ADAPTIVE INTERACTING MULTIPLE MODEL TRACKING
OF MANEUVERING TARGETS

Jeffery R. Layne, Ph.D. and Uditha C. Piyasena
WL/AACF
2241 Avionics Circle
WPAFB, OH 45433
laynejr@aa.wpafb.af.mil

ABSTRACT

In this paper we investigate adaptive interacting multiple model (AIMM) tracking techniques. Here we compare two existing AIMM approaches and propose a new novel technique. Our algorithm is based on the interacting multiple model (IMM) tracking method with the addition of an adaptive acceleration model to track behavior that falls in between the fixed model dynamics. In this research, we found that the adaptive model matches more closely the true system dynamics when the target kinematics lie in between the fixed models thus improving the overall performance of the tracking system. Also, we found that our new AIMM outperforms the classical IMM as well as the existing adaptive approaches with reduced computational complexity.

INTRODUCTION

For targets with fixed kinematic behavior, a single model state estimator is sufficient for tracking targets. However, for targets with varying or multiple kinematic behaviors, (e.g. maneuvering targets) multiple models are sometimes employed. Such is the case in the classical interacting multiple model (IMM) tracking [1] where each model is assigned a fixed deterministic acceleration to deal with different target accelerations. The probability of each model being true is found using a likelihood function for the model. Movement between the models is performed using a transition probability.

The IMM has shown to provide better results than switching schemes because a smooth transition is achieved between models. Further, system accuracies can be improved by increasing the number of models. However, the IMM algorithm will sometimes result in increased computational complexity and degraded performance when the true system dynamics lie between the fixed models.

To reduce these limitation of the classical IMM, some researchers have proposed adaptive IMM approaches. An example of such an algorithm was developed by Munir and Atherton [2]. In this research, Munir and Atherton estimate the target's acceleration and send the result to a bank of IMM filters. They then allow the entire bank of filters to move in acceleration space such that the bank is centered on the acceleration estimate. By doing this a fewer number of models are required to cover the acceleration space.

Another adaptive multiple model estimator is the so-called moving-bank multiple model adaptive estimator (MBMMAE) presented in [3,4]. Because the bank of filters is fixed and does not move, perhaps a more appropriate name might be the moving-window multiple model adaptive estimator. This technique is not based on the IMM; however, the basic idea is easily implemented in the IMM framework. Here all of the models are assume to be fixed. However, to reduce computational complexity the MBMMAE sets a window around a subset of the filter bank which is centered on the estimated acceleration. Only the filters that fall within the window propagate and update their estimates. Everything falling outside the window is turned off.

In this research we propose another approach. First, we reduce the number of filters in the bank so that the acceleration space is covered at courser levels. Then we add a single adaptive acceleration model. The adaptive acceleration model is designed such that it captures the target dynamics when its behavior falls between the fixed models. As illustrated in this paper, we found that our new AIMM outperforms both the classical IMM and the other adaptive IMMs with reduced computational complexity.
This paper is outlined as follows. To introduce notation and to provide background, first we present the basic theory used in a classical IMM. In the next section we present the adaptive IMM approaches studied in this paper, including our newly proposed technique. Then we present an empirical comparison of both the classical IMM and the adaptive IMM approach. Further we show that our new adaptive IMM outperforms both the classical approach and the other adaptive approaches with reduced computational complexity. Finally, we present our conclusions and discuss future research directions.

**INTERACTING MULTIPLE MODEL TRACKING**

Figure 1 shows the architecture for the classical IMM. In IMM tracking it is assumed that the system obeys one of a finite number of fixed models. As shown in Figure 1, the estimate and covariance that is sent to the filter at time \( k-1 \) is the weighted Gaussian mixture of the filters estimate at \( k-1 \). This process is usually referred to as interaction, or mixing of the estimates. Furthermore, the output estimate, usually referred to as the combined estimate, is the weighted Gaussian mixture of each of the IMM models.

\[
x_j(k+1) = \Phi x_j(k) + \Gamma [a_j(k) + w(k)],
\]

where \( x_j(k) = [p_j(k) v_j(k) p_j(k) v_j(k)]' \) is the state vector of \( x \) and \( y \) position and velocity, \( a_j(k) = [a_j(k) a_j(k)]' \) is the acceleration vector, and \( w(k) \in \mathbb{R}^2 \) is the white noise vector such that \( w(k) \sim N(0,Q) \). The matrices \( \Phi \in \mathbb{R}^{4 \times 4} \) and \( \Gamma \in \mathbb{R}^{2 \times 2} \) are the state transition matrix and the measurement matrix, respectively, such that

\[
\Phi = \begin{bmatrix}
1 & T & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & T \\
0 & 0 & 0 & 1 \\
\end{bmatrix}, \\
\Gamma = \begin{bmatrix}
T^2 & 0 \\
0 & T \\
0 & 2 \\
0 & T \\
\end{bmatrix}.
\]

Each model of the IMM assumes a different acceleration. For the IMM studied here we setup nine models with accelerations as shown in Figure 2. A Kalman filter is designed for each of these models.

In this preliminary analysis we assumed that we have noise corrupted measurements of \( x \) and \( y \) position, i.e.

\[
z(k) = Hx(k) + v(k)
\]

where \( v(k) \in \mathbb{R}^2 \) is a white noise vector such that \( v(k) \sim N(0,R) \) and \( H \in \mathbb{R}^{2 \times 2} \) is the measurement matrix given by

\[
H = \begin{bmatrix}
(0,0) & (0,0) \\
(0,0) & (0,0) \\
(0,0) & (0,0) \\
(0,0) & (0,0) \\
\end{bmatrix}
\]

5.3-17
Interaction/Mixing of the Estimates

The mixing probability that model $M_i$ was in effect at $k-1$ given that $M_j$ is in effect at $k$ conditioned on the measurement history up to $k-1$, denoted $p_{ij}(k-1)$, is

$$
\mu_{ij}(k-1) = \frac{p_j \mu_i(k-1)}{\tilde{c}_j}
$$

where $p_j$ is the a priori known mode transition probabilities, $\mu_i$ is the model probability discussed in the next section, and the normalizing constant is

$$
\tilde{c}_j = \sum_{i=1}^{N} p_i \mu_i(k-1).
$$

The mixed estimate and covariance for the filter associated with model $M_j$ is given by

$$
\hat{x}_j(k-1) = \sum_{i=1}^{N} \mu_{ij}(k-1) \hat{x}_i(k-1),
$$

$$
P_j(k-1) = \sum_{i=1}^{N} \mu_{ij}(k-1) \left[ P_i(k-1) - \left[ \hat{x}_i(k-1) - \hat{x}_j(k-1) \right] \left[ \hat{x}_i(k-1) - \hat{x}_j(k-1) \right]^T \right],
$$

respectively.

Kalman Filtering

The updates for the $j^{th}$, $\{j=1,N\}$, filter model are computed via the Kalman filter equations using the mixed estimate and covariance, i.e.

$$
x_j(k|k-1) = \Phi x_j(k-1) + \Gamma \hat{a}_j(k)
$$

$$
P_j(k|k-1) = \Phi P_j(k-1) \Phi^T + \Gamma Q_j \Gamma
$$

$$
S_j(k) = H P_j(k|k-1) H^T + R_j
$$

$$
V_j(k) = z(k) - H \hat{x}_j(k|k-1)
$$

$$
K_j(k) = P_j(k|k-1) H^T S_j^{-1}
$$

$$
P_j(k|k) = P_j(k|k-1) - K_j(k) H P_j(k|k-1)
$$

$$
\hat{x}_j(k|k) = \hat{x}_j(k|k-1) + K_j(k) V_j(k)
$$

where $x_j(k|k) \in \mathbb{R}^d$ is the state estimate for this model, $P_j(k|k) \in \mathbb{R}^{d \times d}$ is the covariance of the state estimate error, $v_j(k) \in \mathbb{R}^d$ is the innovation process, $S_j(k) \in \mathbb{R}^{d \times d}$ is the covariance of the innovation process, and $K_j(k) \in \mathbb{R}^{d \times d}$ is the Kalman filter gain.

Model Probability

The likelihood function of model $\{M_j: j=1,N\}$ at time $k$, under the linear Gaussian assumption is given by

$$
\Lambda_j(k) = p(z(k)|Z^{k-1}, M_j) = p(v_j(k)) = N[v_j(k); 0, S_j(k)].
$$

The posterior probability of model $j$ being correct is obtained recursively

$$
\mu_j(k) = \frac{\Lambda_j(k) \tilde{c}_j}{c}
$$

where the normalizing constant is given by

$$
c = \sum_{j=1}^{N} \Lambda_j(k) \tilde{c}_j.
$$

This recursive algorithm is started with $\mu_j(0)$ specified a priori.

Estimate and Covariance Combination

The estimated probability density function of the system state is the weighted Gaussian mixture of the $N$ model estimates, i.e.

$$
p[x(k)|Z^k] = \sum_{j=1}^{N} \mu_j(k) N[x(k); \hat{x}_j(k|k), P_j(k|k)],
$$

which has a mean given by

$$
\hat{x}(k|k) = \sum_{j=1}^{N} \mu_j(k) \hat{x}_j(k|k),
$$

and covariance given by

$$
P(k|k) = \sum_{j=1}^{N} \mu_j(k) \left[ P_j(k|k) - \left[ \hat{x}_j(k|k) - \hat{x}(k|k) \right] \left[ \hat{x}_j(k|k) - \hat{x}(k|k) \right]^T \right].
$$

ADAPTIVE IMM TECHNIQUES

As mentioned previously, there are two known adaptive techniques in addition to ours presented here. All of these adaptive techniques rely on having good knowledge of the target's acceleration.

In this research we tried several methods for estimating acceleration. First we tried to estimate the acceleration by computing the numerical derivative of the combined velocity estimate. However, this technique led to a very noisy estimate and produced unacceptable results. Then we tried computing the
second numerical derivative of the combined position estimate. Although this technique produced a slightly noisy estimate, the result were reasonably good. To reduce the noise in this acceleration estimate, we tried filtering the estimate using both a moving average window filter and a moving median window filters. They smoothed the estimate but resulted in an unacceptable delay in the acceleration measurement.

Finally, we found that the best technique was to design a separate Kalman filter for estimating the bias (acceleration). To do this let \( \mathbf{x}_b(k) = [p_b(k) v_b(k) a_d(k) p_v(k) v_v(k) a_v(k)]' \), then the acceleration is estimated by the following bias filter:

\[
\begin{align*}
    \mathbf{x}_b(k|k-1) &= \Phi_b \mathbf{x}_b(k - 1|k-1) \\
    \mathbf{P}_b(k|k-1) &= \Phi_b \mathbf{P}_b(k - 1|k-1) \Phi_b^T + \mathbf{Q}_b \Gamma_b \\
    \mathbf{S}_b(k) &= \mathbf{H}_b \mathbf{P}_b(k|k-1) \mathbf{H}_b^T + \mathbf{R}_b \\
    \mathbf{V}_b(k) &= \mathbf{z}(k) - \mathbf{H}_b \hat{\mathbf{x}}_b(k|k-1) \\
    \mathbf{K}_b(k) &= \mathbf{P}_b(k|k-1) \mathbf{H}_b \mathbf{S}_b(k)^{-1} \\
    \mathbf{P}_b(k|k) &= \mathbf{P}_b(k|k-1) - \mathbf{K}_b(k) \mathbf{H}_b \mathbf{P}_b(k|k-1) \\
    \hat{\mathbf{x}}_b(k|k) &= \hat{\mathbf{x}}_b(k|k-1) + \mathbf{K}_b(k) \mathbf{V}_b(k)
\end{align*}
\]

where

\[
\Phi_b = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},
\]

\[
\Gamma_b = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 1 & 0 \\ 0 & T^2/2 \\ 0 & T \\ 1 & 0 \end{bmatrix},
\]

and

\[
\mathbf{H}_b = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.
\]

### Munir and Atherton Adaptive IMM

As previously mentioned, one adaptive IMM that we studied in this research was the Munir and Atherton method [2]. In this method the entire bank of filter was "moved" so that it was centered on the current estimate of acceleration. In other words, the bank was designed so that the model accelerations are as shown in Figure 3.

#### Moving-Window Adaptive IMM

The moving-window adaptive IMM was developed based on [3,4]. As shown in Figure 4, only a subset of the nine filters, those that are contained in a window centered on the acceleration estimate, are operational at any given time. All other filters outside the window are turned off. When the window moves new filters that are brought on line and are initialized by the previous combined state estimate and covariance. Although this technique will not necessarily improve the state estimate over the classical IMM, it does provide a significant reduction in computation.
New Adaptive IMM

The adaptive IMM proposed in this paper is illustrated in Figure 5. It is based on having a number of fixed IMM models and a single adaptive model to capture behavior that falls in between the fixed model dynamics. Further, to reduce computations, the number of fixed models was reduced so that the acceleration space is covered at courser levels. The analysis in this paper is based on only five fixed models.

![y-acceleration diagram](image)

Figure 5 - Location of models in the proposed adaptive IMM.

4.0 Simulation Results

In this section, we compare the performance of our adaptive IMM algorithm to the classic IMM algorithm, the Atherton IMM algorithm, and the moving-window IMM algorithm. In these simulations, we used the target track shown in figure 1. From this figure one can clearly see that the target is making two maneuvers. The first at time 32 and the second at time 48.

![Target Track](image)

Figure 6 - The target track considered in this paper.

The nine model IMM was used as a basis of comparison for the other algorithms considered in this paper. To reduce computations, our adaptive IMM was designed using only six models (five fixed and one adaptive). We chose to use six models to provide a fair comparison with the classical IMM. If we had simply added an adaptive model to the nine model IMM, we would have had ten models in our AIMM. The comparison would not be fair as better performance might be obtained at the cost of additional computations. We also implemented the adaptive IMM presented by Munir and Atherton [2] and the moving-window adaptive IMM [3,4] to determine how they would compare with our adaptive IMM.

For each of the algorithms, we computed the average RMS errors in the x and y axis for 100 Monte Carlo runs. The results are shown in Table 1. Further we compared the number of Matlab FLOPS required by each algorithm to determine which was most efficient.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Flops</th>
<th>RMSx</th>
<th>RMSy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic IMM</td>
<td>86534400</td>
<td>3.55</td>
<td>3.66</td>
</tr>
<tr>
<td>Atherton IMM</td>
<td>98764800</td>
<td>3.72</td>
<td>3.55</td>
</tr>
<tr>
<td>Moving Window IMM</td>
<td>67814400</td>
<td>4.14</td>
<td>3.25</td>
</tr>
<tr>
<td>AIMM</td>
<td>64665600</td>
<td>3.27</td>
<td>3.25</td>
</tr>
</tbody>
</table>

Table 1 - Flops count and the average RMS error for each algorithm.

From this table we see that the classic IMM gives us average RMS errors of 3.55 and 3.66 in the x and y axis, respectively. On the other hand, the Munir and Atherton method gives us average RMS errors of 3.72 and 3.55, respectively. This shows us that the Munir and Atherton technique performs similar to the classic IMM. Also, note that the moving-window model has a much larger RMS error than the classic IMM model. On the other hand, the average RMS errors of our adaptive IMM are much smaller than those obtained for the other methods. Consequently, it is clear that our adaptive IMM outperforms all the other techniques considered in this paper.

Also, the adaptive IMM requires the fewest FLOPS of all the algorithms considered. Further, we see that the FLOPS count is largest in the Munir and Atherton method. The reason for this is clear since we are doing additional computations over the classical IMM when estimating the acceleration. Also note that the flops
count of the moving-window method is also less than the classic IMM method. This is due to the fact that only 4 models are active at any given time.

Figures 7-10 show plots of the RMS errors over 100 Monte Carlo runs for each method. By examining Figure 7 you can see that the RMS error for the classic IMM is increased when the target is doing a maneuver. In Figure 8 we see that the Munir and Atherton model attempts to move its center acceleration model to where the maneuver occurs. Hence the RMS error is large until until the move is completed and the error becomes smaller. That is why the Munir and Atherton method does not outperform the classic IMM method for this case.

Figure 9 shows us that the RMS errors of the moving-window method exhibit a large transient when the window moves as a result of a maneuver. The reason for this maybe due to the fact that the target does a swift transition from one set of acceleration to another. We believe that this model is unable to handle such a fast transition. (We made many attempts to correct this problem without finding a satisfactory solution.)

Figure 10 shows that our adaptive IMM provided the best overall performance with the least number of computations. The reason for this is simple. The adaptive model is able to capture the target’s dynamics when the accelerations fall between the fixed model dynamics.
Figures 11-14 show the plots of the probability of each model in each algorithm. When looking at the plot for the Munir and Atherton method we see that the "center" model is always moving to match the target acceleration. Because of this the probability of the center model rises quickly during a transition in acceleration.

Notice in Figure 13 that the moving-window method is unable to handle the transition from one quadrant to another very well. Finally in Figure 14 it is observed that the adaptive IMM performs well when the target changes maneuver. More specifically we see that our adaptive method captures the maneuver well when the maneuver is not matched exactly by one of the fixed models. It is this feature that makes our method better than the traditional IMM and others.

CONCLUSION

The main goal of this paper is to provide an empirical analysis of adaptive IMM schemes. The results provide evidence that our new adaptive IMM outperforms the classical IMM and other adaptive IMM algorithms with reduced computations. However, we do not suggest that the results of this study represent a real target tracking system. The units are not specified and the sensor models are very crude.

Future work will focus on the processing of real data or "good" synthetic data obtained from high fidelity sensor models. This will give us a better idea if these techniques are truly applicable for tracking maneuvering targets.
REFERENCES


