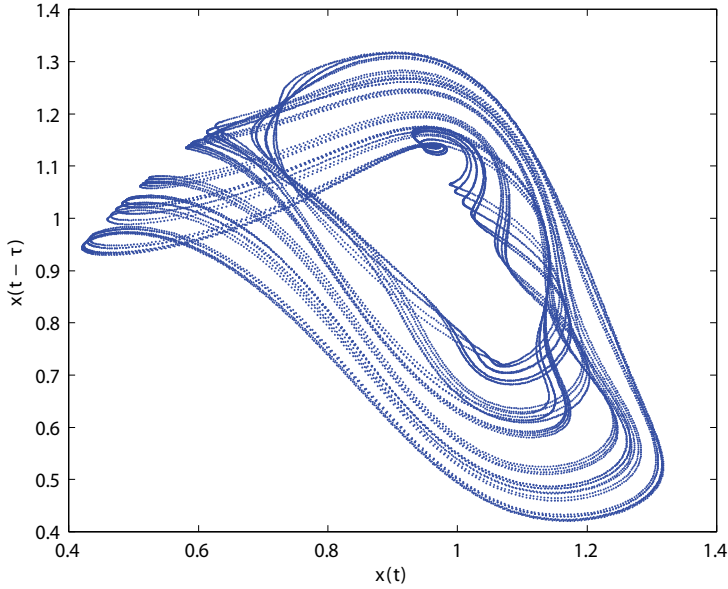
Fig. 6. Mackey-Glass Attractor.

We consider a typical situation where one of the state variables, x_k (for continuous system, it is assumed that we discretize the state variable using appropriate techniques) which is corrupted by channel noise is used for the synchronization. In all the computer simulations, the signal-to-noise ratio (SNR) which is defined as

$$\text{SNR} = \frac{\frac{1}{N}\sum_{k=1}^{N} x_k^2}{\sigma_w^2} \tag{28}$$

where σ_w^2 is the variance of the noise and N is the total number of samples used for evaluation, is varied from -5dB to 50dB for the Lorenz and MG systems and in the case of IM, it is varied from 35dB to 60dB. We define two performance evaluation quantities: the normalized mean square error (NMSE) and the total normalized mean square error (TNMSE) as

$$\text{NMSE}^i = \frac{\sum_{k=1}^{N}(x_k^i - \hat{x}_k^i)^2}{\sum_{k=1}^{N}(x_k^i)^2}, \tag{29}$$

$$\text{TNMSE} = \sum_{i=1}^{n} \text{NMSE}^i \ , \tag{30}$$

respectively. While NMSE gives an idea about the recovery of observed state variable, TNMSE gives how faithfully, the attractor can be reconstructed.

We run numerical simulations on all these three systems and the results are presented here. Figures 7 to 9 show the result of NMSE performances. We restricted the SNR from 35dB to 60dB in the case of IM system due to the observed divergence when both EKF and UKF are used. It is shown in (Kurian, 2006) that since the IM has non-hyperbolic chaotic attractors, this divergence is inevitable. For all other systems, we changed the SNR from -5dB to 50dB. We can see that for all the maps and systems we considered for the analysis, the NMSE is monotonically decreasing with the increase in SNR. The steady decrease in NMSE shows that there are no residual errors. Also, we can see that the UKF based synchronization gives very good NMSE performance due to the better approximation capabilities of the UKF. In the case of MG system, we can see that the EKF nearly fails in synchronizing at all SNRs which is again due to higher approximation error.

We computed the TNMSE values for the Lorenz system and Ikeda Map (Fig. 10 & 11). As mentioned before, the TNMSE values indirectly reflect how well the chaotic attractors could be reconstructed. The TNMSE values for Lorenz systems and Ikeda Map are always greater than the NMSE values at all the SNR values considered due to the error introduced by other state variables. We can see that this increase in error is comparable and behaves exactly like the NMSE: it gradually reduces with an increase in the SNR values. Similarly, the UKF shows better performance for all the SNR values considered.
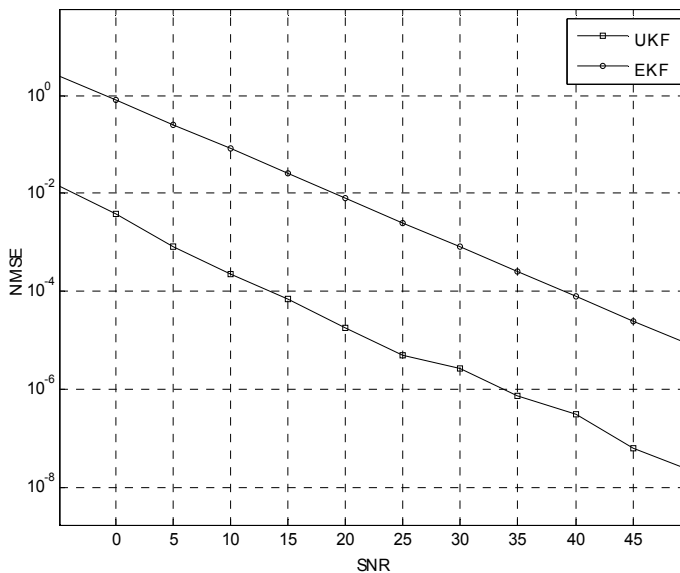

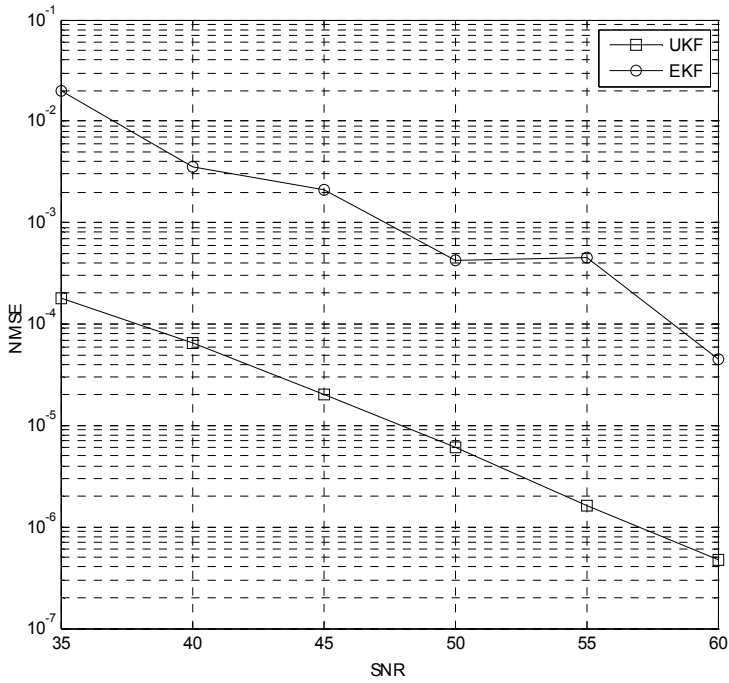Fig. 7. NMSE of UKF and EKF based synchronization schemes for Lorenz.

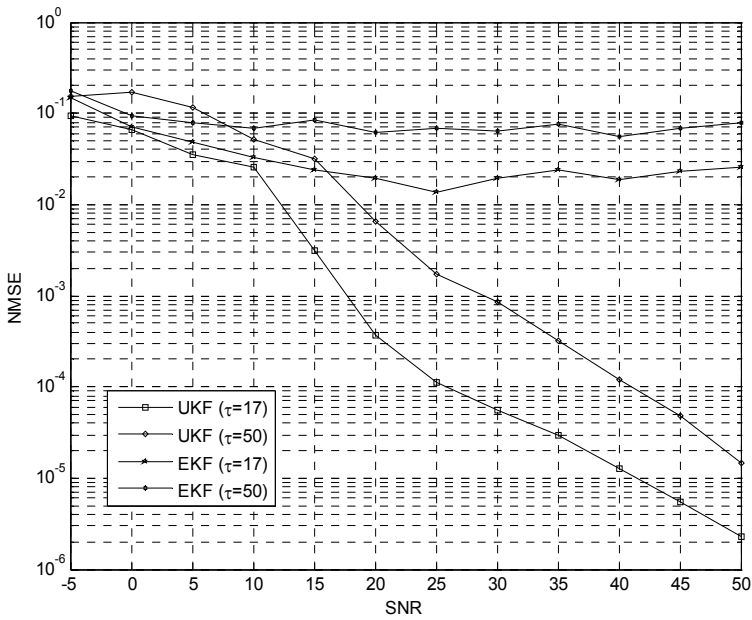Fig. 8. NMSE of UKF and EKF based synchronization schemes for Ikeda map.

Fig. 9. NMSE of UKF and EKF based synchronization schemes for MG systems.
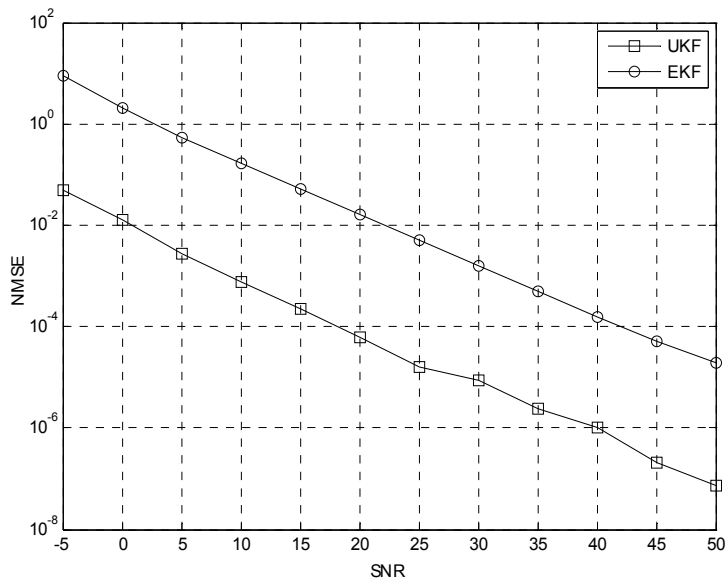
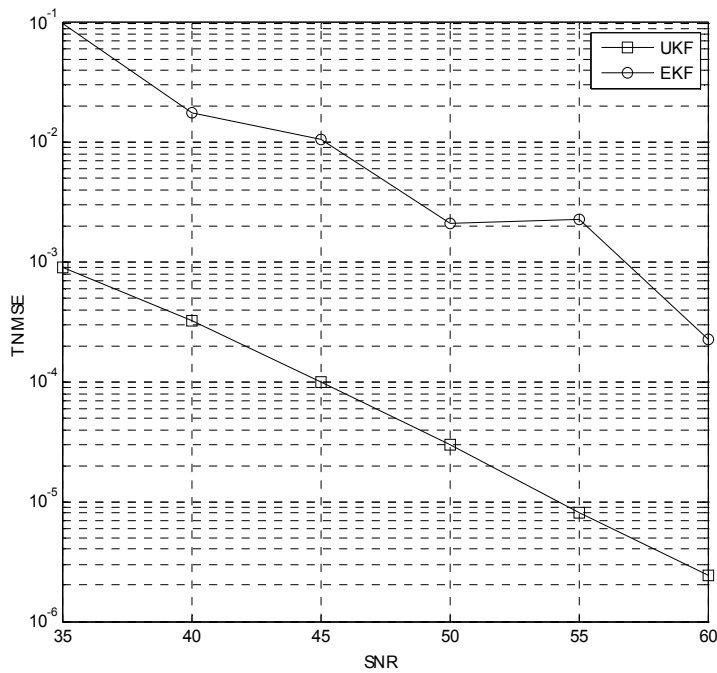Fig. 10. TNMSE of UKF and EKF based synchronization schemes for Lorenz system.

Fig. 11. TNMSE of UKF and EKF based synchronization schemes for Ikeda map.

5. Conclusions

Chaotic systems are simple dynamic systems which can display very complex behavior. One of the defining characteristics of such systems is the sensitive dependence on initial conditions and hence synchronization of such systems possesses certain amount of difficulties. This task will be even more formidable when the channels as well as the measurement noises are present in the system. Stochastic methods are applied to synchronize such chaotic systems. EKF is one of the most widely investigated stochastic filtering methods for chaotic synchronization. However, for highly nonlinear systems, EKF introduces approximation errors causing unacceptable degradation in the system performance. We consider UKF, which has better approximation error characteristics for chaos synchronization and show that it has better error characteristics.

6. References

R. L. Devaney (1985). *An introduction to chaotic dynamical system.* The Benjamin Cummings Publishing Company Inc.

M. P. Kennedy, R. Rovatti, and G. Setti (2000). *Chaotic electronics in telecommunications*. CRC Press.

Fujisaka, H. & Yamada (1983). Stability theory of synchronized motion in coupled oscillator systems. *Progressive Theory of Physics*. Vol. 69, No. 1, pp. 32–47.

Pecora, L. M. & Carroll, T. L. (1990). Synchronization in chaotic systems. *Physical Review Letters*. Vol. 64, pp. 821–824.

Ogorzalek, M. J. (1993). Taming chaos–Part I: Synchronization. *IEEE Trans. Circuits Syst.–I*. vol. 40, No. 10, pp. 693–699.

Sushchik, M. M.; Rulkov, N. F. & Abarbanel H. D. I. (1997). Robustness and stability of synchronized chaos: An illustrative model. *IEEE Trans. Circuits Syst.–I*, vol. 44, pp. 867–873.

Nijmeijer, H. & Mareels, I. M. Y. (1997). An observer looks at synchronization. *IEEE Trans. Circuits Sys.–I*, vol. 44, No. 10, pp. 882–890.

Arulampalam, S.; Maskell, S. ; Gordon, N. & Clapp, T. (2001). A tutorial on particle filters for on–line non–linear/non–Gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188.

Grewal, M. S. & Andrews A. P. (2001). *Kalman filtering: Theory and practice using MATLAB*, 2nd Ed., John Wiley & Sons.

Kurian, A. (2006). *Performance analysis of filtering based chaotic synchronization and development of chaotic digital communication schemes*. PhD Dissertation, National University of Singapore.

Julier, S. J. & Uhlman, J. K. (2004). Unscented Kalman filtering and nonlinear estimation. *Proc. IEEE*, vol. 92, pp. 401–421.

Eric, A. W. & van der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation, *Proc. IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*. Lake Louise, Alberta, Canada, Octobre.

Julier, S. J. & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. *Proc. of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Control.*

H. D. I. Abarbanel, *Analysis of observed chaotic data*. Springer, USA, 1996.

Human Motion Tracking Based on Unscented Kalman Filter in Sports Domain

GuoJun Liu and XiangLong Tang
Harbin Institute of Technology
China

1. Introduction

This work aims to automatically track the movements of ice skaters on a large-scale complex and dynamic rink. From the computer vision point of view, several open challenging problems in the tracking of sports athletes present themselves (Intille & Bobick, 1994): the skaters move rapidly and change direction unpredictably; the estimation of skaters' motions is compounded with non-smooth camera motion; skaters range in size from 30 by 30 pixels to about 70 by 70 pixels, depending on the setting of the camera; competitors flail arms and legs frequently during the match. Additionally, partial occlusions often occur when skaters are close or skaters collide during the overtaking in the short track speed skating games. A one-second-long sequence of a single skater is shown in Fig. 1.

Fig. 1. A one-second-long sequence of a single skater.

There is a substantial literature on tracking for various purposes including video surveillance, smart environments, pedestrian, face tracking, action recognition in sports and many other domains. Haritaoglu et al. (Haritaoglu et al., 2000) use a variation of the background to detect object in the foreground. Rigoll et al. (Rigoll et al., 2004) use stochastic modeling techniques, Pseudo-2D Hidden Markov Models (P2DHMMS) and Kalman filter to estimate the location of a person. In (Ozyildiz et al., 2003; Rasmussen & Hager, 2001; Yilmaz et al., 2004), a technique of fusing multiple cues (e.g. texture, color and shape) is employed to track. The mean shift algorithm (Comaniciu et al., 2003; Yang et al., 2005) has achieved considerable success in object tracking due to its simplicity and robustness, it finds local minima of a similarity measure between the color histograms or kernel density estimates of

the model and the target. Recently, the integration of color distributions into particle filter becomes popular due to the outstanding performance on tracking the non-rigid objects with the non-linear and non-Gaussian motion (Nummiaro et al., 2003). But these well-known traditional tracking algorithms are inadequate for tracking the low-resolution, amorphous, fast erratically moving and colliding skaters in a large-scale, complex and dynamic scene by using a single panning camera.

The remainder of this paper is organized as follows: the next section explores the related work in various other sports fields. Section 3 overview of our system, Section 4 explains how to automatically compute the mappings that transform each frame to the model of the rink and section 5 describes our tracking method which combines the unscented Kalman filter with a hierarchical model based on contextual knowledge. Experiments and results are given in section 6 and the conclusion and discussion in section 7.

2. Related work

As the most popular sport in the world, soccer has enormous potential market value. The related applications, such as broadcast or annotation systems, are receiving more and more attention from computer vision researchers. Misu et al. (Misu et al., 2002) address the problem of tracking soccer players in occlusion by fusing different objects features like textures, color statistics, movement vectors, etc. In (Vandenbroucke et al., 2003), color image segmentation by using pixel classification in an adapted hybrid color space is applied to extract those meaningful regions that represent the players and recognize their team. Needham and Boyel (Needham & Boyel, 2001) describe a CONDENSATION based approach on image sequences obtained from a single fixed camera, Kalman filter and ground plane information are used to improve the prediction of player movement and assist to track occluded players. Junior and Anido (Junior & Anido, 2004) develop a real-time distributed system by using five static cameras to get the whole scene, each player can be tracked individually by an independent module of the system, and each module can apply different tracking techniques depending on specific visual characteristics.

There are many computer vision systems for other sports domains. Pingali et al. (Pingali et al., 1998) develop a real time tracking technology for enhancing the broadcast of tennis matches from stationary cameras. Recently, Yan et al. (Yan et al., 2006) propose a data association algorithm to track a tennis ball in low-quantity tennis video sequences. Pers et al. use two stationary cameras mounted directly above the court and propose a new approach for modeling the radial image distortion more accurately. The tracking algorithm combined with color feature and the template is exploited to track the player, using color feature is to avoid a drift caused by the template tracking. Their systems are applied to many sports domains including squash (Pers et al., 2001), handball (Pers et al., 2002) and basketball (Jug et al., 2003). But there are two limitations in their works: first, the cameras must be placed above the playing court, which is a rigorous condition for regular league or championship matches. Second, how to handle occlusion in tracking process seems not to be solved.

Intille and Bobick (Intille & Bobick, 1994) develop the state-of-art automatic annotation system for American football footage and lay a foundation for research in the automatic annotation of video. In their system, camera motion is to be recovered by using a global model of the football field which consists of some geometrical primitives (e.g. lines) and some features (e.g. number, logo). The "close-world" is defined as "A region of space and

time in which the specific context is adequate to determine all possible objects present in that region", the tracking is performed on those player's pixels extracted from motion blobs by "close-world" analysis. Okuma (Okuma, 2003) develops a hockey annotation system to automatically analyze hockey scenes, track hockey players in these scenes and construct a visual description of these scenes as trajectories of those players. Their system has two components: one is rectification system that transforms the original sequence in broadcast video to the globally consistent map of the hockey rink. The other is a color-based sequential Monte Carlo tracker. Robustness and reliability of this system are shown that the tracker tracks a single target for about 500 frames and the error between the estimated position and the real world position is from 0.3 m to 1 m. Compared with (Okuma, 2003), the framework of our system is similar, but our method is very different from theirs in many aspects such as camera plan, registration and tracking algorithm.

3. Overview of our system

Input video sequence

Registration subsystem	Tracking subsystem
Extract SIFT features and calculate homography between two neighborhood frames	Homography can remove the camera motion
Choose the frame in central angle of view of the whole rink and compute several reference frames	Obtain hierarchical model when the player skates through the different sub-regions of the rink
Generate the panorama of the rink and map the panorama to the rink model manually	Tracking method ...
Calculate homography transforming the current frame to the model of the real rink	Trajectory of athlete in the image coordinate

Spatio-temporal trajectory of the real world stored in the database

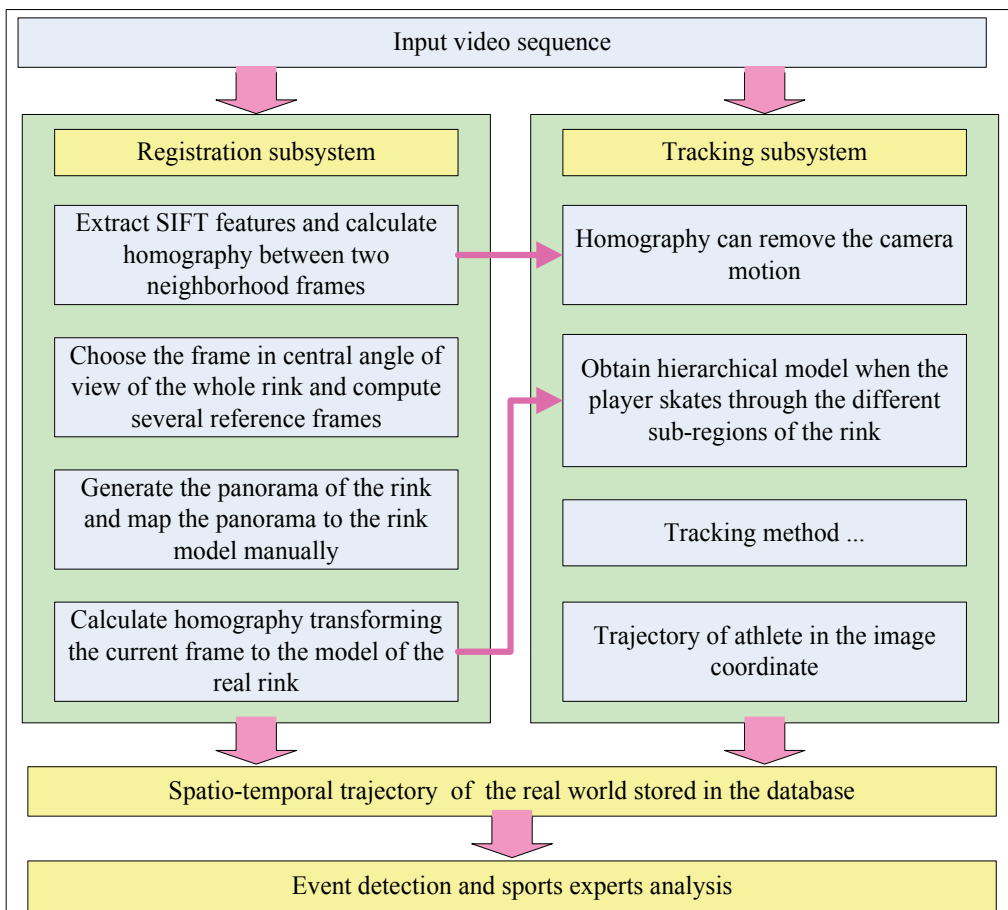| Event detection and sports experts analysis |
| --- |

Fig. 2. The architecture of our system.

Our computer vision system aims to automatically track the movements of skaters on a large-scale complex and dynamic rink. Our aim is to exploit it not only in daily training but also in competitions. We used a single panning camera, which was mounted at the top auditorium of the stadium as close as possible to the center in order to reduce the projection error. Due to little texture information on the rink, unlike (Intille & Bobick, 1994; Okuma, 2003), zooming was abandoned, it can make recording the high-speed target more difficult and enlarge the error of lens distortion. Though the camera center moves by a small amount, due to an offset from the camera's optical center, the approximation of pure rotation is indeed sufficient such as proven in (Hayman & Murray, 2003).

The architecture of our system is shown in Fig. 2, it includes two kernel subsystems. Automatic registration affects directly not only the accuracy of the system's output but also the tracking performance, it can provide two kinds of outputs to other modules: (1) The homography between two neighborhood frames can remove the camera motion to improve the precision of the motion prediction in the process of tracking. (2) Another homography transforming the current frame to the model of the real rink is associated with the output of tracking subsystem, such as the skater's spatio-temporal trajectories in the real world, and that can be stored in the database. At the same time, it can be used in the tracking process to obtain a skater's hierarchical model when the skater moves through different sub-regions of the rink. To improve the tracking performance, the proposed tracking subsystem incorporates the hierarchical model based on contextual knowledge and multiple cues into the unscented Kalman filter, the details will be discussed in section 5.

4. Automatic registration

The goal of automatic registration for sports applications is transforming the positions in the video frame to the real-world coordinates or vice versa. Generally, the registration needs many point or line features extracted automatically from images, then matches them and uses the correspondence to calculate the homography. Farin et al. (Farin et al., 2004) use a model of the arrangement of court lines for registration, similar to the one in (Hayet et al., 2004). However, if no obvious lines are in the fields or courts, it can't work well. Okuma et al. (Okuma, 2003) compute the local displacements of image features by using the Kanade-Lucas-Tomasi (KLT) tracker (Shi & Tomasi, 1994; Tomasi & Kanade, 1991) and determine the local matches, but it needs to predict the current camera motion based on the previous camera motion to reduce the amount of local feature motion, if the camera moves rapidly and asymmetrically such as in our application, the worse prediction can lead to the KLT tracker failing.

Therefore, for a rapid camera, the registration faces two big difficult problems: (1) Detect the more distinctive point features and match them better. (2) How to reduce the accumulative registration error for a long image sequence.

4.1 Homography

A homography is a projective transformation which is a nonsingular matrix H with size 3×3. If x and $x^{'}$ are the images of a world point, belonging to a plane, they are related by a matrix H corresponding to that plane: $x' = Hx$, since the matrix H has 8-Degree of Freedom (DOF), 4 point correspondences can determine H. Obviously, with non-perfect data, more points should be used.

In general, the moving targets in the image can decrease the accuracy of the homography since the good features on the moving ones are imperfect. The RANSAC algorithm (Fischler & Bolles, 1981; Hartley & Zisserman, 2000) can pick out the worse features (namely outliers) easily, as shown in Fig. 3. However, it can only assure the computational accuracy between two adjacent frames, not for a long video sequence. How to reduce the accumulative error is the key issue.
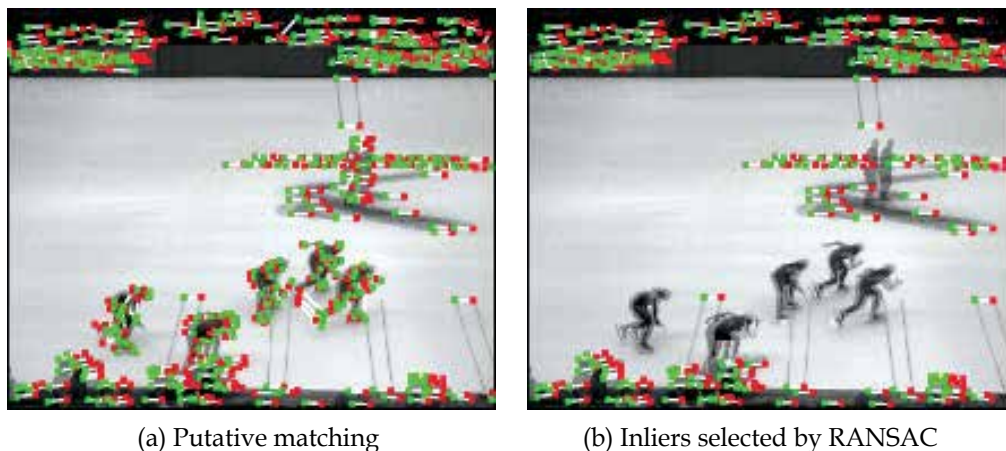

<table>
<tr><td>(a) Putative matching</td><td>(b) Inliers selected by RANSAC</td></tr>
</table>

Fig. 3. The outliers on the moving skaters and the referee are eliminated.

4.2 Selection of point features

Feature extraction and matching determine the computational precision of the homography and the overall reliability of the registration algorithm. Two of the most popular feature extractors are the Harris corner detector (Harris & Stephens, 1988) followed by the Sum of Squared Difference (SSD) matching, and the Kanade-Lucas-Tomasi (KLT) tracker (Shi & Tomasi, 1994; Tomasi & Kanade, 1991). These methods can work well when the baseline is relatively small and the appearance of the features doesn't change too much across the subsequences. A more distinctive feature is desirable for matching two wide baseline images (e.g. a large translation, scaling, or rotation between two frames) such as the current frame and its corresponding reference frame in our application.

Lowe (Lowe, 2004) proposed the Scale Invariant Feature Transformation (SIFT), where a feature's location and scale are determined by extrema of a Difference of Gaussian (DOG) function in the scale space, and its dominant, local image gradient orientation. It is invariant to viewpoint changes, large geometric transformation, changes in illumination, and has been applied in the areas of object recognition (Sivic & Zisserman, 2003) and panorama stitching (Brown & Lowe, 2003). Therefore, it is suited to dealing with the problem of wide baseline matching.

4.3 Registration method for a long image sequence

In order to reduce the accumulation of the errors generated from the set of frame-to-frame homography, many reference frames are introduced and calculated to construct a panoramic image (Brown & Lowe, 2003; Shum & Szeliski, 2000) as illustrated in Fig. 4 and

each frame can be mapped to the most adjacent reference frame. The model of the entire rink is shown at the bottom in Fig. 4, which includes precise measurements of geometrical features: start line, finish line and marking blocks, and all that can be obtained from ISU (International Skating Union, http://www.isu.org/). The corresponding points labeled from 1 to 20 are initialized manually, the detailed procedure is illustrated in Algorithm 1.

Algorithm 1 Registration method for a long image sequence

Input video sequence and perform the following steps:

1. Compute homography $H_{t-1,t}$ between the frame at time $t-1$ and the frame at time t with the RANSAC algorithm (Hartley & Zisserman, 2000).
2. Choose the frame in the central angle of view of the whole rink as a base frame used to construct the panorama.
3. Compute the reference frames which distribute on the both sides of the central frame at regular intervals.
4. Generate the panorama of the rink with all reference frames and calculate homography $H_{ref^i,pano}$ transforming the ith reference frame to the panorama.
5. Map the panorama to the rink in the real world by selecting 20 corresponding points manually shown in Fig. 4 and obtain homography $H_{pano,rink}$.
6. Compute homography H_{t,ref^i} mapping the frame at time t to the corresponding reference frame.
7. Obtain homography $H_{t,rink}$ mapping the frame at time t to the rink in the real world

$$H_{t,rink} = H_{pano,rink} \cdot H_{ref^i,pano} \cdot H_{t,ref^i}.$$

---

5. Tracking method

5.1 Hierarchical model

During the hot short track race, the competitors skate rather quickly about less than 10 seconds one loop (110 meters) and the camera pans rapidly in order to capture them. Therefore, the perfect tracking performance seems to be more and more difficult due to the following reasons: (1) The skater's size in the video sequence changes violently. (2) While skating through the top straight on the ice, the skaters are very close to the miscellaneous advertisements attached to the protection board, its color is similar to the one of the skater's clothing sometimes. (3) Occlusions among high speed skaters often happen on the curves of the rink.

All the above challenge the traditional tracking methods, which can't work perfectly. Therefore, to overcome these problems, more contextual knowledge and multiple cues are to be introduced into the tracker as follows: (1) In our application, the short track skater can be considered as a hierarchical model, it consists of two block components: one is the skater's helmet which is a small block, the other bigger one is the body. The relations of their positions are varied due to the skater's diversified postures, when the skater runs through different sub-regions in the rink, such as straight bottom, right curve or left curve, as illustrated in Fig. 5. On considering the camera plan and the skaters' occlusions in the practical application, the helmet block is more discriminative and more reliable than the

body's. Hence, the skater's model including the helmet and the body has been extracted as the prior knowledge, which can effectively reduce the error during the update of the scale of the model. Inaccurate update can cause the tracking to fail when the target model size changes dramatically. (2) Compared with a single-part model, the hierarchical model can reduce the impact of the interference color coming from the advertisement board. At the same time, the integration of the template matching approach (for helmet model) and the color histogram matching method (for body model, details in the section 5.2) is used to solve the problem of occlusions, the benefit of it is to make the tracker more robust under complex environments.
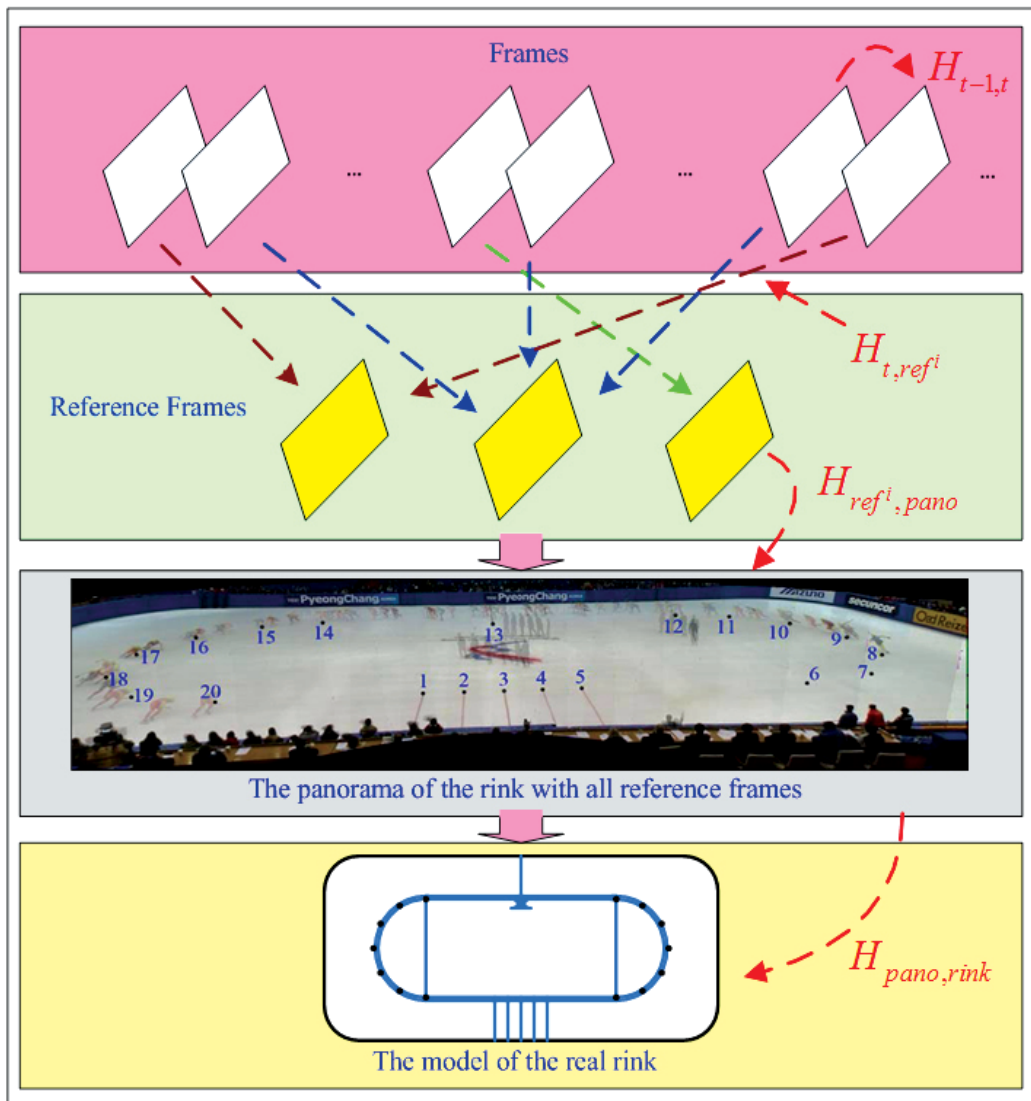
Fig. 4. The flowchart of the registration method for a long image sequence.
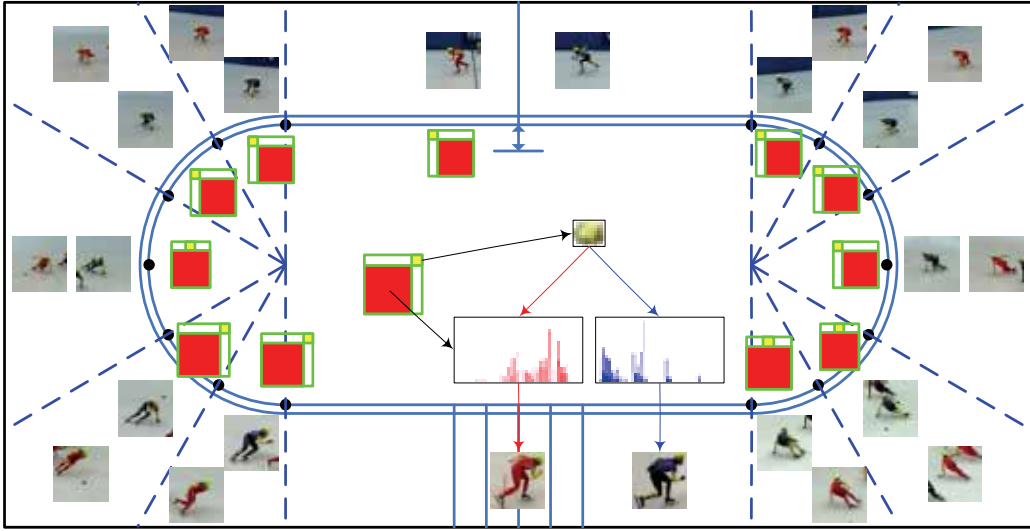
Fig. 5. There are two skaters coming from different countries on the rink. The rink can be divided into 12 sub-regions by the blue dashed lines according to the skater's diversified size and the black markers on the both sides curves. The skater's model of each sub-region differs in its size and the relative position of the helmet and body. The skater's model can be considered as a hierarchical model and represented by two blocks, the smaller yellow block denotes the helmet, the bigger red one denotes the body.

5.2 Color histogram matching method

A skater's body as a target is represented by a rectangle region, its color model can be obtained by using histogram techniques which achieve robustness against non-rigidity, rotation, and partial occlusion (Nummiaro et al., 2003). Let $\{x_i\}_{1,\dots,N}$ be the pixel locations of the target represented by a rectangle region, the region is centered at $\mathbf{z}$ and r is the length of its half diagonal. The function $b(x_i)$ ($\mathbb{R}^2 \to \{1,\dots,m\}$) assigns the color feature at location x_i to the corresponding bin. To increase the reliability of the color histogram, a weighting function $k(\|x\|^2)$ is employed to assign smaller weights to the boundary pixels farther from the center since those pixels are often affected by occlusions and interference from the background, and denoted as

$$k(\|x\|^2) = \begin{cases} c(1-\|x\|^2) & \|x\|^2 < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where c is a constant, for simple implementation, $c = 1$, if $c = \frac{1}{2} c_d^{-1}(d+2)$, c_d is the volume of the unit d-dimensional sphere, $k(\|x\|^2)$ is the Epanechnikov kernel (Comaniciu et al., 2003). The color histogram $p = \{p_u(z)\}_{u=1\dots m}$ at location z is then calculated as

$$p_u(z) = C \sum_{i=1}^{N} k\left(\left\| \frac{z - x_i}{r} \right\|^2 \right) \delta \left[b(x_i) - u \right] \tag{2}$$

where δ is the Kronecker delta function, C is a normalization constant and derived by imposing the condition $\sum_{u=1}^{m} p_u = 1$.

In our application, the Hue Saturation Value (HSV) color space ($8 \times 8 \times 4$ bins) was used to reduce sensitivity to lighting conditions. To avoid the interference from the background (i.e. the rink), those pixels belonging to the rink are to be kept away from the calculation of the histogram. The function $m(x_i)$ is considered as a mask to filter the rink, and defined as

$$m(x_i) = \begin{cases} 1 & S(x_i) > \tau_s \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $S(x_i)$ is the value of Saturation at location x_i, τ_s is a threshold and $\tau_s = 0.15$ in our experiments. Therefore, the new color histogram is calculated by

$$p_u(z) = C_m \sum_{i=1}^{N} m(x_i) k \left(\left\| \frac{z - x_i}{r} \right\|^2 \right) \delta \left[b(x_i) - u \right] \tag{4}$$

where C_m is a normalization constant.

The Bhattacharyya coefficient (Aherne et al., 1997) as a popular similarity measure between two color histograms $p = \{ p_u \}_{u=1...m}$ and $q = \{ q_u \}_{u=1...m}$ is denoted by

$$\rho[p, q] = \sum_{u=1}^{m} \sqrt{p_u q_u} \tag{5}$$

The larger ρ is, the more similar the two color histograms. $\rho = 1$ means a perfect match if and only if the two color histograms are identical.

The benefit of our new color histogram is shown in Fig. 6. (a) includes a skater with red clothing, (f) includes the one with blue clothing and is very different from (a), (k) includes the same skater as (a) distracted by another skater with blue clothing, (k) is more similar to (a) than (f) apparently. As a similarity measure, the Bhattacharyya coefficient between (a) and (k) should be larger than the one between (a) and (f). However, calculated the Bhattacharyya coefficients by using the traditional histogram and the weighted histogram, $\rho[p_c, p_m] = 0.677$ and $\rho[p_d, p_n] = 0.689$ are less than $\rho[p_c, p_h] = 0.684$ and $\rho[p_d, p_i] = 0.704$, respectively. Compared with them, our result is that $\rho[p_e, p_o] = 0.769$ is far larger than $\rho[p_e, p_j] = 0.092$, obviously, our result is better.

5.3 The proposed tracking method

The unscented Kalman filter (UKF) was introduced by Julier (Julier et al., 1995; Julier & Uhlmann, 1997) to address the nonlinear state estimation in control theory as a recursive minimum mean square error estimator. Compared with the extended Kalman filter (EKF), the UKF does not approximate the non-linear process and observation model, it uses a set of sigma points to capture the mean and covariance of the system and propagates these sigma points through the dynamic and measurement models without linearization, the UKF is superior to the EKF both in theory and in many practical applications (van der Merwe et al., 2000; Chen et al., 2006).
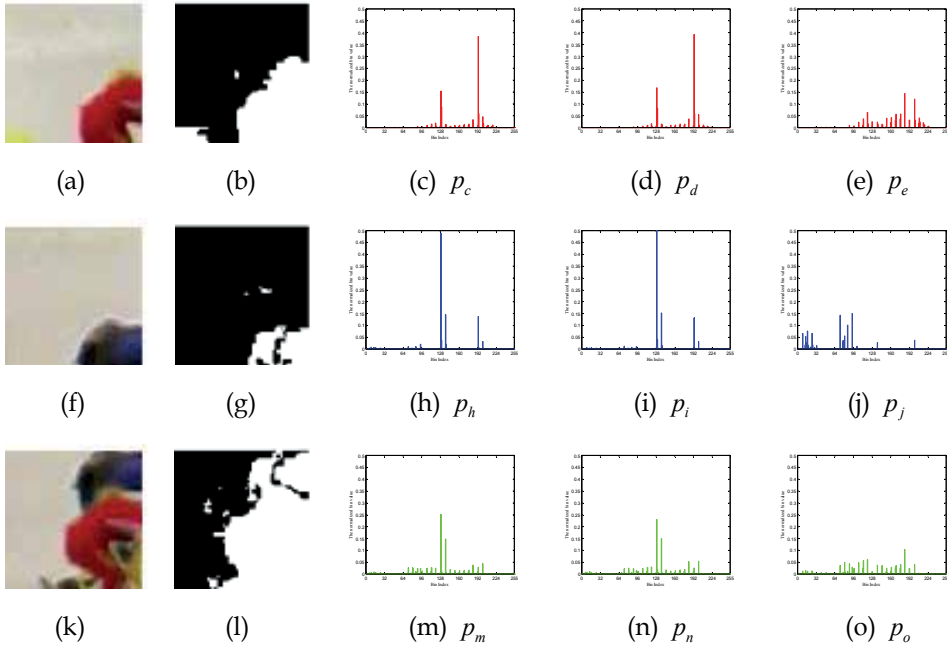
(a)   (b)   (c) p_c (d) p_d (e) p_e

(f)   (g)   (h) p_h (i) p_i (j) p_j

(k)   (l)   (m) p_m (n) p_n (o) p_o

Fig. 6. The benefit of our new color histogram. The first column, (a), (f) and (k) are three different target regions, the second column the results of filtering the rink, the third column the traditional histogram, the fourth column the weighted histogram given by Eq. (2), the last column the new proposed histogram calculated by Eq. (4).

The general state-space model is made up of a state transition $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and state measurement model $p(\mathbf{y}_t \mid \mathbf{x}_t)$, and the dynamic model can be denoted as follows:

$$\mathbf{x}_t = \mathrm{f}(\mathbf{x}_{t-1}) + \mathbf{u}_{t-1} \tag{6}$$

$$\mathbf{y}_t = \mathrm{h}(\mathbf{x}_t) + \mathbf{v}_t \tag{7}$$

where $\mathbf{x}_t \in \mathbb{R}^{n_x}$ represents the system state at time t and $\mathbf{y}_t \in \mathbb{R}^{n_y}$ the observation, $\mathbf{u}_t \in \mathbb{R}^{n_u}$ the process noise, and $\mathbf{v}_t \in \mathbb{R}^{n_v}$ the measurement noise. The mapping $\mathrm{f}(\cdot)$ denotes system state model and $\mathrm{h}(\cdot)$ denotes measurement model, respectively.

Here, the proposed tracker combining the UKF with a hierarchical model based on contextual knowledge is as follows:

State $\mathbf{x}_t$ and observation $\mathbf{y}_t$ can be expressed by:

$$\mathbf{x}_t = \left\{ \mathbf{S}_t^h, \mathbf{V}_{t-1} \right\} \tag{8}$$

$$\mathbf{S}_t^h = \left\{ Pos_x^h, Pos_y^h \right\}, \mathbf{V}_{t-1} = \mathbf{S}_{t-1}^h - H_{t-2,t-1} \mathbf{S}_{t-2}^h \tag{9}$$

$$\mathbf{y}_t = \left\{ \mathbf{S}_t^h \right\} \tag{10}$$

In our application, contextual knowledge (CK) as a control input in the observation process is very important for the proposed tracker, it can be consider as a set of internal parameters determined by $\mathbf{S}_t^h$.

$$\mathbf{CK}_t = \left\{ H_{t,rink}, \mathbf{M}_t^h, \mathbf{M}_t^b, PosRel_t^{hb} \right\} \tag{11}$$

$$\mathbf{M}_t^h = \left\{ \mathbf{S}_t^h, R_x^h, R_y^h \right\}, \mathbf{M}_t^b = \left\{ Hist, \mathbf{S}_t^b, R_x^b, R_y^b \right\}, \mathbf{S}_t^b = \left\{ Pos_x^b, Pos_y^b \right\} \tag{12}$$

where at time t, superscript h and b denote helmet and body, respectively, suffix x and y are the coordinates, $\mathbf{V}$ the velocity, R the length of block, Pos the position of the block center, and $PosRel^{hb}$ the relation of the helmet relative to the body shown in Fig. 5, and $PosRel_t^{hb} \in \{left, middle, right\}$. $Hist$ is the histogram of the skater's body model. Generally, the helmet model $\mathbf{M}_t^h$ is firstly determined by the template matching method, then $\mathbf{S}_t^b$ is calculated by $\mathbf{S}_t^h$ and $PosRel_t^{hb}$, and $Hist$ can be obtained inside a rectangle region which is computed by $\mathbf{S}_t^b$, R_x^b and R_y^b. Finally, $\mathbf{M}_t^b$ is determined.

Here, the mappings $f(\cdot)$ and $h(\cdot)$ can be expressed by the following equations

$$\mathbf{x}_t = f\left(\mathbf{x}_{t-1}, H_{t-1,t} \right) + \mathbf{u}_{t-1} = \mathbf{A} \left[H_{t-1,t} \left(\mathbf{x}_{t-1} \right) \right] + \mathbf{u}_{t-1} \tag{13}$$

$$\mathbf{y}_t = h\left(\mathbf{x}_t \right) + \mathbf{v}_t = \mathbf{C} \mathbf{x}_t + \mathbf{v}_t \tag{14}$$

where $H_{t-1,t}$ as a control input is used to remove the camera motion expressed by $\left[Pos_x \ Pos_y \ 1 \right]^T = H_{t-1,t} \left[Pos_x^h \ Pos_y^h \ 1 \right]^T$, ($Pos_x^h \in \mathbf{x}_{t-1}$ and $Pos_x^h \in \mathbf{x}_{t-1}$), then the value of Pos_x^h and Pos_x^h are replaced by Pos_x and Pos_x as $Pos_x^h = Pos_x, Pos_y^h = Pos_y$. $\mathbf{A}$ is the state transition matrix, $\mathbf{C}$ is a $n_y \times n_x$ matrix which relates the state to the measurement. The noise term $\mathbf{u}_t$ is a Gaussian random variable with zero mean and a covariance matrix $\mathbf{Q}$, i.e., its probability distribution is $p(\mathbf{u}) \sim N(0, \mathbf{Q})$, the covariance matrix $\mathbf{Q}$ is referred to as the process noise covariance matrix. Much like $\mathbf{u}_t$ for the process, $p(\mathbf{v}) \sim N(0, \mathbf{R})$ and $\mathbf{R}$ is measurement noise covariance matrix.

The UKF tracker can be initialized by:

$$\bar{\mathbf{x}}_0 = E\left[\mathbf{x}_0 \right] \tag{15}$$

$$\mathbf{P}_0 = E\left[\left(\mathbf{x}_0 - \bar{\mathbf{x}}_0 \right) \left(\mathbf{x}_0 - \bar{\mathbf{x}}_0 \right)^T \right] \tag{16}$$

$$\lambda = \alpha^2 \left(n_x + \kappa \right) - n_x \tag{17}$$

$$W_0^{(m)} = \lambda / \left(n_x + \lambda \right) \tag{18}$$

$$W_0^{(c)} = \lambda / \left(n_x + \lambda \right) + \left(1 - \alpha^2 + \beta \right) \tag{19}$$

$$W_i^{(m)} = W_i^{(c)} = 1 / \left\{ 2\left(n_x + \lambda \right) \right\}, i = 1, \ldots, 2n_x \tag{20}$$

where the constant α determines the spread of the sigma points around $\bar{\mathbf{x}}$ and is usually set to a small positive value. The constant κ is a secondary scaling parameter which is usually set to 0 or $3 - n_x$ (Julier et al., 1995). β is a non-negative weighting term which can be used to incorporate knowledge of the higher order moments of the distribution of $\mathbf{x}$, for a Gaussian distribution, and $\beta = 2$ is optimal (van der Merwe et al., 2000). W_i is the weight associated with the *ith* sigma point. Here, these parameters were set to $\alpha = 1$, $\beta = 2$ and $\kappa = 0$.

For $t \in \{1,\ldots,\infty\}$, the tracking process combined with UKF and a hierarchical model is detailed as follows:

1. Calculate sigma points

$$\mathcal{X}_{t-1} = \left[\bar{\mathbf{x}}_{t-1} \quad \bar{\mathbf{x}}_{t-1} \pm \sqrt{(n_x + \lambda) \mathbf{P}_{t-1}} \right] \tag{21}$$

where $\sqrt{(n_x + \lambda) \mathbf{P}_{t-1}}$ is the *ith* row or column of the matrix square root of $(n_x + \lambda) \mathbf{P}_{t-1}$ which can be implemented directly by using a Cholesky factorization.

2. Time update:

$$\mathcal{X}_{t|t-1} = \mathrm{f}\left(\mathcal{X}_{t-1}, H_{t-1,t} \right) \tag{22}$$

$$\bar{\mathbf{x}}_{t|t-1} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathcal{X}_{i,t|t-1} \tag{23}$$

$$\tilde{\mathbf{x}}_{i,t|t-1} = \mathcal{X}_{i,t|t-1} - \bar{\mathbf{x}}_{t|t-1} \tag{24}$$

$$\mathbf{P}_{t|t-1} = \sum_{i=0}^{2n_x} W_i^{(c)} \left(\tilde{\mathbf{x}}_{i,t|t-1} \right)\left(\tilde{\mathbf{x}}_{i,t|t-1} \right)^{\mathrm{T}} + \mathbf{Q} \tag{25}$$

$$\mathcal{Y}_{t|t-1} = \mathrm{h}\left(\mathcal{X}_{t|t-1} \right) \tag{26}$$

$$\bar{\mathbf{y}}_{t|t-1} = \sum_{i=0}^{2n_x} W_i^{(m)} \mathcal{Y}_{i,t|t-1} \tag{27}$$

3. Observe in the search regions determined by sigma points and obtain $\mathbf{y}_t$ as a solution of the maximum a posteriori (MAP) based on its contextual knowledge $\mathbf{CK}_t$:

$$\mathbf{y}_t = \operatorname*{argmax}_i P\left(\mathbf{y}_{i,t} \middle| \mathbf{CK}_t \right) \tag{28}$$

$$P\left(\mathbf{y}_{i,t} \middle| \mathbf{CK}_t \right) = P\left(M_{i,t}^b \middle| M_{i,t}^h, \mathbf{CK}_t \right) P\left(M_{i,t}^h \middle| \mathbf{CK}_t \right) \tag{29}$$

a. Calculate the observation probability of the helmet model in the search region $P\left(M_{i,t}^h \middle| \mathbf{CK}_t \right)$ which is the score of the template matching method, and map the skater's position to the rink in the world $Pos_{i,rink}$ by multiplying $H_{t,rink}$.

b. Compute which sub-region the skater skates through according to $Pos_{i,rink}$, and obtain the index $iRegion \in \{1,\ldots,12\}$, detailed in Fig. 5.

c. Obtain $PosRel_{i,t}^{hb}$ according to $iRegion$, and the body model $M_{i,t}^{b}$ is determined by $\{M_{i,t}^{h}, PosRel_{i,t}^{hb}\}$.

d. Calculate the Bhattacharyya coefficient ρ by Eq. (5) between the target histogram $Hist \in M_{t-1}^{b}$ and the observation one $Hist \in M_{i,t}^{b}$, and obtain the probability $P\left(M_{i,t}^{b} \middle| M_{i,t}^{h}, \mathbf{CK}_{t}\right) = \rho$.

4. Measurement update:

$$\tilde{\mathbf{y}}_{i,t|t-1} = \mathcal{Y}_{i,t|t-1} - \overline{\mathbf{y}}_{t|t-1} \tag{30}$$

$$\mathbf{P}_{\tilde{\mathbf{y}}_t \tilde{\mathbf{y}}_t} = \sum_{i=0}^{2n_x} W_i^{(c)} \left(\tilde{\mathbf{y}}_{i,t|t-1}\right)\left(\tilde{\mathbf{y}}_{i,t|t-1}\right)^{\mathrm{T}} + \mathbf{R} \tag{31}$$

$$\mathbf{P}_{\tilde{\mathbf{x}}_t \tilde{\mathbf{y}}_t} = \sum_{i=0}^{2n_x} W_i^{(c)} \left(\tilde{\mathbf{x}}_{i,t|t-1}\right)\left(\tilde{\mathbf{y}}_{i,t|t-1}\right)^{\mathrm{T}} \tag{32}$$

$$\mathbf{K}_t = \mathbf{P}_{\tilde{\mathbf{x}}_t \tilde{\mathbf{y}}_t} \mathbf{P}_{\tilde{\mathbf{y}}_t \tilde{\mathbf{y}}_t}^{-1} \tag{33}$$

$$\overline{\mathbf{x}}_t = \overline{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left(\mathbf{y}_t - \overline{\mathbf{y}}_{t|t-1}\right) \tag{34}$$

$$\mathbf{P}_t = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{P}_{\tilde{\mathbf{y}}_t \tilde{\mathbf{y}}_t} \mathbf{K}_t^{\mathrm{T}} \tag{35}$$

6. Experiments and results

6.1 The output of the proposed system

The trajectory of a skater in an individual race of 500 meters is illustrated in Fig. 7, different color line denotes the trajectory of the skater when he/she skates along different loop, and the order is red, green, blue, black and yellow respectively. In Fig. 8, the velocity of one skater in a individual race of 500 meters is shown, the green line represents the velocity obtained manually as the ground truth, and the black one denotes the result of our tracking algorithm. More information and statistic of the competitions are available such as the trajectory and velocity, which can be further processed and analyzed by the sports experts.

6.2 Tracking results

First, our tracking method compares with the MeanSHIFT and CAMSHIFT algorithm from Open Computer Vision Library (OpenCV, http://sourceforge.net/projects/opencvlibrary/). The results are illustrated in Fig. 9, green box represents the hierarchical model in the corresponding sub-region. If the target color does not change, the MeanSHIFT and CAMSHIFT trackers are quite robust, but they are easily distracted when similar colors appear in the background, however, our tracker can work well by means of a hierarchical model.
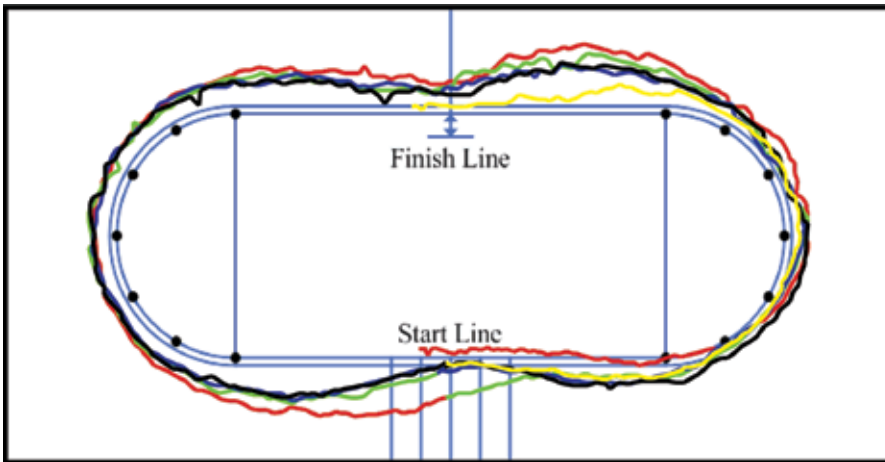
Fig. 7. The trajectory of a skater in an individual race of 500 meters, different color line denotes the trajectory of the skater when he/she skates along different loop.
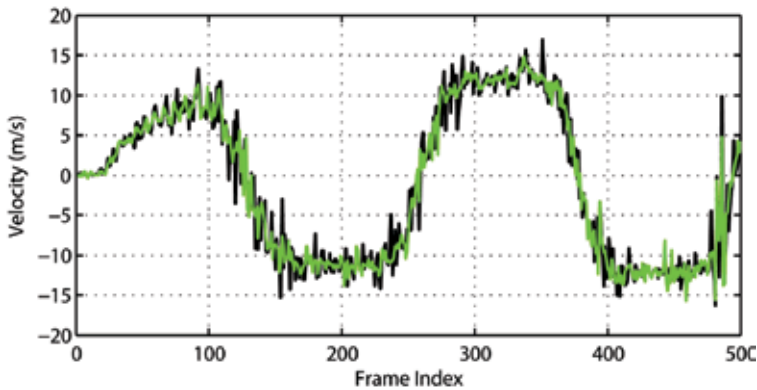

Fig. 8. The velocity of one skater in the individual race of 500 meters.

Second, the comparison results of our tracker and the adaptive color-based particle filter (Nummiaro et al., 2003) are shown in Fig. 10. The failure of the adaptive color-based particle filter tracking is caused by: (1). The model is updated every frame, the patches of the background could be integrated into the model, for a certain time period, the tracker will shift with the background and will not work anymore. (2). Using the single part model (one block or ellipse) and the single cue (color distribution), however, it is too weak for the tracking in the sports domain.

Finally, in Fig. 11, the competitor skating through the curve can be exactly tracked despite successive occlusion. By using the hierarchical model, the proposed tracker would firstly detect the helmet model of the tracked skater by the template matching method. Since the helmet is not occluded, the correct detection result of the helmet model should be observed in the specific search region, then, the color histogram matching method is used to detect the occluded body, compared with other observation results, the observation probability as in Eq. (29) in the specific search region must be the largest, that is, the skater is tracked successfully.

(a) The result of the MeanSHIFT tracker

(b) The result of the CAMSHIFT tracker

(c) The result of our tracker

Fig. 9. MeanSHIFT and CAMSHIFT versus our method.

(a) The result of the adaptive color-based particle filter tracker

(b) The result of our tracker

Fig. 10. The adaptive color-based particle filter versus our method.

| (a) frame 354 | (b) frame 358 | (c) frame 362 |

| (d) frame 366 | (e) frame 370 | (f) frame 374 |
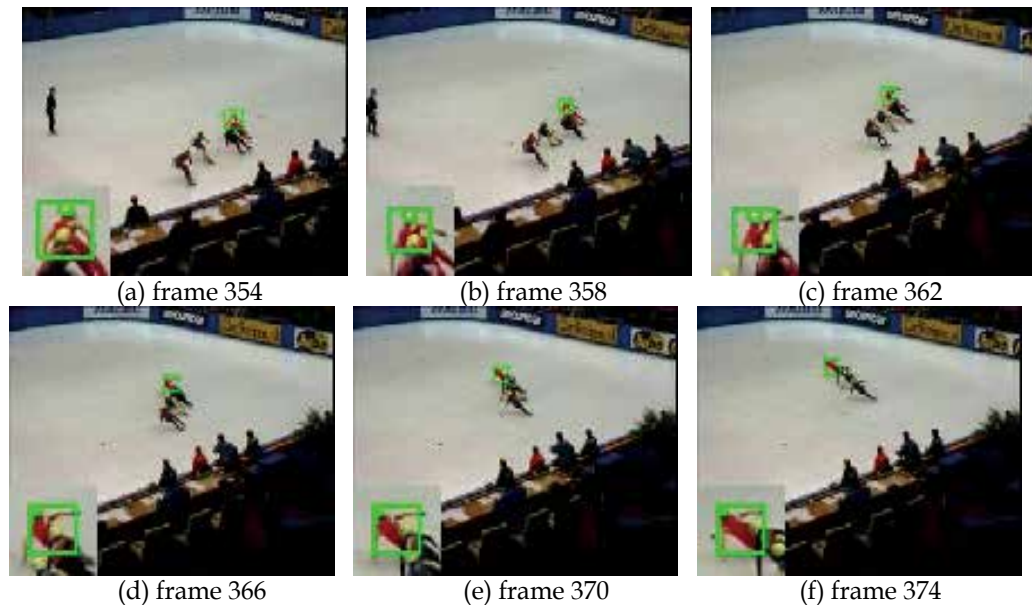
Fig. 11. Tracking results of successive occlusion.

The better performance of the proposed tracker is due to the following three aspects:
1. Using the contextual knowledge, namely, the hierarchical model in each sub-region, it can help the tracker to know when and how to update the hierarchical model.
2. Multiple cues: the template matching method (for helmet) and the color histogram matching (for body), it can make the tracker more robust when a skater moving through the advertisement board or occlusions appear on the curve.
3. The unscented Kalman filer can capture the posterior mean and covariance accurately to the 2*nd* order. In addition, compared with the particle filter, the UKF is more efficient since the number of the sigma points increases linearly with the number of dimension.

6.3 Evaluate the homographies

In our practical system as shown in Fig. 2, there are two kinds of homographies: (1) the homography between two neighborhood frames can remove the camera motion to improve the precision of the motion prediction in the process of tracking. (2) Another homography transforming the current frame to the model of the real rink is associated with not only the output of tracking subsystem but also the tracking process to obtain a skater's hierarchical model when the skater moves through different sub-regions of the rink. The calculation of them is a random process and their accuracy is uncertain, therefore, we have to try to evaluate them and analyze how they affect the tracking performance and the system's output.

Note that our camera pans always at the horizontal direction, in which the accuracy of the homographies is very different from that in the vertical direction. Therefore, for all experiments in this section, statistics of variables are to be derived from the two directions respectively. For simplicity's sake, we only give those equations in the horizontal direction which are the same as that in the vertical direction. $\left| \cdot \right|_x$ denotes the value of the horizontal direction and $\left| \cdot \right|_y$ the value of the vertical direction.

6.3.1 The relation between the accuracy of homography $H_{t-1,t}$ and the tracking performance

	lap 1	lap 2	lap 3	lap 4	lap 5	nframes
BS	1-87	291-348	523-581	756-816	993-1053	326
RC	88-159	349-405	582-636	817-872	1054-1108	295
TS	160-230	406-467	637-700	873-935	1109-1149	301
LC	231-290	468-522	701-755	936-992		227

Table 1. A test video sequence is divided into different groups.

Generally, the tracking process can be considered as two components: *prediction* and *detection* (or observation). Their accuracy can affect the tracking performance directly, however, which one is the main factor leading to the tracking failing: *prediction* or *detection*? It is very important for a practical tracking system. We try to analyze it and test on a video sequence, which are segmented into four groups: bottom straight(BS), top straight(TS), left curve(LC) and right curve(RC), according to the spatio-temporal relation shown in Table 1.
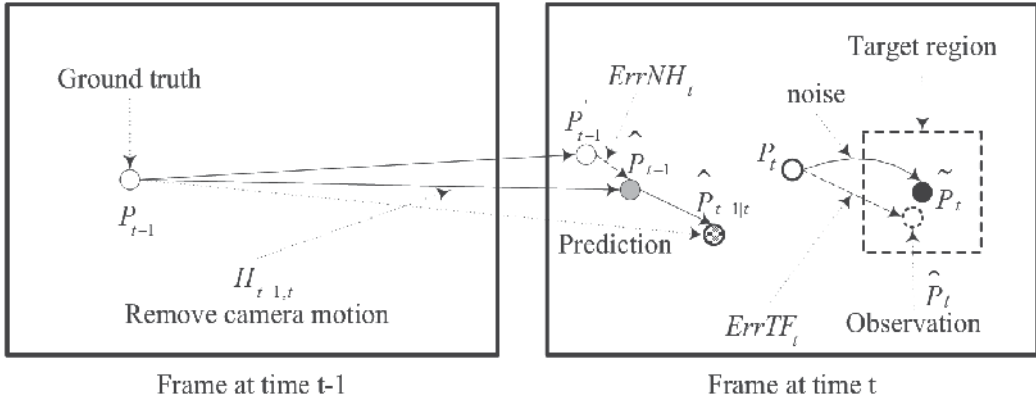
Fig. 12. The diagram of the tracking process between the two neighborhood frames. The ground truth P_t added on Gaussian white noise, i.e. $\tilde{P}_t = P_t + noise$, can be considered as the simulation of the different prediction result $\hat{P}_{t-1|t}$, which assumption is helpful to analyze the tracking performance.

To explain and analyze the tracking process better, some points in the two neighborhood frames are defined, as illustrated in Fig. 12. P_{t-1} is the position of the tracked skater in the frame at time $t-1$ and P_t at time t, which are obtained manually as the ground truth. P_{t-1}' is the corresponding point of P_{t-1} under the ideal transformation if and only if $H_{t-1,t}$ is very accurate, $\hat{P}_{t-1}$ as the approximation of P_{t-1}' is calculated by $\hat{P}_{t-1} = H_{t-1,t}P_{t-1}$. $ErrNH_t$ (the Error of the Neighborhood Homography at time t) is denoted as

$$ErrNH_t = \hat{P}_{t-1} - P_{t-1}' \qquad (36)$$

$\hat{P}_{t-1}$ is transformed to $\hat{P}_{t-1|t}$ by using the dynamic model, $\tilde{P}_t = P_t + noise$ can be considered as the simulation of the different prediction result $\hat{P}_{t-1|t}$, which assumption is helpful to

analyze the tracking performance. $ErrTF_t$ (the Error of the Tracking result in the Frame at time t) is expressed by

$$ErrTF_t = \hat{P}_t - P_t \qquad (37)$$

The tracking result would be considered as the accuracy in the ith frame if the distance of $ErrTF_i$ is less than the threshold value τ ($\tau = 5$ pixels) defined as

$$TA_i = \begin{cases} 1 & \text{if } \left\|ErrTF_i\right\|_x \leq \tau \text{ and } \left\|ErrTF_i\right\|_y \leq \tau \\ 0 & \text{otherwise} \end{cases} \qquad (38)$$

where $\left\|\cdot\right\|_x$ denotes the absolute value of the horizontal distance and $\left\|\cdot\right\|_y$ the absolute value of the vertical distance.

As for the image sequence, the precision of the tracking result can be expressed by the following equation, where $nframes$ is the total number of frames used in statistics.

$$TP = \frac{\sum_{i=1}^{nframes} TA_i}{nframes} \qquad (39)$$

	Noise range					
	0	1-5	1-10	5-10	1-15	10-15
BS	100	100	97.55	95.4	88.04	65.34
RC	100	98.99	95.61	90.88	91.99	67.91
TS	100	99.34	98.01	95.35	93.36	83.72
LC	100	100	93.39	86.34	85.46	66.52
All	100	99.57	96.35	92.43	89.9	71.02

Table 2. The precision percentage of the tracking results given six groups of Gaussian white noise.

The experiments have been implemented with six groups of Gaussian white noise. For a better simulation and qualitative analysis, the noise within some range is selected and others are discarded. The noise range is shown in Table 2, $noise = 0$ means that no noise would be added on the ground truth P_t, i.e. the skater's hierarchical model is detected directly on P_t. In Table 2, when $noise = 0$, the precision of the whole match is 100%, that is to say, if the prediction accuracy is enough, our detection method is very robust. However, with the increase of the $noise$ value, i.e, the prediction accuracy becomes much lower, the value of TP decreases gradually. When $noise < 10$, TP in RC and LC is lower than the one in BS and TS due to the skater's occlusion often occurred at RC and LC which can cause our tracker drift or disabled. When $noise > 10$, TP in BS is much lower. In BS, a skater is closer to the camera, its scale changes quickly, its size and the part of the occlusion become bigger, all of which can make the tracker drift out of the threshold value τ easily. At the same time, with the increase of the $noise$ value, the horizontal error of the tracking result in the frame

($|ErrTF_i|_x$) becomes larger in Fig. 13, and the RMS (Root Mean Square) error of $ErrTR$ (the Error of the Tracking result in the real Rink) defined as Eq. (40) and Eq. (41) is also increasing in Fig. 14.

$$|ErrTR_i|_x = |H_{i,rink}\hat{P}_t - H_{i,rink}P_t|_x = |H_{i,rink}ErrTF_i|_x \qquad (40)$$

$$|RMSErrTR|_x = \sqrt{\frac{\sum_{i=1}^{nframes}(|ErrTR_i|_x)^2}{nframes}} \qquad (41)$$

All the above demonstrate that the higher the prediction precision is, the better the tracking performance is. However, the prediction result in practice is $\hat{P}_{t-1|t}$, instead of $\tilde{P}_t$, *how accurate the prediction* $\hat{P}_{t-1|t}$ *is?* In Fig. 12, the prediction error comes from two sources: (1) one is the assumption of the uniform motion in the tracking process (i.e. $\hat{P}_{t-1} \to \hat{P}_{t-1|t}$), but it can be reasonable and acceptable in the practical systems and (2) the other is the homography $H_{t-1,t}$ used to remove the camera motion (i.e. $P_{t-1} \to \hat{P}_{t-1}$). Therefore, the prediction accuracy depends on the error of the homography $H_{t-1,t}$ (i.e. $ErrNH_t$). To evaluate the homography $H_{t-1,t}$, we have selected 14 still marker blocks on the rink and recorded the positions of those visible markers in each frame as the ground truth, it means that P_{t-1} and P'_{t-1} are prior to be known. The RMS error of $H_{t-1,t}$ is given by

$$|NH|_x = \sqrt{\frac{\sum_{i=1}^{nframes}(|ErrNH_i|_x)^2}{nframes}} \qquad (42)$$
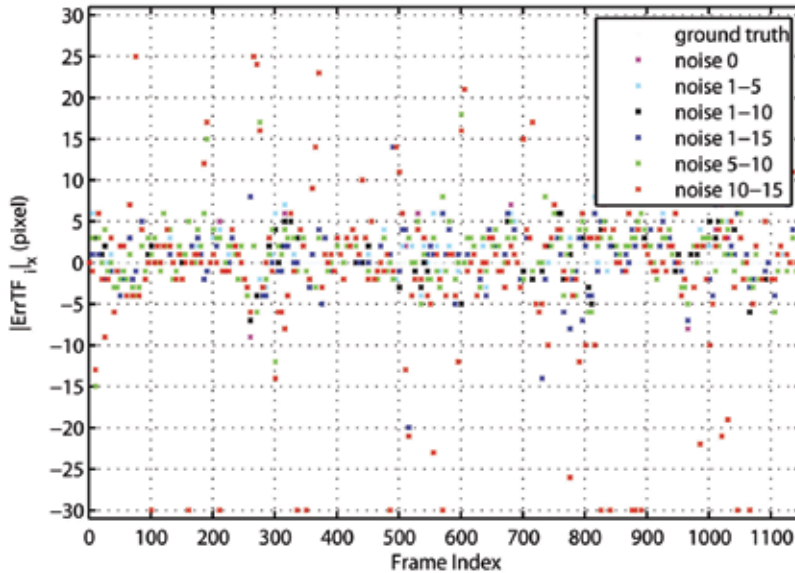
Fig. 13. The horizontal error of the tracking results in the frame given six groups of Gaussian white noise.
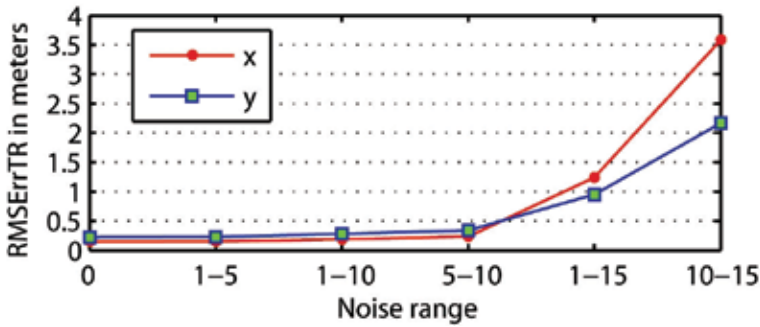
Fig. 14. The RMS error of the tracking result in the real rink given six groups of Gaussian white noise.

	LC							RC						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
$|NH|x$	0.86	1.64	1.28	1.12	1.11	0.99	0.82	1.96	1.92	1.59	1.44	1.15	0.97	1.2
$|NH|y$	0.85	1.21	1.09	1.14	1.13	1.07	0.89	1.29	1.27	1.03	0.95	0.8	0.76	0.81
nframes	181	267	238	187	128	154	231	353	370	381	310	233	248	322

Table 3. The RMS error of $H_{t-1,t}$ of 14 markers on the left curve(LC) and right curve(RC) respectively in pixels.

In Table 3, the accuracy of $H_{t-1,t}$ is enough for the practical application. After all, the calculation of the homography is a random process, but we can increase the number of *good* features (i.e. inliers) to improve the computational stability and reliability of the homograpy (Liu et al., 2007). In a word, the better tracking performance depends on both the robust detection method and the accurate homography $H_{t-1,t}$.

6.3.2 Evaluate the accuracy of homography $H_{i,rink}$ and the precision of our system

The output of our system is the skater's position in the real rink, how to measure the output precision of the practical system needs to be solved. The easiest way is calculating the error between the tracked skater's position and the ground truth (3D position) in the real world directly. However, by this means, we can only obtain the approximation of the former (i.e. $H_{i,rink}P_t$) and can not get the value of the latter without the assistance of the sensors in practice. Therefore, for a more objective evaluation, an indirect approach is used to measure the system's precision (i.e. the accuracy of homography $H_{i,rink}$). In a similar way detailed in section 6.3.1, there are 14 still marker blocks on the rink, their 3D spatial positions are prior to be known according to ISU (http://www.isu.org/) and denoted as RM^j ($j \in \{1,\dots,14\}$). The imagery positions of all visible markers in each frame can be recorded manually and transformed to the real rink by the following equation:

$$\widehat{RM}_i^j = H_{i,rink}FM_i^j \tag{43}$$

where FM_i^j is the position of the jth marker in the ith frame, and $\widehat{RM}_i^j$ the estimated position of FM_i^j in the real rink. If $H_{i,rink}$ is always quite accurate, for $\forall i$, $\widehat{RM}_i^j \equiv RM^j$. However, in the practical system, the accuracy of $H_{i,rink}$ is affected by a variety of factors. As the measurement of the registration error caused by $H_{i,rink}$, the standard deviation of $\widehat{RM}_i^j$ and the RMS error are expressed by the following equations:

$$|std^j|_x = \sqrt{\frac{\sum_{i=1}^{nframes}\left(|\widehat{RM}_i^j|_x - \mathrm{E}\left[|\widehat{RM}_i^j|_x\right]\right)^2}{nframes}} \tag{44}$$

where $\mathrm{E}[\cdot]$ is the expectation function.

$$|rms^j|_x = \sqrt{\frac{\sum_{i=1}^{nframes}\left(|\widehat{RM}_i^j|_x - |RM^j|_x\right)^2}{nframes}} = \sqrt{\frac{\sum_{i=1}^{nframes}\left(|H_{pano,rink}H_{i,pano}FM_i^j|_x - |RM^j|_x\right)^2}{nframes}} \tag{45}$$

where $H_{pano,rink}$ maps the panorama to the real rink, however, $H_{pano,rink}$ is inaccurate in practice, we define the error of $H_{pano,rink}$ as

$$|PRH^j|_x = |H_{pano,rink}PM^j|_x - |RM^j|_x \tag{46}$$

where PM^j is the position of the jth marker based on the panorama, it is recorded manually as the ground truth (see details in Algorithm 1 step 5).

The registration error of 14 markers is shown in Table 4. The values of rms distribute differently, ones are highly accurate and others have large symmetrical errors, all that are derived from the inaccuracy of $H_{pano,rink}$ (i.e. PRH, detailed in last two rows) directly and mostly, since $H_{pano,rink}$ dose not depend on the subscript i in Eq. (45). The value of std (mainly caused by $H_{i,pano}$) is uniformly small, and it means that $H_{i,pano}$ is more accurate than $H_{pano,rink}$. As a result, the accurate $H_{i,rink}$ is due to the better $H_{i,pano}$ and $H_{pano,rink}$. These markers in the rink are closer to the trajectory of a skater, so it is reasonable that the registration error can be considered as an approximation of the output precision of our system. The smaller the values of rms and std are, the more reliable the output of our proposed system is.

	LC							RC						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
|rms|x	0.46	0.31	0.07	0.19	0.16	0.06	0.26	0.43	0.31	0.05	0.26	0.25	0.04	0.29
|rms|y	0.45	0.28	0.14	0.11	0.21	0.21	0.07	0.39	0.17	0.07	0.05	0.11	0.19	0.16
|std|x	0.04	0.04	0.06	0.05	0.04	0.04	0.03	0.05	0.06	0.04	0.04	0.04	0.03	0.02
|std|y	0.06	0.08	0.07	0.06	0.04	0.04	0.03	0.09	0.09	0.07	0.06	0.05	0.03	0.04
nframes	186	280	248	201	139	168	246	365	382	397	326	244	263	339
|PRH|x	0.4	0.21	-0.12	-0.3	-0.18	0.04	0.23	-0.33	-0.27	0.08	0.28	0.31	0.04	0.27
|PRH|y	-0.19	-0.01	0.14	0.13	-0.13	-0.15	0.03	-0.16	0.01	0.15	0.1	-0.02	-0.16	0.08

Table 4. The registration error of 14 markers on the left curve(LC) and right curve(RC) respectively in meters.

7. Conclusion and discussion

In this paper, we propose a novel computer vision system for tracking high-speed non-rigid skaters over a large playing area in short track speed skating competitions. Several important features distinguish the proposed approach from others:

1. Introducing the reference frames as a transition through which each frame can be mapped to the field model to reduce the error accumulation of the projection, and it's very important for a long video sequence and helpful for improving the precision of the system.
2. Incorporating the hierarchical model based on the contextual knowledge and multiple cues into the unscented Kalman filter to improve the tracking performance when occlusions occur.
3. Choosing the unscented Kalman filter for visual tracking in the sports domain, it is superior to EKF in theory, and is more efficient than particle filter.
4. Evaluating the relation between the accuracy of homography $H_{t-1,t}$ and tracking performance.
5. Proposing a novel and objective evaluation method to measure the precision of our practical system.

However, The main problem is remained in our current system: how to improve the tracking performance when skaters are moving in groups during a long and continuously full occlusion.

In future, we can model the skater's uniform of different teams in each sub-region, and the uniform model can be used to assist in tracking the target occluded for a long time.

8. Acknowledgements

9. References

Aherne, F. J.; Thacker, N. A. & Rockett, P. I. (1997). The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data, Kybernetika, vol. 34, No. 4, pp. 363-368, ISSN: 0023-5954

Brown, M. & Lowe, David G. (2003). Recognising panoramas, Ninth IEEE International Conference on Computer Vision, pp. 1218-1225, ISBN: 0-7695-1950-4, Nice, France, October 2003

Chen, Yunqiang; Rui, Yong & Huang, Thomas S. (2006). Multicue HMM-UKF for Real-Time Contour Tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, No. 9, pp. 1525-1529, ISSN: 0162-8828

Comaniciu, D.; Ramesh, V. & Meer, P. (2003). Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, No. 5, pp. 564-577, ISSN: 0162-8828

Fischler, M. A. & Bolles, Robert C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM, vol. 24, No. 6, pp. 381-395, ISSN: 0001-0782

Farin, Dirk; Krabbe, Susanne; de With, Peter H.N. & Effelsberg, Wolfgang (2004). Robust Camera Calibration for Sport Videos Using Court Models, SPIE Storage and

Retrieval Methods and Applications for Multimedia, pp. 80-91, ISBN: 0-8194-5210-6, San Jose CA, January 2004

Haritaoglu, I.; Harwood, D. & Davis, L. S. (2000). W4: Real-Time Surveillance of People and Their Activities, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, No. 8, pp. 809-830, ISSN: 0162-8828

Harris, Chris & Stephens, Mike (1988). A combined corner and edge detector, Fourth Alvey Vision Conference, pp. 147-151, University of Manchester, September, 1988

Hartley, R. & Zisserman, A. (2000). Multiple View Geometry in Computer Vision, Cambridge University Press, ISBN: 0-5216-2304-9

Hayet, Jean-Bernard; Piater, Justus & Verly, Jacques (2004). Robust incremental rectification of sport video sequences, British Machine Vision Conference, Kingston University, London, September, 2004

Hayman, Eric & Murray, David W. (2003). The Effects of Translational Misalignment when Self-Calibrating Rotating and Zooming Cameras, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, No. 8, pp. 1015-1020, ISSN: 0162-8828

Intille, S. S. & Bobick, A. F. (1994). Tracking Using a Local Closed-World Assumption: Tracking in the Football Domain, MIT Media Lab Perceptual Computing Group Technical Report 296, 1994

Jug, M.; Pers, J.; Dezman, B. & Kovacic, S. (2003). Trajectory Based Assessment of Coordinated Human Activity, Third International Conference ICVS 2003, pp. 534-543, ISBN: 3-540-00921-3, Graz, Austria, April 2003

Julier, Simon J. & Uhlmann, Jeffrey K. (1997). A New Extention of the Kalman Filter to Nonlinear Systems, Proceedings of the 11th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, Orlando, Florida, April 1997

Julier, Simon J.; Uhlmann, Jeffrey K. & Durrant-Whyte, Hugh F. (1995). A new approach for filtering nonlinear systems, Proceedings of the American Control Conference, pp. 1628-1632, ISBN: 0-7803-2446-3, Seattle, Washington, June 1995

Junior, B. M. & Anido, R. D. O. (2004). Distributed real-time soccer tracking, Proceedings of the ACM Second International Workshop on Video Surveillance & Sensor Networks, pp. 97-103, ISBN: 1-58113-934-9, New York, October 2004

Liu, GuoJun; Tang, XiangLong; Sun, Da & Huang, JianHua (2007). Robust Registration of Long Sport Video Sequence, Proceedings of the 5th International Conference on Computer Vision Systems, ISBN: 3-0002-0933-8, Bielefeld University, March 2007

Lowe, David G. (2004). Distinctive image features from scale-invariant keypoints International Journal of Computer Vision, vol. 60, No. 2, pp. 91-110, ISSN: 1573-1405

Misu, Toshihiko; Naemura, Masahide; Zheng, Wentao; Izumi, Yoshinori & Fukui, Kazuo (2002). Robust tracking of soccer players based on data fusion, 16th International Conference on Pattern Recognition, pp. 556-561, ISBN: 0-7695-1695-X, Quebec, Canada, August 2002

Needham, C. J. & Boyel, R. D. (2001). Tracking multiple sports players through occlusion, congestion and scale, Proceedings of the British Machine Vision Conference, pp. 93-102, ISBN: 1-901725-16-2, Manchester, UK, September 2001

Nummiaro, K.; Koller-Meier, E. & Gool, L. V. (2003). An adaptive color-based particle filter Image and Vision Computing, vol. 21, No., pp. 99-110, ISSN: 0262-8856

Okuma, K. (2003). Automatic acquisition of motion trajectories, Master's thesis, University of British Columbia, 2003

Okuma, Kenji; Little, James J. & Lowe, David G. (2004). Automatic rectification of long image sequences, Asian Conference on Computer Vision, 2004

Ozyildiz, E.; Krahnstover, N. & Sharma, R. (2002). Adaptive texture and color segmentation for tracking moving objects Pattern Recognition, vol. 35, No., pp. 2013-2029, ISSN: 0031-3203

Pers, J.; Bon, M.; Kovacic, S.; M. Sibila & Dezman, B. (2002). Observation and Analysis of Large-scale Human Motion Human Movement Science, vol. 21, No. 2, pp. 295-311, ISSN: 0167-9457

Pers, J.; Vuckovic, G.; Kovacic, S. & Dezman, B. (2001). A low-cost real-time tracker of live sport events, International symposium on image and signal processing and analysis, pp. 362-365, June 2001

Pingali, G. S.; Jean, Y. & Carlborn, I. (1998). Real time tracking for enhanced tennis broadcasts, International Conference on Computer Vision and Pattern Recognition, pp. 260-265, ISBN: 0-8186-8497-6, Santa Barbara, CA, June 1998

Rasmussen, Christopher & Hager, Gregory D. (2001). Probabilistic data association methods for tracking complex visual objects, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, No. 6, pp. 560-576, ISSN: 0162-8828

Rigoll, G.; Breit, H. & Wallhoff, F. (2004). Robust tracking of persons in real-world scenarios using a statistical computer vision approach, Image and Vision Computing, vol. 22, No., pp. 571-582, ISSN: 0262-8856

Shi, Jianbo & Tomasi, Carlo (1994). Good Features to Track, International Conference on Computer Vision and Pattern Recognition, Seattle, Washington, June 1994

Shum, Heung-Yeung & Szeliski, Richard (2000). Systems and Experiment Paper: Construction of Panoramic Image Mosaics with Global and Local Alignment, International Journal of Computer Vision, vol. 36, No. 2, pp. 101-130, ISSN: 1573-1405

Sivic, Josef & Zisserman, Andrew (2003). Video Google: A Text Retrieval Approach to Object Matching in Videos, Ninth IEEE International Conference on Computer Vision, pp. 1470-1477, ISBN: 0-7695-1950-4, Nice, France, October 2003

Tomasi, Carlo & Kanade, Takeo (1991). Detection and Tracking of Point Features, Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991

van der Merwe, Rudolph; Doucet, Arnaud; de Freitas, Nando & Wan, Eric (2000). The Unscented Particle Filter, Technical Report CUED/FINFENG/TR380, Cambridge University Engineering Department, August 2000

Vandenbroucke, N.; Macaire, L. & Postaire, J. G. (2003). Color image segmentation by pixel classification in an adapted hybrid color space: Applications to soccer image analysis, Computer Vision and Image Understanding, vol. 90, No., pp. 190-216, ISSN: 1077-3142

Yan, Fei; Kostin, Alexey; Christmas, William J. & Kittler, Josef (2006). A Novel Data Association Algorithm for Object Tracking in Clutter with Application to Tennis Video Analysis International Conference on Computer Vision and Pattern Recognition, pp. 634-641, ISBN: 0-7695-2597-0, New York, June 2006

Yang, Changjiang; Duraiswami, R. & Davis, L. (2005). Efficient mean-shift tracking via a new similarity measure International Conference on Computer Vision and Pattern Recognition, pp. 176-183, ISBN: 0-7695-2372-2, San Diego, CA, June 2005

Yilmaz, A.; li, Xin & Shah, M. (2004). Contour-based object tracking with occlusion handling in video acquired using mobile cameras, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, No. 11, pp. 1531-1536, ISSN: 0162-8828

Iterative Forward-Backward Kalman Filtering for Data Recovery in (Multiuser) OFDM Communications[1]

Tareq Y. Al-Naffouri, Muhammad S. Sohail and Ahmed A. Quadeer
King Fahd University of Petroleum & Minerals
Saudi Arabia

1. Introduction

The current era of connectivity and information transfer demands high data rates from broadband wireless systems. The main problem faced by such high data rate systems is multipath fading as it causes inter symbol interference (ISI). Thus, there is a need for a technique that avoids ISI while still providing high speed data. Orthogonal Frequency Division Multiplexing (OFDM) emerged as a technique that combines the advantages of high achievable data rates, relatively easy implementation, high spectral efficiency and robustness to multipath fading. This is reflected by many standards that considered and adopted OFDM including digital audio and video broadcasting (DAB and DVB), WIMAX (Worldwide Inter-operability for Microwave Access), high speed modems over digital subscriber lines, and local area wireless broadband standards such as the HIPERLAN/2 and IEEE 802.11a (Stuber et al., 2004).

In a wireless communication system, data is sent over a channel. This channel is time variant and undergoes fading. The exact state of the channel is unknown at the receiver. The transmitted signal is received at the receiver convolved with the channel and corrupted with noise. The main interest at the receiver is to recover the transmitted data. If we consider that the channel state information is known at the receiver, we can faithfully extract the transmitted data from the received signal with this knowledge (through equalization). In practice however, we do not have prior knowledge of the channel state information and hence we have to employ some estimation technique to obtain an estimate of the channel. Channel estimation is thus an important step in receiver design.

We will start this chapter by reviewing the various approaches to channel estimation available in literature. We will then present an iterative channel estimation technique based on the Forward-Backward Kalman filter for simple OFDM systems and later extend the concept for multi-access OFDM and Multi Input Multi Output (MIMO) OFDM systems.

---

2. Various approaches to channel estimation

An OFDM receiver needs to be able to accurately estimate the transmitted data and for that, it needs accurate channel state information. For time variant channel, an additional problem is that channel state keeps changing. One way to deal with such channels is to periodically send a training (known) sequence which can be used at the receiver to estimate the channel. However this would severely strain the bandwidth efficiency of the system. An alternative approach is to use a priori constraints on the communication system to reduce the training overhead. The most popular of these constraints are summarized in Table 1.1 together with the works that employed them. All channel estimation techniques use all or a subset of these constraints. For example, to reduce the training overhead, we could assume that the channel is sparse or that it exhibits strong time correlation from one instant into another. Similarly, we could use some a priori information about the transmitted data such as the fact that it is drawn from some finite alphabet (see Table 1.1).

Type	Constraints	Reference
Data Constraints	Finite alphabet constraint	(Al-Naffouri et al., 2001); (Shengli and Giannakis, 2001); (Al-Naffouri, 2007) (Yang and Ser, 2004)
	Code	(Al-Rawi et al., 2003); (Zhang and Chen, 2004); (Petropulu et al., 2004); (Gao and Nallanathan, 2007)
	Transmit precoding (e.g., cyclic prefix, zero-padding, virtual carriers)	(Al-Naffouri, 2007); (Bölcskei et al., 2002); (Al-Rawi et al., 2003); (Kim and Eo, 2006); (Muquet et al., 2000); (Shin et al., 2007); (Kunji et al., 2006)
	Pilots	(Edfors et al., 1998) (Negi and Cioffi, 1998); (Biguesh and Gershman, 2004); (Li et al., 1998); (Minn and Al-Dhahir, 2006)
Channel Constraints	Finite delay spread	(Bölcskei et al., 2002); (Negi and Cioffi, 1998)
	Sparsity	(Yang et al., 2001); (Kang et al., 1999)
	Frequency correlation	(Al-Naffouri, 2007); (Edfors et al., 1998); (Al-Rawi et al., 2003); (Chang and Su, 2004); (Cui and Tellambura, 2006)
	Time correlation	(Muquet et al., 2000); (Al-Naffouri et al., 2004); (Necker and Stuber, 2004); (Zhang and Chen, 2004); (Al-Naffouri, 2007)
	Uncertainty information	(Sayed, 2001); (Li et al., 1998)

Table 1.1. Constraints used for channel estimation

2.1 Iterative receivers for channel estimation & data recovery

There are several methods for channel estimation. However, the state of the art receiver is iterative in nature in that it iterates between finding a channel estimate which is used for data detection and between finding a data estimate which is in turn used to enhance channel estimation. Training data kick starts the iterative process by providing an initial channel estimate. Moreover, the a priori information that we have about the channel and data enhance the channel estimation and data detection steps which in turn reduces the required transmission overhead (Al-Naffouri et al., 2002); (Aldana et al., 2003); (Cozzo & Hughes, 2003). In this chapter, we use the Expectation-Maximization (EM) algorithm to design an iterative receiver for the estimation and equalization of time variant channels. We will show that the receiver boils down to a forward-backward Kalman filter. We will discuss the use of Kalman filter for channel and data recovery in single user as well as multiuser OFDM systems. To setup the stage, we introduce our notation in the following section followed by the system model.

3. Notation

In this chapter, scalars are denoted by small-case letters (e.g., y), vectors by small-case boldface letters (e.g., $\boldsymbol{y}$), and matrices by uppercase boldface letters (e.g., $\boldsymbol{X}$). For vectors in frequency domain, calligraphic notation (e.g. $\mathcal{Y}$) is used. An estimate of a variable is indicated by a hat on that variable (e.g., $\hat{\boldsymbol{h}}$ is an estimate of $\boldsymbol{h}$). Also, conjugate transpose is represented by $*$, Kronecker product by $\otimes$, size $N \times N$ identity matrix by $\boldsymbol{I}_N$, and the all zero $M \times N$ matrix by $\boldsymbol{0}_{M \times N}$. For a vector $\boldsymbol{a}$, diag($\boldsymbol{a}$) represents a diagonal matrix with the elements of $\boldsymbol{a}$ on its diagonal. Finally, we use $\boldsymbol{h_0}^{\mathrm{T}}$ to denote the sequence $\boldsymbol{h_0}$, $\boldsymbol{h_1}$, ..., $\boldsymbol{h_T}$.

4. System model

Consider an OFDM system where a sequence of $T + 1$ symbols $\mathcal{X}_0, \mathcal{X}_1, \cdots, \mathcal{X}_T$, each of length N, are to be transmitted. The data bits to be transmitted are first passed through a convolutional encoder, punctured and interleaved. The resulting bit sequence is mapped to QAM symbols using Gray coding. The OFDM symbol is then constructed by inserting these QAM symbols at data positions and pilot symbols at training positions. Here we consider comb type pilots as they are more robust in fading channels than block type pilots (Nee & Prasad, 2000). Each symbol $\mathcal{X}_i$ then undergoes an IDFT operation to produce the time domain symbol $x_i = \sqrt{N}Q^*\mathcal{X}_i$ where Q is the $N \times N$ DFT matrix. A length P cyclic prefix $\underline{x}_i$ is appended to the symbol x_i to counter the effect of inter symbol interference. The transmitter then transmits the resulting super symbol $\overline{x}_i$ of length $N + P$ as shown in figure (1.1)

Fig. 1.1. Simple OFDM System

We consider a block fading channel model, meaning the channel h_i (length < P +1) remains unchanged for each super symbol but varies from one super symbol to the next according to a state space model

$$h_{i+1} = Fh_i + Gu_i \qquad h_0 \sim \mathcal{N}(0, R_n) \tag{1}$$

where $h_o \sim \mathcal{N}(0, \Pi_0)$, and $\mathbf{u}_o \sim \mathcal{N}(0, \sigma_u^2)$. The matrices F and G are square matrices of size $P \times P$ and are a function of Doppler spread (time correlation), power delay profile (frequency correlation) and the transmit filter. These matrices are assumed to be known at the receiver and are given as

$$F = \begin{bmatrix} \alpha(0) & & \\ & \ddots & \\ & & \alpha(P) \end{bmatrix}, \; G = \begin{bmatrix} \sqrt{1-\alpha^2(0)} & & \\ & \ddots & \\ & & \sqrt{(1-\alpha^2(P))e^{-\beta P}} \end{bmatrix} \tag{2}$$

where $\alpha(p)$ is related to the Doppler frequency $f_D(p)$ by $\alpha(p) = J_0(2\pi f_D T(p))$. The variable β corresponds to the exponent of the channel decay profile. The constraints captured by the

state space model (1) include frequency correlation, time correlation, finite delay spread, and sparsity.

The passage of the $\bar{x}_i$ symbols through the channel h produces the received symbol $\bar{y}_i$ at the receiver. The received symbol of length $N + P$ is split into a length N packet y_i and a length P prefix $\underline{y}_i$. The prefix absorbs all the inter symbol interference between $\bar{x}_{(i-1)}$ and $\bar{x}_i$ and is hence discarded. The time domain relation of the input/output equation can thus be expressed as

$$y_i = x_i \circledast h_i + n_i \tag{3}$$

where $\circledast$ represents circular convolution. This relation takes a more transparent form in the frequency domain as

$$\mathcal{Y}_i = \operatorname{diag}(\mathcal{X}_i)\mathcal{H}_i + \mathcal{N}_i \tag{4}$$

$$= \operatorname{diag}(\mathcal{X}_i)Q_P\underline{h}_i + \mathcal{N}_i \tag{5}$$

where Q_P is the FFT matrix comprised of the first P columns of Q, $\mathcal{H}_i$ is related to h_i by the relation

$$\mathcal{H}_i = Q \begin{bmatrix} h_i \\ 0 \end{bmatrix} = Q_P\underline{h}_i$$

and $\mathcal{N}_i$ is additive white Gaussian noise $\mathcal{N}(0, \sigma_n^2 I)$. For ease of notation, let us define X_i as $\operatorname{diag}(\mathcal{X}_i)Q_P$. Thus we can rewrite equation (5) as

$$\boxed{\mathcal{Y}_i = X_i\underline{h}_i + \mathcal{N}_i} \tag{6}$$

This equation gives the input/output relationship of the OFDM system. Now the OFDM symbol X_i consists of both data and pilots. It will be useful for our discussion in the next sections to define a pilot/output relation. Let $I_p = \{i_1, i_2, \cdots, i_{L_p}\}$ be the set of pilot indices within the OFDM symbol known *a priori* at the receiver and let X_{I_p} be the matrix pruned of rows that do not belong to I_p, i.e. X_{I_p} is comprised of the pilot rows only of X. The pilot/output relationship will thus be a pruned version of (6) and is given as

$$\boxed{\mathcal{Y}_{i,I_p} = X_{i,I_p}\underline{h}_i + \mathcal{N}_{i,I_p}} \tag{7}$$

5. Channel estimation

In the following, we consider the channel estimation problem when the data is (*i*) completely known and when it is (*ii*) partially known (i.e. some training data is available).

5.1 Data completely known

Our goal here is to estimate $\underline{h}_i$. We start by performing channel estimation using only one OFDM symbol. We start with the assumption that the transmitted OFDM symbols X_i are

completely known at the receiver. This enables us to use input/output equation (6) to obtain the channel estimate $\hat{\underline{h}}_i$ by maximizing the following log likelihood function

$$\hat{\underline{h}}_i^{MAP} = \arg \max_{\underline{h}_i} \left\{ \ln \ p\left(\mathcal{Y}_i | X_i, \underline{h}_i\right) + \ln \ p\left(\underline{h}_i\right) \right\} \tag{8}$$

where $p\left(z\right)$ stands for probability density function of z. As the noise is white gaussian, the maximization reduces to

$$\hat{\underline{h}}_i^{MAP} = \arg \min_{\underline{h}_i} \left\{ ||\mathcal{Y}_i - X_i \underline{h}_i||^2_{R_{\mathcal{N}}^{-1}} + ||\underline{h}_i||^2_{\Pi^{-1}} \right\} \tag{9}$$

$$= \Pi X_i^* \left[R_{\mathcal{N}} + X_i \Pi X_i^* \right]^{-1} \mathcal{Y}_i \tag{10}$$

Where $\|a\|^2 A \triangleq a^* A a$. The Maximum A Posteriori (MAP) solution (10) makes use of the information at the ith symbol $\mathcal{Y}_i$ only to estimate $\underline{h}_i$. This is clearly suboptimal as $\underline{h}_j$ is correlated with $\underline{h}_j$ for $j = 0, 1, \ldots, T$ and hence it is correlated with all the output symbols $\mathcal{Y}_0$, $\mathcal{Y}_1, \ldots, \mathcal{Y}_T$(and not just $\mathcal{Y}_i$). To make full use of this correlation, we maximize the log likelihood function of the whole sequence $h_0^T = h_0 , h_1 , \ldots, h_T$, given the whole output sequence $\mathcal{Y}_0^T$, i.e. we maximize the MAP estimate of channel tap sequence $\underline{h}_0^T$ by maximizing the following log likelihood function

$$\mathcal{L} = \ln \ p\left(\mathcal{Y}_0^T | X_0^T, \underline{h}_0^T\right) + \ln \ p\left(\underline{h}_0^T\right) \tag{11}$$

It can be shown that since the noise and channel IR sequence are Gaussian, the likelihood function takes the quadratic form (Al-Naffouri, 2007)

$$\mathcal{L} = \sum_{i=0}^{T} \ln \ p\left(\mathcal{Y}_i | X_i, \underline{h}_i\right) + \sum_{i=1}^{T} \ln \ p\left(\underline{h}_i | \underline{h}_{i-1}\right) + \ln \ p\left(\underline{h}_0\right) \tag{12}$$

$$= -\sum_{i=0}^{T} ||\mathcal{Y}_i - X_i \underline{h}_i||^2_{R_{\mathcal{N}}^{-1}} - \sum_{i=1}^{T} ||\underline{h}_i - F\underline{h}_{i-1}||^2_{\frac{1}{\sigma_u^2} GG^*} - ||\underline{h}_0||^2_{\Pi_0^{-1}} \tag{13}$$

up to some additive constant.
This can be formulated as a least square problem in the sequence $\underline{h}_0^T$. Alternatively, since the MAP estimate coincide with the Minimum Mean Square Error (MMSE) estimate for Gaussian data, we can obtain the sequence $\underline{h}_0^T$ by applying the Forward Backward (FB) Kalman filter to the state space model

$$\underline{h}_{i+1} = F\underline{h}_i + G\underline{u}_i \tag{14}$$

$$\mathcal{Y}_i = X_i \underline{h}_i + \mathcal{N}_i \tag{15}$$

The Forward backward Kalman filter which provides the MAP estimate (and also the MMSE estimate) is described by the following set of equations (Kailath et al., 2000):
Forward run: Starting from the initial conditions $P_{0|-1} = \Pi_0$ and $\underline{h}_{0|-1} = 0$ and for $i = 1,\ldots, T$, calculate

$$R_{e,i} = \sigma_n^2 I_{N+P} + X_i P_{i|i-1} X_i^* \tag{16}$$

$$K_{f,i} = P_{i|i-1} X_i^* R_{e,i}^{-1} \tag{17}$$

$$\hat{\underline{h}}_{i|i} = (I_{N+P} - K_{f,i} X_i)\hat{\underline{h}}_{i|i-1} + K_{f,i}\mathcal{Y}_i \tag{18}$$

$$\hat{\underline{h}}_{i+1|i} = F\hat{\underline{h}}_{i|i} \tag{19}$$

$$P_{i+1|i} = F\left(P_{i|i-1} - K_{f,i} R_{e,i} K_{f,i}^*\right) F^* + \frac{1}{\sigma_n^2} G_\alpha G_\alpha^* \tag{20}$$

Backward run: Starting from $\lambda_{T+1|T} = 0$ and for $i = T, T-1, ..., 0$, calculate

$$\lambda_{i|T} = \left(I_{P+N} - X_i^* K_{f,i}^*\right) F^* \lambda_{i+1|T} + X_i R_{e,i}^{-1} \left(\mathcal{Y}_i - X_i \hat{\underline{h}}_{i|i-1}\right) \tag{21}$$

$$\hat{\underline{h}}_{i|T} = \hat{\underline{h}}_{i|i-1} + P_{i|i-1}\lambda_{i|T} \tag{22}$$

where $\hat{\underline{h}}_{i|T}$ is the desired estimate of the channel taps.

5.2 Data partially known

When only the training part of the data is known, we estimate the channel from the state space model

$$\underline{h}_{i+1} = F\underline{h}_i + G\underline{u}_i \tag{23}$$

$$\mathcal{Y}_{i,I_p} = X_{i,I_p}\underline{h}_i + \mathcal{N}_{i,I_p} \tag{24}$$

This dynamical model is different from the model (14) - (15) through the input equation (specifically X_i is replaced by X_{i,I_p} and $\mathcal{Y}_i$ by $\mathcal{Y}_{i,I_p}$). Thus the training based MAP estimate of the channel sequence $\underline{h}_0^T$ is obtained by applying the Forward Backward Kalman filter to the dynamical model (14) - (15), i.e. by applying (16) - (22) with the substitution $X_i \rightarrow X_{i,I_p}$ and $\mathcal{Y}_i \rightarrow \mathcal{Y}_{i,I_p}$.

5.3 Iterative channel estimation

Using a sufficiently large number of pilots would yield a good estimate of channel but it will consume the valuable bandwidth of the system. For this reason, it is desirable to keep the number of pilots in the system to a minimum. The data, which constitutes the larger part of the received symbol, is not being used and hence we are not fully using the constraints on the data. This provides the motivation for using some data aided technique. Since the data is unknown at the receiver, we make use of the expectation maximization (EM) algorithm. The EM algorithm is used when some of the data needed for the estimation process is unavailable. To motivate the EM algorithm, consider the MAP estimate of the channel $\underline{h}_i$ described by (9). Since the data X_i is unknown, we can not minimize (9) for $\underline{h}_i$. To go around this, we minimize (9) averaged over the data X_i i.e.

$$\hat{\underline{h}}_i^{(j)} = \arg\min_{h_i} \left\{ E_{X_i|\mathcal{Y}_i, h_i^{(j-1)}} \left\{ \ln\ p\left(\mathcal{Y}_i | X_i, \underline{h}_i\right) + \ln\ p\left(\underline{h}_i\right) \right\} \right\} \tag{25}$$

As indicated in (25), this is an iterative procedure as it gives the estimate of $\underline{h}_i$ at the jth iteration by utilizing the estimate at the $(j\text{-}1)^{th}$ iteration. Thus, the expectation with respect to X_i is taken given the output data $\mathcal{Y}_i$ and given the channel estimate at the $(j\text{ -}1)^{th}$ step, $\hat{\underline{h}}_i^{(j-1)}$. By evaluating the expectation and completing the squares, we can rewrite (25) as

$$\hat{\underline{h}}_i^{(j)} = \arg\min_{\underline{h}_i} \left\{ \|\mathcal{Y}_i - E[X_i]\underline{h}_i\|^2_{\frac{1}{\sigma_n^2}} + \|\underline{h}_i\|^2_{\frac{1}{\sigma_n^2} Cov[X_i^*]} + \|\underline{h}_i\|^2_{\Pi^{-1}} \right\} \tag{26}$$

where $E[X_i]$ is the expected value of X_i and $Cov[X_i^*]$ is its covariance. Combining the first two terms of the above equation, we can get a concise form of the log likelihood function as

$$\hat{\underline{h}}_i^{(j)} = \arg\min_{\underline{h}_i} \left\{ \left\| \begin{bmatrix} \mathcal{Y}_i \\ 0_{P\times 1} \end{bmatrix} - \begin{bmatrix} E[X_i] \\ Cov[X_i^*]^{1/2} \end{bmatrix} \underline{h}_i \right\|^2_{\frac{1}{\sigma_n^2}} + \|\underline{h}_i\|^2_{\Pi^{-1}} \right\} \tag{27}$$

Comparing the quadratic form (9) for the known data case with the form (27) for the unknown (or partially known) data case, we note that to move from the former to the latter, we need to perform the substitution

$$X_i \rightarrow \begin{bmatrix} E[X_i] \\ Cov[X_i^*]^{1/2} \end{bmatrix} \quad \text{and} \quad \mathcal{Y}_i \rightarrow \begin{bmatrix} \mathcal{Y}_i \\ 0_{P\times 1} \end{bmatrix}$$

This step is the maximization step of the EM algorithm. It remains to perform the expectation step which will be considered below.

As $X_i = \text{diag}(\mathcal{X}_i)Q_P$, we can express the mean and the covariance of X_i in terms of $\mathcal{X}_i$ as

$$E[X_i|\mathcal{Y}_i, \hat{h}_i^{(j-1)}] = \text{diag}(E[\mathcal{X}_i|\mathcal{Y}_i, \hat{h}_i^{(j-1)}])Q_P \tag{28}$$

$$Cov[X_i^*|\mathcal{Y}_i, \mathcal{H}_i] = Q_P^* Cov[\mathcal{X}_i^*|\mathcal{Y}_i, \mathcal{H}_i]Q_P \tag{29}$$

This decoupled form enables us to calculate the above expectations on an element by element basis $\mathcal{X}_i(l)$ $l = 1, 2,...,N$. For the case when $\mathcal{X}_i(l)$ is drawn from a finite alphabet set $A = \{A_1, A_2..., A_{|A|}\}$ with equal probability, we can show that

$$f(\mathcal{X}_i(l)|\mathcal{Y}_i(l), \mathcal{H}_i(l)) = \frac{e^{-\frac{|\mathcal{Y}(l) - \mathcal{H}(l)A_j|^2}{\sigma^2}}}{\sum_{j=1}^{M} e^{-\frac{|\mathcal{Y}(l) - \mathcal{H}(l)A_j|^2}{\sigma^2}}} \tag{30}$$

using Bayes rule. This result leads to the expectations

$$E[\mathcal{X}_i(l)|\mathcal{Y}_i(l), \mathcal{H}_i(l)] = \frac{\sum_{j=1}^{M} A_j e^{-\frac{|\mathcal{Y}_i(l) - \mathcal{H}_i(l)A_j|^2}{\sigma^2}}}{\sum_{j=1}^{M} e^{-\frac{|\mathcal{Y}_i(l) - \mathcal{H}_i(l)A_j|^2}{\sigma^2}}} \tag{31}$$

$$E[|\mathcal{X}_i(l)|^2|\mathcal{Y}_i(l), \mathcal{H}_i(l)] = \frac{\sum_{j=1}^{M} |A_j|^2 e^{-\frac{|\mathcal{Y}_i(l)-\mathcal{H}_i(l)A_j|^2}{\sigma^2}}}{\sum_{j=1}^{M} e^{-\frac{|\mathcal{Y}_i(l)-\mathcal{H}_i(l)A_j|^2}{\sigma^2}}} \tag{32}$$

which can in turn be used to evaluate the expectation and covariance of X_i.

The initial estimate $\hat{\underline{h}}_i^{(0)}$ is obtained using pilots. From this channel estimate, we make an estimate of data and use (26) to estimate the channel again. This process is iterated a number of times to get the final estimate $\hat{\underline{h}}_i$.

6. The EM-based forward backward Kalman filter

The algorithm above performs iterative channel estimation (of $\underline{h}_i$) and data detection (of $\mathcal{X}_i$) using the ith OFDM symbol $\mathcal{Y}_i$ only. However, as we mentioned earlier, this is suboptimal as the channel $\underline{h}_i$ is correlated with the symbols $\mathcal{Y}_0$, $\mathcal{Y}_1$,..., $\mathcal{Y}_T$. Thus, in the partially known data case, we can design an EM based FB Kalman filter as follows

Maximization Step: The maximization step obtains estimate of $\underline{h}_i$ at the jth iteration by applying the FB Kalman to the state space model

$$\underline{h}_{i+1} = F\underline{h}_i + G\underline{u}_i \tag{33}$$

$$\begin{bmatrix} \mathcal{Y}_i \\ 0_{P \times 1} \end{bmatrix} = \begin{bmatrix} E[X_i] \\ Cov[X_i^*]^{1/2} \end{bmatrix} \underline{h}_i + \begin{bmatrix} \mathcal{N}_i \\ \underline{n}_i \end{bmatrix} \tag{34}$$

i.e. by applying the FB Kalman (16) - (22) with the change of variables

$$X_i \rightarrow \begin{bmatrix} E[X_i] \\ Cov[X_i^*]^{1/2} \end{bmatrix} \quad \text{and} \quad \mathcal{Y}_i \rightarrow \begin{bmatrix} \mathcal{Y}_i \\ 0_{P \times 1} \end{bmatrix}$$

Expectation Step: The expectation step in the FB Kalman case is exactly the same as described by equations (28) – (32)

Initialization Step: The initial channel estimate $\hat{\underline{h}}_i^{(0)}$ is obtained by applying the FB Kalman filter to the dynamical model (23) – (24).

There are several possible implementations of incorporating the EM algorithm in the FB Kalman filter. For time correlated channels, there are two dimensions we can iterate (*i*) between channel estimation & data detection or (*ii*) against time. We get different receiver structures depending upon the scheduling of these iterations.

6.1 Cyclic FB Kalman
If we iterate between channel estimation & data detection after the entire forward-backward run of the Kalman, the filter formed is called a cyclic FB Kalman. The pilot based estimate serves as the initial estimate to jump start the data aided version. The channel estimate is then given by the FB Kalman. This estimate is again used to refine the data and the entire process is iterated. The iterations process the OFDM symbols in a circular manner motivating the name of this filter structure as shown in figure 1.2.
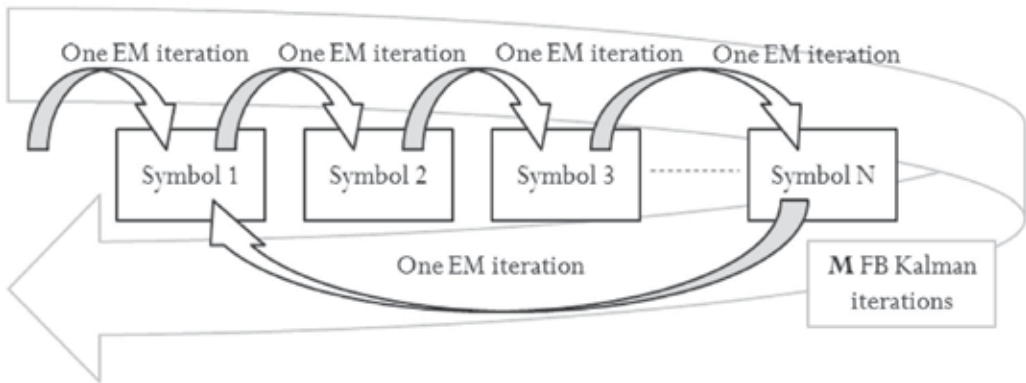
Fig. 1.2. Cyclic FB Kalman
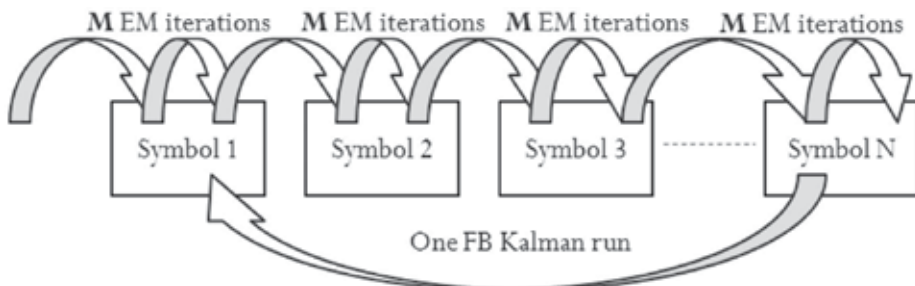
Fig. 1.3. Helix FB Kalman

6.2 Helix FB Kalman

If we structure the filter to iterate between channel estimation and data detection at each step of the Kalman, the filter formed is called a helix FB Kalman. Here, the initial estimate is again provided by the pilots. Now, at each step of the Kalman, the filter iterates several times between channel estimation and data detection at each OFDM symbol, using one to enhance the other and so on, before moving to the next symbol using the Kalman filter. Figure 1.3 shows that the iterations trace a helix which explains the name we chose for this filter.

6.3 Forward only Kalman

Latency and memory is an issue with the FB Kalman filter as the receiver needs all the symbols within an OFDM block to estimate the channel. By implementing the forward part of the Kalman filter only, we can get rid of the storage and latency issues at the cost of decreased performance.

7. The FB-Kalman based receivers

The FB Kalman filter described in the previous section can be applied in several other receiver scenarios. To do so, we need to provide the following

1. A dynamical equation that describes the evolution of the channel impulse response (similar to (1)). The FB Kalman can be applied then to obtain the iterative channel estimate (maximization step).

2. Input/output equation for data recovery. This is used to evaluate the first and second moments of the data (expectation step).
3. A dynamical equation that can be used for initial channel estimation via the FB Kalman filter.

In the following, we shall apply this procedure to two communication scenarios namely channel estimation in multiuser OFDM and channel estimation in MIMO OFDM.

8. Channel estimation in multi-access OFDM systems

In a multi-access OFDM system, the available bandwidth of the system is shared between a number of users. Each user thus has access to only a specific portion of the OFDM spectrum. This has some important ramifications as far as channel estimation is concerned. Specifically, each user is only interested in the estimate of the particular band in which it is operating. In time domain based channel estimation techniques the user is required to estimate the entire spectrum and hence this proves to be computationally expansive in the multi-user case. As such, frequency domain channel estimation methods for the multi-access scenario would make more sense as this would allow each user to estimate the part of spectrum in which it is operating and not the entire spectrum.

The only problem with frequency domain channel estimation is the increased number of parameters to be estimated in this case (from P to N, where P is the number of time domain channel taps and N is the number of frequency bins used). As such, we can use some parameter reduction model to reduce the number of parameters to be estimated.

8.1 A parameter reduction approach

Let k users be operating in a multi-access OFDM system. The frequency response of the entire spectrum is of length N. For simplicity, we will assume that all users share the bandwidth equally. Each user thus operates in a band (or section) of length $L_f = \lceil \frac{N}{k} \rceil$. Let $\underline{\mathcal{H}}_i^{(j)}$ be the jth section of the frequency response, then from (1), the input/output equation of the jth user is given by

$$\underline{\mathcal{Y}}_i^{(j)} = \mathrm{diag}(\underline{\mathcal{X}}_i^{(j)})\underline{\mathcal{H}}_i^{(j)} + \underline{\mathcal{N}}_i^{(j)} \tag{35}$$

where $\underline{\mathcal{Y}}_i^{(j)}$, $\underline{\mathcal{X}}_i^{(j)}$, $\underline{\mathcal{H}}_i^{(j)}$ and $\underline{\mathcal{N}}_i^{(j)}$ are the jth sections of $\underline{\mathcal{Y}}_i$, $\underline{\mathcal{X}}_i$, $\mathcal{H}_i$ and $\mathcal{N}_i$, respectively. We will suppress the dependence on the user index j and time index i for notational convenience. Denoting the pilot locations by the subscript $I_p^{(j)}$, we can write the pilot/output equation as

$$\underline{\mathcal{Y}}_{I_p} = \mathrm{diag}(\underline{\mathcal{X}}_{I_p})\underline{\mathcal{H}} + \underline{\mathcal{N}}_{I_p} \tag{36}$$

There are not enough pilots to estimate $\underline{\mathcal{H}}$. So, we resort to model reduction starting from the autocorrelation function of $\underline{\mathcal{H}}$, $R_{\mathcal{H}}$. The eigenvalue decomposition of $R_{\mathcal{H}}$ is given by

$$R_{\mathcal{H}} = \sum_{l=1}^{L_f} \lambda_l \mathbf{v}_l \mathbf{v}_l^T \tag{37}$$

where $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_{L_f}$ are the (ordered) eigenvalues of $R_{\mathcal{H}}$ and $\mathbf{v}_1, \ldots, \mathbf{v}_{L_f}$ are the corresponding eigenvectors. Using this decomposition, $\underline{\mathcal{H}}$ an be represented as

$$\underline{\mathcal{H}} = \sum_{l=1}^{L_f} \alpha_l \mathbf{v_1} \tag{38}$$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_{L_f}]^T$ is the parameter vector to be estimated, with zero mean and autocorrelation matrix $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{L_f})$. Using (37) and (38), we can represent $\underline{\mathcal{H}}$ in terms of its dominant eigenvalues and treat the insignificant eigenvalues as modeling noise., i.e.

$$\underline{\mathcal{H}} = V_d \alpha_d + V_n \alpha_n \tag{39}$$

Upon substituting (39) in (35), we get

$$\underline{\mathcal{Y}} = \mathrm{diag}(\underline{\mathcal{X}}) V_d \alpha_d + \underline{\mathcal{N}} + \mathrm{diag}(\underline{\mathcal{X}}) V_n \alpha_n \tag{40}$$

$$= \underline{X}_d \alpha_d + \underline{\mathcal{N}}' \tag{41}$$

where $\underline{X}_d = \mathrm{diag}(\underline{\mathcal{X}}) V_d$ and $\underline{\mathcal{N}}' = \underline{\mathcal{N}} + \underline{X}_n \alpha_n$ with $\underline{X}_n = \mathrm{diag}(\underline{\mathcal{X}}) V_n$. The noise $\underline{\mathcal{N}}'$ includes both the *modeling* noise and the additive Gaussian noise. We consider $\underline{\mathcal{N}}'$ to be zero mean white Gaussian noise with autocorrelation

$$R_{\underline{\mathcal{N}}'} = R_{\underline{\mathcal{N}}} + \mathrm{diag}(\underline{\mathcal{X}}) V_n \mathrm{diag}(\lambda_n) V_n^* \mathrm{diag}(\underline{\mathcal{X}})^* \tag{42}$$

Based on (41) we can construct a pilot/output equation similar to (36), as

$$\boxed{\underline{\mathcal{Y}}_{I_p} = \underline{X}_{d,I_p} \alpha_d + \underline{\mathcal{N}}'_{I_p}} \tag{43}$$

Equation (41) is the equation we need for data recovery. Equation (43) can be used for initial channel estimation (See (Al-Naffouri & M. Sohail, 2008) for further details). All that is needed now is to develop a dynamical model for the interpolation parameter.

8.2 Developing a frequency domain time-variant model

In this subsection, we develop a state space model for the parameter α_d. To this end, let us consider the channel model of (1) and assume for simplicity that the diagonal matrices F and G are actually scalar multiples of the identity, i.e.

$$F = fI \qquad G = \sqrt{1 - f^2} I$$

where f is a function of Doppler frequency (see (Al-Naffouri, 2007)). Now as $\mathcal{H}_i$ is just the channel response h_i in frequency domain ($\mathcal{H}_i = Q_P h_i$), the jth section of $\mathcal{H}_i$ (i.e. $\mathcal{H}_i^{(j)}$) is related to h_i by

$$\underline{\mathcal{H}}_i^{(j)} = Q_P^{(j)} h_i \tag{44}$$

where $Q_P^{(j)}$ is the jth section of Q_P. Replacing $\underline{\mathcal{H}}_i^{(j)}$ in (44) by its representation, we get a relation between the time domain channel response and the dominant parameters α_d as

$$V_d \alpha_{d,i} = Q_P^{(j)} h_i \tag{45}$$

$$\alpha_{d,i} = V_d^+ Q_P^{(j)} h_i \tag{46}$$

where V_d^+ is the pseudo inverse of V_d. Combining (1) and (46) yields a dynamical recursion for α_d

$$\boxed{\alpha_{d,i+1} = F_\alpha \alpha_{d,i} + G_\alpha u_i} \tag{47}$$

where $F_\alpha = fI$ and $G_\alpha = \sqrt{1-f^2} V_d^+ Q_P^{(j)}$ and where

$$E[\alpha_{d,0} \alpha_{d,0}^*] = \Lambda_d \tag{48}$$

here we suppress the dependence of G_α and α_d on j for notational convenience.

We are now ready to implement our FB Kalman based receiver, which consists of an initial channel estimation step and an iterative channel estimation step.

8.3 Initial (pilot-based) channel estimation

In multi-access OFDM, the initial estimate is given by applying the FB Kalman filter (16) - (22) to the following state space model:

$$\underline{\mathcal{Y}}_{I_p,i} = \underline{X}_{d,I_p,i} \alpha_{d,i} + \underline{\mathcal{N}}'_{I_p,i} \tag{49}$$

$$\alpha_{d,i+1} = F_\alpha \alpha_{d,i} + G_\alpha u_i \tag{50}$$

8.4 Iterative (data-aided) channel estimation

The iterative channel estimation step is obtained by applying FB Kalman to the state space model

$$\underline{\mathcal{Y}}_i = \begin{bmatrix} E[\underline{X}_{d,i}] \\ Cov[\underline{X}_{d,i}^*]^{\frac{1}{2}} \end{bmatrix} \alpha_{i,d} + \begin{bmatrix} \underline{\mathcal{N}}'_i \\ 0 \end{bmatrix} \tag{51}$$

$$\alpha_{d,i+1} = F_\alpha \alpha_{d,i} + G_\alpha u_i \tag{52}$$

The data expectations in (51) are obtained from the input/output equation (41). Consequently, the FB Kalman is applied to the above set. As mentioned in Section 5, the FB Kalman can be implemented as Cyclic Kalman, Helix Kalman or forward only Kalman.

Figure 1.4 compares the Bit Error Rate (BER) performance of three implementations of Kalman filter with the simple EM based least square estimation method. We consider an OFDM system that transmits 6 symbols with 64 carriers and a cyclic prefix of length $P = 15$ each with a time variation of $f = 0.9$. The data bits are mapped to 16 QAM through Gray coding. The OFDM symbol serves 4 users each occupying 16 frequency bins. In addition, the OFDM symbol carries 16 pilots equally divided between the users. The channel impulse response consists of 15 complex taps (the maximum length possible). It has an exponential delay profile $E[|h_0(k)|^2] = e^{-0.2k}$ and remains fixed over any OFDM symbol. As expected, the

estimate improves when we use time correlation information (by using a Kalman filter). Figure 1.4 is plotted for the uncoded case. Since we are relying on the data to improve the channel estimate, an outer code can improve the data quality and hence the quality of the channel estimate. Figure 1.6 shows the advantage of using coding (1/2 rate convolutional code) during the iterative process over the performance of iterative process that only makes use of the code to correct the data at the last step of the algorithm.
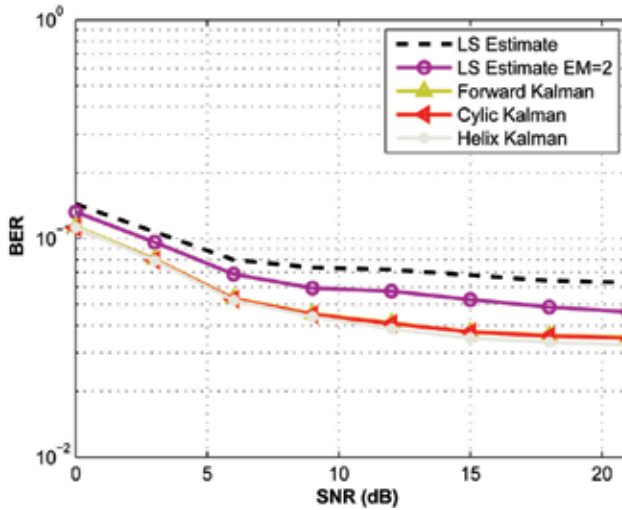
Fig. 1.4. Comparison of various uncoded Kalman implementations.
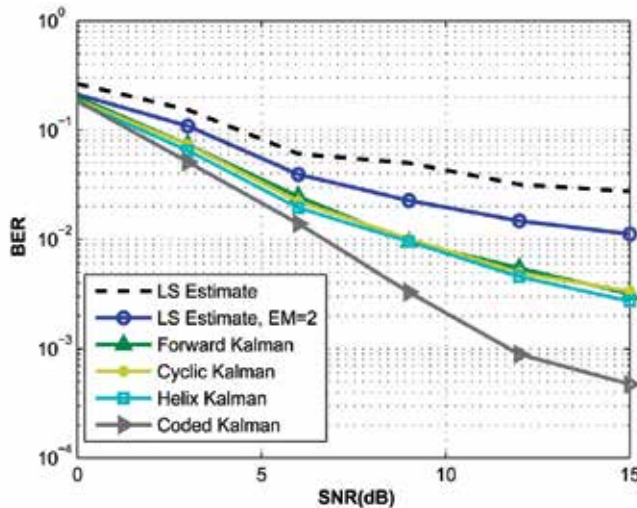
Fig. 1.5. Comparison of Kalman implementations using code.

9. Channel estimation in MIMO OFDM systems

As wireless communication is becoming more and more integrated in everyday life, service operators are offering new and advanced features like video streaming and broadband connections. These services demand high spectral performance. By using Multi Input Multi

Output (MIMO) OFDM techniques, we can increase the spectral efficiency and throughput of wireless systems. As we have seen so far, the problem of channel estimation in a wireless system is a challenging one. In the case of MIMO OFDM systems, it becomes compounded by the fact that transmitters and receivers employ multiple antennas which substantially increase the number of parameters to be estimated. In this section, we will extend the FB Kalman receiver to a Space Time Block Coded (STBC) MIMO OFDM system.

9.1 MIMO channel model

We start by defining the MIMO channel model. Thus consider a MIMO system with T_x transmission antennas and R_x receiver antennas. The time domain input/output relationship for a general MIMO system can be described by

$$y(m) = \sum_{p=0}^{P} H(p)x(m-p) \tag{53}$$

where $H(p)$ is the $R_x \times T_x$ MIMO impulse response at tap p and where m represents the sample time index. The effect of the transmit filter and transmit and receive correlation are incorporated in $H(p)$ making it correlated across space and taps. For simplicity, $H(p)$ is assumed to be iid. Again, a block fading channel model is considered where the changes from the current block ($H_t(p)$) to the next block ($H_{t+1}(p)$) occur according to the dynamical equation

$$H_{t+1}(p) = \alpha(p)H_t(p) + \sqrt{(1-\alpha^2(p))}e^{-\beta p}U_t(p) \tag{54}$$

where $\alpha(p)$ is related to the Doppler frequency $f_D(p)$ by $\alpha(p) = J_0(2\pi f_D(p)T)$ (T being the time duration of one ST block), β is the exponent of the channel decay profile while the factor $\sqrt{(1-\alpha^2(p))}e^{-\beta p}$ represents the exponential decay profile ($e^{-\beta p}$) for all time and $U_t(p)$ $\sim \mathcal{N}(0, 1)$ is an iid matrix. The model approximates the non rational Jakes model by a 1st order AR model. A higher order AR model would give a better approximation but at the expense of increased latency at the receiver. From (54), we can obtain the impulse response $\underline{h}_{r_x}^{t_x}$ between transmit antenna t_x and receive antenna r_x,

$$h_{r_x t+1}^{t_x}(p) = \alpha(p)h_{r_x t}^{t_x}(p) + \sqrt{(1-\alpha^2(p))}e^{-\beta p}u_{r_x t}^{t_x}(p) \tag{55}$$

Stacking (55) over the taps $p = 0, 1,..., P$, we get the dynamical model

$$\underline{h}_{r_x t+1}^{t_x} = F\underline{h}_{r_x}^{t_x} + Gu_{r_x}^{t_x} \tag{56}$$

where $\underline{h}^{tx}{}_{rx}$ is the channel IR at $r_x = 1, ..., R_x$ and $t_x = 1, ..., T_x$, and F and G are the same as given by (2). Stacking (56) for all transmit and receive antennas, we obtain

$$\boxed{\underline{h}_{t+1} = (I_{T_x R_x} \otimes F)\,\underline{h}_t + (I_{T_x R_x} \otimes G)\,u_t} \tag{57}$$

where

$$\underline{h}_t = \begin{bmatrix} \underline{h}_1^1 & \cdots & \underline{h}_1^{T_x} & \underline{h}_2^1 & \cdots & \underline{h}_2^{T_x} & \cdots & \underline{h}_{R_x}^1 & \cdots & \underline{h}_{R_x}^{T_x} \end{bmatrix}^T$$

and where the vectors $\underline{h}_{t+1}$, $\underline{h}_t$, and u_t, are of size $T_x R_x(P) \times 1$. To employ the Kalman filter, we still need to characterize the covariance information of the dynamical model. Specifically we need the covariance of u_t and also the covariance of the channel at $t = 0$. It can be shown that

$$E\left[u_t u_t^*\right] \quad = \quad I_{R_x} \otimes E\left[u_{r_x} u_{r_x}^*\right] \tag{58}$$

$$= \quad I_{R_x} \otimes \left(I_{T_x} \otimes E\left[u_{r_x}^{t_x} u_{r_x}^{t_x *}\right]\right) \tag{59}$$

$$= \quad I_{R_x} \otimes I_{T_x} \otimes I_P = I_{T_x R_x(P)} \tag{60}$$

and

$$E\left[\underline{h}_0 \underline{h}_0^*\right] = I_{T_x R_x} \otimes GG^*$$

It is worthwhile to note that while (54) and (57) are equivalent, the latter model is in vector form and will be useful for Kalman filter operations.

In the channel model described above, we did not consider the transmit/receive correlation between the antennas. When both transmit/receive correlation are incorporated in the channel model, the dynamical equation remains same as in (57) but the covariance of u_t is given by (see (Al-Naffouri & Quadeer, 2008) for further details)

$$E[uu^*] = \sum_{p=0}^{P} R(p) \otimes T(p) \otimes \left(\underline{I}^p B \overline{I}^p\right)$$

where $T(p)$ and $R(p)$ are the transmit and receive correlation matrix (of size T_x and R_x) respectively.

The receiver will perform two functions namely channel estimation and data detection. So we need to derive two forms of the input/output equation. The first is a *channel estimation* form, which treats the channel impulse response as the unknown and which together with the dynamical model (57) forms the state space model that is used by the FB Kalman. The second is a *data detection* form, which treats the input in its uncoded form as the unknown. In the Single Input Single Output (SISO) case, we defined the frequency domain input/output relationship by (4). For the MIMO OFDM case, the input/output relation between transmit antenna t_x and receive antenna r_x is given by

$$\mathcal{Y}_{r_x}^{t_x} = \mathrm{diag}\left(\mathcal{X}_{t_x}\right) Q_P^* \underline{h}_{r_x}^{t_x} + \mathcal{N}_{r_x} \tag{61}$$

By stacking, the input/output equation at receive antenna r_x can be expressed as

$$\mathcal{Y}_{r_x} = \left[\mathrm{diag}(\mathcal{X}_1) \ \cdots \ \mathrm{diag}(\mathcal{X}_{T_x})\right] (I_{T_x} \otimes Q_P^*) \underline{h}_{r_x} + \mathcal{N}_{r_x} \tag{62}$$

In what follows, we describe the channel estimation version of the input/output equation for STBC MIMO OFDM transmission over block fading channels. We omit the data detection version as it is similar to the SISO case and as it does not directly relate to the operation of the FB Kalman, which is the center of attention of this chapter. The reader can find more information in (Al-Naffouri & Quadeer, 2008).

9.2 Input/output equation

The input/output equation in STBC MIMO OFDM transmission can be divided into the following two categories:

1.   Channel Estimation Version
2.   Data Detection Version

Channel Estimation Version

Consider a MIMO OFDM system which has N_c time slots, T_x transmit antennas and R_x receive antennas. Let N_u be the set of uncoded OFDM symbols $\{\boldsymbol{\mathcal{S}}(1),..., \boldsymbol{\mathcal{S}}(N_u)\}$ which are to be transmitted. We can implement the Alamouti (ST) code using the set of $T_x \times N_c$ matrices $\{\boldsymbol{A}(1), \boldsymbol{B}(1),..., \boldsymbol{A}(Nu), \boldsymbol{B}(Nu)\}$ following the procedure in (Larsson & Stoica, 2003). Thus, the OFDM symbol transmitted from antenna t_x at time n_c is given by

$$\boldsymbol{\mathcal{X}}_{t_x}(n_c) = \sum_{n_u=1}^{N_u} a_{t_x,n_c}(n_u)\mathrm{Re}\,\boldsymbol{\mathcal{S}}(n_u) + jb_{t_x,n_c}(n_u)\mathrm{Im}\,\boldsymbol{\mathcal{S}}(n_u) \tag{63}$$

where $a_{t_x,n_c}(n_u)$ is the (t_x, n_c) element of $X(n_u)$ and $b_{t_x,n_c}(n_u)$ is the (t_x, n_c) element of $B(n_u)$. So, instead of (62), the input/output equation at antenna r_x at OFDM symbol n_c of a ST block takes the form

$$\boldsymbol{\mathcal{Y}}_{r_x}(n_c) = [\mathrm{diag}(\boldsymbol{\mathcal{X}}_1(n_c)) \cdots \mathrm{diag}(\boldsymbol{\mathcal{X}}_{T_x}(n_c))](I_{T_x} \otimes \boldsymbol{Q}_P^*)\underline{\boldsymbol{h}}_{r_x} + \boldsymbol{\mathcal{N}}_{r_x}(n_c)$$

Stacking over all symbols yields

$$\boldsymbol{\mathcal{Y}}_{r_x} = X\underline{\boldsymbol{h}}_{r_x} + \boldsymbol{\mathcal{N}}_{r_x} \tag{64}$$

where

$$X = \begin{bmatrix} \{\mathrm{diag}(\boldsymbol{\mathcal{X}}_1(1)) \cdots \mathrm{diag}(\boldsymbol{\mathcal{X}}_{T_x}(1))\}(I_{T_x} \otimes \boldsymbol{Q}_{P+1}^*) \\ \{\mathrm{diag}(\boldsymbol{\mathcal{X}}_1(2)) \cdots \mathrm{diag}(\boldsymbol{\mathcal{X}}_{T_x}(2))\}(I_{T_x} \otimes \boldsymbol{Q}_{P+1}^*) \\ \vdots \\ \{\mathrm{diag}(\boldsymbol{\mathcal{X}}_1(N_c)) \cdots \mathrm{diag}(\boldsymbol{\mathcal{X}}_{T_x}(N_c))\}(I_{T_x} \otimes \boldsymbol{Q}_{P+1}^*) \end{bmatrix} \tag{65}$$

Further stacking this relation for all receive antennas, we get an input/output relationship at *all* frequency bins, for *all* input and output antennas, and for *all* OFDM symbols of the tth ST block.

$$\boxed{\boldsymbol{\mathcal{Y}}_t = (I_{R_x} \otimes X_t)\underline{\boldsymbol{h}}_t + \boldsymbol{\mathcal{N}}_t} \tag{66}$$

Pruning (66), we get the set of those equations where the pilots are present and the pilot/output equation takes the form

$$\boxed{\boldsymbol{\mathcal{Y}}_{t_{I_p}} = (I_{R_x} \otimes X_{t_{I_p}})\underline{\boldsymbol{h}}_t + \boldsymbol{\mathcal{N}}_{t_{I_p}}} \tag{67}$$

Data Detection Version

Signal detection in MIMO case is done in the same fashion as the SISO case, i.e. on a tone-by-tone basis except that the tones are collected for R_x receive antennas and over N_c time slots (the whole ST block). As mentioned above, we will omit the details of the data detection version as it is similar to the SISO case discussed in Section 5.3. The reader is also invited to check (Al-Naffouri & Quadeer, 2008) for more details.

9.3 Channel estimation using EM based FB Kalman

As we showed in Section 5, the MAP estimate for a sequence of $T + 1$ input and output symbols X_0^T and $\mathcal{Y}_0^T$ is given by (11). For the MIMO case, we can use (57) and (66) to express the channel log likelihood as

$$
\begin{aligned}
\mathcal{L} \;=\; & -\sum_{t=1}^{T} \left\| \mathcal{Y}_t - (I_{R_x} \otimes X_t)\underline{h}_t \right\|^2_{\frac{1}{\sigma_n^2}} \\
& -\sum_{t=1}^{T} \left\| \underline{h}_t - (I_{T_x R_x} \otimes F)\underline{h}_{t-1} \right\|^2_{(GR_u G^*)^{-1}} - \left\| \underline{h}_0 \right\|^2_{\Pi_0^{-1}}
\end{aligned}
\tag{68}
$$

In what follows, we present the known and the unknown input data cases for channel estimation.

Known Input Case (Initial Channel Estimation)

Let us start with the simple case of known input i.e. when pilots are used to estimate the channel. The MAP estimate of $\underline{h}_0^T$ for the input and output sequences X_0^T and $\mathcal{Y}_0^T$ respectively can be obtained by applying the FB Kalman (16) - (22) on the following state-space model

$$
\begin{aligned}
\underline{h}_{t+1} \;&=\; (I_{T_x R_x} \otimes F)\underline{h}_t + (I_{T_x R_x} \otimes G)u_t \\
\mathcal{Y}_{t_{I_p}} \;&=\; (I_{R_x} \otimes X_{t_{I_p}})\underline{h}_t + \mathcal{N}_{t_{I_p}}
\end{aligned}
$$

where $\underline{h}_0 \sim \mathcal{N}(0, \Pi)$ and $u_t \sim \mathcal{N}(0, R_u)$.

Unknown Input Case

When input is unknown, the EM algorithm can be used to estimate the channel similar to what we described in Section 6. In this case, we maximize the averaged form of log-likelihood (68). Thus, the jth iteration of the EM algorithm is now obtained by maximizing the averaged log-likelihood

$$
\overline{\mathcal{L}} = E_{X_0^T | \mathcal{Y}_0^T, h_0^{T(j-1)}}[\mathcal{L}]
\tag{69}
$$

As shown in Section 5.3, this is done by representing the input by its first and second moments and applying the FB Kalman (16) - (22) to the following state space model

$$
\underline{h}_{t+1} \;=\; (I_{T_x R_x} \otimes F)\underline{h}_t + (I_{T_x R_x} \otimes G)u_t
\tag{70}
$$

$$
\begin{bmatrix} \mathcal{Y}_t \\ 0_{T_x R_x (P) \times 1} \end{bmatrix}
=
\begin{bmatrix} I_{R_x} \otimes E[X_t] \\ I_{R_x} \otimes \mathrm{Cov}[X_t^*]^{1/2} \end{bmatrix} \underline{h}_t
+
\begin{bmatrix} \mathcal{N}_t \\ \underline{n}_t \end{bmatrix}
\tag{71}
$$

where $\underline{n}_t$ is virtual noise and is independent of the physical noise $\mathcal{N}_t$.

9.4 Data detection

The data is detected by using the data detection version of the input/output equation which in turn is used to obtain the first and second moments of the inputs needed in (70)-(71). This is similar to the SISO case described in Section 5.3. The reader can also refer to (Al-Naffouri & Quadeer, 2008) for more details.

To show the favorable behavior of the algorithm, we simulate a MIMO OFDM system in which a 1/2 rate convolutional encoder is used as an outer encoder. 16-QAM with Gray

coding is used as the modulation scheme. Orthogonal space time block coding (OSTBC) commonly known as Alamouti code (number of time slots, N_s = 2 and number of transmitters, T_x = 2) is used (Alamouti, 1998). Other parameters used are α = 0.985, β = 0.2, and P = 7. Each packet consists of 12 OFDM symbols transmitted over six ST blocks. Each OFDM symbol consists of 64 frequency tones and a cyclic prefix of length 8. The first ST block is comprised of 16 pilots while the number of pilots in subsequent blocks can be varied from zero to 16.
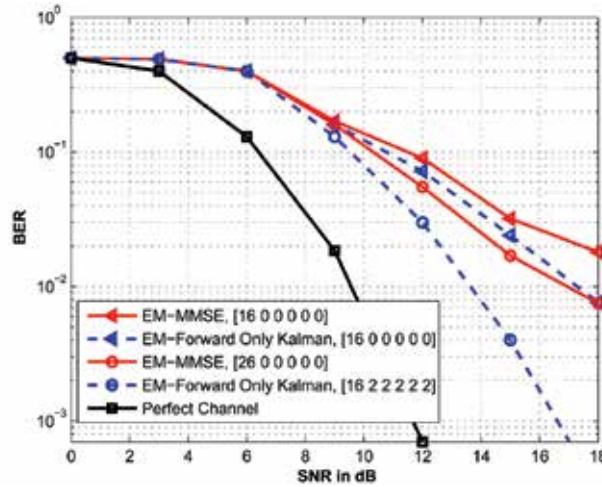
Fig. 1.6. Comparison of EM-MMSE and EM-FB Kalman algorithms

To benchmark the proposed algorithm, it is compared with an EM-based iterative MMSE receiver (proposed in (Li et al., 2001) and (Cozzo & Hughes, 2003)) in Figure 1.6 over a spatially white channel. The Expectation step in this algorithm is calculated through MMSE estimation i.e. by a conditional expectation of the channel given the received symbol and the current estimate of the transmitted data. The Maximization step is simply the maximum likelihood estimate of the transmitted data. In this algorithm, the pilots are confined only to the first space time (ST) block which is used to produce the initial channel estimate. In Figure 1.6, the two algorithms are compared for two scenarios with respect to pilots. In first scenario, 16 pilots are used in the first ST block and zero pilots in the subsequent ST blocks. In the second scenario, the EM-MMSE algorithm has 26 pilots in the first ST block and zero pilots in the subsequent ones while the proposed algorithm (EM-FB Kalman) has 16 pilots in the first ST block and 2 pilots in the subsequent blocks, ensuring the same pilot overhead. It can be easily observed that our algorithm outperforms the EM-MMSE algorithm in both scenarios.

Figure 1.7 describes the effect of spatial correlation over performance of the algorithm for a MIMO OFDM transmission. The parameters used here are same except that channel length, P = 15, α = 0.8, and transmit and receive correlation matrices are given by

$$T(p) = \begin{bmatrix} 1 & \zeta \\ \zeta & 1 \end{bmatrix} \quad \text{and} \quad R(p) = I$$

where ζ = 0.8. The number of pilots in subsequent blocks is fixed at 12 and two EM iterations are used. It can be observed that the performance of both Forward only and FB

Kalman is better over spatially correlated channel (practical scenario) as compared to their performance over spatially white channel.
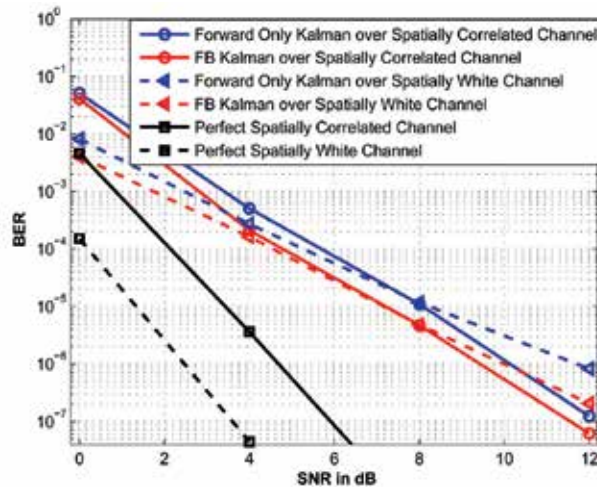
Fig. 1.7. BER performance of Forward only Kalman and FB Kalman using Soft data over spatially white and correlated channel models

Figure 1.8 compares the performance of the different implementations of the Kalman filter (Forward Only Kalman, Cyclic FB Kalman and Helix based FB Kalman) over spatially correlated channel. It can be seen that the Helix based FB Kalman filter outperforms the other two implementations.
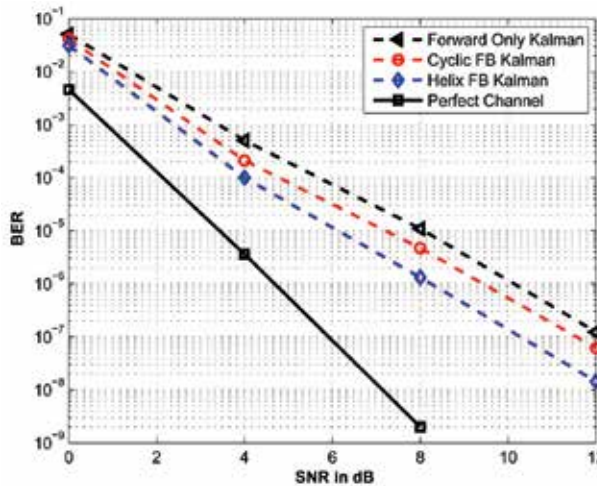
Fig. 1.8. Comparison of Forward only Kalman, Cyclic FB Kalman, and Helix FB Kalman with 12 pilots over spatially correlated channel

10. Conclusion

In this chapter, we presented an iterative receiver for data transmission over time variant channels. Such a receiver needs to perform the two tasks of channel estimation and data

detection. Moreover, since these two tasks can enhance each other, they were run iteratively. The focus in this chapter was on the channel estimation part, whose maximum likelihood estimate boils down to a forward backward Kalman filter. To run this filter we just need to construct an input/output equation that describes the operation of the channel and a dynamical model that describes the time evolution of the channel. In addition we need an input/output equation that can be used for data detection.

We demonstrated the receiver design for 3 OFDM systems: single user single antenna (SISO) OFDM, multi-access OFDM, and multiple antenna (MIMO) OFDM. Moreover the three implementations of the Kalman filter (Cyclic FB Kalman, Helix FB Kalman and Forward only Kalman) were presented along with their comparative performance. The simulation results demonstrated in this chapter indicate that Kalman filter based receivers perform quite well in wireless environment and thus are potential contenders for practical receivers.

11. References

T. Y. Al-Naffouri. An EM-based forward-backward Kalman filter for the estimation of time-variant channels in OFDM. *IEEE Trans. Signal Proc.*, 55(7):3924–3930, Jul. 2007.

T. Y. Al-Naffouri and A. Quadeer. A Forward-Backward Kalman Filter based STBC MIMO OFDM Receiver. accepted in EURASIP Journal Adv. Signal Proc., Nov. 2008.

T. Y. Al-Naffouri and M. S. Sohail. An EM based Frequency Domain Channel Estimation Algorithm for Multi-Access OFDM Systems. submitted to Elsavier Signal Proc. Journal, Jul. 2008.

T. Y. Al-Naffouri, D. Toumpakaris, A. Bahai, and A. Paulraj. An adaptive semi-blind algorithm for channel identification in OFDM. *Asilomar Conf. on Signals, Syst., and Computers*, 2:921–925, Nov. 2001.

T. Y. Al-Naffouri, A. Bahai, and A. Paulraj. Semi-blind channel identification and equalization in OFDM: an expectationmaximization approach. *IEEE Vehicular Tech. Conf.*, 1:13–17, Sep. 2002.

T. Y. Al-Naffouri, O. Awoniyi, O. Oteri, and A. Paulraj. Receiver design for MIMO-OFDM transmission over time variant channels. *IEEE GLOBECOM*, 4:2487–2492, Dec. 2004.

G. A. Al-Rawi, T. Y. Al-Naffouri, A. Bahai, and J. Cioffi. Exploiting error-control coding and cyclic-prefix in channel estimation for coded OFDM systems. *IEEE Commun. Lett.*, 7(8):388–390, Aug. 2003.

S. M. Alamouti. A simple transmit diversity technique for wireless communications. *IEEE J. Select. Areas Commun.*, 16: 1451–1458, Oct. 1998.

C. Aldana, E. de Carvalho, and J. M. Cioffi. Channel estimation for multicarrier multiple input single output systems using the EM algorithm. *IEEE Trans. Signal Proc.*, 51(12):3280–3292, Dec. 2003.

M. Biguesh and A. B. Gershman. MIMO channel estimation: optimal training and tradeoffs between estimation techniques. *IEEE Int. Conf. Commun.*, 5:2658–2662, Jun. 2004.

H. Bölcskei, R. W. Heath, and A. J. Paulraj. Blind channel identification and equalization in OFDM-based multi-antenna systems. *IEEE Trans. Signal Proc.*, 50(1):96–109, Jan. 2002.

M. Chang and Y. T. Su. Blind and semiblind detections of OFDM signals in fading channels. *IEEE Trans. Commun.*, 52(5): 744–754, May 2004.

C. Cozzo and B. L. Hughes. Joint channel estimation and data detection in space-time communications. *IEEE Trans. Commun.*, 51(8):1266–1270, Aug. 2003.

T. Cui and C. Tellambura. Joint data detection and channel estimation for OFDM systems. *IEEE Trans. Commun.*, 54(4): 670–679, Apr. 2006.

O. Edfors, M. Sandell, J. van de Beek, K. S. Wilson, and P. O. B¨orjesson. OFDM channel estimation by singular value decomposition. *IEEE Trans. Signal Proc.*, 46(7):931–939, Jul. 1998.

F. Gao and A. Nallanathan. Blind channel estimation for MIMO OFDM systems via nonredundant linear precoding. *IEEE Trans. Signal Proc.*, 55(2):784–789, Jan. 2007.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 2000.

I. Kang, M. P. Fitz, and S. B. Gelfand. Blind estimation of multipath channel parameters: a modal analysis approach. *IEEE Trans. Commun.*, 47(8):1140–1150, Aug. 1999.

T. Kim and I. Eo. Reliable blind channel estimation scheme based on cross-correlated cyclic prefix for OFDM system. *Int. Conf. Adv. Commun. Technol.*, 1, Feb. 2006.

W. Kunji, Z. Jianhua, L. Chaojun, and H. Chen. Semi-blind OFDM channel estimation using receiver diversity in the presence of virtual carriers. *Int. Conf. Commun. and Networking*, 1:1–4, Oct. 2006.

E. Larsson and P. Stoica. *Space-Time Block Coding for Wireless Communications*. Cambridge University Press, 2003.

Y. Li, L. J. Cimini, and N. R. Sollenberger. Robust channel estimation for OFDM systems with rapid dispersive fading channels. *IEEE Trans. Commun.*, 46(7):902–915, Jul. 1998.

Y. Li, C. N. Georghiades, and G. Huang. Iterative maximum-likelihood sequence estimation for space-time coded systems. *IEEE Trans. Commun.*, 49(6):948–951, Jun. 2001.

H. Minn and N. Al-Dhahir. Optimal training signals for MIMO OFDM channel estimation. *IEEE Trans. Wireless Commun.*, 5(5):1158–1168, May 2006.

B. Muquet, M. de Courville, G. B. Giannakis, Z. Wang, and P. Duhamel. Reduced complexity equalizers for zero-padded OFDM transmissions. *IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, 5:2973–2976, Jun. 2000.

M. C. Necker and G. L. Stuber. Totally blind channel estimation for OFDM on fast varying mobile radio channels. *IEEE Trans. Wireless Commun.*, 3(5):1514–1525, Sep. 2004.

R. V. Nee and R. Prasad. *OFDM for wireless multimedia communications*. MA: Artech house, Boston, 2000.

R. Negi and J. Cioffi. Pilot tone selection for channel estimation in a mobile OFDM system. *IEEE Trans. Consumer Electr.*, 44 (3):1122–1128, Aug. 1998.

A. P. Petropulu, R. Zhang, and R. Lin. Blind OFDM channel estimation through simple linear precoding. *IEEE Trans. Wireless Commun.*, 3(4):647–655, Mar. 2004.

A. H. Sayed. A framework for state-space estimation with uncertain models. *IEEE Trans. Auto. Control.*, 46(7):998–1013, Jul. 2001.

Z. Shengli and G. B. Giannakis. Finite-alphabet based channel estimation for OFDM and related multicarrier systems. *IEEE Trans. Commun.*, 49(8):1402–1414, Aug. 2001.

C. Shin, R. W. Heath, and E. J. Powers. Blind channel estimation for MIMO-OFDM systems. *IEEE Trans. Vehicular Tech.*, 56 (2):670–685, Mar. 2007.

G. Stuber, J. Barry, S. Mclaughlin, Y. Li, M. Ingram, and T. Pratt. Broadband MIMO-OFDM wireless communications. *Proc. IEEE*, 92(2):271–294, Feb. 2004.

B. Yang, K. Ben Letaief, R. Cheng, and Z. Cao. Channel estimation for OFDM transmission in multipath fading channels based on parametric channel modeling. *IEEE Trans. Commun.*, 49(3):467–479, Mar. 2001.

F. Yang andW. Ser. Adaptive semi-blind channel estimation for OFDM systems. *IEEE Veh. Technol. Conf.*, 3:1773–1776, May 2004.

R. Zhang and W. Chen. A mixture Kalman filter approach for blind OFDM channel estimation. *Asilomar Conf. on Signals, Syst., and Computers*, 1:350–354, Nov. 2004.

Robust Kalman-type Filtering in Positioning Applications

Tommi Perälä
Tampere University of Technology
Finland

1. Introduction

Positioning refers to the estimation of one's location by combining many different sources of information. This information is usually obtained in the form of measurements, which may be, for example, pseudorange or deltarange measurements from satellites. In addition, various wireless networks on Earth, for example, cellular networks, WLAN or Bluetooth provide means for positioning in the form of range measurements, received signal strength indicators and sector information. Portable positioning devices may also contain inertial measurement units that provide information about the movements of the user.

The positioning problem may be formulated as a Bayesian filtering problem. The measurements are related to the position of the user, and the relation is approximately known. In addition, there is a model that describes the process dynamics. The measurements are obtained at discrete time intervals, and the process dynamics are also discretized.

The system may be described mathematically as follows. Let x_k denote the stochastic state vector at time step k, and let y_k be the measurement vector. The system is governed by the following equations:

$$y_k = h_k(x_k) + v_k \tag{1}$$

$$x_{k+1} = g_k(x_k) + w_k \tag{2}$$

Functions h_k and g_k are the measurement and state update functions, respectively. The measurement noise v_k and the state update noise w_k are assumed to be white processes. The initial state is denoted by x_0. The noises and the initial state are assumed to be mutually independent.

Using these assumptions, we want to find the conditional probability density function conditioned on all realized measurements $p_{x_k^+}(x_k|y_{1k})$, which is also called the posterior probability density function. The set of all realized measurements up to time step k is denoted by $y_{1k} = \{y_i | i = 1, 2, \ldots, k\}$. The posterior density function contains all the information of the system up to time step k. From this density function we can compute an estimate of the state with respect to any optimality criterion.

Using the above assumptions and the Bayes' rule, the posterior density function may be written as

$$p_{x_k^+}(x_k|y_{1:k}) = \frac{p_{y_k}(y_k|x_k)p_{x_k^-}(x_k|y_{1:k-1})}{p_{y_k^-}(y_k|y_{1:k-1})} ,$$

(3)

where we have denoted the set of past measurements by $y_{1:k-1}$ (Ristic et al., 2004). The conditional probability density function $p_{x_k^-}(x_k|y_{1:k-1})$ is called the prior density function, and it contains all the information about the system before using the measurements of the current time step. The measurement likelihood function $p_{y_k}(y_k|x_k)$ is given by the measurement model, and it is used to incorporate the measurements of the current time step to the estimate. The normalization factor

$$p_{y_k^-}(y_k|y_{1:k-1}) = \int p_{y_k}(y_k|x_k)p_{x_k^-}(x_k|y_{1:k-1})dx_k$$

(4)

is sometimes called the predicted measurement density or the innovation density. The expression (3) for the posterior density function allows the recursive computation of the conditional probability density function, which is very convenient regarding the computational and memory requirements of the algorithm.

In general, it is difficult to find an analytical expression for the posterior density function. However, if the state update function $g_k(x_k) = G_k x_k$ and the measurement function $h_k(x_k) = H_k x_k$ are linear, and the state model noise and the measurement noise are modeled as zero mean Gaussians with covariance matrices Q_k and R_k, respectively, and the initial state is Gaussian with mean x_0^+ and covariance matrix P_0^+, the posterior density function is also Gaussian. Only little computation is required in order to compute the posterior density function, and its parameters are given by the famous Kalman Filter relations $(5)-(9)$.

The prior mean is obtained by applying the linear state update function to the posterior mean of the previous time step:

$$x_k^- = E(x_k|y_{1:k-1}) = G_{k-1}x_{k-1}^+ .$$

(5)

The prior covariance matrix is obtained from the posterior covariance matrix of the previous time step:

$$P_k^- = V(x_k|y_{1:k-1}) = G_{k-1}P_{k-1}^+ G_{k-1}^T + Q_{k-1} .$$

(6)

The posterior mean is obtained by adding a linear transformation of the innovation to the prior mean:

$$x_k^+ = x_k^- + K_k(y_k - H_k x_k^-) .$$

(7)

The posterior covariance becomes:

$$P_k^+ = (I - K_k H_k)P_k^- ,$$

(8)

and the Kalman gain is defined as:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{9}$$

However, in practice the relation between the state and the measurements is rarely linear. Therefore, non-linear extensions of the Kalman Filter have been studied, and in this chapter we will concentrate on one of them, namely, the Extended Kalman Filter (EKF).

EKF linearizes the non-linear measurement and state update functions at the prior mean of the current time step and the posterior mean of the previous time step, respectively. The resulting algorithm is very similar to the Kalman Filter. However, EKF does not solve the posterior density function exactly, but instead, approximates the posterior density function with a Gaussian density function.

EKF has been studied in positioning applications and it is shown to perform poorly when the non-linearities are significant. (Ali-Löytty et al., 2005) show that when using measurements from satellites, the non-linearities do not degrade the performance of EKF, but when using range measurements from terrestrial base stations, EKF may easily veer away from the true solution and get stuck in a wrong solution branch. One reason for this kind of behavior is that the true posterior density might be multimodal, and EKF cannot know which peak represents the correct position of the user. The problem of multimodality has been addressed using Gaussian Mixture Filters (GMF), and they have been shown to perform quite well (Ali-Löytty & Sirola, 2007a); (Ali-Löytty & Sirola, 2007b);(Ali-Löytty, 2008). GMFs approximate the posterior density as a sum of Gaussian densities where each component is an individual EKF. By using GMFs, the problems caused by the non-linearities may be overcome.

Although GMFs perform quite well even in highly non-linear cases, they are still based on the assumption of Gaussian measurement noise. It has been shown that filters based on the assumption of Gaussian noise may perform poorly in cases where the measurement noise is non-Gaussian, and so-called blunder measurements occur (Perälä & Piché, 2007). In positioning applications, blunder measurements occur, for example, due to signal reflections and multipath effects.

In this chapter, we present two methods for making EKF more robust against blunder measurements. The robust modifications of EKF may also be incorporated in Gaussian Mixture Filters that are based on EKF. In the first method, the measurement covariance matrix is modified based on the differences between the predicted and realized measurements, which are also called innovations. The modification is done using weight functions that are derived from M-estimators. In the second method, the predicted measurement density is approximated with a non-Gaussian density and the likelihood score of the corresponding density is used instead of the Gaussian likelihood score that appears in the Kalman Filter. Using these modifications, we try to obtain filters that are robust against blunder measurements.

2. Extended Kalman filter

In this section we present the algorithm of the Extended Kalman Filter and introduce some terminology needed in the following sections. We assume that the state model is linear and only concentrate on linearizing the measurement model. Consider the non-linear measurement equation:

$$y_k = h_k(x_k) + v_k. \tag{10}$$

The first order Taylor series approximation of the measurement function at the prior mean is

$$h_k(x_k) \approx h_k(x_k^-) + H_k(x_k - x_k^-), \tag{11}$$

where the Jacobian of the measurement function is

$$H_k = \frac{\partial h_k(x_k)}{\partial x_k}\bigg|_{x_k = x_k^-}. \tag{12}$$

Denoting

$$\Delta y_k = y_k - h_k(x_k^-) \tag{13}$$

and

$$\Delta x_k = x_k - x_k^-, \tag{14}$$

an approximate measurement equation may be written as

$$\Delta y_k = H_k \Delta x_k + v_k. \tag{15}$$

Applying the Kalman Filter to this linearized measurement model, the posterior mean becomes

$$x_k^+ = x_k^- + K_k(y_k - h_k(x_k^-)) \tag{16}$$

and the posterior covariance is given by

$$P_k^+ = (I - K_k H_k)P_k^- \tag{17}$$

where $K_k = P_k^- H_k (H_k^T P_k^- H_k + R_k)^{-1}$ is the Kalman gain matrix. The only differences to the Kalman Filter are that the innovation is computed using the non-linear measurement function, and that H_k is the Jacobian of the measurement function.

The innovation $s_k := y_k - h_k(x_k^-)$, which appears in the posterior mean recursion, describes how much the measurements differ from those expected when we think the user's state is the prior mean x_k^-, which before taking the measurement into account is our best estimate for the state. In EKF, the state is corrected by applying a linear gain to the innovation. The robust filters presented in this chapter are based on a gain that is computed differently. The next section discusses the score functions that are used later to compute the gain.

3. Score function selection

The robust Kalman Filters discussed in this chapter are essentially based on embedding the score function of a robust M-estimator into the Kalman Filter. We use Huber's concept of minimax robustness to find the robust M-estimators. (Huber, 1964) suggests the minimization of the maximum asymptotic variance of an estimator over a predefined class of densities F. The solutions to this problem are pairs (T^0, f^0), where T^0 is the most robust M-estimator and f^0 is called the least favorable density of the class F. We introduce two

classes for densities for which the minimax solution has been found, namely the ε-contaminated normal neighborhood and the p-point family.

3.1 Huber M-estimator

The ε-contaminated normal neighborhood was first proposed in (Huber, 1964) to be used in robust parameter estimation and it is defined as follows.

Definition 1 (ε-contaminated Normal Neighborhood) The set of density functions F_ε is called ε-contaminated normal neighborhood if $F_\varepsilon = \{(1 - \varepsilon)\phi(x) + \varepsilon H(x) : H \in S\}$, where ϕ is the standard normal probability density function, S is the set of symmetrical probability density functions, and $0 \le \varepsilon < 1$ is the known fraction of contamination.

Huber showed that the least favorable density of this class is Gaussian in the middle, but has exponential tails. We denote this density by f_ε^0 and it is given by

$$
f_\varepsilon^0(t) = \begin{cases} \dfrac{(1-\varepsilon)}{\sqrt{2\pi}}\exp\left(-\tfrac{1}{2}t^2\right), & |t| \le k \\[2mm] \dfrac{(1-\varepsilon)}{\sqrt{2\pi}}\exp\left(\tfrac{1}{2}k^2 - k|t|\right), & |t| > k \end{cases}.
\tag{18}
$$

The connection between the threshold parameter k and the amount of contamination ε is given by

$$
\frac{2\phi(k)}{k} - 2\Phi(-k) = \frac{\varepsilon}{1-\varepsilon},
\tag{19}
$$

where Φ is the standard normal cumulative distribution function. Usually this equation has to be solved numerically. The influence function of an M-estimator is defined as the negative likelihood score of the least favorable density $\psi^0(t) = -\partial \ln f^0(t)/\partial t$, and for the Huber's M-estimator it is

$$
\psi_\varepsilon^0(t) = \begin{cases} t, & |t| \le k \\ k \cdot \text{sign}(t), & |t| > k \end{cases}.
\tag{20}
$$

The weight function of an M-estimator is defined as $\omega(t) = \psi(t)/t$, $t \neq 0$ and $\omega(0)$ is chosen so that $\omega(t)$ is continuous. For the Huber's M-estimator the weight function is

$$
\omega_\varepsilon^0(t) = \begin{cases} 1, & |t| \le k \\ \dfrac{k}{|t|}, & |t| > k \end{cases}
\tag{21}
$$

The influence function and the weight function are needed in the robust filters that are presented in the following sections.

3.2 p-point M-estimator

Another interesting family of densities, namely, the p-point family is used in robust parameter estimation in (Martin & Masreliez, 1975); (Masreliez & Martin, 1977), and it is defined as follows.

Definition 2 (p-point Family) The set of probability density functions F_p is called a p-point family if $F_p = \left\{ f \mid \int_{-\infty}^{-y_p} f(x)dx = p/2 = \Phi(-y_p), f \text{ symmetric and continuous at } \pm y_p \right\}$.

The inclusion of the restriction that F_p contains the standard normal cumulative distribution function $\Phi(x)$ is for standardization purposes, that is, to ensure that F_p is in the neighborhood of the standard normal density.

(Masreliez & Martin, 1977) show that the least favorable density f_p^0 of F_p is

$$f_p^0(t) = \begin{cases} K\cos^2\left(\dfrac{t}{2c_m y_p}\right), & |t| \le y_p \\[4mm] K\cos^2\left(\dfrac{1}{2c_m}\right)\cdot\exp\left(\dfrac{2K}{p}\cos^2\left(\dfrac{1}{2c_m}\right)(y_p - |t|)\right), & |t| > y_p \end{cases} \tag{22}$$

where K is related to p by the following equation

$$K = \frac{1-p}{y_p(1 + c_m \sin(1/c_m))} . \tag{23}$$

For each p there exists s_m that minimizes the asymptotic variance of the estimator. The minimizing value of c_m satisfies the equation

$$2c_m - p\left(1 + \tan^2(1/2c_m)\right)\left(2c_m + \tan(1/2c_m)\right) = 0 . \tag{24}$$

The influence function of the least favorable density of the p-point family F_p is

$$\psi_p^0(t) = -\frac{\partial}{\partial t}\ln f_e^0(t) = \begin{cases} \dfrac{1}{c_m y_p}\tan\left(\dfrac{t}{2c_m y_p}\right), & |t| \le y_p \\[4mm] \dfrac{1}{c_m y_p}\tan\left(\dfrac{1}{2c_m}\right)\cdot\text{sign}(t), & |t| > y_p \end{cases} \tag{25}$$

and the weight function is

$$\omega_p^0(t) = \begin{cases} \dfrac{1}{c_m y_p t}\tan\left(\dfrac{t}{2c_m y_p}\right), & |t| \le y_p \\[4mm] \dfrac{1}{c_m y_p |t|}\tan\left(\dfrac{1}{2c_m}\right), & |t| > y_p \end{cases} . \tag{26}$$

3.3 Damped Hampel M-estimator

(Hampel et al., 1981) propose other M-estimators for robust estimation. However, these M-estimators belong to the class of redescending M-estimators, i.e., they have finite rejection point. M-estimators with finite rejection points discard certain measurements that are assumed to be too far from the true parameter. In hybrid positioning, we cannot always afford to discard measurements, and thus we will present here a modified version of the Hampel's three parts redescending M-estimator, namely the Damped Hampel M-estimator. The influence function of the Damped Hampel M-estimator is defined as

$$\psi_{\mathrm{DHA}}(t) = \begin{cases} t, & |t| \le k_1 \\ k_1 \cdot \mathrm{sign}(t), & k_1 < |t| \le k_2 \\ \dfrac{k_1 k_2^{\mathrm{r}}}{|t|^{\mathrm{r}}} \cdot \mathrm{sign}(t), & |t| > k_2 \end{cases} \tag{27}$$

,

where we assume that $r \ge 0$. The corresponding weight function is defined as

$$\omega_{\mathrm{DHA}}(t) = \begin{cases} 1, & |t| \le k_1 \\ \dfrac{k_1}{|t|}, & k_1 < |t| \le k_2 \\ \dfrac{k_1 k_2^{\mathrm{r}}}{|t|^{\mathrm{r}+1}}, & |t| > k \end{cases} \tag{28}$$

.

By setting $k_1 = k_2 = k$ and $r = 0$, we see that the Huber's M-estimator is obtained as a special case of the Damped Hampel M-estimator.

The Damped Hampel M-estimator is not derived from a least favorable density function, but instead, it is obtained by designing a piecewise weighting for the observations in a continuous manner. It is possible to calculate the density function by using the definition of the influence function, but for our purposes it is enough to know the influence and weight functions.

The Damped Hampel M-estimator presented here is only one example of piecewise influence functions. It is easy to design a variety of different piecewise influence functions and try to find the best one for the problem at hand using optimization techniques. Although the Huber's M-estimator and the p-point M-estimator are most robust in minimax sense, they require some knowledge of the distribution of the errors. Usually we do not have such knowledge, and thus any M-estimator that has proven to perform well in testing could be used.

4. Re-weighted extended Kalman filter

In the Kalman Filter, the posterior density function is Gaussian, and thus, the posterior mean estimate is the value that maximizes the probability density function:

$$x_k^+ = \arg \max_{x_k} p_{x_k^+}(x_k | y_{1:k}) . \tag{29}$$

Using the Bayes' theorem, inserting the prior density function and the measurement likelihood function, and noting that constant multipliers do not affect the maximization problem, (29) may be written as

$$x_k^+ = \arg \max_{x_k} \left(\exp\left(-\tfrac{1}{2} \left\| x_k - x_k^- \right\|_{(P_k^-)^{-1}}^2 \right) \cdot \exp\left(-\tfrac{1}{2} \left\| y_k - H_k x_k \right\|_{(R_k)^{-1}}^2 \right) \right) . \tag{30}$$

Since the exponential function is monotonically increasing, an equivalent problem is the following minimization problem.

$$x_k^+ = \arg\min_{x_k} \left(\left\| x_k - x_k^- \right\|_{(P_k^-)^{-1}}^2 + \left\| y_k - H_k x_k \right\|_{(R_k)^{-1}}^2 \right). \tag{31}$$

Leaving the subscripts out for simplicity, and denoting $n = (P^-)^{-\frac{1}{2}}(x - x^-)$ and $l = R^{-\frac{1}{2}}(y - Hx)$, where the square root is the symmetric square root of a matrix, (31) may be written as

$$x_k^+ = \arg\min_{x_k} \left(\sum_{i=1}^{n_x} n_i^2 + \sum_{j=1}^{n_y} l_j^2 \right). \tag{32}$$

As may be seen in equation (32), the posterior mean estimate of the Kalman Filter is a recursive solution to an ordinary least squares problem. The idea of minimizing the sum of squared errors is tempting since the solution may be computed efficiently. However, the least squares method is not robust, and therefore we modify the quadratic cost function in the second sum by a convex function ρ. The ρ-function is chosen so that the derivative of the score function is the influence function of an M-estimator introduced in Section 3. The minimization problem is changed to

$$x_k^+ = \arg\min_{x_k} \left(\sum_{i=1}^{n_x} n_i^2 + \sum_{j=1}^{n_y} \rho(l_j) \right). \tag{33}$$

The aim is to make the measurement model more robust, and therefore, only the second sum is modified. Since the score function is assumed to be convex, the minimum is found by setting the gradient of the sum to zero

$$\nabla_x \left(\sum_{i=1}^{n_x} n_i^2 + \sum_{j=1}^{n_y} \rho(l_j) \right) = 0. \tag{34}$$

Denoting the derivative of the score function ρ by ψ, equation (34) may be written as

$$\sum_{i=1}^{n_x} 2n_i (P^-)^{-\frac{1}{2}} e_i + \sum_{j=1}^{n_y} \psi_j(l_j) H^T R^{-\frac{1}{2}} e_j = 0, \tag{35}$$

where e_j is a vector whose jth element is one and the others are zeros. Thus, the minimization problem is equal to a vector equation

$$(P^-)^{-\frac{1}{2}} \left[n_1, \cdots, n_{n_x} \right]^T + H^T R^{-\frac{1}{2}} \left[\psi_1(l_1), \ldots, \psi_{n_y}(l_{n_y}) \right]^T = 0. \tag{36}$$

Since ψ is, in general, a non-linear function, equation (36) has to be solved numerically. However, we want to preserve the computationally convenient properties of KF, and thus we proceed as in (Durovic & Kovacevic, 1999) and (Carosio et. al, 2005), and replace the equation by a linear approximation. Thus, we write

$$(P^-)^{-\frac{1}{2}} n + H^T R^{-\frac{1}{2}} \left[\omega_1(l_1^-) l_1, \ldots, \omega_{n_y}(l_{n_y}^-) l_{n_y} \right]^T = 0. \tag{37}$$

where the weights ω_i are given by

$$\omega_i = \begin{cases} \dfrac{\psi_i(l_i^-)}{l_i^-} & l_i^- \neq 0 \\ 1 & l_i^- = 0 \end{cases} \tag{38}$$

and

$$l^- = R^{-\frac{1}{2}}(y - Hx^-) . \tag{39}$$

Now equation (37) may be written as

$$(P^-)^{-\frac{1}{2}}n + H^T R^{-\frac{1}{2}} W_\psi l = 0 , \tag{40}$$

where W_ψ is a diagonal matrix with diagonal elements $\omega_1(l_1^-),\ldots,\omega_{n_y}(l_{n_y}^-)$. Inserting $n = (P^-)^{-\frac{1}{2}}(x - x^-)$ and $l = R^{-\frac{1}{2}}(y - Hx)$ yields

$$(P^-)^{-1}(x - x^-) + H^T R^{-\frac{1}{2}} W_\psi R^{-\frac{1}{2}}(y - Hx) = 0 . \tag{41}$$

We define the re-weighted measurement covariance matrix as

$$R_W = (R^{-\frac{1}{2}} W_\psi R^{-\frac{1}{2}})^{-1} . \tag{42}$$

This matrix exists assuming that W_ψ is positive definite, which is true if $\omega_i > 0, \forall i$. Inserting (42) into (41) yields

$$(P^-)^{-1}(x - x^-) + H^T R_W^{-1}(y - Hx) = 0 , \tag{43}$$

which is the solution for the minimization problem

$$x^+ = \arg\min_x \left(\|x - x^-\|_{(P^-)^{-1}}^2 + \|y - Hx\|_{(R_W)^{-1}}^2 \right) . \tag{44}$$

Equation (44) is similar to equation (31), which was derived from the posterior mean estimate of KF. The only difference is that the measurement covariance matrix R is replaced by the weighted measurement covariance matrix R_W. Thus, the solution of (44) is obtained using the posterior mean relation of KF. The measurement update recursions for the posterior mean estimate may be then written as:

$$x^+ = x^- + K_W(y - Hx^-) , \tag{45}$$

Where

$$K_W = P^- H^T (HP^- H^T + R_W)^{-1} , \tag{46}$$

and the posterior covariance estimate becomes

$$P^+ = (I - K_W H)P^- . \tag{47}$$

The Re-weighted Kalman Filter derived in this section consists of computing the transformed innovation l^- and weighting the measurement covariance matrix using the

weight function of an M-estimator. The prior covariance is left intact since only the measurement model is modified. Thus, the filter derived here may be considered as a robust Kalman Filter, which modifies the given measurement covariances according to the innovations so that bigger transformed innovations result in bigger variances. A drawback of this method is that if the prior mean estimate x^- is far away from the true state, uncorrupted measurements might get weighted down, which could result in bad filter performance.

The filter derived here is for linear systems only. Thus we want to extend the filter for non-linear problems using the ideas of EKF. The only difference is that the innovations are computed using the non-linear measurement function, and that H is the Jacobian of the measurement function computed at the prior mean. The resulting filter is called the Re-weighted Extended Kalman Filter (REKF).

5. Approximate Bayesian extended Kalman filter

In the previous section KF was interpreted as a recursive least-squares algorithm, and was "robustified" by replacing the quadratic cost with (33). In this section we present an alternative approach that is directly based on the Bayesian interpretation of KF presented in Section 1. Consider a linear transformation matrix

$$T_k = (H_k P_k^- H_k^T + R_k)^{-\frac{1}{2}}, \tag{48}$$

where H_k, P_k^- and R_k are the linear measurement function, the prior covariance matrix and the measurement covariance matrix, respectively, that appear in KF. The inverse exists and is symmetric since $H_k P_k^- H_k^T + R_k$ is symmetric and positive definite.

Denote the innovation as $s_k := y_k - H_k x_k^-$. The mean of the innovation is $E(s_k) = 0$ and the covariance matrix is $V(s_k) = H_k P_k^- H_k^T + R_k$. Thus, the mean of the transformed innovation $r_k = T_k s_k$ is $E(r_k) = 0$ and the covariance matrix $V(r_k) = I$. Now consider the posterior mean estimate of KF

$$x_k^+ = \int x_k p_{x_k^+}(x_k | y_{1:k}) dx_k . \tag{49}$$

Using the Bayes' rule this may be written as

$$x_k^+ = \int x_k \frac{p_{x_k^-}(x_k | y_{1:k-1}) p_{y_k}(y_k | x_k)}{p_{y_k^-}(y_k | y_{1:k-1})} dx_k , \tag{50}$$

which may be written after some algebra as

$$x_k^+ = x_k^- - \frac{1}{p_{y_k^-}(y_k | y_{1:k-1})} P_k^- \int p_{y_k}(y_k | x_k)(-(P_k^-)^{-1}(x_k - x_k^-)) p_{x_k^-}(x_k | y_{1:k-1}) dx_k . \tag{51}$$

If the posterior density is approximated at every time step with a Gaussian, the prior density is also Gaussian since the state update function is linear and the noise in the state equation is Gaussian and independent of the state. By noting that for Gaussian prior it holds that

$$\nabla_{x_k} p_{x_k^-}(x_k | y_{1:k-1}) = -(P_k^-)^{-1}(x_k - x_k^-) p_{x_k^-}(x_k | y_{1:k-1}) , \tag{52}$$

the posterior mean may be written as

$$x_k^+ = x_k^- - \frac{1}{p_{y_k^-}(y_k|y_{1:k-1})} P_k^- \int p_{y_k}(y_k|x_k) \nabla_{x_k} p_{x_k^-}(x_k|y_{1:k-1}) dx_k .$$
(53)

Noting that it follows from the measurement model that $p_{y_k}(y_k|x_k) = p_{v_k}(y_k - H_k x_k)$, and integrating by parts yields

$$x_k^+ = x_k^- + \frac{1}{p_{y_k^-}(y_k|y_{1:k-1})} P_k^- \int p_{x_k^-}(x_k|y_{1:k-1}) \nabla_{x_k} p_{v_k}(y_k - H_k x_k) dx_k .$$
(54)

Because

$$\nabla_{x_k} p_{v_k}(y_k - H_k x_k) = -H_k^T \nabla_{y_k} p_{v_k}(y_k - H_k x_k) ,$$
(55)

the posterior mean may be written as

$$x_k^+ = x_k^- - \frac{1}{p_{y_k^-}(y_k|y_{1:k-1})} P_k^- \int p_{x_k^-}(x_k|y_{1:k-1}) H_k^T \nabla_{y_k} p_{v_k}(y_k - H_k x_k) dx_k .$$
(56)

Changing the order of differentiation and integration yields

$$x_k^+ = x_k^- + P_k^- H_k^T (-\nabla_{y_k} \ln p_{y_k^-}(y_k|y_{1:k-1})) .$$
(57)

Noting that $y_k = T_k^{-1} r_k + H_k x_k^-$ it can be shown that

$$x_k^+ = x_k^- + P_k^- H_k^T T_k (-\nabla_{r_k} \ln p_{r_k^-}(r_k|y_{1:k-1})) .$$
(58)

Define the influence function $\psi(r_k)$ as the negative likelihood score of the transformed innovation density

$$\psi(r_k) = -\nabla_{r_k} \ln p_{r_k^-}(r_k|y_{1:k-1}) ,$$
(59)

and insert into (54) to obtain

$$x_k^+ = x_k^- + P_k^- H_k^T T_k \psi(r_k) .$$
(60)

It is easy to see by straightforward calculation that a Gaussian innovation density, which results from the assumptions of KF, produces the familiar posterior mean update relation of KF. (Masreliez & Martin, 1975) study this kind of estimators and show that if the marginal densities of $p_{r_k^-}(r_k|y_{1:k-1})$ are symmetric densities in F, where $F = F_\varepsilon$ or $F = F_p$, the estimator covariance is bounded by

$$C_k \le (I - K_k H_k E_F) P_k^- ,$$
(61)

where $E_F = \int (\frac{\partial}{\partial t} \psi(t)) f_F^0(t) dt$, f_F^0 is the least favorable density in F, and K_k is the Kalman gain matrix. It can be shown that it is possible to come arbitrarily close to the bound (Masreliez & Martin, 1975). Thus, the upper bound is chosen as the posterior covariance estimate P_k^+ for ABKF.

For the Huber's M-estimator and the p-point M-estimator E_F is found easily by straightforward calculation. For the Huber's M-estimator it is

$$E_{F_\varepsilon} = (1 - \varepsilon)(1 - 2\Phi(-k)) , \qquad (62)$$

and for the p-point M-estimator we get

$$E_{F_p} = (c_m y_p)^{-1} \left(1 - p \left(1 + \tan^2 \left((c_m y_p)^{-1} \right) \right) \right) . \qquad (63)$$

For the Damped Hampel M-estimator the integral does not generally have an explicit form, but since the central part of the influence function is the same as in the Huber's M-estimator, we use E_{F_ε} to compute the posterior covariance estimates when using the Damped Hampel M-estimator.

The robust posterior estimates may be computed using (60) and (61), however this does not produce the posterior probability density function. However, we approximate it with a Gaussian density with the mean and covariance matrix as in (60) and (61). This is called the Approximate Bayesian Kalman Filter (ABKF).

In the case where the measurement function is non-linear, the above considerations can be applied in EKF. The only difference is that the innovations are computed using the non-linear measurement function, and that H_k is the Jacobian of the measurement function computed at the prior mean. The resulting non-linear extension is called the Approximate Bayesian Extended Kalman Filter (ABEKF).

6. Positioning example

We consider a positioning scenario where we use satellite pseudorange and deltarange measurements that are obtained at discrete time intervals. The state of the user consists of the 3-dimensional position and velocity vectors of the user and is denoted by $x = [(r^u)^T \ (v^u)^T]^T$. The pseudorange measurements may be written as

$$z_i^p = \left\| r_i^s - r^u \right\| + b^p + \varepsilon_i^p , \qquad (64)$$

where the ith pseudorange measurement is denoted by z_i^p, r^u is the position of the user, r_i^s is the position of the ith satellite, b^p is the unknown clock bias and ε_i^p is Gaussian zero mean noise with variance $(\sigma_i^p)^2$. The deltarange measurements may be written as

$$z_i^d = \frac{(r_i^s - r^u)^T}{\left\| r_i^s - r^u \right\|} (v_i^s - v^u) + b^d + \varepsilon_i^d , \qquad (65)$$

where the ith deltarange measurement is denoted by z_i^d, v^u is the velocity of the user, v_i^s is the velocity of the ith satellite, b^d is the unknown clock drift and ε_i^d is Gaussian zero mean noise with variance $(\sigma_i^d)^2$.

The positions and velocities of the satellites are assumed to be known, but clock bias and the clock drift that appear in (58) and (59) are unknown. They are, however, the same for all the satellites and thus we may deal with them by using so-called difference measurements. Therefore, we have to introduce the concept of difference mapping.

Definition 3 (Difference Mapping) A difference mapping D is a $(n_s - 1) \times n_s$-matrix with full column rank such that $D1 = 0$, where 1 is a vector of ones.

The difference mapping may be chosen to be, for example, $D = \begin{bmatrix} I & -1 \end{bmatrix}$. Denote

$$D_1 = \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix}. \tag{66}$$

Thus, the measurement vector becomes

$$y = D_1 \begin{bmatrix} \left\| r_1^s - r^u \right\| \\ \vdots \\ \left\| r_{n_s}^s - r^u \right\| \\ \dfrac{(r_1^s - r^u)^T}{\left\| r_1^s - r^u \right\|}(v_1^s - v^u) \\ \vdots \\ \dfrac{(r_{n_s}^s - r^u)^T}{\left\| r_{n_s}^s - r^u \right\|}(v_{n_s}^s - v^u) \end{bmatrix} + D_1 \begin{bmatrix} b^p \\ \vdots \\ b^p \\ b^d \\ \vdots \\ b^d \end{bmatrix} + D_1 \begin{bmatrix} \varepsilon_1^p \\ \vdots \\ \varepsilon_{n_s}^p \\ \varepsilon_1^d \\ \vdots \\ \varepsilon_{n_s}^d \end{bmatrix} = D_1 \begin{bmatrix} \left\| r_1^s - r^u \right\| \\ \vdots \\ \left\| r_{n_s}^s - r^u \right\| \\ \dfrac{(r_1^s - r^u)^T}{\left\| r_1^s - r^u \right\|}(v_1^s - v^u) \\ \vdots \\ \dfrac{(r_{n_s}^s - r^u)^T}{\left\| r_{n_s}^s - r^u \right\|}(v_{n_s}^s - v^u) \end{bmatrix} + D_1 \begin{bmatrix} \varepsilon_1^p \\ \vdots \\ \varepsilon_{n_s}^p \\ \varepsilon_1^d \\ \vdots \\ \varepsilon_{n_s}^d \end{bmatrix}. \tag{65}$$

The Jacobian of the measurement function is

$$H = \frac{\partial h(x)}{\partial x} \bigg|_{x = x^-} = D_1 \begin{bmatrix} -U & 0 \\ U_{v^s}^D - U_{v_u}^D & -U \end{bmatrix}, \tag{66}$$

where $U_{v^s}^D$ is the derivative of the vector $\text{diag}(UV^T)$ and V is a matrix of satellite velocities, $U_{v_u}^D$ is the derivative of the vector Uv_u, and U is a matrix whose rows are the unit vectors pointing from the user to the satellites. The derivation of the Jacobian is straightforward but tedious and is omitted here.

The measurement covariance matrix becomes

$$R = D_1 \begin{bmatrix} (\sigma_1^p)^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & (\sigma_{n_s}^p)^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & (\sigma_1^d)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & (\sigma_{n_s}^d)^2 \end{bmatrix} D_1^T, \tag{67}$$

7. Simulations and testing

The robust filters were implemented in MATLAB and tested in simulations and using real GPS data. In Section 7.1, the simulation setup is described and the results of the simulations are discussed. The tests using real GPS measurement data are discussed in Section 7.2.

7.1 Simulations

The simulation test bench was designed to produce dynamic test data similar to what could be expected in real world personal positioning scenarios. The main difference to real data is that in the simulations the true track and correct measurement and motion models are known.

The test process consisted of first generating a true track of 120 points with one second intervals using a velocity-restricted random walk model, where $\sigma_p^2 = (1.41 \frac{m}{s})^2 s^{-1}$ and $\sigma_a^2 = (0.316 \frac{m}{s}) s^{-1}$. Similar values have been used in (Ma, 2004) to model moving vehicles.

Next, a GPS constellation was simulated with an elevation mask and a shadowing profile that were set so that only a couple of satellites were visible at a time. Finally, noisy measurements were generated for each time step.

Satellite pseudorange and deltarange measurements were used with an average of 2.9 pseudorange and deltarange measurements per time step. Measurement variances were set to $(\sigma_i^p)^2 = ((0.1 + 2x_U)^2 \ m^2$ and $(\sigma_i^d)^2 = ((0.01 + 0.05x_U)^2 \ \frac{m^2}{s^2}$. The term x_U denotes the realization of a stochastic variable with standard uniform distribution $U(0,1)$. Altogether, 100 track and measurement sets were generated. These sets were generated using the Personal Navigation Filter Framework (Raitoharju et al., 2008).

Next, some additional noise was generated to the measurements according to different choices of the blunder probability p_b. For each measurement a sample from the standard uniform distribution $U(0,1)$ was drawn. If the realization of the sample was less than p_b, a realization of a sample from $U(-30\sigma, 30\sigma)$, where σ was the standard deviation of the corresponding measurement, was added to the measurement value. The blunder probabilities were chosen to be 0, 0.01, 0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.35 and 0.5.

The test tracks were filtered with the six robust filters described in this chapter, corresponding to two choices of the filter (ABEKF and REKF) and to three choices of the influence and weight functions (Huber (H), p-point (M) and Damped Hampel (DHA)). The parameters for the influence and weight functions used in the simulations and tests are presented in Table 1. Since DHA M-estimator is not derived from a minimax criterion, and thus does not correspond to any least favorable density presented in this chapter, using it in ABEKF is somewhat questionable. However, since DHA is essentially a generalization of the Huber M-estimator, we use E_F in DHA.

Name	Parameters
Huber	$k = 2.2$
p-point	$y_p = 2.2$
Damped Hampel	$k_1 = 2.2$, $k_2 = 3.3$, $r = 1$

Table 1. Estimator parameters used in simulations and testing

The mean and covariance of the posterior distribution were recorded at each time step and compared to the true track. For comparison, the data was also processed with EKF.

Figure 1 shows the mean error of different filters when using the Damped Hampel influence and weight functions. It can be seen that EKF works quite well also when the blunder probability gets bigger. This is not surprising since EKF should be optimal for linear measurements regardless of the density function of the error. Nonetheless, REKF still performs better than EKF. However, ABEKF starts to give meaningless estimates even with moderate blunder probabilities.
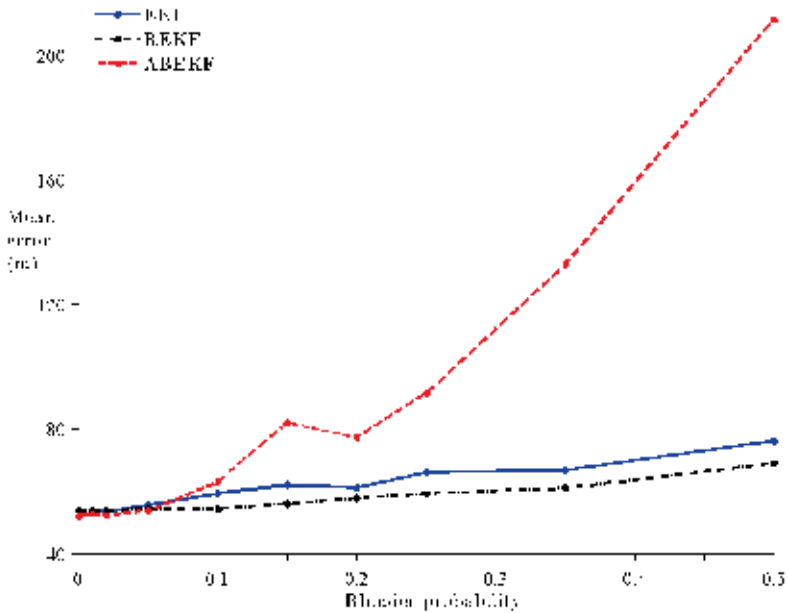
Fig. 1. Comparison of different filters in the simulations

The mean error (m), the 95% percentile of errors (m) and the frequency of inconsistent estimates (%) are presented in Table 2 for the blunder probabilities of 0, 0.1 and 0.25. The inconsistency was determined using the general inconsistency test with risk level 5% (Ali-Löytty et al., 2005), and it tells how often the error estimate was smaller than the actual error.

		p_b = 0%			p_b = 5%			p_b = 25%		
		ME (m)	95% (m)	Inc. (%)	ME (m)	95% (m)	Inc. (%)	ME (m)	95% (m)	Inc. (%)
	H	52.2	287	0.0	53.5	281	0.1	59.7	300	1.5
REKF	M	52.2	285	0.0	53.8	281	0.0	60.9	302	2.0
	DHA	53.6	286	0.0	54.3	279	0.1	59.1	285	1.1
	H	51.9	284	0.0	53.4	277	0.3	64.9	313	10
ABEKF	M	53.0	275	0.0	54.4	279	0.1	62.6	292	7.2
	DHA	51.9	284	0.0	53.5	276	0.5	91.4	461	22
EKF		51.8	284	0.0	55.4	284	0.4	66.0	313	14

Table 2. Results of the simulations

The filtered solutions were consistent when the blunder probability was small, but with large blunder probability ABEKF and EKF produced more inconsistent solutions. REKF, however, does not suffer from inconsistency even with high blunder probabilities. The mean error and the 95% percentile of errors did not vary much between different filters. However, when the blunder probability was large, the best performance was obtained using REKF. ABEKF also performed better than EKF with respect to any criterion with large blunder

probabilities when using the Huber's M-estimator or the p-point M-estimator. Damped Hampel M-estimator did not perform very well with ABEKF, but gave the best results when used in REKF. The fact that EKF performed quite well even with large blunder probabilities was surprising, but may be justified by noting that with almost linear measurements EKF should be optimal regardless of the noise distribution.

7.2 Tests using real GPS data

The filters were also tested using real GPS data. The test bench consists of 40 sets of measurements, which were recorded with a GPS receiver in Tampere, Finland. The receiver used was a Bluetooth Assisted GPS, BAG (Wirola et al., 2006). The sets consisted of cases where the user was standing still, walking or traveling in a bus with the receiver. The true track was only approximately known and it was transformed into digital format using a digital map of Tampere. Thus, the reporter errors are not exact, but instead, should be considered only as indicative.

The results of the tests using real GPS data are presented in Table 3. ABEKF does not seem to work very well with DHA influence function, whereas the influence function of the Huber M-estimator and the influence function of the p-point M-estimator seem to work well. However, ABEKF outperforms EKF with all choices of influence function. REKF outperforms ABEKF with respect to mean error and 95% percentile but ABEKF is more consistent.

The consistency of the solutions for each filter seems to be a lot worse than in the simulations. The inconsistency results from the fact that the variances of the measurements given by the measurement device are too optimistic. This might also be the cause of the poor performance of ABEKF when using the Damped Hampel M-estimator. Nevertheless, ABEKF and REKF perform better than EKF with respect to any criterion. By optimizing the parameters of these filters it might be possible to obtain even better results.

		ME (m)	95% (m)	Inc. (%)
	H	26	55	37.3
REKF	M	27	55	35.4
	DHA	24	55	36.9
	H	29	57	36.3
ABEKF	M	29	56	34.5
	DHA	1046	9308	36.1
EKF		27051	221527	44.7

Table 3. Results of the tests using real GPS data

An example of a real positioning scenario is presented in Figures 2 and 3. In Figure 2 the measurements were processed with EKF using the Damped Hampel weight function, and in Figure 3 the same measurements were processed with REKF. The true track is denoted by a black dashed line, and the mean estimates of the filtered solutions are denoted by blue and red dots. Blue dots represent consistent estimates and red dots are inconsistent estimates. This example shows how poorly EKF might work even in quite good signal conditions, and how significantly REKF improves the estimation process.

Fig. 2. An example of EKF using real GPS-measurements. (Map © Kaupunkimittaus Tampere 2008)

Fig. 3. An example of REKF using real GPS-measurements

8. Conclusions

In this chapter robust filtering techniques for positioning using satellite measurements were presented. The Extended Kalman Filter was chosen as basis for robust filter design. Six filters were presented and tested in the simulations and using real GPS data.

Based on the simulations the proposed filters seem to outperform EKF when blunder measurements occur, and do almost as well in normal cases. However, ABEKF does not seem to work well when using the Damped Hampel M-estimator, but performs better than EKF when using the Huber's M-estimator or the p-point M-estimator. The best performance was obtained using REKF with the Damped Hampel M-estimator, but other M-estimators also seemed to work almost as well.

The tests using real GPS data showed similar trends in the results except that EKF performed very poorly. REKF with the Damped Hampel M-estimator performed best also when using real GPS data.

Therefore, REKF should be taken into consideration when implementing positioning algorithms in mobile positioning devices due to its light computational and memory requirements and relatively high accuracy. The most promising approach was to use the Damped Hampel weight functions, but the performance degrades only a little when using any other influence and weight function proposed in this chapter.

9. References

Ali-Löytty, S.; Sirola, N. & Piché, R. (2005). Consistency of three Kalman Filter extensions inhybrid navigation, *Proceedings of the European Navigation Conference GNSS*, July 2005

Ali-Löytty, S.; Sirola, N. (2007a). Gaussian Mixture Filter in Hybrid Navigation, *Proceedingsof the European Navigation Conference GNSS*, Geneva, Switzerland, May 2007

Ali-Löytty, S.; Sirola, N. (2007b). Gaussian Mixture Filter and Hybrid Positioning,*Proceedings of ION GNSS 2007*, pp. 562—570, Forth Worth, Texas, September 2007

Ali-Löytty, S. (2008). Efficient Gaussian Mixture Filter for Hybrid Positioning, *Proceedings ofIEEE/ION Position Location and Navigation Symposium*, pp. 60—66, Monterey, California, May 2008

Carosio, A.; Cina, A.; Piras, M. (2005). The Robust Statistics Method Applied to the Kalman Filter: Theory and Application, *Proceedings of ION GNSS 2005*, September 2005

Durovic, Z.; Kovacevic, B. (1999). Robust Estimation with Unknown Noise Statistics, *IEEE Transactions on Automatic Control*, Vol. 44, (June 1999), pp. 1292--1296

Huber, P. (1964). Robust Estimation of Location Parameter, *Annals of Mathematical Statistics*, Vol. 35, (1964), pp. 73–101

Ma, C. (2003). Integration of GPS and Cellular Networks to Improve Wireless Location Performance, *Proceedings of ION GPS/GNSS 2003*, pp. 1585—1596, September 2003

Martin, R.; Masreliez, C. (1975). Robust Estimation via Stochastic Approximation, *IEEE Transactions of Information Theory*, Vol. IT-21, (May 1975), pp. 263-271

Masreliez, C.; Martin, R. (1977). Robust Bayesian Estimation for the Linear Model and Robustifying the Kalman Filter, *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 3, (1977), pp. 361—371

Perälä, T.; Piché, R. (2007). Robust Extended Kalman Filtering in Hybrid Positioning Applications, *Proceedings of the 4th Workshop on Positioning, Navigation and Communication (WPNC'07)*, pp. 55—64, Hannover, Germany, March 2007

Raitoharju, M.; Sirola, N.; Ali-Löytty, S.; Piché, R. (2008). PNaFF: A Modular Software Platform for Testing Hybrid Positioning Estimation Algorithms, *Proceedings of the 5th Workshop on Positioning, Navigation and Communication (WPNC'08)*, Hannover, Germany, March 2008

Ristic, B.; Arulampalam, S.; Gordon, N. (2004). *Beyond the Kalman Filter, Particle Filters for Tracking Applications*, Artech House, 1-58053-631-x, Boston, London

Wirola, L.; Alanen, K.; Käppi, J.; Syrjärinne, J. (2006). Bringing RTK to Cellular Terminals Using a Low-cost Single-frequency AGPS Receiver and Inertial Sensors, *Proceedings of 2006 IEEE/ION Position, Location, and Navigation Symposium*, 2006

Object Visual Tracking using Window-Matching Techniques and Kalman Filtering

Flávio B. Vidal and Victor H. Casanova Alcalde
Electrical Engineering Department, University of Brasilia
Brazil

1. Introduction

Object visual tracking aims to determine the image configuration of a target region of an object as it moves through a camera's field of view. The visual tracking process consists on matching the target region in successive frames of a sequence of images taken at closely-spaced intervals. Visual tracking has become an important process on various applications as: vision-based control (Hutchinson et al., 1996; Papanikolopoulos et al., 1992), industrial robotics (Sumi et al., 2007), biomedicine (Shen et al., 2006), surveillance (Urtasun et al., 2006), aerial target tracking (Yau et al., 2001), aircraft and car traffic monitoring and control (Rostamianfar et al., 2006).

Algorithms that combine digital image processing and visual servo control techniques are being applied to the solution of complex problems such as object tracking from a sequence of images (Hager et al., 1998). Visual tracking can be considered an estimation process acting together with digital image processing techniques. For the estimation process a stochastic filtering approach using Kalman filter can be applied (Veeraraghavan et al., 2006) and the particle filter (Shen et al., 2006). A visual tracking algorithm in (Babu et al., 2007) combines mean-shift tracker with a modified window-matching algorithm in order to avoid drift during partial object occlusion. Other algorithm (Brassnet et al., 2007) uses particle filtering for object tracking based on multiple cues with adaptive parameters and its performance is investigated and evaluated with synthetic and natural sequences and compared with the mean-shift tracker. These estimation approaches can be applied to visual servo control in association with window-matching techniques yielding better results (Tan et al., 2005).

Here, an object tracking algorithm is proposed that combines the window-matching techniques and optimal estimation theory based on the linear stochastic Kalman filtering (Kalman, 1960). The window-matching algorithm is modified and a Kalman filtering stage is coupled to improve the tracking performance. The main objective of this work was to develop the structure of a tracking algorithm not yet its final and efficient implementation, so it was developed within the Matlab computational environment.

The chapter is organized as follows, in Section 2 the object visual tracking problem is stated together with the solving methods. Section 3 discusses the window-matching techniques and presents a window-matching algorithm (WM) for tracking purposes. Section 4 deals with the use of the Kalman filtering (K) to improve the object visual tracking, a new algorithm (WM+K) is then presented. Section 5 then shows the application of the WM+K

algorithm to the following tracking situations: a) ball on a warming up table tennis game; b) vehicle in urban traffic scenery; c) somebody on a two-people meeting and walking scene; and d) a bottle floating on the sea.

2. Object visual tracking

Visual tracking is much related to the correspondence subproblem in vision-based motion analysis. The correspondence problem deals with determining which elements of a frame correspond to which elements of the next frame of the sequence, then, it can be applied for tracking purposes by determining the movement of an entire target region over a long sequence of images. Due to the small spatial and temporal differences between consecutive frames, the correspondence problem can also be stated as the problem of estimating the apparent motion of the image brightness pattern, the so called *optical flow*. The solution of the correspondence problem can roughly follow two strategies (Trucco & Verri, 1998): differential methods and window-matching methods. Differential techniques are based on the spatial and temporal variations of the whole image brightness, generating then the optical flow. Methodologies for motion detection based on differential techniques can be modified to perform object tracking in a sequence of images (Vidal & Casanova, 2005). However, these techniques demand numerical calculation of derivatives that could be impracticable in circumstances where there is a high level of noise, reduced number of frames or the effect of aliasing in the image acquisition process.

Window-matching techniques (Anandan, 1989) are based on the assessment of the degree of similarity among regions in sequential images, so that an object may be recognized and its position inferred in subsequent frames. Window-matching techniques can be applied to object tracking and to other issues in computing vision.

3. Visual tracking based on window-matching techniques

Window-matching methods are based upon an analysis of the grey level pattern around a point of interest and the search for the most similar pattern in the subsequent frame. They are also called *region similarity methods*. Having defined a window $W(x, y)$ around the point $\mathbf{p}(x, y)$, similar windows $W'(x + i, y + j)$ displaced an integer number of pixels are considered. The estimated image displacement corresponds to the minimal of a distance function between the intensity patterns of the two considered windows, which is then obtained by minimizing the function

$$f(W, W'(i, j)) \tag{1}$$

In case correlation functions between distances are used, the problem would be to maximize the cost function. The window-matching techniques assume that: a) the grey level intensity pattern is constant between two successive images; b) there is not a high degree of ambiguity between the texture of the region of interest and other regions of the image.

3.1 Window-Matching (WM) methods for motion detection
According to (Barron et al., 1994), there are several ways of evaluating similarities among grey level intensity patterns in sequential images. The nature and rigidity of the performed

motion directly affect the success (or failure) of the method implementation. The choice of the region of interest (ROI) must be a careful task in order to faithfully reproduce the image actual characteristics.

Problems concerned to bidimensional approximations on image tracking usually happen when the ROI is subjected to complex form and illumination changes. One solution for this problem may be using methods for updating the interest region from the preceding image position in order to minimize geometry and lighting changes (Hager et al., 1998). However, this procedure brings up an undesirable effect, known in the literature as *feature drift*. That happens due to the fact that the ROI new position has a small aggregated error, which continuously builds up with the image motion, so compromising the tracking action.

The regions on the image are represented by squared windows of $N{\times}N$ dimension. The idea is to calculate motion between a region center around a certain point of interest $\mathbf{p}(x,y)$ on image I_1 that will be displaced by integer values i, j (along the horizontal and vertical directions, respectively) in the subsequent image I_2.

To measure similarity the well known SSD (Sum of Squared Differences) cost function will be used here, which is defined as

$$\sum_{i,j=-N/2}^{N/2}[I_1(x,y)-I_2(x+i,y+j)]^2 \tag{2}$$

 Minimizing equation 2 represents minimizing the distance of similarity, and then it means finding, on the subsequent image, the most similar region to the current image. On tracking several objects with independent motion, occlusion can happen. Consequently, some objects may partial or totally disappear in some images; this will cause errors in the object trajectory. To deal with this problem local trajectory restrictions can be imposed and/or uncompleted trajectories can be allowed, since this last approach is properly considered.

Solving equation 2 usually encounters the problem of window-matching between regions with little texture information. Anandan (1989) proposed a methodology to evaluate the reliability of the similarity results obtained with the SSD function. In this work small windows of *5×5* pixels are used for window matching purposes with the candidate points for minimizing the SSD function. The method validation is based on the fact that, on establishing a window matching along a scanning direction, if there is a slight difference in the distance of similarity between windows, it will not be possible to determine the matched window. On the other hand, if there are acceptable similarity distance variations along the scanned direction, it can be concluded that there is a matched window.

3.2 A WM tracking algorithm based upon similarity distance measurement

An algorithm applying the window-matching method by using a similarity distance measurement was developed. The stages of such algorithm are shown in Figure 1. The algorithm is a modified version of the one introduced in (Barron et al., 1994). One of the modifications was the insertion of an additional stage to check the result reliability.

To measure the region similarity the algorithm builds an image distance matrix from which the minimum (or maximum, when the correlation functions were used) values can be obtained. A pre-processing stage was not implemented in the proposed algorithm, as suggested in Anandan (1989), because it is desirable to keep the visual information related to the object boundaries. The adopted methodology demands a substantial computational effort. In order to minimize this effort some procedures are reported by Giachetti (2000).
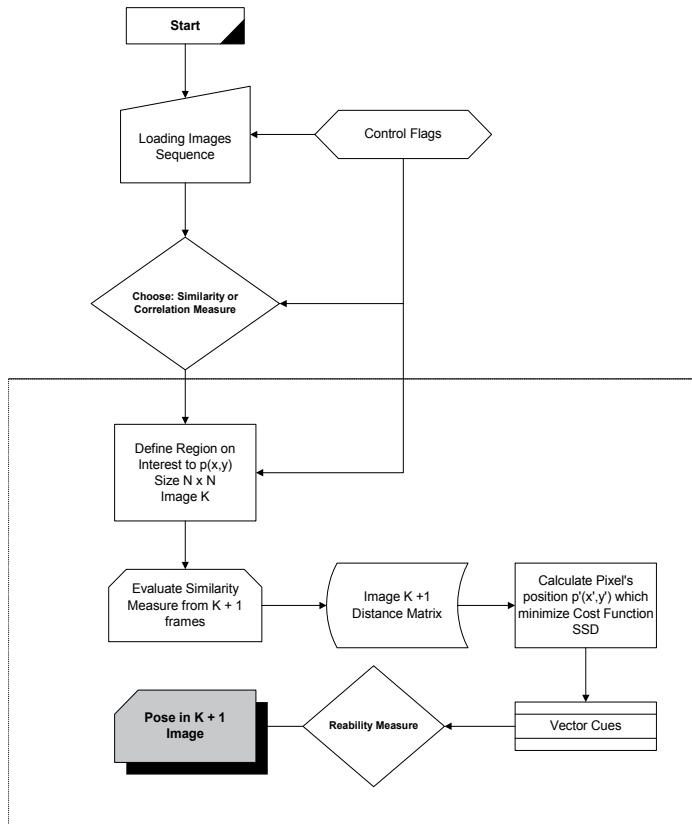
Fig. 1. Window-matching algorithm (WM) for tracking purposes

4. A Kalman filter stage into a WM tracking algorithm

Kalman filtering is a recursive procedure for optimal estimation of the state of a dynamic system, on the basis of noisy measurements and an uncertain model of the system dynamics. For object tracking purposes Kalman filtering can be used to estimate: a) the position of a moving feature point in the next frame, i.e. where to look for the feature; and b) the uncertainty of the estimation, i.e. the degree of confidence of finding the feature in the next frame in a region around the predicted point.

On tracking objects from frame to frame in long sequences of images, there is a fact; the motion of the observed scene is usually continuous, being then possible to make prediction on the motion of the image points, at any instant, based on their previous trajectories. Then object visual tracking can be approached as a problem of state estimation of a dynamic system motion, being the state vector $\mathbf{x} = \begin{bmatrix} x & y & u & v \end{bmatrix}^T$, consisting of the $2D$ position $\mathbf{p} = \begin{bmatrix} x & y \end{bmatrix}^T$ and its respective velocity vector $\dot{\mathbf{p}} = \begin{bmatrix} u & v \end{bmatrix}^T$. As a new frame of the image sequence is acquired and processed at each instant $t_k = t_0 + k.\Delta t$, with $k = 0,1,2,\dots$ and Δt a certain sampling time between frames. Assuming a short sampling time, the state vector does not change much, thus the system model describing the motion dynamics is a time-discrete dynamic equation as

$$x_k = x_{k-1} + u_{k-1}.\Delta t$$
$$y_k = y_{k-1} + v_{k-1}.\Delta t$$
$$u_k = u_{k-1}$$
$$v_k = v_{k-1}$$

(3)

For simplicity (not considering real time) $\Delta t = 1$, thus a state equation that now will include a noise vector $\mathbf{w}_{k-1}$ to represent the system noise is

$$\mathbf{x}_k = \boldsymbol{\Phi}_{k-1}.\mathbf{x}_{k-1} + \mathbf{w}_{k-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.\mathbf{x}_{k-1} + \mathbf{w}_{k-1}$$

(4)

The position vector as determined by the window-matching (WM) procedure will be the measurement vector $\mathbf{z}_k$; it will include a noise vector $\mathbf{v}_k$ to represent the measurement uncertainty. The measurement model will then be

$$\mathbf{z}_k = \mathbf{H}_k.\mathbf{x}_k + \mathbf{v}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.\mathbf{x}_k + \mathbf{v}_k$$

(5)

The noise signals in the noise vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ are considered having Gaussian distribution and zero mean. The corresponding system covariance matrix $\mathbf{Q}_{k-1}$ and the measurement covariance matrix $\mathbf{R}_{k-1}$ are also inputs to the Kalman filter at time t_{k-1}. In the proposed algorithm the window-matching procedure will supply the Kalman filter with "noisy" position observations $\mathbf{z}_k$ from which optimal position and velocity estimates $\hat{\mathbf{x}}_k$ at time t_k will then be obtained.

To initialize the Kalman estimation, arbitrary high values for the process covariance matrix $\mathbf{P}_0$ must be assigned because the filter dynamics takes into account the confidence level of the estimates according $\mathbf{P}_0$. In many cases, undesirable estimates are obtained as a result of bad numerical conditioning of $\mathbf{P}_0$ causing therefore a filter biasing. The implemented filter dynamic equations are

$$\mathbf{P'}_k = \boldsymbol{\Phi}_k.\mathbf{P}_{k-1}.\boldsymbol{\Phi}_k^T + \mathbf{Q}_{k-1}$$
$$\mathbf{K}_k = \mathbf{P'}_k.\mathbf{H}_k^T.\left(\mathbf{H}_k.\mathbf{P'}_k.\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}$$
$$\hat{\mathbf{x}}_k = \boldsymbol{\Phi}_{k-1}.\hat{\mathbf{x}}_{k-1} + \mathbf{K}_k.\left(\mathbf{z}_k - \mathbf{H}_k.\boldsymbol{\Phi}_{k-1}.\hat{\mathbf{x}}_{k-1}\right)$$
$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k.\mathbf{H}_k\right)\mathbf{P'}_k\left(\mathbf{I} - \mathbf{K}_k.\mathbf{H}_k\right)^T + \mathbf{K}_k.\mathbf{R}_k.\mathbf{K}_k^T$$
$$k = 1,2,3,...,n$$

(6)

With $\mathbf{I}$ as the 4×4 identity matrix, and $\mathbf{K}_k$ being the Kalman filter gain. The optimal estimation given by the filter output is vector $\hat{\mathbf{x}}_k$ at time t_k, representing the image position and velocity being its uncertainties described by the diagonal elements of the $\mathbf{P}_k$ matrix.

4.1 A tracking algorithm based on Window-Matching and Kalman filtering (WM+K)
The window-matching (WM) or similarity algorithm shown in Fig. 1 was then modified by inserting a Kalman filter stage. While the window-matching algorithm is running, the Kalman filter processes the resulting measurements generating then outputs to indicate the error tolerance during the WM algorithm execution. In case the similarity algorithm returns values that do not match the conditions previously established, these WM results are dropped and the Kalman filter position estimates are taken as solutions. In this way misleading results, specially those ones produced by feature drift are corrected. Figure 2 shows the developed algorithm (WM+K).
Complementary strategies could be introduced in order to improve this window matching with Kalman filtering. These strategies demand more complex techniques, most of which depend on the use of non-linear models to represent more accurately the attempted tracking. Here linearized models were used to implement the proposed algorithm.

5. Applications of the WM+K tracking algorithm

As mentioned before the objective of this work was to develop a window-matching algorithm with stochastic filtering and verify its performance. For this developing stage the Matlab computational environment was chosen. However, its more critical routines in terms of processing speed were written in *C* language and then converted to *mex* functions (*mexfiles*), a feature in the Matlab environment. *Mexfiles* can be called from within Matlab, so improving the processing speed. The final version of the algorithm will be fully written in C language, provisions for real-time operation will be as well pursued.
The developed tracking algorithms were then applied to these situations with different degree of complexity:
a.   Tracking the ball on a warming up table tennis game;
b.   Tracking a vehicle in urban traffic scenery;
c.   Tracking people meeting and walking in public buildings (two cases); and
d.   Tracking a bottle floating on the sea.
First, the WM tracking algorithm will be applied, then the WM+K algorithm, showing also, in this latter case, the measurement inputs from the WM stage.

5.1 A table tennis sequence[1]
Figure 3 shows the initial, two intermediates and final frames of an image sequence displaying a man practicing for a table tennis game. The problem is tracking the ball during this warming up period of the player. In this sequence the image size is *352×240* pixels with the images in PNG format. For this game sequence the sub-region and searching window sizes were *40×40* and *10×10* pixels respectively.
First, the WM algorithm was applied to the ball tracking problem, then for comparison, the combined WM+K algorithm was applied to the same table tennis sequence. Figure 4 shows, for the same four frames in Fig. 3, the ball tracking results obtained from the WM algorithm. The larger square (blue) is the subregion window and the smaller one (green), the searching window. It can be observed that the WM algorithm is capable of tracking the ball.

---

[1] Downloaded from http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html
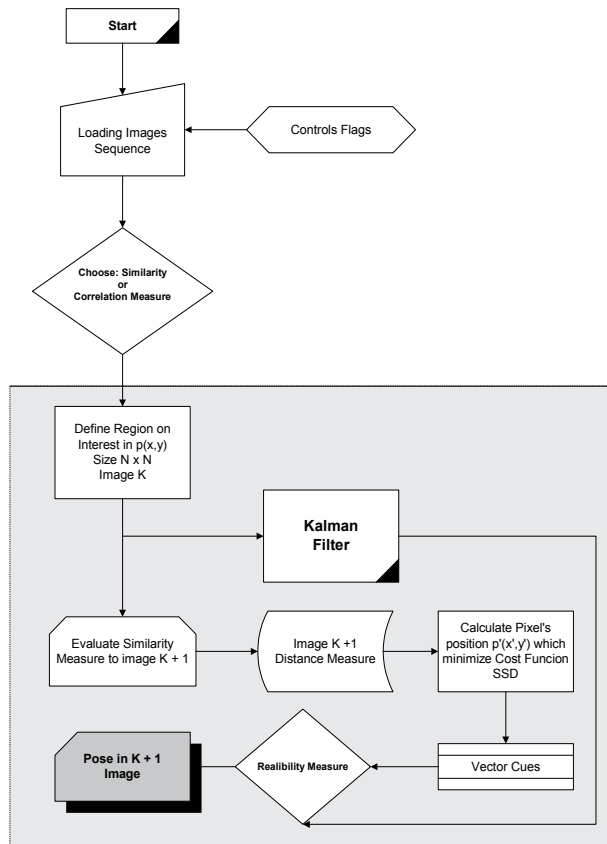
Fig. 2. A Window-matching with Kalman Filter Algorithm (WM+K) for tracking purposes
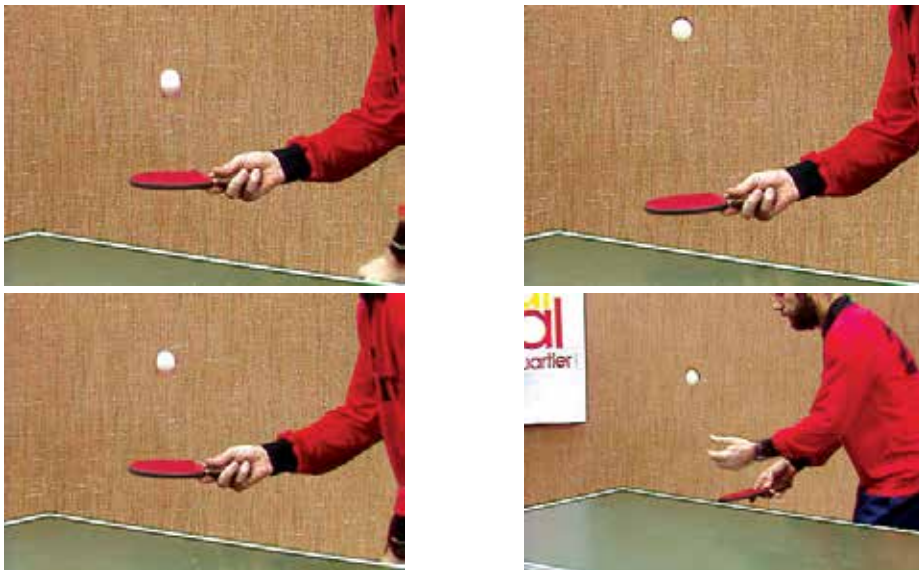
Fig. 3. A table tennis game sequence: initial, two intermediate and final frames
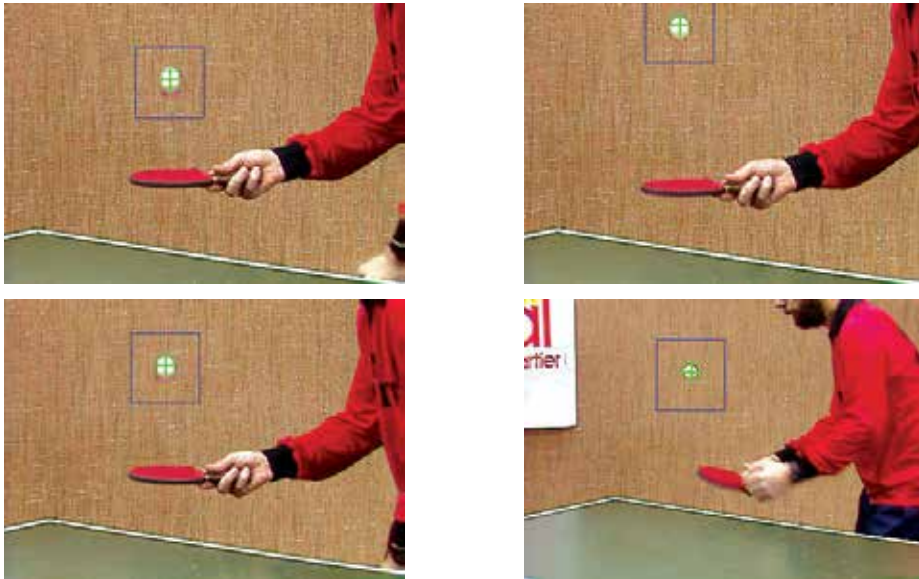
Fig. 4. Tracking the ball with the WM Algorithm

Figure 5 shows, for the same four frames, the ball tracking results from the WM+K algorithm, its tracking window (yellow) is shown together with the now supported WM searching window (green) for comparison. The initial WM+K searching window can be observed on the first frame. For this sequence with well defined environment, both algorithms have a good tracking performance.
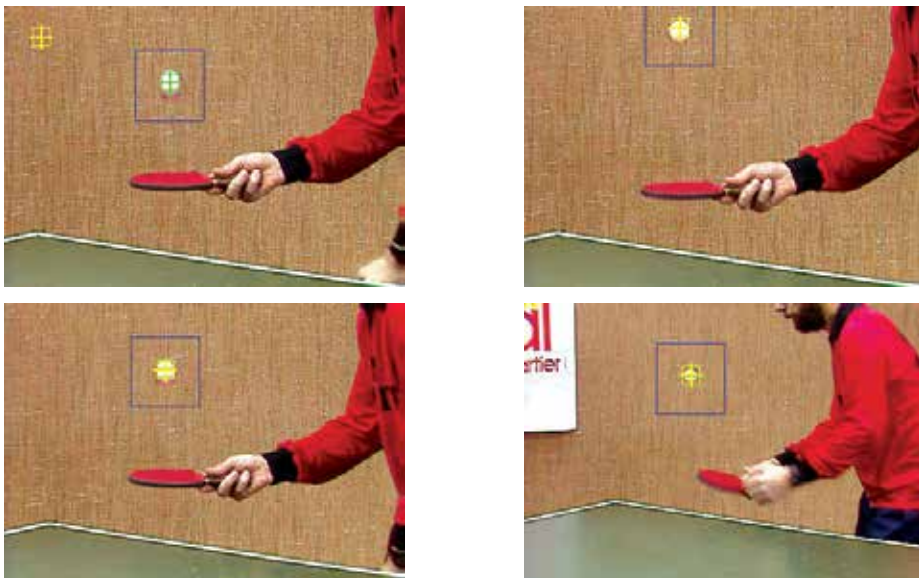
Fig. 5. Tracking the ball with the WM+K algorithm

The ball position (horizontal and vertical) along the sequence of image frames as obtained from both algorithms is shown in Figure 6. There is no much variation along the horizontal direction but is very noticeable along the vertical direction as expected.

	(a) horizontal		(b) vertical
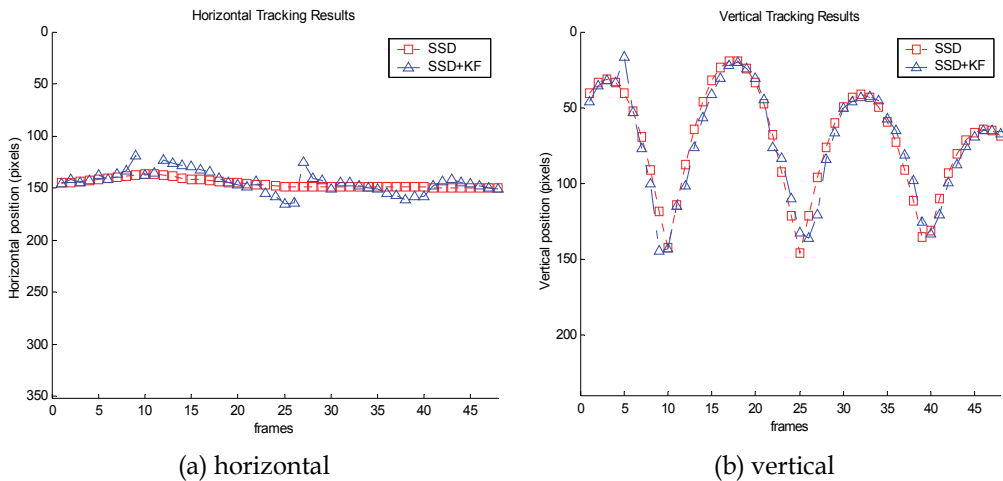
Fig. 6. Ball estimated position from tracking algorithms: □ supported WM algorithm; ∆ WM+K algorithm

5.2 An urban traffic sequence[2]

Figure 7 shows the initial, two intermediates and the final frames of an urban traffic sequence. The scene is a typical urban crossroad with vehicles and pedestrians, commonly found in big cities. The problem now is tracking a particular vehicle. The objects in the scene perform *3D* motion along the vertical and horizontal directions and depth variations. Image size is *320×240* pixels in JPG format. For this sequence the subregion and searching windows were 50×50 pixels and 15×15 pixels respectively.

The choice of the region of interest (ROI) is defined accordingly to the moving objects to be tracked. Other factors that influence the size of the subregions are: the degree of ambiguity, sudden illumination changes, direction, depth, etc. The effects of window size enlargements are: increase in computational cost, window drift caused by ambiguities on ROI and low contrast environments (Giacchetti, 2000). Precisely, the addition of a Kalman filtering stage to a window-matching approach minimizes the effect of these factors. The choice of the window size for the applications in this work was made accordingly to a developed motion detection algorithm (Vidal & Casanova, 2005).

As before, the WM and WM+K algorithms were applied to this sequence. To guarantee a non-biased algorithm on this type of motion, the Kalman, system noise and measurement covariance matrices were randomly initialized. Figure 8 shows, for the same four frames in Fig. 3, the vehicle tracking results from the WM algorithm (green contour). Figure 9 shows the searching results from the WM+K algorithms (yellow) and the now supported WM searching window (green). Here, both algorithms are also capable of tracking the vehicle.

---

[2] Downloaded from http://i21www.ira.uka.de/
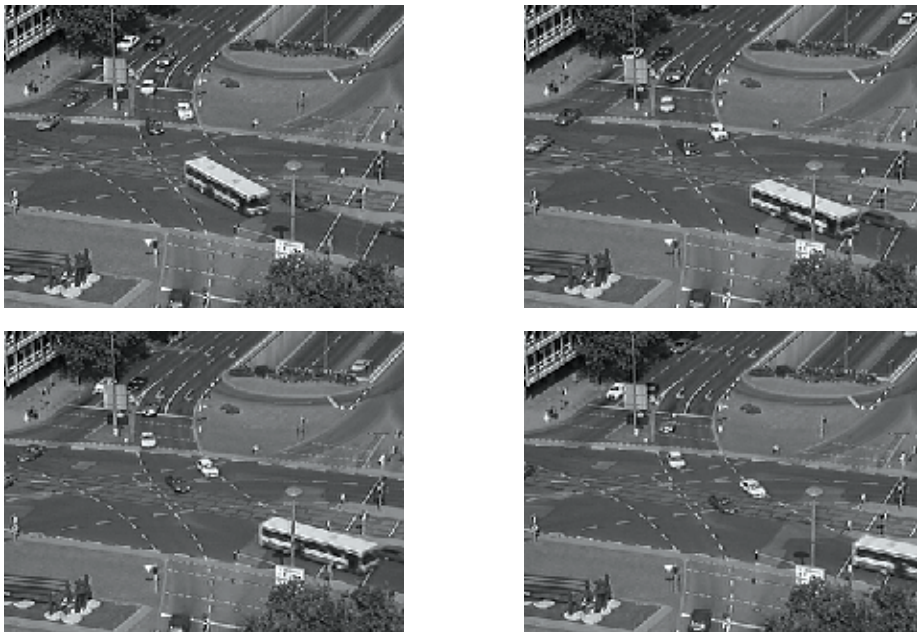
Fig. 7. Vehicle in an urban traffic sequence: initial, two intermediates and final frames
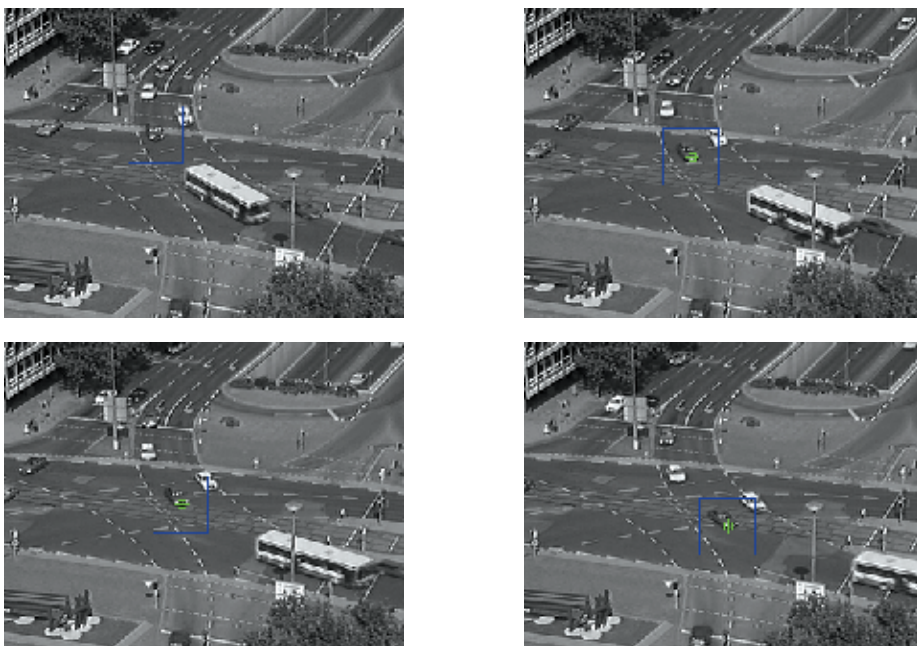
Fig. 8. Tracking a vehicle with the WM algorithm

Figure 10 shows the vehicle position (horizontal and vertical) along the sequence of image frames as obtained from both algorithms. It can be observed that the WM+K algorithm

cannot track the vehicle during the first frames (triangular marks). This is due to the random initialization of the Kalman covariance matrices, but as soon as the WM stage interacts with the filter the algorithm converges quickly.
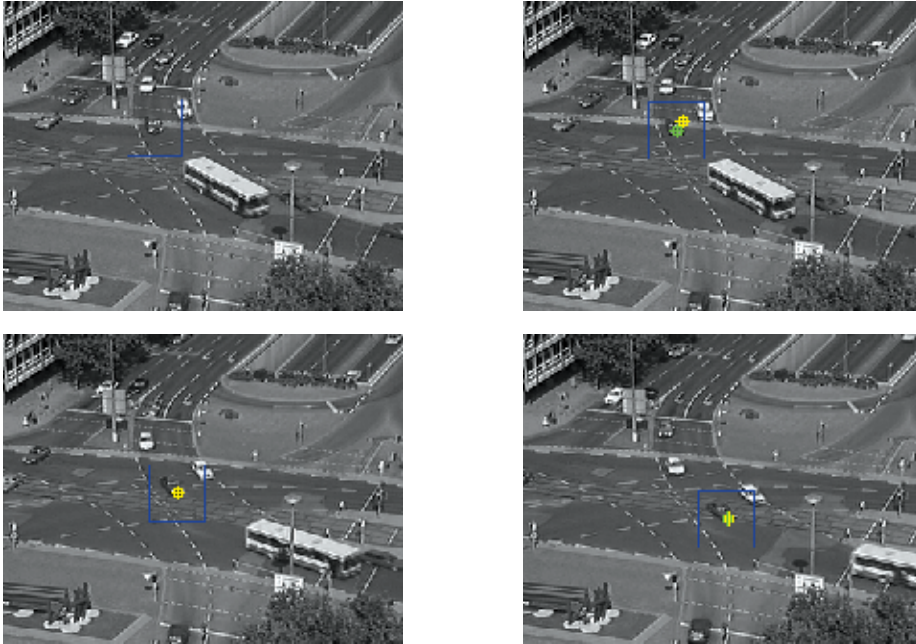
Fig. 9. Tracking a vehicle with the WM+K algorithm

(a) horizontal   (b) vertical
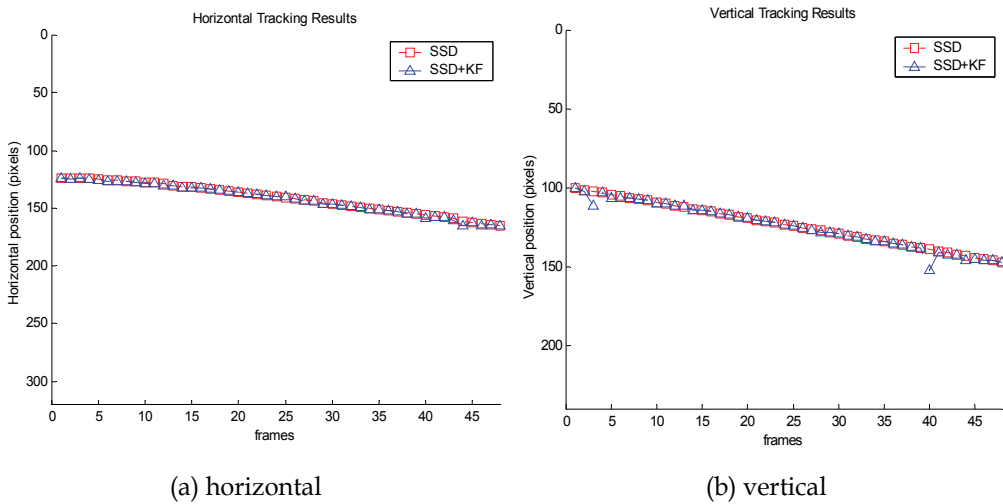
Fig. 10. Vehicle estimated position from tracking algorithms: □ supported WM algorithm; △ WM+K algorithm

5.3 A People Meeting and Walking Sequence[3]

Two cases of a meeting and walking sequence involving people will be considered here. In the first one, two men approach each other and then walk together. Figure 11 shows the initial, two intermediate and final frames of the meeting and walking sequence. This situation is a common scene in public buildings where there is a surveillance system installed. The scene contains problems like photometric distortions, noise, ambiguity and change of scale. For this sequence the image size is *348×288* pixels and the images are in JPG format. A subregion window size of *40×40* pixels and a searching window size of *10×10* pixels were found to be adequate. Figure 12 shows the tracking results of the WM algorithm (red line window), where it can be seen that the algorithm is not capable of tracking the human body, it got lost. On the other hand, Figure 13 shows the searching results (yellow line window) from the WM+K algorithm together with the output from the WM searching window (red line window), now supported by the Kalman filtering stage. Then in spite of the cluttered environment the combined algorithm is able to track the human body till the end of the sequence.

Figure 14 shows the human body estimated position (horizontal and vertical) along the sequence of image frames as estimated by the algorithms. The added Kalman filtering stage then contributes to the robustness of the tracking process.

Fig. 11. People meeting/walking sequence: initial, two intermediates and final frames

---

[3] Downloaded from http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
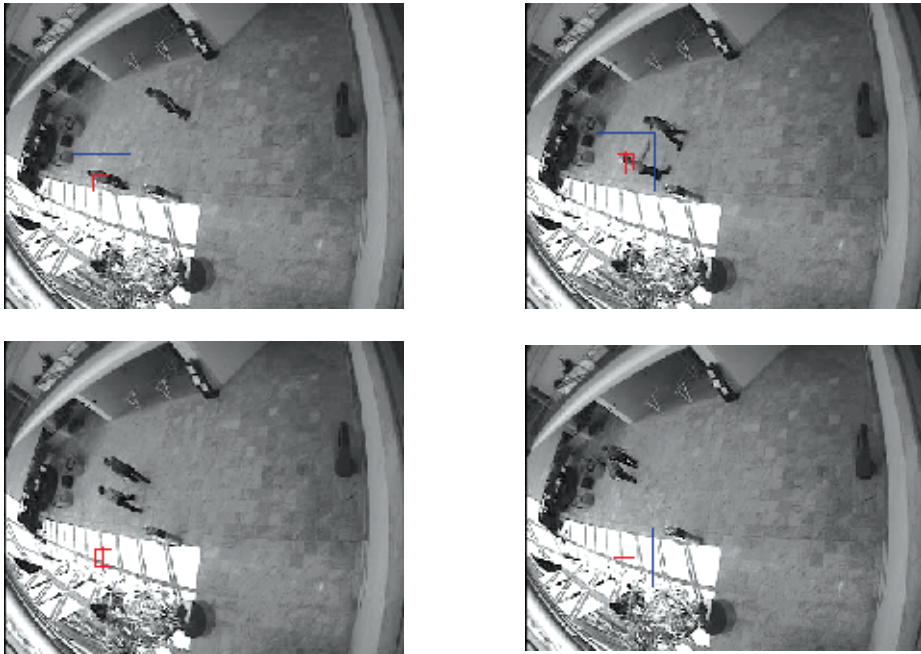
Fig. 12. Tracking people (1) with the WM algorithm
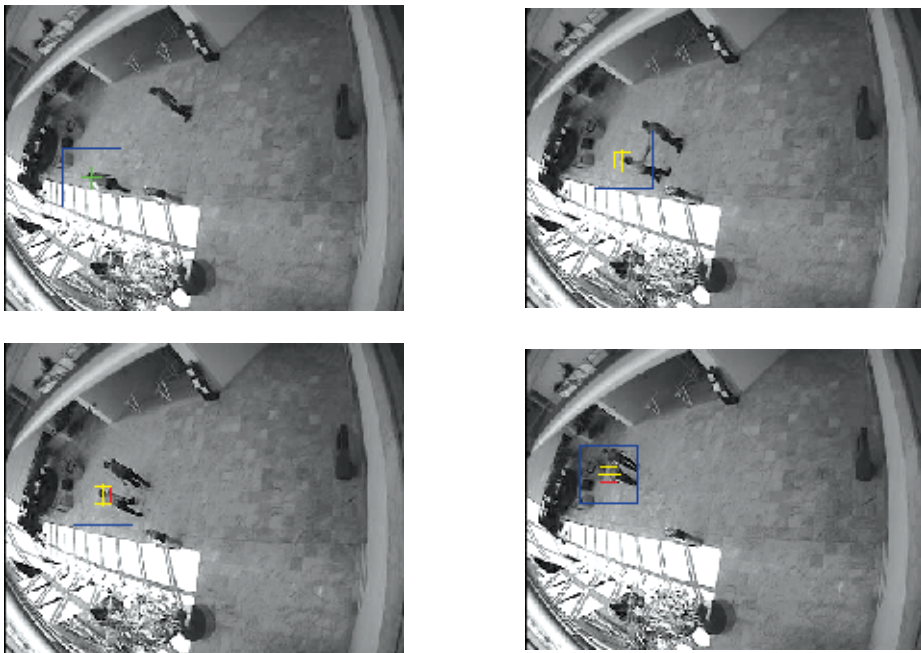
Fig. 13. Tracking people (1) with the WM+K algorithm

(a) horizontal   (b) vertical
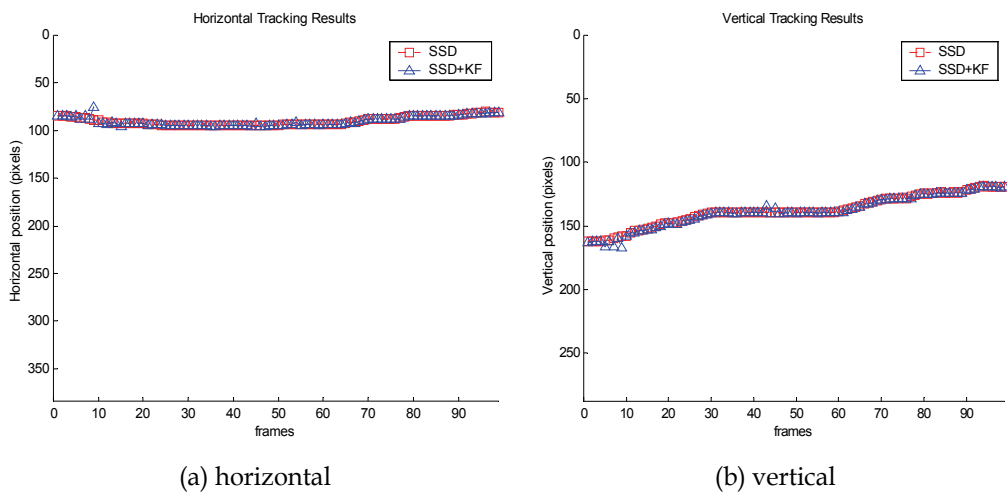
Fig. 14. Human body estimated position from tracking algorithms: □ supported WM algorithm; Δ WM+K algorithm

A second tracking people problem is considered. Figure 15 shows four frames of a sequence where two-people walk side-by-side along a corridor. In this case there are some environmental conditions but not photometric distortion, but there will be a partial occlusion of the tracked human body. The image size is *348×288* pixels and the images are in JPG format. Tracking results are shown in Figures 16 and 17 for the same subregion and searching window sizes as the previous people tracking case.
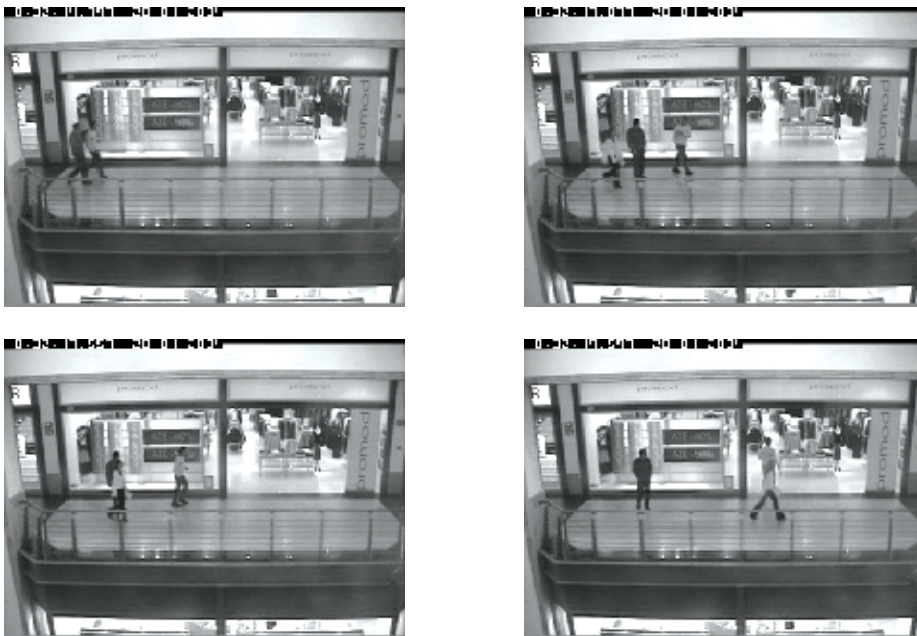
Fig. 15. Two-people walking sequence: initial, two intermediate and final frames
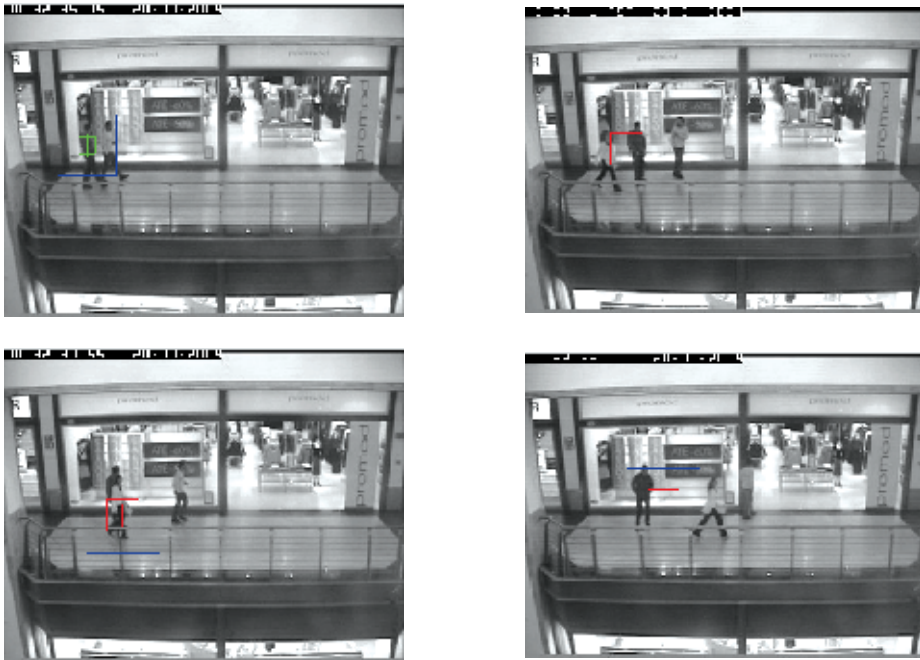
Fig. 16. Tracking people (2) with the WM algorithm
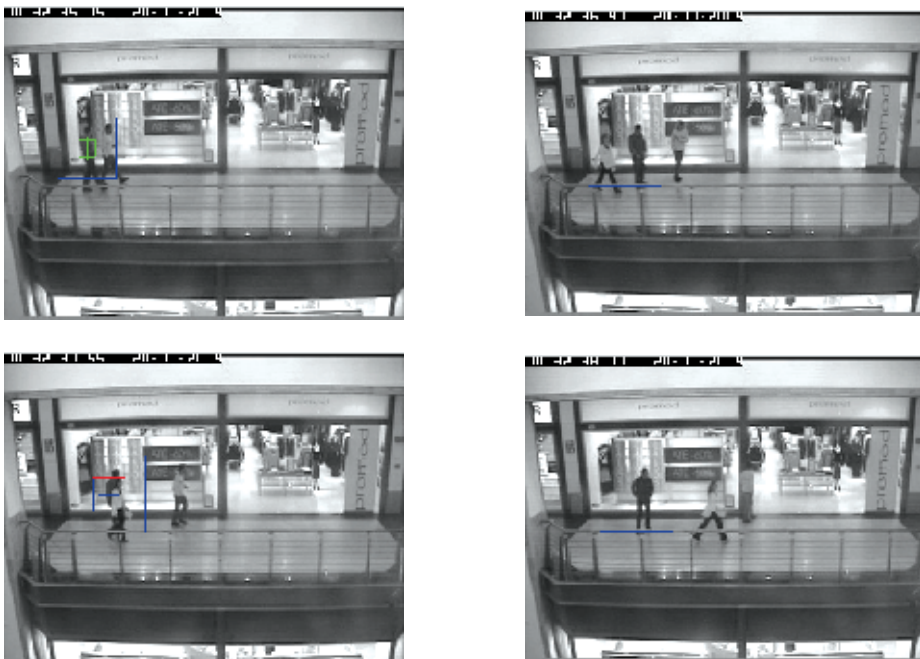
Fig. 17. Tracking people (2) with the WM+K algorithm

Results show that the WM algorithm (red) alone looses the target due to the partial occlusion (frame 3). But the WM+K algorithm (blue) keeps tracking the target in spite of the occlusion, showing that the stochastic filtering adds robustness to the process. The estimated position from the supported WM and the WM+K algorithm are shown in Figure 18.

(a) horizontal   (b) vertical
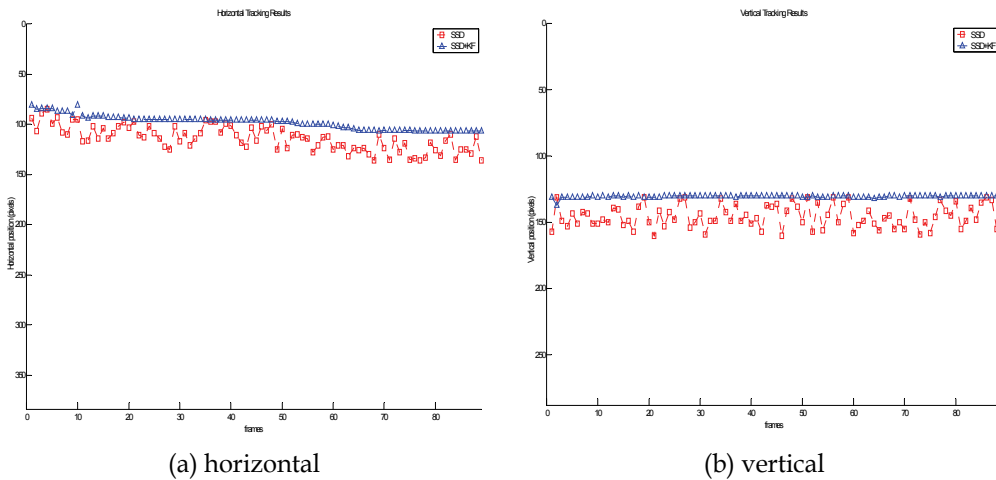
Fig. 18. Human body estimated position from tracking algorithms: □ supported WM algorithm; △ WM+K algorithm

5.4 A Bottle floating on the Sea[4]

This sequence shows a bottle on the sea surface, the same four frames are shown in Figure 19. This is sequence is very particular in the sense that there is a random non-rigid movement with depth variations, blurring effect, scale changes and a high degree of ambiguity. These characteristics demand a robust algorithm to keep tracking the object.

The WM algorithm was applied to the bottle sequence and the tracking results for the four frames (Figure 19) are shown in Figure 20. It is clear that the WM algorithm was not able to keep tracking the bottle, mainly due to the ambiguity with the background. On the other hand the WM+K algorithm keeps tracking the ball despite the particular characteristics of the scene; these tracking results are shown in Figure 21.

Finally, Figure 22 shows the estimated position of the bottle as given by the WM+K algorithm. The bottle trajectory was of random nature and the Kalman stage delivered a smoother trajectory, mainly due to minimization of the *drift* effect.

For the sequences here considered, the larger differences on tracking estimation occurred when the elements of the main diagonal of the error covariance matrix change signs. These changes are caused by the system and measurement noises introduced into the Kalman filter (Jwo, 2007). The proposed WM+K algorithm was capable of tracking targets in these sequences. Thus, it could be said that a WM+K tracking algorithm, consisting of a window matching stage generating measurements for a Kalman estimation filter, produces better tracking results and offers robustness to the object tracking process.

---

[4] Downloaded from
 http://www.cs.bu.edu/groups/ivc/data/DynamicBackgrounds/ICCV2003/water/object7/
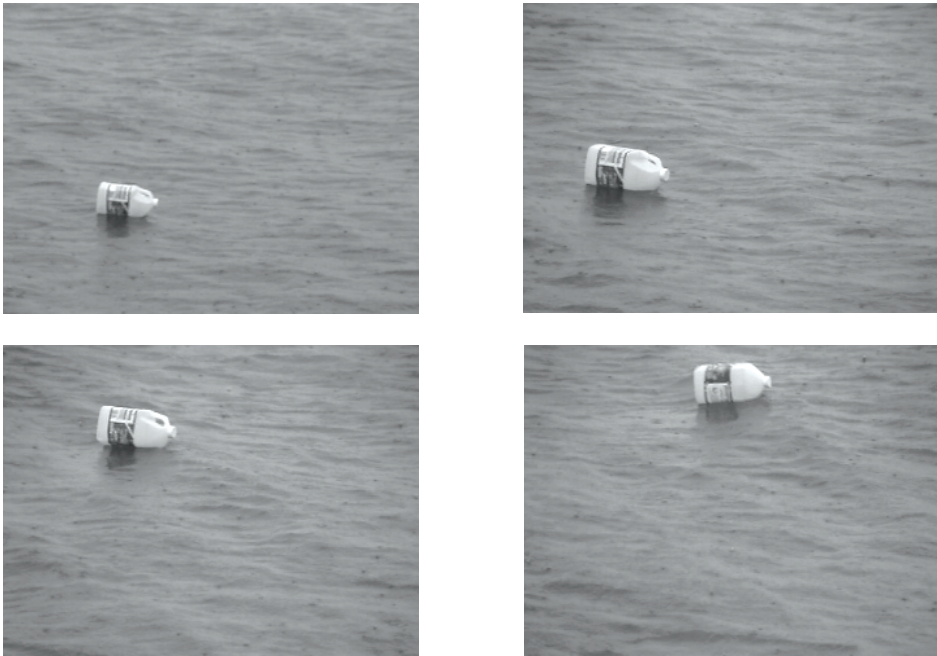
Fig. 19. Bottle floating on the sea sequence: initial, two intermediates and final frames
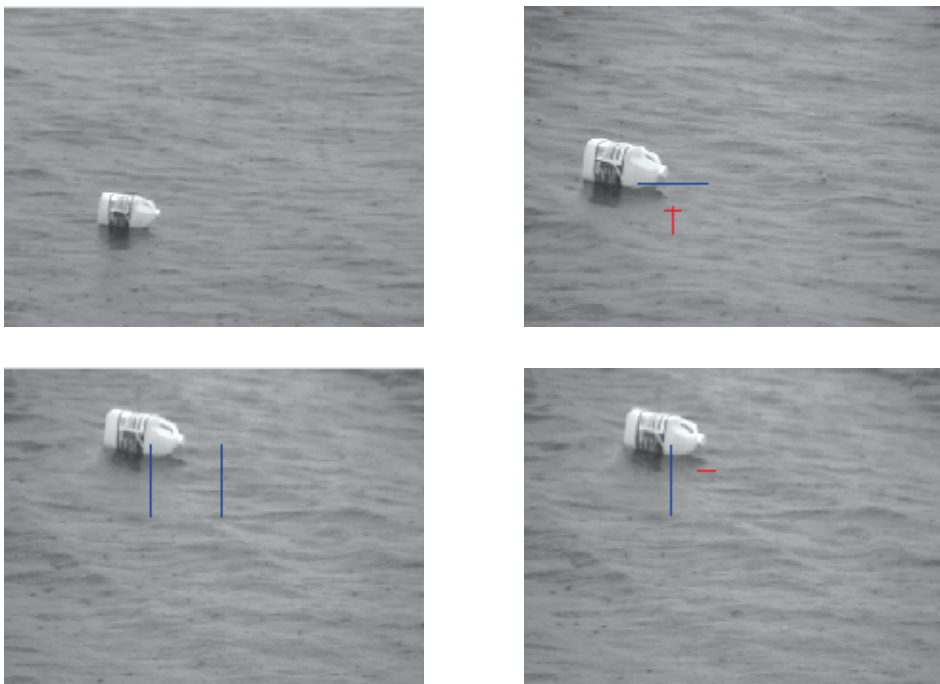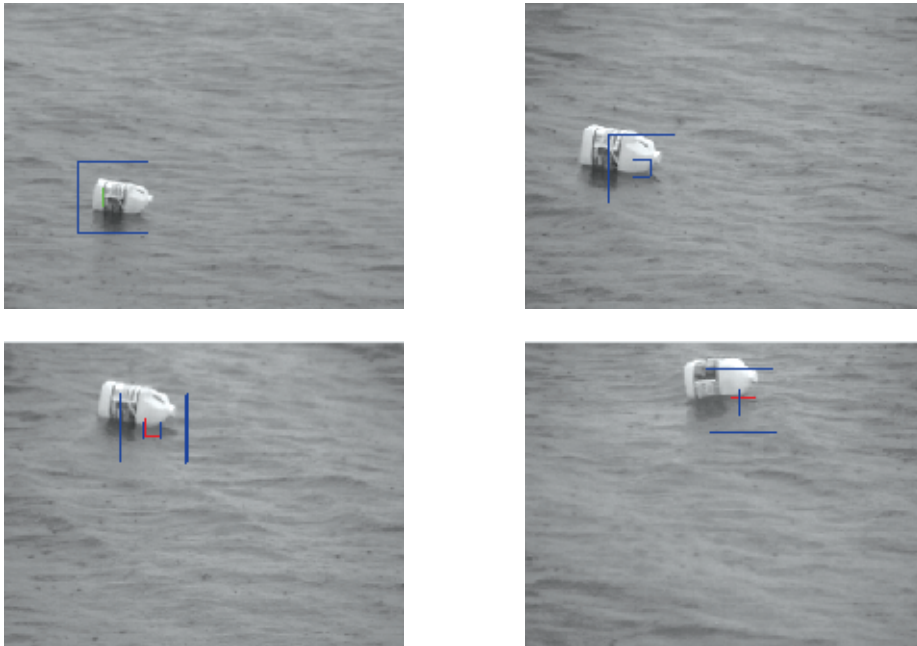

Fig. 20. Tracking a bottle with the WM algorithm
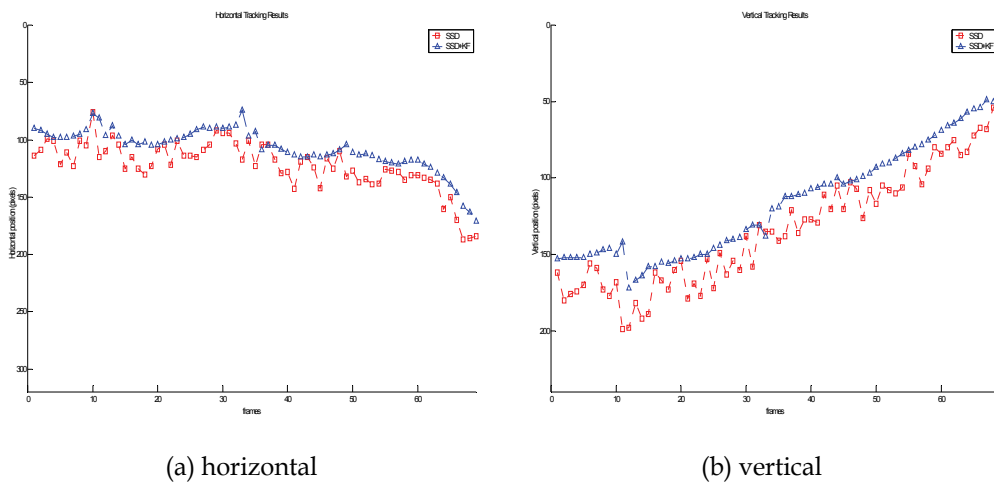
Fig. 21. Tracking a bottle with the WM+K algorithm

(a) horizontal

(b) vertical

Fig. 22. Bottle estimated position from tracking algorithms: □ supported WM algorithm; Δ WM+K algorithm

6. Conclusions

This work presented an algorithm for tracking objects from a sequence of images. The algorithm is based on a window matching approach that uses as a similarity measurement the sum of the square differences (SSD). In order to improve the tracking performance under

disturbances a Kalman filtering stage was incorporated. This joint operation increases the tracking robustness. The algorithm was implemented within the Matlab environment to take advantage of its developing facilities. Assigning scanning subregions contributed to increase the processing speed without compromising the tracking performance. The developed tracking algorithm was applied to track: a) the ball in a table tennis game; b) a vehicle in an urban traffic situation; c) people meeting and walking in buildings; scene; and d) a bottle floating on the sea. The approach presented would provide improvements for visual tracking due to the fact that the tracking is independent of the motion type and of the object shape. The algorithm also offers flexibility in situations where there is no previous information about the object to be tracked. A further work is the algorithm implementation in a high level language and its operation in real time. Also information from additional video cameras could be considered to attack the problem of object occlusion.

7. References

Anandan, P. (1989). A Computational Framework and an Algorithm for the Measurement of Visual Motion, *International Journal of Computer Vision*, Vol. 2, pp. 283–310.

Babu, R. B. ; Perez, P. & Bouthemy, P. (2007). Robust Tracking with Motion Estimation and Local Kernel-based Color Modeling, *Image and Vision Computing*, Vol. 25, No. 8, pp. 1205–1216, ISSN 02628856.

Barron, J.; Fleet, D.J. & Beauchemin, S.S. (1994). Systems and Experiment Performance of Optical Flow Techniques, *International Journal of Computer Vision*, Vol. 12, No. 1.

Chui, C. K. & Chen, G. (1986). *Kalman Filtering with Real-Time Applications*, 3rd edition, Springer, ISBN 038754013X.

Giachetti, A. (2000). Matching Techniques to Compute Image Motion, *Image and Vision Computing*, Vol. 18, pp. 247–260, ISSN 02628856.

Hager, G. D. & Belhumeur, P. N. (1998). Efficient Region Tracking with Parametric Models of Geometry and Illumination, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 10, pp. 1025–1039, ISSN 01628828.

Hutchinson, S., Hager, G.D. & Corke, P. (1996). A Tutorial Introduction to Visual Servo Control, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 651-670.

Jwo, D.J. (2007). Remarks on the Kalman Filtering Simulation and Verification, *Applied Mathematics and Computation*, Vol. 186, No. 1, pp. 159–174, ISSN 00963003.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems, *Journal of Basic Engineering*, pp. 35–45, ISSN 02730979.

Papanikolopoulos, N.P. & Khosla, P.K. (1992). Shared and Traded Telerobotic Visual Control, *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 878–885, ISBN 0818627204.

Rostamianfar, O. ; Janabi-Sharifi, F. & Hassanzadeh, I. (2006). Visual Tracking System for Dense Traffic Intersections, *Proceedings of the Canadian Conference on Electrical and Computer Engineering - CCECE '06*, pp. 2000–2004, ISBN 1424400384.

Shen, H. ; Nelson, G. ; Kennedy, S.; Nelson, D. ; Johnson, J. ; Spiller, D. ; White, R. H. & Kell, D. B. (2006). Automatic Tracking of Biological Cells and Compartments using Particle Filters and Active Contours, *Chemometrics and Intelligent Laboratory Systems*, Vol. 82, No. 1-2, pp. 276–282, ISSN 01697439.

Sumi, Y. ; Ishiyama, Y. & Tomita, F. (2007). Robot-vision Architeture for Real-time 6-dof Object Localization, *Computer Vision and Image Understanding*, Vol. 105, pp: 218–230.

Tan, S. S. & Hart, D. P. (2005). A Fast and Robust Feature-based 3D Algorithm using Compress Image Correlation, *Pattern Recognition Letters*, Vol. 26, No. 11, pp. 1620–1631, ISSN 01678655.

Trucco, E. & Verri, A. (1998). *Introductory Techniques for 3-D computer Vision*, Prentice Hall, ISBN 013261108-2, New Jersey, USA.

Urtasun, R.; Fleet, D. J. & Fua P. (2006). Temporal Motion Models for Monocular and Multiview 3D Human Body Tracking, *Computer Vision and Image Understanding*, Vol. 104, pp.157–177, ISSN 10773142.

Veeraraghavan, H. ; Schrater, P. & Papanikolopoulos, N. (2006). Robust Target Detection and Tracking through Integration of Motion, Color and Geometry, *Computer Vision and Image Understanding*, Vol. 103, No. 2, pp. 121–138, ISSN 10773142.

Vidal, F.B. & Casanova Alcalde, V.H. (2005). Motion Segmentation in Sequential Images based on the Differential Optical Flow. *Proceedings of the International Conference on Informatics in Control, Automation & Robotics*, ICINCO, Barcelona, Spain, pp. 94–100.

Vidal, F.B. & Casanova Alcalde, V.H. (2007). Window-Matching Techniques with Kalman Filtering for an Improved Object Visual Tracking, *Proceedings of the IEEE Conference on Automation Science and Engineering, IEEE CASE 2007*, Scottsdale, AZ, USA, pp. 829–834, ISBN 1424411548.

Yau, W. G.; Fu, L-C. & Liu, D. (2001). Design and Implementation of Visual Servoing System for Realistic Air Target Tracking, *Proceedings of the IEEE International Conference on Robotics and Automation - ICRA*, Vol. 1, pp. 229–234, ISBN 0780365793.

Kalman Filtering for NLOS Mitigation and Target Tracking in Indoor Wireless Environment

Chin-Der Wann
National Sun Yat-Sen University
Taiwan

1. Introduction

Kalman filter and its nonlinear extension, extended Kalman filter provide a feasible solution to mitigating non-line of sight (NLOS) propagation effects, and therefore improving accuracy of mobile target tracking in indoor wireless environments. Most wireless communication systems for indoor positioning and tracking may suffer from different error sources, including process errors, measurement errors, the NLOS propagation effects and dense multipath arrivals. The errors sources, if not properly eliminated or mitigated, generally yield severe degradation of accuracy in ranging, positioning and tracking. Among the factors that cause performance degradation, the NLOS effect is considered the major error source in indoor location systems using one or more types of measured location metrics.

Accurate indoor positioning and tracking play an important role in home safety, public services, and other commercial or military applications (Pahlavan et al., 1998). In recent years, indoor localization has drawn increasing interests from academia and industry. There is an increasing demand of indoor localization systems for tracking persons with special needs, such as the elders and children who may be away from visual supervision. Other applications need the solutions to tracing mobile devices or movable objects in the covered areas of sensor networks, or localizing accurately in-demand portable equipments in hospitals and laboratories. In public safety and military operations, the tracking systems can be used in navigating and coordinating police officers, fire-fighters or soldiers to complete cooperative missions inside buildings.

Various positioning techniques have been developed in the past few years. Handset-based positioning methods generally require that a modified handheld device calculate its own position by using a fully or partially equipped global positioning system (GPS) receiver. The method is, however, unfortunately not suitable for many indoor localization applications. Network-based methods have their advantages in wireless location and indoor positioning. The methods can be used for location estimation in situations where GPS solutions are not applicable. In the network-based wireless location, various schemes using received signal strength indication (RSSI) have also been extensively investigated in the past two decades. In location systems using the RSSI, location estimation is usually obtained from or augmented with the location fingerprinting scheme. Though no complex measurement equipments are involved, the build-up of a "radio map" of RSSI may be time-consuming.

The RSSI data may vary from time to time when the layout or environment becomes different from that when the RSSI data are collected. Due to the limitations of spectral natures of transmitted signals in the location systems, accuracy and precision become a major challenge when applications other than "tour-guiding'" services are targeted.

To tackle the problems of mobile positioning and tracking in indoor wireless NLOS environments, a variety of techniques and systems have been studied in recent years in the hope of attaining better location accuracy. Time of arrival (TOA) and time difference of arrival (TDOA) are two typical time-related parameters usually used in pinpointing the location of a mobile station. In addition, techniques using angle of arrival (AOA) have also been studied by many researchers. When line of sight (LOS) transmission exists between a transmitter and a receiver, the signal arrival time or signal arrival angle may be correctly obtained if the SNR is high and the multi-paths from the propagation channels are resolved properly. In situations where NLOS propagation exists, suitable NLOS mitigation techniques are needed for improving the accuracy of ranging and localization.

For applying the TOA and TDOA parameters in locating mobile stations or targets, the true range between a transmitter and a receiver in the wireless environment is correctly calculated only when the direct path of signal propagation is present, which may not always be possible, especially in indoor environments. In most cases, errors caused by the NLOS effects cannot be ignored in wireless location systems where high accuracy is demanded.

Several NLOS mitigation techniques for mobile positioning systems have been presented in the past few years (Le et al., 2003; Najar & Vidal, 2003; Wylie & Holtzmann, 1996). In (Wylie & Holtzmann, 1996), a simple binary hypothesis testing was used for NLOS identification by exploiting the known statistics of the receiver measurement noise. To mitigate the NLOS effects, polynomial fitting was applied to all available measured range data for data smoothing and variance calculation. Since a whole block of measured data are needed for the process of polynomial fitting, accurate and real-time mobile positioning may not be possible due to the time delay in collecting enough data. In other mobile location estimation methods, biased versions of the Kalman filter used in mitigating the NLOS range error (Le et al., 2003). A coefficient for adjusting the noise covariance matrix needs to be chosen by experiment to obtain good location estimation results. In (Najar & Vidal, 2003), a modified Kalman algorithm with NLOS error estimation was proposed for UMTS mobile positioning. The estimation of range bias in the algorithm provided performance improvement of location tracking in the NLOS environments.

Since most wireless communications systems used in wireless location may suffer from the NLOS and dense multipath situation, it is an important issue to obtain higher accuracy in determining signal arrival time for the time-based location systems. The ultra-wideband (UWB) radio system, in addition to the usage for communications, can provide users with the abilities of high accurate location estimation and tracking. As a good candidate for low-power high-speed wireless communications, the ultra-wideband (UWB) radio technology has gained many interests in recent years for its applications in indoor communications. In indoor environments, the UWB systems, based on the spectral characteristics of the signals, are capable of tackling the multipath effects and providing finer and more accurate measurements in ranging than other narrow-band systems. With the fine time resolution, the accuracy of UWB location systems can be within one inch. The UWB systems though provide potentially accurate ranging for indoor positioning and tracking, the NLOS propagation errors caused by blocked LOS paths between a base station and the mobile

station may still lead to severe degradation of position accuracy, posing a major challenge to positioning and tracking. Suitable NLOS identification and mitigation techniques are basically required in the systems for achieving better positioning performance.

This chapter will first present the UWB signal models in LOS and NLOS indoor environments. The data smoothing schemes using Kalman filter and polynomial fitting technique for identifying the NLOS status of the transmission are discussed. To improve the accuracy of time-based UWB range estimation, the NLOS mitigation techniques using biased Kalman filters will be covered. Some problems of applying the biased Kalman filtering in the NLOS identification and mitigation are addressed, followed by solutions to improving the correctness of hypothesis tests in the identification stage, and reducing the exceeding negative adjustment effects of biased Kalman filtering in the mitigation stage. The applications of extended Kalman filters on wireless positioning and tracking are then presented. A network-based location system, in which location metrics from multiple base stations are used for mobile target location estimation is studied. Positioning techniques using hybrid TDOA/AOA (time difference of arrival/angle of arrival) location metrics in the UWB environments will be presented. Simulation results are included to show the capability of Kalman filter-based architecture in mitigating the NLOS errors and improving the accuracy of target positioning and tracking in the UWB indoor wireless location system.

2. Non-line of sight propagation errors

2.1 Range measurement model

We assume that there are multiple base stations (BS's) in UWB indoor wireless location systems. In dealing with the non-line of sight propagation effects, the range measurement between a rover (or mobile station) and the m-th base station, corresponding to the TOA location metrics of the m-th base station can be modelled as

$$r_m(t_i) = L_m(t_i) + n_m(t_i) + NLOS_m(t_i) \tag{1}$$

where $r_m(t_i)$ is the measured range at the sampling time t_i, $L_m(t_i)$ is the true range, $n_m(t_i)$ is the measurement noise and modelled as a zero-mean additive Gaussian random variable with standard deviation σ_m, and $NLOS_m(t_i)$ is the NLOS error component in the received signal. There will be no NLOS error component if the line-of sight propagation path exists, and $NLOS_m(t_i) = 0$. The measurement error $n_m(t_i)$ becomes the only source of range measured error.

In a dense multipath UWB indoor environment, the estimation of the arrival time of the first path can be directly related to the measured range data at each base station, as in (1). The IEEE UWB channel modelling subcommittee adopted a modified Saleh-Velenzuela (S-V) model, which seemed to best fit the UWB channel measurements (Molisch, et al., 2003). The S-V model was used in modelling the multipath of an indoor environment for wideband channel. The channel measurements showed that multipath arrivals in clusters rather than in a continuous form (Saleh & Valenzuela, 1987), as shown in Fig. 1. Assume that T_0 is the arrival time of the first path in the first cluster. The arrival time T_0 can be related to the positive NLOS error component $NLOS_m(t_i)$ at the time instant t_i. For the LOS cases, we have $T_0 = 0$ and $NLOS_m(t_i) = T_0 \times c = 0$, where c is the speed of light.

The arrival time T_0 for the NLOS cases can be modelled as an exponential distribution and described by the following equation (Molisch, et al., 2003)

$$p(T_0) = \Lambda \exp[-\Lambda(T_0)] \tag{2}$$

where $\Lambda[1/nsec]$ is the cluster arrival rate. The multipath cluster arrival rates under different UWB channel models, CM1 through CM4, are listed in Table 1. The related parameters listed by the IEEE UWB channel modelling subcommittee have been used in many simulations and technical designs of a variety of UWB systems.
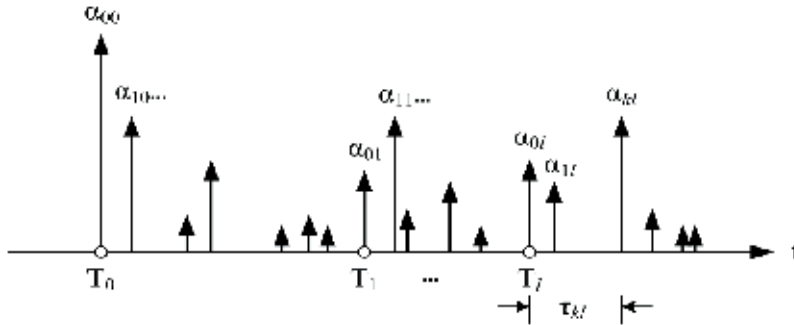
Fig. 1. Multipath delay profile of typical UWB channels

UWB Channel	CM 1	CM 2	CM 3	CM 4
Tx/Rx separation	0-4m	0-4m	4-10m	>10m
Λ (1/nsec)	0.0233	0.4	0.0667	0.0667
LOS/NLOS condition	LOS	NLOS	NLOS	NLOS

Table 1. The cluster arrival rate of multipath used in UWB channel models

2.2 Non-line of sight identification

For the NLOS error problems in mobile position location, several NLOS identification and mitigation techniques have been presented in the past few years (Thomas et al., 2000; Wylie & Holtzmann, 1996). These approaches identified the BS's that have NLOS components in the received range data, and tried to reduce the time-based NLOS errors by using the NLOS mitigation techniques. In (Wylie & Holtzmann, 1996), a simple binary hypothesis test was used for the NLOS identification with an understanding that the standard deviation of the NLOS propagation errors is generally much larger than that of measurement errors in the LOS situation. The understanding may also be applied to the UWB transmission environments. In (Wylie & Holtzmann, 1996), prior to the binary hypothesis testing, polynomial fitting was applied to all available measured mobile range data collected during a block of time interval for variance calculation and data smoothing. Since the whole block of measured data were needed for the process of polynomial fitting, real-time positioning may not be feasible. For mitigation of the NLOS errors $NLOS_m(t_i)$, the existence of non-zero NLOS component may need to be identified first. As shown in Fig. 2, the measured range data are first processed to obtain the smoothed data, which can further be used as averaged values at the corresponding time instants. The standard deviation of the measured range

data with respect to the smoothed values can then be calculated and used in the proceeding LOS/NLOS hypothesis test.

Fig. 2. Identification of LOS/NLOS propagation status

By applying the abovementioned identification scheme to wireless location systems, the range (or TOA) measurements related to each base station can be smoothed by using the least squares technique in solving the coefficients for fitting the modelled N-th order polynomial function (Wylie & Holtzmann, 1996). The standard deviation of the measured range data can be obtained by computation over a block of K range data $r_m(t_i)$ periodically:

$$\hat{\sigma}_m = \sqrt{\frac{1}{K}\sum_{i=1}^{K}\left(r_m\left(t_i\right)-S_m\left(t_i\right)\right)^2} \tag{3}$$

where $S_m(t_i)$ is the smoothed range data, which are obtained from polynomial fitting.

The data smoothing can also be conducted by other methods. In (Le et al., 2003), the range data were smoothed by utilizing biased Kalman filtering approach. In the case, the smoothed range data $S_m(t_i)$ were obtained from the output of the biased Kalman filter. The estimated standard deviation obtained in (3) is then compared with a predetermined threshold in the simple hypothesis testing, represented as

$$\begin{aligned}H_0 &: \ \hat{\sigma}_m(k) < \gamma\sigma_m \qquad \text{LOS case}\\ H_1 &: \ \hat{\sigma}_m(k) \geq \gamma\sigma_m \qquad \text{NLOS case}\end{aligned} \tag{4}$$

where σ_m is the standard deviation of the measurement noise in the LOS environment. The scaling parameter γ is chosen experimentally to reduce the probability of false alarm.

For the hypothesis testing in (Le et al., 2003), a periodical interval checking method was used. Periodical LOS/NLOS checking schemes, however, have some drawbacks. First of all, the block size of data samples for variance calculation needs to be chosen experimentally. Secondly, the period of using the hypothesis testing result may not be easily determined and must be decided experimentally as well. In Fig. 3, the block of C data samples, shown as shaded bars [0, C], are used for status checking. The resulting LOS/NLOS status will be kept unchanged until the time instant N, when another cycle of status identification begins and the new LOS/NLOS status are used. To choose feasible time instants C and N would need consideration of more factors in the transmission channels for the positioning systems. In addition to the problems, since the result of the LOS/NLOS hypothesis testing is used as the channel status until the next new periodical checking result is obtained, it is very likely that an NLOS-to-LOS or LOS-to-NLOS transition time instant is missed or incorrectly detected. The incorrect identification of channel status or the incorrect transition time may yield large estimation errors in ranging and localization.
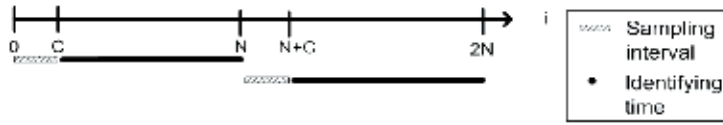
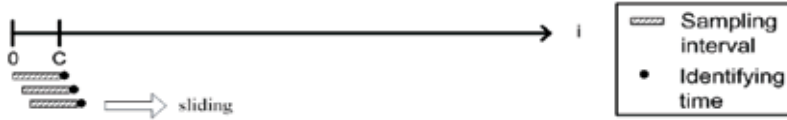Fig. 3. Periodical interval checking for identifying LOS/NLOS status

Fig. 4. Periodical interval checking for identifying LOS/NLOS status

To avoid the drawbacks, an identification scheme using a sliding window can be used in processing the measured range data at the base stations. The function of the sliding window spans from the data smoothing step, the calculation of standard deviation step to the LOS/NLOS identification step. A fixed-length data window of C data samples, shown as the shaded bars $[0, C]$ in Fig. 4, will slide to the right in the time axis as the new data sample is available for processing. The sliding window scheme is considered compatible with the schemes using Kalman filtering.

3. Kalman filtering for data processing

3.1 Data smoothing for the NLOS hypothesis testing
A Kalman filter can be used in estimating the state vector of a mobile target from the observed range data, and therefore smoothing the range data. Assume that the state vector of a mobile can be represented as (Mendel, 1987)

$$\mathbf{X}(k+1) = \mathbf{\Phi}\mathbf{X}(k) + \mathbf{\Gamma}W(k) \tag{5}$$

where $\mathbf{X}(k) = [L(k) \quad \dot{L}(k) \quad \ddot{L}(k)]^T$ is the state vector of the mobile target related to the measured data of a base station at the time t_k, $W(k)$ is the driving noise vector with an assumed covariance $Q = \sigma_w^2$, and the state transition and noise transition matrices can be written as

$$\mathbf{\Phi} = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \text{ and} \tag{6}$$

$$\mathbf{\Gamma} = \begin{bmatrix} 0 \\ 0 \\ \Delta t \end{bmatrix}. \tag{7}$$

The measurement process is represented as

$$Z(k) = \mathbf{H}\mathbf{X}(k) + U(k) \tag{8}$$

where $Z(k)$ is the measured data, measurement matrix $\mathbf{H} = [1\ 0\ 0]$, and $U(k)$ is the measurement noise with covariance $R = \sigma_u^2$. The iterative operations of the Kalman filter can be summarized as follows:

$$\widehat{\mathbf{X}}(k \mid k-1) = \mathbf{\Phi}\widehat{\mathbf{X}}(k-1 \mid k-1) \tag{9}$$

$$\mathbf{P}(k \mid k-1) = \mathbf{\Phi}\mathbf{P}(k-1 \mid k-1)\mathbf{\Phi}^T + \mathbf{\Gamma}\mathbf{Q}\mathbf{\Gamma}^T \tag{10}$$

$$\mathbf{K}(k) = \mathbf{P}(k \mid k-1)\mathbf{H}^T[\mathbf{H}\mathbf{P}(k \mid k-1)\mathbf{H}^T + \mathbf{R}(k)]^{-1} \tag{11}$$

$$\widehat{\mathbf{X}}(k \mid k) = \widehat{\mathbf{X}}(k \mid k-1) + \mathbf{K}(k)[Z(k) - \mathbf{H}\widehat{\mathbf{X}}(k \mid k-1)] \tag{12}$$

$$\mathbf{P}(k \mid k) = \mathbf{P}(k \mid k-1) - \mathbf{K}(k)\mathbf{H}\mathbf{P}(k \mid k-1) \tag{13}$$

where $\mathbf{K}(k)$ is the Kalman gain vector and $P(k \mid k)$ is the covariance matrix of $\widehat{\mathbf{X}}(k \mid k)$.
By using the Kalman filtering, the standard deviation of the observed range data can be calculated and then used in the LOS/NLOS hypothesis testing. To avoid the drawbacks of using polynomial fitting and periodical interval checking methods, as discussed in the previous subsection, the sliding window scheme for processing the measured range data can be integrated with the biased Kalman filtering and hypothesis testing in the UWB location system. The standard deviation of data over a sliding block of C TOA (range) measurements $r_m(t_i)$ at the m-th base station can be obtained as

$$\hat{\sigma}_m(k) = \sqrt{\frac{1}{N} \sum_{i=k-(C-1)}^{k} \left(r_m(t_i) - \hat{L}_m(t_i) \right)^2} \cdot \tag{14}$$

In each processing cycle, the standard deviation calculated at base station m is then used in the hypothesis testing, as represented in (4). If the LOS TOA propagation status is decided, an unbiased Kalman filter is used to smooth the TOA data at each BS. In the contrast, if the NLOS propagation scenario is detected, a biased Kalman filter is used in mitigating the NLOS TOA error.

3.2 Biased Kalman filtering
In a biased Kalman filter, based on the result from LOS/NLOS hypothesis testing, different values of noise covariance are assigned. Under the LOS situation, unbiased smoothing is used for estimating the true TOA. When the NLOS status is detected, the positive NLOS range error in the measured range (or TOA) data can be effectively reduced in the filtering cycle by assigning the diagonal elements of the noise covariance matrix

$$\hat{\sigma}_u(k) = \alpha\sigma_m, \text{ if } Z(k) - \mathbf{H}\mathbf{X}(k \mid k-1) > 0 \text{ and NLOS detected,}$$
$$= \sigma_m, \quad \text{otherwise,} \tag{15}$$

where α is an experimentally chosen scaling factor. The processed TOA data from all base stations are then used in mobile localization and tracking.

3.3 Biased Kalman filtering for TDOA location metrics

Similar to the cases in the TOA location methods, the Biased Kalman filters can also be applied in the TDOA location approaches. Assume that there are M base stations available for measuring the TOA's for locating a mobile station. The first base station, BS1 is selected as the reference base station. The difference of ranges of the m-th base station and the first base station to the mobile station in the location system can be modelled as

$$
\begin{aligned}
d_m(t_i) &= r_m(t_i) - r_1(t_i) \\
&= L_m(t_i) - L_1(t_i) + n_{d,m}(t_i) + NLOS_{d,m}(t_i) \\
m &= 2,...,M
\end{aligned}
\tag{16}
$$

where $n_{d,m}(t_i)$ is defined as the measurement noise of the range difference, $n_{d,m}(t_i) = n_m(t_i) - n_1(t_i)$, and $NLOS_{d,m}(t_i)$ is the NLOS error component of the range difference, $NLOS_{d,m}(t_i) = NLOS_m(t_i) - NLOS_1(t_i)$. Hence, $n_{d,m}(t_i)$ can be modelled as a random variable, at time instant t_i, with independently identical joint probability density function with $n_{d,m}(t_i) \sim N(0, 2\sigma_m^2)$.

To perform range difference estimation in the TDOA data processing, the Kalman filtering scheme discussed above can be applied in the cases here. Three cases with different LOS/NLOS combinations are considered for determining the biased measurement noise covariance $R = \hat{\sigma}_u^2$ in the system.

Case 1: BS m is an NLOS BS, and BS 1 is an LOS BS. We have

$$
Z_m(k+1) - \mathbf{H}\mathbf{X}_m(k+1|k) > 0
$$

Case 2: BS 1 is an NLOS BS, and BS m is an LOS BS. We have

$$
Z_m(k+1) - \mathbf{H}\mathbf{X}_m(k+1|k) < 0
$$

Case 3: Both BS m and BS 1 are NLOS BS's.

When the three different cases are not met, both BS m and BS 1 are considered LOS BS's. From the results of case determination, the value of $\hat{\sigma}_u^2$ can be assigned by using the following rules:

$$
\hat{\sigma}_u = \begin{cases} \alpha\sigma_m, & \text{Case 1 or Case 2,} \\ \sqrt{2}\dfrac{c}{\Lambda}\sigma_m, & \text{Case 3,} \\ \sqrt{2}\sigma_m, & \text{otherwise,} \end{cases}
\tag{17}
$$

where α is experimentally chosen scaling factors, c is the speed of light, Λ is the cluster arrival rate and σ_m is the standard deviation of AWGN measurement noise. If an all-LOS scenario exists, the range difference estimation is constructed by an unbiased Kalman filter; otherwise, the range difference is estimated by using a biased Kalman filter.

4. Range estimation with modified biased Kalman filtering

4.1 Problems in NLOS identification using Kalman filter and sliding window

In data smoothing, the combination of Kalman filtering and sliding window scheme has been considered a possible solution to the shortcomings caused by using polynomial fitting

or periodical interval checking. Problems remain when trying to identify the NLOS-to-LOS channel status transition. In Fig. 5, it can be seen that the range data smoothed by the Kalman filter decrease slower than expected. The phenomenon may cause a time period with identification errors in the proceeding hypothesis test. During the falling edge of transition, earlier values of the smoothed range data may yield standard deviations which will be larger than the threshold in the hypothesis test, as represented in (4). Instead of using the outputs of Kalman filter as the smoothed range data, a modified scheme is considered for directly generating range estimates. The formulation will be discussed in Section 4.3.
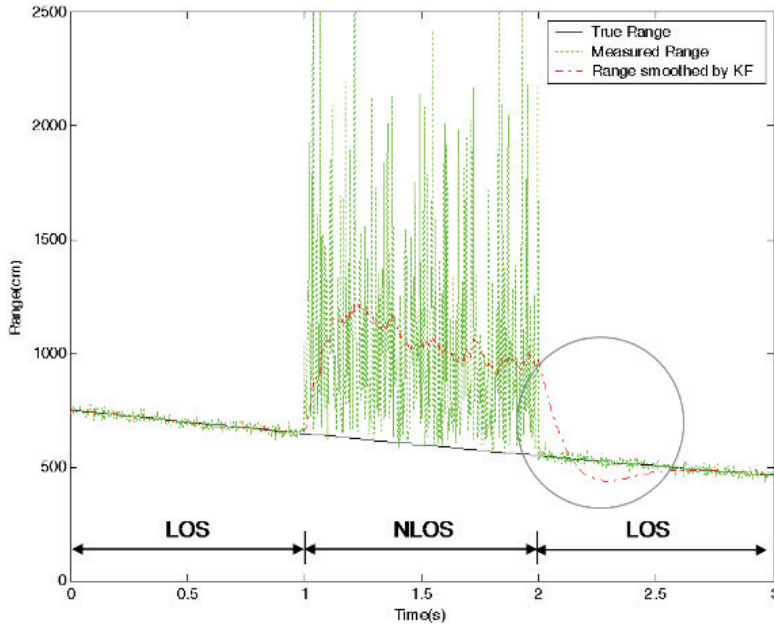
Fig. 5. Slow falling transition of range data smoothed by Kalman filter during the NLOS-to-LOS change of channel status

4.2 Problem in NLOS mitigation using biased Kalman filter

The bias adjusting rule in (15) was used in the hope that the positive NLOS propagation errors be mitigated effectively by increasing the estimated noise standard deviation. The positive NLOS range bias can be reduced by assigning the diagonal elements of noise covariance matrix. However, a resultant side effect is that exceeding negative adjustments may occur in the beginning part of the NLOS status, as shown in Fig. 6. Generally speaking, if an NLOS status is detected and the innovation from the measured data is positive, the measured data may be treated as affected by noise with larger variance and the adjusting rule will achieve the objective of NLOS mitigation in the iteration. In the contrast, if the innovation from the measured data is negative, a possible LOS status would be assumed. The assumption may become incorrect in the next iteration, when the channel is actually in NLOS status. The error would yield an even more negative adjustment in the estimated range value. To tackle the undesirable effect, a new bias adjusting rule is required.
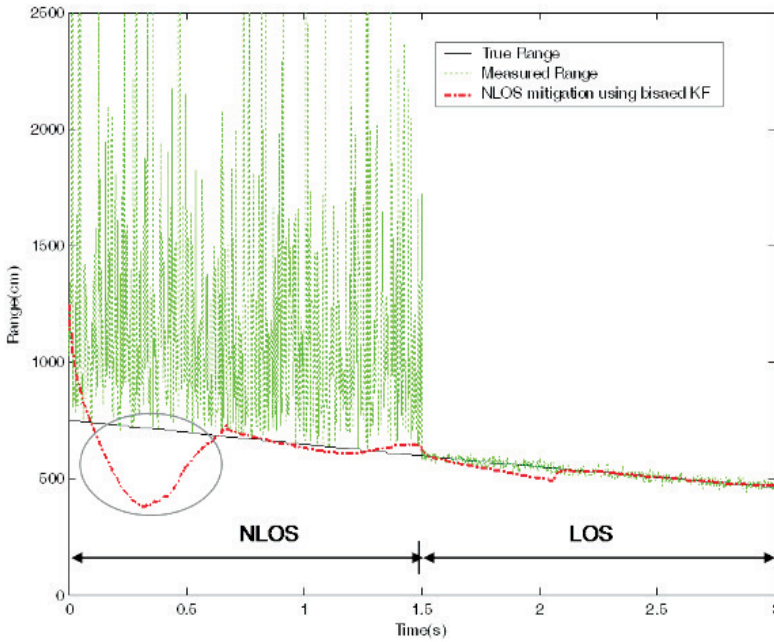
Fig. 6. Exceeding negative adjustments in the beginning part of the NLOS status,

4.3 Functional combination of NLOS identification and mitigation

A novel range estimation scheme with a modified biased Kalman filter for NLOS range mitigation and LOS/NLOS identification is presented and shown in Fig. 7. The modified biased Kalman filter is used to process the range (or TOA) measurement according to the feedback identification result from the processed data in the previous iteration of Kalman filtering. Before computing the Kalman gain in (11), the measurement noise covariance σ_u^2 or the range prediction covariance $P_{1,1}(k \mid k-1)$ must be adjusted by the following adjusting rules.

For NLOS case and $Z(k) - \mathbf{HX}(k \mid k-1) > 0$, let

$$\hat{\sigma}_u^2 = \hat{\sigma}_m^2(k-1) + function(\text{innovation}); \tag{18}$$

For NLOS case and $Z(k) - \mathbf{HX}(k \mid k-1) < 0$, let

$$\hat{\sigma}_u^2(k) = \sigma_m^2, \text{ and} \tag{19}$$

$$P_{1,1}(k \mid k-1) = P_{1,1}(k \mid k-1) + function(\text{innovation}); \tag{20}$$

For LOS case, let

$$\hat{\sigma}_u^2(k) = \sigma_m^2, \tag{21}$$

where $\hat{\sigma}_m^2(k-1)$ is the standard deviation obtained by the sliding window in the previous iteration. By adjusting noise covariance in (18) and (19), the positive NLOS range error can

be significantly reduced. The inclusion of (20) is essential in compensating the range prediction covariance $P_{1,1}(k \mid k-1)$. The biased term avoids inaccurate estimation of the range rate $\dot{L}_m(k)$ from the NLOS mitigation.
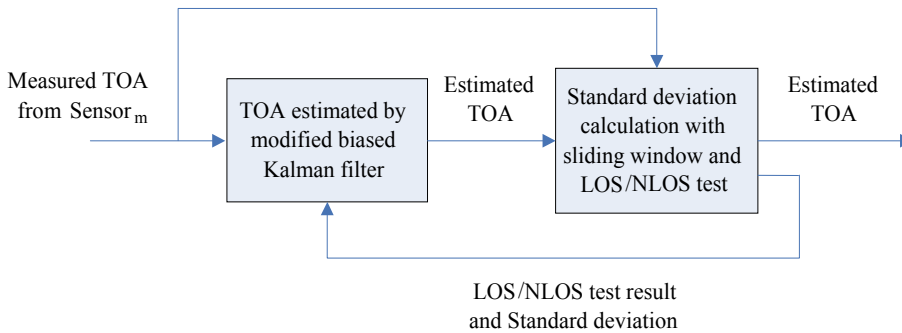
Fig. 7. Range (TOA) mitigation and LOS/NLOS identification

In the range estimation scheme, measured TOA data are first processed by a modified biased Kalman filter. Based on the LOS/NLOS status and the standard deviation feedback from the previous processed data, the modified biased Kalman filter generates an estimated TOA for the current processed data. Under the NLOS situation, the measured TOA's are smoothed by using biased filters, and the positive NLOS errors are mitigated. The standard deviation from the feedback path is regarded as a reference representing the degree of NLOS errors. Under the LOS situation, unbiased smoothing is used for estimating the true TOA value.
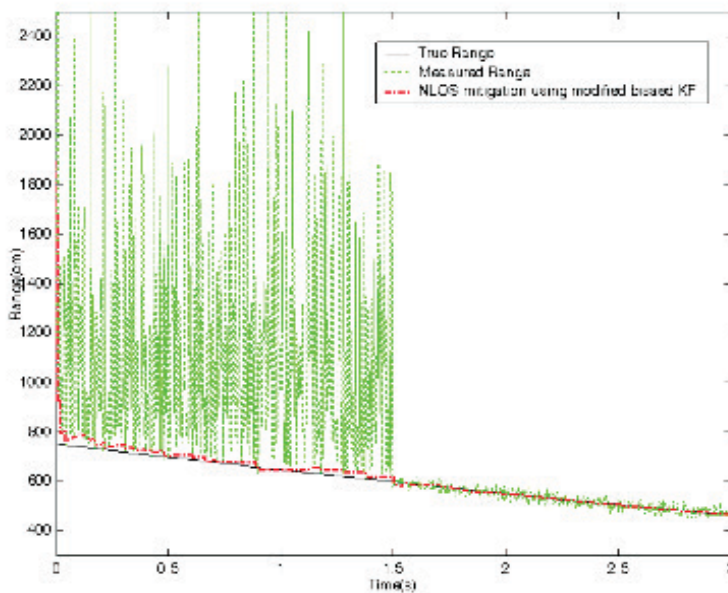
Fig. 8. Results of using the range estimation scheme with a modified biased Kalman filter

In the second functional block in Fig. 7, the standard deviation of the estimated TOA is calculated. With a sliding window, the standard deviation of the last C estimated TOA's is calculated. The obtained standard deviation is passed through an LOS/NLOS hypothesis testing to determine the status of propagation. The resultant LOS/NLOS status and the standard deviation are then used as feedback to the modified biased Kalman filter for data processing in the next cycle. The design objective is that the transition between LOS and NLOS conditions can be detected immediately and the NLOS effects can be effectively mitigated in order to obtain a sequence of estimated TOA's, which are close to the corresponding true range between the mobile and a base station. Fig. 8 shows the results of applying the novel range estimation scheme with a modified biased Kalman filter for NLOS range mitigation and LOS/NLOS identification

For a location system with multiple base stations, the processed TOA data from all participating base stations are then used in positioning and tracking. Formulated TDOA data are also possible if TDOA location system is desired.

5. Hybrid TDOA/AOA indoor positioning and tracking

5.1 Extended Kalman filter for TDOA/AOA positioning

Both time-based and angle-based categories have their own advantages and limitations, it is therefore reasonable to consider hybrid methods to integrate the merits of using the two types of metrics. In (Cong & Zhuang, 2001), a hybrid TDOA/AOA location scheme was presented for wideband code division multiple access (WCDMA) systems. The scheme uses TDOA information from all base stations (BS's) and the AOA information at the serving base station to perform mobile location estimation.

In other mobile location methods, biased versions of the Kalman filter were used in mitigating the NLOS range error. With the rule-determined coefficient for the measurement noise covariance matrix, good location estimation results would be obtained (Le et al., 2003; Thomas et al., 2000). In (Najar & Vidal, 2003), Kalman filtering algorithm with NLOS bias estimation was proposed for UMTS mobile positioning. The estimation of range bias provided performance improvement of location tracking in NLOS environments.

To meet the demand of high location accuracy in indoor positioning applications, the UWB system is considered due to the fine time resolution. The fine resolution of UWB signals provides potentially accurate ranging for indoor location communications, where dense multipath and NLOS errors are the major challenge to the quality of indoor positioning applications. To improve the accuracy of positioning, methods for eliminating or mitigating the effects of NLOS errors and multipath in the UWB environments need to be applied before the TDOA/AOA location technique is used.

For designing suitable NLOS identification and mitigation algorithms for the UWB systems, parameters of the standard UWB channel models provided by the IEEE 802.15.3a standards task group are used in the studies and simulations. In contrast to the scheme in (Cong & Zhuang, 2001), all "good" AOA data along with TDOA information from all BA's are considered in locating the MS position. The AOA and TDOA information are processed centrally by the extended Kalman filter (EKF) for MS positioning and tracking. The architecture of location estimator is illustrated in Fig. 9. For attaining effective NLOS mitigation and obtaining more accurate TOA estimates, the functional blocks in each branch may be replace by the scheme shown in Fig. 7.
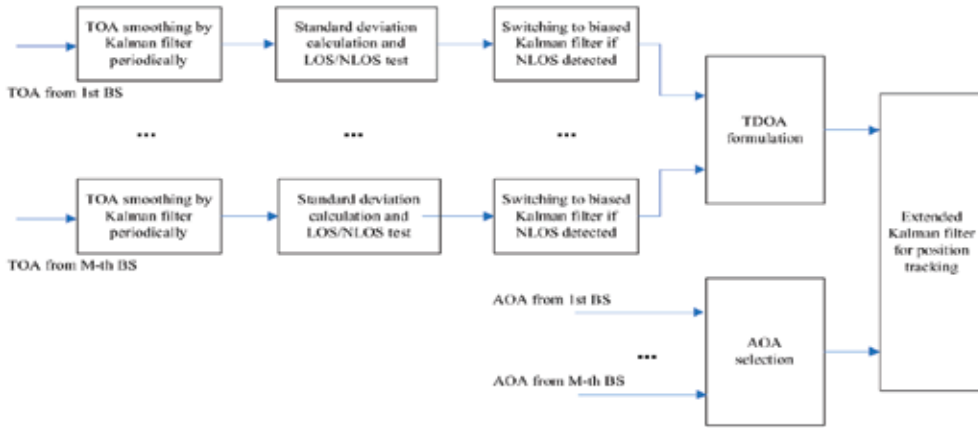
Fig. 9. Hybrid TDOA/AOA positioning and tracking

In Fig. 9, the NLOS error mitigation consists of two parts: the NLOS TOA error mitigation and the AOA information selection. If the LOS TOA propagation scenario is decided, an unbiased Kalman filter is used to smooth the TOA data at each BS. In the contrast, if the NLOS propagation scenario is detected, a biased Kalman filter or modified biased Kalman filter is used in mitigating the NLOS TOA errors. The positive NLOS range bias can be reduced by assigning the diagonal elements of noise covariance matrix. The processed TOA data from all base stations are then used in formulating the TDOA data, which can be further used for mobile positioning and tracking.

The AOA information from all base stations are processed by the AOA selection to avoid introducing large NLOS bearing error into the position tracking stage. Only AOA data from LOS base stations are selected for further processing. In other words, any NLOS AOA data will be discarded.

The formulated TDOA data and the selected AOA data are processed by the extended Kalman filter for the MS location estimation. The state vector of a mobile station is defined as

$$\mathbf{S}(k \mid k-1) = \mathbf{\Phi'S}(k-1 \mid k-1) + \mathbf{W'}(k-1) \tag{22}$$

where $\mathbf{s}(k) = [x(k) \quad y(k) \quad \dot{x}(k) \quad \dot{y}(k)]^{T}$ is the state vector at time instant k. The covariance matrix of the driving noise vector $\mathbf{W'}(k)$ is

$$\mathbf{Q'} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_u^2 & 0 \\ 0 & 0 & 0 & \sigma_u^2 \end{bmatrix}, \tag{23}$$

and the state transition matrix is

$$\mathbf{\Phi'} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{24}$$

In the case where the LOS status exists between the mobile station and all base stations, the TDOA/AOA measurement process can be represented as

$$\mathbf{Z}(k) = f(\mathbf{S}(k)) + \mathbf{U}'(k) ,$$
(25)

where $\mathbf{Z}(k)$ is the measured data vector, $f(\mathbf{S}(k))$ is a nonlinear transformation, and $\mathbf{U}'(k)$ is the measurement noise. The covariance matrix of $\mathbf{U}'(k)$ is

$$\mathbf{R}' = \begin{bmatrix} \mathbf{H}\boldsymbol{\sigma}^2\mathbf{H}^T & \mathbf{0} \\ \mathbf{0} & \sigma_{AOA}^2\mathbf{I} \end{bmatrix}_{(2M-1)\times(2M-1)} ,$$
(26)

in which the TDOA formulation can be written as

$$\mathbf{H} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \cdots & 0 & 1 \end{bmatrix}_{(M-1)\times M} ,$$
(27)

and

$$\boldsymbol{\sigma}^2 = \begin{bmatrix} \sigma_R^2 & 0 & \cdots & 0 \\ 0 & \sigma_R^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_R^2 \end{bmatrix}_{M\times M} ,$$
(28)

where σ_R^2 is the variance of the range related to the output of the unbiased or biased Kalman filter. The AOA variance σ_{AOA}^2 is related to the selected LOS AOA data. The dimension of the matrix $\mathbf{I}$, $m \times m$ is the determined number of LOS base stations from the AOA selection. As shown in Fig. 10, the results from the LOS/NLOS hypothesis are used in selecting the LOS AOA metrics.

When the NLOS situation occurs, the covariance matrices of the processed TDOA and AOA data are different from those in the LOS situation. The dimension of the matrix in measurement process for the NLOS situations will be decreased, and determined by the sum of the number of TDOA and the number of LOS AOA data.

5.2 Simulations results and discussions

The performance of the hybrid TDOA/AOA positioning technique for indoor UWB systems is studied by conducting computer simulations. We assume that three base stations are used in the location system. The coordinates are BS1: (0, 0), BS2: (5m, 8.66m), and BS3: (10m, 0), respectively. The NLOS range error is assumed to be an exponential distribution, which is defined in the standard indoor UWB channel model (Foerster, 2003; Molisch, et al., 2003). A multipath cluster arrival rate 0.0667*10^9 for CM3 is used for the case where the distance
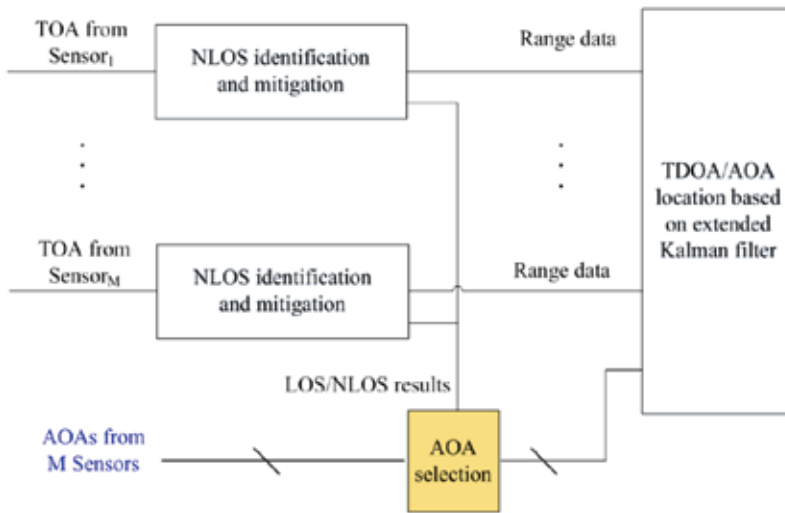
Fig. 10. Hybrid TDOA/AOA positioning and tracking with AOA selection

between any BS and the MS is within the range from *4* to *10* meters. The NLOS bearing is assumed to be uniformly distributed from $-\pi$ to π. It is assumed that an MS travels from location (7m, 4.5m) to (4m, 0.5m) with a constant velocity, *0.5m/s*, as illustrated in Fig. 11. The observed time length is *10s*, and the sample spacing is *25ms*. Two LOS/NLOS propagation scenarios are investigated.

In the first scenario, the LOS propagation between the MS and BS3 turns NLOS at *t = 2s*, and remains NLOS until *t = 10s*. The simulation results in Fig. 12 shows that the hybrid TDOA/AOA positioning scheme with the AOA selection function performs well in terms of root mean square location errors. Without the AOA selection, the NLOS bearing error may lead to severe degradation of position accuracy.
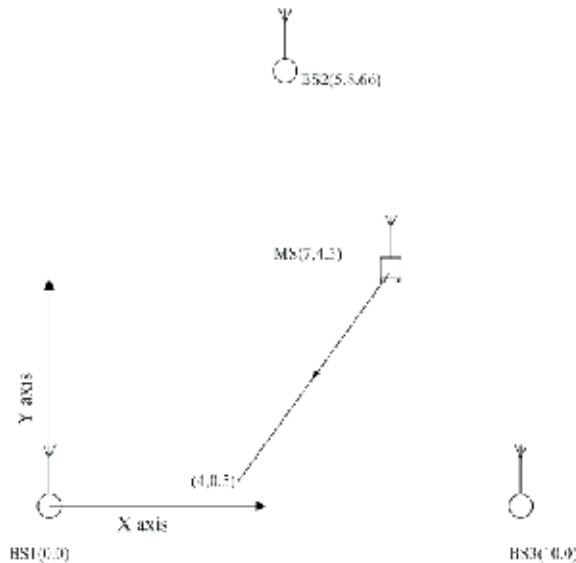
Fig. 11. A simulation example with three BS's and one mobile target
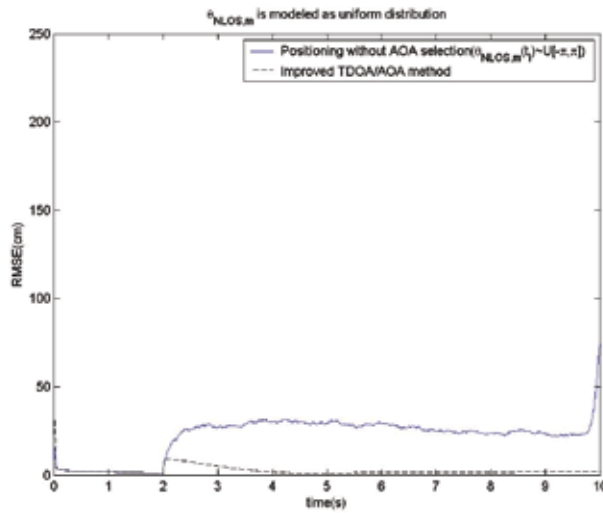
Fig. 12. Performance of positioning with and without AOA selection (Scenario 1: one NLOS BS is assumed)

In the second scenario, the propagation between the MS and two BS's (BS2 and BS3) turn NLOS at $t = 2s$, and remains NLOS until $t = 10s$. In the situation, the data available to the extended Kalman filter will change from three AOA's and two TDOA's to one selected AOA and two TDOAs. The simulation results are shown in Fig. 13. It can be seen that the position errors are a bit larger in this case. The increased errors are mainly due to the larger residual errors at the NLOS TOA mitigation stage and the processing of fewer available AOA data in the adjustable TDOA/AOA-based extended Kalman filter for positioning.
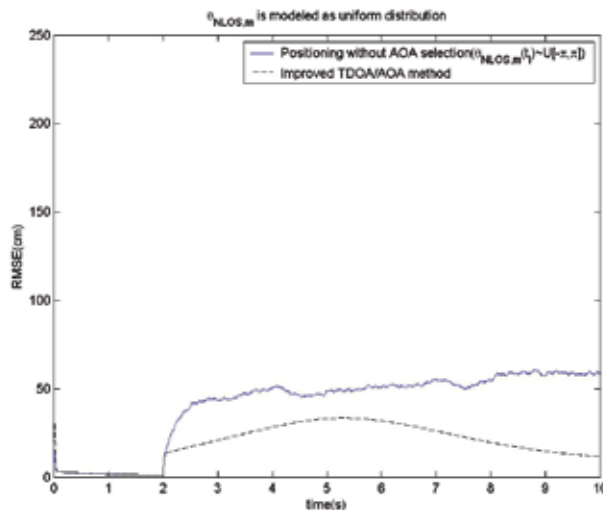
Fig. 13. Performance of positioning with and without AOA selection (Scenario 2: Two NLOS BS's are assumed)

6. Conclusion

We present the applications of Kalman filter and extended Kalman filter on data smoothing, NLOS identification, NLOS mitigation and mobile target tracking in the UWB indoor wireless environments. To improve the accuracy of time-based UWB ranging, data smoothing with Kalman filtering for the NLOS hypothesis testing has been discussed. For the function of NLOS mitigation, biased Kalman filtering is investigated. To tackle the undesirable detection problems of using data smoothing and biased Kalman filtering in range estimation, a novel modified biased Kalman filtering scheme is presented. In the modified biased Kalman filtering scheme, functional combination of NLOS identification and NLOS mitigation is discussed. To investigate mobile target positioning and tracking in the NLOS indoor environments, with a focus on the network-based location system, where multiple base stations are involved in the location estimation, a hybrid TDOA/AOA location system formulated with extended Kalman filters is proposed. Simulation results show the Kalman filter-based scheme is capable of effectively mitigating the NLOS errors and therefore improving the accuracy of target positioning and tracking. Further efforts of NLOS mitigation will lead to improved estimation of signal arrival time and more accurate mobile target positioning and tracking. The integration of multiple Kalman filters and statistical data fusion schemes may also provide a promising solution to the emerging tracking applications using multiple mobile robots and wireless sensor networks in indoor wireless environments.

7. References

Cong, L. & Zhuang W. (2001). Non-Line-of-Sight Error Mitigation in TDOA Mobile Location, *Proceedings of IEEE Global Telecommunications Conference*, Vol. 1, pp. 680-684, San Antonio, TX. USA, Nov. 2001.

Foerster, J. Editor. (2003). Channel Modeling Sub-committee Report Final. *Document IEEE P802.15-02/490r1-SG3a*, IEEE.

Le B. L.; Ahmed K. & Tsuji H. (2003). Mobile Location Estimator with NLOS Mitigation using Kalman Filtering. *Proceedings of IEEE Wireless Communications and Networking Conference*, Vol. 3, pp. 16-20, March 2003.

Mendel, J. M. (1987). *Lessons in Digital Estimation Theory*, Prentice-Hall, Inc., ISBN: 0-534-06660-7, Upper Saddle River, NJ, USA.

Molisch, A. F.; Foerster, J. R. & Pendergrass, M. (2003). Channel Models for Ultrawideband Personal Area Networks. *IEEE Wireless Communications*, Vol. 10, No. 6, December 2003, 14-21, ISSN: 1536-1284.

Najar M. & Vidal J. (2003). Kalman Tracking for Mobile Location in NLOS Situations. *Proceedings of The 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, Vol. 3, pp. 2203-2207, Sept. 2003.

Pahlavan, K.; Krishnamurthy P. & Beneat, J. (1998) Wideband Radio Channel Modeling for Indoor Geolocation Applications. *IEEE Communications Magazine*, Vol. 36, No. 4, April 1998, 60–65.

Saleh, A. & Valenzuela, R. (1987). A Statistical Model for Indoor Multipath Propagation. IEEE Journal on Selected Areas in Communications, Vol. SAC-5, No. 2, Feb. 1987, 128−137, ISSN: 0733-8716.

Thomas, N. J.; Cruickshank, D. G. M. & Laurenson, D. I. (2000). A Robust Location Estimator Architecture with Biased Kalman Filtering of TOA Data for Wireless Systems. *Proceedings of IEEE 6th International Symposium on Spread-Spectrum Techniques and Applications*, pp.296-300, Sept. 2000.

Wylie M. P. & Holtzmann J. (1996). The Non-Line of Sight Problem in Mobile Location Estimation. *Proceedings of IEEE International Conference on Universal Personal Communications*, pp. 827-831, 1996.

Neural Fuzzy Based Indoor Localization by Extending Kalman Filtering with Propagation Channel Modeling

Bing-Fei Wu, Cheng-Lung Jen and Kuei-Chung Chang
National Chiao Tung University
Taiwan

1. Introduction

In this chapter, an indoor localization based on the Received Signal Strength Indication (RSSI) from the Wireless Local Area Network (WLAN) and the Adaptive Neural Fuzzy Inference system (ANFIS) is proposed. Over the last few years, the wireless local area networks based on IEEE802.11b (also named WiFi) and location-based service have been flourishing and increased demand. The wireless networks have become a critical part of the networking infrastructure, and capable for mobile devices equipped with the WLAN to receive the radio signal for networking. Location-aware computing is a recent interesting research issue that exploits the possibilities of modern communication technology due to the location-based service which has a great potential in areas such as library or museum tour-guide, free mobility and nursing at home, patient transporting in the hospital and easy going capability in the shopping mall.

The location of a mobile terminal can be estimated by using the strength of the radio signal with the WLAN. However, the unpredictability of signal propagation through indoor environments is the key challenge in the indoor positioning from the strength of the WLAN signals. It is difficult to provide an adequate likelihood model of signal strength measurements. Thus, the main focus of research in this area have been on the development of the technique that can generate the accurate empirical model from the training data collected in the test area and the real-time recursive estimation for the mobile user.

There are many important propagation models based on the localization techniques. (Bahl & Padmanabhan, 2000; Wassi et al., 2005; Li, B.; Salter & Dempster, 2006) addressed the range based approach, such as pattern matching that relies on the range measurement to compute the position of the unknown node. (Kotanen et al., 2003) offered the design and implementation of a Bluetooth Local Positioning Application (BLPA). First, they converted the received signal power to the distance, according to a theoretical radio propagation model, and then, the Extended Kalman Filter (EKF) is used to compute 3-D position with the basis of distance estimates.

 (Ito & Kawaguchi, 2005) introduced a Bayesian frame work for indoor positioning over the IEEE 802.11 infrastructure, which investigated the direction and distribution of the signal strength for the pre-observation model, and then a location of the mobile device can be computed by using the Bayesian filtering (Fox et al., 2003). (Seshadri, 2005) provided a

probabilistic approach based indoor localization that uses the received signal strength indication as only sensor reading in the Wireless Sensor Networks (WSN), the estimation of location and orientation are computed by a Bayesian filter on the sample set derived using Monte-Carlo sampling. Subject to the power map obtained through the field measurements (Morwlli et al., 2006), the position of a moving user can be estimated and tracked by using the particle filtering that implemented an irregular sampling of a posterior probability space for lower computational power required.

An artificial neural network based work with the supervised learning strategy to reduce the localization error caused by the interference, reflection and other unexpected interruption is presented in (Battiti et al., 2002; Ocana et al., 2005; Ahmad et al., 2006; Ivan & Branka, 2005). During the offline phase, RSSI and the corresponding location coordinates are adopted as the inputs and the targets for the training purpose. The input vector of signal strengths is multiplied by the trained input weight matrix. Thus the location of a mobile device is directly obtained by the outputs of the trained neural network without the detailed knowledge of the access point locations and the specific building characteristics. An indoor localization according to the Maximum Likelihood (ML) algorithm is applied to the measured reference radio map of RSSI (Hatami, 2006), and then an empirical radio map generated form propagation model is compared with the performance of the nearest neighbor method and ML location estimation. (Teuber et al., 2006) proposed a two-stage fuzzy inference system to determine the locations on the basis of Signal-to-Noise Ratio (SNR) in the WLAN measurements and Euclidean distance fingerprints.

In this research, a comprehensive study of the wireless indoor localization based on the three kinds of the propagation channel modeling is addressed. Due to the unpredictability of radio propagation such as the interference, reflection and multipath effects, the adaptive neural fuzzy inference system based on the supervised self-learning strategy is proposed to adapt the indoor environment and reduce the erroneous mapping of the physical distances from RSSI values. In addition, an alternative approach such as interpolation is offered to extend the capability of the propagation model for the small, median and large scale environment, which provides the possibility and the flexibility for the adaption of the indoor positioning. The presented techniques can be developed without any hardware factors of the access point and the knowledge of the test environment. Furthermore, a curve fitting based method is used to compare with the ANFIS and the interpolation. Finally, the extended Kalman filtering is used to perform the location estimation of the mobile terminal in the real indoor environment with 4 access points only.

This chapter is organized as follows. Section 2 introduces the problem formulation for the WLAN indoor positioning and an overview of the proposed appraoch. Section 3 develops three kinds of the radio wave propagation channel models based on the curve fitting, interpolation and neural fuzzy based approach, respectively. Section 4 presents a position estimation computed by using the extended Kalman filtering with the basis of the distance estimates. Section 5 provides a description of the experimental evaluation of the proposed indoor positioning techniques and reports the results. Conclusion and the future work are given in Section 6.

2. Problem formulation

The accuracy and coverage provided by Global Positioning Systems (GPS) or cellular systems are limited in indoor environments. The WLAN standard IEEE 802.11, operating in

the 2.4GHz Industrial, Scientific and Medical (ISM) band, has become a critical part of a public space. Since the wireless information access based on the IEEE 802.11 is now widely available, there is a high demand for accurate positioning in wireless networks, including indoor and outdoor environments. The WLAN positioning is favored because of its cost effectiveness and terminal-based service, while the sensing and positioning are performed on mobile devices. In this study, the WLAN signal strength is used to estimate the corresponding physical distances between the access point and terminal device, and then, a location inference technique based on Kalman filtering is proposed for determining the 2-D location of a mobile device user.

 The indoor environment shown in Fig. 1 consists of one corridor, five classrooms and one bathroom with many walls is utilized to demonstrate the proposed wireless indoor localization scheme. The dimension of the experimentation area is 20.6 meters by 37.4 meters with only 4 APs. Each of the access points provides only partial coverage of the environment. These four access points were mounted on the wall with a height 2.7 meters in the classroom, and the localized mobile device was placed at the height 1.2meter. It is clear that the structure of the test environment is not an ideal free space while there were many unpredictable factors such as the propagation interference, reflection, multipath effects, obstacles, radio noise and clustered pedestrian in this area at the same time.
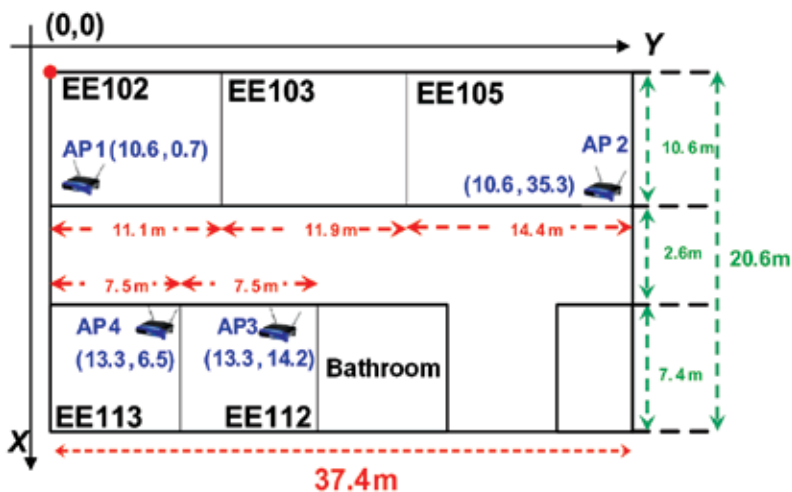
Fig. 1. Map of the experimentation area.

To investigate the characteristics of the WLAN signal propagation, the access points are scanned and the received signal strength indication values are collected. The received signal strength (in the unit of dBm) and noise were obtained by the WLAN adapters in the interval of 2.5 meters, up to about 40 meters. The WLAN client gathered the received signal strength indication value from every access point 200 times. It is observed from Fig. 2, the measurements of WLAN signal of AP3 at the location coordinate (5, 5) is stochastic and complex induced by the radio multipath effect, interference, reflection, and interference.

The histogram and an approximately probability density function of the measurement form AP3 are shown in Fig. 3. It is clear that the distances are converted with the inevasible mapping error from the received signal strength due to the standard deviation of distribution. In particular, it is difficult to avoid the problem under the unpredictability of

signal propagation through the indoor environments. Thus, indeed a propagation channel model with the adaption for the RSSI-physical distances mapping is needed for the accurate evaluation for the mobile device location.
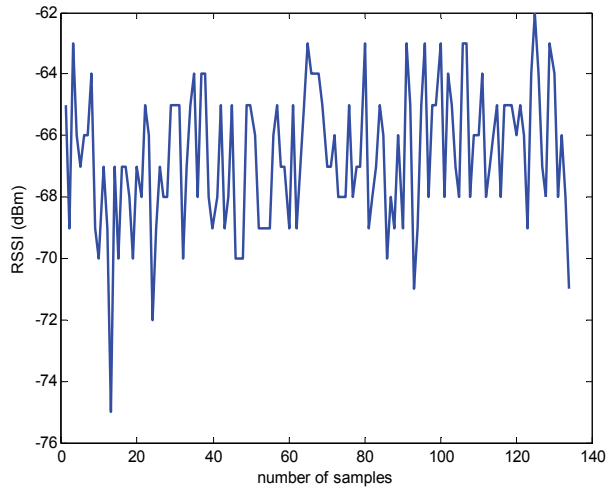

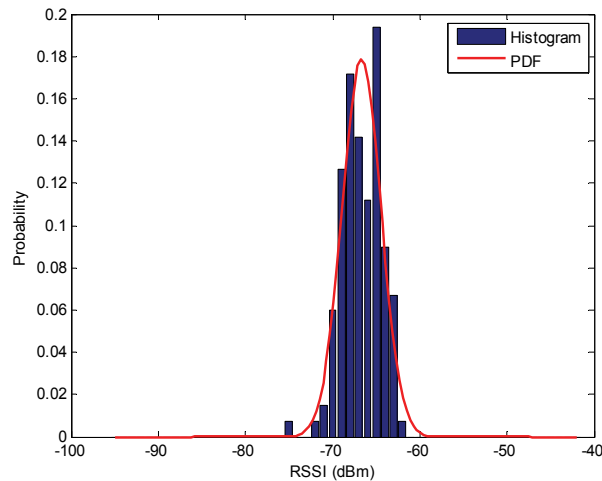Fig. 2. Measurement from the AP3 at room EE102.

Fig. 3. WLAN signal strength distribution for AP3 at room EE102.

The wireless indoor localization scheme proposed herein is shown in Fig. 4, which is divided into two phases, i.e. offline training phase and online estimation phase. The first step, the training phase (also called calibration phase), is the determination of the dependency between the Received Signal Strength (RSS) and the certain location. It is a challenging task in indoors because of the radio interference, multipath, shadowing, non-line-of-sight propagation caused by the environment characteristics such as walls, humans, and other rigid obstalces. In this study, the RSS-position dependency is characterized using the techniqques such as radio propagation model by curve fitting, interpolation, and the neural fuzzy inference system with the training data. The proposed radio propagation

modeling appraoches can be used to capture the combined effects of path loss and non-ideal effects mentioned above relating on the empirical data obtained by a mobile device to the distance from an AP. Then the second step, online estimation phase, which is based on the available distance estimates from the positions localized in the test area, is determined by using the extended Kalman filtering.
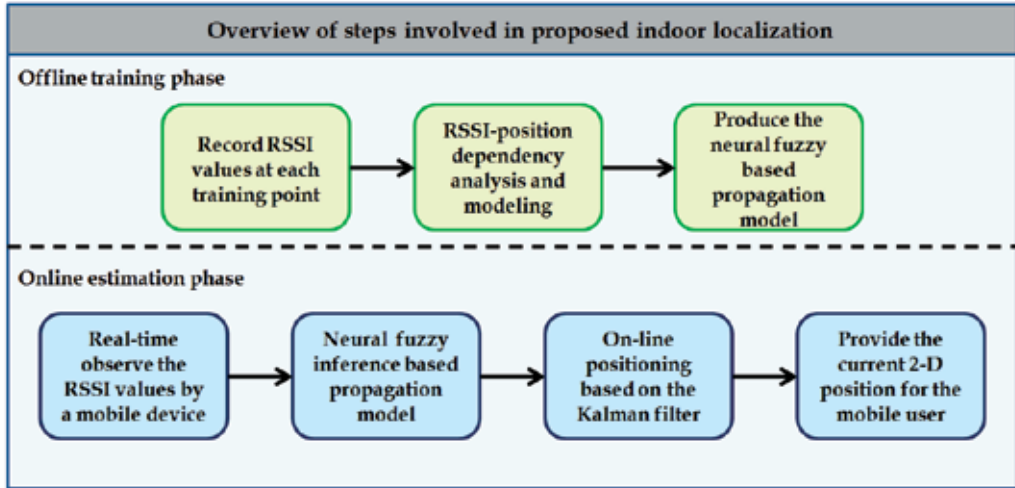

Fig. 4. Overview of the proposed indoor localization.

3. Indoor propagation channel modeling

The WLAN indoor positioning relies on the knowledge obtained in the training phase. A good training procedure should have two features: an accurate mapping relationship based on a large quantity of the empirical data, and a radio map training procedure that is not too complex. These two objectives, however, are often in conflict. To obtain an accurate mapping relationship requires a large number of samples, resulting in a heavy burden on the training procedure and computing load. To trade off these two objectives, three kinds of methodologies for channel modeling in the training scheme are addressed as bellow.

The path loss is the loss between the transmitting antenna and receiving antenna and can be described as

$$L = P_{TX} - P_{RX} , \qquad (1)$$

where L is the total power loss which is caused by indoor channel effects, P_{RX} and P_{TX} are received signal power and transmitted signal power, respectively, in the unit of dBm.

The RSSI values are converted to the physical distances by using a radio wave propagation model (Rappaport, 2001) in model-based localization typically. The model gives the distance d as

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} + 20\log(\lambda) - 20\log(4\pi) - 10n\log(d) - X_{\alpha} . \qquad (2)$$

In (2), d (meter) is the distance between the transmitter and the receiver. P_{TX} (dBm) is the transmitting power and P_{RX} (dBm) is the power level measured by the receiver of the mobile

device. G_{TX} (dBi) and G_{RX} (dBi) are antenna gains respectively to the transmitter and the receiver. Wavelength is λ (meter) and n denotes influence of walls and other obstacles. Error is also included in the equation by X_α which is a normal random variable with standard deviation α. To reduce the converting error between the path loss and the corresponding distance, subject to the training data, three techniques of propagation modeling, curve fitting, interpolation, adaptive neural fuzzy network inference system, are presented as follow to decrease the false positioning.

3.1 Curve fitting

First, the curve fitting technique is applied based on the polynomial regression (Rao, 2002) with the least-squares error sense to realize the propagation model while the RSSI data exhibiting a nonlinear behavior with physical distances at a mobile device to the access points. The polynomial regression can be used by assuming the following relationship:

$$y_{RSS} = a_0 + a_1 d_x + a_2 d_x^2 + \cdots + a_m d_x^m , \tag{3}$$

where y_{RSS} is the path loss approximation function with respect to the traveled distance, the constants $a_0, \ldots, a_m$ are the parameters for polynomial regression and d_x is the distance between the receiver and the transmitter. If there are N training data points (d_{x_i}, y_{RSS_i}), $i = 1, 2, \ldots, N$, the error e_i for the ith data point is defined by

$$e_i = y_{RSS_i} - (a_0 + a_1 d_{x_i} + a_2 d_{x_i}^2 + \cdots + a_m d_{x_i}^m) . \tag{4}$$

The sum of the squared error is given by

$$S = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} \{y_{RSS_i} - (a_0 + a_1 d_{x_i} + a_2 d_{x_i}^2 + \cdots + a_m d_{x_i}^m)\}^2 . \tag{5}$$

The order of polynomial fit, m, is chosen as 2, 3 and 5, respectively, to illustrate the results of propagation modeling by the curve fitting with the training data. It can be observed from Fig. 5, usually the low-order polynomials are used to obtain a better fit. For $m \geq 4$ or 5, the approximation tends to become ill conditioned and the round-off errors make the solution of coefficients $a_0, \ldots, a_m$ inaccurate.

3.2 Interpolation

The theoretical model (2) is usually adopted to present the ideal relationship between the signal power and distance under known parameters of the transmitter and the receiver. The localization method based on such an ideal model is inaccurate since it is impossible in that the perfect modeling for specific conditions of the indoor environment, e.g. multipath and shadowing. Hence, an interpolation method is proposed to solve the propagation modeling without any unknown parameters by using the training data collected in the offline phase. This approach is employed to remove the unknown parameters in (2) by two Reference Nodes (RN). Define RN_j^i as the jth RN with respect to the ith access point, where $i = 1, \ldots, M$, M is the number of detectable access point, (M=4 in this work) and $j = 1, 2$. RN_j^i can be chosen from the training points and corresponding RSSI values. Therefore, it is clear that
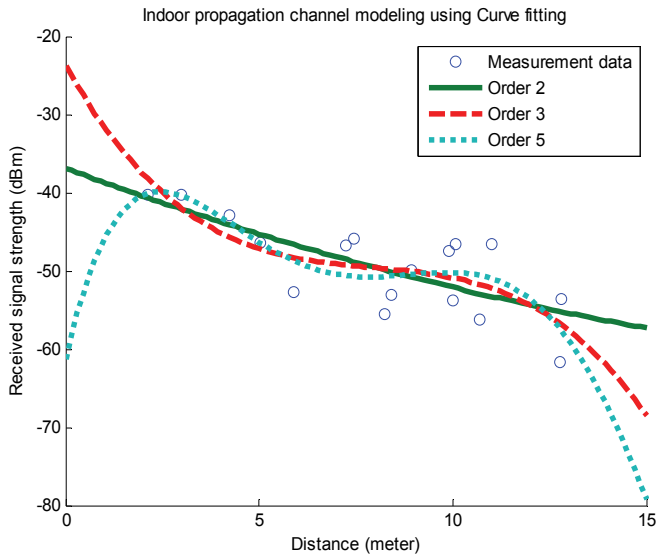
Fig. 5. Propagation modeling based on curve fitting using the polynomial with order 2, 3 and 5.

$$P_{RX1} = P_{TX} + G_{TX} + G_{RX} + 20\log(\lambda) - 20\log(4\pi) - 10n\log(d_1) - X_\alpha , \qquad (6)$$

and

$$P_{RX2} = P_{TX} + G_{TX} + G_{RX} + 20\log(\lambda) - 20\log(4\pi) - 10n\log(d_2) - X_\alpha , \qquad (7)$$

where d_1 and d_2 are two arbitrary reference distance, P_{RX1} and P_{RX2} are corresponding received power computed by (2). From (6) and (7), it is not hard to derive the interpolation model by $(P_{RX} - P_{RX2})/(P_{RX1} - P_{RX2})$ such that

$$P_{RX} = (P_{RX1} - P_{RX2})\frac{\log(d) - \log(d_2)}{\log(d_1) - \log(d_2)} + P_{RX2} . \qquad (8)$$

Figure 6 gives an illustration for the propagation modeling based on the interpolation method which shares the first reference node (RN_1^i, $i = 1,...,4$) that P_{RX1} equals to -44.9 dBm and d_1 is referred to 2.1 meter and combines another RN. The all points, A to L, are the candidates of RN_j^i for the interpolation method.

In Fig. 7, it is obviously that the presented approach using the different reference nodes and referred RSSI values can be used to provide severe propagation models with distinct characteristics and attenuations for the adaption of the environment. The proposed interpolation methodology could be not only extended to perform a large scale test environment wireless positioning but also realize a converting for RSS-position dependency without any hardware parameter and the existing environment knowledge. Therefore, an optimum modeling may be provided by trying for the combination of different reference nodes to yield a better performance for the indoor localization.
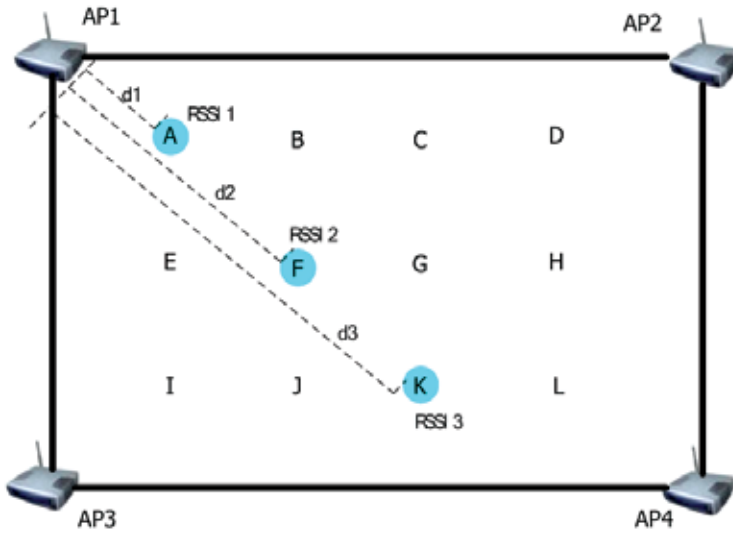
Fig. 6. An example for the interpolation method based modeling with different chosen from *RNs*.
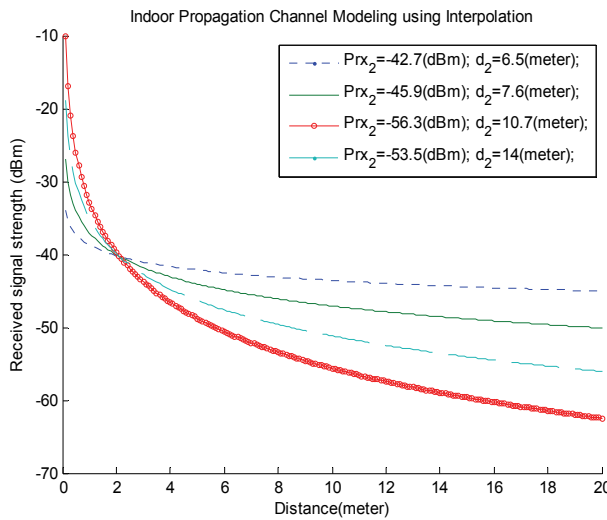

Fig. 7. An Example for interpolation based model by using empirical data collected form AP1.

3.3 Adaptive neural fuzzy inference system

In this subsection, a hybrid intelligent approach for the radio propagation modeling is proposed. We combine the ability of a neural network to learn with the fuzzy logic to reason in order to form an adaptive neural fuzzy inference system (Jang, 1993) with Takagi & Sugeno fuzzy rules whose consequents are linear combinations of their preconditions. A fuzzy inference system is a knowledge representation where each fuzzy rule describes a local behavior of the system. The goal of ANFIS is to find a model or mapping that will correctly associate the inputs (RSSI values) with the target (physical distances). For

simplicity, assume that the system has two inputs x_1 and x_2, and one output y, the rule base contains two TSK fuzzy rules as follows:

$$R^1 : \text{If } x_1 \text{ is } A_1^1 \text{ AND } x_2 \text{ is } A_2^1, \text{ THEN } y \text{ is } f_1 = a_0^1 + a_1^1 x_1 + a_2^1 x_2 ;$$

$$R^2 : \text{If } x_1 \text{ is } A_1^2 \text{ AND } x_2 \text{ is } A_2^2, \text{ THEN } y \text{ is } f_2 = a_0^2 + a_1^2 x_1 + a_2^2 x_2 .$$

The corresponding architecture of ANFIS is shown in Fig. 5, where $\mu_{A_j^i}(x_i)$ is called the Membership Function (MF) for the fuzzy set $A_j^i = \{x, (\mu_{A_j^i}(x)) \,|\, x \in X\}$, for $i = 1,2$ and $j = 1,2$, X is referred to as the universe set. The connected structure shown in Fig. 8, the input and output nodes represent the training values and the predicted values, respectively, the different layers with their meaning are described as bellow:

Layer 1. Every node in this layer acts as an MF, and its output specifies the degree to which the given x_i satisfies the quantifier A_j^i.

Layer 2. Every node in this layer is labeled Π and multiplies the incoming signals $\alpha_j = \mu_{A_1^j}(x_1) \cdot \mu_{A_2^j}(x_2)$ and sends the product out.

Layer 3. Every node in this layer is labeled N calculates the normalized firing strength of a rule such as $\tilde{\alpha}_j = \alpha_j / (\mu_{A_1^j}(x_1) + \mu_{A_2^j}(x_2))$.

Layer 4. Every node in this layer calculates the weighted consequent value $\tilde{\alpha}_j f_j = \tilde{\alpha}_j (a_0^j + a_1^j x_1 + a_2^j x_2)$

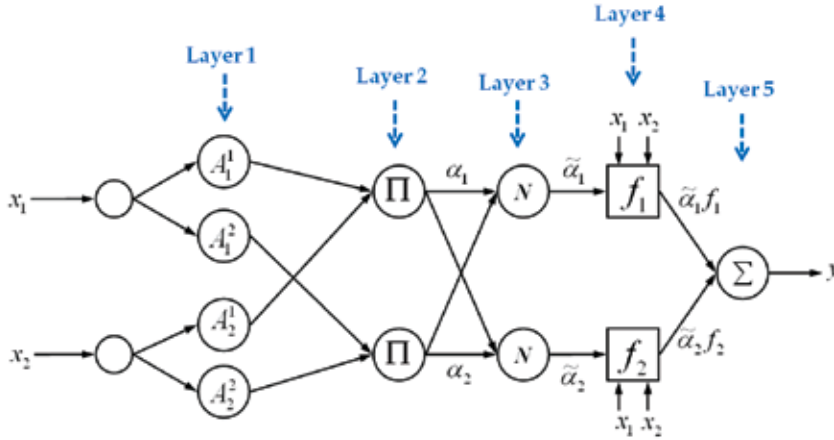Layer 5. This layer sums all incoming signals to obtain the final inferred result.

Fig. 8. An illustration of reasoning mechanism for the ANFIS architecture.

For the given input values $x_1 *$ and $x_2 *$, the inferred output $y*$ is calculated by

$$y* = \frac{\alpha_1 f_1(x_1 *, x_2 *) + \alpha_2 f_2(x_1 *, x_2 *)}{\alpha_1 + \alpha_2} . \tag{9}$$

The learning algorithm for ANFIS is a hybrid algorithm with an error measure $E = \sum_{k=1}^{p} e_k^2$, which is a combination of gradient descent and the least-squares method, where e_k is

$f_k - \hat{f}_k$, f_k and $\hat{f}_k$ are kth desired and estimated output, respectively, and p is the total number of the pairs (inputs-outputs, i.e. RSSI-distances) of data in the training set. An illustration can be observed from Fig. 9, the Gaussian membership function based ANFIS is used to approximate the propagation model according to the training data collected from the received signal strength indication values in the offline training phase. It is clear that a better and reasonable approximation can be provided by lower number of MF whereas the failed mapping is caused by over training that too many MFs and training pairs are applied. The advantages of the neural fuzzy hybrid technique include the sufficient nonlinear regression ability, fast learning from linguistic knowledge, and its adaptation capability. Therefore to reach a good indoor propagation model, the key point is whether the test environment is exactly characterized by the chosen empirical data set with a proper amount and the type of MF.
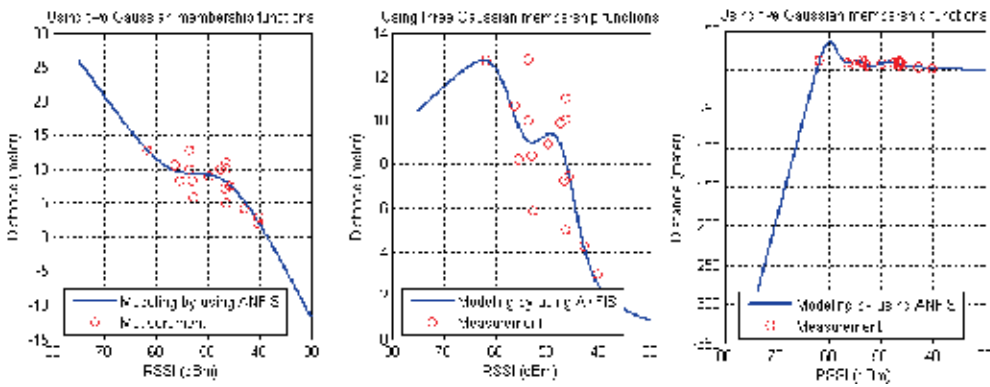

Fig. 9. Propagation modeling by using the ANFIS.

4. Location estimation method

For the online phase, the real-time recursive wireless indoor positioning, the RSS is continuously scanned as a fingerprint or the dependency information on position of the mobile device for the proposed positioning system. Relying on this RSS-position dependency information for wireless positioning, the distance of a mobile device to the access point is provided by the propagation model developed in Section 3, and then, an Bayesian framework estimation, the extended Kalman filter, is used to recursively compute the 2-D mobile device position with the basis of distance estimates. The EKF was selected because it could blend the information optimally with minimizing the variance of the estimation error. A detailed description of the principle of the extended Kalman filter can be found in (Haykin, 2002).

A formulation of the location estimation as a filtering problem in state-space form is addressed here. The general form of the dynamical model is given by

$$x_k = f(x_{k-1}, u_k, w_{k-1}), \tag{10}$$

$$z_k = h(x_k, v_k), \tag{11}$$

where k is the estimation step, u_k is the optional control input, z_k are the output measurements, and x_k is the system state. Further, in (10) and (11), w_{k-1}, v_k are the

discrete white, Gaussian zero-mean, independent state and measurement noise processes, and Q, R are their covariance matrices, respectively.

$$E[w_i w_k^T] = Q_i \delta_{ik} , \tag{12}$$

$$E[v_i v_k^T] = R_i \delta_{ik} , \tag{13}$$

where $\delta(\cdot)$ is the delta function

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i = j \end{cases} . \tag{14}$$

In our case, the state-space is

$$\begin{aligned} x_k &= f(x_{k-1}, 0, w_{k-1}) \\ &= x_{k-1} + w_{k-1} \end{aligned} , \tag{15}$$

$$\begin{aligned} z_k &= h(x_k, v_{k-1}) \\ &= h_d(x_k) + v_{k-1} \end{aligned} , \tag{16}$$

where $h_d(x)$ is the distance between the location of a mobile device user and a access point, which can be written as a function of the device positions:

$$h_{d_i}(x) = \sqrt{\sum_{j=1}^{2} (x_{i,j} - x_j)^2} , \tag{17}$$

where i is the index of the access points, h_{d_i} is the distance between the mobile device user location and the ith access point, $x_{i,j}$ and x_j are the jth coordinate element of the ith access point and mobile device.

Due to the nonlinearity of the mathematical model (17), the EKF is proposed to calculate the mobile device user location $\hat{x}_k$. The algorithm of the EKF can be given as follows:
Time update equations:

$$\hat{x}_k^- = f(x_{k-1}, u_k, 0) \tag{18}$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{19}$$

Measurement update equation:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{20}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \tag{21}$$

$$P_k = (I - K_k H_k) P_k^- \tag{22}$$

where

$$A = \frac{\partial f(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial x}\bigg|_{x=\hat{x}_k} , \qquad (23)$$

$$W = \frac{\partial f(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial w}\bigg|_{x=\hat{x}_k} , \qquad (24)$$

$$H = \frac{\partial h(x_k, 0)}{\partial x}\bigg|_{x=\hat{x}_k} , \qquad (23)$$

$$V = \frac{\partial h(x_k, 0)}{\partial v}\bigg|_{x=\hat{x}_k} , \qquad (26)$$

K is the gain matrix and P is the estimation error covariance. The super minus notation $\hat{x}_k^-$ denotes the a priori state estimate at step k and $\hat{x}_k$ the a posteriori state estimate given measurement z_k.

5. Experimental results

As shown in Fig. 10, the WLAN positioning accuracy of the proposed methods is evaluated using real data from a classroom environment, located on the first floor of EE building at the Chiao Tung University. The dimension of the experimentation area is 37.4 meters by 20.6 meters with a total of 4 access points detectable only, and each one of the detectable access points provides only partial coverage of the environment. The RSSI measurements were scanned and collected from the network card using an application program implemented on the Dopod PDA (200MHz CPU) at the sampling rate of 2 samples/second. The RSS values were collected for 53 training points with a separation of 2.5 meters. For each point, the WLAN client gathered RSSI values from each detectable access point 200 times with a total of 10,600 measurements. As addressed in Section 3, this training set is used to obtain the
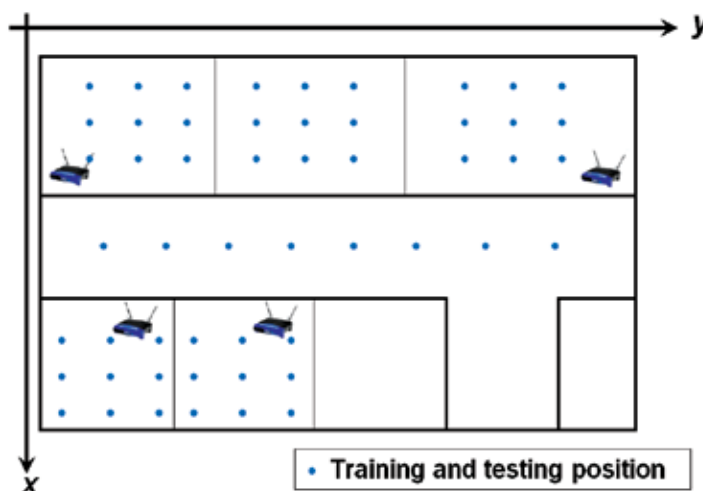

Fig. 10. Experimental area. The blue dots are the training and testing locations.

indoor propagation model in the offline phase, hence the distences of a mobile device to the access points can be estimated for the online phase and recursively compute the position of the mobile device.

The proposed positioning system is evaluated by the real-time estimation at each testing point. The testing data are collected at the same location where the training data are recored. Also, the mobile device gathered the RSSI values from each access point 200 times for the filtering iteration at every testing point. There are four measurement is avaibale for the EKF at each iteration step since only four access points are detectable in this test area. Even the four-demensional measurement state (11) is used, the presented approach is certaintly straightforward and can be implemented by the real-time computing on a PDA.

5.1 Curve fitting

Recall the model developed in the subsection 3.1. The propagation model can be approximated by a curve fitting with nonlinear regression. It is hard to provide the solution of the coefficients of the polynomial fit if the order is high since the high order based nonlinear regression will result an ill conditioned polynomial fit. To omit the trivial test results under the order higher than seven, only four meaningful location error statics are summarized in Fig. 11. On the other hand, the Cumulative Distribution Function (CDF) of the location errors is shown in Fig. 12, it is clear that the best determination of the position is based on first order regression of polynomial fit and achieves improvenments of 7.35 m (70%) over the 3rd order, 5 m (55.6%) over the 5th order, and 2.05 m (30%) over the 7th order at 70th percent. Obviously, the mean error with the first order regression compared to the other three cases are much better not only on the accuracy but also on the precision.

	1st order	3rd order	5th order	7th order
Mean	3.98	10.5	8.98	6.82
STDEV	2.12	5.91	6.34	6.3

Fig. 11. Statistics of the positioning error based on the curve fitting models.

5.2 Interpolation

In this section, the localization scheme using parameter-free propagation models based on the differences in signal attenuations for WLAN signals would be introduced. Due to the individual characteristics of each detectable access point, the propagation models are modified by the combination of the reference nodes. Table. 1 shows the four kinds of choosing reference nodes. However, they might not be the optimal results relate to all possible arrangement of
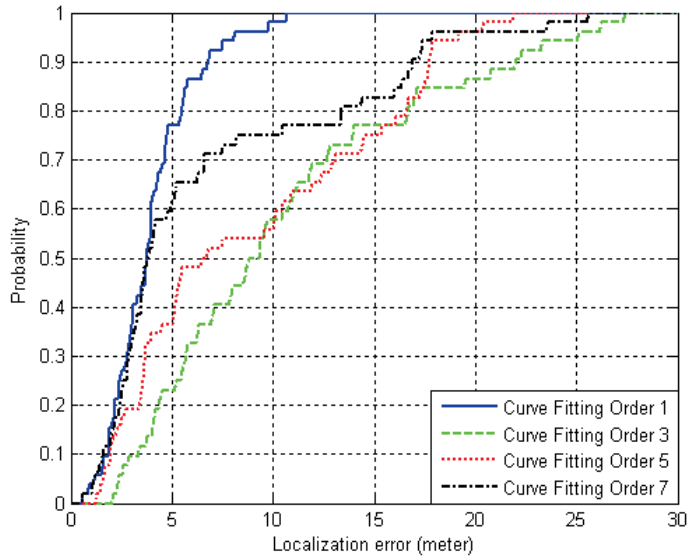
Fig. 12. CDF of the location errors based on the curve fitting models.

RN_j^i while the total number of the possibility of the combination with any two reference nodes from the sampled data is too large to test by trying. In contrast, the empirical data set based interpolation has a high potential to provide a most fit propagation model by associating with some intelligent algorithms.

Positioning case	AP	AP1		AP2		AP3		AP4	
	RN	RN_1^1	RN_2^1	RN_1^2	RN_2^2	RN_1^3	RN_2^3	RN_1^4	RN_2^4
Case1	Distance (m)	5.50	8.54	3.68	18.81	2.37	6.60	3.71	6.54
	RSSI (dBm)	-41.05	-54.61	-37.12	-72.14	-37.76	-60.35	-41.66	-55.66
Case2	Distance (m)	9.24	17.67	5.03	12.39	3.39	5.34	4.28	6.91
	RSSI (dBm)	-50.84	-76.55	-42.30	-66.79	-41.99	-50.65	-41.99	-55.65
Case 3	Distance (m)	6.34	9.94	5.76	19.96	3.52	7.59	7.82	15.51
	RSSI (dBm)	-43.33	-54.2	-40.91	-71.37	-40.00	-54.16	-64.61	-90.00
Case 4	Distance (m)	10.54	19.17	5.80	17.37	5.94	11.51	2.83	8.13
	RSSI (dBm)	-54.09	-74.16	-41.49	-72.22	-49.05	-86.88	-41.38	-61.87

Table 1. Reference nodes and parameters of the interpolation models.

Figure 13 indicates that four better positioning results are picked up to demonstrate the indoor propagation model based on the interpolation with the parameters selected in Table 1. The RN chosen under case 3 presents the best accuracy related to the others due to matching of the real RSS-position dependency for the interpolation model.

The accuracy under case 3 achieves the improvements at least 23.9 percentage over the other cases. It can be seen in Fig. 14, the CDF shows that the reduced location error under case 3 compared to the cases 1, 2 and 4 are 1.93 m (35%), 1.43 m (27.8%) and 2.04 m (36.6%) at 70th percent, respectively. The proposed interpolation modelling technique does not require a known and accurate path loss modeling, reduces the impact of shadowing on location, and is capable of being applied in existing systems without hardware development.

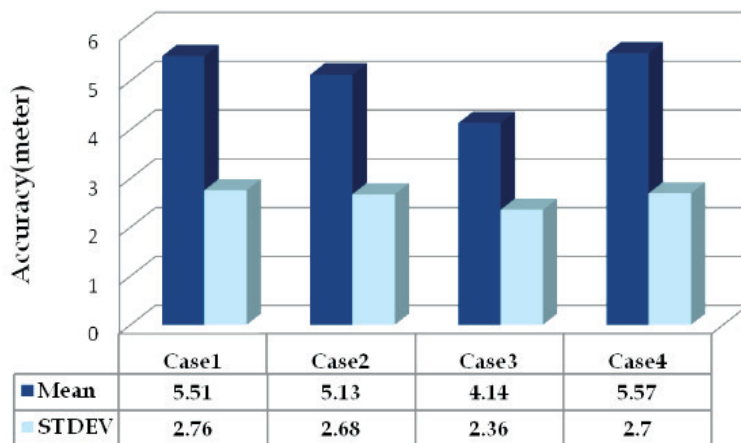|	Case1	Case2	Case3	Case4
■ Mean	5.51	5.13	4.14	5.57
☐ STDEV	2.76	2.68	2.36	2.7

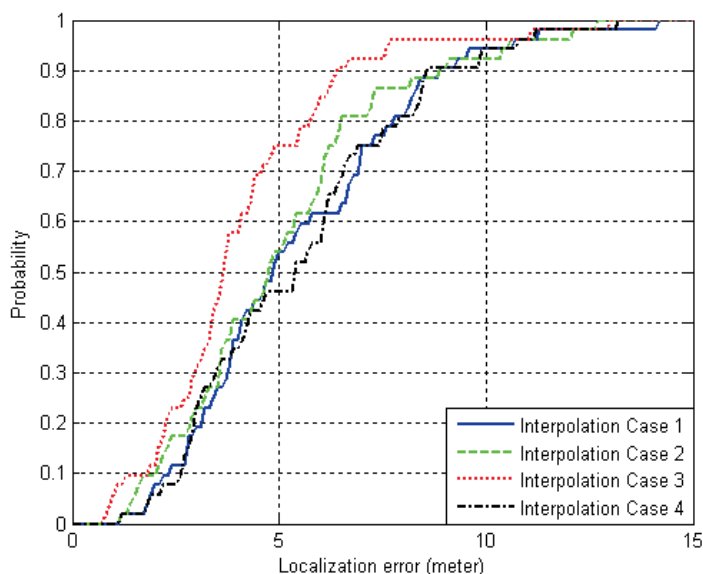Fig. 13. Statistics of the positioning error based on the interpolation models.

Fig. 14. CDF of the location errors based on the interpolation models.

5.3 Adaptive neural fuzzy inference system

During the offline stage, RSS and the corresponding distance between the mobile device and the access point are adopted as the inputs and targets for the learning purpose using the ANFIS, the type and number of the mambership used to train the ANFIS is on the basis of arrangement in Table. 2. There are 53 training pairs consisted of the RSS values and corresponding positions with a total of 10600 sampled data collected from the WLAN positioning area. After training of the ANFIS, the appropriate are obtained. For the online stage, the trained adaptive neural fuzzy inference system acts as a mapping function which converts the RSSI values to the physical distances between the mobile device and the access points in real-time.

Positioning Case	Membership Function Type	Number of Membership Function
1, 2, 3 and 4	Bell-shaped	2, 3, 4 and 5
5, 6, 7 and 8	Triangle	2, 3, 4 and 5
9, 10 ,11 and 12	Trapezoid	2, 3, 4 and 5
13, 14, 15 and 16	Gaussian	2, 3, 4 and 5

Table 2. Positioning cases with different types and the number of MF.

Reducing the complexity of the whole wireless positioning system, the learning results of ANFIS for mapping are summarized in a lookup table which can be used to estimate the physical distances without the loss of accuracy since the ANFIS is optimized under the gradient descent and the least-squares method with the error measure described in the subsection 3.3. The total of 16 cases are tested for verifing the accuracy of the WLAN positioning based on the ANFIS propagation models, which are given in Table. 3. The learning results tend to be over training that diverge the location errors if the the number of the MF is larger than 5, those results are trivial and not mentioned here. It can be seen from Table. 3 that the cases have the best accuracy and precision is the positioning based on the ANFIS models using the bell-shaped MF.

Membership function type	Error evaluation	Localization error (m)				Averaged performance (m)	
		Number of membership function				Mean	STDEV
		2	3	4	5		
Bell-shaped	Mean	3.76	3.57	3.60	3.64	3.64	0.08
	STDEV	2.03	2.07	2.06	2.33	2.12	0.13
Triangle	Mean	6.47	7.82	8.40	4.72	6.85	1.63
	STDEV	6.71	10.96	11.93	2.93	8.13	4.14
Trapezoid	Mean	3.80	3.68	3.93	4.22	3.90	0.23
	STDEV	2.02	2.06	2.20	2.60	2.22	0.26
Gaussian	Mean	4.01	3.63	3.70	3.79	3.78	0.16
	STDEV	2.08	2.04	2.06	2.35	2.13	0.14

Table 3. Statistics of the positioning error based on the ANFIS models.

The average performance shown in Table 3 is yielded by averaging the mean location errors with respect to different numbers of MF, and indicates that the results based on Gaussian, bell-shaped and trapezoid type MF are similar because of similar accuracy. Figure 15 shows the CDF for the best location error regard to the each MF type, it can be seen that the positioning based on the Gaussian, bell-shaped and trapezoid MF almost have the same good statistics and achieve the improvements of 31.9% over estimation using triangle MF.

Clearly, there are many benefits of the proposed ANFIS approach for the modelling or the prediction. It combines two techniques, the neural networks and fuzzy systems, by using the fuzzy techniques, both numerical and linguistic knowledge can be combined into a fuzzy rule base, the combined fuzzy rule base represents the knowledge of the network structure so that structure learning techniques can be accomplished easily. Fuzzy membership functions can be tuned optimally by using learning methods, and the architecture

requirements are fewer and simpler compared to neural networks, which require extensive trails and errors for optimization of their architecture. Therefore the accurate indoor propagation approximation can be provided by the proposed ANFIS based method.
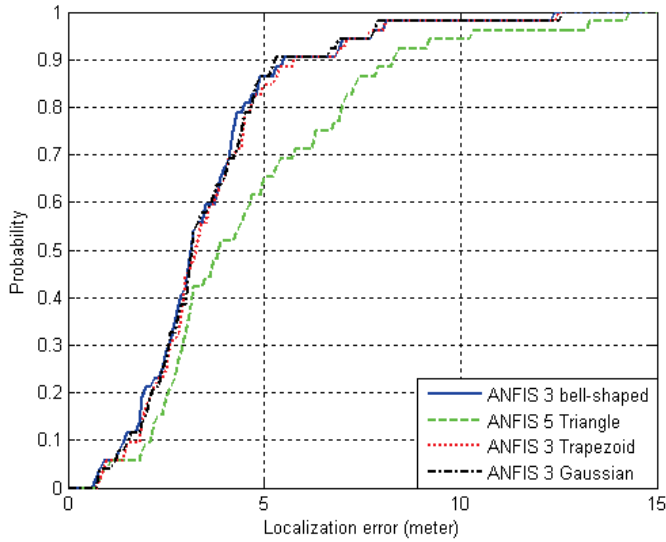

Fig. 15. CDF of the location errors based on the ANFIS models.

5.4 Comparisons

1) Related work: (Bahl & Padmanabhan, 2001) proposed an in-building user location and tracking system - RADAR, which adopts the nearest neighbors in signal-space technique, which is the same as *k*NN positioning based on averaging *k* closest location candidates. They proposed the method based on the empirical measurement of access point signal strength in offline phase. By these experiments, it is reported that user orientations, number of nearest neighbors used, number of data points, and the number of samples in real-time phase would affect the accuracy of location determination. The proposed WLAN positioning is compared with RADAR in case *k*=3,5,7, and 9.

	3NN	5NN	7NN	9NN
■Mean	4.31	4.49	4.54	4.63
STDEV	3.07	3.13	3.25	3.28
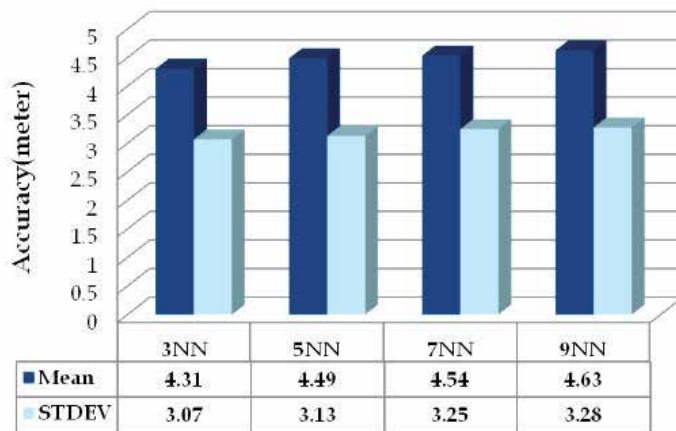
Fig. 16. Statistics of the positioning error based on RADAR.

The RADAR based WLAN positioning is very popular because of its low complexity of algorithm and efficiency. Figure 16 shows the statistics of the location error based on the RADAR, the best accuracy error mean of 4.31 m is provided by using three nearest neighbors while the positioning errors tend to be increasing if more neighbors are referred. This is caused by the referring fault by ill conditioned neighbor candidates which are too far to the real position of the mobile device location.

2) Comparison and Verification of accuracy in a mobile device: In order to ensure fair comparison of the positioning results, the proposed approaches mentioned before and the related work RADAR were implemented on the same simple mobile device, a Dopod PDA (200MHz). To carefully verify the performance of the accomplished experimentation, the best positioning results regard to each approach mentioned above in this chapter are compared hereon to demonstrate the performance of the proposed method. For convenience, a summary of error statistics for each method is provided in Fig. 17.

	ANFIS with three bell-shaped MFs	Interpolation Case 3	Curve Fitting with order 1	RADAR (Bahl et al) with 3NN
■Mean	3.57	4.14	3.98	4.31
STDEV	2.07	2.36	2.12	3.07
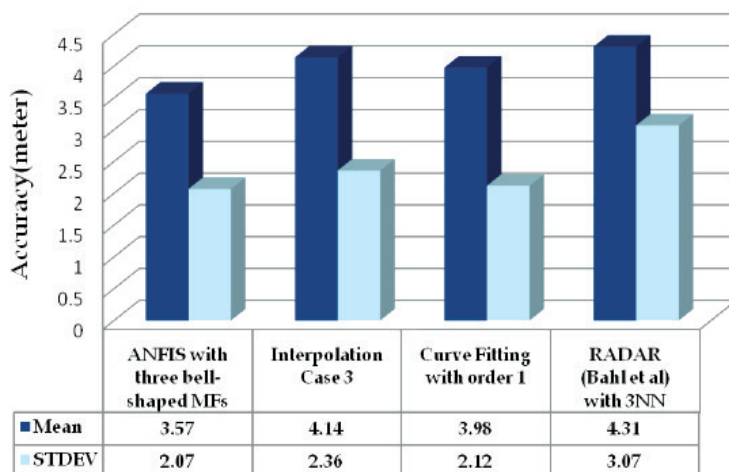
Fig. 17. Statistics of the positioning error for each method.

The improvements of error mean of WLAN positioning based on ANFIS with three bell-shaped MFs compared to the interpolation, curve fitting and RADAR are 0.57m (13.7%), 0.41m (10.3%) and 0.74m (23.2%). The precision of the proposed WLAN localization using the ANFIS based propagation model has a relatively best standard deviation 2.07 m whereas the poorest one is 3.07m provided by the RADAR algorithm, so that ANFIS based appraoch achivies the improvements of 12.2% over the interpolation, 2% over the curve fitting and 32.5% over the RADAR algorithm.

In addition, Fig. 18 shows that the CDF of the location errors for the best cases of each approach. Using ANFIS, interpolation, curve fitting and RADAR, 50% of the location errors are within 3.2 m, 3.7 m, 3.81 m, and 3.65 m, respectively; 90% of the location errors are below 5.52 m, 6.41 m, 6.83 m, and 7.03 m, respectively. It is clear that the CDF of the ANFIS based result is the most left respected to the other approaches. Investigating the proposed WLAN localization by the time response is another point of view for verifing the performance, which is shown in Fig. 19.
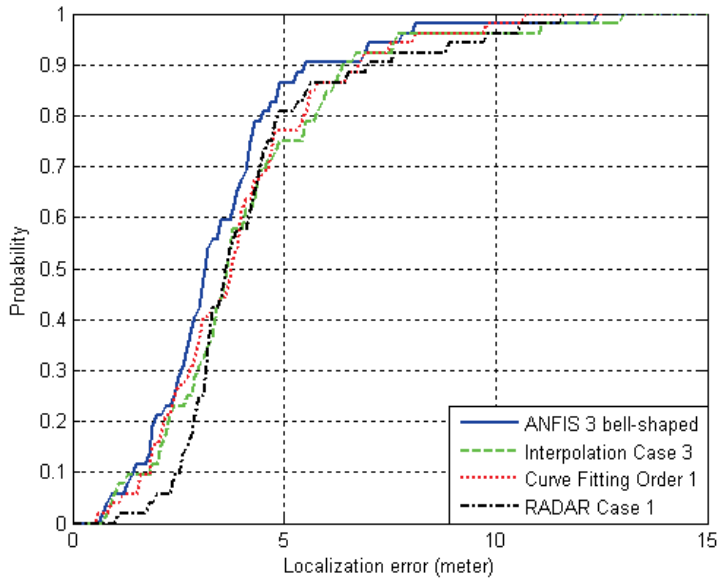
Fig. 18. CDF of the location errors for the best cases of each approach.

Since the structure of the test area is complex due to many walls surrand of the calssrooms, rigid obstacles and clustered human, the noise and shawdoing from the environment are deteriorating the positioning precision, especially the RADAR algorithm is easily to be infuluenced because there is no filtering scheme used to enhance and track the position state by referring previously states. In constrast, the positioning based on the proposed ANFIS propagation model with the well adaptation capability has a fast convergence and tend to be the best accuracy after about 20 transition iterations.
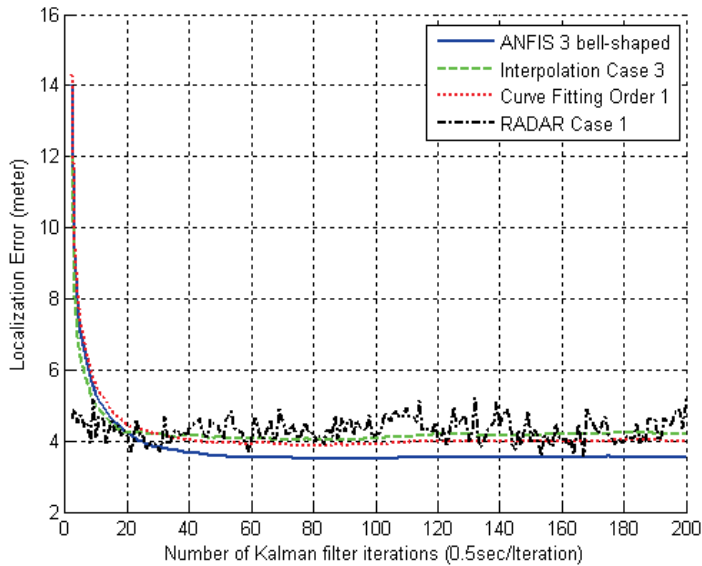
Fig. 19. Time response of error mean for the best case of each method.

6. Conclusions

In this study, a low complexity WLAN indoor positioning system based on the curve fitting, the interpolation, and an adaptive neural fuzzy system is addressed. The proposed modeling techniques are used to reduce the false distance estimation of a mobile device to the access points induced by the unpredictable interference, reflection and multipath effects. It is clear that the experimental results show that a empirical propagation model can be easily provided by the curve fitting based approach but involves a limited flexibility and adaptation; the interpolation based approach has a simple computation for the distance estimation and a high flexibility due to the possible arragement of the reference nodes but requires horrible trails and errors for optimization. Therefore a hybrid learning algorithm under a combination of gradient descent and the least-squares method based neural fuzzy inference system is proposed to solve the problems mentioned above. Due to its nonlinear ability, fast learning capacity and adaptation capability, the accurate distance estimates of a mobile device to the access points can be obtained. The distances are firstly converted by the addressed ANFIS propagation model during the the online stage in real-time with the currently observed RSSI value from the tested WLAN area. The available distances from the positions localized in the test bed are recursively estimated by an extended Kalman filter while it could blend the information optimally minimizing the variance of the estimation error.

Consequently, the presented approach is verified to provide the accurate localization in the real indoor environment of 20.6m×37.4m with 4 access points only. The addressed positioning algorithm is straightforward and can be implemented by simple mobile devices, such as PDAs or mobile phones, to provide the location-aware services without any hardware cost since WLAN signals are popular sensor sources for indoor environments. There are many potential applications for location positioning and network accessing, such as library or museum tour-guide, free mobility and nursing at home, patient transporting in the hospital and easy going capability in the shopping mall.

7. Acknowledgment

8. References

Bahl, P. & Padmanabhan V. N. (2000). RADAR: An In-Building RF-Based User Location and Tracking System, *Proceedings of IEEE International Conference on Computer Communications*, pp. 775–784, March, 2000, Israel.

Wassi, G. I.; Despins, C. & Grenier, D. (2005). Indoor location using received signal strength of 802.11b access point. *Proceedings of the IEEE International Conference on Electrical and Computer Engineering*, pp. 1367-1370, May, 2005, Canadian.

Li, B.; Salter, J. & Dempster, A. G. (2006). Indoor positioning techniques based on Wireless LAN. *Proceedings of IEEE International Conference on Wireless Broadband & Ultra Wideband Communications*, pp. 13 – 16, March, 2006, Australia.

Kotanen, A.; Hännikäinen, M. & Leppäkoski, H. (2003). Experiments on local positioning with Bluetooth. *Proceedings of International Conference on Information Technology: Coding and Computing*, pp. 28 – 30, April, 2003, Las Vegas.

Kotanen, A.; Hännikäinen, M. & Leppäkoski, H. (2003). Positioning with IEEE 802.11b wireless LAN. *Proceedings of IEEE International Symposium on Persona1, indoor and Mobile Radio Communications*, pp2218-2222, September, 2003, China.

Ito, S. & Kawaguchi, N., (2005). Bayesian based location estimation system using wireless LAN. P*roceedings of IEEE International Conference on Pervasive Computing and Communications,* pp. 273 – 278, March, 2005, Hawaii.

Fox, V.; Hightower, J.; Lin, L.; Schulz, D. & Borriello, G. (2003). Bayesian filtering for location estimation. *IEEE Transactions on Pervasive Computing*, Vol. 2, No. 3, pp. 24 – 33.

Seshadri, V.; Záruba, G. V. & Huberpp M. (2005). A Bayesian sampling approach to in-door localization of wireless devices using received signal strength indication. *Proceedings of IEEE International Conference on Pervasive Computing and Communications*, pp. 75 – 84, March, 2005, Hawaii.

Morwlli, C.; Nicoli, M.; Rampa, V.; Spagnolini, U. & Alippi, C. (2006). Particle filters for rss-based localization in wireless sensor networks: an experimental study. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 957 – 960, May, 2006, France.

Battiti, R.; Nhat, T. L. & Villani, A. (2002). *Location-Aware Computing - A Neural Network Model For Determining Location In Wireless Lans*. The University of Trento, Technical Report: DIT-02-0083, 2002.

Ocana, M.; Bergasa, L. M.; Sotelo, M. A.; Nuevo, J. & Flores, R. (2005). Indoor Robot Localization System Using WiFi Signal Measure and Minimizing Calibration Effort. *Proceedings. of IEEE International Symposium on Industrial Electronics*, pp. 1545 – 1550, June, 2005, Croatia.

Ahmad,U.; Gavrilov, A.; Nasir, U.; Iqbal, M.; Seong, J. C. & Sungyoung, L. (2006). In-building Localization using Neural Networks. *Proceedings of IEEE International Conference on Engineering of Intelligent Systems*, pp. 1 – 6, April, 2006, Pakistan.

Ivan, V. & Branka, Z. C. (2005). WLAN location determination model based on the artificial neural networks. *Proceedings of IEEE International Symposium on Multimedia Systems and Applications*, pp.287 – 290, June, 2005, Croatia.

Hatami, A. (2006). Application of channel modeling for indoor localization using TOA and RSS. Ph.D. dissertation, Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, England, 2006.

Teuber, A.; Eissfeller B. & Pany, T. A. (2006). Two-Stage Fuzzy Logic Approach for Wireless LAN Indoor Positioning. *Proceedings of IEEE International Conference on Position, Location, And Navigation Symposium*, pp. 730 – 738, April, 2006, California.

Rappaport T. S. (2001). *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall, ISBN 013-0422-32-0, 2001, New Jersey.

Jang, J. S. (1993). Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, 1993.

Haykin, S. (2002). *Adaptive Filter Theory*, fourth edition, Prentice Hall, ISBN 0-13-090126-1, New Jersey.

Rao, S. S. (2002). *Applied Numerical Methods for Engineers and Scientists*, Prentice Hall, ISBN 0-13-0879480-X, New Jersey.

Application of Kalman Filters for the Fault Diagnoses of Aircraft Engine

Wei Xue and Ying qing Guo
Northwestern Polythechnical University
China

1. Introduction

Fault detection and isolation (FDI) logic plays a crucial role in enhancing the safety and reliability, and reducing the operating cost of aircraft propulsion systems. However, it is a challenging problem achieving the FDI task with high reliability. For this purpose, various approaches have been proposed in the literature.

In an on-line engine fault diagnoses, two tasks may use Kalman filter to carry out: 1) evaluation of on-line engine state variables to renew the on-board model; 2) diagnoses of on-line aircraft engine sensor/actuator fault. How to solve the above problems through application of Kalman filter is discussed in this paper.

A challenge in developing an on-line fault detection algorithm is making it adaptive to engine health degradation. If the algorithm has no adaptation capability, it will eventually lose its diagnostic effectiveness. To address this problem, the integration of on-line diagnostic algorithms was investigated. The Kalman filter estimates engine health condition over the course of engine's life. Then the on-board model could be re-constructer based on the estimated values of Kalman filter.

After all of the above, A Robust Kalman filter and a bank of Kalman filters are applied in fault detection and isolation (FDI) of sensor and actuator for aircraft gas turbine engine. A bank of Kalman filters are used to detect and isolate sensor fault, each of Kalman filter is designed based on a specific hypothesis for detecting a specific sensor fault. In the event that a fault does occur, all filters except the one using the correct hypothesis will produce large estimation errors, from which a specific fault is isolated. When the Kalman filter is used, failures in the sensors and actuators affect the characteristics of the residual signals of the Kalman filter. While a Robust Kalman filter is used, the decision statistics changes regardless the faults in the sensors or in the actuators, because it is sensitive to sensor fault and insensitive to actuator fault.

W. C. Merrill, J. C. Delaat, and W. M. Bruton used a bank of Kalman filters for aircraft engine sensor FDI. This study successfully improved control loop tolerance to sensor failures, which were considered the most likely engine failures to happen under the harsh operating environment. In this study, actuator failure was not considered. In the study done by T. Kobayashi and D. Simon, a fault detection and isolation (FDI) system which utilizes a bank of Kalman filters is developed for aircraft engine sensor and actuator FDI in conjunction with the detection of component faults. The results indicate that the proposed

FDI system is promising for reliable diagnostics of aircraft engine sensor and actuator. An analytical redundancy-based approach for detecting and isolating sensor, actuator, and component faults in complex dynamical systems, such as aircraft and spacecraft is developed by E. C. Larson, E.B. Jr. Parker, and B. R. Clark. This method has limited applications in practice. A Kalman filter was applied to aircraft sensor and actuator fault diagnosis by C. Hajiyev and F. Caliskan. Two different fault detection algorithms, namely multiple hypotheses testing and neural networks that analyze the sensor residuals generated with an extended Kalman filter (EKF) based on an un-faulted engine model were developed and implemented by R. Randal et al. These two algorithms have complementary performance, which is exploited in a fusion algorithm to enhance the overall detection & classification performance. An observer-based robust sensor fault detection approach was applied to a jet engine simulation by R. J. Patton and J. Chen. This method has limited applications in practice. A Kalman filter was applied for aircraft sensor and actuator fault diagnosis by C. Hajiyev and F. Caliskan.Those approach were based on the faults affected the mean of the Kalman filter innovation sequence. A sensor fault that shifted the mean of the innovation sequence could be detected and isolated. A Roubst Kalman filter was used to distinguish the sensor and actuator faults. But, this method could not used to isolate which actuator is faulty.

In general, in–flight diagnostic systems are designed at a nominal health, or non-degraded condition. This design condition becomes a reference health baseline for diagnostics. Any observed deviations in engine outputs from their reference condition values may indicate the presence of a fault. As the real engine degrades over time, in-flight diagnostic systems may lose their effectiveness. Engine health degradation is a normal aging process that occurs in all aircraft engines due to usage and therefore is not considered as a fault. However, similar to various faults, degradation causes the engine outputs to deviate from their reference condition values. When engine output deviations eventually exceed a certain level, the diagnostic system may misinterpret the health degradation as a fault and consequently generate a false alarm.

One approach to maintaining the effectiveness of in-flight diagnostic algorithms, when applied to degraded engines, is to periodically update or re-design the diagnostic algorithms based on the estimated amount of health degradation. Health degradation can be estimated by trend monitoring systems. Through the update based on the estimated health degradation, the health baseline of an in-flight diagnostic system can be shifted to the degraded engine, and thereby the system is able to effectively diagnose the presence of a fault.

The diagnosis approaches based on Kalman filter is the analysis of the residual signals. When the system operates normally, normalized residual signal in a Kalman filter is a Gaussian white noise with a zero mean and a unit covariance matrix. Faults change the system dynamics by causing surges of drifts of the state vector components, abnormal measurements, sudden shifts in the measurement sensor, and other difficulties such as decrease of instrument accuracy, an increase of background noise, reduction in actuator effectiveness etc., effect the characteristics of the normalized residual signals by changing its white noise nature, displacing its zero mean, and varying unit covariance matrix.

For linear dynamic system with white process and measurement noise, the Kalman filter is known to be an optimal estimator. Kalman filters are largely used in the jet engine community for condition monitoring purpose. At the same time Kalman filter are used in

the turbine engine for sensor fault diagnostics purpose. However this method can not or hardly distinguish the fault between sensor and actuator. A bank of Kalman filters and a robust Kalman filter are used to detect sensor and actuator faults. In addition, a bank of Kalman filters is used to detect which sensor is fault. Such technical are easy to implement in a real-time environment.

In the following sections of this paper, the problem setup for sensor fault diagnostics based on the engine health degradation. The deterioration can be estimated by one Kalman filter. Then the on-board model can be re-constructer based on the estimated values of Kalman filter. At last a bank of Kalman filters is applied in fault detection and isolation (FDI) of sensors for aircraft gas turbine engine. At the same time, we assumed that only one of the sensors will fail at a time, and just only one actuator. Hence, detection and isolation between different actuators is not considered. The mean of the residual signals which from sensor measurements and their estimated values applied to detect and isolate sensor failures. An effective approach previously discussed in literature is to distinguish the sensor and actuator fault during a linear engine simulation.

2. Engine model

The engine model being used for this research is the nonlinear simulation of an advanced military twin-spool turbofan engine. Engine performance deterioration is modeled by adjustments to efficiency or flow coefficient scalars of the following four components: Fan (FAN), Booster (BST), High-Pressure Turbine (HPT), and Low-Pressure Turbine (LPT). These scalars representing the component performance deterioration are the health parameters. The engine state variables, health parameters, actuator, and sensor used in the current research are shown in Table 1.

State variable	Health parameters	Actuators	Sensors
XNL	FAN efficiency	W_{FB}	XNL
XNH	BST efficiency	A_8	XNH
	HPT efficiency		P_{31}
	LPT efficiency		P_6
			T_{45c}

Table 1. State Variables, Health Parameters, Actuators, and Sensors of the Engine Model

The FDI (Fault detection and isolation) logic uses the Kalman filter approach in order to estimate the state variables, health parameters, and engine output values from a given set of sensor measurements and control commands. A linear model under consideration is represented by the following state-space equations:

$$\dot{x} = Ax + Bu + Lh + w$$
$$y = Cx + Du + Mh + v$$

(1)

where the vectors x, h, and u represent the state variables, health parameters and control commands, respectively. y is sensor measurement vector, w and v are the process and sensor noise, respectively, they are both assumed to represent Gaussian white noise. Their covariance matrices:

$$E\left[w(k)\right]=0; E\left[v(k)\right]=0$$
$$E\left[w(k+\tau)w^T(k)\right]=Q\delta(k\tau)$$
$$E\left[v(k+\tau)v^T(k)\right]=R\delta(k\tau)$$

(2)

3. The estimation of health degradation

As shown in Fig.1, the on-board model and tracking filter are important part in the model-based control and diagnostics logic. This part uses two sets of input signals; sensor measurements and actuator position. The degradation of the real engine can be tracked by one Kalman filter based on the input signals. After the estimation of the Kalman filter, the on-board model can be shifted to the vicinity of the degraded engine.
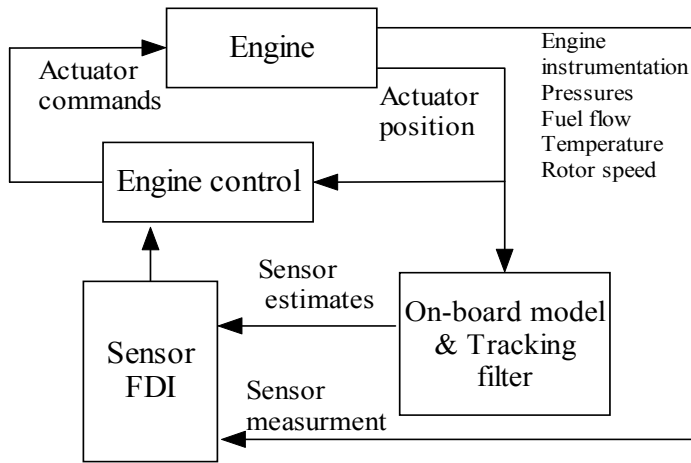
Fig. 1. Model-Based control and diagnostics logic

In the Kalman filter problem setup, the engine state vector is augmented with health parameters as follows:

$$\tilde{x} = \tilde{A}\tilde{x} + \tilde{B}u + w$$
$$y = \tilde{C}\tilde{x} + Du + v$$

(3)

where

$$\tilde{x} = \begin{bmatrix} x \\ h \end{bmatrix}, \tilde{A} = \begin{bmatrix} A & L \\ 0 & 0 \end{bmatrix}, \tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \tilde{C} = \begin{bmatrix} C & M \end{bmatrix}$$

The estimated state vector x_e, the sensor measurements of y_e and the Kalman filter gain matrix can be found with the Kalman filter of the form:

$$\tilde{x}_e = \tilde{A}\tilde{x}_e + \tilde{B}u + K(y - y_e)$$
$$y_e = \tilde{C}\tilde{x}_e + Du$$
$$K = P\tilde{C}^T R^{-1}$$

(4)

where matrix P is the solution of the following steady-state Riccati equation:

$$\tilde{A}P + P\tilde{A}^T - P\tilde{C}^T R^{-1}\tilde{C}P + Q = 0 \qquad (5)$$

4. Fault detection and isolation logic

When a fault occurs, the first step is to detect it as soon as possible. The approach used for model-based fault detection is composed of two steps as follow.
1. Generate residual signals from the sensor measurements and their Kalman filter estimated values.
2. Compare the residuals with thresholds to make fault detection detections.

System noise, measurement noise and modeling uncertainty are key factors that affect detection performance.

A propulsion system with fault detection and isolation logic is shown in Fig. 2. The Kalman filters use two sets of input signals; sensor measurements and control commands. Sensor measurements are corrupted by noise. The difference between them is simply defined as a fault. In this paper, the sensor and actuator failures are "soft failures". Soft failure is defined as inconsistencies between true and measured sensor values that are relatively small in magnitude and thus difficult to detect by a simple range-checking approach.
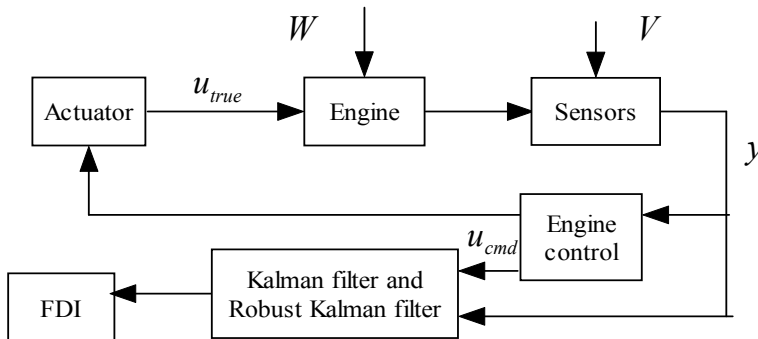
Fig. 2. Fault detection and isolation logic.

4.1 Fault detection algorithm for sensor

In this paper, an approach based on a model with a bank of Kalman filters is used for sensor fault detection and isolation. The sensor and actuator fault are "soft fault". Soft fault is defined as inconsistencies between true and measured sensor values that are relatively small in magnitude and thus difficult to detect by a simple range-checking approach, whereas "hard" fault are larger in magnitude and thus more readily detectable.

Each Kalman filter is designed for a specific sensor fault. In the event that a fault does occur, all filters except the one using the correct hypothesis will produce large estimation errors. By monitoring the residual of each filter, the specific fault that has occurred can be detected and isolated. The structure for sensor FDI using a bank of Kalman filters is shown in Fig. 3. The bank of Kalman filters contains 5 Kalman filters where 5 is the number of sensors being monitored. The control input and a subset of the sensor measurements are fed to each of the 5 Kalman filters. The sensor which is not used by a particular filter is the one being

mentioned by that filter for fault detection. For instance, the ith filter uses the sensor subset that excludes the ith sensor. Hence each Kalman filter estimates the augmented state vector using 4 sensors. Filter #1 uses all sensors except sensor #1, filter#2 uses all sensors except sensors#2, and so on. So, filter #1 is able to estimate the augmented state vector from fault-free sensor measurements, whereas the estimates of the remaining filters are distorted by the fault in sensor #1.

For each filter, the residual vector:

$$e^i = y_e^{\ i} - y^i \qquad (6)$$

When we got the residual, the weighted sum of squares residuals for each of the Kalman filters were calculated as:

$$WSSR^i = V^i \left(e^i \right)^T \left(\Sigma^i \right)^{-1} e^i \qquad (7)$$

where $\Sigma^i = diag \left[\sigma^i \right]^2$. The vector σ is the noise standard deviation, and the additional weigh V^i is the weighting factor.

The statistical function as in (6) has χ^2 distribution consider the following two hypotheses :

H_0: system operates normally,

H_1: fault occurs in the system.

If a confidence probability a is given, the threshold can be found as in. The following gives the detection theory:

$$H_0 : WSSR^i \le \lambda_i$$
$$\qquad\qquad\qquad\qquad (8)$$
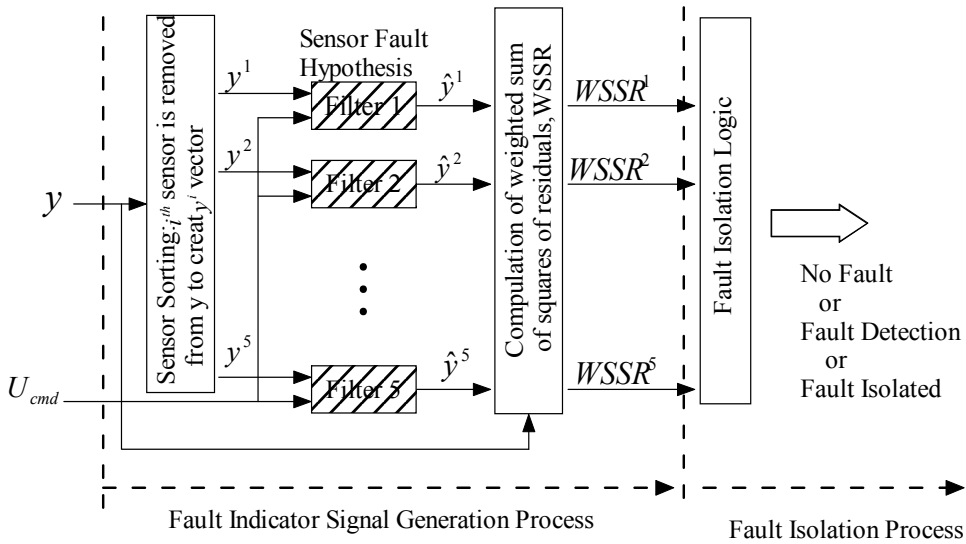$$H_1 : WSSR^i \ge \lambda_i$$

where λ_i is the threshold.

Fig. 3. Sensor fault detection isolation using bank of kalman filters

4.2 Fault detection algorithm for actuator

When a large discrepancy between commanded and true actuator positions does exist due to an actuator fault, it may cause significant errors. A Robust Kalman filter may be designed in order to isolate the sensor and actuator faults. A Kalman filter that satisfies the Dolye-Stein condition is referred to as Robust Kalman filter.

The Doly-Stein condition is expressed as follow.

$$K\left(I + H\phi K\right)^{-1} = B\left(H\phi B\right)^{-1} \tag{9}$$

Here K is Kalman filter gain, I is unit matrix, $\phi = \left(sI - A\right)^{-1}$, A is the system matrix in continuous time, B is the control distribution matrix in continuous time. H is the system measurement matrix. The Kalman filter satisfies the Doly-Stein condition called Robust Kalman filter.

For Kalman filters, $K = P_q C^T R^{-1}$,

With P_q defined by the Riccati equation

$$AP + PA^T - PC^T R^{-1} CP + Q_q = 0$$

As usual we take $Q = Q^T > 0$ and $R = R^T > 0$ with $\left(A, Q^{1/2}\right)$ and $\left(C, A\right)$ stable and observable, respectively. For Kalman filters, they represent given process noise and measurement noise intensities. They are treated more freely as design parameters which we can select to suit broader purposes. In particular, let

$$\begin{aligned} Q_q &= Q_0 + q^2 BVB^T \\ R &= R_0 \end{aligned} \tag{10}$$

Where Q_0 and R_0 are noise intensities matrix for the nominal plant, V is any positive definite symmetric matrix. With these selections, the observer gain for $q = 0$ corresponds to the nominal Kalman filter gain. However, as q approaches infinity, the gains are to satisfy as follow

$$\frac{KRK^T}{q^2} \to BVB^T \tag{11}$$

Solutions of (11) must necessarily be of the form: $\dfrac{1}{q} K \to BV^{1/2}\left(R^{1/2}\right)^{-1}$

Where $V^{1/2}$ and $R^{1/2}$ denote square root of V and R, respectively, i.e.

$$\left(V^{1/2}\right)^T V^{1/2} = V, \quad \left(R^{1/2}\right)^T R^{1/2} = R.$$

Then a Kalman filter satisfying with (9) will be a Robust Kalman filter.

Because of the q factor, the Robust Kalman filter (RKF) is not an optimum filter. The value of the q must be chosen carefully, if q is chosen small the RKF is a Kalman filter and becomes sensitive to actuator failures, on the other hand, if it is chosen large, noise effects increase and unexpected result occur in the RKF.

5. Simulation results 1

The bank of Kalman filters was implemented on the nonlinear dynamical model of an aircraft engine with faults in sensors and the estimation of degraded engine as shown in Fig.4. The nonlinear dynamical model generates five sets of real signals at a given state. The sensor fault can be added on those signals directly. There are five sensors may be fault: High-pressure spool speed (XNH) sensor, Low- pressure spool speed (XNL) sensor, Booster exit pressure (P31) sensor, LPT exit pressure (P6) sensor, LPT inlet temperature (T45c) sensor.

Health degradation can be estimated by trend tracking filter. One Kalman filter was used to estimate the degradation of the real engine. If there were no degradation and no fault, the values of $WSSR1-5$ go to zero, as shown in Fig.5. However, if the HPT efficiency degrades by 2% and the on-board model does not shift to the vicinity of the degraded engine then the in-flight diagnostic systems may lose their effectiveness, as shown in Fig.6. The values of the $WSSR1-5$ grow rapidly and all of them exceed the threshold. It causes a false alarm. This is because shifts in measured engine outputs are induced not only by faults but also by engine degradation. Estimation of the degraded engine is critical to the fault detection and isolation system.

Fig.7 shows that Kalman filter can estimate the degradation accurately. After this the on-board model can be shifted to the vicinity of the degraded engine, and the in-flight diagnostic system may be effective. The Kalman filter estimates engine health condition over the course of engine's life. Based on the estimated health condition, the on-board model is updated. When we add the fault at 10 steps and stop at 200 steps in the LPT inlet temperature measurement sensor and at the same time HPT efficiency degrades by 2%, the in-flight diagnostic system can detect and isolate the fault. As shown in Fig.8, $WSSR1$-$WSSR4$ grow rapidly but the $WSSR5$ remains small. The results indicate that there is a fault in T45c sensor.
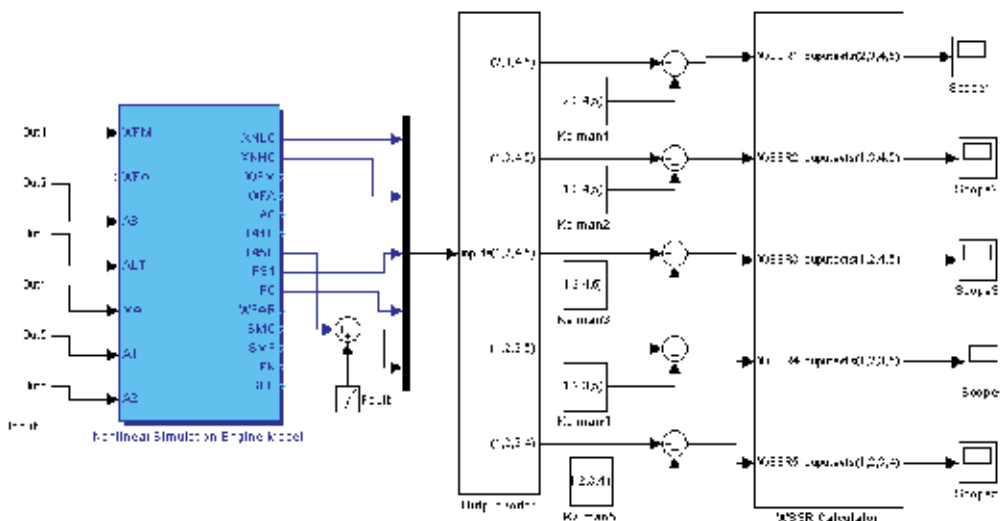

Fig. 4. The simulation architecture of Sensor fault detection isolation using bank of kalman filters
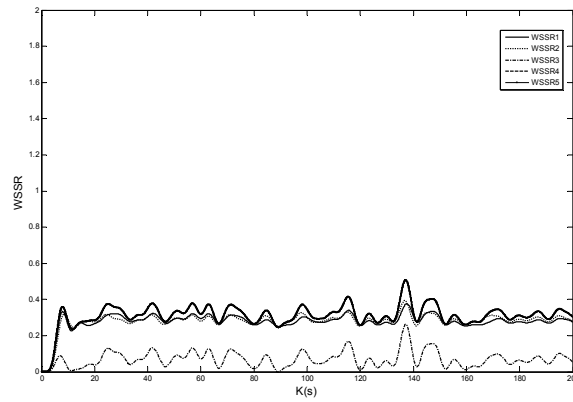
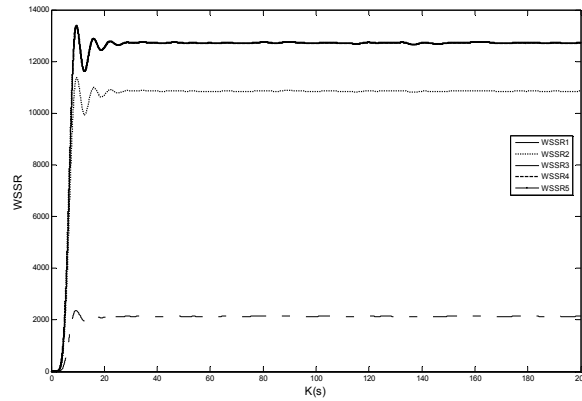Fig. 5. The value of $WSSR1-5$ when no fault and no degradation exist

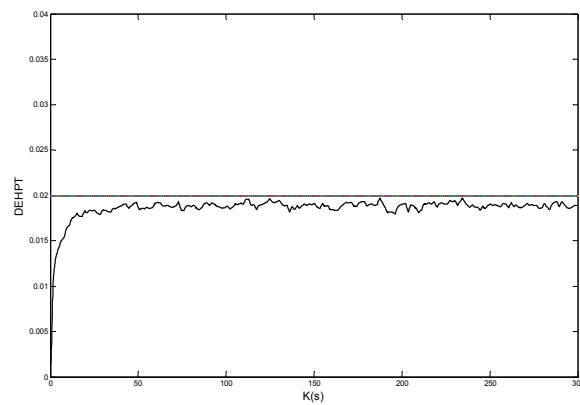Fig. 6. The value of $WSSR1-5$ when the HPT efficiency degrades by 2% and no fault exist.

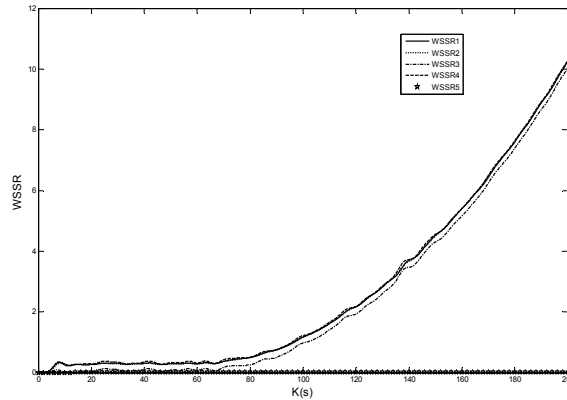Fig. 7. The estimation of Kalman filter when the HPT efficiency degrades by 2%

Fig. 8. The value of *WSSR* -5 when there is a fault in T45c sensor and HPT efficiency degrades by 2%

5. Simulation results 2

The bank of Kalman filters and a Robust Kalman filter (RKF) were implemented on the nonlinear dynamical model of an aircraft with faults in sensors and actuator. The use of the RKF is very useful in the isolation of sensor and actuator as it is insensitive to the latter failures. The RKF was used to isolate whether the detected fault is a sensor fault or an actuator fault, when we add the fault at 20 steps and stop at 200 steps in the low-pressure spool speed measurement sensor, the plot for the RKF estimate is shown in Fig. 10 and when a fault occurs in the sensor, *WSSR* grows rapidly, and after 50 steps it exceeds the threshold. Then, when the fault is in the actuator, the plot for the RKF estimate is similarly shown in Fig. 9. The detection of actuator fault is not possible when the RKF is used. Hence, Fig. 9 and Fig.10 illustrate that the RKF can detect the sensor faults, and cannot detect the actuator faults. On the other hand, if we use Kalman filter (KF) to isolate sensor or actuator fault, the values of *WSSR* are shown in Fig. 11 and Fig. 12. Whatever there is a fault in the sensor or in the actuator, the value of *WSSR* exceeds the threshold. So, KF is sensitive to both sensor and actuator fault and RKF are not sensitive to actuator fault. In this case, RKF and KF should be united to distinguish sensor or actuator fault.
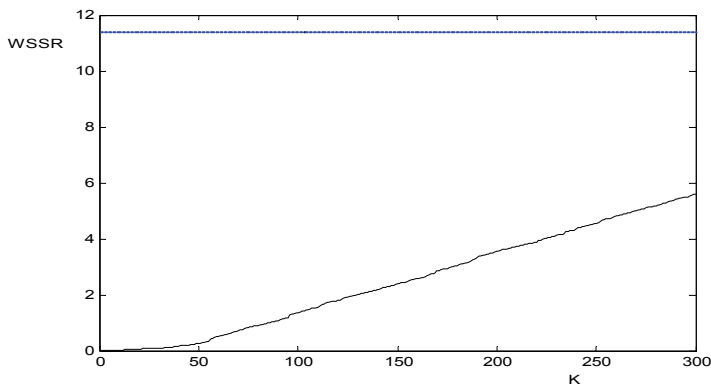

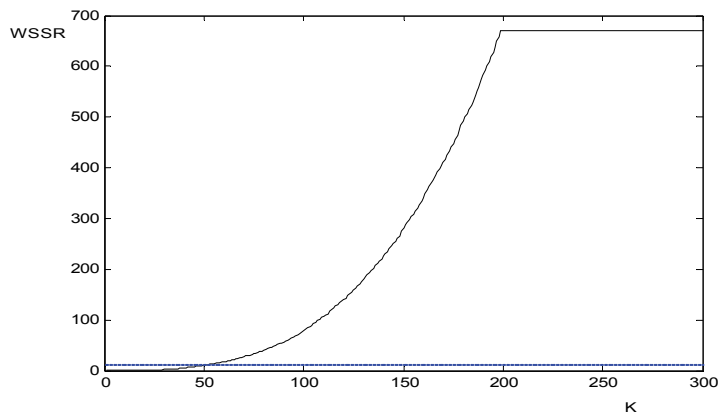Fig. 9. Detection of actuator fault with RKF.

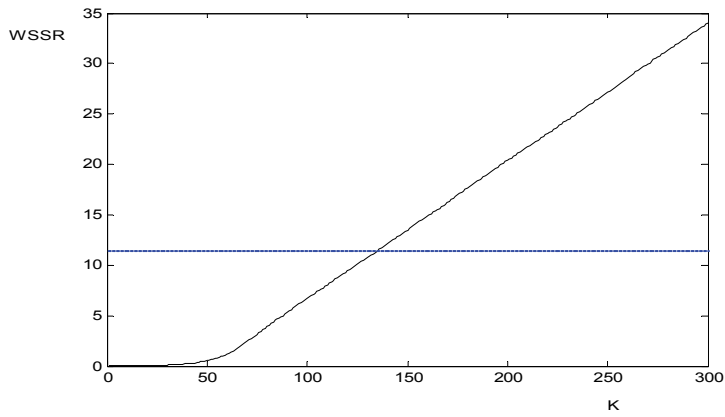Fig. 10. Detection of sensor fault with RKF.

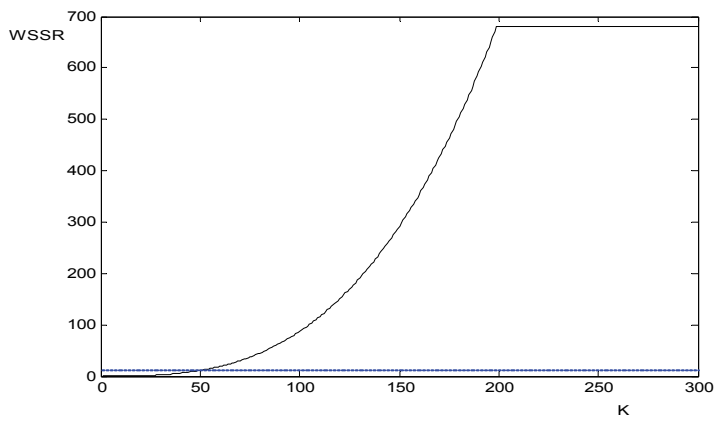Fig. 11. Detection of actuator fault with KF.

Fig. 12. Detection of sensor fault with KF.

In this paper, there are four sensors may be fault, i.e. low-pressure spool speed sensor, high-pressure spool speed sensor, high-pressure compressor exit pressure sensor, low-pressure turbine exit temperature sensor.

When the low-pressure spool speed measurement sensor is faulty, as above mentioned, all filters except for filter 1 will use a corrupted measurement. Filter 1 will be able to estimate the engine outputs from fault-free sensor measurements, whereas the output estimates of the remaining filters (i.e., filters 2, 3 and 4) will be distorted by the fault in sensor 1. The value of *WSSR* and threshold for the 4 Kalman filters are shown in Fig. 13(a)-(d) respectively. The values of *WSSR* for Kalman filter 2, 3 and 4 are also seen to be high whereas the value of *WSSR* for the Kalman filter 1 goes to zero. In this way we can successfully detect which sensor is faulty. The low-pressure spool speed measurement sensor is not used by filter 1. Hence, this sensor is faulty.
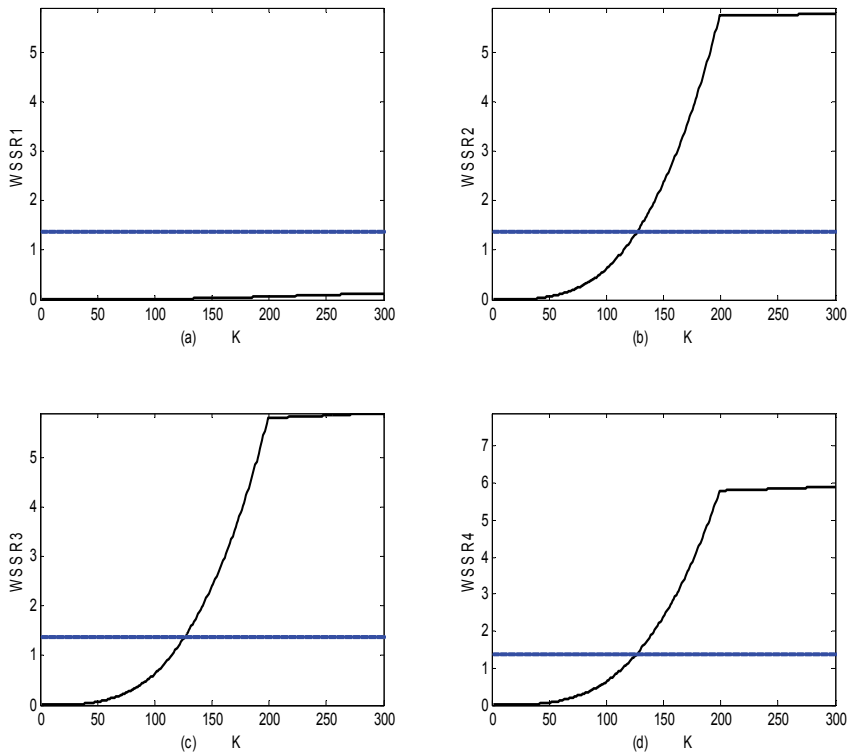

Fig. 13. Fault detection of low-pressure spool speed measurement sensor when a bank of Kalman filters is used.

6. Conclusion

In this paper, aircraft engine sensor fault diagnostics based on the estimation of health degradation was investigated. The tracking filter estimates engine health condition over the course of engine's life. Through this integration, the on-line fault detection algorithm is able to maintain its diagnostic effectiveness as the aircraft engine degrades over its lifetime.

The integrated approach was investigated in a simulation environment using a nonlinear engine model. The evaluation result showed that this approach is essential to maintain on-line fault detection capability in the presence of health degradation.

In this paper, an approach has been proposed to detect and isolate the aircraft sensor and actuator failures occurred in the aircraft control system. A bank of Kalman filters were used to detect and isolate sensor failures, each of Kalman filter is designed based on a specific hypothesis for detecting a specific sensor fault. In the event that a fault does occur, all filters except the one using the correct hypothesis will produce large estimation errors, from which a specific fault is isolated. Failures in the sensors and actuators affect the characteristics of the residual signals of the Kalman filter. When the Kalman filter is used, the decision statistics changes regardless the faults in the sensor or in the actuator. While a Robust Kalman filter is used, it is easy to distinguish the sensor and actuator fault.

7. References

Lakshminarasimha, A. N., Boyce,M. P., and Meher-Homji, C. B. (1994). Modeling and Analysis of Gas Turbine Performance Deterioration. *Journal of Engineering for Gas Turbines and Power*, Vol.116, January 1994, pp46-52.

Diakunchak, I. S. (1992). Performance Detrioration in Industrial Gas Turbines. *Journal of Engineering for Gas Turbines and Power,* Vol. 114, April 1992, pp. 161-168,.

Tabakoff, W., L akshminarasimha, A. N., and Pasin, M. (1990). Simulation of Compressor Performance Deterioration Due to Erosion. *Journal of Turbomachinery*, pp. 78-83, Vol.112, January 1990.

Batcho, P. F., Moller, J.C., Padova, C., and Dunn, M.G. (1987). Interpretation of Gas Turbine Response Due to Dust Ingestion. *Journal of Engineering for Gas Turbines and Power*. Vol.109, July 1987 pp. 344-352.

Richardson, J.H., Sallee, G. P., and Smakula, F. K. (1979). Causes of High Pressure Compressor Detrioration in Service. Paper AIAA-79-1234, AIAA 15th Joint Propulsion Conference, Las Vegas, NV, 1979.

Luppold R H, Roman J R, Gallops G W, Kerr L J.(1989). Estimating in-Flight Performance Variations Using Kalman Filter Concepts. AIAA-89-2584, AIAA/SAE/ASME/ASEE 25th Joint Propulsion Conference, July 10-12 1989, Monterey, CA,1989.

Lambert H.H. A Simulation Study of Turbofan Engine Deterioration Estimation Using Kalman Filter Techniques. NASA-TM-104233.

Kobayashi, T., and Simon, D.L.(2004), Evaluation of an enhanced bank of Kalman filters for in-flight aircraft engine sensor fault diagnostics. NASA/TM_2004 - 213203.

Xue Wei, Guo Ying-qing. (2007). Troubleshooting soft failure of aircraft engine sensor with sequential probability ratio. *Journal of Aerospace Power*, Vol.22, 2007, 1925-1929.

Wei Xue, female, The Han nationality,from Yuncheng, Shan xi province, born in 1981, PhD.candidate, Magoring in aircraft engine health monitring and fault diagnoses.

Application of a Strong Tracking Finite-Difference Extended Kalman Filter to Eye Tracking

Zutao Zhang and Jiashu Zhang
Southwest Jiaotong University
China

1. Introduction

Since eye tracking was first introduced by Mowrer in 1936, it has been gaining in popularity over the past decades as a window into observers' visual and cognitive process. For instance, researchers have utilized eye tracking to study behavior in such domains as driver fatigue detection (Qiang et al., 2004; Horng et al., 2004; Dong et al., 2004), eye typing for helping users with movement disabilities interact with computers (Majaranta & Raiha, 2002), eye tracking analysis of user behavior in WWW search (Laura et al., 2004), using eye tracking techniques to study collaboration on physical tasks for medical research, VR system for measuring inspection methods, and image scanning (Noton & Stark, 1971). Above all applications, two types of human-computer interfaces utilize eye tracking, passive and active interfaces. Passive interfaces monitor the user's eye movements and automatically adapt themselves to the user. For example in driver fatigue detection, the researchers track the driver eyes to fatigue detection, because the human eyes express the most direct reaction when dozing, inattention and yawning. On the other hand, Active interfaces allow users to explicitly control the interface though the use of eye movements. For example, eye typing has users look at keys on a virtual keyboard to type instead of manually depressing keys as on a traditional keyboard (Majaranta & Raiha, 2002; Takehiko et al., 2003). Such active interfaces have been quite effective at helping users with movement disabilities interact with computers. Not surprisingly, eye tracking has attracted the interest of many researchers, and eye trackers have been commercially available for many years (Qiang et al., 2004; Horng et al., 2004; Takehiko et al., 2003; John et al., 2005).
In the past decades, many researchers have paid attention to the eye tracking in human computer interaction. There have been many methods that support non-invasive eye tracking. In (Li et al., 2005), all of these eye tracking algorithms can be classified into two approaches: feature-based and model-based approaches. Feature-based approaches detect and localize image features related to the position of the eye. A commonality among feature-based approaches is that a criteria (e.g., a threshold) is needed to decide when a feature is present or absent. The determination of an appropriate threshold is typically left as a free parameter that is adjusted by the user. The tracked features vary widely across algorithms but most often rely on intensity levels or intensity gradients. For example, in infrared (IR) images created with the dark-pupil technique, an appropriately set intensity threshold can

be used to extract the region corresponding to the pupil. The pupil center can be taken as the geometric center of this identified region. The intensity gradient can be used to detect the limbus in visible spectrum images (Zhu & Yang, 2002) or the pupil contour in infrared spectrum images (Tohno et al., 2002). In (Qiang et al., 2004; Zhu et al., 2002; Ebisawa, 1995; Morimoto & Flickner, 2000), several active IR based eye trackers were proposed. The authors thought that eye tracking based on the active remote IR illuminations is a simple and effective approach. But most of them require distinctive bright pupil effect to work well because they all track the eyes by tracking the bright pupils. Qiang et al. has also made significant improvement of eye tracking over existing techniques (Qiang et al., 2004; Zhu et al., 2002). However, their methods need IR eye detector, or bright pupils and steady illumination. Their eye tracking method using Kalman filtering is linear system estimation algorithm. In realistic driving environments, the eye motion is the high nonlinearity of the likelihood model, the stand Kanlman filter is no longer optimal.

On the other hand, model-based approaches do not explicitly detect features but rather find the best fitting model that is consistent with the image. For example, integrodifferential operators can be used to find the best-fitting circle (Daugman, 1993) or ellipse (Nishino & Nayar, 2004) for the limbus and pupil contour. Michael Chau and Margrit Betke use correlation with an online template to eye tracking in (Michael & Margrit, 2005). The authors (Horng et al., 2004) use the dynamic templates for eye tracking. After finding the eye templates, they are used for eye tracking by template matching. And the minimum value within the search area is the most matching position of the eye. The model- based approach can provide a more precise estimate of the pupil center than a feature-based approach given that a feature-defining criteria is not applied to the image data.

Eye tracking has not reached its full potential even though the general-purpose eye tracking technology has been explored for decades. The first obstacle to integrating these techniques into human-computer interfaces is that they have been too expensive for routine use. Currently, a number of eye trackers are available on the market and their prices range from approximately 5,000 to 40,000 US Dollars (Li et al., 2005). The second factor is that it's very difficult to model to eye tracking because of the eye motion being the high nonlinearity. The third factor is the robustness of eye tracking should be improved because of the variety of head and eyes moving fast, external illuminations interference and realistic lighting conditions. The accuracy of eye tracking can't satisfy the realistic requirement of HCI.

To tackle some of those problems, we propose a strong tracking finite-difference extended Kalman filter algorithm to eye tracking. In this paper, a strong tracking factor is introduced to modify prior covariance matrix to improve the accuracy of the algorithm. And the finite-difference method is presented to calculate partial derivatives of nonlinear functions to eye tracking. At the same time, we overcome the eye tracking modeling in nonlinear system. The last experimental results show that the average correct rate of eye tracking can achieve 99.4% on three video.

The organization of the paper is as follows. Strong Tracking Finite-Difference Extended Kalman Filter algorithm is given in next section. Section 3 gives STFDEKF based Eye Tracking algorithm and experimental results. Final conclusion is in section 4.

2. Strong tracking finite-difference extended Kalman filter

Extended Kalman filter (EKF) is one of the most common and popular filtering approach in nonlinear target tracking and state estimation. It includes state estimation of a nonlinear

dynamic system, parameters estimation for nonlinear system identification and dual estimation where both states and parameters are estimated simultaneously. However, EKF simply linearizes all nonlinear functions to the first order by using the Talyor series expansions. At the same time, EKF may cause more errors for the nonlinear system while estimating system state and its variance. Moreover, the linearization may lead to divergence of filtering process. In a nonlinear mismatched model and limited applications scope, EKF filter will lead the divergence problem of state estimation. For these reasons, two improved EKF algorithms are introduced to tackle some of those problems.

2.1 Suboptimal fading extended Kalman filter

In this section, an adaptive extended Kalman filter - a suboptimal fading extended Kalman filter (SFEKF) is presented. The derivation of the filter is presented (Zhou et al., 1991; Zhou et al., 1990) in detail. SFEKF has the following good properties: 1) lower sensitivity to the statistics of the initial states and the statistics of the system and/or measurement noise, 2) stronger tracking ability to the suddenly changing states and bias no matter whether the filter operates in dynamic or stationary fashion, 3) acceptable computational complexity. Considering a class of nonlinear discrete-time dynamical system,

$$x_{k+1} = f(x_k, u_k, v_k) \tag{1}$$

$$y_k = g(x_k, w_k) \tag{2}$$

where, x_k is the state vector, y_k is the measurement vector, u_k is control input vector, v_k is process noise and w_k is measurement noise. v_k and w_k are statistically independent. The equations of mean and covariance are as follow.

$$
\begin{aligned}
E[v_k] &= q_k, \quad \text{cov}[v_k, v_j] = Q_k \delta(k-j) \\
E[w_k] &= r_k, \quad \text{cov}[w_k, w_j] = R_k \delta(k-j)
\end{aligned}
\tag{3}
$$

The extended Kalman filter is based on the assumption that sensor noises and, propagation errors are driven by zero-mean, Gaussian-distributed, white, random process. Retaining only the first-order terms in the Taylor series expansion, one obtains

$$
\begin{cases}
x_{k+1} \approx f(\hat{x}_k, u_k, q_k) + F_x(k)(x_k - \hat{x}_k) + F_v(k)(v_k - q_k), \\
y_k \approx g(\overline{x}_k, r_k) + G_x(k)(x_k - \hat{x}_k) + G_w(k)(w_k - r_k),
\end{cases}
\tag{4}
$$

where $F_x(k)$ and $F_v(k)$ are the partial derivatives of $f(\bullet)$ to x and v, $G_x(k)$ and $F_w(k)$ are the partial derivatives of $f(\bullet)$ to x and w.

So the suboptimal fading extended Kalman filter (SFEKF) is deduced as follows:
The predicted state estimation equations are

$$\overline{x}_{k+1} = f(\hat{x}_k, u_k, v_k) \tag{5}$$

$$\overline{y}_k = g(\overline{x}_k, r_k) \tag{6}$$

The predicted covariance estimation equations are

$$\bar{P}_{k+1} = \lambda(k+1)F_x(k)\hat{P}_k F_x(k)^T + F_v(k)Q_k F_v(k)^T \qquad (7)$$

Where $\lambda(k+1) \geq 1$ is the suboptimal fading factor, which is used to fade the bypast datum and adjust predictable state estimation covariance matrix.

With this model (Zhou et al., 1993), $\lambda(k+1)$ can be directly determined as follows:

$$\lambda(k+1) = \begin{cases} \lambda_0, \lambda_0 \geq 1; \\ 1, \lambda_0 < 1; \end{cases} \qquad (8)$$

Where

$$\lambda_0 = tr[N(k+1)]/tr[M(k+1)], \qquad (9)$$

$$N(k+1) = V_0(k+1) - G_x(k)F_v(k) \bullet F_v(k)^T - G_w(k)R_k G_w(k)^T, \qquad (10)$$

$$M(k+1) = G_x F_x(k)\hat{P}_k F_x^T(k)G_x^T(k). \qquad (11)$$

$$V_0(k+1) = \frac{1}{k}\sum_{j=1}^{k}\gamma_j\gamma_j^T = \begin{cases} G_x(0)\hat{P}_0 G_x(0) + G_w(0)^T, k=0; \\ \dfrac{\rho V_0(k) + \gamma_j\gamma_j^T}{1+\rho}, k \geq 1 \end{cases} \qquad (12)$$

with $0 \leq \rho \leq 1$ is the preselected forgetting, it may be selected according to the real processes. For fast changing processes, a smaller ρ should be selected, and vice versa. As that pointed out in the paper (Zhou et al., 1993), $\lambda(k+1)$ is insensitive to the value of ρ.

2.2 Strong tracking finite-difference extended Kalman filter

Deriving the ideas in papers (Fan et al., 2006; Zhou et al., 1997), the authors proposed a finite-difference method to replace partial derivatives of nonlinear functions. From further improving the self-covariance and between-covariance, we obtain the algorithm based on strong tracking filter-difference enhanced kalman filter.

We adopt cholesky to decompose Q_k, R_k, $\bar{P}_K$, $\hat{P}_k$,

$$\begin{aligned} Q_k &= S_v S_v^T, R_k = S_w S_W^T, \\ \bar{P}_k &= \bar{S}_x \bar{S}_x^T, P_k = \hat{S}_x \hat{S}_x^T. \end{aligned} \qquad (13)$$

central difference of partial derivative in nonlinear function $Fx(k)$:

$$F_x(k) = \{f_{ij}\} = \left\{ \left(f_i\left(x_{k,j} + \Delta x_{k,j}, u_k, q_{k-} f\left(x_{k,j} - \Delta x_{k,j}, u_k, q_k\right)\right) / 2\Delta x_{k,j} \right) \right\} \qquad (14)$$

Where $\Delta \hat{x}_{k,j} = h\hat{s}_{x,j}, h$ is the step adjustment coefficient; $\hat{s}_{x,j}$ represents the j column of $\hat{S}_x$, then

$$F_x(k)\hat{S}_x = S_{x\bar{x}} = \left\{ \left(f\left(\hat{x}_k + hs_{x,j}, u_k, q_k, \right) - f_i(\hat{x}_k + hs_{x,j}, u_k, q_k) \right) / 2h \right\}. \qquad (15)$$

$$F_v(k)S_v = S_{xv} = \left\{ \left(f_i\left(\hat{x}_k, u_k, q_k, +hs_{v,j} \right) - f_i\left(\hat{x}_k, u_k, q_k - hs_{v,j} \right) \right) / 2h \right\}, \qquad (16)$$

$$G_x\left(k \right)\bar{S}_x = S_{y\bar{x}} = \left\{ \left(g_i\left(\bar{x}_k + h\bar{s}_{x,j,rk} \right) - g_i\left(\bar{x}_k - h\bar{s}_{x,j,rk} \right) \right) / 2h \right\}, \qquad (17)$$

$$G_w\left(k \right)S_w = S_{yw} = \left\{ \left(g_i\left(\bar{x}_{k,rk} + hs_{w,j} - g_i\left(\bar{x}_{k,rk} - hs_{w,j} \right) \right) / 2h \right) \right\} \qquad (18)$$

The predicted covariance matrix, gain matrix and covariance estimate of suboptimal fading extended Kalman filter (SFEKF) are mended as follows

$$\begin{aligned}
\bar{P}_{k+1} &= \lambda(k+1)F_x\left(k \right)\hat{P}_k F_x\left(k \right)^T + F_v\left(k \right)Q_k F_v\left(k \right)^T = \\
&\lambda\left(k+1 \right)F_x\left(k \right)\hat{S}_x \hat{S}_x^T F_x\left(k \right)^T + F_v\left(k \right)\hat{S}_v \hat{S}_v^T F_v\left(k \right)^T = \\
&\lambda\left(k+1 \right)S_{x\hat{x}}S_{x\hat{x}}^T + S_{xv}S_{xv}^T;
\end{aligned} \qquad (19)$$

$$\begin{aligned}
K_{k+1} &= \bar{P}_k G_x\left(k \right)^T \left[G_x\left(k \right)\bar{P}_k G_x\left(k \right)^T + G_w\left(k \right)R_k G_w^T\left(k \right) \right]^{-1} = \\
&\bar{S}_x \bar{S}_x^T \left(S_{y\bar{x}}S_x^{-1} \right)^T \left[S_{y\bar{x}}S_{y\bar{x}}^T + S_{yw}S_{yw}^T \right]^{-1} = \bar{S}_x S_{y\bar{x}}^T \left[S_{y\bar{x}}S_{y\bar{x}}^T + S_{yw}S_{yw}^T \right]^{-1};
\end{aligned} \qquad (20)$$

$$\begin{aligned}
\hat{P}_{k+1} &= \\
&\left[I - K_{k+1}G_x\left(k \right)\bar{P}_{k+1} \right] = \\
&\bar{S}_x \bar{S}_x^T - K_{k+1}G_x\left(k \right)\bar{S}_x \bar{S}_x^T = \\
&\bar{S}_x \bar{S}_x^T - \bar{S}_x \bar{S}_{y\bar{x}}^T K_{k+1}^T - K_{k+1}\bar{S}_{y\bar{x}}^T \bar{S}_x^T + \bar{S}_x \bar{S}_{y\bar{x}}^T K_{k+1}^T = \\
&\bar{S}_x \bar{S}_x^T - \bar{S}_x \bar{S}_{y\bar{x}}^T - K_{k+1}\bar{S}_{y\bar{x}}^T \bar{S}_x^T + \\
&K_{k+1}\bar{S}_{y\bar{x}}\bar{S}_{y\bar{x}}^T K_{k+1}^T + K_{k+1}S_{yw}S_{yw}^T K_{k+1}^T = \\
&\left[\bar{S}_x - K_{k+1}\bar{S}_{y\bar{x}}K_{k+1}S_{yw} \right] \times \left[\bar{S}_x - K_{k+1}\bar{S}_{y\bar{x}}K_{k+1}S_{yw} \right]^T
\end{aligned} \qquad (21)$$

From before-mentioned deduction, we can infer to that all calculations of above include the process noise impact and the error problem of model linearization. The step number which nonlinear function is linearized also changes with last time covariance matrix, process noise and observation noise. The filter becomes very simple because of replacing partial derivatives calculation using finite-difference value. The new strong finite-difference Kalman filter (STFDEKF) has more accuracy and covariance estimation, and improves the

robustness of target tracking. The last experiment results show that the STFDEKF can be used for the high nonlinear stochastic systems such as eye tracking.

3. STFDEKF based eye tracking and results

In this section, we develop the following eye tracking using STFDEKF. Because the eye motion is the high nonlinearity of the likelihood model, it's very difficult to model human eye movement dynamics. In our tracking system, the following nonlinear equations are used to model the eye movement dynamics.

$$x = x_0 + vt + \frac{1}{2}at^2 \tag{22}$$

$$\dot{x}_{k+1} = v_0 + A_k \sin(\omega_k t) \tag{23}$$

$$a_{k+1} = \ddot{x}_{k+1} = A_k \omega_k \cos(\omega_k t) \tag{24}$$

where, the initial value x_0 and v_0 are zero. The acceleration a follows the sine distribution, and a will be considered process noise (v_k), respectively $A_k = 0.08 m/s$ and $\omega_k = \pi rad/s$.

The proposed eye tracking experiment is developed in platform of OPEN CV. Our system uses a ViewQuest VQ680 video camera to capture human images. The experiment is tested on a Pentium III 1.7G CPU with 128MB RAM. Eye tracking based on the proposed method can reach 10 frames per second. The format of input video is 352×288. Fig.1 represents the eye tracking using STFDEKF algorithm. The Correct Rate of eye tracking is shown in Table.1.
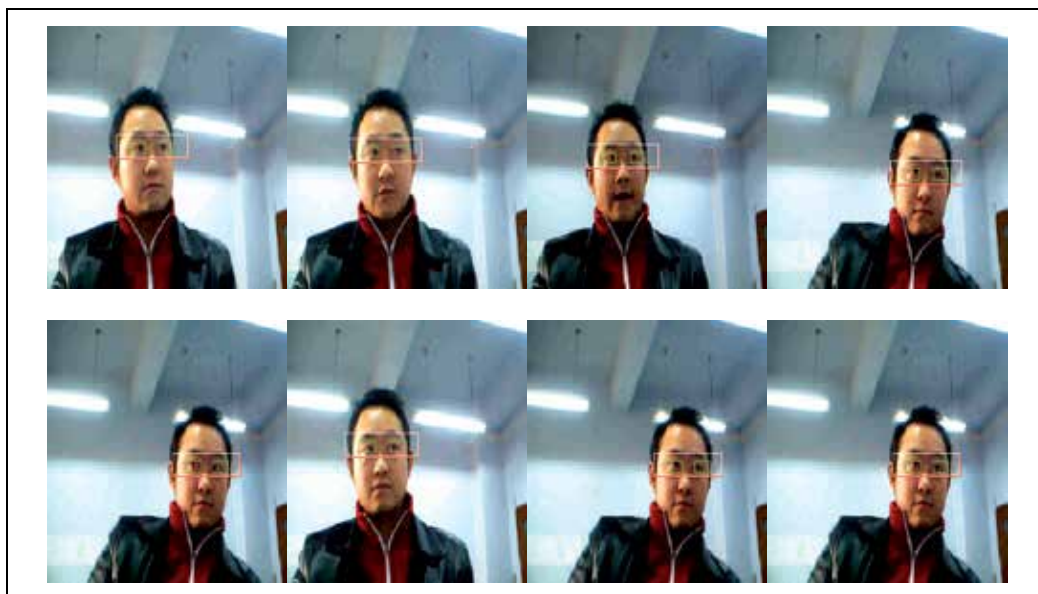
Fig. 1. Eye tracking using STFDEKF algorithm

	Video 1 (without glasses)	Video2 (glasses)	Video3(long hair)
Total Frames	1999	2941	2889
Tracking Failure	9	16	18
Correct Rate	99.45%	99.35%	99.4%
Average Correct Rate	99.4%		

Table 1. Result of eye tracking using STFDEKF algorithm

Correct Rate of eye tracking is defined as in equation (25).

$$Correct \quad Rate = \frac{Total \quad Frames - Tracking \quad Failure}{Total \quad Frames} \tag{25}$$

The following experimental results evaluate the performance of our proposed method and other eye tracking methods.

Algorithm	Correct rate	remark
Templates Match	99.1%	Refer to (Horng et al., 2004)
Kalman and mean shift algorithm	99.1%	Refer to (Zhu et al., 2002)
EKF tracking algorithm	99%	
STFDEK algorithm in this paper	99.4%	Refer to Table.1

Table 2. Comparison of eye tracking algorithms

In order to qualitatively gauge performance and discuss resulting issues, we consider using the traditional measures of performance: the RMSE (Root Mean Square Error) and MSE (Mean Square Error). The simulation results of RMSE and MSE are depicted in Table 3.

Algorithm	RMSE	MSE
Kalman Filter algorithm	0.13155	0.164661
EKF tracking algorithm	0.1222	0.0904
STFDEK algorithm in this paper	0.0989	0.0780

Table 3. RMSE and MSE of eye tracking filtering algorithms

The results of above experiments indicate that the proposed method has better performance. So we can use STFDEK algorithm for eyes tracking.

4. Conclusion

This paper proposes a new eye tracking method using strong finite-difference Kalman filter. Firstly, strong tracking factor is introduced to modify priori covariance matrix to improve the accuracy of the eye tracking algorithm. Secondly, the finite-difference method is proposed to replace partial derivatives of nonlinear functions to eye tracking. From above deduction, the new strong finite-difference Kalman filter becomes very simple because of replacing partial derivatives calculation using finite-difference value. The last experiment results show that STFDEKF has more accuracy and covariance estimation, improves the robustness of target tracking, and can be used for the high nonlinear stochastic systems such as eye tracking.

5. References

Qiang, J.; Zhu, Z.W. & Peilin Lan.(2004). Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue, *IEEE Transactions on Vehicular Technology*, VOL. 53, NO. 4, July 2004, pp.1052-1068, ISSN:0018-9545

Horng, W.B.; Chen, C.Y. & Chang, Y.(2004). Driver Fatigue Detection Based On Eye Tracking and Dynamic Template Matching, *Proceeding of the 2004 IEEE International Conference on Networking, Sensing & Control,* pp.7-12,ISBN:0-7803-8193-9, Taipei, Taiwan, March 21-23,2004

Dong, W.H. & Wu,X.J.(2004). Driver Fatigue Detection Based on the Distance of Eyelid, *Proceeding of the 2005 IEEE International Workshop VLSI Design & Video Tech,* pp.365-368,ISBN:0-7803-9005-9, Suzhou, China, May 28-30,2005

P.Majaranta & K.Raiha. (2002). Twenty year of eye typing: system and design issues, *Proceeding in ACM eye tracking research and applications symposium*, pp.15-22, ISBN: 1-58113-467-3, New Orleans, Louisiana,USA, March 2002

Laura A.; Granka & Thorsten Joachims. (2004). Eye-Tracking Analysis of User Behavior in WWW Search, *Proceeding of the 200 SIGIR*, pp.478-479, ISBN:1-58113-881-4, Sheffield, South Yorkshire, UK, July 25-29, 2004

Noton & Stark, L.(1971). Scanpaths in accadic eye movements while viewing and recognizing patterns vision research, PMID: 5538847, Jan 22,1971

Takehiko Ohno; Naoki Mukawa & Shinjiro Kawato.(2003).Just Blink Your Eyes: A Head-Free Gaze Tracking System, *Proceeding of the 2003 CHI*, pp.115-122, ISBN:1-58113-825-3, Ft. Lauderdale, Florida, USA, April 5–10, 2003

John D. McCarthy; Jens Riegelsberger & M. Angela Sasse.(2005). Commercial uses of eye tracking, *Proceeding of the 2005 British HCI Group Annual Conference*, Napier University, Edinburgh, Monday 5th September 2005

Li, D.H.; David Winfield &Derrick J.(2005). A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches, *Proceeding of the* IEEE Vision for Human-Computer Interaction, 2005

Zhu, J. & Yang, J.(2002). Subpixel eye gaze tracking, *Proceeding of the 2002 IEEE conference on Automatic Face and Gesture Recognition*, pp.124-129, ISBN:0-7695-1602-5, May 2002

Tohno; N.Mukawa & A.Yoshikawa.(2002). Freegaze: a gaze tracking system for everyday gaze interaction, *Proceeding of the 2002 Eye tracking research and applications symposium*, pp.15-22, ISBN:1-58113-467-3, New Orleans, Louisiana, March 2002

Zhu, Z.W.; Qiang, J. & Kikuo Fujimura.(2002). Combining Kalman Filtering and Mean Shift for Real Time Eye Tracking under Active IR Illumination, *Proceeding of the 2002 International Conference on Pattern Recognition (ICPR'02)*, ISBN:0-7695-1695-X, 2002

Y. Ebisawa.(1995). Unconstrained pupil detecting technique using two light sources and the image difference method, Visualization and Intelligent Design in engineering and architecture,pages 79-89, 1995

C.H. Morimoto & M. Flickner.(2000). Real-time multiple face detection using active illumination, *Proceeding of the 4th IEEE International Conference on Automatic Face and Gesture Recognition,* pp.8-13, ISBN:0-7695-0580-5, Grenoble, France, March 2000

Daugman.J.(1993). High confidence visual recognition of persons by a test of statistical independence, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, ISSN: 0162-8828

Nishino.K & S. Nayar.(2004). Eyes for relighting, *ACM SIGGRAPH 2004*, vol. 23, no. 3, pp. 704–711, 2004

Michael Chau & Margrit Betke.(2005). Real Time Eye Tracking and Blink Detection with USB Cameras, Boston University Computer Science Technical Report No. 2005-12

Zhou, D. H.; Xi, Y. G. & Zhang, Z. J.(1991). A suboptimal multiple fading extended Kalman filter, *Chinese J. Automation*, pp.689-695, ISSN: CN11-2109/TP, 1991

Zhou, D. H. (1990). Fault detection and diagnostics for a class of nonlinear systems, Ph.D. dissertation, Shanghai Jiao Tong Univ., P.R. China, 1990.

Zhou, D. H.; Sun, Y.X. & Zhang, Z. J.(1993). Extension of Friedland's Separate-Bias Estimation to Randomly Time-Varying Bias of Nonlinear Systems, *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, VOL. 38, NO. 8, AUGUST 1993

Fan,W.B.; LIU, C.F. & Zhang, S.Z.(2006). Improved Method of STRONG Tracking Extended
        Kalman Filter, Control and Decision Jan,2006

Zhou, D. H. & Q.L.(2006). Strong Tracking Filter of Nonlinear Systems with Colored Noise,
        Journal of Beijing Institute of Technology, vol.17, no.3, pp.321-326,1997

Estimation of Continuous-time Nonlinear Systems by using Unscented Kalman Filter

Min Zheng, Kenji Ikeda and Takao Shimomura
The University of Tokushima
Japan

1. Introduction

The discrete-time model is often used for the system identification. However, the controlled plant is a continuous-time system in many cases. In addition, there are some disadvantages in the discrete-time model, such as the discrete-time model has a complex representation of the continuous-time model parameters, and can't reflect the structure of the plant. Especially for the nonlinear system, if the sampling period is large, system nonlinearity will be enlarged, and the nonlinear discrete-time model can't be identified well. Because of these reasons, the method for estimating the parameter of the continuous-time system from the sampled I/O data directly has attracted attention.

Estimation in nonlinear system is very important, because almost all practical systems involve nonlinearities. The Unscented Kalman Filter (UKF) is a nonlinear estimation method, which propagates mean and covariance information through nonlinear transformation. It is accurate, and has superior implementation properties. Plant parameters can be estimated based on the UKF like algorithm by defining an augmented state as the state and the unknown parameters. As it is well known, the UKF uses sigma points to capture the statistics of a Gaussian random variable, instead of calculating the Jacobian matrices, and the UKF does not use linear approximation. Furthermore, it does not matter if the plant is based on continuous-time model, because the one-step-ahead estimate in continuous-time model can be calculated by numerical integration. From these reasons, it is possible to estimate the state and the parameters of a continuous-time system by using the UKF.

In order to demonstrate the validity, the Rotary Pendulum is provided to estimate the unknown parameter of the continuous-time nonlinear system. For the numerical simulation, system parameters have been almost exactly estimated. From the experimental I/O data, system parameter has been estimated within one percent Relative Root Squared Error (RRSE) by using the UKF like algorithm.

2. Continuous-time model and discrete-time model

2.1 Dynamical system

System is an object in which variables of different kinds interact and produce observable signals. As shown in Fig.1, system can be represented by input $u(t)$, output $y(t)$, and disturbance $v(t)$. As a general rule, $u(t)$ and $y(t)$ can be measured possibly, however, $v(t)$ can not be measured usually.

Dynamical system is a system whose output depends   not only the current input but also the past input.

Fig. 1. Dynamical system

In general, the dynamical system shown in Fig.1 is represented with n-dimensional state variables $\{x_1, \ldots, x_n\}$ by a first-order differential equation as follows:

$$
\left.
\begin{aligned}
\dot{x}_1(t) &= f_1(t; x_1, \ldots, x_n; u_1, \ldots, u_l) \\
\dot{x}_2(t) &= f_2(t; x_1, \ldots, x_n; u_1, \ldots, u_l) \\
&\vdots \\
\dot{x}_n(t) &= f_n(t; x_1, \ldots, x_n; u_1, \ldots, u_l)
\end{aligned}
\right\}
\tag{1}
$$

where $\{u_1, \ldots, u_l\}$ is l-dimensional system input signals, and m-dimensional output

$$
\left.
\begin{aligned}
\dot{y}_1(t) &= g_1(t; x_1, \ldots, x_n; u_1, \ldots, u_l) \\
\dot{y}_2(t) &= g_2(t; x_1, \ldots, x_n; u_1, \ldots, u_l) \\
&\vdots \\
\dot{y}_n(t) &= g_n(t; x_1, \ldots, x_n; u_1, \ldots, u_l)
\end{aligned}
\right\}
\tag{2}
$$

can be measured. From

$$
x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_m(t) \end{bmatrix}, u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_l(t) \end{bmatrix}
\tag{3}
$$

and vector functions

$$f(t, x, u) = \begin{bmatrix} f_1(t, x, u) \\ f_2(t, x, u) \\ \vdots \\ f_n(t, x, u) \end{bmatrix}, g(t, x, u) = \begin{bmatrix} g_1(t, x, u) \\ g_2(t, x, u) \\ \vdots \\ g_m(t, x, u) \end{bmatrix} \tag{4}$$

eqs.(1) and (2) can be expressed briefly as follows

$$\dot{x}(t) = f[t, x(t), u(t)] \tag{5}$$

$$y(t) = g[t, x(t), u(t)] \tag{6}$$

$x(t)$ is called state vector, eqs.(1) and (5) are called state equation, and eqs.(2) and (6) are called output equations.

In order to analyze the discrete-time model, f and g are assumed to be linear in section 2.2. At that time, eqs.(7) and (8) are used instead of eqs.(5) and (6)

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{7}$$

$$y(t) = C(t)x(t) + D(t)u(t) \tag{8}$$

where $A(t) \in R^{n \times n}, B(t) \in R^{n \times l}, C(t) \in R^{m \times n}, D(t) \in R^{m \times l}$ respectively. It is called a time-varying linear system if the system is represented by eqs.(7) and (8). When $A(t)$, $B(t)$, $C(t)$, $D(t)$ are constant matrices and do not depend on the time variable t, the system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{9}$$

$$y(t) = Cx(t) + Du(t) \tag{10}$$

is called a time-invariant linear system. In many cases, $D(t) = 0$, because phase of a physical system always delays in high frequency range. Output eq.(8) can be changed as

$$y(t) = C(t)x(t) \tag{11}$$

Vis-a-vis the linear system, the system which is represented with eqs.(1), (2) or the eqs.(5), (6) is called a nonlinear system.

2.2 Continuous-time model and discrete-time model

The behavior of a dynamic system evolves over time. The discrete-time model is often used for the system identification. However, in many cases the controlled plant is a continuous-time system, which its descriptive equations are defined for all values of time and the system dynamic properties shown by the differential equations. There are some properties of continuous-time model,

1. In the control design, the important parameters are easy to be grasped.

2. If the physical structure of the plant is known, the mathematical continuous-time model can be obtained beforehand.

The focus in this chapter targets the modeling of continuous-time systems. As for the second property, if the differential equations of the continuous-time model can be obtained, the identification problem afterwards can be replaced by the parameter estimation problem.

In many applications, particularly in physical modeling, the design of a discretetime model starts from the description of a physical continuous-time model by means of differential equations and constraints. Therefore, for this section, the continuous-time model is represented as

$$\dot{x}(t) \;=\; Ax(t) + Bu(t) \tag{12}$$

$$y(t) \;=\; Cx(t) \tag{13}$$

where $u \in R^m$, $y \in R^l$, $A \in R^{n \times n}$, $B \in R^{n \times m}$, $C \in R^{l \times n}$. On the other hand, for performing a system by using a digital computer, it has become prerequisite to handle the sampled data. Discrete-time model is the mathematical model in which the I/O relation of the sampled data is shown by a difference equation. Corresponding to eqs.(14) and (15), the discrete-time model can be described by

$$x_{k+1} \;=\; Fx_k + Gu_k \tag{14}$$

$$y_k \;=\; Hx_k \tag{15}$$

Similarly to the continuous-time model, x_k, u_k and y_k are state variable vector, input and output at a time step k respectively. And $G \in R^{n \times m}$ and $H(=C) \in R^{l \times n}$ are the system matrices of the discrete-time model.

The solution of eq.(14) can be solved as

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \tag{16}$$

where t_0 is the initial time. Let t_k, k=0, 1, . . ., denote the sampling time. If the input $u(t)$ is a constant u_k for the sampling interval $[t_k, t_{k+1})$, i.e.

$$u(t) = \begin{cases} u_k & (t_k \leq t < t_{k+1} \quad k = 0,1,2,\ldots) \\ 0 & (t \leq t_0) \end{cases} \tag{17}$$

Then

$$x(t_{k+1}) = e^{A(t_{k+1}-t_k)}x(t_k) + \left[\int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}d\tau \right] Bu_k \tag{18}$$

can be obtained. Here, assume $t_{k+1} - t_k = \Delta$(const.), the discrete-time system representated as follow

$$x(t_{k+1}) = Fx(t_k) + Gu_k \tag{19}$$

where

$$F = e^{A\Delta}, \quad G = \left[\int_0^\Delta e^{A\sigma} d\sigma \right] B \tag{20}$$

Δ is the sampling period.

For zero-order hold of the sampling, it is possible to calculate without the approximate error. However, if unlimitedly reduces Δ,

$$\lim_{\Delta \to 0} e^{A\Delta} = I, \quad \lim_{\Delta \to 0} \left[\int_0^\Delta e^{A\sigma} d\sigma \right] B = 0 \tag{21}$$

F and G approach to identity matrix and zero matrix, regardless of the elements of A and B. Moreover, if the obtained discrete-time model is clear to identity matrix or zero matrix, the backward calculation from the discrete-time model to the continuous-time model becomes numerically unstable. Therefore, for the discrete-time model, the value of the model depends on the sampling period. And it is not limited to represent the dynamics of the actual system accurately, though the sampling period be diminished simply.

In a summary, for the discrete-time model and continuous-time model, there are some problems

1. The discrete-time model does not reflect the structure of the plant.
2. The discrete-time model has a complex representation of the continuous-time model parameters.
3. Especially for the nonlinear system, if the sampling period is large, system nonlinearity will be enlarged, and the nonlinear discrete-time model can't be identified well.

Because of these reasons, the method for estimating the parameter of the continuous-time system from the sampled I/O data directly has attracted attention.

3. Unscented Kalman filter

Estimation in nonlinear system is very important because many practical systems involve nonlinearities. The Extended Kalman Filter (EKF) which applies the KF to nonlinear system by linearizing all nonlinear models, has become a most widely used method for estimation of nonlinear system. However, more than 35 years of experience in the estimation community, although the EKF maintains the elegant and computationally efficient recursive update form of the KF, it suffers a number of serious limitations.

1. Only reliable for systems which are almost linear on the time scale of the updates.
2. Linearization can be applied only if the Jacobian matrix exists. However, this is not always the case.
3. Calculating Jacobian matrices can be a very difficult and error-prone process.

It means the EKF is difficult to implement, difficult to tune, and the reliability is limited. To address the limitations, the Unscented Kalman Filter (UKF) was proposed by Julier and Uhlmann in 1996.

The UKF is a nonlinear estimation method which propagates mean and covariance information of the parameter recursively through nonlinear transformation. As it is well known, the UKF is a straightforward extension of the Unscented Transformation (UT) to the recursive estimations. It uses sigma points to capture the statistics instead of calculating the Jacobian matrices, and the UKF does not use the linear approximation of functions. It is accurate, and has superior implementation properties. As a nonlinear estimation method, the UKF has been widely applied in nonlinear control applications.

3.1 Unscented Transformation

The UT is a nonlinear method for calculating the statistics of a random variable which undergoes a nonlinear transformation and builds on the principle that it is easier to approximate a PDF(probability distribution function) than to approximate an arbitrary nonlinear function.

The approach is illustrated in Fig.2. And the the principle of the UT is as follows:
1. Sigma points are chosen from the mean and covariance.
2. The nonlinear function is applied to each points in turn to yield a cloud of transformed points.
3. The statistics of the transformed points can then be calculated to form an estimate of the nonlinearly transformed mean and covariance.
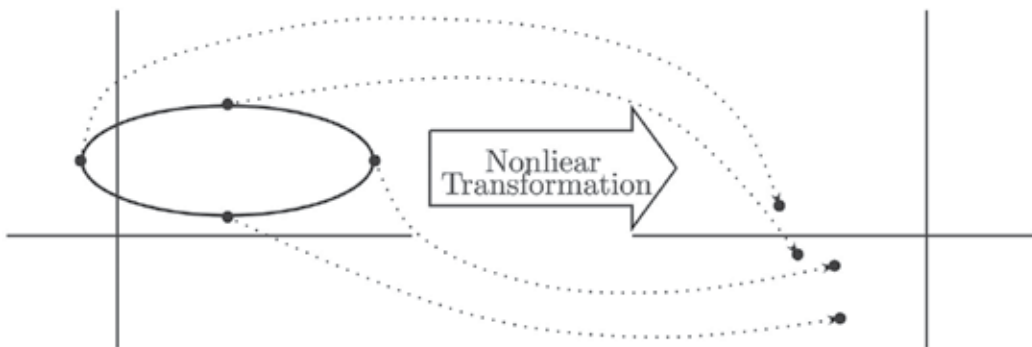
Fig. 2. The principle of the UT

Different from other methods, there are two important distinctions of the UT.
1. The sigma points are chosen deterministically from the statistics of transformation, and not drawn at random.
2. The approximation itself can be interpreted more generally than as probability distribution.

3.2 Calculating sigma points

As described in section 3.1, the UT method is founded as a nonlinear method to calculate the sigma points which are deterministically chosen from the propagated mean and covariance through nonlinear transformations. This section describes how to calculate $2n + 1$ weighted sigma points of the n-dimensional samples, and illustrate the coordinate transformation to demonstrate the improved accuracy of the UT.

Consider a n-dimensional random variable x through a nonlinear functuon. Assume x has mean $\bar{x}$ and covariance P. Calculate a set of $2n + 1$ weighted sigma points $\{\mathcal{X}^i; i = 1, 2, \ldots, 2n\}$, where $\mathcal{X}^i \in R^n$. The first sigma point is

$$\mathcal{X}_k^0 = \bar{x}_k \tag{22}$$

and the other sigma points are calculated using the following general selection scheme:

$$\mathcal{X}_k^i = \bar{x}_k + \left(\sqrt{(n+\kappa)P}\right)_i \tag{23}$$

$$\mathcal{X}_k^{i+n} = \bar{x}_k - \left(\sqrt{(n+\kappa)P}\right)_i \tag{24}$$

$$i \in \{1, 2, \ldots, n\}$$

where $\kappa \in R$ is a scaling parameter, $\left(\sqrt{(n+\kappa)P}\right)_i$ is ith row or column of the matrix square root of $(n + \kappa)P$, κ scales the third and higher order terms of this set. If $(n + \kappa) = 3$, it is possible to match some of the fourth order terms when x is Gaussian.

The scaled result is a different sigma set, with different higher moments, but with the same mean and covariance. The weight is provided to weight the point for controlling some aspects of distribution of the sigma points. By convention, let W^0 be the weight on the mean point, adjusting the weights as follows:

$$W^0 = \kappa/(n+\kappa) \tag{25}$$

$$W^i = 1/2(n+\kappa) \tag{26}$$

$$W^{i+n} = 1/2(n+\kappa) \tag{27}$$

$$i \in \{1, 2, \ldots, n\}$$

Calculate the mean:

$$\sum_{i=0}^{2n} W^i \mathcal{X}_k^i = \bar{x}_k \tag{28}$$

and the covariance:

$$\begin{aligned}
& \sum_{i=0}^{2n} W^i (\mathcal{X}_k^i - \bar{x}_k)(\mathcal{X}_k^i - \bar{x}_k)^T \\
=\ & \sum_{i=0}^{2n} 2W^k(n+\kappa)(\sqrt{P})_i(\sqrt{P})_i^T \\
=\ & \sum_{i=0}^{2n} (n+\kappa)(\sqrt{P})_i(\sqrt{P})_i^T \\
=\ & P
\end{aligned} \tag{29}$$

From the results, the mean and covariance of $\mathcal{X}_k^i$ are same to them of x_k is found.

3.3 The coordinate transformation problem

The problem of converting uncertain information between polar and Cartesian coordinate is a special case of the general problem for applying a nonlinear projection to a random variables. Here, illustrates coordinate transformation to describe the properties of the UT and demonstrates the improved accuracy of the UT.

An example of a coordinate system is to describe polar information (r, θ) returned in its local coordinate frme that has to be converted into an (x, y) position in Cartesian coordinate frme

$$f(r, \theta) = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r\cos\theta \\ r\sin\theta \end{pmatrix} \tag{30}$$

where samples are 1000 polar coordinate range, and standard deviation $\sigma_r = 0.02[m]$, $\sigma_\theta = \pi/12[rad]$ for the true value $(r^*, \theta^*)^T = (1.0, \pi/2)^T$. The Fig.3 plots the (r, θ) samles, the mean, and the covariance ellipse. The performance of the UT, sigma points $\mathcal{X}^0, \ldots, \mathcal{X}^4$ is shown in Fig.4.

One thousand (x, y) samples from the transformation and the statistics calculated though the nonlinearization are plotted in the Fig.5. As can be seen, the points lie on a "banana"-shaped arc. Fig.6 plots the mean and standard deviation ellipses for the true statistics, and the set of sigma points $\mathcal{X}^0, \ldots, \mathcal{X}^4$ which have undergone the nonlinear transformation by the UT.
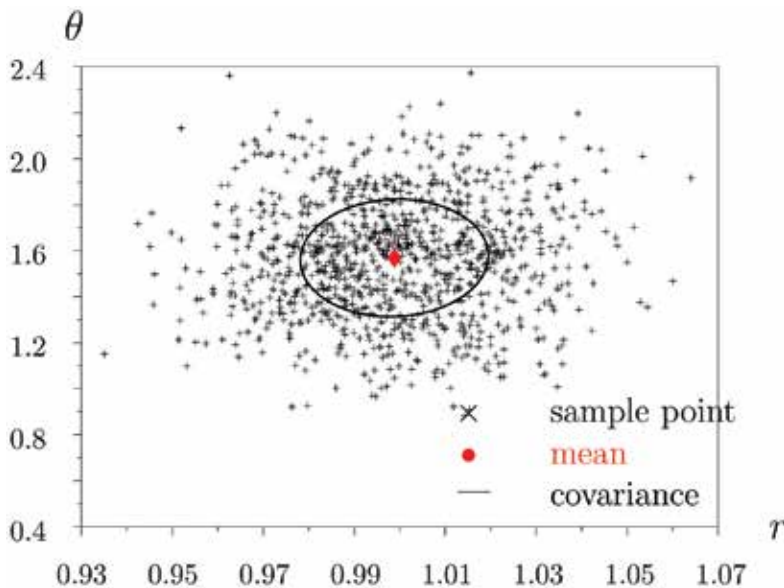
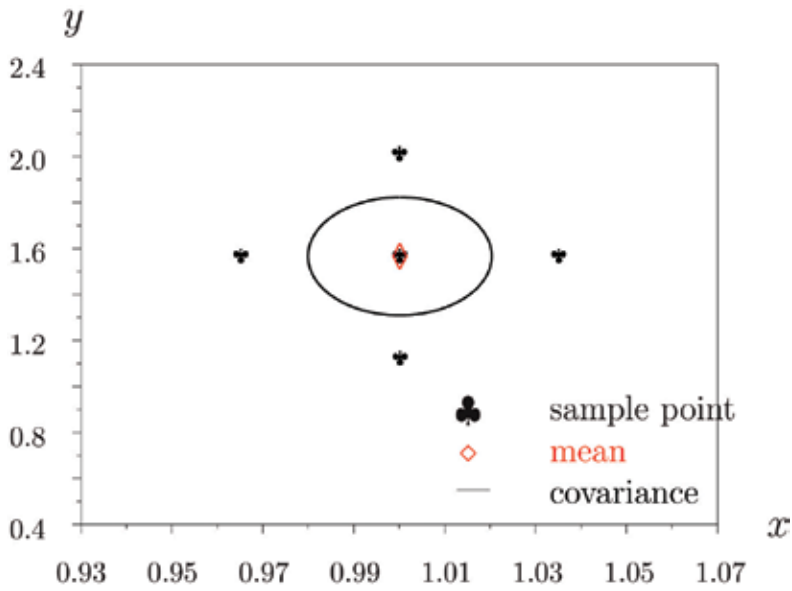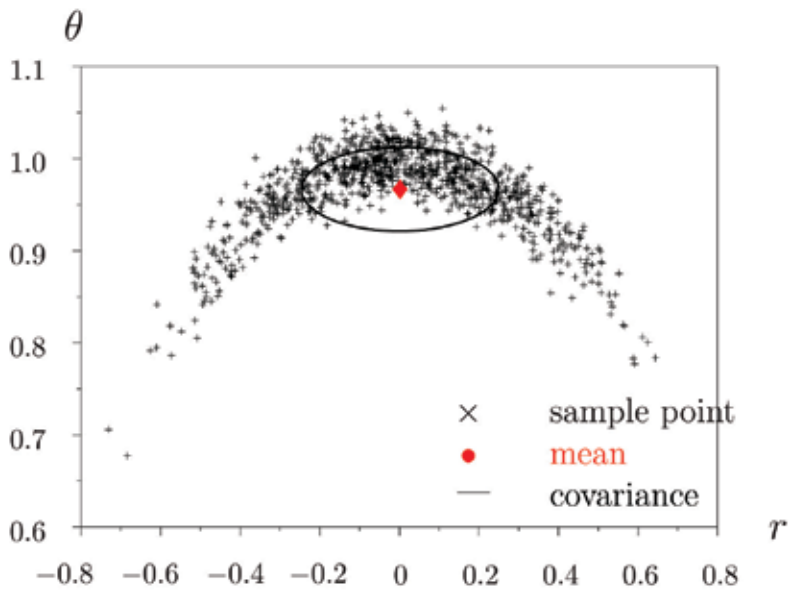

Fig. 3. Polar Coordinate Samples

Fig. 4. Sigma Points

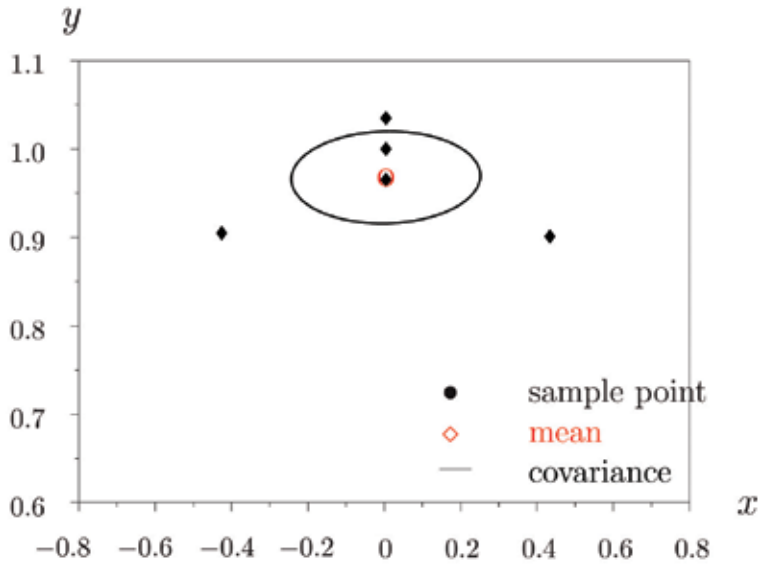Fig. 5. Transformed samples

Fig. 6. Transformed sigma points

3.4 Formulation of problem

Consider a continuous-time nonlinear sysytem,

$$\dot{x}(t) = f(x(t), u(t), \theta) \tag{31}$$

$$y(t) = g(x(t), \theta) + \nu(t) \tag{32}$$

where $x(t) \in R^{n_x}$ is the system state, $u(t) \in R$ is the control input, $y(t) \in R$ is the system output, θ denotes the unknown plant parameters, and $\nu(t) \in R$ is the measurement noise with zero mean and its covariance matrix R. The discrete-time model of the system can be represented as:

$$x_{k+1} = f_d(x_k, u_k, \theta) \tag{33}$$

$$y_k = g(x_k, \theta) + \nu_k \tag{34}$$

where $x_k = x(kT)$, the subscript k is a discrete time, $k \in \{1, \ldots, N-1\}$, and T is the sampling period. An explicit formula of f_d is not required, but a calculation procedure such as numerical integration is required. Let $X_k = (x_k^T, \theta_k^T)^T$, the equations are rearranged as:

$$X_{k+1} = F(X_k, u_k) \tag{35}$$

$$y_k = G(X_k) + \nu_k \tag{36}$$

3.5. UKF algorithm

Denote an estimate of $\hat{X}_k$ at a time step l as $\hat{X}_{k|l}$. For the general formulation of the UKF, the n-dimensional state with mean $\hat{X}_{k|k}$ and covariance $P_{k|k}$ are approximated by $2n + 1$ weighted sigma points. The index i takes values over $\{1, \ldots, n\}$.

The estimation will be performed as fllows:

1. Initialization of $\hat{X}_{0|0}$, $P_{0|0}$, and R_0.
2. Calculatation of sigma points:

$$\mathcal{X}_{k|k}^0 = \hat{X}_{k|k} \tag{37}$$

$$\mathcal{X}_{k|k}^0 = \hat{X}_{k|k} \tag{38}$$

$$\mathcal{X}_{k|k}^i = \hat{X}_{k|k} + \left(\sqrt{(n+\kappa)P_{k|k}}\right)_i \tag{39}$$

$$W^i = 1/(2(n+\kappa)) \tag{40}$$

$$\mathcal{X}_{k|k}^{i+n} = \hat{X}_{k|k} - \left(\sqrt{(n+\kappa)P_{k|k}}\right)_i \tag{41}$$

$$W^{i+n} = 1/2(n+\kappa) \tag{42}$$

3. Time update:

$$\mathcal{X}_{k+1|k}^i = F(\mathcal{X}_{k|k}^i, u_k) \tag{43}$$

$$\hat{X}_{k+1|k} = \sum_{i=0}^{2n} W^i \mathcal{X}_{k+1|k}^i \tag{44}$$

$$P_{k+1|k} = \sum_{i=0}^{2n} W^i (\mathcal{X}_{k+1|k}^i - \hat{X}_{k+1|k}) \times (\mathcal{X}_{k+1|k}^i - \hat{X}_{k+1|k})^T \tag{45}$$

$$\mathcal{Y}_{k+1|k}^i = G(\mathcal{X}_{k+1|k}^i, u_k) \tag{46}$$

$$\hat{y}_{k+1|k} = \sum_{i=0}^{2n} W^i \mathcal{Y}_{k+1|k}^i \tag{47}$$

4. Measurement update:

$$P_{k+1|k}^{yy} = \sum_{i=0}^{2n} W^i (\mathcal{Y}_{k+1|k}^i - \hat{y}_{k+1|k}) \times (\mathcal{Y}_{k+1|k}^i - \hat{y}_{k+1|k})^T \tag{48}$$

$$P_{k+1|k}^{xy} = \sum_{i=0}^{2n} W^i (\mathcal{X}_{k+1|k}^i - \hat{X}_{k+1|k}) \times (\mathcal{Y}_{k+1|k}^i - \hat{y}_{k+1|k})^T \tag{49}$$

$$v_{k+1} = y_{k+1} - \hat{y}_{k+1|k} \tag{50}$$

$$P_{k+1|k}^{vv} = R_{k+1} + P_{k+1|k}^{yy} \tag{51}$$

$$W_{k+1} = P_{k+1|k}^{xy} P_{k+1|k}^{vv}{}^{-1} \tag{52}$$

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + W_{k+1} v_{k+1} \tag{53}$$

$$P_{k+1|k+1} = P_{k+1|k} - W_{k+1} P_{k+1|k}^{vv} W_{k+1}^T \tag{54}$$

Summarily, the UKF uses sigma points to capture the mean and covariance of a Gaussian random variable, instead of calculating the Jacobian matrices. Plant parameters can be estimated based on the UKF like algorithm by augmenting the state with the unknown parameters. Furthermore, it does not matter if the estimation is based on continuous-time model, because the one-step-ahead estimate in continuous-time model can be calculated by numerical integration. From these reasons, it is possible to estimate the state and the parameters of a continuous-time system by using the UKF like algorithm.

4. Numerical example

In order to demonstrate the effectiveness of the proposed method, the Rotary Pendulum is provided to estimate the parameters of the continues-time nonlinear system by using the UKF from the sampled I/O data.

The schematic representation of the Rotary Pendulum system is shown in the Fig.7, where m is the pendulum mass, r is the arm length, l denotes half the length of the pendulum. The total effective moment of base inertia is J_b. Each of the angle of the pendulum α and the angle of the arm φ is measured by the potentiometer.
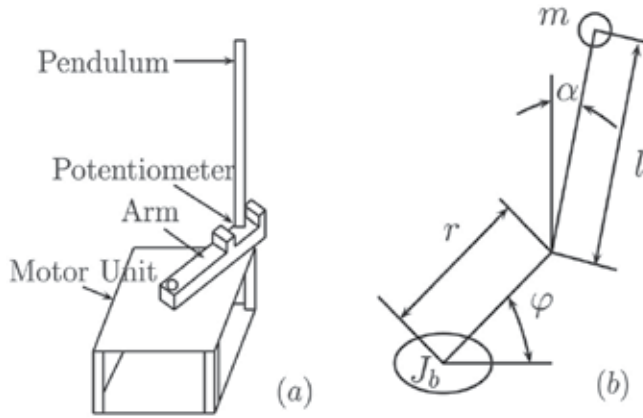

Fig. 7. Schematic Diagram of Rotary Pendulum

Consider the nonlinear model of this system, the nonlinear equations can be derived by Lagrange equations:

$$(mr^2 + J_b)\ddot{\varphi} - (mrl\cos\alpha)\ddot{\alpha} + (mrl\sin\alpha)(\dot{\alpha})^2 = \tau \tag{55}$$

$$-(mrl\cos\alpha)\ddot{\varphi} + \frac{4}{3}ml^2\ddot{\alpha} + mgl\sin\alpha = 0 \tag{56}$$

Take consider of the torque τ as follow:

$$\tau = \frac{KgKm}{R}v - \frac{Kg^2Km^2}{R}\dot{\varphi} \tag{57}$$

where K_g is Gear ratio in motor, K_m denotes Motor torque constant, and R_m is Motor Torque DC resistance,

and rearrange the equations into the state space representation given as:

$$\frac{d}{dt} \begin{pmatrix} \varphi \\ \alpha \\ \dot{\varphi} \\ \dot{\alpha} \end{pmatrix} = \begin{pmatrix} \dot{\varphi} \\ \dot{\alpha} \\ J(m,r,l,J_b,\alpha)^{-1} L(m,r,l,K_g,K_m,R_m,\alpha,\varphi) \end{pmatrix} \quad (58)$$

where

$$J(m,r,l,J_b,\alpha) = \begin{pmatrix} mr^2 + J_b & -mrl\cos\alpha \\ -r\cos\alpha & \frac{4}{3}l \end{pmatrix} \quad (59)$$

$$L(m,r,l,K_g,K_m,R_m,\alpha,\varphi) = \begin{pmatrix} \frac{K_g K_m}{R_m}v - \frac{K_g^2 K_m^2}{R_m}\dot{\varphi} - mrl\cos\alpha(\dot{\alpha})^2 \\ -g\sin\alpha \end{pmatrix} \quad (60)$$

The parameters of the plant can be estimated from eq.(58) by using the UKF like algorithm. The numerical values of parameters are provided in Tabel.1.

Desription	Symbol	value	Units
Pendulum mass	m	0.05	K_g
Pendulum arm length	r	0.175	m
1/2 of Pendulum length	l	0.198	m
Gravitational const	g	9.8	m/sec^2
Base inertia monment	J_b	0.0122	$K_g m^2$
Motor Torque DCresistance	R_m	8.3	Ω
Motor torque constant	K_m	0.0625	N_m/Amp
Gear ratio	K_g	7.5	/

Table 1. Parameters of the experiment Rotary Pendulum system

4.1 Numerical simulation result
Offer the system an input voltage to observe the system output as in Fig.8. The sampling period T is 0.008668 seconds, and 1001 sampled I/O data used.
Assume there is no noise in the system to make the simulations.
Estimate the augmented state $X = [x, \theta]$ of the continuous-time model by using the UKF from the sampled I/O data to demonstrate the effectiveness of the UKF, where $x = [\varphi, \alpha, \dot{\varphi}, \dot{\alpha}]^T$ is the system state and $\theta = [m, r, l]^T$ is the system parameter. The estimated parameters can be seen in Fig.9.
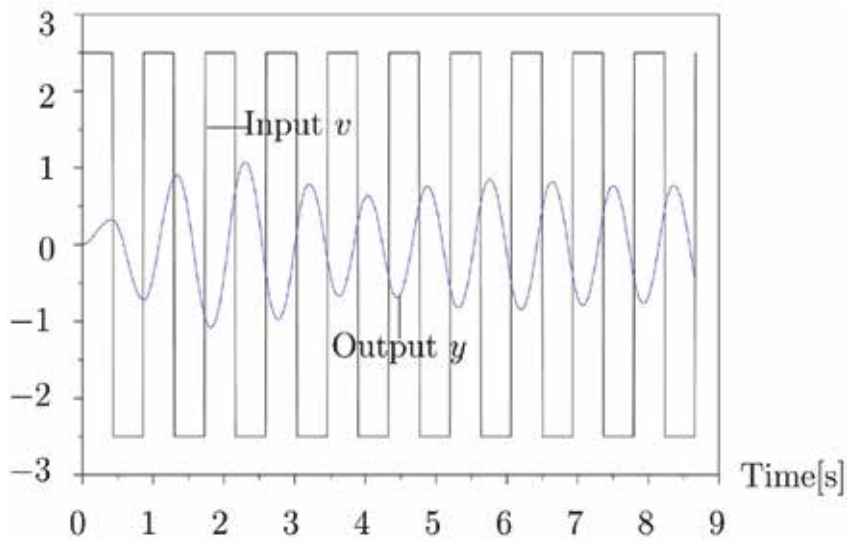
Fig. 8. Input and Output of the Rotary Pendulum system
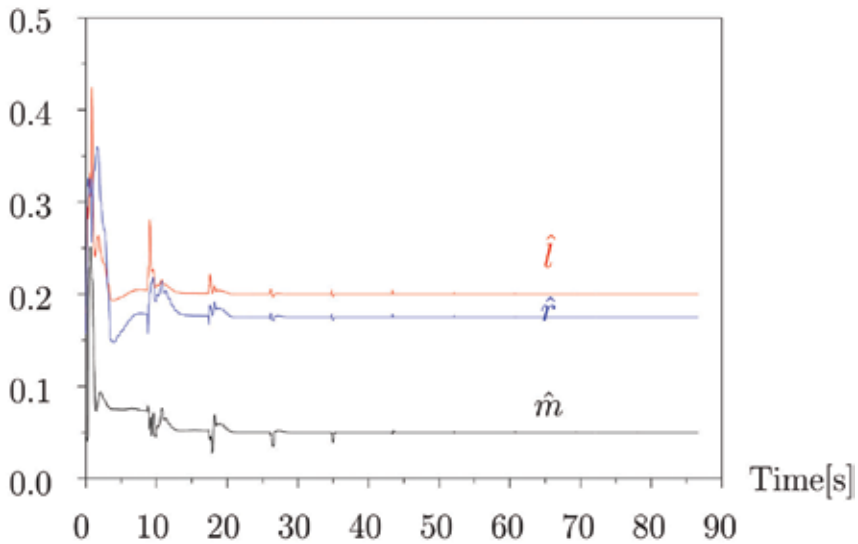

Fig. 9. Estimated parameters

Fig.10 shows the Relative Root Squared Error (RRSE) of the estimate for each sampling time which is defined by:

$$RRSE_k = \frac{\|X_k^* - \hat{X}_k\|}{\|X_k^*\|} \tag{61}$$

where X_k^* is the true value and $\hat{X}_k$ is the estimate at a time step k. The RRSE reduced to 1.493×10^{-14}, an extremely small value, indicated that this method has very high precision.
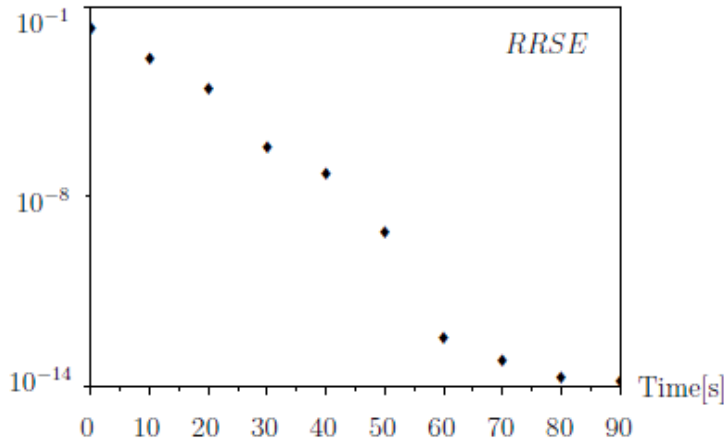
Fig. 10. RRSE of the parameter esimation

4.2 Experimental result of the actual system

The above results are the numerical simulation results. In fact, the actual result is most important. Therefore, estimate the plant parameter from the experimental I/O data of actual system next.

For the actual experiment, the Rotary Pendulum system is excited by a voltage input signal which is plotted in Fig.11, while the pendulum position is measured. Subsequently estimate the parameter of the Rotary Pendulum from the collected data. This time, the unknown parameter is the length of pendulum l.
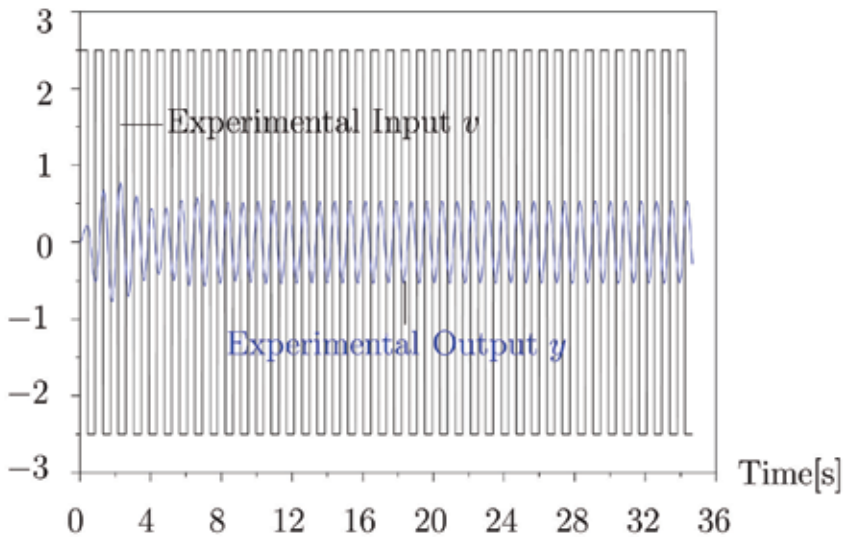

Fig. 11. Experimental I/O data of the actual system

First step, estimate the system parameter by using the EKF. In this step, calculate the Jacobian matrix, then perform the estimation based on the EKF algorithm by the sampled

I/O data. The estimation result based on experimental output is plotted in Fig.12. From the plot, as can be seen, the estimated parameter converges to the different value from the true value. And the RRSE is about 0.128, it means the low-precision of the estimation based on the EKF.
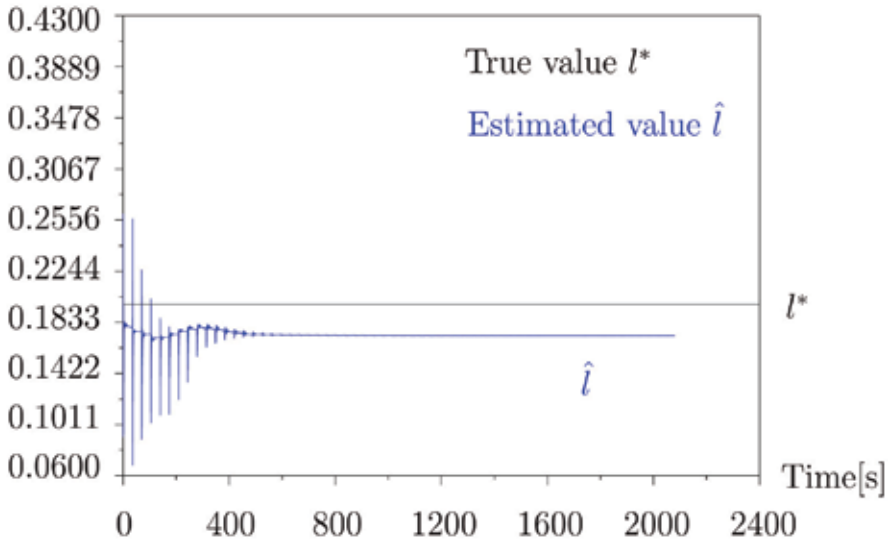

Fig. 12. Esimation result based on the experimental I/O data by using the EKF

Next step, estimate the parameter by using the UKF like algorithm. Fig.13 is the estimated parameter. The RRSE of the estimation is less than one percent, about 5.866×10^{-3}.
Comparing the result of these two methods, the high-precision of the estimation based on the UKF like algorithm is known.
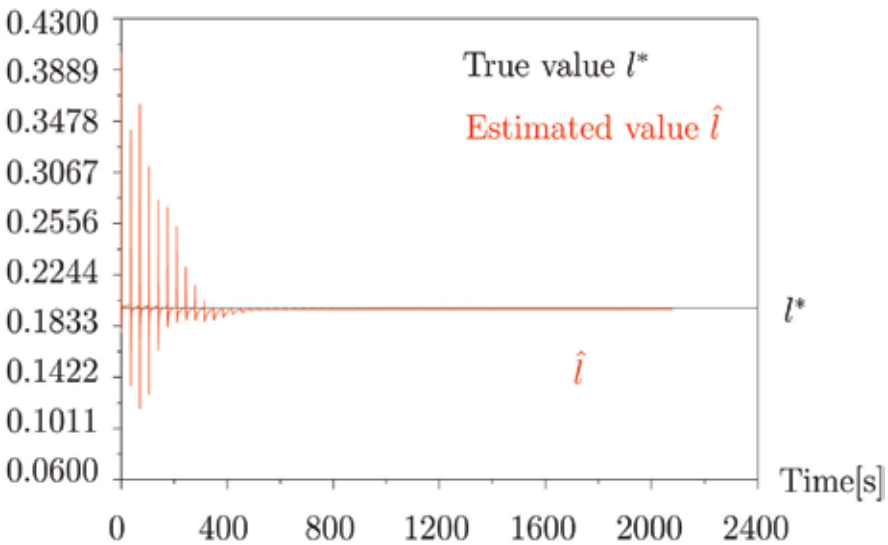

Fig. 13. Esimation result based on the experimental I/O data by using the UKF

5. Conclusion

In this chapter, direct estimation of the continuous-time systems from the sampled I/O data by using the UKF like algorithm is paid attention, and the Rotary Pendulum is provided to estimate the parameters of the continuous-time nonlinear system for demonstrating the validity of the UKF. Through the simulation and the experiment results, we found that, for the numerical simulation, system parameters have been almost exactly estimated, and from the experimental I/O data, system parameter has been estimated within one percent RRSE by using the UKF like algorithm.

All the simulations were set up under the condition that the initial value is known. The estimation of initial states is very important for obtaining the correct estimates of the system parameters. However, for the practical plants, the initial state may not be measured because there is a dead zone of the sensor. If the initial state is unknown, the covariance of the initial state has to be set large, and it leads to low precision of the parameter estimation. Therefore, we are to propose a continuous-time model estimation method by using the UKF like algorithm, in which the initial state as well as the paramters are estimated, as a future research.

6. References

[1] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," Proceeding of the IEEE, vol1. 92 No. 3, pp. 401-422, Mar, 2004.

[2] S. Julier, J. Uhlmann, and H. F. Durrant Whyte, "A New Method for the Nonlinear Transfor- mation of Means and Covariances in Filters and Estimators," IEEE Trans. Automat. Contr., Vol. 45, pp. 477-482, Mar, 2000.

[3] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Comment on "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimations"," IEEE Trans. Automat. Contr., vol. 47, pp. 1406-1408, Aug, 2002.

[4] Eric A. Wan and Rudolph van der Merwe "The Unscented Kalman Filter for Nonlinear Estimation," Proc of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control, Lake Louise, Alberta, 2000:153- 158.

[5] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Trans. ASME J.Basic Eng., vol1. 82, pp. 35-45, Mar, 1960.

[6] Greg Welch, Gary Bishop, "An Introduction to the Kalman Filter" UNCChapel Hill, TR 95-041, July 24, 2006

[7] Lefebvre, Tine, Bruyninckx, Herman, De Schutter, and Joris: "Nonlinear Kalman Filtering for Force-Controlled Robot Tasks", Springer Tracts in Advanced Robotics, Vol. 19, Nov, 2005.

[8] K. Reif, S. Günther, E. Yaz, and R. Unbehuaen: "Stochastic Stability of the Discrete-Time Extended Kalman Filter", IEEE Trans. Automat. Contr., Vol.44, No.4, Apr, 1999.

[9] Sy-Miin Chow, Emilio Ferrer, John R. Nesselroade, "An Unscented Kalman Filter Approach to the Estimation of Nonlinear Dynamical Systems Models" Multivariate Behavioral Research, v42.n2, p283-321, 2007.

[10] Maria Isabel Ribeiro, "Kalman and Extended Kalman Filters: Concept, Derivation and Properties" Institute for Systems and Robotics, p43, February, 2004.

[11] Joseph J. LaVio la Jr., "A Comparison of Unscented and Extended Kalman Filter for Estimating Quaternion Motion" American Control Conference, 2435-2440 vol.3, Jun, 2003.

[12] Lonnie C. Ludeman, "Random Processes: Filtering, Estimation, and Detection", ISBN 7-121-00866-1, January, 2003.

[13] Kenji Ikeda, Yoshio Mogami and Takao Shimomura, "Continuous-time model identification by using adaptive observer, − Estimation of the initial state −", Proc. of SICE-ICCAS 2006, pp.1796-1799, Busan, Oct, 2006.

[14] Kenji Ikeda, Yoshio Mogami and Takao Shimomura, "Continuous-time model identification by using adaptive observer", of 13th IFAC Symposium on System Identification, pp.325-330, Rotterdam, Aug, 2003.

[15] Rudolph van der Merwe and Eric A. Wan, "Sigma-Point Kalman Filters for Intergrated Navigation", Proceedings of the 60th Annual Meeting, pp.641-654, Jun, 2004.

[16] S. J. Julier: "Reduced Sigma Point Filters for the Propagation of Means and Covariances Through Nonlinear Transformations", available in http://citeseer.ist.psu.edu/

[17] S. J. Julier: "The Scaled Unscented Transformation", available in http://citeseer.ist.psu.edu/

[18] S. J. Julier, and J. K. Uhlmann: " A Consistent, Debiased Method for Converting Between Polar and Cartesian Coordinate Systems", available in http://citeseer.ist.psu.edu/

[19] S. J. Julier, and J. K. Uhlmann: "A New Extension of the Kalman Filter to Nonlinear Systems", available in http://citeseer.ist.psu.edu

[20] Brian D.O. Anderson, and John B. Moore, "Optimal Filtering", Dover 0-486- 43938-0

[21] Mohinder S. Grewal, Angus P. Andrews, "Kalman Filtering: theory and practice Using Matlab, second edition", Wiley-Interscience, Jan, 2001.

[22] Karl J.Åström, BJörnWittenmark, "Computer-Controlled Systems: theory and design, second edition", Prentice Hall, November, 2006.

Edited by Vedran Kordic

The Kalman filter has been successfully employed in diverse areas of study over the last 50 years and the chapters in this book review its recent applications. The editors hope the selected works will be useful to readers, contributing to future developments and improvements of this filtering technique. The aim of this book is to provide an overview of recent developments in Kalman filter theory and their applications in engineering and science. The book is divided into 20 chapters corresponding to recent advances in the filed.

Photo by monsitj / iStock

IntechOpen