Unifying Bayesian Networks and IMM Filtering for Improved Multiple Model Estimation

Robin Schubert
Professorship of Communications Engineering
Chemnitz University of Technology
Chemnitz, Germany
robin.schubert@etit.tu-chemnitz.de

Gerd Wanielik
Professorship of Communications Engineering
Chemnitz University of Technology
Chemnitz, Germany
gerd.wanielik@etit.tu-chemnitz.de

Abstract – Multiple model filtering has become an important concept for various applications such as maneuvering target tracking or road vehicle positioning. Algorithms like the Interacting Multiple Model (IMM) filter allow an adaptation of the filter bandwidth to different motion patterns of the target. In general, the individual probabilities of each model are derived from the estimation itself and the incorporated measurements. In this paper, an approach to exploit additional uncertain knowledge for multiple model filtering is presented. This method is modeling the additional information in a Meta Model using a Bayesian network. Thus, two important concepts of information fusion are unified to a holistic approach for multiple model filtering. The proposed method is demonstrated on the example of maneuver recognition for road vehicles.

Keywords: IMM, Bayesian Network, Meta Model Filter, Multiple Model Estimation

1 Introduction

Estimating the parameters of dynamic systems from uncertain observations is one of the most fundamental problems in information fusion. In general, specializations of the well-known Bayes filter (e.g. the Kalman, histogram, or particle filter) are used to perform this estimation based on appropriate models of the system dynamics and the sensor characteristics. While those methods show a reasonable performance if the system behavior corresponds at least to some extent to the chosen model, problems occur if the dynamic patterns of the systems are varying over time. Thus, multiple model estimators have been proposed in order to overcome this problem.

One of the most widely used algorithms for that purpose is the Interacting Multiple Model filter (IMM) proposed by Blom and Bar-Shalom [1]. The IMM – as well as multiple model estimation in general – has received considerable attention recently, as it provides a valuable concept for different applications such as maneuvering target tracking [2], ego positioning [3], or maneuver recognition [4, 5].

The main concept of the IMM (which will be described more detailed in section 2) is to estimate the system state using different models in parallel. In addition, the probabilities of each model are derived from the model-specific innovations, that is, the difference between the predicted and the observed measurements. This approach makes the IMM completely self-contained, which means that no additional information is necessary for the estimation. However, there are many situations in which additional knowledge is available. With the traditional IMM approach, such knowledge cannot be exploited for the estimation process (unless it can be regarded as additional observation provided by an appropriately modeled sensor).

In this paper, a method is proposed to incorporate additional, uncertain information into the multiple model estimation process. For that, a Meta Model Filter is presented which models the transitions between single system modes using a Bayesian network. With this approach, uncertainties concerning both the additional information itself and the meta model structure can be treated in a statistically sound way. Altogether, a generalization of the IMM is proposed which unifies two important concepts of information fusion: Bayesian networks and the multiple model filtering.

Some related ideas for extending the classical IMM have been proposed by different authors recently: In [6], an IMM implementation for ground target tracking is presented using variable model sets, which depend on additional information about the road. However, these additional information are influencing the choice of possible models only, but not the transition between them. In [7], a Bayesian network is used to incorporate additional measurements which are called attributes. However, while those attributes are used for data association and track identification purposes, they do not influence the tracking directly. Furthermore, an approach which combines multiple model filtering and classification in a Dynamic Bayesian Network is presented in [8]. However, as this results in a hybrid system (containing both discrete and continuous random variables), it can only be solved with high effort using sequential Monte Carlo methods.

This paper is organized as follows: in sections 2 and 3, the fundamentals of multiple model filtering and Bayesian networks are briefly discussed. The proposed Meta Model Filter is described in section 4. After that, the presented method is demonstrated on the example of road vehicle lane change maneuver recognition in section 5. Finally, the paper concludes with section 6.
2 Multiple Model Filtering

The main idea of multiple model filtering is to run several filters in parallel, each of them representing a separate model of the system under consideration [9]. With \( x_k \) and \( P_k \) denoting the system state’s mean value and covariance matrix at time \( k \) and \( Z_k \) representing all observations up to \( k \), the desired posterior probability density \( p(x_k | Z_k) \) can be approximated as a Gaussian mixture of the different model-matched filters, that is:

\[
p(x_k | Z_k) \approx \sum_{i=1}^{N} w_k^i \cdot N(x_k, x_k^i, P_k^i),
\]

where the weights \( w_k^i \) sum up to unity.

Depending on the concrete procedure for determining the weights, different multiple model (MM) estimators can be defined. The static MM estimator assumes that the system is following one constant model which is unknown to the estimator. In contrast, dynamic MM estimators explicitly take into account the possibility that the system may change its model over time. Under this category, it is possible to add and prune mixture components in each estimation step (as for instance done by the Multiple Model Pruning estimator [10]) or to merge a constant set of components. The most popular implementation of the latter approach is the Interacting Multiple Model filter (IMM), which will be described in the following.

The IMM models the system state as a Hidden Markov Model (HMM) with \( N \) mode states (compare figure 1). The model is hidden because the mode states cannot directly be observed. Instead, every mode is producing an uncertain system state prediction \( x_{k|i|k-1}^i \) by a stochastic process (described by the system model). In addition, every system state produces predicted observations \( y_{k|i|k-1}^i \) by another stochastic process (described by the observation model). The task is to derive the state of the Markov model from the observations. As the Kalman Filter already estimates the system state from the observations, the task reduces to determining the model from the system state vector and its covariance.

The transition probabilities of the Markov model are described by the Markovian matrix II:

\[
\Pi = \begin{pmatrix}
\pi_{11} & \cdots & \pi_{1N} \\
\vdots & \ddots & \vdots \\
\pi_{N1} & \cdots & \pi_{NN}
\end{pmatrix}
\]

In the Interacting Multiple Model (IMM) framework, a separate filter is running for each of the \( N \) models with a predefined initial model probability \( \mu_k^j | k = 0 \) (\( j = 1 \ldots N \)). Every time before the Kalman filter equations are applied, this probability has to be updated using the transition probabilities defined in \( \Pi \):

\[
\mu_{k-1}^{ij} = \frac{\pi_{ij} \mu_{k-1}^j}{\sum_{\ell=1}^{N} \pi_{i\ell} \mu_{k-1}^\ell}
\]

In the interaction process, those probabilities are used to update the state vectors of the filters and their covariances. Thus, despite of the filters running in parallel, they are not independent from each other. The results of the interaction process are calculated as follows:

\[
x_k^j = \sum_{i=1}^{N} \mu_{k-1}^{ij} x_{k-1}^i
\]

\[
P_{k-1}^j = \sum_{i=1}^{N} \mu_{k-1}^{ij} P_{k-1}^i + \left( P_{k-1}^i + (x_{k-1}^i - x_{k-1}^j)(x_{k-1}^i - x_{k-1}^j)^T \right)
\]

In the next step, the standard Kalman filter equations can be applied. After the correction step, the model probabilities need to be updated. For that, a measure is necessary which evaluates the conditional likelihood of each filter. This problem is comparable to the well-known data association problem where one observation could possibly belong to different filters. A common solution for that is to calculate the normalized Mahalanobis-distance [11] between the predicted observation \( y_{k|i|k-1}^j \) and the real measurement \( y_k \).

Thus, both the residuals of each filter and their covariances are taken into account. Under a Gaussian assumption, the model probability can be calculated as follows:

\[
\eta^j = \frac{1}{\sqrt{(2\pi)^n |S_k^j|}} e^{-\frac{1}{2} (v_k^j)^T (S_k^j)^{-1} (v_k^j)}
\]

\[
\mu_k^j = \frac{\mu_{k|i|k-1}^j \eta^j}{\sum_{i=1}^{N} \mu_{k|i|k-1}^i \eta^i}
\]

where \( v^j \) and \( S^j \) denote the model-specific innovation and its covariance. Finally, the output of the tracking system is

\[\text{Figure 1: A Hidden Markov Model describing a system with two possible modes which are producing system state predictions and predicted observations by stochastic processes.}\]
calculated using the probability-weighted sum of all filters:

\[ x_k = \sum_{i=1}^{N} \mu_i^k x_k^i \]  

(8)

\[ P_k = \sum_{i=1}^{N} \mu_i^k (P_k^i + (x_k - x_k^i)(x_k - x_k^i)^T). \]

(9)

### 3 Bayesian networks

A Bayesian network (BN) is a graphical representation of a set of random variables [12]. It consists of nodes which represent random variables and edges which represent probabilistic relationships (and, generally speaking, causality). For instance, in figure 2, the node pairs A&B, B&D, and C&D are not conditionally independent, whereas pairs such as A&C show that property.

The main advantage of Bayesian networks compared to other representations (for instance joint probability tables) is the possibility to model conditional independence, which drastically decreases the size of the necessary conditional probability tables (CPT). Furthermore, the graphical modeling makes it easier for a human designer to adapt the model to his perception of the system characteristics. At the same time, the concept of BN is formalized enough to allow efficient computational evaluation.

After defining the structure of the network and the CPTs, full joint distributions may be calculated for any particular assignment to the random variables of the network. That is, given that the number of random variables (denoted by \( X_i \)) in the network is \( N \), the joint probability that they are assigned to a certain value \( x_i \) is given by

\[ p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i|pa(X_i)), \]

(10)

where \( pa(X_i) \) represents the set of parent nodes of \( X_i \) (that is, all nodes that posses a directed edge to \( X_i \)).

The main feature of Bayesian networks is the ability to perform probabilistic reasoning. That is, knowledge about the state of certain nodes (which is generally called evidence) can be incorporated into the network for the purpose of calculating the probability distributions of all other nodes under the condition of the entered evidence. The reasoning can be performed regardless of the defined causality, that is, it is possible to deduce possible reasons for observed effects or vice versa. Uncertain evidence (so-called likelihood evidence) can also be modeled inside the network.

BN can be regarded as probabilistic extensions of expert systems and allow a statistically sound combination of uncertain knowledge, taking into account both model and sensor uncertainties. While for small networks reasoning can be performed in a simple way using equation 10, efficient algorithms exist in order to allow real-time operations for larger models (compare for instance [13]).

![Figure 2: Example of a Bayesian network. As the states B and D are not directly connected, conditional independence among them is assumed.](image)

### 4 The Meta Model Filter

The main idea of this paper is to establish an adaptive model transition probability matrix \( \Pi_k \) which depends on additional knowledge on object or situation level. For that, it is proposed to construct a model of the different possible modes and their transitions. As this means modeling other models on a higher abstraction level, the term Meta Model will be used in the following.

In order to obtain a statistically sound representation of this Meta Model, it is proposed to construct it in a probabilistic way using a Bayesian network. While the concrete implementation obviously depends on the application, it is nevertheless possible to define a general structure of the Meta Model which is illustrated in figure 3.

The core of this generic Meta Model are \( N \) nodes that represent the possible nodes of the system. For each of these nodes, \( N \) states are defined which represent all possible transitions to other modes. That is, each element of \( \Pi \) is represented by one single state of a certain node. For instance, the following transition probability matrix can be derived from the exemplary probability distribution illustrated in figure 3:

\[ \Pi = \begin{pmatrix}
0.9 & 0.05 & 0.05 \\
0.05 & 0.9 & 0.05 \\
0.05 & 0.05 & 0.9
\end{pmatrix} \]

(11)

These mode nodes are the core of every Meta Model. Furthermore, additional nodes can be specified which are statistically related to the mode nodes. Taking into account the implications of causality, these nodes can either be a possible cause or a possible effect of the current mode transition. Thus, they are summarized in the Generic Meta Model shown in figure 3 to two subnets which may contain an arbitrary number of individual nodes.

In order to emphasize the fact that knowledge about these additional nodes will in general be obtained by imperfect data sources, two additional subnets are included in the Generic Meta Model that contain the observations, which are connected to the cause and effect subnets via likelihood functions (representing appropriate sensor models).

Having specified a concrete Meta Model for the targeted application, the Meta Model Filter algorithm is proposed as follows:
Figure 3: A Generic Meta Model for incorporating additional information into the multiple model filtering process.

1. Enter all available evidences of time step $k$ into the observation subnets of the Meta Model.

2. Perform probabilistic reasoning within the Bayesian network using equation 10 or appropriate sophisticated algorithms.

3. Derive $\Pi_k$ from the network as shown in equation 11.

4. Perform the IMM algorithm described in section 2 using $\Pi_k$ as mode transition matrix.

This approach yields several advantages:

- It provides an extension of the IMM and can be applied in addition to existing IMM implementations.

- The current mode probability does still depend on the observed behavior of the system (represented by the innovations), but is supported by additional knowledge.

- The computational load of the presented approach is rather limited and allows real-time performance.

- The method provides a generalization which can be adapted to various implementations.

- Learning algorithms can be applied to the Meta Model based on training data.

Furthermore, the MMF can be regarded as a general tool for the widely discussed Multi Level Fusion [14] which aims (among others) to exploit situational awareness for object tracking, as it combines two important concepts from object and situation level into a unified algorithm.

5 Application & Results

In order to demonstrate the usage of the presented Meta Model Filter, an example implementation is presented in this section. The problem is to recognize a lane change maneuver of a vehicle traveling on a highway from uncertain velocity and yaw rate measurements. The basic approach to solve this problem using an IMM has been presented already in [4] and will be repeated here for convenience. Furthermore, the extensions of the IMM-based approach and especially the MMF solution will be presented.

The main idea of the approach is to represent the maneuvers Lane Keeping, Lane Change Left, and Lane Change Right by different motion models. The mode with the highest probability can then be regarded as the current maneuver. For lane keeping, a Constant Turn Rate and Acceleration (CTRA) model\(^2\) is applied, that is, the system model can be described by

$$
\ddot{x}(t+T) = \dot{x}(t) + \begin{pmatrix}
\Delta x(T) \\
\Delta y(T) \\
\Delta \theta(T) \\
\Delta v(T)
\end{pmatrix},
$$

(12)

with

$$
\Delta x(T) = \frac{1}{\omega^2} \left[ (v(t) \omega + a \omega T) \sin(\theta(t) + \omega T) + a \cos(\theta(t) + \omega T) \\
- v(t) \omega \sin \theta(t) - a \cos \theta(t) \right]
$$

(13)

\(^2\)Details about appropriate motion models for road vehicles can for instance be found in [15] and [16].
and

\[ \Delta y(T) = \frac{1}{\alpha} \left[ (-v(t)\omega - a\omega T) \cos(\theta(t) + \omega T) + a\sin(\theta(t) + \omega T) + v(t)\omega \cos(\theta(t) - a\sin(\theta(t))) \right] \]  

(14)

For the lane change maneuvers, a constant, pre-defined yaw acceleration of \( \alpha = \pm 0.06 \text{ rad/s}^2 \) is assumed. That is, the applied system model is

\[ \bar{x}(t+T) = \begin{pmatrix} x(t+T) \\ y(t+T) \\ \theta(t+T) \\ v(t+T) \end{pmatrix} = \bar{x}(t) + \begin{pmatrix} \Delta x(T) \\ \Delta y(T) \\ \omega T \\ aT \end{pmatrix} \]  

(15)

For the filtering itself, the Unscented Kalman Filter [17] is applied. As proposed in [18], the state space is augmented by error variables in order to incorporate the process noise.

In order to further increase the performance of the system, additional knowledge from the lane recognition system is incorporated into the filter using the proposed MMF approach. In particular, the lane recognition system is able to classify lane markings into the types dashed and solid. This knowledge may be considered useful for the tracking process as in the presence of solid lane markings (which indicate a lane change prohibition in many countries) the probability of a lane change maneuver is assumed to be significantly lower than in the contrary case. Thus, the Meta Model illustrated in figure 4 has been constructed.

In figure 4(a), the initial Meta Model is shown (the conditional probability tables are given in appendix A). In this model, the causal subnet consists of two nodes representing the type of lane markings on the left and right side of the vehicle. As this information is derived from an imperfect classifier, a sensor model is used in order to cope with classification errors. Furthermore, this model also accounts for the possibility that the classifier cannot provide a result at all (state Unknown).

Figure 4(b) shows the transition probabilities for the case that the type of both lane markings is known to be dashed. It is shown that in that case the probabilities for keeping or changing the mode are equal for every mode node. Thus, this situation corresponds to the classical IMM approach.

In figures 4(c) and 4(d), the influence of the classification result on the transition probabilities is demonstrated. For instance, in the case of a dashed/solid lane marking on the left/right side, the probability for a mode transition from Lane Keeping to Lane Change Left is much higher than the one from Lane Keeping to Lane Change Right. However, the latter probability is still not zero in order to cope with classification errors and unconventional driving behavior.

In summary, the Meta Model represents typical driving behavior observations and traffic regularities in an uncertain way. However, the model is obviously only a very simple example in order to illustrate the Meta Model concept. For the presented application, it is also imaginable to incorporate other data sources (e.g. digital maps) or model additional rules (for instance, a vehicle will only rarely initiate a lane change maneuver while another vehicle is driving in its lateral vicinity). Additionally, due to the Bayesian network structure of the MM, it is possible to apply learning algorithms [12] in order to automatically learn the structure or the conditional probability tables of the network (for instance based on traffic surveillance data). Thus, the MMF

---

3This is a theoretical case which is used just for illustration purposes, as such certain knowledge will never be available in any practical system.
approach yields an enormous additional potential beyond the scope of the provided example.

For evaluating the MMF’s benefits, real test data have been recorded using the concept vehicle Carai of Chemnitz University of Technology (compare figure 5). In particular, velocity and yaw rate measurements as well as grayscale images with VGA resolution have been used. In order to obtain reference data about the true maneuver of the vehicle, a lane recognition algorithm was applied on the camera images. The algorithm is capable of estimating the position and course of the lane relative to the ego vehicle. Thus, information about the exact time of a lane change can be provided, where a lane change is defined as the moment in which the center of the vehicle’s rear axis is passing a lane border.

Figure 6 shows the resulting model probabilities of all three models for both the classical IMM and the MMF approach during an exemplary time period. It illustrates that both approaches are capable of recognizing the start of the lane change maneuver (approximately 2.8 s before the passing of the lane border). However, it turns out that the knowledge about the right lane markings’ type (which are solid during this period) suppresses the lane change right model probability. The same holds for the lane change left model probability after the lane change (where the left lane markings are solid).

In addition, the figure illustrates a false alarm of the IMM at time 17.86 s which is caused by the beginning of a left curve. However, the MMF suppresses this false alarm successfully – even though the effect of the curve is still visible. The evaluation of the complete data set (around 30 km of driving distance) showed that the suppression of false alarms is one of the main benefits in the presented example.4 Furthermore, the tests confirmed the real-time capability of the proposed approach. In fact, the additional computational effort of the Bayesian network evaluation turned out to be almost negligible compared to the filtering itself.

4It also turned out that the MMF recognized the lane change maneuvers slightly earlier, however, this effect is nearly negligible.

6 Conclusions & Future Work

In this paper, an algorithm for exploiting additional, uncertain knowledge for multiple model tracking purposes has been presented. In particular, a Generic Meta Model structure has been proposed in order to unify Bayesian networks and the Interacting Multiple Model Filter into a holistic framework for MM filtering. The approach extends the classical IMM without the need to change the IMM algorithm itself. Thus, it can be regarded as a generalization of Multiple Model estimation taking into account situational awareness.

The utilization of the Meta Model Filter has been illustrated on the example of a lane change maneuver recognition system, where knowledge about the type of lane markings was incorporated into the filter. It has been shown how this additional knowledge influences the mode transition probabilities of the MMF and, thus, the filtering itself.

Future work will be carried out in different directions: Taking into account the generality of the presented approach, one task will be to exploit this algorithm for additional applications such as ego vehicle positioning or pedestrian recognition. Furthermore, more complex models for the illustrated example will be developed which incorporate additional knowledge from different data sources. Finally, quantitative studies will be conducted in order to evaluate the benefits and performance gains provided by the Meta Model Filter.

A Conditional Probability Tables

In the following, the conditional probability tables used for the example implementation from section 5 will be given:

1. Observation Border Left:

<table>
<thead>
<tr>
<th></th>
<th>Dashed</th>
<th>Solid</th>
<th>Unknown</th>
<th>BorderLeft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashed</td>
<td>0.99</td>
<td>0.002</td>
<td>0.008</td>
<td>Dashed</td>
</tr>
<tr>
<td>Solid</td>
<td>0.002</td>
<td>0.99</td>
<td>0.008</td>
<td>Solid</td>
</tr>
</tbody>
</table>

2. Observation Border Right:

<table>
<thead>
<tr>
<th></th>
<th>Dashed</th>
<th>Solid</th>
<th>Unknown</th>
<th>BorderRight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashed</td>
<td>0.99</td>
<td>0.002</td>
<td>0.008</td>
<td>Dashed</td>
</tr>
<tr>
<td>Solid</td>
<td>0.002</td>
<td>0.99</td>
<td>0.008</td>
<td>Solid</td>
</tr>
</tbody>
</table>

3. Type of left border:

<table>
<thead>
<tr>
<th></th>
<th>Dashed</th>
<th>Solid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashed</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Solid</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

4. Lane Change Left:

<table>
<thead>
<tr>
<th></th>
<th>Dashed</th>
<th>Solid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dashed</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

5. Lane Change Left:
Figure 6: Model probabilities for both the IMM and MMF algorithm.

<table>
<thead>
<tr>
<th></th>
<th>LC Left</th>
<th>KL</th>
<th>LC Right</th>
<th>B Left</th>
<th>B Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMM</td>
<td>0.96</td>
<td>0.03</td>
<td>0.01</td>
<td>Dashed</td>
<td>Dashed</td>
</tr>
<tr>
<td></td>
<td>0.97</td>
<td>0.03</td>
<td>0.0</td>
<td>Solid</td>
<td>Solid</td>
</tr>
<tr>
<td>MMF</td>
<td>0.96</td>
<td>0.03</td>
<td>0.01</td>
<td>Solid</td>
<td>Solid</td>
</tr>
</tbody>
</table>

6. Keep Lane:

<table>
<thead>
<tr>
<th></th>
<th>LC Left</th>
<th>KL</th>
<th>LC Right</th>
<th>B Left</th>
<th>B Right</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.02</td>
<td>0.96</td>
<td>0.02</td>
<td>Dashed</td>
<td>Dashed</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
<td>0.98</td>
<td>0.0</td>
<td>Dashed</td>
<td>Solid</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.98</td>
<td>0.02</td>
<td>Solid</td>
<td>Dashed</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>Solid</td>
<td>Solid</td>
</tr>
</tbody>
</table>

7. Lane Change Right:

<table>
<thead>
<tr>
<th></th>
<th>LC Left</th>
<th>KL</th>
<th>LC Right</th>
<th>B Left</th>
<th>B Right</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.01</td>
<td>0.03</td>
<td>0.96</td>
<td>Dashed</td>
<td>Dashed</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.03</td>
<td>0.97</td>
<td>Dashed</td>
<td>Solid</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>0.03</td>
<td>0.96</td>
<td>Solid</td>
<td>Dashed</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.03</td>
<td>0.97</td>
<td>Solid</td>
<td>Solid</td>
</tr>
</tbody>
</table>

References


