2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

A Novel Interacting Multiple Model Algorithm Based on Multi−sensor Optimal Information Fusion Rule

Xiaoyan Fu, Yingmin Jia, Junping Du and Shiying Yuan

Abstract— **In this paper, a novel interacting multiple model (IMM) algorithm is proposed, which utilizes a multi-sensor optimal information fusion rule to combine multiple models in the linear minimum variance sense instead of famous Bayes' rule. Furthermore, the diagonal matrices are used as the updated weights of models, which are applied to distinguish the effects produced by different dimensions of state, so the new algorithm is named as diagonal interacting multiple model (DIMM) algorithm. Extensive Monte Carlo simulations indicate that the proposed DIMM algorithm has better accuracy of estimation than the IMM algorithm with no increase in the execution time, which confirm that the DIMM algorithm is a competitive alternative to the classical IMM algorithm.**

I. INTRODUCTION

The Interacting Multiple Model (IMM) algorithm is a well-known state estimation algorithm for hybrid systems, whose unique feature is that the state estimates and the covariance matrices from multiple models are combined according to a Markov model for the transition between target maneuver states. It has been widely applied in many fields [1-8]. For example, in [6], the IMM method is applied to the problem of object tracking with a video system in a car. The authors in [7] develop the IMM algorithm to detect lane change maneuvers based on laser, radar, and vision data. Craig O. Sayage and Bill Moran propose waveform selection for maneuvering targets within the IMM framework [8].

However, in the classical IMM algorithm, multiple models are combined based on the analysis of 'non-pure' probability, to be accurate, the normalized product of the likelihood function of target measurement and the prior model probability mass is used as the updated weight of model. It is known that any probability mass must be in the interval [0,1], but any likelihood function has no such restriction. The two values are at different levels (magnitudes), so the resulting weight of model is only an approximate probability mass. Moreover, the scalar weight used in the IMM algorithm can not distinguish the effects of different dimensions of state, especially, the position dimensions and the velocity dimensions.

This work is supported by the NSFC (60374001,60727002,60774003), the MOE (20030006003), the COSTIND (A2120061303), and the National 973 Program (2005CB321902).

Xiaoyan Fu and Yingmin Jia are with the Seventh Research Division, Beihang University (BUAA), Beijing 100083, P.R. China fuxiaosg@ss.buaa.edu.cn, ymjia@buaa.edu.cn

Junping Du is with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, School of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China junpingdu@126.com

Shiying Yuan is with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454000, Henan, P.R. China yuansy@hpu.edu.cn

In this paper, an improved IMM algorithm reweighted by diagonal matrices is proposed, which is named as diagonal interacting multiple model (DIMM) algorithm. The novelty of DIMM algorithm can be compressed into two aspects.

First, a multi−sensor optimal information fusion rule is introduced into the DIMM algorithm for interacting models, which can obtain the optimal estimation in the linear minimum variance sense [9]. We utilize the fusion rule instead of the likelihood function of target measurement at the step of updating weight. This can avoid the mixture of likelihood function and probability mass.

Second, the weight of model is the diagonal matrix, instead of the scalar in the classical IMM algorithm, through which we can distinguish the effects produced by different dimensions of state. This point is motivated by the work in [10], which proposes the reweighted IMM (RIMM) algorithm that uses some matrices as updated weights. Nevertheless, the RIMM algorithm needs to compute a mass of inverse matrices, which not only increases the computational complexity but also constrains the choice of the initial state and the initial error covariance. Extensive Monte Carlo simulations indicate that the DIMM algorithm has better accuracy of estimation than the classical IMM algorithm with no increase in the execution time, which confirm that the DIMM algorithm is a competitive alternative to the IMM algorithm.

The remaining part of the paper is outlined as follows. The classical IMM algorithm is reviewed in section II. Section III describes the multi−sensor optimal information fusion rule. The DIMM algorithm is proposed in section IV. Section V shows the computer simulation results. The conclusions and future work are provided in section VI.

II. REVIEW OF THE IMM ALGORITHM

We consider the following jump Markov linear system:

$$\mathbf{x}(k+1) = \mathbf{A}_j\mathbf{x}(k) + \mathbf{B}_j\omega_j(k) \tag{1}$$

$$z(k) = \mathbf{C}_j\mathbf{x}(k) + \nu_j(k) \tag{2}$$

Without loss of generality, the exogenous input $\mathbf{D}(r_{k+1})u_k$ can be considered in (1). For notational convenience, we omit it here. The state vector $\mathbf{x}(k)$ is an $n-$dimension vector, whereas the observation process $z(k)$ is an $m-$dimension vector, and the subscript $j \in \mathbb{S}$ ($\mathbb{S} = \{1,2,\cdots,s\}$) denotes the model. The matrix functions $\mathbf{A}_j(\cdot)$, $\mathbf{B}_j(\cdot)$ and $\mathbf{C}_j(\cdot)$ are assumed known. The model-dependent process noise is assumed to be a Gaussian random process with the following mean and covariance:

$$E[\omega_j(k)] = 0, \qquad E[\omega_j(k)\omega_j(k)^T] = Q_j \tag{3}$$

The model-dependent measurement noise term is also assumed to be a Gaussian random process with the following mean and covariance:

$$E[\nu_j(k)] = 0, \qquad E[\nu_j(k)\nu_j(k)^T] = R_j \tag{4}$$

The model-independent system parameters for each of the models see [11,12].

Let $M_j(k)$ denotes the model j at time k. The model dynamics are modeled as a finite Markov chain with known model-transitions probabilities from model i at time k to model j at time $k+1$ [13].

$$\pi_{ij} \triangleq Prob\{M_j(k+1) \mid M_i(k)\} \tag{5}$$

$$0 \le \pi_{ij} \le 1, \qquad \sum_{j=1}^{s} \pi_{ij} = 1, \qquad i,j \in \mathbb{S} \tag{6}$$

The initial state distribution of the Markov chain is $\varphi = [\varphi_1, \cdots, \varphi_s]$, where

$$0 \le \varphi_j \le 1, \qquad \sum_{j=1}^{s} \varphi_j = 1, \qquad j \in \mathbb{S} \tag{7}$$

This Markov chain description of the models is used to model as the unknown input.

The steps of classical IMM algorithm are as follows:

1) Calculation of the mixing weight ($i,j \in \mathbb{S}$): Here, the weight is defined as a probability mass, which is that model M_i was in effect at k given that M_j is in effect at $k+1$ conditioned on the sequence of measurement $Z^k = [z_1, \cdots, z_k]$.

$$\begin{aligned} \mu_{i|j}(k|k) &\triangleq P\{M_i(k)|M_j(k+1), Z^k\} \\ &= \frac{1}{\bar{c}_j} P\{M_j(k+1)|M_i(k), Z^k\} \cdot P\{M_i(k)|Z^k\} \end{aligned} \tag{8}$$

The formula is the mixing weight, which can be written as

$$\mu_{i|j}(k|k) = \frac{1}{\bar{c}_j} \pi_{ij} \mu_i(k) \tag{9}$$

where $\bar{c}_j$ is the normalization constant.

2) Mixing ($j \in \mathbb{S}$): Starting with $\hat{\mathbf{x}}_i(k)$ and $\mathbf{P}_{ii}(k)$, the mixed initial estimation for the filter matched to $M_j(k+1)$ is

$$\hat{\mathbf{x}}^{0j}(k|k) = \sum_{i=1}^{s} \hat{\mathbf{x}}_i(k) \mu_{i|j}(k|k) \tag{10}$$

The initial covariance corresponding to the above estimation is

$$\begin{aligned} \mathbf{P}^{0j}(k|k) = & \sum_{i=1}^{s} \mu_{i|j}(k|k) \cdot \{\mathbf{P}_{ii}(k) + [\hat{\mathbf{x}}_i(k) - \hat{\mathbf{x}}^{0j}(k|k)] \\ & \times [\hat{\mathbf{x}}_i(k) - \hat{\mathbf{x}}^{0j}(k|k)]^T\} \end{aligned} \tag{11}$$

where $\hat{\mathbf{x}}_i(k)$ is the estimation of state based on the ith Kalman filter at time k, and the corresponding covariance is $\mathbf{P}_{ii}(k)$.

3) Updating the weight of model ($j \in \mathbb{S}$): The estimate (10) and covariance (11) are used as input to the filter matched to M_j, which uses $z(k+1)$ to yield $\hat{\mathbf{x}}^{0j}(k+1|k+1)$ and $\mathbf{P}^{0j}(k+1|k+1)$. The likelihood function

$$\Lambda_j(k+1) = p[z(k+1)|M_j(k+1), Z^k] \tag{12}$$

corresponding to the jth filter is computed using the mixed initial estimation and covariance as

$$\begin{aligned} &\Lambda_j(k+1) \\ &= p[z(k+1)|M_j(k+1), \hat{\mathbf{x}}^{0j}(k+1|k+1), \mathbf{P}^{0j}(k+1|k+1)] \end{aligned} \tag{13}$$

that is,

$$\begin{aligned} &\Lambda_j(k+1) \\ &= \mathcal{N}[z(k+1); \hat{\mathbf{x}}^{0j}(k+1|k+1), \mathbf{P}^{0j}(k+1|k+1)] \end{aligned} \tag{14}$$

where $\mathcal{N}(x; \bar{x}, P)$ denotes a probability density function (PDF) of x with mean $\bar{x}$ and covariance P.

Updated weight of model $M_j(k+1)$ is

$$\begin{aligned} \mu_j(k+1) &\triangleq P\{M_j(k+1)|Z^{k+1}\} \\ &= \frac{1}{c} p[z(k+1)|M_j(k+1), Z^k] P\{M_j(k+1)|Z^k\} \\ &= \frac{1}{c} \Lambda_j(k) \bar{c}_j \end{aligned} \tag{15}$$

where c is the normalization constant.

4) Estimate and covariance combination: Combination of the model-conditioned estimates is done according to the mixture equations

$$\hat{\mathbf{x}}(k+1|k+1) = \sum_{j=1}^{s} \hat{\mathbf{x}}_j(k+1|k+1) \mu_j(k+1) \tag{16}$$

Combination of the model-conditioned covariances:

$$\begin{aligned} \mathbf{P}(k+1|k+1) = & \sum_{j=1}^{s} \mu_j(k+1) \{\mathbf{P}_{jj}(k+1|k+1) \\ & + [\hat{\mathbf{x}}_j(k+1|k+1) - \hat{\mathbf{x}}(k+1|k+1)] \\ & \times [\hat{\mathbf{x}}_j(k+1|k+1) - \hat{\mathbf{x}}(k+1|k+1)]^T\} \end{aligned} \tag{17}$$

Note that the combination is only for output, it is not a part of the algorithm recursions.

III. OPTIMAL INFORMATION FUSION RULE

In this section, an optimal information fusion rule is described. The rule will be as the origin of updated weights of proposed IMM algorithm, which can obtain the optimal estimation in the linear minimum variance sense based on the following lemma.

Lemma 1:[9] If $\hat{\mathbf{x}}_i$ ($i \in \mathbb{S}$), be an unbiased estimate of the $n-$dimensional stochastic column vector x based on the ith sensor, and the estimation error is $\tilde{\mathbf{x}}_i = \hat{\mathbf{x}}_i - \mathbf{x}$, the covariance matrix of error is $\mathbf{P}_{ij} = E[\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j^T]$ ($i,j = 1, \cdots, \mathbb{S}$), $\hat{\mathbf{x}}_d$ is the fused estimation of state $\mathbf{x}$, elements of state and estimations are described as follows.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \hat{\mathbf{x}}_j = \begin{bmatrix} \hat{x}_{j1} \\ \vdots \\ \hat{x}_{jn} \end{bmatrix}, \hat{\mathbf{x}}_d = \begin{bmatrix} \hat{x}_{d1} \\ \vdots \\ \hat{x}_{dn} \end{bmatrix} \tag{18}$$

then the optimal fusion estimate is

$$\hat{\mathbf{x}}_d = \sum_{j=1}^{s} \mathbf{B}_j \hat{\mathbf{x}}_j \tag{19}$$

which minimizes the performance index J for $\mathbf{x}$

$$J = tr\mathbf{P}_d \tag{20}$$

where

$$\mathbf{B}_j = diag(b_{j1}, \cdots, b_{jn}), \quad \mathbf{P}_d = E[\tilde{\mathbf{x}}_d \tilde{\mathbf{x}}_d^T], \quad \tilde{\mathbf{x}}_d = \mathbf{x} - \hat{\mathbf{x}}_d \tag{21}$$

and it is equivalent to fused weight by diagonal matrices in the linear minimum variance sense as

$$\hat{x}_{di} = \sum_{j=1}^{s} b_{ji} \hat{x}_{ji} \tag{22}$$

where b_{ji} is derived from vector $\mathbf{b}_i$

$$\begin{aligned} \mathbf{b}_i &= [b_{1i}, b_{2i}, \cdots, b_{si}] \\ &= \frac{\mathbf{e}^T (\mathbf{P}^{ii})^{-1}}{\mathbf{e}^T (\mathbf{P}^{ii})^{-1} \mathbf{e}}, \qquad i = 1, \cdots, n \end{aligned} \tag{23}$$

we define

$$\mathbf{e} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{s\times 1}, \quad \mathbf{P}^{ii} = \begin{bmatrix} P_{11}^{(ii)} & \cdots & P_{1s}^{(ii)} \\ \vdots & \ddots & \vdots \\ P_{s1}^{(ii)} & \cdots & P_{ss}^{(ii)} \end{bmatrix} \tag{24}$$

where $P_{kj}^{(ii)}$ is the ith diagonal element of matrix $\mathbf{P}_{kj}$. The covariance of error $\tilde{x}_{di}$ is

$$\begin{aligned} P_{di} &= E[\tilde{x}_{di} \tilde{x}_{di}^T] \\ &= E[(x_i - \hat{x}_{di})(x_i - \hat{x}_{di})^T] \\ &= [\mathbf{e}^T (\mathbf{P}^{ii})^{-1} \mathbf{e}]^{-1} \end{aligned} \tag{25}$$

Through Lemma 1, we can obtain the optimal fused estimation and the weight of estimation from each sensor. In face, there are many information fusion rules can be used in the IMM algorithm, but only the information fusion rule (lemma 1) is weighted by diagonal matrices, which can distinguish the effects produced by different dimensions of state with no increase in the execution time.

IV. DIMM ALGORITHM

As can be seen from step 3 (updating the weight of model) in IMM algorithm, the likelihood function of target measurement model is used to obtain the updated model weight $\mu_j(k)$, which is not a real probability mass but a *PDF*. Because any probability mass must be a scalar in the interval [0,1], but any *PDF* has no such restriction. Moreover, used *PDF*s come from different distributions, which are at different levels (magnitudes), therefore the resulting weight of model is only an approximate probability mass. In this case, we aim at proposing a new IMM algorithm to avoid the mixture of *PDF* and probability mass, and obtaining the optimal estimation of state simultaneously.

In this section, the new IMM algorithm—DIMM algorithm is proposed in detail. We regard the combination of multiple models as the fusion of multiple sensors, so the updated weights of models can be obtained from the fuse rule in the previous section. The steps of DIMM algorithm are as follows:

1) Calculation of the mixing diagonal weight ($i, j \in \mathbb{S}$): The probability that model M_i was in effect at k given that M_j is in effect at $k+1$ conditioned on the sequence of measurement $Z^k = [z_1, \cdots, z_k]$ about each dimension of state is one of elements of following diagonal matrix (26), so the weight of model can be viewed as

$$\begin{aligned} \mathbf{B}_{i|j}(k|k) &\triangleq P\{M_i(k)|M_j(k+1), Z^k\} \\ &= \frac{\pi_{ij} \cdot \mathbf{B}_i(k)}{\sum_{i=1}^{s} \pi_{ij} \cdot \mathbf{B}_i(k)} \\ &= \begin{pmatrix} \frac{\pi_{ij} b_{i1}}{\sum_{i=1}^{s} \pi_{ij} b_{i1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\pi_{ij} b_{in}}{\sum_{i=1}^{s} \pi_{ij} b_{in}} \end{pmatrix} \end{aligned} \tag{26}$$

where

$$\begin{aligned} \mathbf{B}_i(k) &= diag(b_{i1}, b_{i2}, \cdots, b_{in}) \\ &\triangleq P\{M_i(k)|Z^k\} \end{aligned} \tag{27}$$

2) Mixing ($j \in \mathbb{S}$): Starting with $\hat{\mathbf{x}}_i(k)$ and $\mathbf{P}_{ii}(k)$, the mixed initial estimation for the filter matched to $M_j(k+1)$ is

$$\hat{\mathbf{x}}^{0j}(k|k) = \sum_{i=1}^{s} \mathbf{B}_{i|j}(k|k) \cdot \hat{\mathbf{x}}_i(k) \tag{28}$$

The initial covariance corresponding to the above estimation is

$$\begin{aligned} \mathbf{P}^{0j}(k|k) = & \sum_{i=1}^{s} \mathbf{B}_{i|j}(k|k) \cdot \{\mathbf{P}_{ii}(k) + [\hat{\mathbf{x}}_i(k) - \hat{\mathbf{x}}^{0j}(k|k)] \\ & \times [\hat{\mathbf{x}}_i(k) - \hat{\mathbf{x}}^{0j}(k|k)]^T\} \end{aligned} \tag{29}$$

3) updating model weight ($j \in \mathbb{S}$): The weight updating method is based on the optimal information fuse rule (in Section III) which is weighted by diagonal matrices.

3.1) Bank of kalman filters produce outputs using the mixed initial estimation $\hat{\mathbf{x}}^{0j}(k|k)$ and covariance $\mathbf{P}^{0j}(k|k)$ are:

$$\hat{\mathbf{x}}_j(k+1), \qquad \mathbf{P}_{jj}(k+1) \tag{30}$$

3.2) updated model weight is:

$$\mathbf{B}_j(k+1) = diag(b_{j1}, b_{j2}, \cdots, b_{jn}) \tag{31}$$

where

$$\begin{aligned} \mathbf{b}_j &= [b_{1j}, b_{2j}, \cdots, b_{sj}] \\ &= [\mathbf{e}^T (\Sigma^{jj})^{-1} \mathbf{e}]^{-1} \mathbf{e}^T (\Sigma^{jj})^{-1} \end{aligned} \tag{32}$$

$$\mathbf{e} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{s\times 1}, \quad \Sigma^{jj} = \begin{bmatrix} \mathbf{P}_{11}^{(jj)}(k+1) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{P}_{ss}^{(jj)}(k+1) \end{bmatrix} \tag{33}$$

$\mathbf{P}_{ii}^{(jj)}(k+1)$ is the jth diagonal element of matrix $\mathbf{P}_{ii}(k+1)$ that comes from (30), and $\mathbf{0}$ denotes the null matrix.

4) Estimate and covariance combination: Combination of the model-conditioned estimates is done according to the mixture equations

$$\hat{\mathbf{x}}_D(k+1) = \sum_{j=1}^{s} \mathbf{B}_j(k+1) \cdot \hat{\mathbf{x}}_j(k+1) \tag{34}$$

Combination of the model-conditioned covariances:

$$\begin{aligned} \mathbf{P}_D(k+1) = \quad & \sum_{j=1}^{s} \mathbf{B}_j(k+1) \cdot \{\mathbf{P}_{jj}(k+1) + [\hat{\mathbf{x}}_j(k+1) \\ & -\hat{\mathbf{x}}_D(k+1)] \times [\hat{\mathbf{x}}_j(k+1) - \hat{\mathbf{x}}_D(k+1|k+1)]^T\} \end{aligned} \tag{35}$$

Remark: we regard the diagonal matrix as the probability in the DIMM algorithm. Though the diagonal matrix is not a value in the interval [0,1], the every element of diagonal matrix must be a scalar in the interval [0,1]. In fact, the diagonal matrices used in the DIMM algorithm are the joint probability, the every element denotes a probability mass.

Assume that $\mathbf{x}$ is the n dimensional stochastic vector, and $\mathbf{y}$ is the m dimensional measurement vector. We denote the linear space spanned by $\mathbf{y}$ as $L(\mathbf{y})$. The optimal estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ based on $L(\mathbf{y})$ in the linear minimum variance sense is defined by minimizing

$$J = trE[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T], \quad \hat{\mathbf{x}} \in L(\mathbf{y}) \tag{36}$$

Let $\hat{\mathbf{x}}_D$ and $\hat{\mathbf{x}}_S$ are estimates of $\mathbf{x}$ based in linear spaces $L_D(\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_s)$ and $L_S(\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_s)$, respectively, where $L_\theta(\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_s)(\theta = D, S)$ are the set $\{\Omega_1\hat{\mathbf{x}}_1 + \cdots + \Omega_s\hat{\mathbf{x}}_s\}$, Ω_i denotes diagonal matrix and scalar, respectively. Denote the error covariances of estimates $\hat{\mathbf{x}}_D$ and $\hat{\mathbf{x}}_S$ as $\mathbf{P}_D$ and $\mathbf{P}_S$, respectively.

Theorem 1: Under the linear minimum variance sense, the optimal estimates $\hat{\mathbf{x}}_D$ and $\hat{\mathbf{x}}_S$ weighted by diagonal matrices and probability masses respectively, satisfy the accuracy relation

$$tr\mathbf{P}_D \leq tr\mathbf{P}_S \tag{37}$$

Proof : Notice that a probability mass is a scalar, a scalar weight is a special case of diagonal matric weight, and $\hat{\mathbf{x}}_i \in L(\mathbf{y}_i)$, so we have the relation

$$L_S(\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_s) \subseteq L_D(\hat{\mathbf{x}}_1, \cdots, \hat{\mathbf{x}}_s) \tag{38}$$

From (38), we obtain (37). This completes the proof.

In the section, we present the DIMM algorithm based on a multi−sensor optimal information fusion rule in the linear minimum variance sense. There is no need to calculate the likelihood function of target measurement, so the DIMM algorithm keeps away the mixture of probability mass and likelihood function in the classical IMM algorithm. Moreover, the diagonal matrices are used as the updated weights of models in the DIMM algorithm, which distinguish the effects of different dimensions of state, such as position dimensions and velocity dimensions. Because the updated weight matrices are derived from the optimal information fusion rule in the sense of linear minimum variance, the estimation of state is also optimal in the linear minimum variance sense in the DIMM algorithm.

V. SIMULATION RESULTS

In this section, we evaluate the performance of the DIMM and the classical IMM algorithms by using the target tracking example in [14]. Assume a target moves along a plane with constant course and speed until $t = 400s$. Then it starts to maneuver a slow 90 degree turn in the x direction with acceleration inputs $u_x = u_y = 0.075m/s^2$ and completes the turn at $t = 600s$. From then on the accelerations are zero. The second turn, also 90 degree, is fast: it starts at $t = 610s$ with acceleration inputs $u_x = -0.3m/s^2$ and $u_y = 0.3m/s^2$ and is completed at $t = 660s$. The $x-y$ position trajectory of the target is shown in Fig. 1. In addition, the velocity trajectory of the target is depicted in Fig. 2. For target tracking, two constant velocity (CV) models with different process noises are employed. The state representations for each CV model are as follows:

$$\mathbf{A}_i = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}; \tag{39}$$

$$\mathbf{B}_i = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{40}$$

The covariances are

$$Q_i = \begin{bmatrix} q_i & 0 \\ 0 & q_i \end{bmatrix}; \quad R_i = \begin{bmatrix} r & r/20 \\ r/20 & r \end{bmatrix} \tag{41}$$

where $q_1 = 0.01$; $q_2 = 50$; whereas r is a parameter in the simulation. The initial state for the target is $\mathbf{x}(0) = [2100; 0; 10000; -15]$. The initial error covariance is chosen based on the result in [14]

$$\mathbf{P}_i(0) = \begin{bmatrix} r & r/T & 0 & 0 \\ r/T & 2 & 0 & 0 \\ 0 & 0 & r & r/T \\ 0 & 0 & r/T & 2 \end{bmatrix} \tag{42}$$

the target's position is sampled every T=10s.

The two-model algorithms have mode transition matrix

$$[\pi_{ij}] = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \tag{43}$$

The initial distribution is $\varphi_1 = 0.5$, $\varphi_2 = 0.5$.

The results are obtained from 100 Monte carlo runs. Fig. 3 and Fig. 5 show that the root mean square (RMS) error of DIMM algorithm is virtually indistinguishable from that of the IMM algorithm' in the dimensions of position ($x-position$ and $y-position$), when the measurement noise is low (r=1). Fig. 4 and Fig. 6 show that the performance of DIMM algorithm is better than the IMM algorithm' in the dimensions of $x-velocity$ and $y-velocity$. As a whole, the accuracy of estimation of the DIMM algorithm is higher than the IMM algorithm's. The average errors of 100 sample times in Table 1 can validate it, and computational complexity of the algorithms are also shown from Run-time statistics in Table 1. The algorithm was implemented in MATLAB 7.0.4

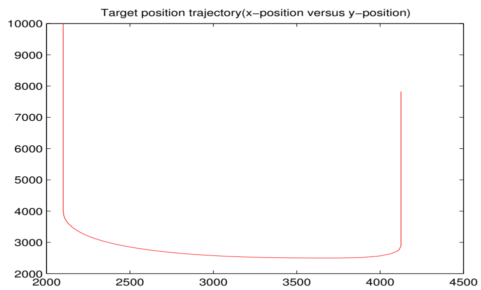

Fig. 1. Target position trajectory

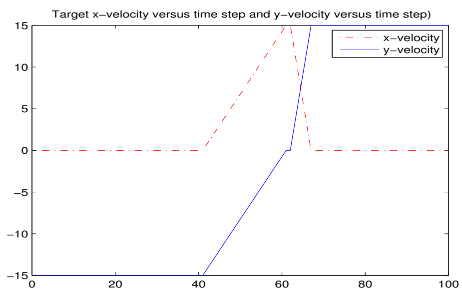

Fig. 2. Target velocity versus time step

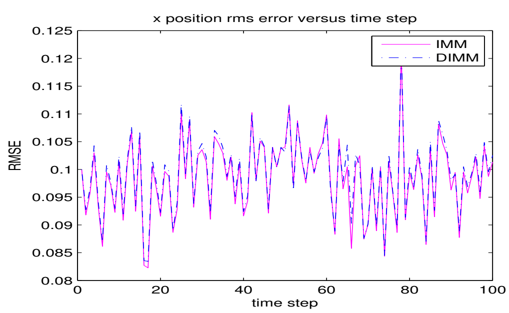

Fig. 3. $x-position$ RMS Error (r=1)

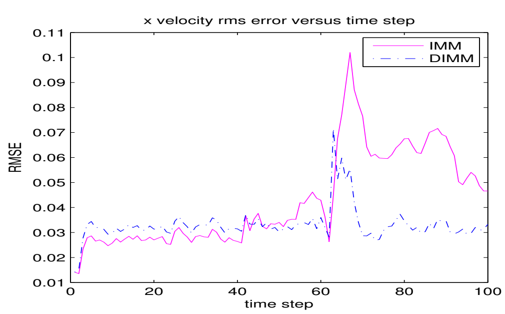

Fig. 4. $x-velocity$ RMS Error (r=1)

on a 2.79 GHz 4 CPU Pentium-based computer operating under Windows XP (Professional). The Run-time in table 1 is the running time for 100 times' simulation, each time 100 steps.

When the measurement noise is high (r=1000), Fig. 7-Fig. 10 show that the proposed DIMM algorithm is obvious superior to the IMM algorithm in almost all of the time.

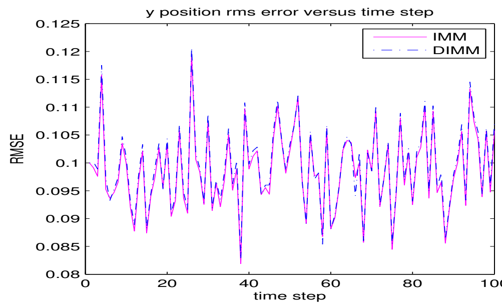

Fig. 5. $y-position$ RMS Error (r=1)

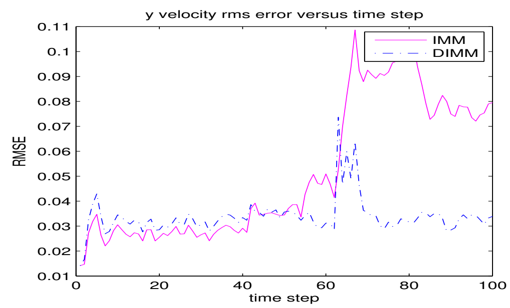

Fig. 6. $y-velocity$ RMS Error (r=1)

The IMM algorithm is better than the DIMM algorithm in the dimensions of $x-position$ and $y-position$ during the second turn (samples time 62-66), which may be due to that the DIMM algorithm pays more attention to the rapid change of the velocity in the high maneuver.

Comparing the IMM algorithm with the DIMM algorithm, we can find that the Run-time of DIMM algorithm is evident less than the IMM algorithm's. Moreover, the accuracy of the DIMM algorithm is better than the IMM algorithm, especially about the estimations of velocity dimensions ($x-velocity$ and $y-velocity$) or in the situation with high measurement noise. Therefore, the proposed DIMM algorithms is more flexible and applicable than the classical IMM algorithm.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In the paper, the DIMM algorithm is presented, which is schematically similar to the classical IMM algorithm. The

TABLE I

RMS ERROR VALUES IN EACH DIMENSION AND RUN-TIME STATISTICS

	IMM (r=1)	DIMM (r=1)	IMM (r=1000)	DIMM (r=1000)
x position	0.0993	0.1001	2.8386	2.5834
x velocity	0.0527	0.0331	1.0077	0.1458
y position	0.0977	0.0983	2.7986	2.5657
y velocity	0.0514	0.0328	1.0142	0.1489
Run-time	7.64712	4.4526	8.1469	4.6595

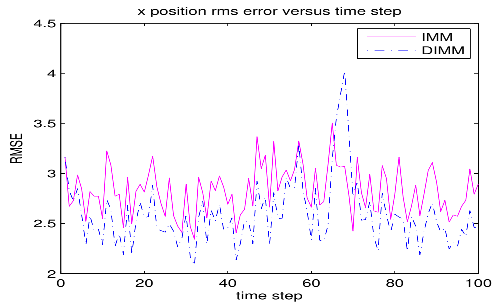

Fig. 7. $x-position$ RMS Error (r=1000)

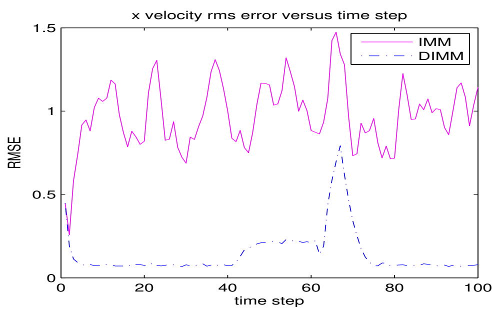

Fig. 8. $x-velocity$ RMS Error (r=1000)

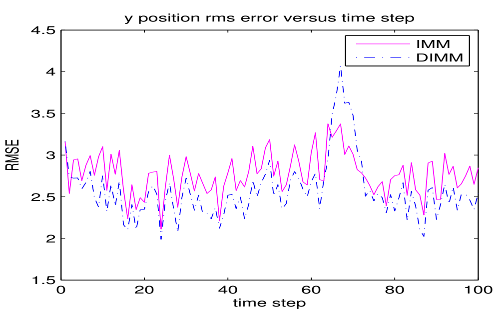

Fig. 9. $y-position$ RMS Error (r=1000)

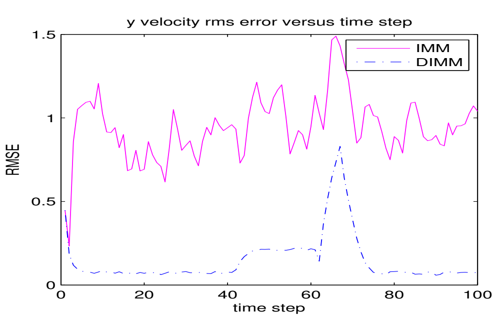

Fig. 10. $y-velocity$ RMS Error (r=1000)

difference lies in the method of updating model weight. The DIMM algorithm is reweighted by diagonal matrices based on a multi−sensor optimal information fusion rule in the linear minimum variance sense instead of Bayes' rule. By the combination of the filter input and output based on this optimal fusion rule, the DIMM algorithm obtains the optimal state estimation in the linear minimum variance sense. Simultaneously, the algorithm can avoid the hybrid computation of likelihood function and probability mass, and distinguish the effects produced by different dimensions of state. Computer simulations indicate that the DIMM algorithm gained better track accuracy than the popular IMM algorithm. Furthermore, the DIMM algorithm is much more flexible and has less computational complexity than the IMM algorithmm.

B. Future Works

In future works, the classical IMM algorithm will be further improved based on information fusion theory. We will propose various IMM algorithms that are reweighted by scalars, diagonal matrices, and general matrices based on various information fusion rules respectively, and analyze the stability of the various IMM algorithms [15].

REFERENCES

[1] X.R.Li and V.Jilkov, Estimation and tracking: principles, techniques, and software, *IEEE Trans. Aerosp. Electron. Syst.*, vol.39, no.4, Oct.2003, pp 1333-1364.

[2] S.Puranik and J.K.Tugnait, Tracking of multiple maneuvering targets using multiscan JPDA and IMM filtering, *IEEE Trans. Aerosp. Electron. Syst.* , vol.43, no.1, Jan.2007, pp.23-35.

[3] A.Sinha,T.Kirubarajin and Y.Bar-Shalom, Application of the Kalman-Levy filter for tracking maneuvering targets, *IEEE Trans. Aerosp. Electron. Syst.*, vol.43, no.3, Jul.2007, pp.1099-1107.

[4] R.Toledo-Moreo, M.A.Zamora-Izquierdo and A.F.Gomez-Skarmeta, , IMM-EKF based Road Vehicle Navigation with Low Cost GPS/INS *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, Sep.2006, pp.433 - 438

[5] J.A.Besada, J.Garcia and G.De Miguel, Design of IMM filter for radar tracking using evolution strategies, *IEEE Trans. Aerosp. Electron. Syst.*, vol.41, no.3, Jul.2005, pp.1109-1122.

[6] N. Kaempchen, K. Weiss, M.Schaefer, and K. C. J. Dietmayer, IMM object tracking for high dynamic driving maneuvers, *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, Jun.2004, pp.825C830.

[7] K. Weiss, N.Kaempchen and A. Kirchner, Multiple-model tracking for the detection of lane change maneuvers, *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, Jun.2004, pp.937C942.

[8] Craig O. Sayage and Bill Moran, Waveform selection for maneuvering targets within an IMM framework, *IEEE Trans. Aerosp. Electron. Syst* ., vol.43, no.3, Jul.2007, pp.1205-1214.

[9] S.L.Sun and Z.L.Deng, Multi-sensor optimal information fusion Kalman filter, *Automatica* , vol.40, 2004, pp.1017-1023.

[10] A.L.Johnston and V.Krishnamurthy, An Improvement to the interacting multiple model (IMM) Algorithm, *IEEE Trans. Sign. Proc.*, vol.49, Dec.2001, pp 2909-2923.

[11] Javier Lovera Yepes, Inseok Hwang and Mario Rotea, New algorithms for aircraft intent inference and trajectory prediction, *AIAA Journal of Guidance, Control, and Dynamics*, vol.30, no.2, March-April 2007, pp.370-382.

[12] X.R.Li and Vesselin P. Jilkov, Survey of maneuvering target tracking. Part I: dynamic models, *IEEE Trans. Aerosp. Electron. Syst.*, vol.39, no.4, Oct.2003, pp.1333-1364.

[13] I.Hwang,H.Balakrishnan and Tomlin, State estimation for hybrid systems: applications to aircraft tracking, *IEE Proceedings Control Theory and Application*, 2006.

[14] T.J.Ho and M.Farooq, Comparing an interacting multiple model algorithm and a multiple-process soft switching algorithm: equivalence relationship and tracking performance *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, vol.1, Jul.2000, pp.MOD2/17-MOD2/24.

[15] Chze Eng Seah and Inseok Hwang, Stability analysis of the Interacting Multiple Model algorithm, *American Control Conference, 2008*, Jun.2008, pp.2415-2420.