Radar Target Tracking Using
An IMM-EV Estimators-Based Switching Scheme
T.-J. Ho
Department of Electrical Engineering
Chung Yuan Christian University
Chungli, Taiwan, R.O.C.
E-mail: tjho@dec.ee.cycu.edu.tw

Abstract – In this paper, we propose a method of combining some interacting multiple model-extended Viterbi (IMM-EV) algorithms for target tracking. The objective of the proposed scheme is to take the maximum advantage of the combined strengths of some IMM-EV algorithms so as to achieve better performance and/or computational efficiency than the IMM and some tracking algorithms. Simulation results demonstrate the aforementioned goal can be achieved.

I. INTRODUCTION
The interacting multiple-model (IMM) algorithm in [1] has been a popular method for state estimation of discrete-time Markov jump linear systems (MJLS). Recently, various fixed structure MM (FSMM) and variable structure MM (VSMM) algorithms have been proposed as improvements to the IMM algorithm. The FSMM approach, e.g., [2]-[4], utilizes the total set of models throughout MM estimation. In contrast, the VSMM approach, e.g., [5], divides the total set of models into a number of model groups for state estimation, each of which represents a cluster of closely related system behavior pattern. In this paper, we will introduce a hybrid approach for developing a new alternative to the IMM algorithm, which integrates the concept of variable structure into a family of IMM-EV algorithms in [3]-[4] for combining some strengths of FSMM and VSMM approaches. The hybrid approach is proposed specifically based on switching among some IMM-EV(m) algorithms.

It has been shown in [3]-[4] that given positive integers m and n with \( m < n \), the IMM-EV(m) algorithms can handle properly the degradation of the estimation performance and/or the increase of computational burden due to the excessive competition from some unnecessary models of the n models. The IMM-EV algorithms span the continuum between the hard-decision methods of merged-hypothesis-tree style tracking algorithms and the IMM algorithm. The IMM-EV(m) algorithm computes and selects the ‘m’ most probable model paths at any given time instant based on the extended Viterbi algorithm, and merges the information from the ‘m’ most likely model paths into state estimation. The IMM-EV(m=1) algorithm is the most computationally efficient estimator in the family of IMM-EV(m) algorithms. However, it generally yields estimation errors larger than IMM-EV(m) algorithms with some small integers \( m > 1 \), during the occurrence of a maneuver. To take maximum advantage of some strengths of IMM-EV(m) algorithms, it is more desirable that the IMM-EV(m=1) algorithm is utilized for state estimation when the system resides in or reverts to a quiescent mode whereas the IMM-EV(m) algorithms with some small integers \( m > 1 \) are employed when a maneuver is declared. Accordingly, switching among the IMM-EV(m) algorithms is performed whenever a maneuver is detected or terminated. In this manner, the proposed approach has a high potential for achieving a satisfactory tradeoff between estimation performance and efficient computation in comparison with the IMM algorithm.

This paper is organized as follows: Section II describes a Markov-transitioned multiple-model framework. Section III reviews the IMM-EV(m) algorithms. In Section IV, we introduce the proposed algorithm. Section V demonstrates the feasibility of the proposed approach in comparison with the IMM algorithm and the Viterbi tracking method. We conclude this work in section VI.

II. SYSTEM MODEL
We consider discrete-time MJLS. Assume the overall system dynamics can be covered by \( n \) discrete-time linear models \( M_1 \ldots M_n \). We denoted \( M_k \) as the \( j \) th discrete-time model in effect at the \( k \) th sampling period with \( j = 1, \ldots, n \) and \( k = 0,1,2,\ldots \). Moreover, each model \( M_k \) is described by the state equation

\[
x_{k+1}^j = A_{k}^j x_{k}^j + B_{k}^j u_{k}^j + \Gamma_{k}^j v_{k}^j
\]

and the measurement equation

This work is supported in part by National Science Council of Taiwan under Grant NSC 95-2221-E-033-041.
\[ z^j_k = H^j_k x^j_k + D^j_k \omega^j_k \]  

Assume the system matrices \( A^j_k, B^j_k, \Gamma^j_k, H^j_k, D^j_k \) are known. \( x^j_k \in \mathbb{R}^{n_x} \) is a state of the system. The \( B^j_k \) are exogenous inputs. \( z^j_k \in \mathbb{R}^{n_z} \) is a vector-valued measurement. The \( \Gamma^j_k \) and the \( D^j_k \) denote process noises and measurement noises, respectively, where the \( \Gamma^j_k \) and the \( D^j_k \) are independent vector zero-mean white Gaussian processes. The covariance matrices of process noises are \( Q^j_k = E[\Gamma^j_k \Gamma^j_k^T] \) which are positive semi-definite, and the covariance matrices of measurement noises are \( R^j_k = E[D^j_k \omega^j_k D^j_k \omega^j_k^T] \) which are positive definite where the notation \( E[.] \) denotes the expectation. The initial state \( x_0 \) is assumed to be independent of \( \Gamma^j_k \) and \( D^j_k \).

III. IMM-EV(m) ESTIMATORS

The IMM-EV(m) algorithms developed in [3]-[4] are recursive estimators. The derivations of the IMM-EV(m) algorithms are based on the incorporation of the mechanism of the extended Viterbi algorithm(m) introduced in [3] into hypothesis reduction of the IMM algorithm. Given positive integers \( m \) and \( n \) with \( m < n \), the IMM-EV(m) algorithm seeks the \( m \) most probable model paths from \( n \) model paths at any given time instant based on the extended Viterbi algorithm(m). Then it merges the information from the \( m \) most likely model paths into state estimation. The IMM-EV(m) algorithms are summarized in Table I and described below. Given \( n \) and \( m \) with \( 1 \leq m \leq n \).

**Step 1.** Mixing probability calculation: the process involves in determining the \( m \) largest elements of \( p_{ij} \mu_{k-1}(i) \) for \( i = 1, \ldots, n \) and their indices corresponding to \( i \) for a fixed index \( j \), \( j = 1, \ldots, n \). For indices \( i \) corresponding to the \( m \) largest elements of \( p_{ij} \mu_{k-1}(i) \), mixing probabilities \( \mu_{k-1}(i | j) \) are computed by only taking those \( m \) largest elements into account. **Step 2.** Interaction/mixing of the estimates: the mixed state estimate \( \hat{x}^j_{k-1} \) are the inputs to the filtering at the **Step 3** and are computed.

<table>
<thead>
<tr>
<th>Table I.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialization</strong></td>
</tr>
<tr>
<td>Given ( m, n ) with ( 1 \leq m \leq n ). For ( j = 1, \ldots, n ), given</td>
</tr>
<tr>
<td>( 0 \leq \eta_j \leq 1 ), ( \sum_{j=1}^{n} \eta_j = 1 ); and given ( \tilde{x}^0_0 ), ( P^0_0 ).</td>
</tr>
</tbody>
</table>

**Step 1.**

For \( j = 1, \ldots, n \), \( s = 1, \ldots, m \),

\[
\mu_{k-1}(s | j) = \frac{\max_{s=1}^{m}(s) \{ p_{ij} \mu_{k-1}(i) \} \sum_{s=1}^{m} \max_{s=1}^{m}(s) \{ p_{ij} \mu_{k-1}(i) \}}{\sum_{s=1}^{m} \max_{s=1}^{m}(s) \{ p_{ij} \mu_{k-1}(i) \}}
\]

\[
\tilde{x}_{k-1} = \arg\{ \max_{s=1}^{m}(s) \{ p_{ij} \mu_{k-1}(i) \} \}.
\]

**Step 2.**

For \( j = 1, \ldots, n \),

\[
x^0_{k-1} = \sum_{s=1}^{m} \tilde{x}_{k-1}^{s} \mu_{k-1}(s | j)
\]

\[
P^0_{k-1} = \sum_{s=1}^{m} \mu_{k-1}(s | j) [P_{k-1}^{s} - \sum_{j=1}^{n} \tilde{x}_{k-1}^{s} (\tilde{x}_{k-1}^{s} - \sum_{s=1}^{m} \tilde{x}_{k-1}^{s} - \sum_{s=1}^{m} \tilde{x}_{k-1}^{s})].
\]

**Step 3**

For \( j = 1, \ldots, n \), the state estimate \( \tilde{x}_{k-1}^j \) and state error covariance \( P_{k-1}^{j} \) are utilized for computing predicted state and predicted state covariance \( P_{k-1}^{j} \) of the Kalman filter matched to the model \( M^j_k \) given by (3)-(4).

Then \( \tilde{x}_{k-1}^j \) and \( P_{k-1}^{j} \) are utilized for computing updated state estimate \( \tilde{x}_{k}^j \) and updated state covariance \( P_{k}^{j} \). The corresponding likelihood function is \( L_k(j) = \mathcal{N}[v^j_k, S^j_k] \) where \( v^j_k = z_k - H^j_k (\tilde{x}_{k-1}^j) \) is innovation with zero mean and covariance \( S^j_k \).

**Step 4.**

For \( j = 1, \ldots, n \),

\[
\mu_k(j) = \frac{\max_{j=1}^{n} \{ \tilde{x}_{k}^j \mu_{k-1}(j) \}}{\sum_{j=1}^{n} \max_{j=1}^{n} \{ \tilde{x}_{k}^j \mu_{k-1}(j) \}}
\]

**Step 5.**

For \( r = 1, \ldots, m \),

\[
\tilde{\mu}_r = \arg\{ \max_{1 \leq j \leq n} (r) \{ \mu_k(j) \} \}
\]

**Step 6.**

\[
\tilde{x}_k = \sum_{r=1}^{m} \tilde{\mu}_r \tilde{x}_k^r.
\]
models given by (1)-(2). For and their corresponding indices. For are based on model, the corresponding to the 'm' largest elements of \( jk \). Determining the 'm' largest elements of the weighted sum of updated state estimates constructed based on the obtained by renormalizing the 'm' largest elements of indices \( jk \). In \( jk \) dynamics fitting the current channel dynamics are calculate. In \( jk \) likelihoods

EV(m) algorithms with some integers \( m \), \( n \) \( m \) efficient computation but switch to employ some IMM-algorithm be utilized for most of the time to achieve algorithms, it is suggested that the IMM-EV(m=1) algorithm. To detect the onset of a maneuvering motion, we adapt the fading-memory detection strategy as follows. The fading-memory average of the innovations herein is based on the innovations of the most likely model path from the Step 3 of the IMM-EV(m=1) algorithm. Suppose a maneuver is significant enough to be detected at time \( k \). In general, there is a delay between the time of an acceleration change and the time that the trajectory deviation is significant enough to determine that change unambiguously. A switch from Set 1 to Set 2 takes place and the maneuver is estimated to occur at time \( k - \Delta \). However, appropriate state estimates and errors caused by Set 1 during the occurrence of a maneuvering motion is tested. Then Set 2 is utilized for computing the state estimates from time \( k - \Delta \) up to time \( k \) so that the peak estimation errors caused by Set 1 during the occurrence of a maneuver can be reduced after the data processed by the Set 2. However, appropriate state estimates and model probabilities that should be utilized by the Set 2 at time \( k - \Delta \) are not known a priori. Thus, we activate and initialize the Set 2 using the data from some time prior to time \( k - \Delta \) obtained by the Set 1. And from then on up to time \( k \) we utilize the Set 2. The computed data at time \( k \) obtained by the Set 2 will be used for initializing the algorithm in the Set 3 at time \( k + 1 \). The Set 3 continues in real-time state estimation until the termination of the maneuver is declared. Assume the system dynamics is described in the x-y plane. Of course, the description can be extended to 3-dimensions. Our strategy for detecting the termination

![Fig 1. Schematic view of the proposed approach](image)
of the maneuvering motion is based on whether each acceleration component of the system simultaneously falls below its corresponding threshold. The termination of a maneuvering motion is declared at time \( \tilde{k} \) if \( \tilde{k} > k \) and \( \tilde{k} \) is the first time instant since time \( k \) such that the estimated acceleration components in both coordinates, i.e., \( \hat{a}_x(\tilde{k}) \) and \( \hat{a}_y(\tilde{k}) \), are insignificant (i.e., both \( |\hat{a}_x(\tilde{k})| \) and \( |\hat{a}_y(\tilde{k})| \) drop below its respective threshold, say, below 2% of the absolute value of its corresponding estimated acceleration component at time \( k \)). Of course, other choices of threshold settings are possible.

Accordingly, the propose algorithm is summarized in Table II.

**Remark:** The quantities \( Thl, \alpha, \text{guard}, \text{Th}_-.a_x, \) and \( \text{Th}_-.a_y \) are parameters whose values determine whether a switch between algorithms occurs. The quantity \( \frac{1}{1-\alpha} \) with \( 0 < \alpha < 1 \) may be regarded as the effective window length over which the presence of a motion change is tested. Under the Gaussian assumptions, \( e_i(k) \) is chi-square distributed with \( n_z \) (dimension of the measurement) degree-of-freedom which in turn yields \( \rho(k) \) approximately \( \frac{n_z(1+\alpha)}{1-\alpha} \) degree-of-freedom chi-square distributed. Thus, as long as the parameters \( \alpha \) and \( n_z \) are given, we can set the threshold \( Thl \), say, the 95% confidence region for \( \rho(k) \). \( \text{Th}_-.a_x \) and \( \text{Th}_-.a_y \) are chosen as (15). It is to be noted that there are some chances that a maneuver may not be detected.

### V. NUMERICAL EXAMPLES

In this section, we use two examples in [7],[2] to demonstrate the feasibility of the proposed approach. The performance comparisons for each example are made based on 100 Monte Carlo runs.

**Example 1:** The state of a target \( x_k \) is of the form

\[
x_k = [x(k) \, \dot{x}(k) \, \ddot{x}(k) \, y(k) \, \dot{y}(k) \, \ddot{y}(k)]^T
\]

with position in meters, velocity in m/sec and acceleration in m/sec\(^2\) in the \((x,y)\) plane. The state evolves in accordance with eqn. (1) with the initial state \( x_0 = [2000 \\ 0 \\ 10000 \\ -15 \\ 0] \). The position of moves with constant velocity during the 0-39th sampling periods, undergoes the slow maneuvering motion with \( x \)-acceleration 0.075m/sec\(^2\) and \( y \)-acceleration 0.075m/sec\(^2\) during the 40th-60th sampling periods and the fast one with \( x \)-acceleration 0.3m/sec\(^2\) and \( y \)-acceleration -0.3m/sec\(^2\) during the 61th-66th sampling periods, and thereafter moves with constant velocity. In this example, three models are employed and \( B_k u_j = 0 \), \( j = 1,2,3 \) in (1)

\[
A_k^1 = I_2 \otimes \begin{bmatrix} 1 & T \\ 0 & 0 \end{bmatrix} ; \quad \Gamma_k^1 = I_2 \otimes \begin{bmatrix} T^2/2 \\ 0 \end{bmatrix} ;
\]

\[
A_k^2 = I_2 \otimes \begin{bmatrix} 1 & T^2/2 \\ 0 & 0 \end{bmatrix} ; \quad \Gamma_k^2 = I_2 \otimes \begin{bmatrix} T^2/2 \\ 1 \end{bmatrix} ;
\]

\[
A_k^3 = A_k^2 \text{ and } \Gamma_k^3 = \Gamma_k^2, \text{ where } I_2 \text{ is the } 2 \times 2 \text{ identity matrix and } \otimes \text{ is the Kronecker product.}
\]

**Table II.**

<table>
<thead>
<tr>
<th>( k = 0 )</th>
<th>Index=0.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1.</strong></td>
<td>( k = k + 1 ).</td>
</tr>
<tr>
<td>If Index=0, Set 1 is utilized at ( k ) and for ( k &gt; 0 ),</td>
<td>( \rho(k) = ap(k-1) + e_i(k) )</td>
</tr>
<tr>
<td>with ( 0 &lt; \alpha &lt; 1 ), ( \rho(0) = 0 ), and</td>
<td>( e_i(k) = \nu(j \max(k))S^{-1}(j \max(k))\nu(j \max,k) )</td>
</tr>
<tr>
<td>( \nu(k) ) for the IM-EV(m=1) algorithm.</td>
<td>( j \max = \arg \max_{1 \leq j &lt; n} [\mu_k(j)] ) is</td>
</tr>
<tr>
<td></td>
<td>obtained in step 5 of</td>
</tr>
<tr>
<td></td>
<td>( \rho(k) &gt; Thl ), then</td>
</tr>
<tr>
<td></td>
<td>( \text{start}_\text{time} = )</td>
</tr>
</tbody>
</table>
| | \( \begin{cases} 
1 & \text{if } k - \frac{1}{1-\alpha} - 1 - \text{guard} \leq 0 \\
1 - \frac{1}{1-\alpha} - 1 - \text{guard} & \text{otherwise}
\end{cases} \) |
| | \( \text{ref}_\text{time} = k \), and Index=1. |
| **Step 2.** | If Index=1, then |
| let Set 2 be activated to process data obtained by Set 1 sequentially from \( \text{start}_\text{time} \) to \( \text{ref}_\text{time} \). |
| **Step 3.** | If Index=2, then a) Set 3 is activated and b) if |
| \( |\hat{a}_x(k)| < \text{Th}_-.a_x \) and \( |\hat{a}_y(k)| < \text{Th}_-.a_y \) | \( |\hat{a}_x(\text{ref}_\text{time})| \) \( \text{and} \)
| \( \text{Th}_-.a_x = 0.02 \times |\hat{a}_x(\text{ref}_\text{time})| \) \( \text{and} \)
| \( \text{Th}_-.a_y = 0.02 \times |\hat{a}_y(\text{ref}_\text{time})| \) |
| then the maneuvering motion is declared to terminate, let Index=0. |
| **Step 4.** | Go to Step 1. |

\[
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{for} \quad j = 1,2,3.
\]

| \( Q_k^j \), \( j = 1,2 \) and \( Q_k^3 = 0.001 \times \Gamma_k^3 \times \Gamma_k^3 \).
| \( H_k^j = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \), \( D_k^j = 1 \).
The measurement noise covariance is of $10^4 \text{m}^2$. The model switching probability matrix is given by

$$
\begin{bmatrix}
0.95 & 0.05 & 0 \\
0.33 & 0.34 & 0.33 \\
0 & 0.05 & 0.95
\end{bmatrix}.
$$

In Table I, $\mu_0(j) = 1/3$, $\hat{\mu}_0 = x_0$, and

$$
\bar{p}_0^j = I_2 \otimes \frac{r}{T} I_2, \quad j = 1, 2, 3 \text{ where } r = 10^4.
$$

In the proposed procedure, let Set 2 and Set 3 be singleton sets consisting of the IMM-EV(m=2) algorithm. Let $\alpha$ be 2/3. Since $n_z = 2$, $\rho(k)$ is approximately 10 degree-of freedom chi-square distributed and thereby $\chi^2_{18.3}$ due to 95% confidence region. Let the parameter $\text{guard}$ be 7.

The tracking performance measures of interest are root-mean-square (rms) x-y position errors and rms x-y velocity errors. Fig. 2 shows that the proposed approach outperforms the IMM algorithm during the 10th-40th and 80th-100th sampling periods. The proposed approach yields slightly larger peak position errors than the IMM algorithm during the 45th-48th sampling periods while their position tracking performance can be considered comparable during the 40th-60th and 60th-80th sampling periods. Fig. 3 shows that the x-y tracking performance of the proposed approach outperforms the IMM algorithm during the 10th-40th, 40th-60th and 80th-100th sampling periods while their performance is comparable during the 60th-80th sampling periods.

The ratio of the execution times of the IMM-EV(m=1), the proposed approach and the IMM algorithm is 1:1.29:1.25. Accordingly, the proposed approach can be regarded as a viable alternative to the IMM algorithm.

**Example 2:** Suppose the target’s movement is sampled every $T=1\text{ sec}$. The target remains a constant speed except during the 50th-60th sampling periods. Fig. 2 shows that the proposed approach can be regarded as a viable alternative to the IMM algorithm.

Twelve models are employed. The system matrices in (1)-(2) are given by

$$
A_k^j = I_2 \otimes \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_k^j = I_2 \otimes \begin{bmatrix} T^2/2 \\ T \\ 0 \end{bmatrix},
$$

$$
A_k^j = I_2 \otimes \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_k^j = I_2 \otimes \begin{bmatrix} T^2/2 \\ T \\ 0 \end{bmatrix},
$$

$$
\Gamma_k^j = I_2 \otimes \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix}, \quad j = 1, \ldots, 12
$$

and $Q_k^j = I_k^j \star \Gamma_k^j$. Furthermore, maneuver inputs are

$$
u_k^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad u_k^2 = \begin{bmatrix} 20 \\ 0 \\ 0 \end{bmatrix}, \quad u_k^3 = \begin{bmatrix} -20 \\ 0 \\ 0 \end{bmatrix}, \quad u_k^4 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},
$$

$$
u_k^5 = \begin{bmatrix} 0 \\ -20 \\ 0 \end{bmatrix}, \quad u_k^6 = \begin{bmatrix} 20 \\ 20 \\ 0 \end{bmatrix}, \quad u_k^7 = \begin{bmatrix} -20 \\ 20 \\ 0 \end{bmatrix}, \quad u_k^8 = \begin{bmatrix} 0 \\ 0 \\ 20 \end{bmatrix},
$$

$$
u_k^9 = \begin{bmatrix} 0 \\ -20 \\ 0 \end{bmatrix}, \quad u_k^{10} = \begin{bmatrix} 40 \\ 0 \\ 0 \end{bmatrix}, \quad u_k^{11} = \begin{bmatrix} 0 \\ 0 \\ 40 \end{bmatrix}, \quad u_k^{12} = \begin{bmatrix} 0 \\ 0 \\ -40 \end{bmatrix}
$$

Henceforth, $u_k^j, 1$ and $u_k^j, 2$ denote the first and the second components of the input vector $u_k^j$, respectively.
For each model, the measurement equation is given by

\[ z_k = \left[ \sqrt{x^2(k) + y^2(k)} + \omega_k^j \right], \quad j=1,2,...,12 \]

and the covariance matrices of measurement noises are

\[ R_k^j = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 2 \times 10^{-3} \end{bmatrix} \]

where

\[ \sigma = \sqrt{\frac{x^2(k) + y^2(k)}{100}} \]

\[ \text{if } \sqrt{x^2(k) + y^2(k)} > 5000 \]

\[ \text{otherwise} \]

The model transition probabilities are given by

\[ p_{ij} = \begin{cases} 0.97 & \text{if } i = j \\ 0.03 \frac{1}{(p-1)} & \text{otherwise} \end{cases}, \quad i,j=1,...,12 \]

where \( p \) is the number of models. In Table I, \( \mu_0 = 1/12 \), \( \hat{x}_0 = x_0 \),

where \( r = 50^2 \). Since the measurement equation is nonlinear, we employ the first-order extended Kalman filters which incorporate the measurements in their polar form into state estimation in Cartesian coordinates.

For the proposed approach, we take Set 2 consisting of the IMM-EV(m=2) algorithm. Let \( \alpha \) be 0.8. Since \( n_z = 2 \), we obtain \( \rho(k) \) approximately which in turn yields the threshold \( T_{th} \) as 16.55 corresponding to 95% confidence region. Let the parameter \( \text{guard} = 5 \). Figures 4-5 show that the Viterbi tracking method [2] fails to improve the performance of the IMM during and after the occurrence of the maneuver despite its efficient computation. Figs. 4-5 show that the proposed approach yields much larger peak position errors than the IMM sporadically whereas the former outperforms or performs as well as the latter for most of the time. Moreover, the ratio of the execution times of the IMM-EV(m=1), the proposed approach and the IMM is 1:1.68:1.49.

![Figure 4. Performance comparison based on rms x-y position errors (100 runs)](image1.png)

**Figure 4. Performance comparison based on rms x-y position errors (100 runs)**

![Figure 5. Performance comparison based on rms x-y velocity errors (100 runs)](image2.png)

**Figure 5. Performance comparison based on rms x-y velocity errors (100 runs)**

**V. CONCLUSIONS**

We have presented a tracking approach based switching among some IMM-EV algorithms. The proposed approach takes the combined strengths of the IMM-EV algorithms for achieving a satisfactory tradeoff between estimation performance and computational load. Simulation results have demonstrated that the proposed approach can be a viable alternative to the IMM algorithm. Moreover, the proposed filter may be further extended to the incorporation of data association, missed contacts, and the impact of false contacts.

**REFERENCES**


