Multiple model estimation: A convex model formulation

R. Hallouzi¹, *, †, M. Verhaegen¹ and S. Kanev²

¹Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628CD, Delft, The Netherlands
²ECN Wind Energy, P.O. Box 1, 1755ZG Petten, The Netherlands

SUMMARY
The multiple model (MM) framework provides an elegant solution to adaptive filtering problems. An important issue in the MM framework is how the estimation is performed. In this paper, a brief overview is given of the mainstream methods for MM estimation and a new method is proposed. Contrary to existing methods that mostly adopt a hybrid model structure, the newly proposed method uses a more general MM framework that allows for weighted combinations of the local models. The main advantage of this framework is that it has better model interpolation properties. These improved properties allow for smaller model sets, which are very useful in, for example, fault detection and identification (FDI) of partial faults. The improved interpolation properties are demonstrated by means of two simulation examples, one in which an FDI problem is addressed, and one in which a target tracking problem is addressed. Monte Carlo simulation results of these two examples are given. In these simulations, the well-known interacting IMM filter is compared with two estimation algorithms based on the proposed model structure. Copyright © 2008 John Wiley & Sons, Ltd.

Received 16 October 2006; Revised 23 April 2007; Accepted 22 January 2008

KEY WORDS: multiple model systems; model interpolation; joint parameter and state estimation; jump Markov linear systems

1. INTRODUCTION

Research on the multiple model (MM) approach has attracted considerable interest in the last decades. The reason for this is the elegant solutions that the MM approach provides for estimation, control and modeling problems [1–3]. A well-studied example of the application of MM to estimation is the target tracking problem. In this problem, the local models usually correspond to kinematic modes, such as straight flight and coordinated turns of the target. An elaborate explanation

*Correspondence to: R. Hallouzi, Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628CD Delft, The Netherlands.
†E-mail: r.hallouzi@tudelft.nl

Contract/grant sponsor: Dutch Technology Foundation (STW); contract/grant number: DEL.4506

Copyright © 2008 John Wiley & Sons, Ltd.
of different MM algorithms applied to target tracking problems can be found in [4]. Another important estimation application of the MM framework is fault detection and identification (FDI).

Different research efforts have been made in the field of FDI with MM systems, see, e.g. [5–10]. The main motivation for using the MM framework for FDI is that it allows a large class of fault conditions to be modeled. The reason for this is that, in principle, each of the local models might have totally different dynamics. Therefore, from an FDI perspective, the MM framework allows the modeling of actuator, sensor and component faults. The basic idea of performing FDI with MM systems is as follows: a model set must be created that contains local models corresponding to different fault conditions of the monitored system. In addition to the fault models, the model set usually includes the nominal model. Faults are identified by estimating which of the local models is valid using MM estimation algorithms. When there are no faults present in the monitored system, the nominal model will be valid. In case of a fault, one of the other models in the model set will become valid.

Most of the existing MM estimation algorithms provide a solution to the problem of estimating the state and the mode of a jump Markov linear system (JMLS). Numerous solutions are reported for this estimation problem ranging from particle filters [11] to the well-known interacting MM (IMM) filter [12]. A thorough overview of these different solutions is provided in [13]. The underlying model structure of a JMLS is hybrid. This means that it consists of a number of local models that do not interact with each other. Interaction between the different models can be added by the MM estimation algorithms themselves. However, it is important to note that this interaction is not inherent to the model structure itself. Both the MM estimation algorithms that do not have interaction between the models and the ones that do, display deteriorated performance in case the model set does not contain a model corresponding to the true system. This is the result of the assumption that the model corresponding to the true system should be in the model set [13].

In case the weighted combinations of the models in the JMLS model set correspond to physically relevant conditions, it is desirable to interpolate between models. For example, in [14] the MM adaptive estimation (MMAE) algorithm, which is also based on JMLS, is used to identify partial actuator faults. For this purpose, model sets are used that contain models of the same fault with different sizes in order to be close to the true system in case of a fault with an arbitrary magnitude. This indicates that the MMAE is not able to interpolate well between models. Otherwise, the partial faults could have been modeled by a weighted combination of only the nominal model and the total fault model. Another example in which the poor interpolation properties of the MM methods based on JMLS are recognized is [15]. In this reference, an extra feature is added to the IMM filter for identifying partial faults. This feature introduces model sets with a finer parameterization (which means a larger model set) after the detection of the fault.

A possible remedy for the poor interpolation properties of MM algorithms based on JMLSs is to use another model structure that does explicitly interpolate between local models. Such a structure is the blended MM structure [16]. In this structure, the model that is valid at a certain time is a weighted combination of the local models in the model set. When the combinations of the local models are restricted to be convex, a subset of the blended MM structure is created that is named the convex model (CM) structure. The convexity restriction is added because only convex combinations of the local models, which also include hybrid combinations (i.e. one of the local models is fully valid and the rest is not), correspond to physically relevant conditions. The MM estimation using the CM structure entails estimating both the state and the model weights of the local models. This estimation problem is nonlinear due to products of the state and the model weights.
MULTIPLE MODEL ESTIMATION: A CONVEX MODEL FORMULATION

The main contribution of this paper is to present the CM structure as an alternative to the hybrid model structure. The CM structure is an improvement to the hybrid model structure in the following ways:

1. The CM structure has better model interpolation properties because it explicitly allows for interpolations of local models. Better interpolation properties are very desirable when the weighted combinations of local models also correspond to physically relevant conditions. This is the case, for example, in partial fault modeling in FDI problems. In this case, having better interpolation properties allows for smaller model sets.

2. The CM structure does not use a transition probability matrix. This matrix contains the transition probabilities between the different local models in a JMLS. In theory, this matrix is usually assumed to be known for MM estimation in a JMLS. However, in practice, the transition probability matrix is considered to be a ‘design parameter’ due to insufficient information. Since this parameter can be difficult to design in practice, recent methods have been proposed for online estimation of the transition probability matrix [17]. Instead of using the transition probability matrix, MM estimation methods based on the CM structure rely more on measured data and less on a priori information (i.e. the transition probability matrix).

The IMM filter is chosen in this paper as the representative filter for MM estimation with JMLSs because it is widely accepted for this purpose and because of its simplicity. For the nonlinear estimation problem related to the CM structure, two approximate filters are used. One filter solves the problem in two linear filtering steps using dual filtering methods [18] and the other filter uses the augmented extended Kalman filter (EKF) [19], which uses linearization. These two straightforward filters are chosen to allow an honest comparison with the IMM filter. A comparison between the three filters is presented based on an FDI problem and a target tracking problem.

This paper is organized as follows. First, the JMLS and the CM will be described in Section 2 together with the estimation objectives of the MM estimation algorithms based on these two structures. Subsequently, in Section 3 the MM estimation algorithms themselves are described. Section 4 provides two Monte Carlo simulation examples that have the purpose to demonstrate the pros and cons of the conventional and newly proposed model structure. Finally, Section 5 will end this paper with some concluding remarks.

2. PROBLEM FORMULATION

Consider the following linear time-varying system:

\[ x_{k+1} = A(f_k)x_k + B(f_k)u_k + Q(f_k)^{1/2}w_k \]  
\[ y_k = C(f_k)x_k + D(f_k)u_k + R(f_k)^{1/2}v_k \]  

where \( x_k \in \mathbb{R}^n \) is the state, \( u_k \in \mathbb{R}^m \) is the input, \( y_k \in \mathbb{R}^\ell \) is the output, \( w_k \in \mathbb{R}^n \) is the process noise and \( v_k \in \mathbb{R}^\ell \) is the measurement noise. Both \( v_k \) and \( w_k \) are assumed to be zero-mean white noise sequences with unit variance. \( A(f_k), B(f_k), C(f_k) \) and \( D(f_k) \) are the system matrices that depend on parameter \( f_k \). \( Q(f_k) \) and \( R(f_k) \) are noise covariance matrices that also depend on \( f_k \). The parameter \( f_k \) can take values in the bounded set \( \mathcal{F} \). Although the set \( \mathcal{F} \) is bounded, the parameter \( f_k \) can have infinitely many values. Let the infinite set of models defined by (1)–(2)
for all \( f_k \in \mathcal{F} \) be denoted by \( \mathcal{M} \) and let \( \mathcal{M} \) be represented by the shaded area in Figure 1. The goal of MM methods is to approximate \( \mathcal{M} \) by as little models as possible. The JMLS is often used for approximating \( \mathcal{M} \). The local models used by the JMLS to approximate \( \mathcal{M} \) can be chosen heuristically or according to certain design methods \[2\]. In Figure 1, the JMLS model set chosen to approximate the shaded area contains six models. These models are represented by stars and denoted by \( \mathcal{M}(1) - \mathcal{M}(6) \). Furthermore, model \( \mathcal{M}(c) \in \mathcal{M} \) is also depicted. If this model corresponds to the true system, then MM estimation algorithms based on the JMLS with model set \( \mathcal{M}(1) - \mathcal{M}(6) \) perform less well. In order to maintain performance in this case, \( \mathcal{M}(c) \) should be added to the existing model set. This principle can lead to large model sets in practice, which is not desirable because of the increased computational cost. However, it is possible to represent \( \mathcal{M}(c) \) by a convex combination of \( \mathcal{M}(1) - \mathcal{M}(6) \). If an alternative model structure is adopted that allows convex combinations of models, then all models from \( \mathcal{M} \) could be represented with the same model set as the JMLS in this case. In this section, first the JMLS and the estimation objectives related to it will be described. Subsequently, an alternative structure that allows convex combinations of models is proposed. Also for this alternative structure estimation objectives are given.

### 2.1. Hybrid model structure

The MM system that is commonly adopted for MM estimation algorithms has a hybrid model structure and is given by

\[
x_{k+1} = A(s_k)x_k + B(s_k)u_k + Q(s_k)^{1/2}w_k \\
y_k = C(s_k)x_k + D(s_k)u_k + R(s_{k+1})^{1/2}v_k
\]

where \( A(\cdot), B(\cdot), C(\cdot), D(\cdot), Q(\cdot) \) and \( R(\cdot) \) are matrices, which are functions of the first-order homogenous Markov chain \( s_k \), referred to as the system mode. \( w_k \) and \( v_k \) represent unit variance white noise sequences with means \( w_k \) and \( v_k \), respectively. \( Q(s_k) \) and \( R(s_{k+1}) \) are noise covariance matrices. The system mode \( s_k \) has \( N \) states and the entries of the transition probability matrix

---

**Figure 1.** Schematic diagram of a multiple model system.
MULTIPLE MODEL ESTIMATION: A CONVEX MODEL FORMULATION

\(\Pi \in \mathbb{R}^{N \times N}\) are given by

\[\Pi_{ij} = P(s_{k+1} = j | s_k = i)\]  \hspace{1cm} (5)

where \(i, j \in S\), with \(S = \{1, 2, \ldots, N\}\). The system defined by (3)–(4) combined with (5) is known as a JMLS.

The estimation objective of most MM estimation algorithms based on a JMLS is to obtain minimum mean square error (MMSE) estimates of both the state and the mode of the JMLS [11, 13]. There are also MM estimation algorithms that use the maximum a posteriori (MAP) criterion. For example, in [20] a non-recursive algorithm for MAP estimation is proposed followed up by a recursive MAP estimation algorithm in [21]. In this paper, the MMSE estimate of the state is considered. This estimate is given by [4]

\[\hat{x}_{k|k} = E[x_k | Y_k] = N_k \sum_{i=1}^{N_k} E[x_k | M^{(k,n)}, Y_k] P(M^{(k,n)} | Y_k)\]  \hspace{1cm} (6)

where \(Y_k = [y_k \ y_{k-1} \ldots \ y_0]\) and \(M^{(k,n)}\) denotes the \(n\)th possible mode sequence of \(N_k\) sequences at time instant \(k\). It can be seen that evaluating (6) is a very computationally intensive task due to its exponential complexity. Therefore, most MM estimation algorithms apply approximations to deal with this complexity. Different strategies exist for making such approximations resulting in different filters. These strategies differ in the way in which all possible mode scenarios in the past are used for estimating the mode at the current time step. An elaborate explanation of these different strategies is given in [13]. In this paper, the approximation of the IMM filter for obtaining the MMSE estimate of the state is considered. This approximation is given by

\[\hat{x}_{k|k} = E[x_k | Y_k] \approx N \sum_{i=1}^N E[x_k | s_k = i, Y_k] \mu_k^{(i)}\]  \hspace{1cm} (7)

where \(\mu_k^{(i)} = P(s_k = i | Y_k)\) is the ‘model weight’ of the \(i\)th model, which is computed in the course of computing \(\hat{x}_{k|k}\) [4]. The full weight vector \(\mu_k\) is composed of the individual model weights as \(\mu_k = [\mu_k^{(1)}\ \mu_k^{(2)}\ \ldots\ \mu_k^{(N)}]^T\). Note that the complexity of (7) is drastically less than the complexity of (6). This complexity reduction has been obtained by not considering all possible mode sequences in the past (as was the case in (6)) but instead considering only each possible current model based on the information from the previous time step.

2.2. CM structure

An alternative to the hybrid system defined in (3)–(5) is given by

\[x_{k+1} = \sum_{i=1}^N \mu_k^{(i)} [A^{(i)} x_k + B^{(i)} u_k] + Q_k^{1/2} w_k\]  \hspace{1cm} (8)

\[y_k = \sum_{i=1}^N \mu_k^{(i)} [C^{(i)} x_k + D^{(i)} u_k] + R_k^{1/2} v_k\]  \hspace{1cm} (9)

\[\mu_k^{(i)} \geq 0, \quad \sum_{i=1}^N \mu_k^{(i)} = 1\]  \hspace{1cm} (10)
where \( \mu_k^{(i)} \in [0, 1] \) is the model weight corresponding to the \( i \)th model represented by the state-space quadruple \( \{A^{(i)}, B^{(i)}, C^{(i)}, D^{(i)}\} \) and \( N \) is the number of local models. The model structure defined by (8)–(10) is named the CM structure because of the convex combination of the local models in this structure. The CM structure explicitly uses weighted combinations of the local models contrary to the hybrid model structure. Also note that the CM structure allows modeling of the same systems as the JMLS.

If the constraints on the model weights (10) are removed, a model structure results that is able to model an even larger class of models. However, when the same type of models are used as for the JMLS (i.e. each local model corresponds to a physical condition of the system), non-convex combinations of these local models correspond to models that do not have a physical meaning. Therefore, these constraints are explicitly included.

The estimation objective for MM estimation using the CM structure is defined in the following optimization problem:

\[
(\hat{\mu}_k, \hat{x}_{k|k}) = \arg \max_{\mu_k, x_k} \mathcal{P}(\mu_k, x_k | y_k)
\]

which is a MAP estimation approach. However, because the considered noise sequences in the CM structure have a Gaussian distribution, the MAP estimate also corresponds to the MMSE estimate. The reason for this is that the point at which the Gaussian distribution has maximum probability (location of the MAP optimum) corresponds to its center of probability mass (location of the MMSE optimum).

3. MM ESTIMATION ALGORITHMS

In this section, three algorithms for MM estimation will be described. The IMM filter is probably the most popular solution for the problem of estimating the state and mode of a JMLS. Although there are many variants of the IMM filter, each of which is aimed at improving performance in specific (practical) situations, the main core of these variants is the same. Therefore, the basic version of the IMM filter is considered as the benchmark algorithm for MM estimation using a JMLS and it will be summarized in this paper. For the CM structure, two MM estimation algorithms will be described that provide an approximate solution for (11). These filters differ in the sequence of estimation of the state and the weights. The EKF estimates the state and weights at the same time, whereas the dual CM filter (CMF) estimates the state and weights separately in two steps.

3.1. IMM

The IMM filter belongs to the second category of a set of three categories [13]. It was preceded by the autonomous MM (AMM) filter in the first category. The third category is the category of the variable structure MM filter. The AMM filter is characterized by the fact that there is no interaction between the different filters, each of which is based on a different local model from the model set. Interaction between filters was introduced in the second category, which explains the name of the IMM filter. In addition to this, the third category allows for a time-varying model set. This means that the model set can contain a different number of models at different times.
The IMM filter consists of four steps that are performed each cycle of the filter to compute (7). These four steps are [13] as follows:

1. model-conditioned re-initialization;
2. model-conditioned filtering;
3. model probability update;
4. fusion of estimates.

In Table I, the different steps that are performed in one cycle of the IMM filter are described. For convenience, throughout this table the notation $A(s_k = j) = A_k^{(j)}$ is used for the system matrices.

<table>
<thead>
<tr>
<th>Table I. One cycle of the IMM filter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Model-conditioned re-initialization (for $j = 1, 2, \ldots, N$)</td>
</tr>
<tr>
<td>Predicted mode probability: $\mu_{k</td>
</tr>
<tr>
<td>Mixing weight: $\mu_{k</td>
</tr>
<tr>
<td>Mixing estimate: $\hat{x}_{k</td>
</tr>
<tr>
<td>Mixing covariance: $\tilde{P}_{k</td>
</tr>
<tr>
<td>2. Model-conditioned filtering (for $j = 1, 2, \ldots, N$)</td>
</tr>
<tr>
<td>Predicted state: $\hat{x}_{k</td>
</tr>
<tr>
<td>Predicted covariance: $P_{k</td>
</tr>
<tr>
<td>Measurement residual: $\tilde{z}_k^{(j)} = y_k - C_k^{(j)} \hat{x}_k^{(j)} - \hat{D}_k^{(j)} u_k - \tilde{R}_k^{(j)} v_k$</td>
</tr>
<tr>
<td>Residual covariance: $\Sigma_k^{(j)} = C_k^{(j)} (C_k^{(j)})^T + R_k^{(j)}$</td>
</tr>
<tr>
<td>Filter gain: $K_k^{(j)} = P_{k</td>
</tr>
<tr>
<td>Updated state: $\hat{x}_{k</td>
</tr>
<tr>
<td>Updated covariance: $P_{k</td>
</tr>
<tr>
<td>3. Model probability update (for $j = 1, 2, \ldots, N$)</td>
</tr>
<tr>
<td>Model likelihood: $L_k^{(j)} = \exp \left[ -1/2 (\tilde{z}_k^{(j)} - \Sigma_k^{(j)}) \Sigma_k^{(j)-1} \tilde{z}_k^{(j)} \right] / [2\pi \Sigma_k^{(j)}]^{1/2}$</td>
</tr>
<tr>
<td>Model probability: $\mu_k^{(j)} = \sum_{i=1}^{N} \mu_{k</td>
</tr>
<tr>
<td>4. Fusion of estimates</td>
</tr>
<tr>
<td>Overall state: $\hat{x}_{k</td>
</tr>
<tr>
<td>Overall covariance: $P_{k</td>
</tr>
</tbody>
</table>
In the first step, mixing of the estimated states $\hat{x}_k^{(j)}$ and error covariances $P_k^{(j)}$ takes place. The transition probability matrix $\Pi$ forms the basis for this mixing step. In the second step, one cycle of the Kalman filter is performed for each local model. In the third step, the likelihood for each local model is computed based on the probability distribution, which is assumed to be Gaussian:

$$L_k^{(j)} \triangleq p(y_k^{(j)}|s_k^{(j)}, Y_{k-1}) \overset{\text{assume}}{=} \mathcal{N}(\tilde{z}_k^{(j)}, 0, \Sigma_k^{(j)})$$

(12)

where $s_k^{(j)}$ is the mode sequence $\{s_1 = j, \ldots, s_k = j\}$ and $\mathcal{N}(z; \tilde{z}, \Sigma) = \exp\left[-(z - \tilde{z})^T\Sigma^{-1}(z - \tilde{z})/2\right]/(\sqrt{2\pi}\Sigma)$. In the fourth and final step, the fusion of the local estimates of the state vector and covariance matrix takes place in order to obtain the overall values of these two quantities. The fusion is based on the model weights computed in the third step of the algorithm. Note that the overall values of the state and covariance are not used in the next cycle of the filter. The main tuning parameter of the IMM filter is the transition probability matrix $\Pi$.

### 3.2. Dual CMF

In the following, an objective function is derived that needs to be minimized to solve the estimation problem formulated in (11). Using Bayes’ rule, the joint conditional density $p(\mu_k, x_k|y_k)$ from (11) can be expanded as

$$p(\mu_k, x_k|y_k) = \frac{p(y_k|x_k, \mu_k)p(x_k, \mu_k)}{p(y_k)}$$

$$= \frac{p(y_k|x_k, \mu_k)p(x_k|\mu_k)p(\mu_k)}{p(y_k)}$$

(13)

Next, the following relation [18]

$$p(y_k, x_k|\mu_k) = p(y_k|x_k, \mu_k)p(x_k|\mu_k)$$

(14)

can be substituted in (13) to obtain

$$p(\mu_k, x_k|y_k) = \frac{p(y_k|x_k, \mu_k)p(\mu_k)}{p(y_k)}$$

(15)

The problem of maximizing $p(\mu_k, x_k|y_k)$ as a function of $\mu_k$ and $x_k$ (i.e. solving the problem formulated in (11)) boils down to maximizing $p(y_k, x_k|\mu_k)$. The reason for this is that $p(y_k)$ is a function of neither $x_k$ nor $\mu_k$ and furthermore the term $p(\mu_k)$ can also be discarded if it is assumed that there is no prior information available on the distribution of the weights [22]. This assumption is motivated by the observation that the transition probability matrix in a JMLS (i.e. the prior information on the weights) in practice often is not known in advance and is therefore considered as a tuning parameter [17]. The same lack of a priori information is also assumed for the CM structure.
MULTIPLE MODEL ESTIMATION: A CONVEX MODEL FORMULATION

If the relation from (14) is considered, the following objective function can be obtained:

\[
p(y_k, x_k | \mu_k) = \frac{1}{\sqrt{2\pi Q_k}} \exp\left( -\frac{1}{2} (x_k - \bar{x}_k)^T Q_k^{-1} (x_k - \bar{x}_k) \right) \times \frac{1}{\sqrt{2\pi R_k}} \exp\left( -\frac{1}{2} (y_k - \bar{y}_k)^T R_k^{-1} (y_k - \bar{y}_k) \right)
\]

with

\[
x_k^- = A_{\mu_k} \hat{x}_{k-1} + B_{\mu_k} u_{k-1}
\]

\[
\bar{y}_k = C_{\mu_k} x_k + D_{\mu_k} u_k
\]

and

\[
A_{\mu_k} = \sum_{i=1}^{N} \mu_k^{(i)} A^{(i)}, \quad B_{\mu_k} = \sum_{i=1}^{N} \mu_k^{(i)} B^{(i)}
\]

\[
C_{\mu_k} = \sum_{i=1}^{N} \mu_k^{(i)} C^{(i)}, \quad D_{\mu_k} = \sum_{i=1}^{N} \mu_k^{(i)} D^{(i)}
\]

The notation from (19) to (20) will also be used in the remainder of the paper in a similar manner for indices other than \( \mu_k \). Taking the logarithm of (16), dropping the terms independent of both \( \mu_k \) and \( x_k \) and changing the sign, reduces it to the following objective function:

\[
J(x_k, \mu_k) = (x_k - \bar{x}_k)^T Q_k^{-1} (x_k - \bar{x}_k) + (y_k - \bar{y}_k)^T R_k^{-1} (y_k - \bar{y}_k)
\]

Now, (11) can be rewritten as the following optimization problem:

\[
(\hat{\mu}_k, \hat{x}_k | k) = \arg \min_{\mu_k, x_k} J(x_k, \mu_k)
\]

This problem is nonlinear due to the product of \( x_k \) and \( \mu_k \) that is present in (21). In general, solving problem (22) can be done by using two approaches. These approaches can be decoupled or direct [18]. In the direct approach, both \( x_k \) and \( \mu_k \) are estimated jointly by solving a multivariate nonlinear optimization problem. The principle of decoupled approaches is to optimize with respect to one variable at a time, while keeping the other variable fixed and vice versa [18].

In the following, a filter that uses the decoupled approach is described. This filter, the dual CMF, uses two linear filtering steps to minimize \( J(x_k, \mu_k) \). In the first step of the dual CMF, the weight vector is assumed to be fixed and it is chosen to be the current estimate, i.e. \( \mu_k = \hat{\mu}_{k-1} \). Subsequently, an estimate of the state is obtained by minimizing the following objective function with respect to the state \( x_k \):

\[
J(x_k, \hat{\mu}_{k-1}) = (x_k - \bar{x}_k)^T Q_k^{-1} (x_k - \bar{x}_k) + (y_k - \bar{y}_k)^T R_k^{-1} (y_k - \bar{y}_k)
\]
with

\[ \hat{x}_k = A \hat{x}_{k-1} + B \hat{u}_{k-1} \]
\[ \hat{y}_k = C \hat{x}_{k-1} + D \hat{u}_k \]

The minimization of (23) corresponds to applying one step of the Kalman filter to the CM structure in which the current estimates of the weights are used (i.e. \( \hat{\mu}_k = \hat{\mu}_{k-1} \)). This has been implemented in Step 1 of Table II by using one iteration of the square root covariance filter (SRCF) implementation of the Kalman filter. The SRCF is a numerically more robust implementation, which solves exactly the same problem as the Kalman filter. See [23] for a detailed treatment of the SRCF. In Step 1, the matrices \( S_{k|k} \) and \( S_{k|k-1} \) correspond to the error covariance matrices defined as

\[ S_{k|k} S_{k|k}^T = E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T] \]
\[ S_{k|k-1} S_{k|k-1}^T = E[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T] \]

In Step 2, the state is assumed to be fixed and it is chosen to be the current estimate, i.e. \( x_k = \hat{x}_{k|k} \). Subsequently, the following objective function is minimized with respect to \( \hat{\mu}_k \):

\[ J(\hat{x}_{k|k}, \hat{\mu}_k) = (\hat{x}_{k|k} - x_k)^T Q_k^{-1} (\hat{x}_{k|k} - x_k) + (\hat{y}_k - y_k)^T R_k^{-1} (\hat{y}_k - y_k) \]

with

\[ \hat{y}_k = C \hat{x}_{k|k} + D \hat{u}_k \]

The minimization of objective function (28) is performed in Step 2 of the dual CMF. In this step, one iteration of a (square root information) recursive least-square (RLS) estimator [24] is used to obtain \( \hat{\mu}_k \). The RLS estimator has been implemented with a forgetting factor \( \lambda \in [0, 1] \). The forgetting factor determines to what extent old data are taken into account. In case \( \lambda = 1 \), the RLS estimator equally weights all data from the past. The smaller \( \lambda \) is chosen, the more past data are ‘forgotten’. The forgetting factor \( \lambda \) forms the main tuning parameter of the dual CMF (Table II).

3.2.1. Covariance propagation. In the two previously described steps of the dual CMF, a direct substitution of the current estimates of the weights and states into the objective function (21) is performed. With this direct substitution, the uncertainty of the current estimates is overlooked. If this uncertainty were also to be considered, in the first step the substitution \( \hat{\mu}_k = \tilde{\mu}_{k-1} \) would have been made. In this substitution, the uncertainty of \( \hat{\mu}_{k-1} \) is expressed by the term \( S_{k-1}^{\mu} \tilde{\mu}_k \) consisting of the error covariance matrix \( S_{k-1}^{\mu} \) and a zero-mean white noise term \( \tilde{\mu}_k \). For the second step of the dual CMF, the substitution \( x_k = \hat{x}_{k|k} - S_{k|k} \hat{x}_k \) would have been made. In this substitution \( x_k \) is a zero-mean white noise sequence. With these two alternative substitutions, the two steps of the dual CMF would have changed to nonlinear estimation steps due to cross-products of the introduced noise terms and the quantities to be estimated. Therefore, in this paper the direct substitution of current estimates is preferred as is also done in, e.g. [18]. In [25] also an attempt is made to incorporate the errors of the estimates. This attempt resulted in modified versions of (23) and (28), which had to be linearized to allow the two estimation steps to be linear again. However, it was concluded that the performance of the resulting algorithm was not better than the algorithm that used direct substitution of the current estimates.
MULTIPLE MODEL ESTIMATION: A CONVEX MODEL FORMULATION

Table II. The dual CMF.

1. State estimation: Given $A_{\hat{p}_k-1}, B_{\hat{p}_k-1}, C_{\hat{p}_k-1}, D_{\hat{p}_k-1}, Q_k, R_k, S_k, \hat{y}_k, u_k$ and $y_k$, compute
   
   - Using the $QR$-decomposition, find the orthogonal matrix $U_1$ and matrices $R_e, G_{k,1}, G_{k,2}, S_k, V_{k,1}$ and $V_{k,2}$, such that
     
     $$
     \begin{bmatrix}
     R_e & 0 & 0 \\
     G_{k,1} & S_k & 0 \\
     G_{k,2} & V_{k,1} & V_{k,2}
     \end{bmatrix} =
     \begin{bmatrix}
     C_{\hat{p}_k-1} S_k & -R_k^{1/2} & 0 \\
     S_k & 0 & 0 \\
     A_{\hat{p}_k-1} S_k & 0 & Q_k^{1/2}
     \end{bmatrix} U_1
     $$
     
     (30)
   
   - Compute
     
     $$
     \hat{x}_{k|k} = \hat{x}_{k|k-1} - G_{k,1} R_e^{-1} (C_{\hat{p}_k-1} \hat{x}_{k|k-1} + D_{\hat{p}_k-1} u_k - y_k)
     $$
     
     (31)
   
   - Using the $QR$-decomposition, find orthogonal matrix $U_2$ and matrix $S_{k+1|k}$, such that
     
     $$
     [S_{k+1|k} \ 0] = [V_{k,1} \ V_{k,2}] U_2
     $$
     
     (32)
   
   - Compute
     
     $$
     \hat{x}_{k+1|k} = A_{\hat{p}_k-1} \hat{x}_{k|k-1} + B_{\hat{p}_k-1} u_k - G_{k,2} R_e^{-1} (C_{\hat{p}_k-1} \hat{x}_{k|k-1} + D_{\hat{p}_k-1} u_k - y_k)
     $$
     
     (33)

2. Model weight estimation: Given
   
   $$
   \begin{bmatrix}
   A^{(1)} & B^{(1)} \\
   C^{(1)} & D^{(1)} \\
   \vdots & \vdots \\
   A^{(N)} & B^{(N)} \\
   C^{(N)} & D^{(N)}
   \end{bmatrix}
   $$
   
   $\hat{x}_{k-1|k-1}, \hat{x}_{k|k}, \hat{z}_{k-1}, T_k-1, \hat{z}_{k-1}, u_k$ and $y_k$, compute
   
   - Let
     
     $$
     \Phi =
     \begin{bmatrix}
     Q_k^{-1} [A^{(1)} B^{(1)}] [\hat{x}_{k-1|k-1} u_{k-1}] & \cdots & [A^{(N)} B^{(N)}] [\hat{x}_{k-1|k-1} u_{k-1}] \\
     R_k^{-1} [C^{(1)} D^{(1)}] [\hat{x}_{k|k} u_k] & \cdots & [C^{(N)} D^{(N)}] [\hat{x}_{k|k} u_k]
     \end{bmatrix}
     $$
     
     (34)
   
   and
     
     $$
     \Gamma =
     \begin{bmatrix}
     Q_k^{-1} \hat{x}_{k|k} \\
     R_k^{-1} \hat{y}_k
     \end{bmatrix}
     $$
     
     (35)
   
   Using the $QR$-decomposition, find an orthogonal matrix $U_3$ and matrices $T_k, \hat{z}_k$ and $e_k$, such that
     
     $$
     \begin{bmatrix}
     T_k & \hat{z}_k \\
     0 & e_k
     \end{bmatrix} = U_3 \begin{bmatrix}
     \sqrt{T} T_k^{-1} & \sqrt{T} \hat{z}_{k-1} \\
     \Phi & \Gamma
     \end{bmatrix}
     $$
     
     (36)
   
   - Compute
     
     $$
     \hat{\mu}_k = T_k^{-1} (\hat{z}_k)
     $$
     
     (37)
3.2.2. Constraint implementation. The algorithm described in Table II does not take the constraints from (10) into account. This can be done in an additional step by using the theory for implementing equality and inequality constraints in Kalman filters as described in [26, 27], respectively. For implementing the equality constraint, the projection method described in [26] is used. This method directly projects the unconstrained estimate $\hat{\mu}_k$ onto the constraint surface to obtain the constrained estimate $\hat{\mu}_k^c$. This problem is formulated as follows:

$$
\min_{\hat{\mu}_k} (\hat{\mu}_k^c - \hat{\mu}_k) \; W \; (\hat{\mu}_k^c - \hat{\mu}_k) \; \text{such that} \; D\hat{\mu}_k^c = d \tag{38}
$$

where $W$ is a symmetric positive-definite weighting matrix, which is chosen as an identity matrix. The solution to this problem is given by

$$
\hat{\mu}_k^c = \hat{\mu}_k - W^{-1} D^T (D W^{-1} D^T)^{-1} (D \hat{\mu}_k - d) \tag{39}
$$

The inequality constraint can be also cast as an equality constraint if it is a priori known that the inequality constraint indeed holds. This idea was also suggested in [27]. For estimating $\hat{\mu}_k^c$ this means that first it has to be determined for which $i$ it holds that $\hat{\mu}_k^{(i)} < 0$. Subsequently, for these $i$’s an extra equality constraint is added in the form of $\hat{\mu}_k^{(i)} = 0$ to the already existing equality constraint $\sum_{k=1}^{N} \mu_k^{(i)} = 1$. With this augmented set of constraints a constrained estimate $\hat{\mu}_k^c$ is computed. This constrained estimate is to be checked again for $\hat{\mu}_k^{(i)}$’s that violate the inequality constraint. If this is the case, then a new constrained estimate should be computed in the same way as the previous one. This sequence is repeated until the full constrained weight vector $\hat{\mu}_k^c$ satisfies the constraints.

3.3. EKF

The nonlinear problem of estimating both the state and weights as is formulated in (11) has been reformulated as (22). In the previous section, this objective function is minimized in a decoupled fashion. In this section, the direct approach in which the state and model weights are jointly estimated is adopted. A novel algorithm for this nonlinear estimation problem is proposed in [28]. The main feature of this algorithm is that it reduces the number of parameters in the nonlinear estimation problem. In this algorithm, only the weight vector is estimated in a nonlinear estimation step. The state is estimated in a linear estimation step. Because the nonlinear estimation step can become computationally intensive, especially if the model set is large, another algorithm will be explained in this paper. This algorithm uses linearization as an approximation to the nonlinear problem. This approximation is performed by an EKF. In this filter, the state is augmented with the weights. At each time instant, the nonlinear augmented state-space model is linearized around the current estimate. This approach is quite common and is proposed in, e.g. [4, 19] for general parameter estimation problems. In the augmented state, the weights are assumed to evolve as a random walk process:

$$
\mu_k = \mu_{k-1} + w_{\mu_k} \tag{40}
$$
where $w_{\mu_k}$ is a zero-mean white noise sequence. The resulting augmented state model obeys the following equations:

$$
\begin{bmatrix}
    z_k \\
    x_k \\
    \mu_k 
\end{bmatrix} =
\begin{bmatrix}
    A_{\mu_{k-1}} & 0 & 0 \\
    0 & I & 0 \\
    0 & 0 & I 
\end{bmatrix}
\begin{bmatrix}
    x_{k-1} \\
    x_{k-1} \\
    \mu_{k-1} 
\end{bmatrix} +
\begin{bmatrix}
    B_{\mu_{k-1}} \\
    0 \\
    0 
\end{bmatrix}
\begin{bmatrix}
    u_{k-1} \\
    0 \\
    0 
\end{bmatrix} +
\begin{bmatrix}
    Q_{k-1} \\
    0 \\
    0 
\end{bmatrix}
\begin{bmatrix}
    w_{k-1} \\
    \tilde{w}_{k-1} \\
    \tilde{w}_{k-1} 
\end{bmatrix}
$$

(41)

$$
y_k = [C_{\mu_k} 0]
\begin{bmatrix}
    x_k \\
    \mu_k 
\end{bmatrix} + D_{\mu_k} u_k + R_k \tilde{v}_k
$$

(42)

where $\tilde{w}_{k-1}$ and $\tilde{v}_k$ are zero-mean white noise sequences with unit variance of appropriate dimensions and $Q_{k-1}^\mu$ is the covariance of the weight vector $\mu_{k-1}$. The augmented state and covariance matrix are denoted by $z_k$ and $Q_{k-1}^a$, respectively. Let (41) and (42) be given by the following shorthand notation:

$$
z_k = F(z_{k-1}, u_{k-1}, \tilde{w}_{k-1})
$$

(43)

$$
y_k = H(z_k, u_k, \tilde{v}_k)
$$

(44)

then the Jacobian matrices required for the EKF are given by

$$
\bar{A}_{k-1} = \frac{\partial F}{\partial z} \bigg|_{z = \hat{z}_{k-1} | k-1}
$$

(45)

$$
\bar{C}_k = \frac{\partial H}{\partial z} \bigg|_{z = \hat{z}_{k} | k-1}
$$

(46)

The equations for the EKF are given in Table III. The main tuning parameter of the EKF is the covariance matrix $Q_{k}^\mu$. The convexity constraints can be included in the same manner as explained in Section 3.2.2 for the dual CMF.

3.4. Properties of MM filters

At this stage three different MM filters are described: the IMM filter, which uses the hybrid model structure, and the dual CMF and EKF, which use the CM structure. In Table IV an overview is given of the properties of the described filters. One of the main differences between the IMM filter and the filters based on the CM structure is the number of local filters. Contrary to the IMM, which uses $N$ local filters, the CM-based filters use only one global filter. The main tuning parameter of the IMM filter is the transition probability matrix $\Pi$. In practice, tuning this parameter might prove to be a difficult task [17]. The counterparts of $\Pi$ in the other two filters are $\hat{\lambda}$ and $Q_{k}^\mu$. These parameters are much easier to tune in practice since $\hat{\lambda}$ is a single scalar and $Q_{k}^\mu$ can be chosen as a diagonal matrix of which the size of its entries relative to $Q_k$ is of importance. More information on how to choose the tuning parameters of the different filters can be found in Section 4. Model interaction between the different local models is present in the IMM filter only as a step of the filter. There is no interaction between models in the model structure (JMLS) itself. In the other two filters, there is an explicit interaction present in the CM structure.
Given
\[
\begin{bmatrix}
A^{(1)} & B^{(1)} \\
C^{(1)} & D^{(1)}
\end{bmatrix}
\cdots
\begin{bmatrix}
A^{(N)} & B^{(N)} \\
C^{(N)} & D^{(N)}
\end{bmatrix}
\]
compute

**Time update**
\[
\hat{x}_{k|k-1} = A \hat{x}_{k-1|k-1} + B u_{k-1}
\]
\[
\hat{p}_{k|k-1} = \hat{p}_{k-1|k-1}
\]
\[
P_{k|k-1} = \bar{A}_{k-1} P_{k-1|k-1} \bar{A}_{k-1}^T + Q_k
\]

**Measurement update**
\[
K_k = P_{k|k-1} C_k (C_k P_{k|k-1} C_k^T + R_k)^{-1}
\]
\[
\begin{bmatrix}
\hat{x}_{k|k} \\
\hat{p}_{k|k}
\end{bmatrix} =
\begin{bmatrix}
\hat{x}_{k|k-1} \\
\hat{p}_{k|k-1}
\end{bmatrix} + K_k (y_k - C_k \hat{x}_{k|k-1} - D_k u_k)
\]
\[
P_{k|k} = (I - K_k C_k) P_{k|k-1}
\]

with
\[
\bar{A}_{k-1} =
\begin{bmatrix}
A \hat{x}_{k-1|k-1} \\
0 \\
1
\end{bmatrix}
\]
\[
\bar{C}_k = [C \hat{x}_{k|k-1} + D u_k]
\]

Table III. The EKF.

Table IV. Overview of the properties of the described filters.

<table>
<thead>
<tr>
<th>Structure</th>
<th>IMM</th>
<th>Dual CMF</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of filters</td>
<td>N</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Tuning parameters</td>
<td>(\Pi)</td>
<td>(\hat{\lambda})</td>
<td>(Q_k)</td>
</tr>
<tr>
<td>Model interaction</td>
<td>Filter</td>
<td>Structure</td>
<td>Structure</td>
</tr>
</tbody>
</table>

4. EXPERIMENTAL RESULTS

In order to compare the performance of the different previously explained filters, Monte Carlo-type simulation experiments have been performed using two problem cases. Monte Carlo simulations in the context of this paper are characterized by the consideration of a number of different simulation...
MULTIPLE MODEL ESTIMATION: A CONVEX MODEL FORMULATION

runs, based on the same model, but each with different random noise sequences for signals such as process noise, measurement noise and input signals. The first simulation experiment considers an FDI problem in a linearized model of a Boeing 747 aircraft. The second simulation experiment considers a target tracking problem.

4.1. Aircraft FDI

For this simulation experiment, a linearized model of a Boeing 747 aircraft is used. This model is being used as a benchmark in action group 16 of the Group for Aeronautical Research and Technology in Europe project. This action group aims at integrating advanced FDI methods with control reconfiguration schemes. Within this action group, a benchmark scenario is defined consisting of different flight phases [29]. In the performed simulations, a model is linearized during the first flight phase of this benchmark scenario. During this first flight phase, the aircraft flies at a constant altitude of 600 m with a constant velocity of 133.8 m/s. The continuous linear time-invariant model of the longitudinal dynamics of the Boeing 747 at the previously mentioned operating point is given by

\[
\dot{x}(t) = Ax(t) + Bu(t) + w(t) \tag{55}
\]

\[
y(t) = Cx(t) + v(t) \tag{56}
\]

where the system matrices \( A, B \) and \( C \) are as given in the Appendix. The state of the model is given by \( x = [q \ V_{TAS} \ \alpha \ \theta \ h]^T \), where \( q \) (rad/s) is the pitch rate, \( V_{TAS} \) (m/s) is the true airspeed, \( \alpha \) (rad) is the angle of attack, \( \theta \) (rad) is the roll angle and \( h \) (m) is the altitude. The input of the model is given by \( u = [\delta_c \ \delta_s \ T_{n1} \ T_{n2} \ T_{n3} \ T_{n4}]^T \), where \( \delta_c \) (rad) is the column deflection, \( \delta_s \) (rad) is the stabilizer deflection and \( T_{n1} - T_{n4} \) (N) are the thrust inputs for the four engines. The measurement vector \( y \) contains measurements of \( q, V_{TAS}, \alpha \) and \( h \), respectively.

The goal of this simulation experiment is to identify four classes of faults. These fault classes include two types of actuator faults: loss of effectiveness (LOE) in the column and stabilizer commands. Furthermore, two classes of sensor faults are modeled: multiplicative faults in the \( V_{TAS} \) and \( \alpha \) sensors. A model set consisting of five models is used for the purpose of identifying these four classes of faults. Table V gives a description of this model set. The system matrices of the nominal model are given in the Appendix by (A1)–(A3). The system matrices for Model 2 in the model set are the same as the nominal model except for the \( B \)-matrix, which is given by \( B^{(2)} \) in Equation (A4) of the Appendix. The same holds for Model 3, for which the \( B \)-matrix is given by \( B^{(3)} \) defined in (A5). Note that \( B^{(2)} \) and \( B^{(3)} \) are created by zeroing the relevant column. Models 4 and 5 differ from the nominal model in the \( C \)-matrices, which are defined by \( C^{(4)} \) and \( C^{(5)} \), respectively, in Equation (A6) of the Appendix. The differences of the matrices defined in (A4)–(A6) with respect to the nominal matrices are pointed out in the Appendix by an underlining of the changed entries.

For MM estimation, the models in the model set are discretized using a sample time of \( T_s = 0.01 \) s. Using the discretized models, a fault scenario is simulated in closed-loop. An LQR state feedback controller synthesized for the nominal system is used in the closed-loop system. This state feedback controller uses the nominal states, which means that it is assumed that the non-faulty state signal is always available for computation of the control input. Additional noise is added to the LQR control signal to provide different input realizations for the Monte Carlo simulations and to provide system excitation, which is desirable for FDI. The simulated fault scenario consists of a sequence of faults corresponding to partial occurrences of the faults modeled by the models in the model.
set. These partial faults do not correspond exactly to the models from the model set. Instead, they correspond to weighted combinations of the models from the model set. Therefore, this experiment allows evaluation of the interpolation properties of the different filters. The simulated fault scenario is described in Table VI. Noise with a covariance of $R = \text{diag}(10^{-5} \ 10^{-4} \ 10^{-5} \ 10^{-3})$ is added to the measurements and noise with a covariance of $Q = 10^{-6}I_5$ is added to the process. This particular choice of parameters results in a signal-to-noise (SNR) ratio of approximately 35 dB for all four measurements.

The tuning parameter $\Pi$ of the IMM filter in this experiment is chosen as follows:

$$
\Pi = \begin{bmatrix}
0.9 & 0.025 & 0.025 & 0.025 & 0.025 \\
0.1 & 0.9 & 0 & 0 & 0 \\
0.1 & 0 & 0.9 & 0 & 0 \\
0.1 & 0 & 0 & 0.9 & 0 \\
0.1 & 0 & 0 & 0 & 0.9
\end{bmatrix}
$$

(57)

The rationale behind this choice is that if the system is in the nominal (fault-free) condition, the probability of occurrence for each of the four possible faults is 0.025. Once one of the four faults has occurred, the system remains in the same fault condition with a probability of 0.9. The probability that the system returns from each of the faulty conditions to the nominal condition is 0.1. The covariance matrices for the IMM filter are chosen as $Q_k^{\text{IMM}} = Q$ and $R_k^{\text{IMM}} = R$ for all $k$ and for all five models. The covariance matrices for the dual CMF are chosen as $Q_k^{\text{DCMF}} = Q$, $R_k^{\text{DCMF}} = R$ for all $k$, and the forgetting factor is chosen as $\lambda = 0.75$. The covariance matrices for the EKF are chosen as $Q_k^{\text{EKF}} = Q$, $R_k^{\text{EKF}} = R$ and $Q_k^{\text{EKF}} = 0.01I_5$ for all $k$. The values of the main tuning parameters of the different filters ($\Pi$, $\lambda$ and $Q_k^{\text{EKF}}$) are obtained by careful tuning. Tuning of $\lambda$ and $Q_k^{\text{EKF}}$ is relatively easy: if the entries of diagonal matrix $Q_k^{\text{EKF}}$ are chosen too large and $\lambda$ is chosen too small, then the estimated model weights vary too much. However, if the diagonal entries of
$Q^k$ are chosen too small and $\lambda$ is chosen too large (e.g. $\lambda = 1$), then the estimated model weights hardly vary. Hence, a good compromise between these two extremes must be found by proper tuning. For the IMM filter, the tuning is not so straightforward. The transition probability matrix $\Pi$ contains information on the probabilities of jumping from one model to other models. These probabilities can be chosen different for each jump, which makes $\Pi$ difficult to tune. Therefore, usually a number of jumps are chosen to have the same probability as is done in this experiment.

The performance of the MM estimation algorithms has been evaluated by using 100 Monte Carlo simulations. The root mean square error (RMSE), which is closely related to standard deviation, is used as a performance measure. The RMSE is defined as

$$\text{RMSE}(\hat{p}_k) = \sqrt{\frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} (p_k - \hat{p}_{k,i})^T (p_k - \hat{p}_{k,i})}$$

where $N_{MC}$ is the number of Monte Carlo simulations, $p_k$ is the real value of quantity $p$ at time instant $k$ and $\hat{p}_{k,i}$ is the estimation of $p_k$ in the $i$th Monte Carlo simulation run. $p_k$ can also be a vector, in which case the RMSE may only be physically interpretable if the entries of the vector have the same units.

The RMSE values for the state and the weight vector for the defined fault scenario are depicted in Figures 2 and 3, respectively. The shaded intervals in these two figures correspond to regions in which faults are injected into the system. The RMSE value for the state is unitless since the state consists of physical quantities with different units. Although the state RMSE is not physically interpretable, it gives a good overview of the state estimation performance. Furthermore, the computation of the state RMSE in this way is justified by the fact that the errors in the five individual signals that form the state are of the same order of magnitude. In Figure 2 it can be

![Figure 2. RMSE of the estimated states during the FDI scenario.](image-url)
observed that no clear pattern can be recognized on which to base conclusions on state estimation performance. However, Figure 2 is still important since it shows that the estimation performance of all three filters is acceptable; the largest RMSE error is about $10^{-0.85}$. Figure 3 gives a clearer view on how the filters perform relative to each other. In this figure, it can be observed that the weight estimation performance of the two filters based on the CM structure is always better than the performance of the IMM filter in the intervals in which faults have occurred. Especially, in the sample intervals $[400, 500]$ and $[600, 700]$, the RMSE of the weights of the IMM filter becomes relatively large. The reason for this is that in these intervals the weight estimates switch quickly between two models (i.e. in these intervals the two models interchange a full probability of 1 in an abrupt manner) because the IMM filter has difficulties interpolating the two concerning models. Furthermore, it can be seen that the IMM filter generally performs better than the two filters based on the CM structure in the intervals without faults.

Additionally, from Figure 3 some insights on the convergence properties of the different filters can be gained. From the fact that the weight RMSE of the EKF becomes very small during faults, it can be concluded that the model weights estimated with the EKF get very close to the right value for all faults. The dual CMF performs worse in this aspect, especially the first fault in the interval $[200, 300]$ is not identified well. The same holds for the IMM filter; it does not converge to the right weight for any of the faults. However, a benefit of the dual CMF with respect to the EKF is that it performs better in the cases in which the true system corresponds exactly to a model in the model set, i.e. the sample intervals in which no faults occur. The reason for this is that the EKF is slower to adapt to a nominal condition again after a fault has occurred. Hence, a general design consideration is that of the two filters based on the CM structure, the dual CMF is to be preferred when most of the expected system conditions are modeled by the models in the model set. If it is expected that many of the expected system conditions are represented by convex combinations of the models in the model set (such as is the case for partial faults), then the EKF is filter to be
preferred. Therefore, under the conditions in this simulation example, the EKF is the filter to be preferred for identifying the partial faults.

4.2. Tracking of a maneuvering target

In this section, the problem of tracking a maneuvering target in the presence of noisy measurements is considered. The goal of tracking a target is to obtain a consistent estimate of the state even in case of noisy measurements and maneuvers of the target. The target tracking problem considered in this section is largely based on the target tracking problem also considered in [11, 30]. The state of the target is given by \( x = [p_x \ v_x \ p_y \ v_y]^T \), where \( p_x \) (m) and \( p_y \) (m) represent the positions in the \( x \) and \( y \) directions of the target and \( v_x \) (m/s) and \( v_y \) (m/s) represent the velocities in the \( x \) and \( y \) directions, respectively. The maneuvering target evolves according to a JMLS model, with parameters

\[
A = \begin{bmatrix}
1 & T_s & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & T_s \\
0 & 0 & 0 & 1
\end{bmatrix}, \quad C = I_4
\]  

(59)

The switching term in this model is the \( B \)-matrix. This matrix can have three possible values that correspond to three different maneuver commands: straight flight, left turn and right turn. These three matrices are given by

\[
B^{(1)} = [0 \ 0 \ 0 \ 0]^T \\
B^{(2)} = \sqrt{2} [-0.25 -0.5 \ 0.25 \ 0.5]^T \\
B^{(3)} = \sqrt{2} [0.25 \ 0.5 -0.25 -0.5]^T
\]  

(60)

The fact that the \( B \)-matrix is the only system matrix that is time varying has some consequences for the two filters based on the CM structure. The consequence for the dual CMF is that the assumption that is required for the first step (i.e. \( \hat{\mu}_{k-1} = \hat{\mu}_k \)) is not necessary anymore. This can be seen by observing that the first three equations of the dual CMF described in (30)–(32) do not contain any instances of the \( B \)-matrix. The matrix \( B \) appears for the first time in the fourth equation (33). This equation can now be postponed until the vector \( \mu_k \) is evaluated in the second step of the dual CMF. This means that (33) can be evaluated after (37). The consequence for the EKF is that it simplifies to an ordinary Kalman filter with an augmented state consisting of the state \( x_k \) and the weight vector \( \mu_k \). The reason for this is that there is no product between \( x_k \) and \( \mu_k \) in the system equations since the \( B \)-matrix is the only system matrix that is time varying.

For the simulation experiment, process and measurement noise are used with covariances:

\[
Q = 10^{-3} I_4, \quad R = \begin{bmatrix}
2.5 \times 10^4 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 2.5 \times 10^4 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(61)
This particular choice results in an SNR of approximately 20 dB for all four measurements. The system is simulated for 400 samples using $T_s = 1$ s. In this simulation a left turn is simulated in the sample interval $[100, 150]$ and a right turn is simulated in the sample interval $[250, 300]$. The left turn is simulated with half the acceleration that corresponds to $B^{(2)}$. This means that for samples $k \in [100, 150]$, $B_k = 0.5 B^{(2)}$ is simulated, where $B_k$ denotes the $B$-matrix used at sample $k$. The right turn is simulated with the same acceleration as $B^{(3)}$. The initial state of the target is chosen as $[-500 0 -500 5]^T$.

As is discussed before, one of the benefits of the CM structure is its interpolation property. To illustrate this, the model set used for the two filters based on the CM structure is composed of only two models. These models are the two turn models. This can be done because the possible convex combinations of these two models also include the model for straight flight ($B^{(1)} = 0.5 B^{(2)} + 0.5 B^{(3)}$). The model set used for the IMM filter contains all three models. This choice has been made to allow an honest comparison with the other two filters. The transition probability of the IMM filter in this experiment is chosen as

$$
\Pi = \begin{bmatrix}
0.98 & 0.01 & 0.01 \\
0.01 & 0.98 & 0.01 \\
0.01 & 0.01 & 0.98
\end{bmatrix}
$$

The covariance matrices are chosen as $Q_k^{\text{IMM}} = Q$ and $R_k^{\text{IMM}} = R$ for all $k$ and for all three models. The covariance matrices for the dual CMF are chosen as $Q_k^{\text{DCMF}} = Q$, $R_k^{\text{DCMF}} = R$ for all $k$, and the forgetting factor is chosen as $\lambda = 0.9$. The covariance matrices for the EKF are chosen as $Q_k^{\text{EKF}} = Q$, $R_k^{\text{EKF}} = R$ and $Q_k^{\mu} = 10^{-2} I_2$ for all $k$. The state estimation results obtained with 100 Monte Carlo simulations of the three filters are given in Figure 4. In this figure, the RMSE values of the position states and velocity states have been separately computed to obtain physically interpretable RMSE values. The shaded intervals in Figure 4 correspond to the intervals in which a maneuver takes place. It can be observed that the RMSE values of the position are approximately the same for all three filters. However, the RMSE values of the velocity show significant differences. In general, the IMM filter has the lowest RMSE value, but in the sample interval $[100, 150]$ it clearly does not. The reason for this is that in this interval a maneuver is performed for which the explicit model is not present in the model set used by the IMM filter. However, the same also holds for the other two filters. In the sample interval $[100, 150]$, the IMM filter tends to switch rapidly between two models, which clearly has a negative influence on the state estimation performance. The two CM estimation algorithms, on the other hand, manage to interpolate between the two turn models, resulting in better state estimation performance.

Since the three filters do not all use the same model set, the weight estimation performance cannot be evaluated by analyzing the weights directly. Therefore, the weight estimation performance is evaluated by reconstructing the $B$-matrix using the estimated weights. For the two filters using the CM structure, the $B$-matrix is reconstructed by $\hat{B}_k = \hat{\mu}_k^{(1)} B^{(2)} + \hat{\mu}_k^{(2)} B^{(3)}$. For the IMM filter, which uses a model set containing three models, the $B$-matrix is reconstructed by $\hat{B}_k = \hat{\mu}_k^{(1)} B^{(1)} + \hat{\mu}_k^{(2)} B^{(2)} + \hat{\mu}_k^{(3)} B^{(3)}$. The RMSE of the reconstructed $B$-matrix for the three filters is depicted in Figure 5. Note that this RMSE is dimensionless since it is based on physical quantities with different units. In Figure 5, the same important observation can be made as in Figure 4. This
observation is that in the sample interval [100, 150] the IMM filter is clearly outperformed by the other two filters. In general, the performance of the three filters is in the same order of magnitude.

An important conclusion that can be drawn from this simulation experiment is that the CM structure is to be preferred when interpolation between models from the model set is required. Furthermore, it can be concluded that the CM structure can indeed allow for smaller model sets,
while at the same time maintaining the same estimation performance as an IMM filter with a larger model set. In fact, for situations in which model interpolation is required the estimation performance can be even better with smaller model sets when using the CM structure.

5. CONCLUSIONS

In this paper, an alternative model structure is proposed for the hybrid model structure of JMLS. This structure, named convex model (CM) structure, explicitly allows for interpolation between models. The benefit of the estimation algorithms based on this structure is that they have better model interpolation properties. Having better model interpolation properties allows for smaller model sets. Furthermore, the CM structure does not require transition probabilities, which greatly simplifies the tuning process. Two estimation algorithms based on the proposed CM structure are explained and compared with the well-known IMM filter. Monte Carlo simulation results of an FDI problem and a target tracking problem show the usefulness of the improved interpolation properties.

APPENDIX A

State-space matrices for the linearized Boeing 747 model

\[
A = \begin{bmatrix}
-5.909 \times 10^{-1} & -1.564 \times 10^{-4} & -1.358 & 0 & -1.229 \times 10^{-6} \\
-1.467 \times 10^{-1} & -8.723 \times 10^{-3} & 4.827 & -9.805 & 5.310 \times 10^{-5} \\
9.800 \times 10^{-1} & -1.181 \times 10^{-3} & -6.719 \times 10^{-1} & 0 & 6.831 \times 10^{-1} \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1.338 \times 10^2 & 1.338 \times 10^2 & 0
\end{bmatrix}
\]  

(A1)

\[
B = \begin{bmatrix}
1.412 & 2.6806 & 2.076 \times 10^{-8} & 5.588 \times 10^{-8} & 5.588 \times 10^{-8} & 2.076 \times 10^{-8} \\
0 & 0 & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} \\
4.695 \times 10^{-2} & 0 & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]  

(A2)

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(A3)
MULTIPLE MODEL ESTIMATION: A CONVEX MODEL FORMULATION

\[
B^{(2)} = \begin{bmatrix}
1.412 & 0 & 2.076 \times 10^{-8} & 5.588 \times 10^{-8} & 5.588 \times 10^{-8} & 2.076 \times 10^{-8} \\
0 & 0 & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} \\
4.695 \times 10^{-2} & 0 & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

(A4)

\[
B^{(3)} = \begin{bmatrix}
0 & 2.6806 & 2.076 \times 10^{-8} & 5.588 \times 10^{-8} & 5.588 \times 10^{-8} & 2.076 \times 10^{-8} \\
0 & 0 & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} & 3.314 \times 10^{-6} \\
0 & 0 & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} & -2.911 \times 10^{-9} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

(A5)

\[
C^{(4)} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}, \quad C^{(5)} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

(A6)

REFERENCES


