Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Multiscale Fuzzy State Estimation using Stationary Wavelet Transforms

Hazem N. Nounou, *Member, IEEE*

Abstract— **Multiscale representation of data is a powerful data analysis tool, which has been successfully used to solve several data filtering problems. For nonlinear systems, which can be represented by a Takagi-Sugeno fuzzy model, several Fuzzy Kalman filtering algorithms have been developed to extend Kalman filtering for such systems. In this paper, a multiscale Fuzzy Kalman (MSFK) filtering algorithm, in which multiscale representation is utilized to improve the performance of Fuzzy Kalman filtering, is developed. The idea is to apply fuzzy Kalman filtering at multiple scales to combine its advantages with those of the low pass filters used in multiscale data representation. Starting with a fuzzy model in the time domain, a similar fuzzy model is derived at each scale using the scaled signal approximation of the data obtained by stationary wavelet transform (SWT). These multiscale fuzzy models are then used in fuzzy Kalman filtering, and the fuzzy Kalman filter with the least cross validation mean square error among all scales is selected as the optimum filter. Finally, the performance of the developed MSFK filtering algorithm is illustrated through a simulated example.**

I. Introduction

Fuzzy systems have been widely used and are known to perform well in modeling nonlinear dynamical systems because of their accuracy and ability to incorporate pre-defined knowledge into their estimation [1-6]. A fuzzy system is an approximator which consists of a set of IF-THEN type rules, each of which has a premise and a consequent part. In standard fuzzy systems, the consequent part is a scalar. A more general class of fuzzy systems includes the functional fuzzy systems, which are usually referred to as the Takagi-Sugeno (TS) fuzzy systems [1,7]. In such systems, the consequent part is a crisp function, which gives the approximator the ability to incorporate knowledge about the model.

Fuzzy models have been found very useful for control purposes due to their ability to describe complex systems in an efficient manner. However, in order to achieve good fuzzy control, reliable state estimation is essential. The problem of state estimation in systems which can be described by fuzzy models has received little attention by researchers. For example, the authors in [8-10] have addressed the noise-free case of the fuzzy state estimation problem, in which the system is assumed to be unaffected by noise. Also, they require a solution of set of Riccati equations, which sometimes are very challenging to solve. Other researchers dealt with the noise corrupted systems. For example, the author in [11] developed a TS fuzzy Kalman filtering algorithm, in which local Kalman filters are designed for all local sub-models, which are then interpolated to give the overall state estimate. Also, the authors in [12] developed a fuzzy version of the interval Kalman filter. Their developed algorithm, which they refer to as the fuzzy Kalman filter, provides scalar estimates of the states and possesses the same recursive mechanism as the standard Kalman filter.

Unfortunately, the above fuzzy filtering techniques are single scale methods because they all assume that the measured process data only contain features with fixed contributions over time and frequency. In practice, however, measured data usually contain multiscale features. For example, a sudden change in the data spans a wide range in the frequency domain and a narrow range in the time domain, while a slow change spans a wide range in the time domain and narrow range in the frequency domain. Filtering such data using single scale methods is not very effective because they do not account for the multiscale nature of the data. Thus, effective filtering of multiscale data requires a multiscale filtering method.

The objective of this paper is to develop a Multiscale Fuzzy Kalman (MSFK) filtering algorithm that combines the advantages of multiscale filtering with those of the Fuzzy Kalman filter to further improve its performance. The MSFK algorithm relies on applying fuzzy Kalman filtering at multiple scales using the scaling function coefficients of the data obtained using Stationary Wavelet Transform (SWT), and then selecting the optimum fuzzy Kalman filter, among all scales, which minimizes a cross validation estimation error criterion. The multiscale model used in this MSFK filtering approach is also derived using stationary wavelet transform, which is not restricted to any filter type.

The rest of the paper is organized as follows. In Section II, the problem of multiscale fuzzy Kalman filtering is specifically stated for the TS fuzzy model. Then, in Section III, a brief description of the Fuzzy Kalman filter is presented, followed by an introduction to multiscale representation of data and a derivation of the multiscale fuzzy state space model in Section IV. Then, in Section V, a

H. N. Nounou is with the department of Electrical Engineering at the United Arab Emirates University, Al-Ain, UAE. P.O. Box 17555 (phone: +9713-713-3061; e-mail: hnounou@uaeu.ac.ae).

0-7803-9568-9/05/$20.00 ©2005 IEEE

derivation of the MSFK filter equations is presented, followed by an outline of the MSFK filtering algorithm. Then, the performance of the MSFK filter is demonstrated and compared to that of the standard fuzzy Kalman filter through a simulated example in Section VI, followed by few concluding remarks in Section VII.

II. Problem Statement

In this paper, the problem of state estimation is addressed from a multiscale perspective for the nonlinear dynamic systems which can be described by the following Takagi-Sugeno fuzzy state space model, whose i[th] rule is defined as:

If $\bar{z}_1(k)$ is $\bar{\alpha}_{1,r1}$ and $\bar{z}_2(k)$ is $\bar{\alpha}_{2,r2}$ …and $\bar{z}_p(k)$ is $\bar{\alpha}_{p,r}$

Then, $x_i(k+1) = A_i x(k) + B_i u(k) + \Phi_i w(k)$,

$$y_i(k) = C_i x(k) + \nu(k). \qquad \forall i = 1,\ldots,M \tag{1}$$

where, M is the number of rules. Here, $x(k) \in \Re^n$, $u(k) \in \Re^m$, $y(k) \in \Re^p$, $\nu(k) \sim N(0,R) \in \Re^p$, and $w(k) \sim N(0,Q) \in \Re^n$ are the actual state, process input, measured output, measurement noise, and process noise, respectively. The premise of each rule is defined such that $\bar{\alpha}_{p,r}$ is the r[th] linguistic value of the linguistic variable, $\bar{z}_p(k)$, defined over the universe of discourse, U_p. The input to the fuzzy system, $\bar{z}(k) = [\bar{z}_1, \bar{z}_2, \cdots, \bar{z}_p]^T$, is a p-dimensional vector that can be a function of past states, inputs, outputs, or any external input. On the other hand, the consequent part of each rule is a crisp function that is defined as a state space discrete time-invariant system. Assume that $\xi_i(\bar{z}(k))$ represents the certainty that the premise of the i[th] rule matches the input information. For simplicity, $\xi_i(k)$ is used instead of $\xi_i(\bar{z}(k))$. Assuming that $0 \le \xi_i(k) \le 1, \forall i = 1,2,\cdots,M$, and $\sum_{i=1}^{M} \xi_i(k) \neq 0$, then $\mu_i(k)$, which is the certainty of the i[th] rule, can be defined as, $\mu_i(k) = \frac{\xi_i(k)}{\sum_{i=1}^{M} \xi_i(k)}$, where $\sum_{i=1}^{M} \mu_i(k) = 1$. With this in mind, the overall fuzzy model which can be thought of as nonlinear interpolator between M linear systems such that $x(k) = \sum_{i=1}^{M} \mu_i(k) x_i(k)$ and $y(k) = \sum_{i=1}^{M} \mu_i(k) y_i(k)$. Hence, the Takagi-Sugeno fuzzy system can be expressed as,

$$x(k+1) = \sum_{i=1}^{M} \mu_i(k)[A_i x(k) + B_i u(k) + \Phi_i w(k)]$$
$$y(k) = \sum_{i=1}^{M} \mu_i(k)[C_i x(k) + \nu(k)], \tag{2}$$

or equivalently,

$$x(k+1) = \bar{A}(k)x(k) + \bar{B}(k)u(k) + \bar{\Phi}(k)w(k)$$
$$y(k) = \bar{C}(k)x(k) + \nu(k), \tag{3}$$

where, $\bar{A}(k) = \sum_{i=1}^{M} \mu_i(k) A_i$, $\bar{B}(k) = \sum_{i=1}^{M} \mu_i(k) B_i$, $\bar{C}(k) = \sum_{i=1}^{M} \mu_i(k) C_i$, and $\bar{\Phi}(k) = \sum_{i=1}^{M} \mu_i(k) \Phi_i$.

III. Fuzzy Kalman (FK) Filtering

Several approaches have been used to derive the Kalman filter equations for nonlinear systems represented by fuzzy models [11,12]. In this section, the fuzzy Kalman filtering equations are derived for the Takagi-Sugeno fuzzy state space model defined in equation (3). The system represented by equation (3) can be thought of as a time-varying linear state space model, for which the matrices, $\{\bar{A}, \bar{B}, \bar{C}, \bar{\Phi}\}$, are time varying since they depend on the local or sub-models certainties $\mu_i(k)$. Therefore, the derivation of the fuzzy Kalman filter equations can follow a similar approach to that used for linear time invariant (LTI) systems. The basic idea is that the Kalman filter recursively estimates the state at any time instant in two steps (prediction and correction) such that the trace of the estimation error covariance matrix is minimized. In the prediction step, the state space model shown in equation (3) is used to predict the state at time instant $(k+1)$ given the estimated state at the previous time instant as follows,

$$x^-(k+1) = \bar{A}(k)\hat{x}(k) + \bar{B}(k)u(k). \tag{4}$$

In the second step, the estimated state shown in equation (4) is corrected using the following correction equation,

$$\hat{x}(k+1) = x^-(k+1) + K(k+1)\{y(k+1) - \bar{C}(k+1)x^-(k+1)\} \tag{5}$$

where, $y(k+1)$ is the measured process output, $K(k+1)$ is the Kalman filter gain, which can be determined by minimizing the trace of the estimation error covariance matrix. Defining the estimation error vector as,

$$e(k+1) = \hat{x}(k+1) - x(k+1), \tag{6}$$

its covariance matrix can be written as,

$$P(k+1) = E[e(k+1)e^T(k+1)], \tag{7}$$

which can be easily expressed as,

$$P(k+1) = \{I - K(k+1)\bar{C}(k+1)\}P^-(k+1)\{I - K(k+1)\bar{C}(k+1)\}^T$$
$$+ K(k+1)RK(k+1)^T, \tag{8}$$

where, $P^-(k+1) = E[e^-(k+1)e^-(k+1)^T]$

and, $e^-(k+1) = x^-(k+1) - x(k+1)$.

Taking the partial derivative of the trace of $P(k+1)$ with respect to the Kalman filter gain and then setting it to zero gives the following expression for the Kalman filter gain,

$$K(k+1) = P^-(k+1)\overline{C}(k+1)^T\left[\overline{C}(k+1)P^-(k+1)\overline{C}(k+1)^T + R\right]^{-1}. \quad (9)$$

Substituting equation (9) into equation (8), the following update equation for the error covariance matrix is obtained,

$$P(k+1) = [I - K(k+1)\widetilde{C}(k+1)]P^-(k+1). \quad (10)$$

Thus, the fuzzy Kalman filtering algorithm can be outlined as follows,

1. Prediction Step (given $\hat{x}(k)$ and $P(k)$):
   a. Predict the state ahead using equation (4).
   b. Project the error covariance matrix as follows,
   $$P^-(k+1) = \overline{A}(k)P(k)\overline{A}(k)^T + \overline{\Phi}(k)Q\overline{\Phi}(k)^T.$$
2. Correction Step (given $y(k+1)$):
   a. Compute the FK filter gain using eq. (9).
   b. Correct the state vector estimate using eq. (5).
   c. Update the estimation error covariance matrix using equation (10).
   d. Set $k = k+1$ and go to step 1.

IV. Multiscale Data and Fuzzy State Space Model Representations

Real system measurements are usually multiscale in nature, which means that they contain features with varying contributions over time and frequency. Therefore, a proper analysis of such measurements requires their representation at multiple scales, which can be achieved by expressing the data as a weighted sum of orthornomal basis functions, such as wavelets. In this section, some of the wavelet representation algorithms are introduced [13,14].

A. Discrete Wavelet Transform (DWT)

A discrete dyadic signal can be represented at multiple resolutions by decomposing it on a family of wavelet and scaling functions. For example, a coarser approximation (also called the scaled signal) of a signal $x(t)$, at any scale, can be computed by projecting that signal on a set of orthonormal scaling functions of the form, $\phi_{jk}(t) = \sqrt{2^{-j}}\phi(2^{-j}t - k)$, or equivalently by filtering the signal using a low pass filter of length r, $h = [h_1 \quad h_2 \quad . \quad . \quad h_r]$, derived from the scaling functions. On the other hand, the difference between the original signal and its coarser approximation (also called the detail signal), can be computed by projecting the signal on a set of wavelet basis functions of the form, $\psi_{jk}(t) = \sqrt{2^{-j}}\psi(2^{-j}t - k)$, or equivalently by filtering the original signal using a high pass filter of length r, $g = [g_1 \quad g_2 \quad . \quad . \quad g_r]$, derived from the wavelet basis functions. Repeating the process to any scale, J, the original signal can be represented as the sum of all detail signals and the last scaled signal as follows,

$$x(t) = \sum_{k=1}^{n2^{-J}} x_{Jk}\phi_{Jk}(t) + \sum_{j=1}^{J}\sum_{k=1}^{n2^{-j}} dx_{jk}\psi_{jk}(t), \quad (11)$$

where, j, k, J, and n are the dilation parameter, translation parameter, maximum number of scales, and the length of the original signal, respectively. Fast wavelet transform algorithms of $O(n)$ complexity for a discrete signal of dyadic length have been developed [13].

B. Stationary Wavelet Transform (SWT)

In the DWT algorithm described above, the length of the scaled and detail signals decreases dyadically (by half) with scale, due to the down-sampling of the convoluted scaled signal. This can become a problem in certain applications which require large data sets, such as image processing. Also, DWT is time-variant, which means that in some cases, the multiscale representation of a certain feature in the original signal depends on the location of that feature in time. A general solution, which results in both time-invariant and fixed length decomposition at all scales, is the stationary wavelet transform (SWT) [14]. SWT does not use down-sampling as in DWT. Instead, it up-samples the low pass and high pass filters at every subsequent coarser scale before convoluting the filter with the data. Therefore, the filter length increases dyadically at coarser scales, but the lengths of the scaled and detail signals stay the same at all scales.

As an example to illustrate the SWT algorithm, consider the following time domain discrete signal,

$$X_o = [x_o(1) \quad x_o(2) \quad . \quad x_o(k) \quad x_o(k+1) \quad . \quad x_o(n)] \quad (12)$$

Assume that the original low pass and high pass filters are of length (r), as in DWT, i.e., $h = [h_1 \quad h_2 \quad . \quad . \quad h_r]$, and $g = [g_1 \quad g_2 \quad . \quad . \quad g_r]$. Then, the filters that are used to obtain the SWT scaled and detail signals at scale (j) from the scaled signal at scale ($j-1$), are of length $p = 2^{j-1}r$, because of the up-sampling of the filters at each subsequent coarser scale. For example, the low pass and high pass filter used to obtain the *second* scale SWT coefficients from the first scale coefficients are of length ($2r$), and are of the form,

$$\begin{aligned}\overline{h}^2 &= [\overline{h}_1^2 \quad \overline{h}_2^2 \quad \overline{h}_3^2 \quad . \quad . \quad . \quad \overline{h}_{p-1}^2 \quad \overline{h}_p^2] \\ &= [0 \quad h_1 \quad 0 \quad h_2 \quad . \quad . \quad 0 \quad h_r]\end{aligned} \quad (13)$$

and,

$$\bar{g}^2 = [\bar{g}_1^2 \quad \bar{g}_2^2 \quad \bar{g}_3^2 \quad . \quad . \quad . \quad \bar{g}_{p-1}^2 \quad \bar{g}_p^2] \\ = [0 \quad g_1 \quad 0 \quad g_2 \quad . \quad . \quad 0 \quad g_r] \quad (14)$$

respectively, where the superscript "2" denotes the second scale, $j = 2$. Now, these filters can be used to compute the SWT scaling function coefficients at scale (j) at time steps k and $k+1$, $x_j(k)$ and $x_j(k+1)$, as follows,

$$x_j(k) = \bar{h}_1^j x_{j-1}(k-p+1) + \bar{h}_2^j x_{j-1}(k-p+2) + ... + \bar{h}_p^j x_{j-1}(k), \quad (15)$$

and, $$x_j(k+1) = \bar{h}_1^j x_{j-1}(k-p+2) + \bar{h}_2^j x_{j-1}(k-p+3) + ... + \bar{h}_p^j x_{j-1}(k+1). \quad (16)$$

The notation used in this SWT wavelet decomposition algorithm will be used in the derivation of SWT-based multiscale state space models in the next section.

C. State Space Modeling using SWT

In this section, a multiscale version of the Takagi-Sugeno fuzzy state space model relating the scaled signal approximations of the data at any scale (j) is derived using SWT given the state space model in the time domain. Starting with the individual i^{th} local model given in eq. (1),

$$x_{i,o}(k+1) = A_i x_o(k) + B_i u_o(k) + \Phi_i w_o(k), \\ y_{i,o}(k) = C_i x_o(k) + v_o(k)., \quad (17)$$

where, $w_o(k) \sim N(0,Q)$, $v_o(k) \sim N(0,R)$, and the subscript "o" denotes scale zero or the time domain, it can be shown that the above model can also be used to relate the scales signal approximations of the data obtained using SWT as stated in the following Theorem.

Theorem 1:

If the i^{th} sub-model shown in equation (17) can be used to relate the time domain data, then the scaled signal approximations of the data obtained using SWT at any scale (j) can be related by the following state space model,

$$x_{i,j}(k+1) = A_i x_j(k) + B_i u_j(k) + \Phi_i w_j(k), \\ y_{i,j}(k) = C_i x_j(k) + v_j(k).,$$

where, $w_j \sim N(0,Q)$, and $v_j \sim N(0,R)$. (18)

Theorem 1, which can be proved by substituting equation (18) into equation (16) for the i^{th} model, shows that if the time domain data are related by a discrete forced linear state space model, then the same state space model will be valid at all scales.

Now since all individual state space models used in the TS-fuzzy systems are valid at all scales, then these individual models can be used to define a new fuzzy model whose i^{th} rule has the following form,

If $\bar{z}_{1,j}(k)$ is $\bar{\alpha}_{1,r1,j}$ and $\bar{z}_{2,j}(k)$ is $\bar{\alpha}_{2,r2,j}$ ..and $\bar{z}_{p,j}(k)$ is $\bar{\alpha}_{p,r,j}$

Then, $x_{i,j}(k+1) = A_i x_j(k) + B_i u_j(k) + \Phi_i w_j(k)$,

$$y_{i,j}(k) = C_i x_j(k) + v_j(k). \quad \forall i = 1,...,M \quad (19)$$

The above TS-fuzzy model can also be written similar to equation (3) as follows,

$$x_j(k+1) = \bar{A}_j(k) x_j(k) + \bar{B}_j(k) u_j(k) + \bar{\Phi}_j(k) w_j(k) \\ y_j(k) = \bar{C}_j(k) x_j(k) + v_j(k),$$

where, $\bar{A}_j(k) = \sum_{i=1}^{M} \mu_{i,j}(k) A_i$, $\bar{B}_j(k) = \sum_{i=1}^{M} \mu_{i,j}(k) B_i$,

$\bar{C}_j(k) = \sum_{i=1}^{M} \mu_{i,j}(k) C_i$, and $\bar{\Phi}_j(k) = \sum_{i=1}^{M} \mu_{i,j}(k) \Phi_i$. (20)

Note that since the data are transformed at multiple scales, the sub-models certainties, $\mu_{i,j}$, are different from those of the time domain fuzzy model. However, the multiscale TS-fuzzy model shown in equation (20) has the same form as the time domain TS-fuzzy model, and thus can be directly used to derive the MSFK filter as in the case of the conventional or time domain fuzzy Kalman filter.

V. Multiscale Fuzzy Kalman (MSFK) Filtering

In this section, the multiscale TS-fuzzy model derived in Section IV.C is used to develop a multiscale fuzzy Kalman filtering algorithm that combines the advantages of the conventional time-domain fuzzy Kalman filter with those of multiscale representation of data to further enhance its state estimation accuracy. The MSFK filtering algorithm relies on decomposing the data at multiple scales using SWT, and then using the scaled signal approximations of the data along with the multiscale state space models derived earlier to obtain estimates of the plant states at each scale. The estimated states are then reconstructed to the time domain, and the states obtained from the scale that minimizes a cross validation mean square error criterion are taken as the optimum estimates.

A. Derivation of the MSFK Equations

In this section, the MSFK filter equations are derived using the SWT scaled signal approximations of the data (x_j, y_j, and u_j) and the multiscale TS-fuzzy model, relating these variables at scale (j). Since the multiscale TS-fuzzy model has the same form as the time domain fuzzy model, it is expected that the MSFK filter equations to be very similar to the time domain ones. For the prediction step, the predicted state at scale (j) can be computed as follows,

$$x_j^-(k+1) = \overline{A}_j(k)\hat{x}_j(k) + \overline{B}_j(k)u_j(k). \quad (21)$$

Then, the predicted state is updated using the SWT scaling coefficient of the measured output at scale (j) as follows,

$$\hat{x}_j(k+1) = x_j^-(k+1) + K_j(k+1)\{y_j(k+1) - \overline{C}_j(k+1)x_j^-(k+1)\} \quad (22)$$

where, $K_j(k+1)$ is the fuzzy Kalman filter gain at scale (j). Let the estimation error covariance matrix at scale (j) be defined as,

$$P_j(k+1) = E[e_j(k+1)e_j(k+1)^T], \quad (23)$$

where, $e_j(k+1) = \hat{x}_j(k+1) - x_j(k+1)$.

Then, the fuzzy Kalman filter gain at scale (j) can be determined by taking the partial derivative of $P_j(k+1)$ and setting it to zero, which gives,

$$K_j(k+1) = P_j^-(k+1)\overline{C}(k+1)^T\left[\overline{C}(k+1)P_j^-(k+1)\overline{C}(k+1)^T + R\right]^{-1}. \quad (24)$$

where, $P_j^-(k+1) = E[e_j^-(k+1)e_j^-(k+1)^T]$

and $e_j^-(k+1) = x_j^-(k+1) - x_j(k+1)$.

B. MSFK Algorithm

The MSFK filtering algorithm can be outlined as follows,

1) Decompose the input, output, and state data at multiple scales using the wavelet and scaling function filter of choice.
2) At each scale, apply Kalman filtering as follows:
   a) Given previously estimated states and measured output, compute the time-varying system matrices, $\{\overline{A}_j(k), \overline{B}_j(k), \overline{C}_j(k)$, and $\overline{\Phi}_j(k)\}$ using equation (20).
   b) Prediction step (given $x_j(k)$ and $P_j(k)$) :
      i) Predict the state ahead using equation (21).
      ii) Project the error covariance matrix as follows,
      $$P_j^-(k+1) = \overline{A}_j(k)P_j(k)\overline{A}_j(k)^T + Q$$
   c) Correction step (given $y_j(k+1)$) :
      i) Compute the Kalman filter gain using eq. (24).
      ii) Correct the state vector estimate using eq. (22).
      iii) Update the estimation error covariance matrix as follows,
      $$P_j(k+1) = [I - K_j(k+1)\overline{C}_j(k)]P_j^-(k+1)$$
      iv) Set $k = k+1$ and go to step (a).
3) Reconstruct the estimated state and output vectors, $\hat{x}_j$ and $\hat{y}_j$, from all scales.
4) Compute the output cross validation mean square error at each scale (j) as follows [23]:

$$CVMSE(j) = (1/n)\sum_{k=1}^{n}\left[(y^*(k) - \hat{y}(k))^2\right], \quad (25)$$

where, $y^*(k) = (1/2)[y(k-1) + y(k+1)]$.

5) Select the MSFK filter with the minimum cross validation mean square error as the optimum filter.

Note that in the above algorithm, the optimum filtering scale is selected such that the estimated output minimizes a cross validation mean square error with respect to a pre-smoothed output signal (not the measured one), which is suggested in [15]. The rational behind this choice is that the optimum MSFK filter should minimize the output mean square error with respect to the noise-free data, regardless of the measurement noise content.

VI. Illustrative Example

The fuzzy model used in this simulated example represents the rotation of a human leg that pivots from the hip joint. The nonlinear dynamical model describing the human leg can be expressed by the following differential equation [16],

$$J\frac{d^2\theta(t)}{dt^2} + D\frac{d\theta(t)}{dt} + \frac{1}{2}MgL\sin[\theta(t)] = T_m(t), \quad (26)$$

which relates the output angular rotation about the hip joint, $\theta(t)$, to the applied input muscular torque, $T_m(t)$. Also, the parameters, D, J, M, L, and g represent the viscous damping at the hip joint, the inertia around the hip joint, the mass of the leg, the length of the leg, and the gravitational acceleration, respectively.

Here, it is assumed that the state vector $x^T(t) = [x_1(t) \quad x_2(t)] = [\theta(t) \quad \dot{\theta}(t)]$, the input $u(t) = T_m(t)$, and the output $y(t) = \theta(t)$. By linearizing the system at the following three operating conditions, $\{x^T{}_{01} = [0 \quad 0], u_{01} = 0\}$, $\{x^T{}_{02} = [\pi/4 \quad 0], u_{02} = 0\}$, and $\{x^T{}_{03} = [-\pi/4 \quad 0], u_{03} = 0\}$, and then discretizing it with a sampling time, $T_s = 0.1s$, two linear discrete-time models are obtained. Assuming the following model parameters: $D = 20Nms$, $J = 5kgm^2$, $M = 10kg$, $L = 0.5m$, and $g = 9.8m/s^2$, the overall fuzzy model can be written as,

$$x(k+1) = \overline{A}(k)x(k) + \overline{B}(k)u(k) + \overline{\Phi}(k)w(k)$$
$$y(k) = \overline{C}(k)x(k) + v(k), \quad (27)$$

where the matrices $\{\overline{A}, \overline{B}, \overline{C}, \overline{\Phi}\}$ are defined as in eq. (3). The state space matrices are found to be:

$$A_1 = \begin{bmatrix} 0.9785 & 0.0818 \\ -0.4006 & 0.6515 \end{bmatrix}, B_1 = \begin{bmatrix} 0.0009 \\ 0.0164 \end{bmatrix}, C_1 = [1 \quad 0],$$

$$A_2 = \begin{bmatrix} 0.9848 & 0.0819 \\ -0.2839 & 0.6570 \end{bmatrix}, B_2 = \begin{bmatrix} 0.0009 \\ 0.0164 \end{bmatrix}, \text{and } C_2 = [1 \quad 0]. \quad (28)$$

Also, both of the matrices, Φ_1 and Φ_2, are assumed to be, $0.01I$, and the membership function certainties are:

$$\mu_1(x) = \left[1 - \frac{1}{1+\exp[-15(x_1 - \pi/8)]}\right]\left[\frac{1}{1+\exp[-15(x_1 + \pi/8)]}\right],$$

and, $$\mu_2(x) = 1 - \mu_1(x). \quad (29)$$

In this example, it is desired to estimate the angular rotation, which is the first state (x_1), given measurement of the output, which also represent the same variable. The data used in this simulation are generated by applying a pseudo random binary sequence (PRBS) changing between -10 and 10 as an input to the fuzzy model described above to generate an output, which is assumed to be noise-free. Then, this noise-free output is contaminated with zero-mean Gaussian noise of variance R, representing the output measurement noise. Different measurement noise variances are used ($R = 0.01$, 0.05, and 0.1, which correspond to output signal to noise ratios of 18, 3.6, and 1.8, respectively) to test the robustness of the MSFK filtering algorithm under different levels of noise.

For statistically valid comparisons between the fuzzy Kalman and multiscale fuzzy Kalman filters, a Monte Carlo simulation of 100 realizations is performed. Table I, which lists the estimation means square errors with respect to their noise-free values shows that the MSFK filter outperforms the Fuzzy Kalman filter for all noise levels, especially for the state of interest (first state), which represents the angular rotation. This advantage can be clearly seen from Figure 3,

TABLE I
COMPARISON OF THE MEAN SQUARE ERRORS OF THE ESTIMATED STATES USING THE FUZZY KALMAN AND MSFK FILTERS

	$\sigma_e = 0.01$		$\sigma_e = 0.05$		$\sigma_e = 0.1$	
	x_1	x_2	x_1	x_2	x_1	x_2
FK	5.3	2.9	10.5	9.3	13.0	11.9
MSFK	2.6	2.8	4.8	4.5	6.3	9.4

which compares the filtered states for the case where $R =$ 0.05.

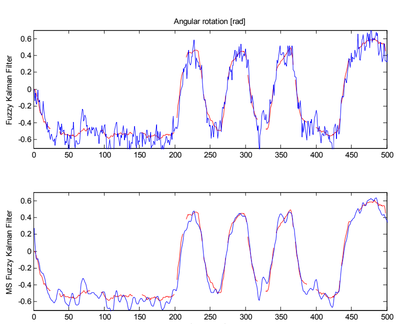

Fig. 3. Comparison of the state estimation by the Fuzzy and multiscale fuzzy Kalman filters for the case where $R =$ 0.05.

VII. CONCLUSIONS

This paper presents a multiscale fuzzy Kalman (MSFK) filtering algorithm, which enhances the performance of the conventional fuzzy Kalman filter by its application at multiple scales using the scaled signal representations of the data obtained from stationary wavelet transform (SWT). First, multiscale fuzzy models are derived at each scale given the time domain fuzzy model representation of the system. Then, these models are used to implement fuzzy Kalman filtering at all scales, and the filter that minimizes a cross validation mean square error criterion is selected as the optimum filter. The MSFK filtering algorithm is shown to outperform the conventional fuzzy Kalman filter through a simulated example, and the reason behind this enhancement is that MSFK filtering combines the advantages of both fuzzy Kalman filtering and multiscale data smoothing.

REFERENCES

[1] Passino, K.M., and Yurkovich, S., "Fuzzy Control", Addison-Wesley Longman, Menlo Park, CA, 1998.
[2] Johansen, T., Shorten, R., and Murray-Smith, R., "On the interpretation and Identification of Dynamic Takagi-Sugeno Fuzzy Models, IEEE Trans. Fuzzy Syst., 8, 3, 297-313, 2000.
[3] Ying H., "Sufficient Conditions on Uniform Approximation of Multivariate Functions by General Takagi-Sugeno Fuzzy Systems With Linear Rule Consequent", IEEE Trans. Man Cybernetics, 28, 515-520, 1998.
[4] Ying, H., "General SISO Takagi-Sugeno Fuzzy Systems with Linear Rule Consequent are Universal Approximators", IEEE Trans. Fuzzy Sys., 6, 582-587, 1998.
[5] Ying, H., "General Takagi-Sugeno Fuzzy Systems with Simplified Linear Rule Consequent are Universal Controllers", J. Inf. Sci., 108, 91-107, 1998.
[6] Zeng. K., Zhang, N., and Xu, W., "A Comparative Study on Sufficient conditions for Takagi-Sugeno Fuzzy Systems as Universal Approximators", IEEE Trans. Fuzzy Syst., 8, 6, 773-780, 2000.
[7] Takagi, T. and Sugeno, M., "Fuzzy Identification of Systems and its Application to Modeling and Control", IEEE Transactions on Systems, Man, and Cybernetics, 15, 1, 116-132, 1985.
[8] Feng, G., Cao, S., Rees, N., and Chak, C., "Design of Fuzzy Control Systems with Guaranteed Stability", Fuzzy Sets Sys., 85, 1-10, 1997.
[9] Ma, X., Sun, Z., and He, Y., "Analysis and Design of Fuzzy Controller and Fuzzy Observer", IEEE Trans. Fuzzy Sys, 6, 1, 41-51, 1998.
[10] Tanaka, K., Ikeda, T., and Wang, H., "Fuzzy Regulators and Fuzzy Observers: Relaxed Stability Conditions and LMI-Based Designs", IEEE Trans. Fuzzy Sys., 6, 2, 250-265, 1998.
[11] Simon, D., "Kalman Filtering of Fuzzy Discrete Time Dynamic Systems", Applied Soft Computing, 3, 191-207, 2003.
[12] Chen, G., Xie, Q., and shieh, L., "Fuzzy Kalman Filtering", J. Inform. Sci., 109, 197-209, 1998.
[13] Mallat, S. G., "A Theory of Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Transaction on Pattern Analysis and Machine Intelligence, 11, 7, 764, 1989.
[14] Nason, G. and Silverman, B., "The Stationary Wavelet Transform and some Statistical Applications", Lecture Notes in Statistics: Wavelets and Statistics, 281, 1995.
[15] Nason, G., "Wavelet shrinkage using cross-validation", J. R. Statist. Soc. B, 58, 463, 1996.
[16] Nise, Norman S., "Control Systems Engineering", Fourth edition, John Wiley and Sons, Inc., 2004.