Incorporating World Information into the IMM Algorithm via State-Dependent Value Assignment

Rastin Rastgoufard Vesselin P. Jilkov X. Rong Li
Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148, U.S.A.
Email: {rrastgou, vjilkov, xli}@uno.edu

***Abstract*—We propose two methods of incorporating world information as modifications to the Interacting Multiple Model (IMM) algorithm via state-dependent value assignment. The value of a state is a measure of its worth, so, for example, waypoints have high value and regions inside obstacles have small value. The two methods involve modifying the model probabilities in the update step and modifying the transition probability matrix in the mixing step based on the assigned values of target states. The state-dependent value assignment modifications to the IMM algorithm are simulated and compared with the standard IMM algorithm over a large number of game player-controlled trajectories for obstacle avoidance, as ground truth, and are shown experimentally to perform better than the standard IMM algorithm in both target's current state estimation and next state prediction. The proposed modifications can be used for improved trajectory estimation or prediction in real-life applications such as, e.g., Air Traffic Control, ground target tracking and robotics, where additional (world) information is available.**

***Index Terms*—IMM, state-dependent transition probabilities, constraints, penalty function, waypoints, obstacles**

I. Introduction

Multiple model (MM) estimation is the state-of-the-art approach to maneuvering target tracking [1], [2] wherein maneuvering target tracking problems are typically modeled through "hybrid systems" [3] in which the target kinematic state is continuous, but the target mode of motion at any time is described by a finite number of kinematic models. MM estimation can be categorized into three generations—autonomous, cooperating and variable-structure, the first two of which use a fixed-structure of a model set. A very popular MM algorithm is the (fixed structure) Interacting Multiple Model (IMM) algorithm [4], [5] which runs several Kalman filters in parallel and merges their results. The IMM algorithm is widely-used because it is cost-effective, meaning that it performs relatively well and is computationally inexpensive.

In many real-world applications the modes of motion of a target depend on its state, and additional (world) information on the target behavior, as a function of its state, is available. For example, almost never does a civilian aircraft approach a runway for landing or a car on a highway move in an arbitrary way. The airplane must respect the rights of way of all other airplanes and follow instructions of the air traffic controller, and the car is likely to maintain its lane on the road if it does not wish to cause an accident. Often the airplane or car is the target to be tracked, and in both cases, information about the world around the target affects its motion. Furthermore, the impact that the world has on the target depends not only on the world but also on the state of the target: the target responds to the current conditions of the world around it. A way of improving the prediction/tracking performance is to incorporate as much as possible any available world information (which can depend on the target's state) into the tracking algorithm.

A simple illustration of using additional information within the fixed structure IMM algorithm is given in [6] in which a vehicle is moving along a highway. Two conditions are of interest – maintaining a lane or changing lanes. There is a motion model and associated "directional" process noise that corresponds to maintaining a lane, and there is a different motion model with a different type of process noise associated with the lane change maneuver. [6] shows that the IMM algorithm tracks the vehicle well under both conditions and quickly determines when lane changes happen. The target behavior is captured by just two models. Most practical problems, however, require many more modes of operation to characterize a target's range of motion. The IMM algorithm's performance suffers when there are too many motion models that overlap and compete [7]. The variable-structure multiple model (VSMM) approach and algorithms [2], [7] can significantly outperform the traditional fixed-structure IMM algorithm, (e.g, [7]–[11]). Moreover, those VSMM algorithms inherently posses the capability of capturing world information because of the fact that the set of models can be adapted based on the target's current state. For example, [7] describes a problem in which the acceleration of the target cannot change rapidly. An overarching set of models is designed to cover all of the possible target accelerations, but at any time the VSMM algorithm uses the set of accelerations that are "near" the target's current acceleration. As the target's acceleration changes, the VSMM algorithm chooses different sets of models accordingly.

VSMM algorithms with model selection or switching rules that are based not only on the target's state but also on the properties of the world around the target have been already applied to real-word problems with great success. For example,

This research was supported in part by NASA/LEQSF(2013-15)-Phase3-06 through grant NNX13AD29A.

Rastin is currently a Ph.D. student in the Department of Electrical Engineering at the University of New Orleans.

[12] and [13] limit the available modes of operation based on the presence of roads and whether or not the target is on a road. [12] describes a general ground target tracking problem where a target may be navigating in an unconstrained environment (off-road); it may be near a road, or it may be constrained to be on a piecewise linear road. Furthermore, roads may have junctions, where the target can choose different branches. Each condition, including motion at a junction, is captured by a different set of models, and there is a very intricate method of selecting which models are applicable. [13] expands the on-road condition and describes how to incorporate the actual curvature of road segments as constraints.

In the VSMM methods, the state-dependent information is captured by the strategic selection and omission of models. The work presented in this paper is focused on embedding state-dependent and world information into the model transition probabilities or the model probabilities of the IMM algorithm directly. Similar approaches within the fixed structure IMM algorithm have been reported in the literature, e.g., [14]–[16], where the model transition probabilities or model likelihoods are modified based on the target's state. [15], [16] use a transition probability matrix (TPM) that depends on the proximity to a waypoint through guard conditions. An example is given in [15] where an airplane should turn toward a new destination when it arrives at a waypoint. There are two guard conditions to model this desired behavior. The first is to switch from constant velocity to constant turn when the plane's position is near the waypoint. The second is to switch away from constant turn back to constant velocity when the plane's heading is near a specific angle. [14] considers a problem in which a target can choose to stop randomly as an evasive maneuver. The authors make the argument that real world targets cannot instantaneously cease their motion, and thus the probability that the target will stop is small when its speed is large. The TPM governing the switching of motion models depends on the speed of the target. The methods proposed in this paper, along with some of the aforementioned methods, can complement the VSMM methods and would operate after the set of models is selected.

Briefly, the main idea of our approach is as follows. We consider two locations in the IMM algorithm that allow world information to be incorporated. The first is in the update step of the model probabilities (MPs) that uses the likelihoods of each model. The second is in the TPM used to predict the model probabilities for re-initialization of the conditional filters. Just before the end of one cycle of the IMM algorithm there is one estimated state for each of the models in the algorithm. Each of these states is assigned a value, based on the world information, and the values then modify the weights of their respective models in the final update step of the algorithm. The details of this process are given in Sec. III.B (State-Dependent Model Probabilities). Furthermore, the estimated states will interact in the mixing step to obtain the next cycle's re-initialization. Before that happens, each estimated state is propagated by all motion models to determine "what-if" predicted states. The number of these predicted states is equal to the number of elements in the TPM. The assigned values of these predicted states characterize the values of the transitions, and thus they are used to modify the TPM, as given in Sec. III.C (State-Dependent Transition Probabilities).

II. Outline of the IMM Algorithm

We give here a brief outline of the IMM algorithm as far as it is needed for the remainder of the paper.

The IMM algorithm runs several Kalman filters in parallel. The individual filters are re-initialized at each time step (cycle) using a probabilistic mixture of the results from the previous step using the predicted model probabilities and the TPM, and after each filter is run the model probabilities are updated through the model likelihoods. For output, an overall state estimate is computed as a mixture of the individual filters' estimates.

More specifically, consider the following Markov jump linear systems where the ith model of the finite MM set $\mathbb{M} = \{m^{(1)}, \ldots, m^{(M)}\}$ obeys the following equations:

$$x_{k+1} = F_k^{(i)} x_k + G_k^{(i)} w_k^{(i)} \tag{1}$$

$$z_k = H_k^{(i)} x_k + v_k^{(i)} \tag{2}$$

where $E[w_k^{(i)}] = \bar{w}_k^{(i)}$, $\text{cov}(w_k^{(i)}) = Q_k^{(i)}$, $E[v_k^{(i)}] = \bar{v}_k^{(i)}$, $\text{cov}(v_k^{(i)}) = R_k^{(i)}$. Superscript (i) denotes quantities pertinent to model $m^{(i)}$ in $\mathbb{M}$, and the jumps of the system mode are assumed to have transition probabilities

$$P\{m_{k+1}^{(j)} | m_k^{(i)}\} \triangleq \pi_{ij} \tag{3}$$

where $m_k^{(i)}$ denotes the event that model $m^{(i)}$ matches the system mode in effect at time k. A complete recursion of the IMM algorithm with Kalman filters as its elemental filters is summarized in Table I [1] for the Markov jump linear system (1)–(3) with white Gaussian independent process and measurement noises.

III. State-Dependent Value Assignment

To every possible state that the target can take we assign a penalty or benefit value depending on the world information. For example, we could define a simple mapping such that every location inside of an obstacle is assigned a zero and every other location is assigned a one. The state-to-value mapping is used to modify the update of mode probability (see Table I) and the TPM $\Pi = (\pi_{ij})_{i,j=1}^{M}$ in (3) that is used for model-conditioned re-initialization (see Table I).

A. A State's Value

To explain the idea, let us consider a scenario of obstacle avoidance, as illustrated in Fig. 1. Such scenarios are typical in, e.g., aircraft flight re-routing in order to avoid regions of bad weather and in mobile robotics to avoid obstacles.

The state-to-value mapping described below is only one of many possible mappings for the specific type of obstacles in Fig.1. Every problem may have many possible mappings, and the design of the state-to-value mapping is something to be carefully considered by the designer for any specific application. In Fig. 1, there are two factors that give hints towards the design. First, the obstacles are assumed, for simplicity, to be circles

TABLE I: One cycle of the IMM estimator [1].

1. Model-conditioned reinitialization (for $i = 1, 2, \ldots, M$):

Predicted mode probability: $\mu_{k|k-1}^{(i)} \triangleq P\{m_k^{(i)}|z^{k-1}\} = \sum_j \pi_{ji}\mu_{k-1}^{(j)}$

Mixing weight: $\mu_{k-1}^{j|i} \triangleq P\{m_{k-1}^{(j)}|m_k^{(i)}, z^{k-1}\} = \pi_{ji}\mu_{k-1}^{(j)}/\mu_{k|k-1}^{(i)}$

Mixing estimate: $\bar{x}_{k-1|k-1}^{(i)} \triangleq E[x_{k-1}|m_k^{(i)}, z^{k-1}] = \sum_j \hat{x}_{k-1|k-1}^{(j)}\mu_{k-1}^{j|i}$

Mixing covariance:
$\bar{P}_{k-1|k-1}^{(i)} = \sum_j [P_{k-1|k-1}^{(j)} + (\bar{x}_{k-1|k-1}^{(i)} - \hat{x}_{k-1|k-1}^{(j)})(\bar{x}_{k-1|k-1}^{(i)} - \hat{x}_{k-1|k-1}^{(j)})']\mu_{k-1}^{j|i}$

2. Model-conditioned filtering (for $i = 1, 2, \ldots, M$):

Predicted state: $\hat{x}_{k|k-1}^{(i)} = F_{k-1}^{(i)}\bar{x}_{k-1|k-1}^{(i)} + G_{k-1}^{(i)}\bar{w}_{k-1}^{(i)}$

Predicted covariance: $P_{k|k-1}^{(i)} = F_{k-1}^{(i)}\bar{P}_{k-1|k-1}^{(i)}(F_{k-1}^{(i)})' + G_{k-1}^{(i)}Q_{k-1}^{(i)}(G_{k-1}^{(i)})'$

Measurement residual: $\tilde{z}_k^{(i)} = z_k - H_k^{(i)}\hat{x}_{k|k-1}^{(i)} - \bar{v}_k^{(i)}$

Residual covariance: $S_k^{(i)} = H_k^{(i)}P_{k|k-1}^{(i)}(H_k^{(i)})' + R_k^{(i)}$

Filter gain: $K_k^{(i)} = P_{k|k-1}^{(i)}(H_k^{(i)})'(S_k^{(i)})^{-1}$

Updated state: $\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k^{(i)}\tilde{z}_k^{(i)}$

Updated covariance: $P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)}S_k^{(i)}(K_k^{(i)})'$

3. Mode probability update (for $i = 1, 2, \ldots, M$):

Model likelihood: $L_k^{(i)} \triangleq p[\tilde{z}_k^{(i)}|m_k^{(i)}, z^{k-1}] \overset{\text{assume}}{=} \mathcal{N}(\tilde{z}_k^{(i)}; 0, S_k^{(i)})$

Mode probability: $\mu_k^{(i)} = \frac{\mu_{k|k-1}^{(i)}L_k^{(i)}}{\sum_j \mu_{k|k-1}^{(j)}L_k^{(j)}}$

4. Estimate fusion:

Overall estimate: $\hat{x}_{k|k} = \sum_i \hat{x}_{k|k}^{(i)}\mu_k^{(i)}$

Overall covariance: $P_{k|k} = \sum_i [P_{k|k}^{(i)} + (\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)})']\mu_k^{(i)}$

with known radii. Second, the moving target is "allowed" to be inside an obstacle, but such a maneuver is discouraged. These two factors can be handled by a sigmoid function, defined below in (4).

Assume there are N circular obstacles, each with radius r_i and center $(\mathsf{x}_i, \mathsf{y}_i)$, where $i \in \{1, \ldots, N\}$. Then, consider only the position (x, y) of a state x. The distance between the position of the state x and the ith obstacle center is

$$d_i(x) = \sqrt{(\mathsf{x} - \mathsf{x}_i)^2 + (\mathsf{y} - \mathsf{y}_i)^2}$$

If $d_i(x) > r_i$, then the position is outside the ith obstacle. Define a function $s(x, i)$ as

$$s(x, i) = \frac{1}{1 + \exp(-\beta(d_i(x) - r_i))} \tag{4}$$

It has a sigmoidal shape. A state with position that is outside the ith obstacle has $d_i > r_i$ and $s(x, i)$ is approximately one. A state that is inside the ith obstacle has $d_i < r_i$ and $s(x, i)$ is near zero. The parameter β controls the steepness of the transition between the outside and the inside regions. One minor but nice property of this sigmoidal shape is that the gradient always points away from the center of an obstacle. Fig. 2 shows the values of all states (x, y) with respect to an obstacle centered at $(\mathsf{x}_i, \mathsf{y}_i) = (0, 0)$ with radius $r_i = 2$.

The function $s(x, i)$ is the value of the state x with respect to a single obstacle i. In order to find an overall value of the state x with respect to the world, we define $s(x)$ as the minimum of all N functions $s(x, i)$:

$$s(x) = \min_{1 \le i \le N} s(x, i) \tag{5}$$

B. State-Dependent Model Probabilities

The function $s(x)$ gives the value of every state x. This information can be incorporated into the model probabilities in the update step (see Table I) under the assumption that an intelligent target will want to maneuver toward high-valued states.

Specifically, each model probability $\mu_k^{(i)}$ is updated by

$$\mu_k^{(i)} = \frac{\mu_{k|k-1}^{(i)}L_k^{(i)}s_k^{(i)}}{\sum_j \mu_{k|k-1}^{(j)}L_k^{(j)}s_k^{(j)}} \tag{6}$$

where $s_k^{(i)} = s(\hat{x}_{k|k}^{(i)})$ and $\hat{x}_{k|k}^{(i)}$ is the estimate from the ith conditional filter. In other words, the likelihood of model $m^{(i)}$ is changed from $L_k^{(i)}$ to $L_k^{(i)}s_k^{(i)}$.

C. State-Dependent Transition Probabilities

The states' value information can be embedded into the TPM $\Pi = (\pi_{ij})_{i,j=1}^M$ in (3) that is used for model-conditioned re-initialization (see Table I) under the assumption that an intelligent target will want to maneuver toward high-valued states.

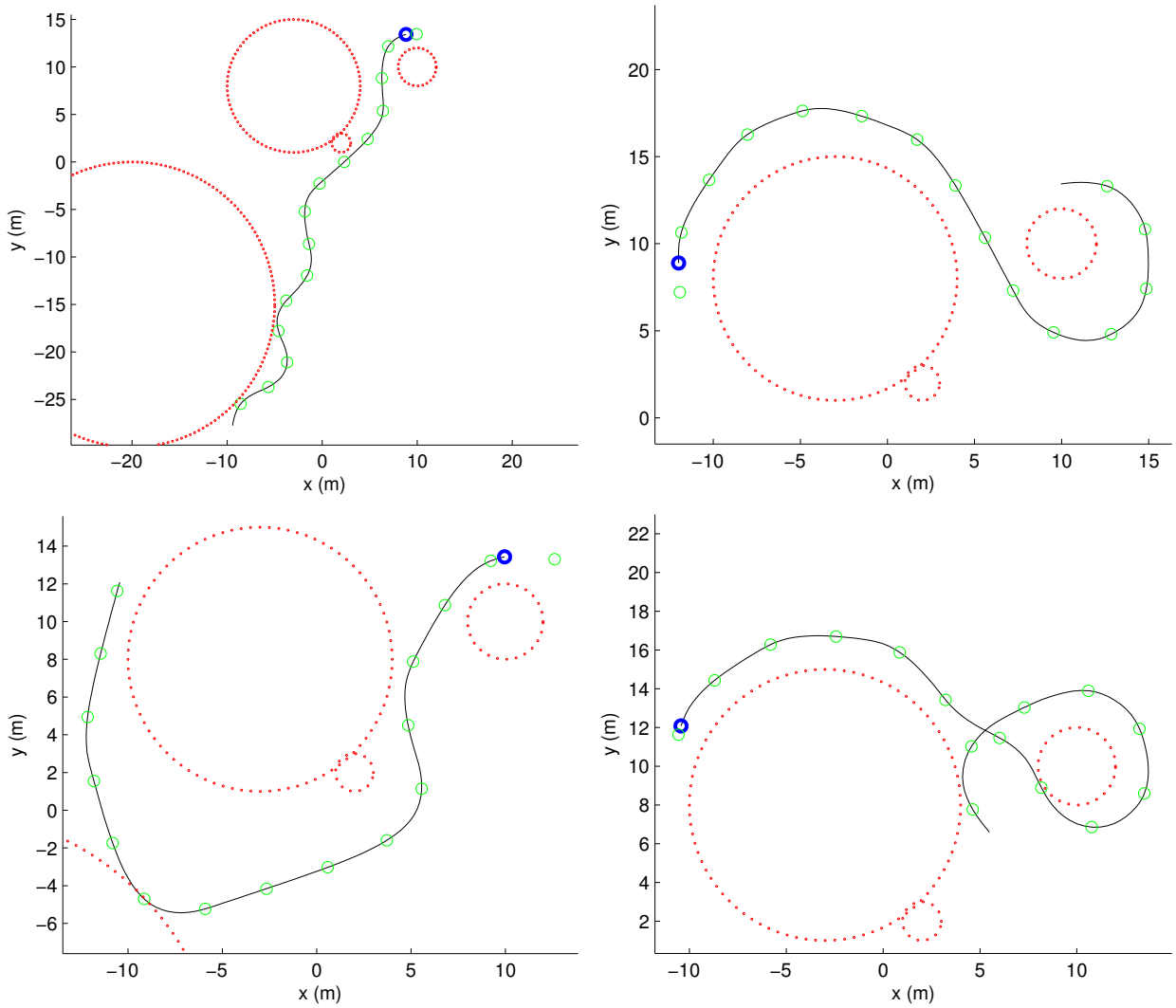

Fig. 1: Sample ground truth trajectories. The black solid line shows the continuous time trajectory, the green circles show error free samples (taken once every $T = 0.35$ seconds) and the heavy blue circle shows the starting location of the trajectory. The red dotted circles are obstacles that the target should avoid.

Before the recursive cycle $(k-1 \to k)$ of the IMM algorithm (Table I), the TPM Π is modified as follows:

$$\pi_{ij,k} = \frac{\pi_{ij} s^{(ij)}_{k|k-1}}{\sum_j \pi_{ij} s^{(ij)}_{k|k-1}} \tag{7}$$

where $s^{(ij)}_{k|k-1} = s(\hat{x}^{(ij)}_{k|k-1})$ and

$$\hat{x}^{(ij)}_{k|k-1} = F^{(j)}_{k-1} \hat{x}^{(i)}_{k-1|k-1} + G^{(j)}_{k-1} \bar{w}^{(j)}_{k-1} \tag{8}$$

That is, the transition probability π_{ij} is modified to $\pi_{ij} s^{(ij)}_{k|k-1}$.

Then the recursive cycle $(k-1 \to k)$ of the IMM algorithm is completed as given in Table I with the modified $\Pi_k = (\pi_{ij,k})_{i,j=1}^{M}$ instead of Π.

Clearly, both modifications (6) and (7) make good sense.

IV. Experiment and Results

Fig. 1 shows four examples of ground truth trajectories. Samples come from the position part of the ground truth every $T = 0.35$ seconds, and these samples are used as the basis of the tracking experiment. The purpose of the experiment is to compare the performance of the standard IMM algorithm with the IMM algorithm with state-dependent modifications.

A. Experimental Design

The first step for setting up the experiment is to design the IMM algorithm's model set. This model set should capture the possible maneuvers that the target can take while simultaneously being as simple as possible. In order to simplify the design, we chose sections of ground truth in which the speed of the target is constant, thus avoiding the need to model linear accelerations. (In all cases shown in Fig. 1, the target is at full throttle the entire time.)

We track the position and the velocity of the target. The state vector is $x = [\mathsf{x}, \mathsf{y}, \dot{\mathsf{x}}, \dot{\mathsf{y}}]'$ where x and y represent coordinates. The model set that we chose consists of five constant turn models with different turn rates. Two left-turn models have $\omega_1 = 1.42 \times 2\pi$ rad/s and $\omega_2 = 0.71 \times 2\pi$ rad/s. Two right-turn

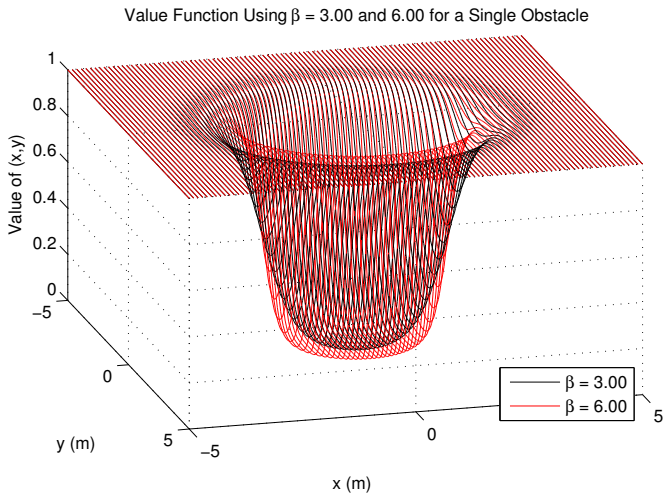

Fig. 2: The function $s(x, i)$, given by (4), for an obstacle with $r = 2$ and two values of β. The region outside the obstacle has a neutral value of 1, but the region inside the obstacle is penalized to have value close to zero.

models have $\omega_4 = -\omega_2$ and $\omega_5 = -\omega_1$. A final model has $\omega_3 = 0$, which corresponds to going straight. The dynamics matrix of a constant turn model [17] is as follows.

$$A(\omega, T) = \begin{pmatrix} 1 & 0 & \frac{\sin(\omega T)}{\omega} & \frac{\cos(\omega T) - 1}{\omega} \\ 0 & 1 & \frac{1 - \cos(\omega T)}{\omega} & \frac{\sin(\omega T)}{\omega} \\ 0 & 0 & \cos(\omega T) & -\sin(\omega T) \\ 0 & 0 & \sin(\omega T) & \cos(\omega T) \end{pmatrix} \quad (9)$$

Fig. 3 shows the possible combinations of the five models after two time steps. The process noise is omitted in the figure just for clarity of illustration. The green circles represent the end position, and each black line shows the trajectory that the target would have taken to get to an endpoint. Even though the mode sequences $[\omega_2, \omega_2]$ and $[\omega_1, \omega_5]$ have the same endpoint, the resulting orientations are different.

The second step for setting up the experiment is to design various filters. We implemented and tested four versions of the IMM algorithm, all of which use the same model set. The first two versions have a constant TPM with large diagonal elements. The first algorithm is the standard IMM algorithm with no state-dependent features. This is called "Normal." The second algorithm is the IMM algorithm with the world information embedded into the model probabilities, as described in Section III.B (State-Dependent Model Probabilities). This is called "SD MPs."

The third and fourth versions have a new TPM at every time step. The third algorithm is the IMM algorithm with the world information embedded into the TPM, as described in Section III.C (State-Dependent Transition Probabilities). This is called "SD TPM." The fourth variation is the IMM algorithm with world information embedded into both the model probabilities and the TPM. This is called "SD Both."

All three of the state-dependent variations use the value function $s(x)$ given by (5). The function $s(x)$ knows the locations and sizes of the obstacles of the world shown in Fig. 1.

The final step for the design of the experiment is to prepare true trajectory samples with corresponding noisy measurements. Fig. 1 shows four example ground truth trajectories. Each green circle represents the truth at a particular time step. Those serve as reference samples and are corrupted by noise of a varying degree to obtain the measurements.

The measurement model used in the IMM algorithms matches the mechanism by which the measurements are created. It uses position-only measurements with additive white Gaussian noise v_k, i.e.,

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \; v_k \sim \mathcal{N}(0, R), \; R = diag\{\sigma_z^2, \sigma_z^2\}$$

The experiment runs all of the filters for a fixed value of σ_z^2 and then repeats with a different σ_z^2 to see the effect of different noise levels.

B. Results

All four versions of the IMM algorithm estimate and predict the motion of the target. Fig. 4 shows the algorithms' estimated and predicted states over two consecutive time steps in one sample run. There are several ground truth trajectories, some of which are shown in Fig. 1.

On each run (trial), the estimation and prediction errors $\tilde{x}_{k|l} = x_k - \hat{x}_{k|l}$ are computed for $l = k, k-1$, respectively, and then scalar time-averaged estimation and prediction errors,

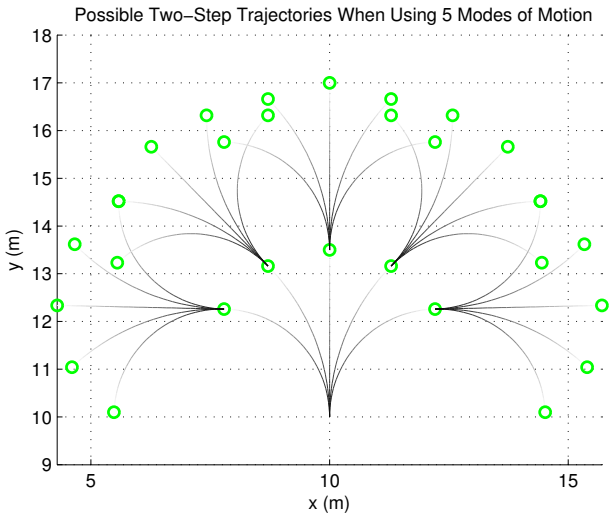

Fig. 3: Possible sequences over two time steps. The initial state is $[\mathsf{x}, \mathsf{y}, \dot{\mathsf{x}}, \dot{\mathsf{y}}]' = [10, 10, 0, 10]'$. Each time step is $T = 0.35$ seconds long. The green circles indicate a possible position (x, y) at each time step.

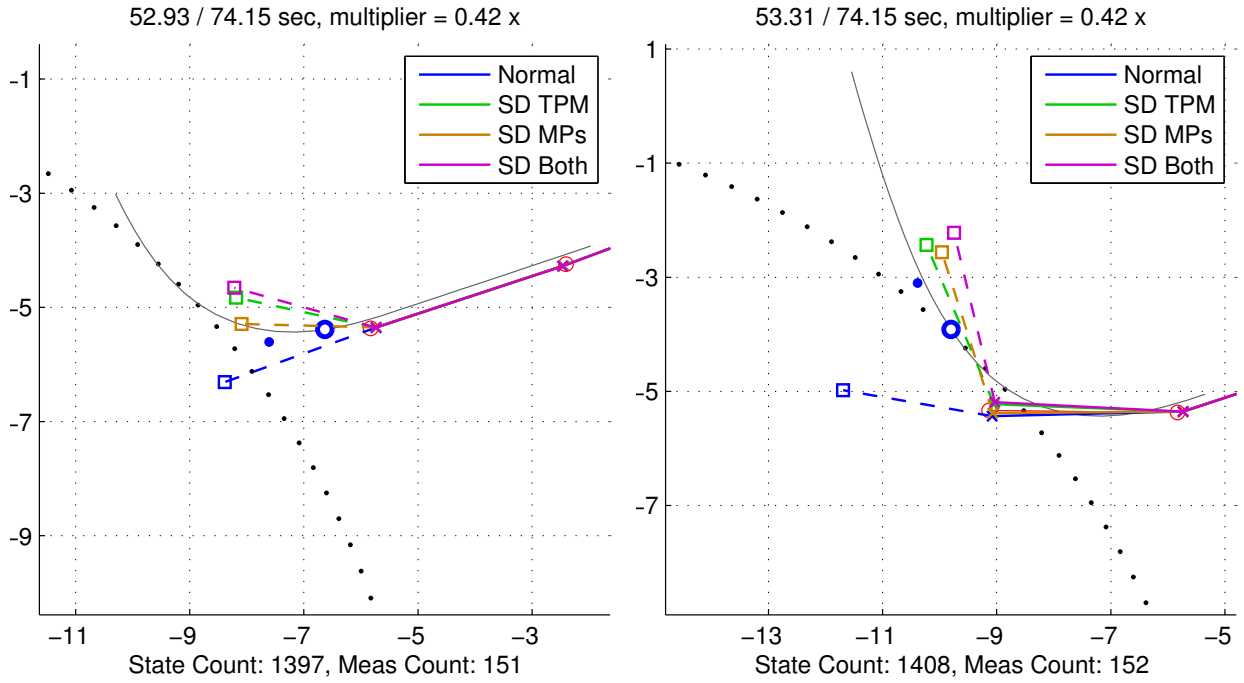

Fig. 4: Close-ups of two consecutive time steps in one run. The heavy blue circle at the center of each frame is the current location of the target, and the heavy blue dot shows its orientation. The thin black line is its actual continuous time trajectory. The two red circles indicate the two most recent measurements. The colored crosses show each algorithm's estimated position output. The four colored squares show each of the algorithm's one-step predicted position. The dotted arc of a circle is an obstacle that the target should avoid. Notice that the "Normal" IMM algorithm tries to continue the motion while the three state-dependent variations predict that the target will maneuver to avoid the obstacle.

respectively, are computed as follows:

$$e_e = \frac{1}{N}\sum_{k=1}^{N}\sqrt{(H\tilde{x}_{k|k})'(H\tilde{x}_{k|k})} \tag{10}$$

$$e_p = \frac{1}{N}\sum_{k=1}^{N}\sqrt{(H\tilde{x}_{k|k-1})'(H\tilde{x}_{k|k-1})} \tag{11}$$

There is one value of e_e per trial for each of the four algorithms in the experiment, and there is one value of e_e for the measurements themselves for each trial. Similarly, there is one value of e_p per trial for each of the four algorithms. In order to assess the four algorithms, each trial is repeated 300 times and the errors are averaged over all trials to obtain $\bar{e}_e$ and $\bar{e}_p$. Table II shows the average errors for three different levels of σ_z^2. In Table II, the measurement error is in bold face because it is the baseline reference. The other bolded values correspond to the filter that had the lowest error, on average, for a specific level of σ_z^2.

C. Discussion

Table II shows the results of the experiment for a typical ground truth case. There are twelve similar ground truth cases, each of which contains cases for seven different measurement noise levels. Thus, the experiment contains a total of eighty-four parameter combinations. All eighty four combinations are documented in detail in [18]. One purpose of this section is to try to identify patterns in the results, and to that end two major questions need to be addressed.

TABLE II: Three sets of estimation and prediction errors averaged over 300 runs for varying amounts of noise σ_z^2.

		Estimation	Prediction
a) $\sigma_z^2 = 0.005$	Measurements	**0.08873**	-
	Normal	0.08827	1.57677
	SD TPM	0.08811	1.43514
	SD MPs	0.08826	**1.40860**
	SD Both	**0.08811**	1.41925
		Estimation	Prediction
b) $\sigma_z^2 = 0.05$	Measurements	**0.27937**	-
	Normal	0.27229	1.78459
	SD TPM	0.27028	1.62484
	SD MPs	0.27196	1.61392
	SD Both	**0.27012**	**1.59280**
		Estimation	Prediction
c) $\sigma_z^2 = 0.5$	Measurements	**0.88759**	-
	Normal	0.85571	2.51311
	SD TPM	0.83711	2.39872
	SD MPs	0.84918	**2.34708**
	SD Both	**0.83372**	2.34748

1) Does knowledge of the environment actually make a difference in tracking performance?
2) Is there a difference between implementing the world information in the model probabilities as opposed to in the TMP?

It is seen from Table II that if the measurement noise is very small, the estimation improvement by the SD versions is not significant, because the normal algorithm is already very accurate (due to almost perfect measurements), but as the measurement noise increases the estimation accuracy improvement of the SD algorithms also increases considerably. With respect to the prediction accuracy, the SD algorithms are clearly better.

Table III provides some counts that help make a more thorough comparison.

TABLE III: Performance summaries over a total of 84 cases. The entry in the ith row and jth column shows how many times algorithm i was better than algorithm j.

	Normal	SD TPM	SD MPs	SD Both
Normal		4	2	3
SD TPM	73		70	4
SD MPs	52	7		5
SD Both	74	48	72	

Estimation

	Normal	SD TPM	SD MPs	SD Both
Normal		1	0	1
SD TPM	76		21	10
SD MPs	77	56		41
SD Both	76	67	36	

Prediction

1) State-Dependent Performance: The experiment contains four versions of the IMM algorithm. In addition to calling them "Normal," "SD TPM," "SD MPs," and "SD Both," we can refer to them as the first, second, third, or fourth algorithm, respectively. This is the order in which the algorithms were implemented as well as the order in which the results are displayed. Table III counts how many times algorithm i was better than algorithm j over the *eighty-four* cases.

The Estimation columns of Table II show the values of $\bar{e}_{e,i}$, and the Prediction columns show the values of $\bar{e}_{p,i}$ for a 40-second-duration trajectory. (The additional subscript refers to the choice of algorithm.) For every pair $i, j \in \{1, 2, 3, 4\}$, add one to the i, jth entry of the Estimation portion of Table III if $\bar{e}_{e,i} < \bar{e}_{e,j}$. The process is repeated again using $\bar{e}_{p,i}$ for Prediction. The elements on the diagonal show the total number of cases summarized in Table III.

Table III shows that the three state-dependent variations of the IMM algorithm very frequently perform better, on average, than the "Normal" algorithm both for estimation and for prediction. This can be seen in two ways. The entries in the first column that correspond to the state-dependent algorithms are very high, with the exception of "SD MPs" in the Estimation case, meaning that the state-dependent variations frequently perform better than the "Normal" version. A similar conclusion comes from the first row of each of the two tables; the "Normal" version is almost never better than the state-dependent versions. In Estimation, "SD MPs" is better than "Normal" 52 times, "Normal" is better than "SD MPs" only 2 times, and the remaining 30 times the two were equal (within five decimal places). Seven of the equivalent performances come from a ground truth in which there are no obstacles.

It is important to note that Table III does not give an indication of how much better the state-dependent algorithms performed. It merely counts how many times the state-dependent algorithms performed better by any amount.

2) State-Dependent Comparison: According to Table III, there are differences between using "SD MPs," "SD TPM," and "SD Both." The relative differences between the variations depend as much on the choice of the state-to-value mapping as they do on the methods of incorporating that information.

Table III shows that "SD TPM" performs better than "SD MPs" a majority of the time in estimation. Further, it shows that "SD Both" is at least as good as, if not better than, "SD TPM" in the majority of cases. Conversely, it shows that "SD MPs" often predicts better than "SD TPM." It also shows that "SD MPs" is roughly equivalent in prediction performance to "SD Both." The combination algorithm "SD Both" can take both the good parts and the bad parts of the individual variations "SD TPM" and "SD MPs," and thus the challenge becomes to differentiate between "SD TPM" and "SD MPs."

It seems that "SD TPM" is not as good as "SD MPs" for prediction, but it is better for estimation. Generally, the IMM algorithm at the $(k+1)$th time step has the ability to lessen the strength of the kth measurement because of the mixing step. The "SD TPM" algorithm has even more of that power because of the fact that the TPM can be very heavily modified, as in (7), before the mixing step takes place. We believe this causes the predictions of the "SD TPM" algorithm to be more wild than the predictions of the "SD MPs" algorithm. Even though the predictions of "SD TPM" are wild, they appear to be good mixing candidates once a new measurement arrives to tame them.

V. Summary and Conclusions

Three state-dependent variations of the IMM algorithm have been proposed and studied. State-dependent model probabilities, "SD MPs," embeds the world information into the mixing step of the IMM algorithm. State-dependent TPM, "SD TPM," incorporates the world information into the TPM before the mixing step of the IMM algorithm. Their combination, "SD Both," modifies both the model probabilities and the TPM.

Based on the set of ground truth trajectories and the simulation results, "SD TPM" seems to perform better than "SD MPs" for estimation but worse for prediction. The combination, "SD Both," often performs at least as well as the other two. All three of the variations always outperform the "Normal" algorithm. The estimation performance improvement is not always significant, but increases considerably as the measurement noise increases. The prediction performance is always clearly better than that of the "Normal" version.

The specific value function proposed in Section III.A (A State's Value) is relatively simple and not quite matched to the ground truth generator's rules. A better value function certainly would improve the tracking performance of the state-dependent

IMM algorithms. Regardless, the simple value function is enough to improve the performances of the state-dependent algorithms when compared to the standard IMM algorithm.

The value function must be tailored to a specific problem, but the method of incorporating the world information into the IMM algorithm is generally applicable. Specific performance details depend on both the problem and the choice of value function.

REFERENCES

[1] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part V: Multiple-Model Methods," *IEEE Trans. Aerospace and Electronic Systems*, vol. 41, pp. 1255–1321, Oct. 2005.

[2] X. R. Li, "Engineer's Guide to Variable-Structure Multiple-Model Estimation for Tracking," in *Multitarget-Multisensor Tracking: Applications and Advances* (Y. Bar-Shalom and W. D. Blair, eds.), vol. III, ch. 10, pp. 499–567, Boston, MA: Artech House, 2000.

[3] X. R. Li, "Hybrid Estimation Techniques," in *Control and Dynamic Systems: Advances in Theory and Applications* (C. T. Leondes, ed.), vol. 76, pp. 213–287, New York: Academic Press, 1996.

[4] H. A. P. Blom, "An Efficient Filter for Abruptly Changing Systems," in *Proc. 23rd IEEE Conf. Decision and Control*, (Las Vegas, NV), Dec. 1984.

[5] H. A. P. Blom and Y. Bar-Shalom, "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients," *IEEE Trans. Automatic Control*, vol. AC-33, pp. 780–783, Aug. 1988.

[6] R. Toledo-Moreo and M. A. Zamora-Izquierdo, "IMM-Based Lane-Change Prediction in Highways with Low-Cost GPS/INS," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 180–185, 2009.

[7] X. R. Li and Y. Bar-Shalom, "Multiple-Model Estimation with Variable Structure," *Automatic Control, IEEE Transactions on*, vol. 41, no. 4, pp. 478–493, 1996.

[8] X. R. Li, Y. Zhang, and X. Zhi, "Multiple-Model Estimation with Variable Structure: Model-Group Switching Algorithm," in *Proc. 36th IEEE Conference on Decision and Control,*, vol. 4, pp. 3114–3119, 1997.

[9] X. R. Li, X. Zhi, and Y. Zhang, "Multiple-Model Estimation with Variable Structure. III. Model-Group Switching Algorithm," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 35, no. 1, pp. 225–241, 1999.

[10] X. R. Li, "Multiple-Model Estimation with Variable Structure. II. Model-Set Adaptation," *Automatic Control, IEEE Transactions on*, vol. 45, no. 11, pp. 2047–2060, 2000.

[11] X. Wang, S. Challa, R. Evans, and X. R. Li, "Minimal Submodel-Set Algorithm for Maneuvering Target Tracking," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1218–1231, 2003.

[12] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground Target Tracking with Variable Structure IMM Estimator," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 36, no. 1, pp. 26–46, 2000.

[13] M. Zhang, S. Knedlik, and O. Loffeld, "An Adaptive Road-Constrained IMM Estimator for Ground Target Tracking in GSM Networks," in *Information Fusion, 2008 11th International Conference on*, pp. 1–8, 2008.

[14] S. Zhang and Y. Bar-Shalom, "Tracking Move-Stop-Move Targets with State-Dependent Mode Transition Probabilities," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 47, no. 3, pp. 2037–2054, 2011.

[15] I. Hwang and C. E. Seah, "An Estimation Algorithm for Stochastic Linear Hybrid Systems with Continuous-State-Dependent Mode Transitions," in *Decision and Control, 2006 45th IEEE Conference on*, pp. 131–136, 2006.

[16] C. E. Seah and I. Hwang, "State Estimation for Stochastic Linear Hybrid Systems with Continuous-State-Dependent Transitions: An IMM Approach," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, no. 1, pp. 376–392, 2009.

[17] X. R. Li and Y. Bar-Shalom, "Design of an Interacting Multiple Model Algorithm for Air Traffic Control Tracking," *Control Systems Technology, IEEE Transactions on*, vol. 1, no. 3, pp. 186–194, 1993.

[18] R. Rastgoufard, "The Interacting Multiple Models Algorithm with State-Dependent Value Assignment," Master's thesis, University of New Orleans, 2012.