Tracking a Maneuvering Target Using Neural Fuzzy Network

Fun-Bin Duh and Chin-Teng Lin, *Senior Member, IEEE*

***Abstract*—A fast target maneuver detecting and highly accurate tracking technique using a neural fuzzy network based on Kalman filter is proposed in this paper. In the automatic target tracking system, there exists an important and difficult problem: how to detect the target maneuvers and fast response to avoid miss-tracking? The traditional maneuver detection algorithms, such as variable dimension filter (VDF) and input estimation (IE) etc., are computation intensive and difficult to implement in real time. To solve this problem, neural network algorithms have been issued recently. However, the normal neural networks such as backpropagation networks usually produce the extra problems of low convergence speed and/or large network size. Furthermore, the way to decide the network structure is heuristic. To overcome these defects and to make use of neural learning ability, a developed standard Kalman filter with a self-constructing neural fuzzy inference network (KF-SONFIN) algorithm for target tracking is presented in this paper. By generating possible target trajectories including maneuver information to train the SONFIN, the trained SONFIN can detect when the maneuver occurred, the magnitude of maneuver values and when the maneuver disappeared. Without having to change the structure of Kalman filter nor modeling the maneuvering target, this new algorithm, SONFIN, can always find itself an economic network size with a fast learning process. Simulation results show that the KF-SONFIN is superior to the traditional IE and VDF methods in estimation accuracy.**

***Index Terms*—Doppler shift, feature extraction, Kalman filter, maneuvering, neural fuzzy network, system covariance, target tracking.**

I. Introduction

FOR the surveillance purpose, a radar system, such as plane pulsed search radar, is installed to search for targets and provide reliable detection within the given region. The radar can measure the range of the target by calculating the delay time between the transmission of a pulse signal and the reception of the echo of target, the bearing of the target by the angle measurement device such as synchro and radial component of the target speed by the measurement of the Doppler frequency shift. However, the search radar isn't provided with the ability of target tracking and the operator must decide whether targets detected on the current scan are the same as that detected on a previous scan or scans. The track-while-scan (TWS) system is constructed for automatic target tracking in a plane pulsed search radar system. With the function of TWS, the search radar can track a target or targets while scanning. Such an elegant combination of searching and tracking in the radar system is referred to the TWS radar [1]–[3] as shown in Fig. 1. In the TWS radar system, the radar signals' pulse repetitive frequency (PRF), video echo and bearing are processed by the radar signal processor and plot extractor. The former is to reduce the noise interference and degrades the effect of clutter, while the latter extracts the plot position data which are then sent to the radar data processor by bearing and range sequences at a discrete interval of time. As the searching radar passes by each target of interest, the radar data processor, which is with the TWS program, gives a track and predicts the target's trajectory [4], [5].

The main function of the radar data processor is to execute the tracking program in which the tracking algorithms are developed such as α-β filter [5], [6], α-β-γ filter [6], [7] and Kalman filter [8]–[15], [23]–[40]. The performance of a tracking algorithm is mainly governed by the performance of the state estimator. The Kalman filter is a recursive algorithm developed to solve the state estimation problems of a known system based on certain assumptions about the system's mathematic model. These assumptions include the input forcing functions and noise statistics. As we shall describe in Section II, the process of state estimation in the Kalman filter comprises two parallel iteration cycles that are estimation of the state and estimation of the state covariance. The Kalman filter has high complexity and large computation requirement, but it also has fast convergence ability and an optimal mean-square-error (MSE) filter process. Due to the faster computing speed of current computers, more and more systems use the Kalman filter to track the target. In practical systems there are many factors originating from the tracked targets and the tracking system that lead to target loss. The tracking performance degradation arising from the tracking system itself can be improved by promoting the sensor's accuracy, increasing the signal processing ability to decrease the measurement noise input and setting the parameters of the tracking filter properly. However, in spite of the recent advances in sensor technology, there are no devices that can detect the manned maneuvers of a tracked target in the surveillance and guidance systems. This sudden maneuver of a target implies to a tracking system that it is accelerating unexpectedly and that acceleration may be time-varying and following an unknown profile. Even a short-term acceleration can cause a bias in the measurement sequence and will result in divergence, if no compensations are used in time. This is the key reason that the simple Kalman filter always misses the tracked target if the target produces maneuvers. To solve this problem, a number of researchers have proposed techniques to modify the conventional Kalman filter for maneuvering target tracking.

Manuscript received May 14, 2001; revised March 7, 2002. This work was supported by the MOE Program for Promoting Academic Excellent of Universities under Grant 91-E-FA06-4-4, Taiwan, R.O.C. This paper was recommended by Associate Editor N. Pal.

The authors are with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C. (e-mail: william@falcon3.en.nctu.edu.tw).

Digital Object Identifier 10.1109/TSMCB.2003.810953

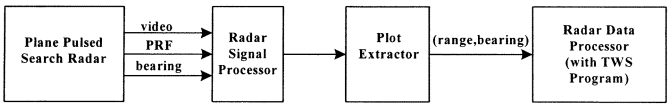


Fig. 1. Block diagram of the TWS radar system.

For good maneuvering target tracking, there are three key steps that must be done well to avoid miss-tracking. First, when the target maneuver occurs it must be detected immediately. Second, the magnitude of the maneuver value must be estimated accurately at as short time intervals as possible. Third, according to the magnitude of the maneuver value, the tracking filter state estimate must be compensated correctly. This is the mechanism to be developed for detecting, estimating and compensating the unexpected maneuvers in an automatic target tracking system. To get the best tracking performance the maneuver compensation must be done well and to get the best compensation the initial maneuver detection and estimation must be done well.

Based on these steps, several approaches have been proposed. Generally, the Kalman filter-based ones were used to track a maneuvering target by modifying the Kalman filter parameters or by using different structures to predict the maneuvers. These we call adaptive tracking filters. In the past three decades, several algorithms were proposed. Among these algorithms the most familiar adaptive filters are introduced as follows. For detecting and obtaining the maneuvers of a tracked target, Chan *et al.* [10] utilized the generalized least square estimation approach to estimate the acceleration input and used the estimate to update the Kalman filter directly. This algorithm is called input estimation (IE). It can track a constant velocity target to convergence very well, but under a noisy environment the target maneuver cannot be estimated accurately enough. A different approach by Bar-Shalom and Birmiwal [11] does not estimate the maneuver value to compensate the filter, but introduces extra state components in the state model when a maneuver is detected and reverts back to the quiescent state when it disappears. This is called variable dimension filter (VDF). It employs a sliding window to find the fading memory average of innovation from the estimator based on the quiescent state to detect the maneuver. However, because the fading memory average is unable to detect target maneuver immediately, it leads to a time delay. Moreover, the augmented state results in computational loading and the switches between the quiescent and nonquiescent states may give rise to a discontinuity problem. All of these will degrade the tracking performance, especially while the noise is changing greatly.

Recently, neural networks applied to maneuvering targets were proposed with information fusion capabilities and fuzzy logic schemes with intelligent adaptation were also proposed [12]–[19]. However, many complex factors must be considered when detecting target maneuvers. It is not easy to train the neural network well enough and the forward structure of fuzzy logic is neither easy to find the exact parameters nor is it easy to partition the parameters. Chin [12] employed the backpropagation (BP) neural network in a forward loop to aid the Kalman filter to reduce the estimation error. During operation, the output of the trained neural network is used to compensate the state estimate. This algorithm does not change the structure or parameter of the standard Kalman filter. The defect is that it will not be easy to compensate well enough when large errors are generated by the Kalman filter tracker.

Jing [13] did not attempt to find out the quality of maneuver to compensate the tracking filter, but rather he simply employed a trained neural network to adjust the system variance of the filter. He used a BP network to fuse all state information from two filters. All the fused state information of both filters is used as samples to train the BP network off-line. When in application, the trained BP network's output is used to adjust the filter variance according to the input fused data.

To cope with the drawbacks encountered in neural networks and fuzzy logic schemes whilst still keeping their advantages, a new novel solution to the problem of tracking the maneuvering target algorithm with a neural fuzzy network is proposed in this work. It is a self-constructing neural fuzzy inference network called SONFIN that we proposed previously in [20]. The SONFIN is a feedforward multilayer network that integrates the basic elements and functions of a traditional fuzzy system into a connectionist structure. In this connectionist structure, the input nodes represent the desired signal process and, in the hidden layers, there are nodes functioning as membership functions and fuzzy logic rules. The proposed algorithm can find the proper fuzzy logic rules dynamically on the fly. Also the SONFIN can always find itself an economical network size in high learning speed. Therefore, it can avoid the need of empirically determining the number of hidden layers and nodes in ordinary neural networks. Since the structure of the SONFIN is constructed from fuzzy IF–THEN rules, expert knowledge can be put into the network as a priori knowledge, which can usually increase its learning speed and estimation accuracy [21], [22]. These properties make SONFIN an attractive candidate for constructing an inverse mapping.

The SONFIN here is used as a feedforward mechanism in a closed loop with a Kalman filter and it is applied to approximate the function relationship between the target maneuvers and some features extracted from radar outputs and the innovation of the filter. With well trained SONFIN, the onset time and the quality of the maneuver can be estimated accurately. As well as the fact that the acceleration inputs into the Kalman filter can compensate the maneuvering bias directly, we can also modify the system variance adaptively according to the estimated maneuver quality so that both the Kalman filter structure and its parameter are compensated concurrently. The simulation results show that this algorithm produces a great accuracy with no time delay. The proposed scheme is called KF-SONFIN algorithm.

The rest of this paper is organized as follows. Section II gives the problem formulation. In Section III, the basic structure and function of the SONFIN is briefly introduced and the features input to SONFIN are proposed. The system covariance of Kalman filter is given also. Section IV describes the performance of KF-SONFIN by comparing it with the IE and VDF algorithms. Conclusions are summarized in Section V.

II. Problem Formulation

In the plane search radar, the measurement sequences are in polar coordinates for the range and the bearing. As the target

dynamics are best described in Cartesian coordinates, coordinates transformation is needed. This can be found in [1]. Here we assume the measurement sequences described in Cartesian coordinates are available.

Because of the measured data and computational loading, most of tracking systems using position and velocity two-state models are considered, but these models cannot reflect target maneuver actions practically. Models with target acceleration as a state variable can track the maneuvering target well as maneuver occurs, but in the absence of target maneuver such models tend to be inaccurate and waste the computational time. For example, if the target does not have acceleration, using a third model can only increase the estimation errors for both position and velocity. And if employing the jerk model [23] in which the third derivative of the target position exists, more errors will be generated. To improve the tracking performance of the maneuvering target, the acceleration inputs must be considered in the system model because the filter bias caused by the target maneuver implies that the target is just deviating from the assumed constant velocity, straight line motion. Unlike Bar-Shalom *et al.* [11] using two state models and switching them in the system, we model the maneuver as the driving input in this paper. This model is similar to that used by [10] and is attractive as it has a simple structure. It can compensate the maneuver bias directly and it does not have to assume any *a priori* knowledge of the maneuver target.

The problem of interest is described by the linear discrete time-invariant measurement equation as

$$\begin{aligned} \mathbf{x}(k+1) =& F\mathbf{x}(k) + G_1\mathbf{u}(k) + G_2\mathbf{v}(k) \qquad (1)\\ E\{\mathbf{v}(k)\mathbf{v}(j)^T\} =& Q(k)\delta_{kj}\\ \text{with } \delta_{kj} =& \begin{cases} 1 & k=j \\ 0 & k\neq j \end{cases}\\ \mathbf{z}(k+1) =& H\mathbf{x}(k+1) + \mathbf{w}(k+1) \qquad (2)\\ E\left\{\mathbf{w}(k)\mathbf{w}(j)^T\right\} =& R(k)\delta_{kj} \end{aligned}$$

where k is the time index, $\mathbf{x}(k) = [x(k), \dot{x}(k), y(k), \dot{y}(k)]^T$ is the state vector representing the relative positions and velocities of the target in the two-dimension plane, $\mathbf{z}(k)$ is the radar measurement vector, $\mathbf{u}(k) = [u_x(k), u_y(k)]^T$ is the input vector consisting of the acceleration components in the x and y directions and $\mathbf{v}(k)$ and $\mathbf{w}(k)$ are the process noise and measurement noise; both sequences are assumed to be uncorrelated with white Gaussian noise sequence with zero means and the variance matrices $Q(k)$ and $R(k)$, respectively.

In (1) and (2), F is the model state transition matrix, G_1 is the coupling matrix for maneuver inputs, G_2 is the process noise input matrix, is the model output matrix, H is the model output matrix. The related matrices are given by

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G_1 = G_2 = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix},$$
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

where T is the sampling time interval.

The variance matrices of system noise and measurement noise are given by

$$GQ(k)G^T = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & 0 & 0 \\ \frac{T^3}{2} & T^2 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^3}{2} \\ 0 & 0 & \frac{T^3}{2} & T^2 \end{bmatrix} \cdot q$$

where q is the variance of the noise process and

$$R(k) = \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} \end{bmatrix}$$

where R_{11} and R_{22} are predefined variances in the x and y directions, respectively. For adapting the maneuver, $Q(k)$ is changed to correspond to the magnitude of the maneuver. We will explain this in Section III.

According to the maneuver target model above, the Kalman filter receives the measurement sequence $\{\mathbf{z}(k)\}$, $k = 1, 2, 3, \ldots$, to get the minimum mean square error (MMSE) estimates and fast convergence by recursive performance. The recursive steps are derived as follows.

The one-step predicted state is formed by taking the expected value of state (1) conditioned on $\{\mathbf{z}(k)\}$ sequence and results in

$$\hat{\mathbf{x}}(k+1|k) = F\hat{\mathbf{x}}(k|k) + G_1\mathbf{u}(k). \qquad (3)$$

The state prediction error is

$$\begin{aligned} \tilde{\mathbf{x}}(k+1|k) =& \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\\ =& F\tilde{\mathbf{x}}(k|k) + G_2\mathbf{v}(k) \qquad (4) \end{aligned}$$

and the state prediction covariance becomes

$$P(k+1|k) = FP(k|k)F^T + G_1Q(k)G_1^T. \qquad (5)$$

The predicted measurement is

$$\hat{\mathbf{z}}(k+1|k) = H\hat{\mathbf{x}}(k+1|k). \qquad (6)$$

The measurement prediction error (or measurement residual) is

$$\begin{aligned} \tilde{\mathbf{z}}(k+1|k) =& \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k)\\ =& H\tilde{\mathbf{x}}(k+1|k) + \mathbf{w}(k+1) \qquad (7) \end{aligned}$$

and the measurement prediction covariance (or innovation covariance) is

$$S(k+1) = HP(k+1|k)H^T + R(k+1). \qquad (8)$$

The filter gain is defined as $E\{\tilde{\mathbf{x}}(k+1|k)\tilde{\mathbf{z}}(k+1|k)^T\}S(k+1)^{-1}$ and can be obtained by

$$W(k+1) = P(k+1|k)H^TS(k+1)^{-1}. \qquad (9)$$

The updated state estimate is

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + W(k+1)\tilde{\mathbf{z}}(k+1|k). \qquad (10)$$

Finally, the updated covariance is

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)^T. \qquad (11)$$

It is obvious that the modified Kalman filter equations above are similar to the simple Kalman filter except for the predicted state. The addition of $G_1\mathbf{u}(k)$ term in the predicted state is used to compensate the acceleration input. In the case that the acceleration input $\mathbf{u}(k)$ is unknown, especially the unexpected man-made maneuver motion, the filter will not produce good tracking performance (or even miss-tracking) unless the acceleration input is estimated immediately and correctly.

The key problem of the maneuvering target tracking in the modified Kalman filter is to estimate the acceleration input. The traditional algorithms to estimate the acceleration input, such as input estimation, utilize the measurement residual to find its least mean square over a finite sliding window for estimating onset time and magnitude of the target maneuver and then develop a decision logic to determine whether to update the state and covariance of the simple Kalman filter or not. The conventional methods produce time delay and require a significant amount of computation, because they cannot detect the target maneuver immediately. To improve these issues, a new approach is needed.

From a different point of view, the input acceleration estimation problem can be considered as a mapping of the feature vectors extracted from plot output to the estimated acceleration value for the Kalman filter

$$\hat{\mathbf{u}}(k) = f\Big(\Delta\theta(k), \Delta\dot{R}(k)\cos\alpha(k), \Delta\dot{R}(k)\sin\alpha(k), \tilde{z}_x(k), \tilde{z}_y(k)\Big) \quad (12)$$

where α is the angle between the target position and radar site, $\Delta\theta$ is the trajectory heading change, $\Delta\dot{R}(k)$ is the change of range rate and $\tilde{z}_x(k)$ and $\tilde{z}_y(k)$ are the measurement residuals in the x and y directions, respectively. We will explain these feature vectors in the following section.

From the intuitive justification, a target accelerating in both cross track and along track directions naturally generates the heading change of the target; the larger heading change is reflected by larger acceleration in cross track and/or along track directions and vice versa. It is a fact that a standard Kalman filter can track a nonmaneuvering target very well with a low measurement residual sequence, i.e., the Kalman filter can predict the next position of the target correctly. However, when a maneuver occurs, the measurement residual in the Kalman filter will increase rapidly and this is the reason that the Kalman filter cannot track the maneuvering target well. In other words, we can get the maneuvering information of the target from the observation of the measurement residual. The Doppler effect causes a shift in the frequency. When a radio wave is reflected from an object moving toward the radiating source, the wave is compressed. On the contrary, the wave is spread out, when the object is moving away from the radiating source. In the case of a Doppler radar, when the transmitted wave illuminates a moving target that has a radial velocity component relative to the radar, the signal of Doppler frequency shift is reflected from the target, which gives the information about the target's velocity. The greater the target's radial velocity, the greater the effect will be. From sensing the Doppler frequency shift and calculating its change, the Doppler radar can acquire the change of the target's radial velocity. This change implies the acceleration of the target.

In each recursive filtering step, the available feature vectors are fed into the trained network and the maneuver estimate can be obtained from the network output directly without any time delay. Once the maneuver is gained, the system noise covariance is adjusted according to the maneuver in that recursive step.

In our application, we use a neural fuzzy network (SONFIN) to approximate the mapping function f for acceleration estimation of a tracked target. As shown in (12), there are five input elements and one output vector in the SONFIN. Prior to this network working, it must be well-trained with the training data pair

$$\big[\Delta\theta(k), \Delta\dot{R}(k)\cos\alpha(k), \Delta\dot{R}(k)\sin\alpha(k), \tilde{z}_x(k), \tilde{z}_y(k); u_x(k), u_y(k)\big].$$

The objective of learning is to minimize the error function

$$E = \frac{1}{2}\Big[(\hat{u}_x(k) - u_x(k))^2 + (\hat{u}_y(k) - u_y(k))^2\Big] \quad (13)$$

where $u_x(k)$ and $u_y(k)$ are the desired network outputs and $\hat{u}_x(k)$ and $\hat{u}_y(k)$ are the actual outputs in the x and y directions, respectively.

III. Estimation of Maneuver Using a Neural Fuzzy Network

In this section, we shall introduce a neural fuzzy network called SONFIN and then propose a high-accuracy maneuver estimation scheme based on this network integrated into a Kalman filter. Besides, adaptive system covariance of Kalman filter is derived. The proposed estimation mechanism is, thus, called KF-SONFIN.

A. KF-SONFIN for Maneuver Estimation

As we know, there are no exact relations between the radar output signal and the target maneuver (including onset time and magnitude), but there exists a complex nonlinear mapping between them. The plane pulsed Doppler radar can provide not only the position measurement sequence of the target in two dimensions but also its range rate sequence. To map the input vector to the target acceleration vector accurately, it is important to find the effective input elements, which are acquired from radar output and tracking filters. Fig. 2 shows the flow chart of the maneuver estimation using the proposed KF-SONFIN based on a feature vector composed of five elements that are acquired from plane pulsed Doppler radar and Kalman filter.

The purpose of feature extraction is to extract the feature data from different existing data available and accessible to generate the useful inputs for the estimation network. As we know, except the radar sensor, there are other sensors integrated in a modern surveillance system, such as infrared, laser, TV, etc. We can acquire more data with different characteristics from these sensors to generate other features that are helpful for the neural fuzzy network to find the man-made target maneuver. In our system, we use three feature extraction processes to produce five features for our estimation system as shown in Fig. 2. For

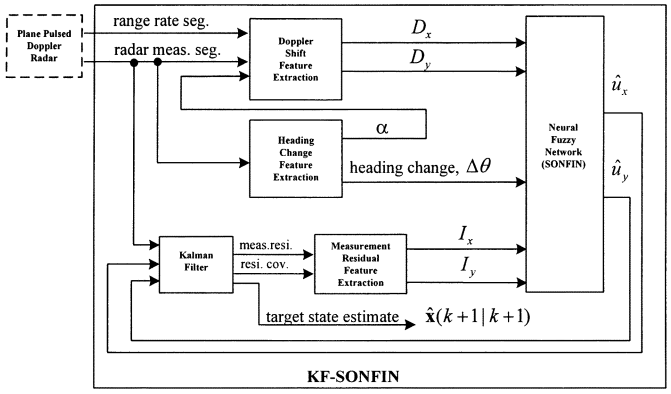


Fig. 2. Flowchart of KF-SONFIN estimation.

the purpose of separating the effects in the x and y directions, we calculate the Doppler-shift-difference feature and measurement-residual feature in the x and y directions, respectively, in addition to the heading-change feature.

In Fig. 2, the neural fuzzy network (SONFIN) implements data fusion from five input feature elements and estimates the unexpected maneuver motion of the target including the maneuver onset time and magnitude. Upon the target maneuver occurring, the SONFIN will produce the estimated accelerations as a response. The estimated acceleration will not only be fed into the Kalman filter to compensate the error caused by the target maneuver, but also be used to adjust the system covariance of Kalman filter and forms a closed loop mechanism for the maneuver target tracking system.

In the following Sections III-B– III-D, we shall introduce the feature extraction processes, SONFIN and the adaptive system covariance of Kalman filter in details, which constitute the proposed KF-SONFIN.

B. Feature Extraction

We construct three feature extraction processes to extract useful feature data for our maneuver estimation network, SONFIN. They are measurement residual feature extraction, heading change feature extraction and Doppler shift feature extraction [15]. We now explain the relations between these features and the maneuver of the target as follows.

1) Measurement Residual Feature Extraction: Measurement residual (innovation) is one of the most useful features to detect target maneuver. Most researchers in maneuvering target tracking used this concept to detect maneuvers, [11], [12], [24]–[29].

For the Kalman filter, the innovation sequence is represented as $\{\tilde{\mathbf{z}}(1), \tilde{\mathbf{z}}(2), \ldots, \tilde{\mathbf{z}}(k+1), \ldots\}$, where $\tilde{\mathbf{z}}(k+1) = \mathbf{z}(k+1) - H\hat{\mathbf{x}}(k+1|k)$, $k = 0, 1, 2, \ldots$. When a target is in nonmaneuver state, the mean of the innovation $\tilde{\mathbf{z}}(k)$ is zero. However, when the target begins to maneuver the mean of the innovation is no longer zero. This is a fact that measurement residual can be utilized to detect maneuvers [29].

In many previous researches, the concept of measurement residual was utilized to find the maneuver, based on the fact that a maneuver manifests itself as a large innovation. Chan's IE algorithm [10] used the innovations as a linear measurement of the unknown input that models the target maneuvers. Bar-Shalom's VDF algorithm [11] employed a *fading memory average* based on the *normalized innovation squared* from the estimator in the quiescent model and according to this fading memory average the *effective window length* is obtained over which the presence of a maneuver is tested. A maneuver is declared when a fading memory average of the normalized innovations exceeds a threshold and the onset of the maneuver is then taken as the beginning of the sliding effective window. R. K. Mehra [24] used the so-called covariance-match techniques to implement the adaptive filter. The basic idea of the covariance of innovation sequence must be consistent with their theoretical covariance. If the actual covariance of $\tilde{z}(k+1)$ is much larger than $(HP(k+1|k)H^T + R)$ obtained from the Kalman filter, then the adaptive actions are taken.

Chang and Tabaczynski [27] regarded the maneuver detection problem as the problem of discriminating two hypotheses, the maneuvering target and nonmaneuvering target hypotheses, based upon filter residuals. A generalized likelihood ratio test is defined to compare with a threshold value and therefore the maneuver is declared when the generalized likelihood ratio exceeds this threshold.

From the above practical research examples, it is evident that the measurement residual has a strong relationship to the target maneuver. In this paper, the measurement residual feature is extracted from normalizing the measurement residual with respect to its covariance in the x and y directions, respectively

$$I_x(k+1) = \frac{\tilde{z}_x(k+1)}{S_{xx}(k+1)} \tag{14}$$

$$I_y(k+1) = \frac{\tilde{z}_y(k+1)}{S_{yy}(k+1)} \tag{15}$$

where $\tilde{z}_x(k+1)$ and $\tilde{z}_y(k+1)$ are the measurement residuals in the x and y directions, respectively and $S_{xx}(k+1)$ and $S_{yy}(k+1)$ are the diagonal elements of the covariance matrix.

2) Heading Change Feature Extraction: We use the change in heading to estimate the target maneuver, which depends on the fact that the first derivative of the target heading is angular velocity (turn-rate), which represents a turning motion of target. When a target performs a maneuver, it can be modeled as cross-track $a_n(t)$ and along-track $a_t(t)$ accelerations, respectively. The target model, as described in [1], [30], uses the track-oriented coordinates to explain the relation of accelerations with the Cartesian coordinates

$$\frac{d\theta}{dt} = \frac{a_n(t)}{V(t)} \tag{16}$$

$$\frac{dV}{dt} = a_t(t) \tag{17}$$

$$\frac{dx}{dt} = V(t)\sin\theta(t) \tag{18}$$

$$\frac{dy}{dt} = V(t)\cos\theta(t) \tag{19}$$

$$a_x(t) = a_t(t)\sin\theta(t) + a_n(t)\cos\theta(t) \tag{20}$$

$$a_y(t) = a_t(t)\cos\theta(t) - a_n(t)\sin\theta(t) \tag{21}$$

where $a_x(t)$ and $a_y(t)$ are the Cartesian components of acceleration, $V(t)$ is the target velocity, $\theta(t)$ is the heading of the target path and $d\theta/dt \triangleq \omega(t)$ is the turn-rate of the target.

If $a_n(t) \neq 0, a_t(t) = 0$ the target is in circular motion and we get

$$a_x(t) = a_n(t)\cos\theta(t) \tag{22}$$

$$a_y(t) = -a_n(t)\sin\theta(t). \tag{23}$$

In this case, it is obvious that the magnitude of the target acceleration $||\mathbf{a}|| = \sqrt{a_x^2 + a_y^2} = \omega V$, is the function of the target turn-rate.

J. P. Helferty [31] extended the work of Singer [32] by modeling the maneuver process as a linear system with the constant forward velocity assumption. He used the turn-rate distribution for acceleration modeling to improve maneuvering target tracking. Although this model gave an increase in the number of states for the maneuver model and hence increased the computational load of the Kalman filter, it has illustrated that the turn-rate cannot be neglected when processing maneuver behavior of a target. Other researches, such as [33]–[35], used the turn-rate to deal with the target maneuver tracking problem.

There are many methods to estimate the heading of the target path from noisy position measurements, which can be found in [36], [37]. For simplicity and taking use of the robustness of neural fuzzy networks, we take $\Delta\theta(k) = \theta(k) - \theta(k-1)$ as the heading change of the target path and use the following equation to estimate the heading of the target:

$$\theta(k) = \tan^{-1}\frac{y(k) - \bar{y}}{x(k) - \bar{x}} \tag{24}$$

where $\bar{y} = (1/N)\sum_{i=1}^{N} y(i)$, $\bar{x} = (1/N)\sum_{i=1}^{N} x(i)$ and N is the number of measurements used to estimate the heading of a target. In this paper, we set $N = 6$.

3) Doppler Shift Feature Extraction: The radial velocity (range rate) measurement of the pulsed Doppler radar has been applied in target tracking for a long time, [5], [38]–[40]. For example, A. Farina and S. Pardini [5] developed a TWS algorithm with and without radial velocity measurement. From the simulations, it has proved that the radial velocity makes an improvement of track mean life for the strong acceleration target.

The feature of Doppler shift is extracted from the Doppler radar, which provides a frequency shift to measure the radial velocity of a moving target. The Doppler shift frequency is given by [40]

$$f_d(k) = \frac{2}{\lambda}\dot{R}(k) = \frac{2}{\lambda}||V(k)||\cos\phi(k) \tag{25}$$

where $\dot{R}(k)$ is the radial velocity of the target relative to radar, λ is the wavelength of the radar transmitter, $||V(k)||$ is the magnitude of the target velocity and $\phi(k)$ is the angle between the target velocity and line of sight to the sensor.

Simplifying the target trajectory model to allow movement in a straight line which includes acceleration, $\mathbf{a}$

$$V(k) = V_0 + \mathbf{a}kT, \quad k = 0, 1, 2\ldots \tag{26}$$

then the change of the Doppler shift is given by

$$\nabla f_d(k) \triangleq f_d(k) - f_d(k-1) = \frac{2}{\lambda}||\mathbf{a}||T\cos\phi(k). \tag{27}$$

That is, the change of the Doppler shift is proportional to acceleration of the target.

In this paper, we use the change in Doppler shift normalized by its variance as a feature, which is related to the variance of the range rate, i.e.,

$$\mathbf{D}(k) = \frac{1}{\sigma_{f_d}}\nabla f_d(k) = [D_x(k), D_y(k)] \tag{28}$$

where $D_x(k) = \mathbf{D}(k)\cos\alpha(k)$, $D_y(k) = \mathbf{D}(k)\sin\alpha(k)$ and α is the angle between the target position and radar site.

C. SONFIN

The neural fuzzy network used in our maneuver estimation system is called self-constructing neural fuzzy inference network (SONFIN) that we proposed previously in [20]. The SONFIN is a general connectionist model of a fuzzy inference system, which can find its optimal structure and parameters automatically. There are no rules initially in the SONFIN. They are created and adapted as on-line learning proceeds via simultaneous structure and parameter learning, so the SONFIN can be used for normal operation at any time as learning proceeds without any assignment of fuzzy rules in advance. A novel network construction method for solving the dilemma between the number of rules and the number of consequent terms is developed. The number of generated rules and membership functions is small even for modeling a sophisticated system. The SONFIN always produces an economical network size and the learning speed and modeling ability are superior to ordinary neural networks.

A key feature of the SONFIN structure is that a high-dimensional fuzzy system is implemented with a small number of rules and fuzzy terms. This is achieved first by partitioning the input and output spaces into clusters efficiently through learning proper fuzzy terms for each input/output variable and then by constructing fuzzy rules optimally through finding proper mapping between input and output clusters in the SONFIN. In addition, due to the physical meaning of the fuzzy IF–THEN rule, each input node in the SONFIN is only connected to its related rule nodes through its term nodes instead of being connected to all the rule nodes in Layer 3 of the SONFIN. This results in a small number of weights to be tuned in the SONFIN.

The structure of the SONFIN is shown in Fig. 3. This six-layered network realizes a fuzzy model of the following form.

Rule i: IF x_1 is A_1^i and … and x_n is A_n^i THEN y is $m_0^i + a_j^i x_j + \cdots$, where A_j^i is the fuzzy set of the ith linguistic term of input variable x_i, m_0^i is the center of a symmetric membership function on y and a_j^i is the consequent parameter. It is noted that unlike the traditional TSK model where all the input variables are used in the output linear equation, only the significant ones are used in the SONFIN; i.e., some a_j^i's in the above fuzzy rules are zero.

The SONFIN consists of nodes, each of which has some finite fan-in of connections represented by weight values from other nodes and fan-out of connections to other nodes. Associated with the fan-in of a node is an integration function f which serves to combine information, activation, or evidence from other nodes. This function provides the net input for this node

$$net_{input} = f\left(u_1^{(k)}, u_2^{(k)}, \ldots, u_p^{(k)}; w_1^{(k)}, w_2^{(k)}, \ldots, w_p^{(k)}\right) \tag{29}$$

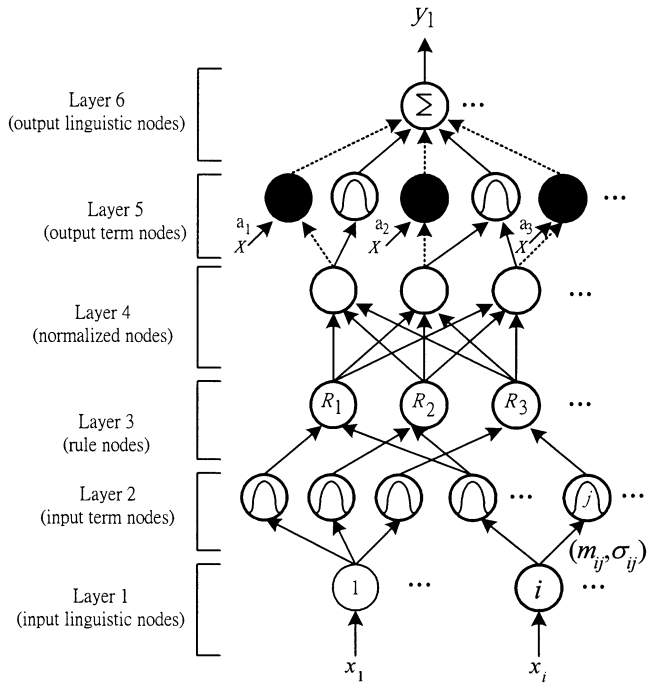


Fig. 3. Structure of the SONFIN.

where $u_1^{(k)}, u_2^{(k)}, \ldots, u_p^{(k)}$ are inputs to this node and $w_1^{(k)}, w_2^{(k)}, \ldots, w_p^{(k)}$ are the associated link weights. The superscript (k) in the above equation indicates the layer number. This notation will also be used in the following equations. A second action of each node is to output an activation value as a function of its net_{input}

$$output = o_i^{(k)} = a(net_{\text{input}}) = a(f) \tag{30}$$

where $o_i^{(k)}$ denotes the output of the ith node in the layer k and $a(\cdot)$ denotes the activation function. We shall describe the functions of the nodes in each of the six layers of the SONFIN as follows.

Layer 1: No computation is done in this layer. Each node in this layer, which corresponds to one input variable, only transmits input values to the next layer directly. That is

$$f = u_i^{(1)} \quad \text{and} \quad a^{(1)} = f. \tag{31}$$

From the above equation, the link weight in layer one ($w_i^{(1)}$) is unity.

Layer 2: Each node in this layer corresponds to one linguistic label (small, large, etc.) of one of the input variables in Layer 1. In other words, the membership value, which specifies the degree to which an input value belongs to a fuzzy set is calculated in Layer 2. With the choice of Gaussian membership function, the operation performed in this layer is

$$f_j\left(u_i^{(2)}\right) = -\frac{\left(u_i^{(2)} - m_{ij}\right)^2}{\sigma_{ij}^2} \text{ and}$$
$$a^{(2)}(f) = e^f \tag{32}$$

where m_{ij} and σ_{ij} are, respectively, the center (or mean) and the width (or variance) of the Gaussian membership function of the jth partition for the ith input variable u_i. Hence, the link weight in this layer can be interpreted as m_{ij}.

Layer 3: A node in this layer represents one fuzzy logic rule and performs precondition matching of a rule. Here we use the following AND operation for each Layer-3 node

$$f\left(u_i^{(3)}\right) = \prod_{i=1}^{q} u_i^{(3)} = e^{-[D_i(x-m_i)]^T[D_i(x-m_i)]}$$
$$\text{and } a^{(3)}(f) = f \tag{33}$$

where q is the number of Layer-2 nodes participating in the IF part of the rule, $D_i = diag(1/\sigma_{i1}, 1/\sigma_{i2}, \ldots, 1/\sigma_{in})$, $m_i = (m_{i1}, m_{i2}, \ldots, m_{in})^T$. The weights of the links in Layer 3 ($w_i^{(3)}$) have the value of one. The output of a Layer-3 node represents the firing strength of the corresponding fuzzy rule.

Layer 4: The number of nodes in this layer is equal to that in Layer 3 and the firing strength calculated in Layer 3 is normalized in this layer by

$$f\left(u_i^{(4)}\right) = \sum\nolimits_{i=1}^{r} u_i^{(4)} \text{ and } a^{(4)}(f) = \frac{u_i^{(4)}}{f} \tag{34}$$

where r is the number of rule nodes in Layer 3. Like Layer 3, the link weight ($w_i^{(4)}$) in this layer is unity too.

Layer 5: This layer is called the consequent layer. Two types of nodes are used in this layer and they are denoted as blank and shaded circles in Fig. 3, respectively. The node denoted by a blank circle (blank node) is the essential node representing a fuzzy set (described by a Gaussian membership function) of the output variable. Only the center of each Gaussian membership function is delivered to the next layer for the local mean of maximum LMOM defuzzification operation and the width is used for output clustering only. Different nodes in Layer 4 may be connected to a same blank node in Layer 5, meaning that the same consequent fuzzy set is specified for different rules. The function of the blank node is

$$f = \sum\nolimits_{i=1}^{s} u_i^{(5)} \text{ and } a^{(5)}(f) = f \cdot a_0^i \tag{35}$$

where s is the number of nodes in Layer 4 and $a_0^i = m_0^i$ is the center of a Gaussian membership function. As to the shaded node, it is generated only when necessary. Each node in Layer 4 has its own corresponding shaded node in Layer 5. One of the inputs to a shaded node is the output delivered from Layer 4 and the other possible inputs (terms) are the input variables from Layer 1. The shaded node function is

$$f = \sum\nolimits_{j=1}^{n} a_j^i x_j \text{ and } a^{(5)}(f) = f \cdot u_i^{(5)} \tag{36}$$

where the summation is over the significant terms connected to the shaded node only and a_j^i is the corresponding parameter. Combining these two types of nodes in Layer 5, we obtain the whole function performed by this layer as

$$a^{(5)}(f) = \left(\sum_{j=1}^{n} a_j^i x_j + a_0^i\right) u_i^{(5)}. \tag{37}$$

Layer 6: Each node in this layer corresponds to one output variable. The node integrates all the actions recommended by Layer 5 and acts as a defuzzifier with

$$f\left(u_i^{(6)}\right) = \sum_{i=1}^{t} u_i^{(6)} \text{and } a^{(6)}(f) = f \tag{38}$$

where t is the number of nodes in Layer 5.

Two types of learning, structure and parameter learning, are used concurrently for constructing the SONFIN. The structure learning includes both the precondition and consequent structure identification of a fuzzy IF-THEN rule. Here the precondition structure identification corresponds to the input-space partitioning and can be formulated as a combinational optimization problem with the following two objectives: to minimize the number of rules generated and to minimize the number of fuzzy sets on the universe of discourse of each input variable. As to the consequent structure identification, the main task is to decide when to generate a new membership function for an output variable and which significant terms (input variables) should be added to the consequent part (a linear equation) when necessary. For the parameter learning, based upon supervised learning algorithms, the parameters of the linear equations in the consequent parts are adjusted by either least mean squares (LMS) or recursive least squares (RLS) algorithms and the parameters in the precondition part are adjusted by the backpropagation algorithm to minimize a given cost function. The SONFIN can be used for normal operation at any time during the learning process without repeated training on the input/output patterns when on-line operation is required. There are no rules (i.e., no nodes in the network except the input/output nodes) in the SONFIN initially. They are created dynamically as learning proceeds upon receiving on-line incoming training data by performing the following learning processes simultaneously: A) input/output space partition; B) construction of fuzzy rules; C) optimal consequent structure identification; D) parameter identification. In the above, processes A, B, and C belong to the structure learning phase and process D belongs to the parameter learning phase.

In the structure identification of the precondition part of the SONFIN, the input space is partitioned in a flexible way according to an aligned clustering-based algorithm. As to the structure identification of the consequent part, only a singleton value selected by a clustering method is assigned to each rule initially. Afterwards, some additional significant terms (input variables) selected via projected-based correlation measure for each rule will be added to the consequent part (forming a linear equation of input variables). The combined precondition and consequent structure identification scheme can set up an economical and dynamically growing network automatically. This means the SONFIN can grow its rule nodes, term nodes and link weights upon necessary on the fly and, thus, own the so-called self-construction capability. The details of the learning processes for SONFIN can be found in Section IV.

D. Adaptive System Covariance of Kalman Filter

Consider the situation when a mismatch exists between the true acceleration and the estimated acceleration used in the tracking filter and this error causes the inferior tracking accuracy. In order to further compensate the tracking filter and reduce the tracking error arising from the maneuver estimation error we would like to add an additional error covariance term to help compensate for this uncertainty [41].

Referring back to the Kalman filter described in Section II, the state equation with a deterministic input $\mathbf{u}(k)$ and random disturbance $\mathbf{v}(k)$ can be given by

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + G_1\mathbf{u}(k) + G_2\mathbf{v}(k) \tag{39}$$

then the predicted covariance matrix $P(k+1|k)$ is

$$P(k+1|k) = FP(k|k)F^T + G_2Q(k)G_2^T. \tag{40}$$

In practice, we cannot find the deterministic input $\mathbf{u}(k)$ exactly; there exists a difference between the true input and the estimated deterministic input, $\hat{\mathbf{u}}(k) - \mathbf{u}(k)$. The state prediction and update state estimate of the Kalman filter are

$$\hat{\mathbf{x}}(k+1|k) = F\hat{\mathbf{x}}(k|k) + G_1\hat{\mathbf{u}}(k) \tag{41}$$

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k+1) =& \hat{\mathbf{x}}(k+1|k) + W(k+1)\\ &\cdot [\mathbf{z}(k+1) - H\hat{\mathbf{x}}(k+1|k)]\\ =& \hat{\mathbf{x}}(k+1|k) + W(k+1)\\ &\cdot [H\mathbf{x}(k+1) + \mathbf{v}(k+1) - H\hat{\mathbf{x}}(k+1|k)]\\ =& [I - W(k+1)H]\hat{\mathbf{x}}(k+1|k)\\ &+ W(k+1)H\mathbf{x}(k+1)\\ &+ W(k+1)\mathbf{v}(k+1)\\ =& [I - W(k+1)H]\,[F\hat{\mathbf{x}}(k|k) + G_1\hat{\mathbf{u}}(k)]\\ &+ W(k+1)H[F\mathbf{x}(k) + G_1\mathbf{u}(k) + G_2\mathbf{v}(k)]\\ &+ W(k+1)\mathbf{v}(k+1).\end{aligned} \tag{42}$$

The update covariance matrix is

$$\begin{aligned}P(k+1|k+1) =& E\Big\{[\hat{\mathbf{x}}(k+1|k+1) - \mathbf{x}(k+1)]\\ &\cdot [\hat{\mathbf{x}}(k+1|k+1) - \mathbf{x}(k+1)]^T\Big\}\\ =& [I - W(k+1)H]\\ &\cdot \big[FP(k|k)F^T + G_2QG_2 + D\big]\\ &\cdot [I - W(k+1)H]^T\\ &+ W(k+1)R(k+1)W^T(k+1)\end{aligned} \tag{43}$$

where

$$\begin{aligned}D =& G_1E\left\{[\hat{\mathbf{u}}(k) - \mathbf{u}(k)]\,[\hat{\mathbf{u}}(k) - \mathbf{u}(k)]^T\right\}G_1^T\\ &+ G_1E\left\{[\hat{\mathbf{u}}(k) - \mathbf{u}(k)][\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)]^T\right\}F^T\\ &+ FE\left\{[\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)]\,[\hat{\mathbf{u}}(k) - \mathbf{u}(k)]^T\right\}G_1^T.\end{aligned} \tag{44}$$

The state prediction variance is

$$\begin{aligned}P(k+1|k) =& E\{\tilde{\mathbf{x}}(k+1|k)\tilde{\mathbf{x}}^T(k+1|k)\}\\ =& E\Big\{[\hat{\mathbf{x}}(k+1|k) - \mathbf{x}(k+1)]\\ &\cdot [\hat{\mathbf{x}}(k+1|k) - \mathbf{x}(k+1)]^T\Big\}\\ =& FP(k|k)F^T + G_2QG_2^T + D.\end{aligned} \tag{45}$$

If it is assumed that the estimate of the target states can be so close to the target states, $E\{\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)\} \cong 0$, then

$$\begin{aligned} D =& G_1 E\left\{[\hat{\mathbf{u}}(k) - \mathbf{u}(k)]\,[\hat{\mathbf{u}}(k) - \mathbf{u}(k)]^T\right\} G_1^T \\ =& G_1 Q_u G_1^T \end{aligned} \tag{46}$$

where $Q_u = E\{[\hat{\mathbf{u}}(k) - \mathbf{u}(k)][\hat{\mathbf{u}}(k) - \mathbf{u}(k)]^T$ and $P(k+1|k) = FP(k|k)F^T + G_2 Q G_2^T + G_1 Q_u G_1^T$. For $G_1 = G_2 = G$, we have

$$P(k+1|k) = FP(k|k)F^T + G(Q + Q_u)G^T. \tag{47}$$

Assuming that the KF-SONFIN can estimate the acceleration input $\hat{\mathbf{u}}(k)$ no less than half of the true value in each axis and here we take $\hat{\mathbf{u}}(\mathbf{k}) = 1/2\mathbf{u}(k)$, then $\hat{\mathbf{u}}(k) - \mathbf{u}(k) = -\hat{\mathbf{u}}(k)$

$$\begin{aligned} Q_u =& E\left\{[\hat{\mathbf{u}}(k) - \mathbf{u}(k)]\,[\hat{\mathbf{u}}(k) - \mathbf{u}(k)]^T\right\} \\ =& \frac{1}{N}\sum_{k=1}^{N}\left\{[-\hat{\mathbf{u}}(k)]\,[-\hat{\mathbf{u}}(k)]^T\right\} \\ =& \frac{1}{N}\sum_{k=1}^{N}\hat{\mathbf{u}}(k)\mathbf{u}(k)^T \end{aligned} \tag{48}$$

where (N) is the interval from the time the target starts to maneuver to the time it stops maneuvering [42]. Here, we call N the length of acceleration.

IV. Simulation Results

The estimation improvement obtained by the KF-SONFIN presented in this paper is illustrated by the following examples. In the experiments reported in this section, the following assumptions and parameter values are used. The sampling time interval is assumed to be 10 s, which is the time of the radar antenna scanning a revolution. The radar measurement sequence is assumed to have been transformed from polar coordinates to Cartesian coordinates before the TWS tracking process. The variances directions, R_{11} and R_{22}, are set to be 10^4 m^2 and the covariance R_{12} set to be 500 m^2. For extracting the Doppler shift in the KF-SONFIN, the wavelength of the radar transmitter λ is known as 0.008 57 m (corresponding to k_a -band radar) and the standard variance of Doppler shift σ_{f_d}, is assumed to be 30 m/s.

The SONFIN is trained off-line with the three features mentioned in Section III. The training data are generated as follows. We divide the range of the heading of the target path, 0° to 90°, equally into 90 intervals, with each interval being 1.0°. In these 90 trajectories, each trajectory has 30 position measurements, with random accelerations being generated from 0 to 20 m/s^2. We select the points with the maximum or minimum acceleration as the training data in each trajectory. The corresponding input feature vectors (with five-elements) and the desired output vectors (with two-elements) are acquired. These two vectors construct a training pattern pair in the form of (feature inputs, desired outputs). Hence, as a total, we get 180 training patterns. With the same procedure, we can obtain 172 (feature inputs, desired outputs) pairs as testing patterns by dividing the range of heading values, 1.5° to 89.5°, into 1° intervals.

The fuzzy rules resulting from the trained SONFIN are listed as follows.

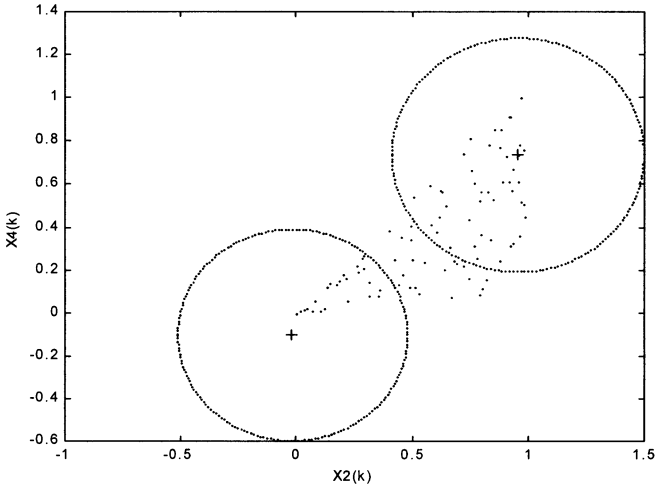


Fig. 4. Input training patterns and the final assignment of rules for x_2 and x_4.

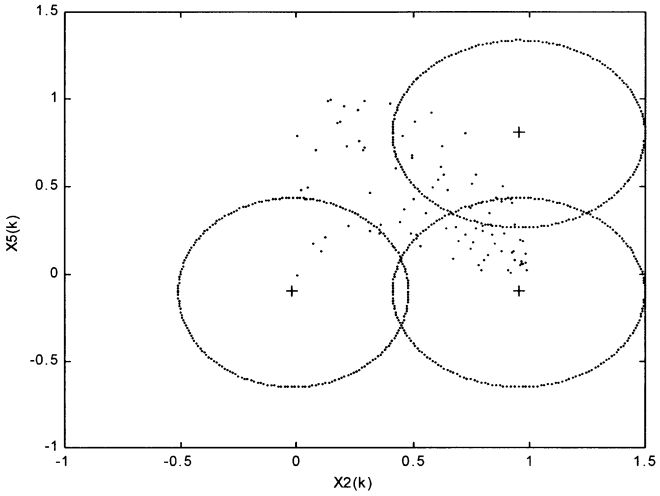


Fig. 5. Input training patterns and the final assignment of rules for x_2 and x_5.

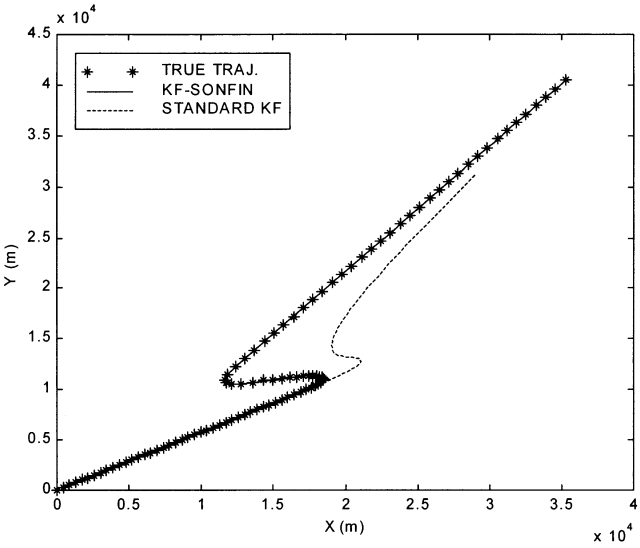
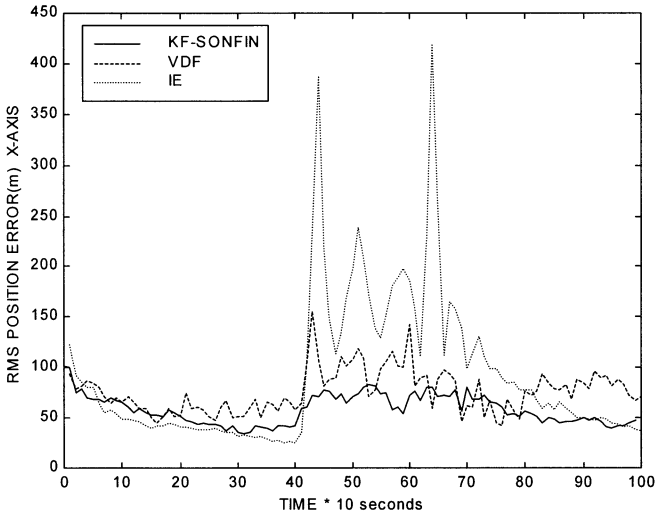
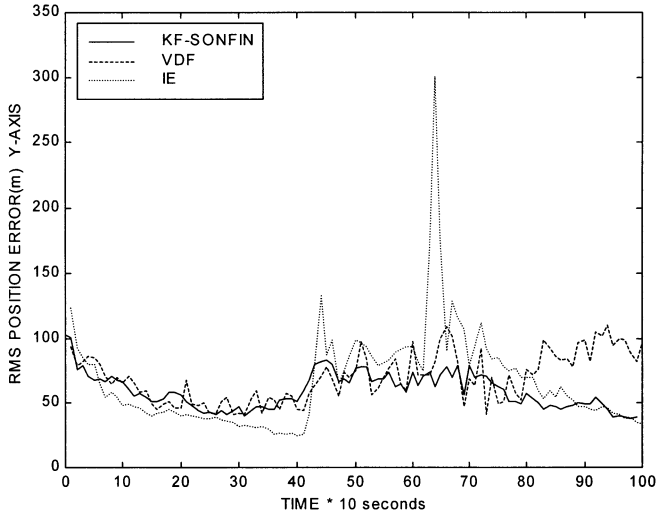


Fig. 6. Target's trajectory versus the tracking results of KF-SONFIN and standard Kalman filter with the mean values over 50 runs in Experiment 1. (a) x direction. (b) y direction.



(a)



(b)

Fig. 7. Comparisons of RMS position tracking error in Experiment 1. (a) x direction. (b) y direction.

Rule 1: IF x_1 is $\mu(0.1111, 0.9881)$ and x_2 is $\mu(0.9497, 1.0859)$ and x_3 is $\mu(0.6907, 0.2500)$ and x_4 is
$\mu(0.7382, 1.0859)$ and x_5 is $\mu(0.8075, 1.0702)$
THEN $y_1 = 0.8329 + 0.0105x_1 + 0.0512x_2 + 0.0107x_3 + 0.0377x_4 + 0.9139x_5$
$y_2 = -0.0245 + 0.0102x_1 + 0.0289x_2 + 0.0103x_3 + 0.2023x_4 + 0.2016x_5$.

Rule 2: IF x_1 is $\mu(0.1111, 0.9881)$ and x_2 is $\mu(-0.0200, 0.9881)$ and x_3 is $\mu(-0.0300, 0.9881)$ and x_4 is
$\mu(-0.1000, 0.9881)$ and x_5 is $\mu(-0.1000, 1.0859)$
THEN $y_1 = -0.0185 - 0.0326x_1 + 0.9749x_2 - 0.0126x_3 - 0.1062x_4 + 0.1062x_5$
$y_2 = 0.0203 + 0.0642x_1 + 0.0327x_2 + 1.0251x_3 + 0.1058x_4 + 0.1059x_5$.

Rule 3: IF x_1 is $\mu(0.1111, 0.9881)$ and x_2 is $\mu(0.9497, 1.0859)$ and x_3 is $\mu(-0.0300, 0.9881)$ and x_4 is
$\mu(0.7382, 1.0859)$ and x_5 is $\mu(-0.1000, 1.0859)$
THEN $y_1 = 0.0989 + 0.0747x_1 + 0.9545x_2 - 0.0335x_3 - 0.4021x_4 + 0.1013x_5$
$y_2 = -0.1541 - 0.1285x_1 + 0.0704x_2 + 1.0499x_3 + 0.2036x_4 - 0.2030x_5$.

Rule 4: IF x_1 is $\mu(0.1111, 0.9881)$ and x_2 is $\mu(0.9497, 1.0859)$ and x_3 is $\mu(0.6925, 1.0702)$ and x_4 is
$\mu(0.7382, 1.0859)$ and x_5 is $\mu(0.8075, 1.0702)$
THEN $y_1 = -0.1001 + 0.0951x_1 + 1.0450x_2 - 0.0050x_3 + 0.2012x_4 + 0.2006x_5$
$y_2 = 0.2192 - 0.2488x_1 - 0.0876x_2 + 1.0273x_3 - 0.2037x_4 + 0.2002x_5$.

Rule 5: IF x_1 is $\mu(0.8075, 1.0702)$ and x_2 is $\mu(-0.0200, 0.9881)$ and x_3 is $\mu(0.6925, 1.0702)$ and x_4 is
$\mu(-0.1000, 0.9881)$ and x_5 is $\mu(-0.1000, 1.0859)$
THEN $y_1 = 0.0010 - 0.0006x_1 + 1.0045x_2 + 0.0037x_3 - 0.5000x_4 - 0.5002x_5$
$y_2 = 0.0021 - 0.0013x_1 + 0.0018x_2 + 0.9959x_3 + 0.2005x_4 + 0.2001x_5$.

In the above rules, $\mu(m_i, \sigma_i)$ represents a Gaussian membership function with center m_i and width σ_i, x_1 is the heading change, x_2 is the change of the Doppler shift in x direction, x_3 is the change of the Doppler shift in y direction, x_4 is the innovation in x direction, x_5 is the innovation in y direction, y_1 is the estimated acceleration in x direction and y_2 is the estimated acceleration in y direction.

The input training patterns and the final assignment of rules for x_2 and x_4 and x_2 and x_5 are shown in Figs. 4 and 5, respectively. In Fig. 4, the lower ellipse is corresponding to Rules 2 and 5 and the upper ellipse is corresponding to Rules 1, 3, and 4. The boundary of each ellipse represents the product of the membership degrees of x_2 and x_4 with value $(e^{-0.25})$. The major and minor axes of the ellipses represent half the width of Gaussian membership function corresponding to x_2 and x_4. In Fig. 5, the lower left ellipse is corresponding to Rules 2 and 5, the right ellipse is corresponding to Rule 3 and the upper ellipse is corresponding to Rules 1 and 4. The boundary of each ellipse represents the product of the membership degrees of x_2 and x_5 with value $e^{-0.25}$. The major and minor axes of the ellipses represent half the width of Gaussian membership function corresponding to x_2 and x_5.

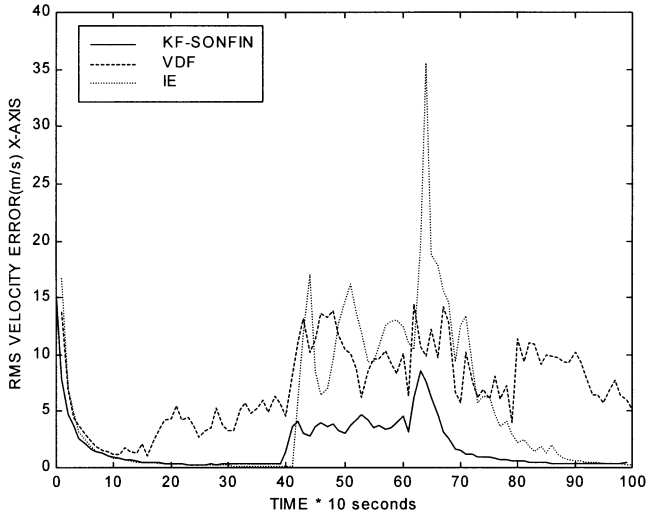
The initial estimates of the states in the x and y directions are obtained from [11] for our proposed method as well as the compared algorithms

$$\hat{x}(0|0) = \hat{x}_1(0|0) = x(0) + N(0, R_{11}) \tag{49}$$

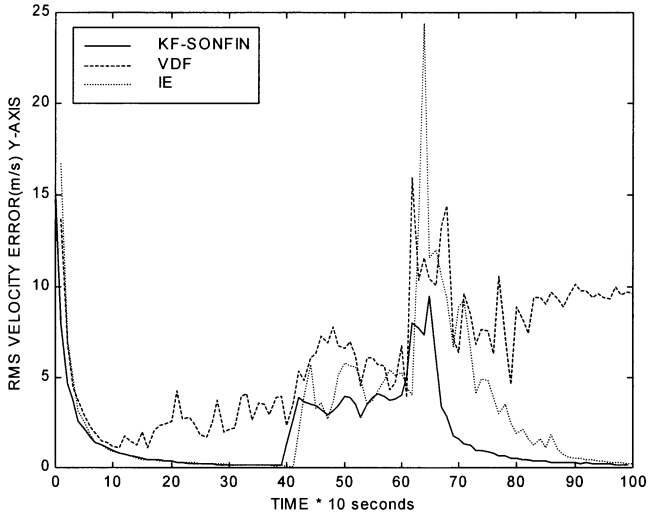
$$\hat{y}(0|0) = \hat{y}_1(0|0) = y(0) + N(0, R_{22}) \tag{50}$$

$$\hat{\dot{x}}(0|0) = \hat{x}_2(0|0) = \frac{\{x_1(0|0) - [x(0) - \dot{x}(0)T + N(0, R_{11})]\}}{T} \tag{51}$$

$$\hat{\dot{y}}(0|0) = \hat{y}_2(0|0) = \frac{\{y_1(0|0) - [y(0) - \dot{y}(0)T + N(0, R_{22})]\}}{T} \tag{52}$$

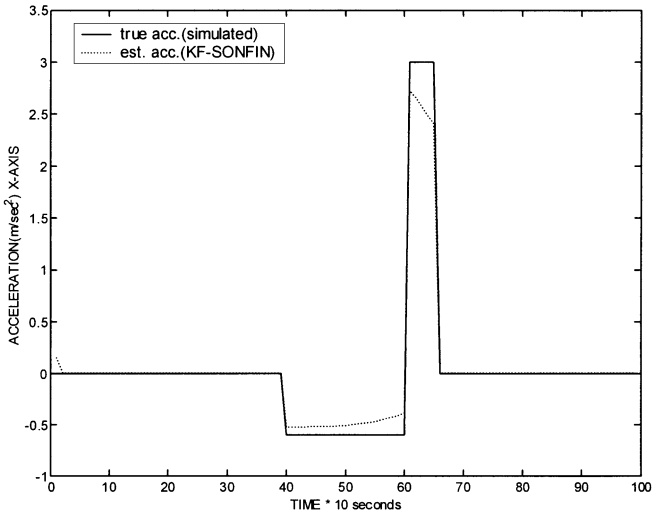


(a)

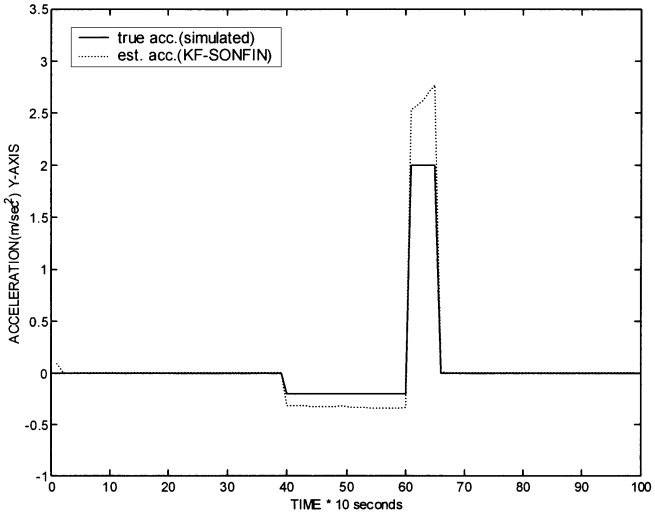


(b)

Fig. 8. Comparisons of RMS velocity tracking error in Experiment 1. (a) x direction. (b) y direction.



(a)



(b)

Fig. 9. Mean value of acceleration estimated by SONFIN versus true acceleration value in Experiment 1. (a) x direction. (b) y direction.

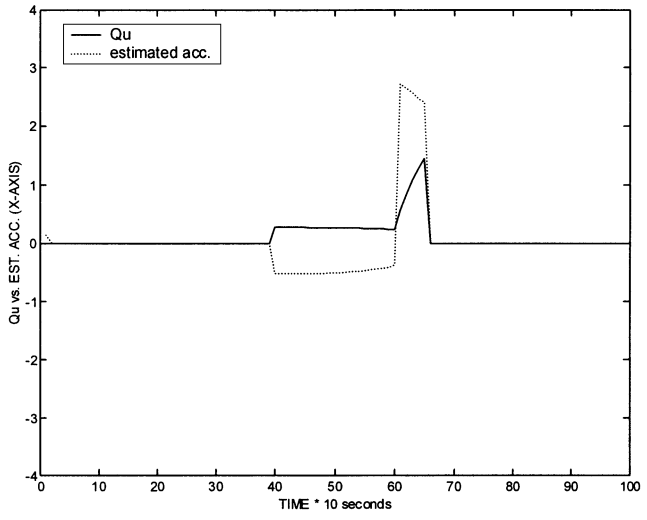
where $N(\cdot)$ is the Gaussian noise. The initial covariance matrix is set as

$$P(0|0) = \begin{bmatrix} R_{11} & \frac{R_{11}}{T} & 0 & 0 \\ R_{11} & \frac{2R_{11}}{T^2} & 0 & 0 \\ 0 & 0 & R_{22} & \frac{R_{22}}{T} \\ 0 & 0 & \frac{R_{22}}{T} & \frac{2R_{22}}{T^2} \end{bmatrix}. \tag{53}$$

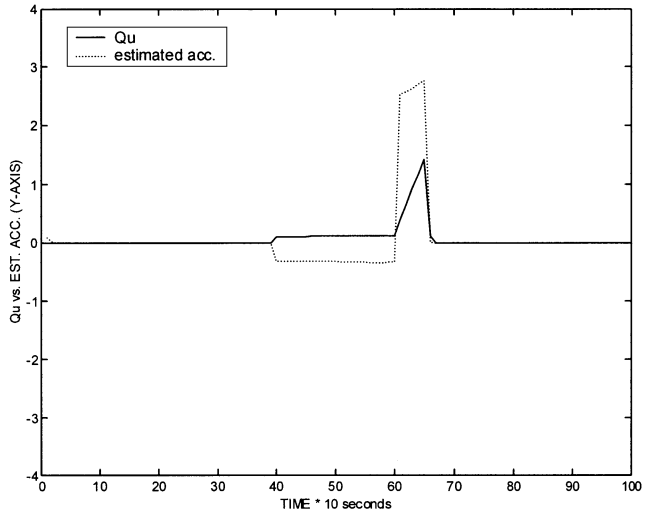
Experiment 1: As an evaluation of the new tracking scheme and for the purpose of comparison with two existing algorithms, VDF and IE, we modify the target scenario considered in [10], [11] for simulation testing here. The initial position of the target scenario is given by $(x, y) = (0.1\text{ m}, 0.1\text{ m})$ with an initial speed of $(v_x, v_y) = (43.3\text{ m/s}, 25.0\text{ m/s})$ and on a constant course of 30° and speed until $t = 400$ s, then it starts to accelerate with values $u_x = -0.6\text{ m/s}^2$, $u_y = -0.2\text{ m/s}^2$. This maneuver completes at $t = 600$ s and from then on it starts the other maneuver at $t = 610$ s with accelerations of $u_x = 3\text{ m/s}^2, u_y = 2\text{ m/s}^2$. This maneuver stops at $t = 660$ s. For the purpose of comparison on the convergence performance after the above two maneuvers, the target scenario lasts for 1000 s (100 samples). The trajectory of the target with the corresponding accelerations, which make the trajectory with a sharp left turn in five scans and then a sharp right turn in 4 scans, is shown in Fig. 6.

With the same Gaussian noise input, a Monte Carlo simulation of 50 runs is performed for VDF, IE, and trained KF-SONFIN algorithms. The RMS of the estimation values is computed. Fig. 7 is the plot of the RMS errors of x and y positions versus time for the three algorithms and Fig. 8 is the plot of RMS errors of v_x and v_y versus time. Fig. 9 is the plot of the mean value of acceleration estimated by SONFIN over 50 runs. Fig. 10 shows that the system covariance of KF-SONFIN is adaptive to the estimated acceleration value. All of these results are the mean values over 50 runs.

Experiment 2: In this experiment, the initial position of the target is $(x, y) = (5\text{ km},\ 4\text{ km})$ with respect to the radar site



(a)



(b)

Fig. 10. Adapted system covariance of KF-SONFIN by the estimated acceleration in Experiment 1. (a) x direction. (b) y direction.

and with an initial speed of $(v_x, v_y) = (38 \text{ m/s}, 38 \text{ m/s})$ that is radially moving away from the radar site with a heading angle of 45°. The sampling time $T = 10$ s is chosen. The maneuver starts at time 470 s with $u_x = -1.3$ m/s^2, $u_y = -3.0$ m/s^2 and lasts for three scans, which is followed by a second maneuver occurring at 500 s with accelerations of $u_x = -1.9$ m/s^2, $u_y = -3.3$ m/s^2 for two scans. The 5 scan maneuvers make a U-turn trajectory as shown in Fig. 11. The measurement noise and the tracking parameters are set as those in Experiment 1 for all KF-SONFIN, VDF and IE algorithms. The tracking results are shown in Figs. 12 and 13 for comparisons in RMS errors of position and velocity. Fig. 14 is the plot of the mean value of acceleration estimated by SONFIN over 50 runs versus time. Fig. 15 shows that the system covariance of KF-SONFIN is adaptive to the estimated acceleration value. All of these results are the mean values over 50 runs.

Experiment 3: In this experiment, the initial position of the target is $(x, y) = (10 \text{ m}, 10 \text{ m})$ apart from the radar site with an initial speed of $(v_x, v_y) = (4 \text{ m/s}, 2 \text{ m/s})$. The target is then assumed to undergo a convention coordinate-turn maneuvers

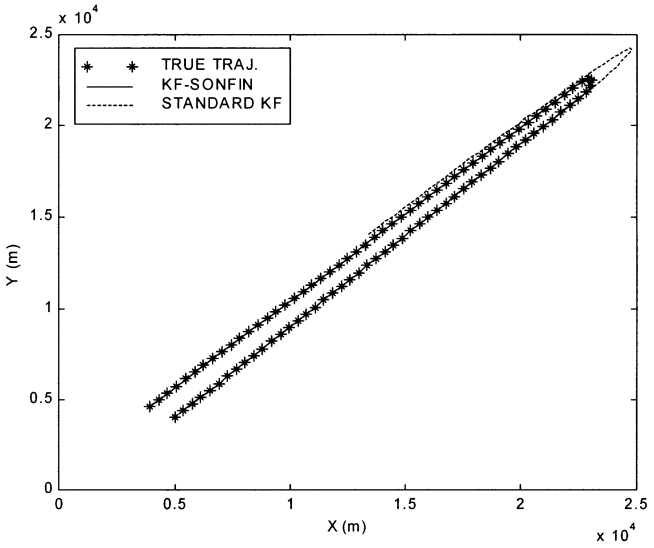
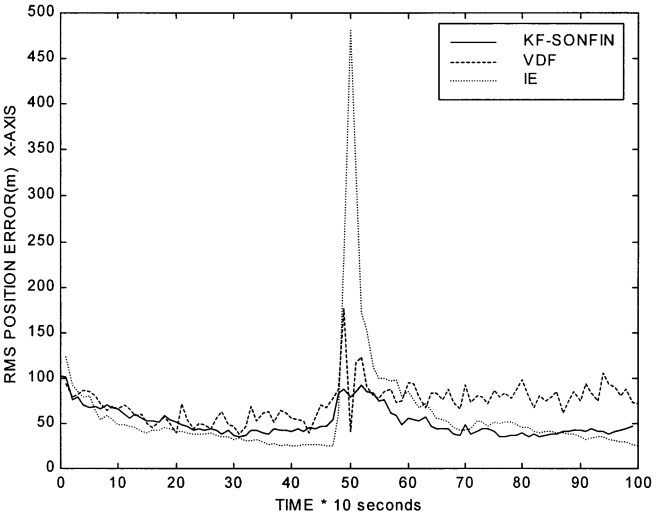
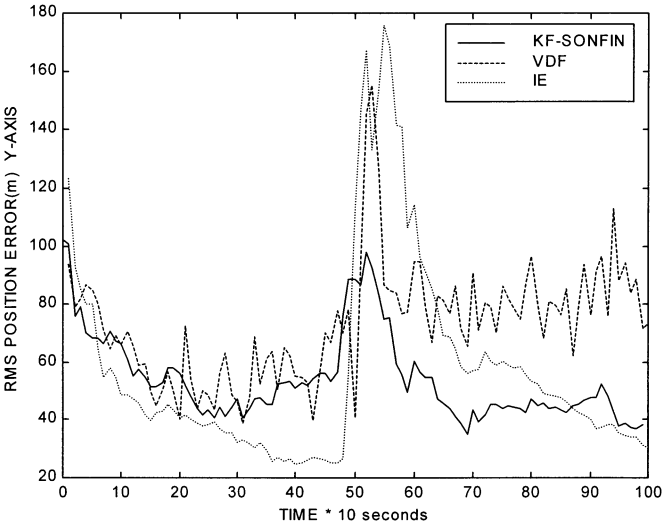


Fig. 11. Target's trajectory versus the tracking results of KF-SONFIN and standard Kalman filter with the mean values over 50 runs in Experiment 2.

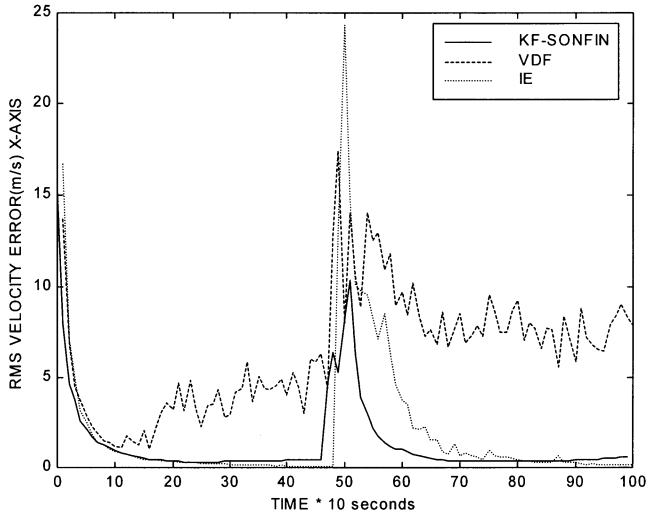


(a)

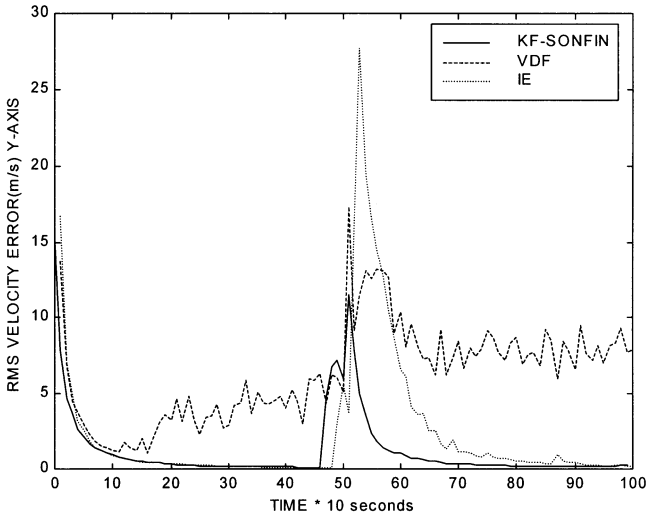


(b)

Fig. 12. Comparisons of RMS position tracking error in Experiment 2. (a) x direction. (b) y direction.

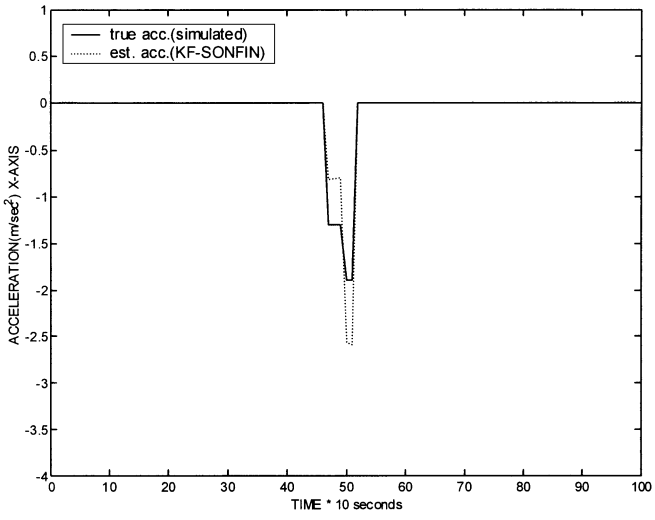


(a)

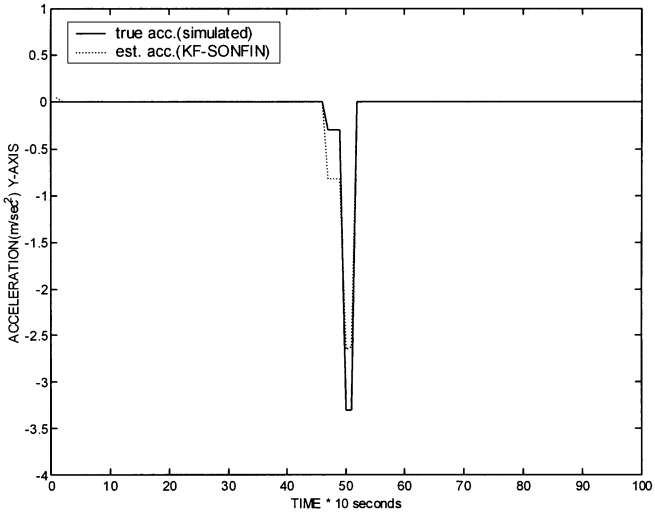


(b)

Fig. 13. Comparisons of RMS velocity tracking error in Experiment 2. (a) x direction. (b) y direction.



(a)



(b)

Fig. 14. Mean value of acceleration estimated by SONFIN versus true acceleration value in Experiment 2. (a) x direction. (b) y direction.

during the 200th s and 400th s (i.e., the 20th scan to the 40th scan with sampling time $T = 10$ s) by

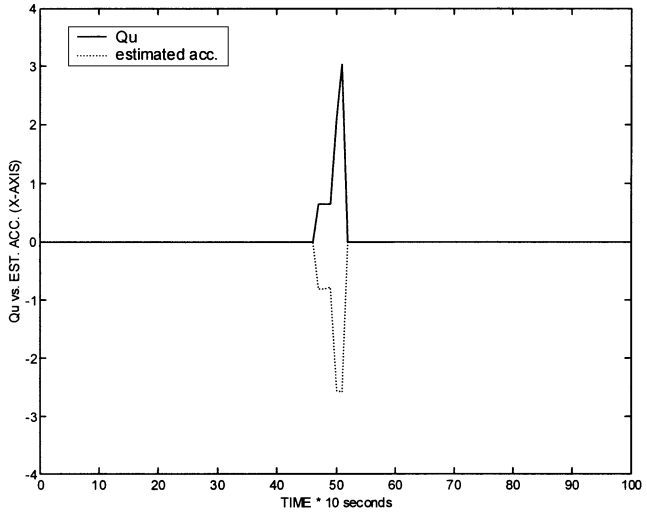
$$u_x = u_{x0} \times \sin[w_t(t - 200) + \phi_x]$$
$$u_y = u_{y0} \times \cos[w_t(t - 200) + \phi_y]$$

where u_{x0} and u_{y0} are the amplitudes of the acceleration in x and y directions and are set to 0.2 m/s^2 and 0.16 m/s^2, respectively. The turning rate w_t is set to 0.2 rad/s, the phase shift in x direction ϕ_x is set to 90° and in y direction ϕ_y is set to 0°. This maneuver makes the trajectory a more sharp left turn in four scans. After the above maneuver, the target path is with no maneuver during the 410th s and 500th s and then undergoes a second maneuver with $u_x = 0.2$ m/s^2, $u_y = 0.1$ m/s^2 from the 510th s to 550th s. This maneuver makes the trajectory a second sharp turn. The trajectory of the target and its corresponding accelerations with time-varying functions during the maneuvering period are shown in Figs. 16 and 19, respectively. The measurement noise and the tracking parameters are set as those in Experiment 1 for KF-SONFIN, VDF and IE algorithms. The tracking results are shown in Figs. 17 and 18 for comparisons. Fig. 19 is the plot of the mean value of acceleration estimated by SONFIN over 50 runs versus time. Fig. 20 shows that the system covariance of KF-SONFIN is adaptive to the estimated acceleration value. All of these results are the mean values over 50 runs.

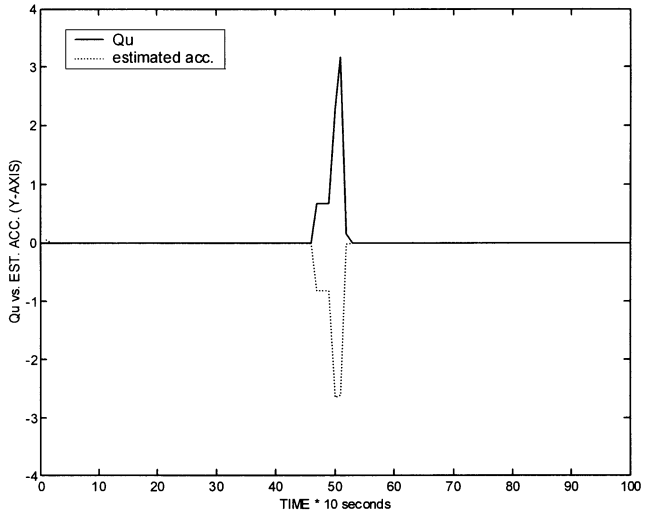
V. Discussions

From the results of Figs. 6, 11, and 16, it is clear that the standard Kalman filter cannot track the target while maneuver has occurred; however, the KF-SONFIN, due to its precise estimation of acceleration values and compensation, can always keep tracking the target. The experimental results show the validity of prediction of acceleration values, which are needed before the system is applied to particular moving targets.

Figs. 7–8, 12–13 and 17–18 show the performance comparisons of IE, VDF, and the proposed KF-SONFIN methods based



(a)



(b)

Fig. 15. Adapted system covariance of KF-SONFIN by the estimated acceleration in Experiment 2. (a) x direction. (b) y direction.

on the indexes of position and velocity RMS errors in Experiments 1, 2, and 3, respectively. The results indicate that the KF-SONFIN has the best performance. We shall analyze these results according to three kernel factors in target tracking.

1) real-time *detection* of maneuver occurrence;
2) accurate *estimation* of maneuver values;
3) efficient *compensation* to the tracking filter for reducing the maneuver's effects.

These three factors form the heart of the target-tracking mechanism and have direct impact on tracking errors.

First, let us consider the detection of maneuver. The IE and VDF methods consider innovation variations and use the technology of sliding window with significance test to detect the abrupt occurrence of maneuver for a constantly accelerating target moving straightly. This approach cannot real-time detect the onset time of maneuver efficiently and thus will prolong the response to the maneuver. This phenomena can be observed from Fig. 7 in Experiment 1, which indicates that the target changes

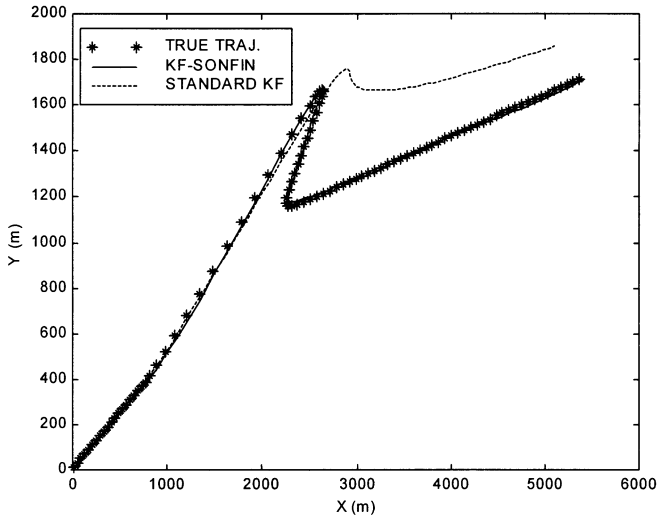
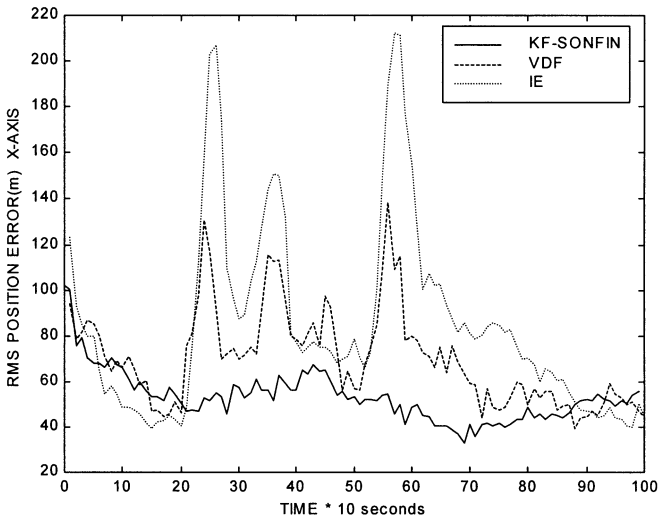
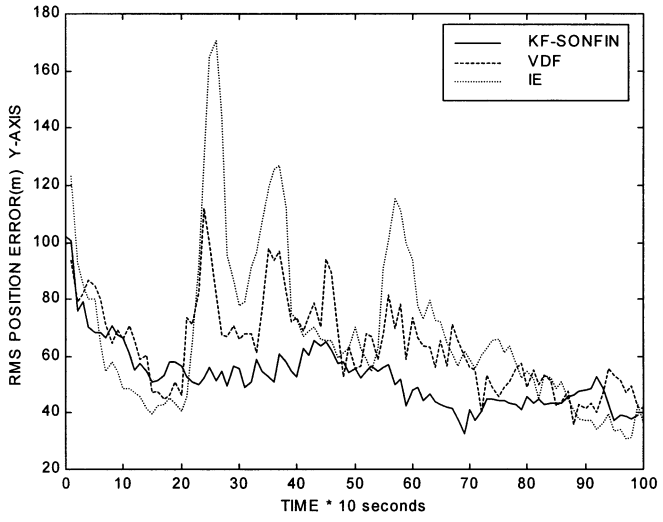


Fig. 16. Target's trajectory versus the tracking results of KF-SONFIN and standard Kalman filter with the mean values over 50 runs in Experiment 3.

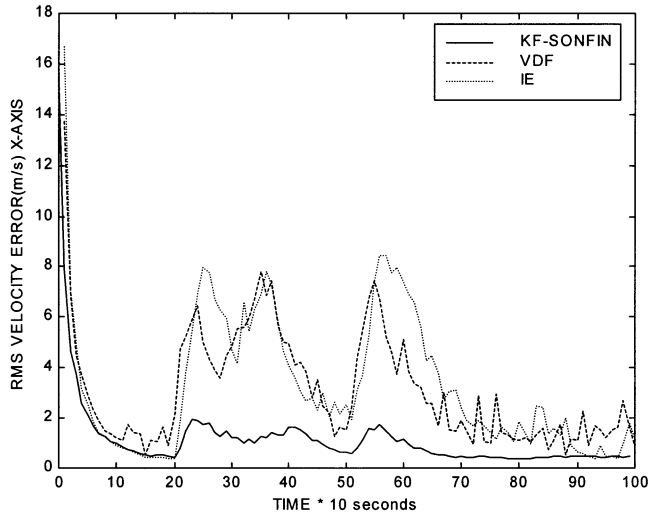


(a)

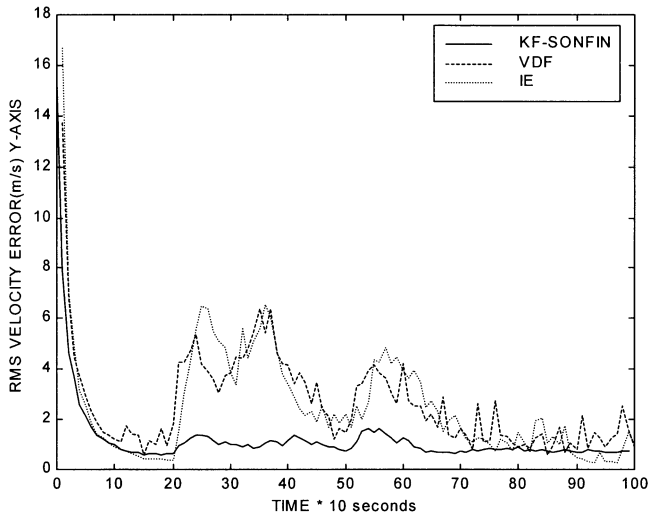


(b)

Fig. 17. Comparisons of RMS position tracking error in Experiment 3. (a) x direction. (b) y direction.



(a)



(b)

Fig. 18. Comparisons of RMS velocity tracking error in Experiment 3. (a) x direction. (b) y direction.

its acceleration and heading direction two times at time 400 s and 610 s, respectively. The IE and VDF methods detect each maneuver at about the 5th sample after its occurrence and thus produce large tracking errors. The situation is even worse for the IE method. On the contrary, the proposed KF-SONFIN method detects the occurrence of maneuver in real time and produces much less position and velocity tracking errors. Fig. 9 shows the trained SONFIN can detect the maneuver immediately. Experiments 2 and 3 show the same results. This is mainly due to the fact that the KF-SONFIN method considers the features of "heading change" and "Doppler frequency shift" in addition to the innovation values. The fusion of these three features can indicate the occurrence of maneuver accurately. Moreover, Figs. 7–8 and 12–13 indicate that the trajectories detected by the VDF method contain obvious pulsation. This is because the external stochastic disturbance existing in the testing measurement sequence makes the VDF switch alternatively between quiescent and nonquiescent states. This is the inherent "discontinuity" problem of the VDF approach.

Second, let us focus on the factor of estimation of maneuver values. After detecting the maneuver occurrence, the VDF method uses the formula, $(2/T^2)\cdot[\mathbf{z}(k-s)-\hat{\mathbf{z}}(k-s|k-s-1)]$, to estimate the acceleration at time $(k-s)$, where s is the effective length. The estimated acceleration value is used to calculate the velocity at that time. With the estimated velocity and the measured position, $\mathbf{z}(k-s)$, at time $(k-s)$, we can obtain the term $\mathbf{x}_m(k-s|k-s)$, which becomes the state estimate when the VDF enters its nonquiescent state (maneuver state). It is noted that when the noise disturbance at time $(k-s)$ is large, the estimated acceleration value will deviate from the true one greatly. This in turn will make the obtained state estimate, $\mathbf{x}_m(k-s|k-s)$, further away from the true one. This incorrect state estimate will obviously cause the Kalman filter to produce erroneous estimation. In the IE method, the estimation of maneuver values is based on the assumption of constant acceleration within the sliding window and the acceleration is estimated by the least square estimation algorithm. Obviously, the constant-acceleration assumption will make the IE method produce large error in the situations with varying accelerations. On the contrary, the error of KF-SONFIN comes directly from the training error of SONFIN, without the IE- or VDF- like chaining error propagation. Also, the training error can be further reduced by collecting more on-site training information and performing more extensive training on SONFIN.

Finally, the maneuver compensation scheme is considered. In our simulations, the VDF and KF-SONFIN methods adopt the same compensation scheme. They apply the estimated maneuver values to the state equation of Kalman filter and to tune the standard deviation of the filter's process noise for calculating the system covariance, Q. This scheme can compensate the tracking filter directly and efficiently to overcome the miss-tracking problem caused by maneuver. In other words, the values in Q are adaptive according to the estimated maneuver values. Traditionally, like the scheme used in the IE method, Q is constant, which is difficult to choose. A fixed larger Q will make the bandwidth of tracking filter large, which in turn will produce larger tracking errors in nonmaneuver period. On the other hand, a fixed small Q will result in larger tracking errors in the maneuver period. Figs. 10, 15, and 20 show that the system covariance of KF-SONFIN is adaptive according to the estimated maneuver. From the comparisons of the tracking error, they obviously indicate that such a Q-adaptive mechanism makes the tracking filter produce good compensation effects. It is noted that the KF-SONFIN outperforms the VDF since the former has more accurate estimation of maneuver values.

According to the above performance analysis based on the three factors of detection, estimation and compensation of maneuver, we find out the reasons why the proposed KF-SONFIN methods can outperform the other two compared counterparts. The fusion of three features, including "heading change" and "Doppler frequency shift" in addition to the common "innovation variation" and the learning capability of SONFIN are believed to play the crucial roles in contributing to the superiority of the proposed KF-SONFIN method for target tracking with maneuver.

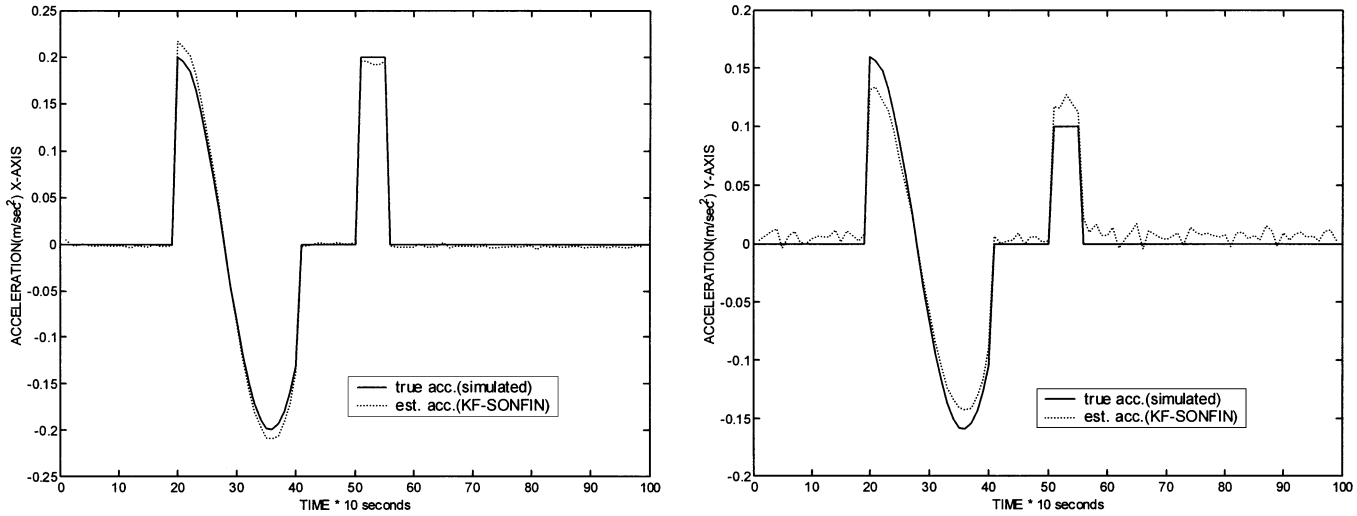


Fig. 19. Mean value of acceleration estimated by SONFIN versus true acceleration value in Experiment 3. (a) x direction. (b) y direction.

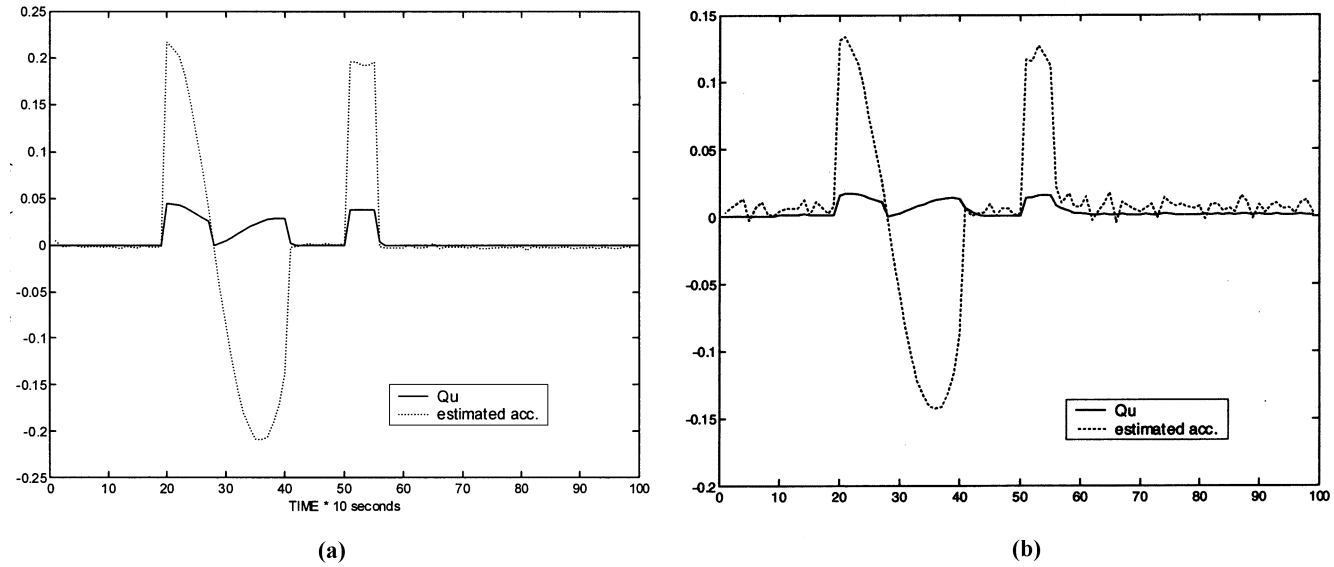


Fig. 20. Adapted system covariance of KF-SONFIN by the estimated acceleration in Experiment 3. (a) x direction. (b) y direction.

VI. Conclusion

In this paper we have proposed a neural fuzzy scheme for estimating the acceleration of a maneuvering target based on the measurement residual, heading change and Doppler shift feature extractions from the pulsed Doppler radar and Kalman filter. Although the features used for the neural fuzzy network are not unique, the features taken in this paper have strong relationships to the target maneuver and have ever been used in previous researches as described in Section III. These features are considered and fused simultaneously to estimate the accelerations in our KF-SONFIN scheme, which is different from many previous researchers who have used only one of these features to find the accelerations in their schemes. The major contribution of this study is a demonstration of the ability of the proposed SONFIN to fuse the information from different sources which are nonlinear to the desired output in a radar automatic tracking system. In addition, the scheme of the adaptive system covariance of Kalman filter is derived when the acceleration of the tracked target is detected and estimated. Using the developed algorithm, when a tracked target maneuver occurred it can be detected immediately to estimate the magnitude and the maneuver value accurately in short time intervals and then the tracking filter can be compensated correctly by the estimated acceleration and system covariance. Simulation results show that a well-trained SONFIN always produces estimated output very close to the true maneuver values that lead to good tracking performance and avoid miss-tracking when the maneuver occurs.

Although some intelligent adaptation schemes such as neural networks and fuzzy logic etc. have been proposed recently, they cannot easily to train the neural network well enough and obtain good fuzzy rules. However, the self-constructing neural fuzzy inference network (SONFIN) can find its optimal structure and parameters automatically and always produces an economical networks size and learning speed. Comparing

with the traditional schemes IE and VDF, SONFIN does not need to model the maneuver input nor augment the dimension of the tracking filter. In addition, from the simulation results SONFIN shows that the performance is superior to the traditional methods. This is an important benefit in a practical military surveillance system or a civil air traffic control (ATC) system.

REFERENCES

[1] A. Farina and F. A. Studer, *Radar Data Processing*. New York: Wiley, 1985, vol. 1, Introduction and Tracking.

[2] B. Edde, *Radar: Principles, Technology, Applications*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[3] ——, *Fundaments of Radar*. Piscataway, NJ: Institute of Electrical and Electronic Engineers, Inc., 2000.

[4] G. Galati, M. Naldi, and M. Ferri, "Airport surface surveillance with a network of miniradars," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 5, pp. 331–338, Jan. 1999.

[5] A. Farina and S. Pardini, "Track-while-scan algorithm in a clutter environment," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-14, pp. 769–779, Sept. 1978.

[6] P. R. Kalata, "The tracking index: A generalized parameter for α-β and α-β-γ target trackers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-20, pp. 174–182, Mar. 1984.

[7] W. D. Blair, "Two-stage alpha-beta-gamma estimator for tracking maneuvering targets," in *Proc. 1992 Amer. Contr. Conf.*, Chicago, IL, June 1992, pp. 842–846.

[8] R. G. Brown, *Introduction to Random Signals and Applied Kalman Filtering*. New York: Wiley, 1997.

[9] M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[10] Y. T. Chan, A. G. C. Hu, and J. B. Plant, "A Kalman filter based tracking scheme with input estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, pp. 237–244, Mar. 1979.

[11] Y. Bar-Shalom and K. Birmiwal, "Variable dimension filter for maneuvering target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-18, pp. 621–629, Sept. 1982.

[12] L. Chin, "Application of neural networks in target tracking data fusion," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, pp. 281–287, Jan. 1994.

[13] Z. Jing, H. Xu, and X. Zhou, "Information fusion and tracking of maneuvering targets with artificial neural network," in *Proc. IEEE Int. Conf. Neural Networks (ICNN'94)*, 1994, pp. 3403–3408.

[14] T. Tao, "A neural decision estimator for maneuvering targets," in *Proc. IEEE Int. Conf. Neural Networks (ICNN'94)*, 1994, pp. 3926–3931.

[15] M. K. Sundareshan and F. Amoozegar, "Neural network fusion capabilities for efficient implementation of tracking algorithms," *Opt. Eng.*, vol. 36, no. 3, pp. 692–707, Mar. 1997.

[16] F. Amoozegar, "Neural-network-based target tracking state-of-the-art survey," *Opt. Eng.*, vol. 37, no. 3, pp. 836–846, Mar. 1998.

[17] J. Lalk, "Intelligent adaptation of Kalman filters using fuzzy logic," in *Proc. IEEE 3rd Conf. Fuzzy Syst.*, 1994, pp. 744–749.

[18] K. C. C. Chan, V. Lee, and H. Leung, "Radar tracking for air surveillance in a stressful environment using a fuzzy-gain filter," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 80–89, Feb. 1997.

[19] S. McGinnity and G. W. Irwin, "Fuzzy logic approach to maneuvering target tracking," *Proc. Inst. Elect. Eng.*, vol. 145, pp. 337–341, Dec. 1998.

[20] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 12–32, Feb. 1998.

[21] C. T. Lin, *Neural Fuzzy Control Systems With Structure and Parameter Learning*, Singapore: World Scientific, 1994.

[22] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[23] K. Mehrotra and P. R. Mahapatra, "A jerk model for tracking highly maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, pp. 1094–1105, Oct. 1997.

[24] R. K. Mehra, "Approaches to adaptive filtering," *IEEE Trans. Automat. Contr.*, vol. AC-17, pp. 693–698, Oct. 1972.

[25] N. H. Gholson and R. L. Moose, "Maneuvering target tracking using adaptive state estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-13, pp. 310–317, May 1977.

[26] F. R. Castella, "An adaptive two-dimensional Kalman tracking filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-16, pp. 822–829, Nov. 1980.

[27] C. B. Chang and J. A. Tabaczynski, "Application of state estimation to target tracking," *IEEE Trans. Automat. Contr.*, vol. AC-29, pp. 98–109, Feb. 1984.

[28] Y. N. Chung, D. L. Gustafson, and E. Emre, "Extended solution to multiple maneuvering target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-26, pp. 876–887, Sept. 1990.

[29] T. C. Wang and P. K. Varshney, "A tracking algorithm for maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, pp. 910–924, July 1993.

[30] R. A. Best and J. P. Norton, "A new model and efficient tracker for a target with curvilinear motion," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, pp. 1030–1037, July 1997.

[31] J. P. Helferty, "Improved tracking of maneuvering targets: The use of turn-rate distributions for acceleration modeling," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-32, pp. 1355–1361, Oct. 1996.

[32] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-6, pp. 473–483, July 1970.

[33] A. Munir and D. P. Atherton, "Maneuvering target tracking using different turn rate models in the interacting multiple model algorithm," in *Proc. 34th IEEE Conf. Decision Contr.*, New Orleans, LA, Dec. 1995, pp. 2747–2751.

[34] M. Efe and D. P. Atherton, "Maneuvering target tracking using adaptive turn rate models in the interacting multiple model algorithm," in *Proc. 35th IEEE Conf. Decision Contr.*, Kobe, Japan, Dec. 1996, pp. 3151–3156.

[35] ——, "Maneuvering target tracking with an adaptive Kalman filter," in *Proc. 37th IEEE Conf. Decision Contr.*, Tampa, FL, Dec. 1998, pp. 737–742.

[36] S. R. Rogers, "Tracking targets with constant heading and variable speed," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, pp. 543–547, May 1990.

[37] R. J. Evans, F. Barker, and Y. C. Soh, "Maximum likelihood estimation of constant heading trajectories," in *Proc. IEEE Int. Radar Conf.*, London, U.K., Oct. 1987, pp. 489–493.

[38] F. R. Castella, "Tracking accuracies with position and rate measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-17, pp. 433–437, May 1981.

[39] K. V. Ramachandra, B. R. Mohan, and B. R. Geetha, "A three-state Kalman tracker using position and rate measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, pp. 215–222, Jan. 1993.

[40] B. Armstrong and B. S. Holeman, "Target tracking with a network of Doppler radars," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, pp. 33–48, Jan. 1998.

[41] R. L. Moose, H. F. Vanlandingham, and D. H. McCabe, "Modeling and estimation for tracking maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, pp. 448–456, May 1979.

[42] P. L. Bogler, "Tracking a maneuvering target using input estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-23, pp. 298–310, May 1987.

Fun-Bin Duh received the B.S. degree in electronic engineering and the M.S. degree in automatic control engineering from Feng-Chia University, Taichung, Taichung, R.O.C., in 1981 and 1983, respectively, and is currently pursuing the Ph.D. degree in the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan.

Since 1983, he has been with the Chung-Shan Institute of Science and Technology (CSIST), Taiwan, R.O.C., where he is currently an Associate Scientist and Senior Specialist. His current research interests include fuzzy systems, neural networks, estimation theory, radar tracking, signal processing, and servo control systems.

Chin-Teng Lin (S'88–M'91–SM'95) received the B.S. degree in control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., in 1986 and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1989 and 1992, respectively.

Since August 1992, he has been with the College of Electrical Engineering and Computer Science, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., where he is currently a Professor and Chairman of Electrical and Control Engineering Department. He served as the Deputy Dean of the Research and Development Office of the National Chiao-Tung University, from 1998 to 2000. His current research interests are fuzzy systems, neural networks, intelligent control, human-machine interface, image processing, pattern recognition, video and audio (speech) processing, and intelligent transportation system (ITS). He is the co-author of *Neural Fuzzy Systems—A Neuro-Fuzzy Synergism to Intelligent Systems* (Englewood Cliffs, NJ: Prentice-Hall) and the author of *Neural Fuzzy Control Systems with Structure and Parameter Learning* (New York: World Scientific). He has published over 60 journal papers in the areas of soft computing, neural networks and fuzzy systems, including 40 IEEE TRANSACTIONS papers.

Dr. Lin received the Outstanding Research Award from the National Science Council (NSC), Taiwan, R.O.C., since 1997, the Outstanding Electrical Engineering Professor Award from the Chinese Institute of Electrical Engineering (CIEE) in 1997, and the Outstanding Engineering Professor Award from the Chinese Institute of Engineering (CIE), in 2000. He was also elected to be one of the 38th Ten Outstanding Young Persons in Taiwan, R.O.C., in 2000. He is a member of Tau Beta Pi and Eta Kappa Nu. He is also a member of the IEEE Computer, IEEE Robotics and Automation, and IEEE Systems, Man, Cybernetics Societies. He was the Executive Council Member of the Chinese Fuzzy System Association (CFSA), from 1995 to 2001. He has been the President of CFSA since 2002 and Supervisor of the Chinese Automation Association since 1998. Since 2000, he has been the Chairman of the IEEE Robotics and Automation Society, Taipei, Chapter and the Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, the IEEE TRANSACTIONS ON FUZZY SYSTEMS, and *AUTOMATICA*.