International Journal of Innovative
Computing, Information and Control
Volume **14**, Number **4**, August **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 1481–1493

SWITCHED MULTI-MODEL ESTIMATION USING PROBABILISTIC NEURAL NETWORK DECISION FOR MANEUVERING TARGET TRACKING

Zhiguo Shi[1], Yuankai Li[2,*] and Zeying Wang[1]

[1]School of Computer and Communication Engineering
University of Science and Technology Beijing
No. 30, Xueyuan Road, Haidian District, Beijing 100083, P. R. China

[2]School of Aeronautics and Astronautics
University of Electronic Science and Technology of China
No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chengdu 611731, P. R. China
*Corresponding author: yuankai.li@alumni.sjtu.edu.cn

Received November 2017; revised March 2018

Abstract. *Target motion model is usually unknown due to the uncertainty of target maneuver. For maneuvering target tracking, a typical way is to use the multi-model (MM) based methods, where multiple filters work in parallel on different motion models with different state transition functions. However, paralleled working of multiple models is unnecessary for target with typical maneuver and remarkably increases the computation complexity and algorithm structure. In this paper, we propose a novel multi-model method based on probabilistic neural network (PNN) decision. In the method, target model switches among a designed model set and only one model runs during any sample time. It contains two main steps. First, a model set of target motion is trained and classified by a detector built on PNN. Second, the detector determines the current motion model and based on it uses a standard Kalman filter (KF) for optimal estimation. Different from the way of paralleled working of the usual MM methods, the proposed PNN-MM method runs only one filter at any single moment, which makes the algorithm structure simplified and calculation complexity decreased. Simulation results have shown the improved performance of the PNN-MM method for tracking targets with typical maneuver and less computational load compared with traditional MM methods.*

Keywords: Decision-based tracking, Probabilistic neural network, Multiple models, Kalman filter, Maneuvering target tracking

1. Introduction.

Maneuvering target tracking (MTT) is a fundamental problem in both military and civil fields. The main objective of MTT is to confirm existence of target in complex environment and to identify the target motion in real time. Presently, almost all the MTT methods are model-based where the modeling of target motion is essential to the quality of tracking. However, in practical situation, the targets having maneuvering capability may change motion types at any arbitrary moment, so it is quite hard for a tracking scheme to determine the target motion model quickly and accurately.

A typical way to cope with the uncertainty of target motion is to use a multi-model (MM) based estimation method [1], which has undergone several stages of development. Magill derived the first generation of MM method [2]. It uses a static model set composed of a bundle of models to describe target motion. Blom and Bar-Shalom put it forward and presented an interacting multi-model (IMM) algorithm [3]. The IMM algorithm allows filters working jointly with calculated possibilities, making it adaptive to more complicated

DOI: 10.24507/ijicic.14.04.1481

target motion [4]. However, under practical environments, target tracking requires a huge number of models to describe the target motion. That will cause large amount of calculation and increase algorithm structure. Additionally, in the process of estimation fusion, competition among models will lead to degradation of tracking performance. To solve the problem, Li and Vesselin built a variable structure multi-model (VSMM) algorithm [5-8] where multiple model sets are introduced. The algorithm switches models among model sets, making the algorithm focused on the most possible part of models rather than all the models in model space. The VSMM method has better adaptability to target motion due to hierarchical structure of model sets, and the critical task in VSMM estimation is to determine all the candidate model sets and decide the one with the most possibility. Compared with the IMM algorithm where all the models in model space must be used for calculation, the VSMM method has less computational load but more complicated structure [9].

For the VSMM method, if target motion is quite complex and has numerous mixed maneuver types, the most possible model set will approximate to the range of all the candidate sets, and the method will degenerate to the IMM problem. However, if target motion has no mixed maneuver type, i.e., the target maneuver has only one primary type at each time span, the most possible model set will shrink to one single model and the other models do not participate in estimation calculation. That will revert to a decision-based single model method.

Therefore, for the target without mixed maneuver type, it is unnecessary to use hierarchical structure of the VSMM method. Any choice of model sets is inaccurate, leading to tracking error. In order to give a convenient solution for tracking such kind of maneuvering target, in this paper, a novel switched multi-model tracking method is proposed. In the method, a model switching mechanism is introduced to replace the complicated model interaction mechanism in IMM and VSMM methods. The algorithm structure becomes much simpler and requires less computation. The switched model is adopted to establish an optimal Kalman filter (KF) for target motion estimation and determined by using a decision technique based on probabilistic neural network (PNN).

PNN is a desired tool that can help obtain the optimal Bayesian decision result [10-13], arising from neural network theory. In early studies, the back propagation (BP) network [14] is expressed first to solve the problem of learning in multilayer network and realize the idea of multilayer neural network. Powell built a multi-variate differential method of radial basis function (RBF) [15], based on which the radial basis function neural network is established [16]. Hopfield proposed a neural network model with a feedback loop inter-connection, where the concept of energy function is introduced. The Hopfield model is used as a base to produce recurrent neural network (RNN) [17,18]. Specht early researched the probabilistic neural network (PNN) [12,13]. By combining statistical signal processing methods with the neural networks to make decision, PNN can realize Bayesian optimization. The PNN-based decision technique has been applied successfully in many fields, such as feature extraction [19], image classification [20], fault recognition [21], battery state estimation [22], and sound source localization [23], and verified its advantages on decision performance.

As pointed out in [24,25], we may list the advantages of PNN as follows: 1) easy training and fast convergence, much suitable for real-time processing; 2) adaptive to arbitrary nonlinear transformation with similar rule surface of Bayes optimal decision; 3) good robustness to uncertainty; 4) flexible design of kernel function for estimating probability density, making classification results insensitive to the kernel function; 5) fixed number of neurons in each layer for easy implementation.

Considering the advanced characteristics above, in the proposed method we adopt PNN to form the decision maker to achieve the tracking objective. We name the proposed method as PNN-MM. It is a decision-based method, using switched multiple models rather than candidate model sets as in VSMM, so at each time point only one single model runs for filtering estimation. Simulation results have been shown to illustrate the advantage of the PNN-MM compared with other typical MM methods.

The rest of the paper is organized as follows. The overall structure of the PNN-MM tracking scheme is presented in Section 2. The detailed algorithm including model space, feature extraction, PNN detector and Kalman filtering is presented in Section 3. Section 4 shows the simulation results and discussion, and Section 5 concludes the paper.

2. **Structure of the Tracking Scheme.** Considering the target motion without mixed maneuver type, we minimize the number of models in the possible model set. That is to say, at each sampling time, there is only one possible model in the model space required for calculation. When target maneuvers, target movement pattern will jump from one model to another, and each model can be regarded as a Gaussian random subsystem [26]. To each subsystem, we can use a Kalman filter to estimate the states of target motion. On the other hand, if we can obtain the current most possible motion model by using decision technique before filtering, all the subsystems will require just one filter for state estimation. That will eliminate the unnecessity of the complexity caused by paralleled computation of a model set in IMM and VSMM methods, simplifying the algorithm structure and avoiding the performance loss from model competition.

In order to determine the model with most possibility, it is necessary to design a decision maker, which can detect and classify the measurements acquired from sensors before filtering estimation. Through classification, the decision maker can determine the most possible target motion model of the current time and use filter to yield the most appropriate estimation result under the current motion model.

To achieve the measurement classification and model decision, a desired method is to use neural networks (NN). The NN has been already used in MM tracking [10,11], and in recent decade, statistical method is introduced into NN and generates a type of probabilistic neural network (PNN) [12,13]. Applied in the classification and decision problem, PNN can help obtain the desired optimal Bayesian results. For this reason, we use PNN for switching decision among multiple models.

The structure of the PNN decision-based MM tracking scheme is designed as in Figure 1. It is composed of two parts: the model decision and the filtering estimation. In the part of model decision, there are a model space, a feature extraction module and a PNN decision maker module. Z_k and S_k represent the input measurements to target before and after feature extraction, respectively. The filtering estimation part contains a switching module and a filter set. The filters are dependent with the models in the model space, X_k showing the state estimates.

As shown in Figure 1, the tracking scheme is a switched jump linear system. It can be degraded to a fundamental linear stochastic system if the model m_j is determined. Note the model space as M and suppose $m = m_j \in M$. The system can be described as

$$X_{k+1} = F^j(k)X_k + G^j(k)w_k^j \tag{1}$$

$$Z_k = H^j(k)X_k + v_k^j \tag{2}$$

where X_k and Z_k are the state and measurement of the target motion at time k, respectively. F^j and H^j are the transition matrix and the measurement matrix for the model m_j, and G^j the process noise matrix. w^j and v^j are the process and measurement noises. Suppose w^j and v^j are uncorrelated Gaussian white with covariance matrices Q and R,

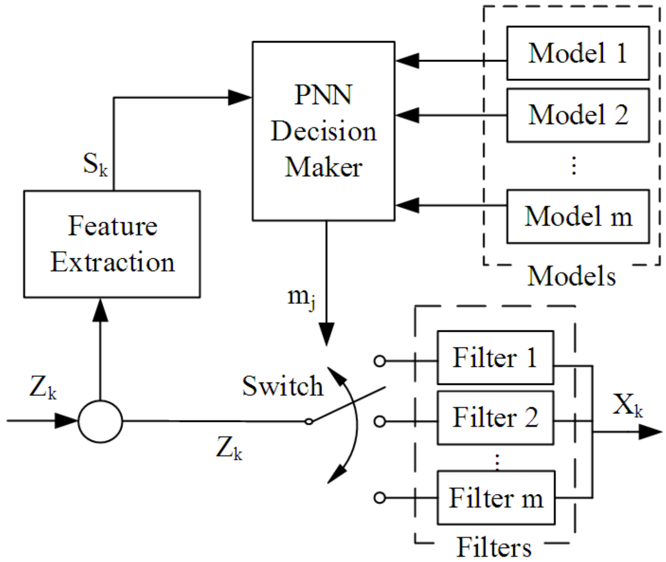

FIGURE 1. Structure of the PNN-MM tracking scheme

respectively. For the model m_j that has been determined, we may use Equations (1) and (2) to establish the Kalman filter for state estimation. Clearly, the remained objective is making switching decision to determine the right model of current target motion. Because target motion jumps among different models in M, we design a PNN as the decision maker to control the model switching and let the corresponding filter be connected.

Therefore, according to the framework shown in Figure 1, the tracking scheme can be designed as a four-step process.

Step 1. Feature Extraction. The feature information is extracted from the measurement data in order for PNN use. As in Figure 1, the measurement Z_k is transformed into S_k with a given rule.

Step 2. PNN Decision. Use known model types to train the PNN and initialize the network weight. Then, use the trained PNN to classify the feature data S_k, and judge which kind of target motion currently has the highest probability. The PNN outputs the current model m_j from the model space.

Step 3. Model Switching. Based on the PNN decision result, the switching module switches the target motion model to the objective model, and connect the corresponding filter for state estimation.

Step 4. State Estimation. With the most possible target motion model and measurement Z_k input to the filter set, use the yielded optimal Kalman filter to estimate the target motional state as the output of target tracking.

The above 4 steps present one cycle of PNN-MM tracking algorithm at the sampling time k. Repeating them at the time $k + 1$, we may complete the PNN-MM method for maneuvering target tracking.

There are several advantages that we can know from the tracking scheme structure. First, only one single filter works all through the tracking procedure. Computation complexity caused by paralleled filtering in IMM and VSMM is reduced remarkably. Second, model competition is prevented. For the target without mixed maneuver type, no performance loss is generated from model fusion. Third, PNN decision makes the switched jump system degrade to a linear stochastic system where we can use an optimal Kalman filter conveniently for state estimation. Fourth, the scheme structure is simple and can be easily applied in practical implementation.

3. The PNN-MM Tracking Algorithm. In this section, we present the specific PNN-MM algorithm for tracking a maneuvering target without mixed maneuver type. Because PNN has many advantages such as easy training, fast convergence and robustness for real-time processing, capability of obtaining the Bayesian optimal decision result, and fixed structure for convenient implementation that are pointed out in [27-31], the PNN-based algorithm has such fundamental properties as well. Generally, it can be expressed by four sequenced parts that are model space design, measurement feature extraction, PNN-based model decision and Kalman filtering estimation.

3.1. Model space design. For the sake of simplicity, assume that the target maneuvers alternatively among 4 typical motions, i.e., constant acceleration (CA), constant velocity (CV), constant turn to left (CTL) and constant turn to right (CTR). So, we may use the four types of motion models to construct the model space.

Note the state vector $\boldsymbol{x} = [x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}]^T$, the components of which represent position, velocity and acceleration, respectively, on x axis and y axis. The discrete CA motion model can be written as

$$X_{k+1} = F_{CA}X_k + G_{CA}(k)w_k \tag{3}$$

If let $\ddot{x} = \ddot{y} = 0$, the CA motion model will revert to the CV model as

$$X_{k+1} = F_{CV}X_k + G_{CV}(k)w_k \tag{4}$$

In constant turn (CT) motion, target has a constant turning rate noted as ω. The signal of ω shows the turn direction. Positive signal means turning to left and negative for right. The CT model can be expressed as

$$X_{k+1} = F_{CT}(\omega)X_k + G_{CT}(\omega)w_k \tag{5}$$

Typically, we use multiple CT models with different constant values of turning rate to describe CT motion. The sequence of turning rate ω_k can be modeled as a Markov chain taking values in the set $\{\omega_1, \omega_2, \ldots, \omega_n\}$. In this paper, in order to simplify the problem, we assume that the target has two kinds of turning model which are left- and right-turning motion with turning rate ω_1 and ω_2 and $\omega_1 = -\omega_2$.

Denote the model space of target motion as M. Then, M can be composed of the above 4 types of motion models: CA, CV, CTR and CTL, noted as m_1, m_2, m_3 and m_4, respectively. That is, $M = \{m_1, m_2, m_3, m_4\}$.

3.2. Measurement feature extraction. In the model space designed above, we need identify the real target motion from real-time measurements. The first step is to evaluate the measurements by feature extraction.

Suppose that only relative position is detected by sensor. The measurement can be denoted as $Z_k = [x_k, y_k]^T$. We define a 9-dimensional feature vector

$$S_k = \{s_1, s_2, \ldots, s_9\} \tag{6}$$

where

$$s_1 = \begin{cases} 1, & \dot{x}_k > 0 \\ -1, & \dot{x}_k < 0 \end{cases} \text{ and } s_2 = \begin{cases} 1, & \dot{y}_k > 0 \\ -1, & \dot{y}_k < 0 \end{cases}$$

showing directions of target motion, i.e., four quadrants in the coordination system. $s_3 = \ddot{x}_k$, $s_4 = \ddot{y}_k$ and $s_5 = a_k$. a_k represents amplitude of the acceleration. $s_6 = \Delta\ddot{x}_k$, $s_7 = \Delta\ddot{y}_k$ and $s_8 = \Delta a_k$ are the difference value of s_3, s_4 and s_5, respectively. s_9 is a bias value. All the components in S_k should be normalized by a sigmoid function as

$$f_{sigmoid}(s) = \frac{2}{1 + e^{-\alpha s}} - 1 \tag{7}$$

making them distributed at $[-1, 1]$ for easy use. With such transformation, the critical features of measurement vector Z_k can be extracted and shown by the 9-dimensional vector S_k.

It is noticeable that the rule of feature extraction is built upon physical characteristics of target movement. To acquire desirable results of feature extraction, target motion types should be distinguishable. In this paper, the assumed motion types are typical and different, so the feature extraction works desirably. If the motion types are quite similar, the dimension of feature vector should be augmented to improve the distinguishing capability.

Based on the extracted feature, we may move to the second step for model decision. If using S_k as the input, the output should be the determined target model, named current motion model m_j, $j = 1, 2, 3$ or 4. It can be written with matrix form as $m_1 = [1\ 0\ 0\ 0]^T$, $m_2 = [0\ 1\ 0\ 0]^T$, $m_3 = [0\ 0\ 1\ 0]^T$ and $m_4 = [0\ 0\ 0\ 1]^T$. Then, a decision scheme is necessary to be developed.

3.3. **PNN-based model decision.** We adopt a PNN as the decision mechanism. The typical structure of a PNN is a four-layer structure [32,33], including an input layer, a pattern layer, a summation layer and a competitive layer. The structure is shown in Figure 2.

In the input layer, neural units receive independent variables as input. Nevertheless, these units do not perform computation. They just distribute the input data to the neurons. For the case of maneuvering target tracking, the input data is the extracted feature vector from measurements, i.e., S_k and sent it to the pattern layer. The pattern layer is a layer storing all the training samples and categorizes them into several groups. For the assumed case, training samples will be grouped into four motion classes. The

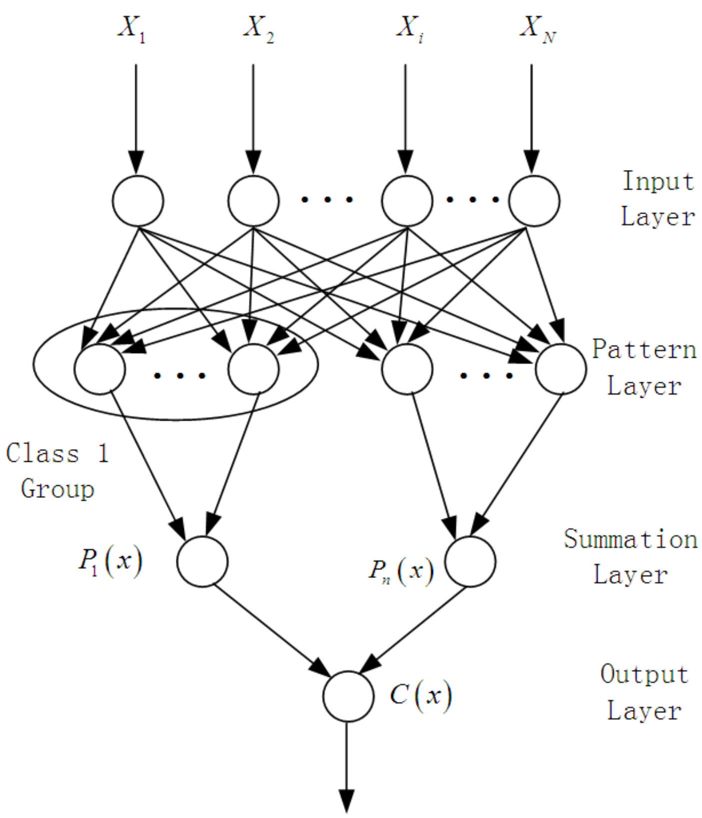

FIGURE 2. The structure of probabilistic neural network

output of the pattern layer passes through the neural unit of the summation layer and generates the probability of each possible category. Finally, the competitive layer uses a decision criterion to pick out the category with the largest probability and outputs the decision result of the PNN.

Consider a PNN having p-dimensional input feature vector $\chi = \{x_1, x_2, \dots, x_p\}$ and K types of classifiers $\{m_1, m_2, \dots, m_K\}$. The decision result of the PNN is determined based on a Bayes criterion as

$$h_j f_j(\chi) = \max\{h_k f_k(\chi)\}, \quad k = 1, 2, \dots, K \tag{8}$$

where $f_k(\chi)$ is the probability density function (PDF) of the test vector χ belonging to the category k, and h_k is a prior probability of the class k. If Equation (8) is satisfied, the PNN will output $C(\chi) = C_j$ as the decision result.

To enhance the smoothness and continuity, class group is also used as parent probability density of a single pattern [22]. For example, the PDF of a test set χ belonging to class A can be written as

$$f_A(\chi) = \sum_{l=1}^{n^A} h_l f_{Al} \tag{9}$$

where

$$f_{Al} = \exp\left(-\frac{(\chi - \chi_{Al})^T(\chi - \chi_{Al})}{2\sigma^2}\right) \tag{10}$$

In the above two equations, n^A is the number of training samples. l is the sequence number of training sample. χ_{Al} is the lth test sample in class A and σ the smoothing parameter.

With Equations (8), (9) and (10) considered, in the case of maneuvering target tracking, the motion model can be determined by the PNN decision as follows. The input feature vector χ is the 9-dimensional vector S_k. All the training samples are grouped into 4 categories and stored in the pattern layer. Once the sensor acquires new measurement, the network detects the input and makes classification immediately. The summation layer calculates the probability density of each category, and the final competitive layer compares the probability densities $h_k f_k(\chi)$ and finds the highest one, letting $C_j = 1$ and others setting 0, and outputs m_j as the decision result of the current motion model.

3.4. **Decision-based Kalman filtering.** After PNN determined the target motion model, we may use Kalman filter to estimate the motional states. Kalman filtering is a recursive linear minimum variance estimation method [12]. The major function of KF is to extract useful part from the noised measurements based on the proper state model.

According to the four different patterns of target motion, we can design the filter set as four Kalman filters running on four motion models, respectively. Once the model is determined or updated in real time, the filter works alternatively. The CV model m_1 with the transition matrix F_{CV} is used in Filter$_1$. Similarly, the CA model m_2 with F_{CA} is used for Filter$_2$. The CTR model m_3 with turning rate ω_1 and transition matrix $F_{CT}(\omega_1)$ is for Filter$_3$, and the CTL model m_4 with ω_2 and $F_{CT}(\omega_2)$ for Filter$_4$. The filters switch according to the model determined by PNN decision.

In Kalman filtering, it needs to give the initial state estimation $\hat{X}(0|0)$ and covariance matrix $P(0|0)$. Given the measurement Z_{k+1} at the next time step, it can obtain the state estimation $\hat{X}_{k+1|k+1}$ and its covariance matrix $P_{k+1|k+1}$. The detailed KF algorithm is well-known and will not be given in this paper.

However, it should be noticed that estimation accuracy will decrease at the initial and each filter switching time. It is because the KF aims to reduce the covariance of the state prediction and the measurement, and the state prediction and the measurement should have the same tendency of convergence. However, when target maneuvers, state model changes so that the state prediction will no longer follow the variation of the measurement. At that time point, the KF needs reset the state covariance matrix P to start a new motion stage. That results in the estimation deviation fluctuated temporarily.

Combining the Kalman filtering estimation with previous feature extraction and PNN model decision, the PNN-MM tracking algorithm is presented completely.

4. Numerical Simulations.

In this section, we use a numerical example of maneuvering target tracking to illustrate the feasibility of the PNN-MM tracking method. Suppose that the target is a particle moving on a 2-dimentional coordinate system. The target motion follows the CA, CV, CTL and CTR models with random switching. A PNN is adopted to judge the current motion model and switch to the corresponding Kalman filter for target state estimation. Also, we compared the PNN-MM tracking method with other typical MM methods to demonstrate the advantages on tracking performance.

4.1. Decision and estimation result.

Considering that the neural network should be trained before testing, we designed a training trajectory of target motion as in the left figure of Figure 3. In the simulation case, the target has 4 motion types corresponding to $m_1 = 1$, $m_2 = 2$, $m_3 = 3$ and $m_4 = 4$. m_1: go straight with constant speed; m_2: go straight with constant acceleration or deceleration; m_3: turn left with constant turning rate that equals -0.2 rad/s; m_4: turn right with the turning rate 0.2 rad/s. These 4 types are typical and can demonstrate most two-dimensional motion. The training trajectory includes all the 4 motion types and all the moving directions (4 quadrants in the coordinate system plane). After network training, the PNN is initialized for real testing of the target tracking case.

The real testing trajectory of the target is shown in the right figure of Figure 3. In order for clear description, we note each section in the trajectory. The tested target motion lasts 82 seconds, and the measurement frequency is 0.1 seconds, totally measuring 820 times. The target starts from the origin and takes CA as the initial motion. The specific target motion is shown in Table 1. Clearly, all the motion models can be expressed as $m^1 = \cdots =$

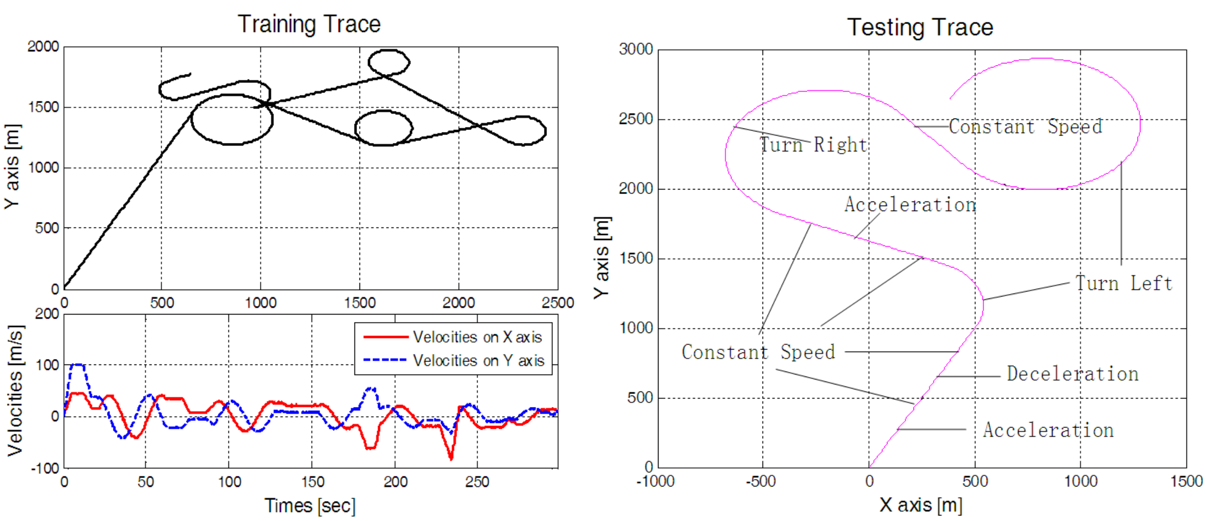

FIGURE 3. Trajectory of target motion for network training (left) and for real testing (right)

TABLE 1. Target motion types

Times (sec)		Movement	Motion models	Types
Start	End			
0.1	5.0	Acceleration	CA	2
5.1	8.0	Constant speed	CV	1
8.1	10.0	Deceleration	CA	2
10.1	15.0	Constant speed	CV	1
15.1	23.0	Turn left	CTL	4
23.1	27.0	Constant speed	CV	1
26.1	28.0	Acceleration	CA	2
28.1	33.0	Constant speed	CV	1
33.1	51.0	Turn right	CTR	3
51.1	56.0	Constant speed	CV	1
56.1	82.0	Turn left	CTL	4

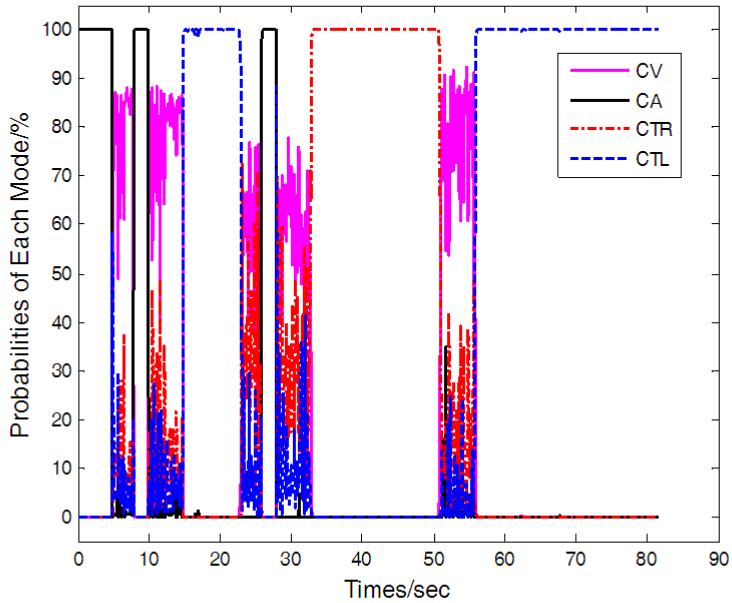

FIGURE 4. The probabilities of each candidate model

$m^{50} = \{2\}$, $m^{51} = \cdots = m^{80} = \{1\}$, $m^{81} = \cdots = m^{100} = \{2\}$, $\ldots$, $m^{561} = \cdots = m^{820} = \{4\}$, and the model sequence as $\{m^1, \ldots, m^{50}, m^{51}, \ldots, m^{80}, m^{81}, \ldots, m^{561}, \ldots, m^{820}\}$.

The target motion models $\{m^1, \ldots, m^{820}\}$ are input to the PNN in sequence and real time. After feature extraction and model decision, the PNN outputs the judgement result of what type of motion the target follows currently. The probabilities of the 4 models at each measurement time are shown in Figure 4. Obviously, the model with the largest probability means that the target follows the corresponding motion.

The PNN outputs the motion type with the largest probability as the current target motion model. The results are shown in Figure 5. Compared with the true motion types, the results of PNN decision have an accuracy of 96.57%. Due to existence of stochastic noises, some motion types may have very closed probabilities so that misjudgment appears occasionally. Nevertheless, that can be solved well if using a low pass filter to restrain glint-noise-like decision glitches. In the simulation case, a decision result is regarded as glitch if it lasts less than 5 sampling time span. That can help smooth the result and

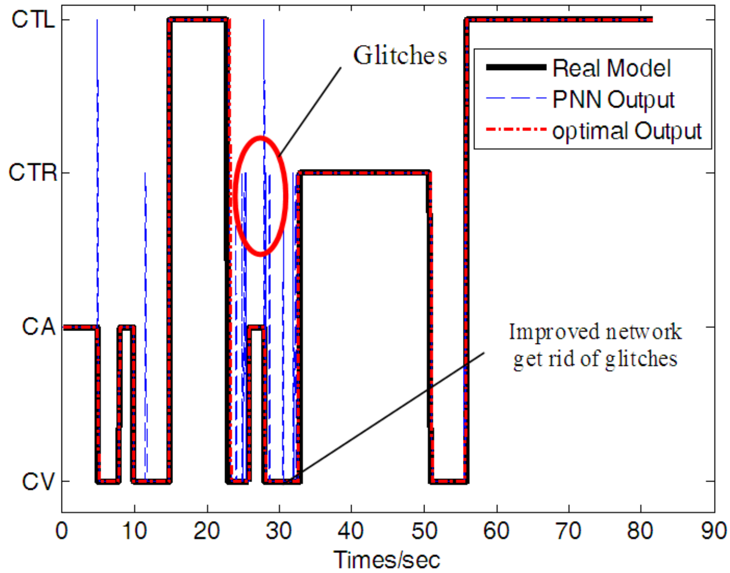

FIGURE 5. PNN-based decision results of target motion model

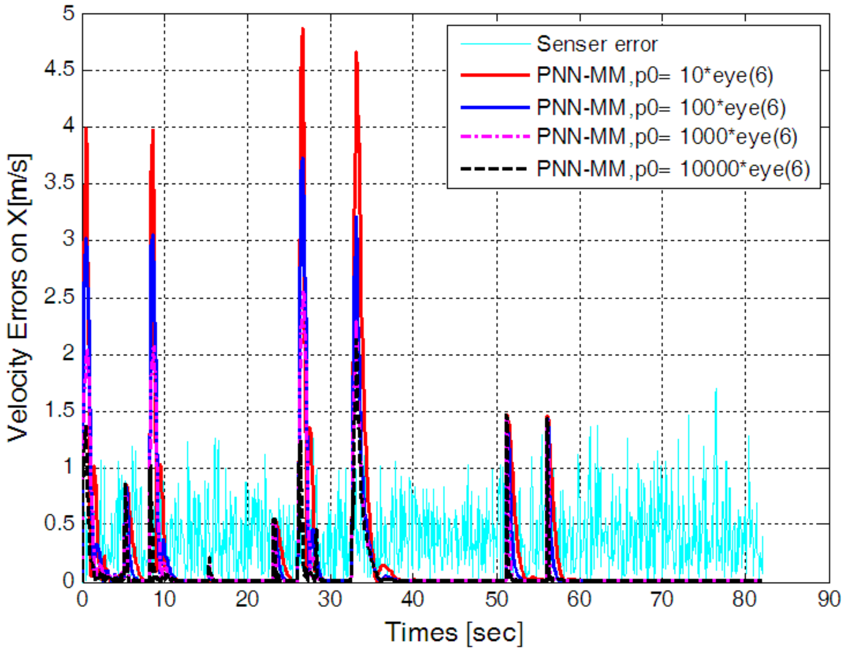

FIGURE 6. Estimation error of target velocity on x axis with variant reset values of the covariance matrix P_0

increase the decision accuracy. As shown in Figure 5, the improved PNN result has an accuracy of 99.51%, which is quite close to the ideal decision accuracy.

According to the model decision result, we use the KF with corresponding motion model to estimate the target motional states. The estimation error of the velocity on x axis is presented in Figure 6. Compared with the original sensor error we can know, filtering estimation makes the velocity error remarkably reduced. Figure 6 also provides the results under variant reset values of the covariance matrix P_0 at the initial time and each time point of model switching. Clearly, convergence speed of the estimation result is increased with the value of P_0. The state covariance matrix is required reset when target motion type changes or target maneuver appears. It is because at least two time steps are needed to generate the PNN input. That leads to a time delay of the PNN decision output and yields the fluctuation of the estimation result.

4.2. **Tracking error comparison.** We choose 4 typical tracking methods for comparison. They are the traditional KF with CA model, the standard IMM method, the PNN-MM method, and the ideal PNN-MM method where the PNN has an idea output with accuracy 100%. In the ideal PNN-MM method, no time delay appears during KF estimation.

The root-mean-square errors (RMSE) of position and velocity of the compared tracking methods are shown in Table 2. The method of KF with CA model has no decision mechanism and uses KF and CA model without filter switching. The RMSE is quite large and the estimation precision is low. In the table we can see, the IMM method can obtain better tracking performance whereas the PNN-MM method further improves the tracking error. The KF based on PNN decision greatly reduces the error covariance and achieves a higher precision. The last line of the table presents the RMSE of the PNN-MM method with decision accuracy of 100%, which actually gives the infimum of the RMSE of all the feasible PNN-MM methods.

Figure 7 compares the x-axis velocity errors of IMM method and PNN-MM method. Both methods are suitable for maneuvering target tracking. When target changes motion type, the filtering errors get sudden increments and quickly fall back to a low level. However, as the figure shows, the increment of the PNN-MM method is clearly smaller than that of the IMM method and the convergence speed of PNN-MM is also faster

TABLE 2. RMSE of the compared tracking methods

RMSE	x (m)	y (m)	v_x (m/s)	v_y (m/s)
CA	0.1001	0.1225	0.8105	1.0034
IMM	0.0908	0.0908	0.7039	0.8439
PNN-MM	0.0346	0.0552	0.2443	0.4180
Ideal PNN-MM	0.0087	0.0091	0.1168	0.1663

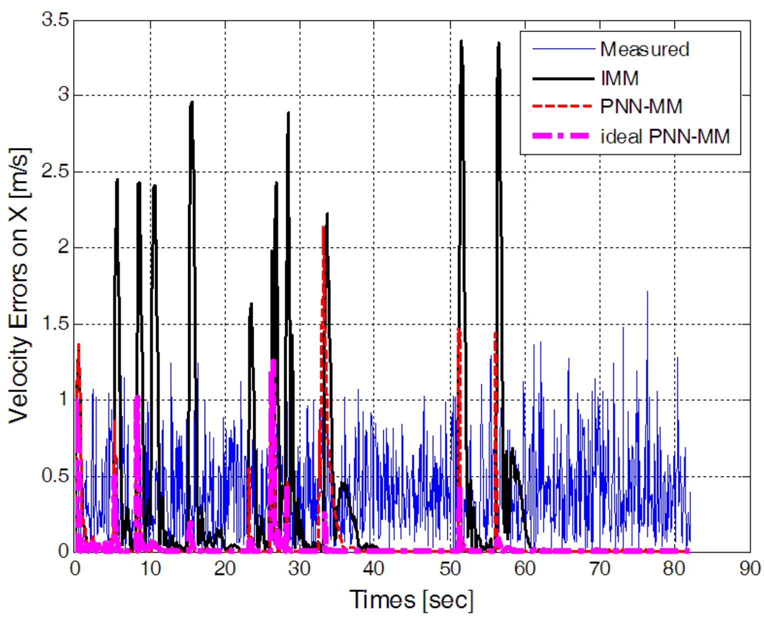

FIGURE 7. The x-axis velocity errors of the IMM, PNN-MM and the ideal PNN-MM methods

than that of IMM. The estimated velocity error of the ideal PNN-MM method is also given in Figure 7. Without negative effect of model decision misjudgment, it provides the optimal estimation result of using KF for maneuvering target tracking. Nevertheless, optimal Kalman gain cannot eliminate fast fluctuation caused by target maneuver. Using an adaptive or robust Kalman filter may further improve the estimation accuracy and convergence speed.

5. **Conclusions.** In this paper, a switched multi-model estimation method based on PNN model decision is developed for tracking a target with typical maneuver. The method contains two primary parts that are model decision and filtering estimation. For the model decision part, PNN is a feasible way to making pattern classification and decision of the target motion type, and in the filtering estimation part, the determined target model makes filter set switched to the KF with corresponding state transition matrix and generate the state estimation result. This PNN-MM method is decision-based with only one single model and filter running at any arbitrary time. That greatly reduces the computation complexity caused by paralleled-working filters in the traditional MM methods. The structure of the PNN-MM algorithm is much simpler and easier for practical application. PNN-MM estimation is an efficient and convenient method for tracking target with one primary maneuver type at each time span, showing superior estimation precision and accuracy compared with typical single model method and the IMM methods. Nevertheless, we have to notice that the maneuver capability of target is increasing fast and many mixed maneuver types appear, so the future work should focus on tracking target with complex maneuvering types, establishing decision-based tracking schemes as simple and efficient as the PNN-MM method.

Acknowledgment. This work is jointly supported by Beijing Natural Science Foundation under Grant No. 4122049, Beijing Higher Education Young Elite Teacher (No. YETP0 381), the Project Sponsored by SRF for ROCS, SEM and the Fundamental Research Funds for the Central Universities (ZYGX-2014-J098, FRF-JX-12-002, FRF-TP-12-083A and FRF-TP-14-049A2).

REFERENCES

[1] R. Murraysmith and T. A. Johansen, *Multi-Model Approaches to Nonlinear Modelling and Control*, CRC Press, Boca Raton, 1997.

[2] D. Magill, Optimal adaptive estimation of sampled stochastic processes, *IEEE Trans. Automatic Control*, vol.10, no.4, pp.434-439, 1965.

[3] H. A. P. Blom and Y. Bar-Shalom, The interacting multi-model algorithm for systems with Markovian switching coefficients, *IEEE Trans. Automatic Control*, vol.33, no.8, pp.780-783, 1988.

[4] E. Mazor, A. Averbuch and Y. Bar-Shalom, Interacting multi-model methods in target tracking: A survey, *IEEE Trans. Aerospace and Electronic Systems*, vol.34, no.1, pp.103-123, 1998.

[5] X. R. Li and P. Vesselin, A survey of maneuvering target tracking. Part V. Multiple-model methods, *IEEE Trans. Aerospace and Electronic Systems*, vol.41, no.4, pp.1255-1321, 2005.

[6] X. R. Li and P. Vesselin, A survey of maneuvering target tracking. Part I. Dynamic models, *IEEE Trans. Aerospace and Electronic Systems*, vol.39, no.4, pp.1333-1362, 2003.

[7] X. R. Li and P. Vesselin, A survey of maneuvering target tracking. Part II. Ballistic targets models, *Proc. of the SPIE Conference on Signal and Data Processing of Small Targets*, San Diego, pp.1-23, 2001.

[8] X. R. Li and P. Vesselin, A survey of maneuvering target tracking. Part IV. Decision-based methods, *Proc. of the SPIE Conference on Signal and Data Processing of Small Targets*, vol.4728, pp.511-534, 2002.

[9] X. R. Li and Y. Bar-Shalom, Multiple-model estimation with variable structure, *IEEE Trans. Automatic Control*, vol.41, no.4, pp.478-493, 1996.

[10] W. Kazimierski and A. Stateczny, Optimization of multi-model neural tracking filter for marine targets, *Proc. of the International Radar Symposium*, Warsaw, pp.543-548, 2012.

[11] W. Kazimierski, G. Zaniewicz and A. Stateczny, Verification of multi-model neural tracking filter with ship's radar, *Proc. of International Radar Symposium*, Warsaw, pp.549-553, 2012.

[12] D. F. Specht, Probabilistic neural networks for classification, mapping, or associative memory, *Proc. of the IEEE International Conference on Neural Networks*, pp.525-532, 1988.

[13] D. F. Specht, Probabilistic neural networks, *Neural Networks*, vol.3, no.1, pp.109-118, 1990.

[14] D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning representations by back-propagating errors, *Nature*, vol.323, no.10, pp.533-536, 1986.

[15] M. D. Powell, *The Theory of Radial Basis Function Approximation in 1990*, Cambridge University Press, 1992.

[16] D. S. Broomhead and D. Lowe, Radial basis functions, multi-variable functional interpolation and adaptive networks, *Advances in Neural Information Processing Systems*, vol.74, no.2, pp.728-734, 1988.

[17] X. Chen and K. Shi, Target threat assessment based on fuzzy recurrent wavelet neural network, *International Journal of Innovative Computing, Information and Control*, vol.12, no.3, pp.885-898, 2016.

[18] X. Chen, K. Shi and X. Li, Target threat assessment based on modified particle swarm optimized fuzzy recurrent wavelet neural network, *International Journal of Innovative Computing, Information and Control*, vol.13, no.2, pp.605-620, 2017.

[19] D. Evagorou, A. Kyprianou, P. L. Lewin et al., Feature extraction of partial discharge signals using the wavelet packet transform and classification with a probabilistic neural network, *IET Science, Measurement and Technology*, vol.4, no.3, pp.177-192, 2010.

[20] A. Tahir, A system based on ratio images and quick probabilistic neural network for continuous cloud classification, *IEEE Trans. Geoscience and Remote Sensing*, vol.49, no.12, pp.5008-5015, 2011.

[21] N. Perera and A. D. Rajapakse, Recognition of fault transients using a probabilistic neural-network classifier, *IEEE Trans. Power Delivery*, vol.26, no.1, pp.410-419, 2011.

[22] H. T. Lin, T. J. Liang and S. M. Chen, Estimation of battery state of health using probabilistic neural network, *IEEE Trans. Industrial Informatics*, vol.9, no.2, pp.679-685, 2013.

[23] Y. X. Sun, J. J. Chen, C. Yuen and S. Rahardja, Indoor sound source localization with probabilistic neural network, *IEEE Trans. Industrial Electronics*, 2017.

[24] K. Z. Mao, K. C. Tan and W. Ser, Probabilistic neural-network structure determination for pattern classification, *IEEE Trans. Neural Networks*, vol.11, no.4, pp.1009-1016, 2000.

[25] B. Yadav and V. S. Devi, Novelty detection applied to the classification problem using probabilistic neural network, *Proc. of the IEEE Symposium on Computational Intelligence and Data Mining*, pp.265-272, 2014.

[26] R. E. Kalman, A new approach to linear filtering and prediction problems, *Journal of Fluids Engineering*, vol.82, no.1, pp.35-45, 1960.

[27] C. H. Chen, C. M. Kuo and S. H. Hsieh, RETRACTED: Highly efficient very-large-scale integration (VLSI) implementation of probabilistic neural network image interpolator, *Journal of Vibration and Control*, vol.20, no.2, pp.218-224, 2014.

[28] M. F. Othman and M. A. M. Basri, Probabilistic neural network for brain tumor classification, *Proc. of the IEEE International Conference on Intelligent Systems, Modelling and Simulation*, pp.136-138, 2011.

[29] M. Ahmadlou and H. Adeli, Enhanced probabilistic neural network with local decision circles: A robust classifier, *Integrated Computer-Aided Engineering*, vol.17, no.3, pp.197-210, 2010.

[30] G. Minchin and A. Zaknich, A design for FPGA implementation of the probabilistic neural network, *Proc. of the IEEE International Conference on Neural Information Processing*, pp.556-559, 1999.

[31] N. Aibe, M. Yasunaga and I. Yoshihara, A probabilistic neural network hardware system using a learning-parameter parallel architecture, *Proc. of the IEEE International Joint Conference on Neural Networks*, pp.2270-2275, 2002.

[32] E. Parzen, On estimation of a probability density function and mode, *The Annals of Mathematical Statistics*, pp.1065-1076, 1962.

[33] D. F. Specht, Enhancements to probabilistic neural networks, *Proc. of the IEEE International Joint Conference on Neural Networks*, pp.761-768, 1992.