

Stochastic Filtering in Jump Systems With State Dependent Mode Transitions

Agostino Capponi, Concetta Pilotto

Division of Engineering and Applied Sciences

California Institute of Technology

Email: acapponi@caltech.edu pilotto@cs.caltech.edu

Abstract—We introduce a new methodology to construct a Gaussian mixture approximation to the true filter density in hybrid Markovian switching systems. We relax the assumption that the mode transition process is a Markov chain and allow it to depend on the actual and unobservable state of the system. The main feature of the method is that the Gaussian densities used in the approximation are selected as the solution of a convex programming problem which trades off sparsity of the solution with goodness of fit. A meaningful example shows that the proposed method can outperform the widely used interacting multiple model (IMM) filter in terms of accuracy at the expenses of an increase in computational time.

Keywords: Tracking, Kalman filtering, estimation.

I. INTRODUCTION

Many applications require to estimate the state of a system with time varying dynamics. Applications of these type of systems can be found in target tracking [11], change point detection [2] and financial engineering [7]. A specific class of these systems is the class of jump Markovian systems. The state of such systems consists of a continuous part, the kinematics, and a discrete part, the mode which determines the dynamics in place. In this paper we consider systems in which the mode is not directly observed, also referred in the literature as hidden Markov models. In the simplest case in which the mode switching process is a Markov chain, we obtain the so called Markov jump linear system. In this case, the optimal filter density can be computed using a weighted mode-matched sequence of Kalman filters, one for each trajectory of modes. Since the optimal filtering density cannot be directly computed due the computational complexity which grows exponentially with time, a number of approximation techniques have been developed to deal with these systems, such as the IMM estimator [3], [10], Gaussian mixture reduction techniques [13] and particle filtering methods [8]. An excellent survey of all these techniques can be found in [11].

We consider models in which the mode jump process is not a Markov chain, but state dependent. This relaxes quite a restrictive assumption, which is unrealistic in many practical applications. For example, if a target is constrained to travel on a road network, a maneuver can only be performed at time instances at which the target reaches an intersection. Whether or not the target reaches an intersection in each sampling interval depends on the proximity of the target to it and its speed in traveling towards the intersection. Nevertheless in target tracking, most of the filtering studies have continued

to focus on Markovian jump systems. Some exceptions are Blom [4], who proposes a particle filtering approach for jump systems with non Markovian switching probabilities, and Rozowskii [12] who develops a greedy selection procedure based on matching pursuit to manage the non-linearity arising from the non-Markovian mode switching process.

We develop an approach to solve filtering problems arising in jump systems with state dependent mode transition probabilities. The output of our scheme at each sampling time is an approximation of the true unnormalized posterior density with a mixture of Gaussian densities. The novelty of the approach is that such densities are selected as the solution of a convex programming problem aiming at finding the sparsest possible expansion in terms of Gaussian densities, which is as close as possible to the actual unnormalized density when both are evaluated on a discrete set of training points.

The proposed approach differs from the mixture reduction algorithm based on clustering proposed by Salmond [13] in that we do not set any a-priori upper bound on the number of components in the reduced mixture. Such number is adaptively selected by the convex optimization algorithm based on the maximum error tolerance specified by the user. The number of Gaussian mixture components tends to be low in cases when the density is sufficiently close to a low dimensional Gaussian mixture, while it tends to be high when the shape of the density is far from a Gaussian mixture.

The novel approach has in common with the IMM and the IMM particle filtering, see [4], that the mode conditional densities are approximated right after the mixing step. The proposed method is shown to outperform the widely used IMM estimator in terms of prediction accuracy on a meaningful tracking scenario. However, the methodology is based on an explicit gridding of the state space, which makes it slower than IMM as discussed later in the paper.

The rest of the paper is organized as follows. Section II introduces the general state and observation model along with the mode transition process. Section III develops the exact recursive Bayesian filter. Section IV describes the proposed approximation approach used in computing the filter density estimator. Section V evaluates the methodology on a tracking scenario. Section VI concludes the paper.

II. SETUP AND PROBLEM FORMULATION

The dynamics of the state is modeled by the jump linear switching system

$$\mathbf{x}_k = \mathbf{A}(\theta_{k-1})\mathbf{x}_{k-1} + \mathbf{Q}(\theta_{k-1})\mathbf{w}_k \quad (1)$$

where \mathbf{x}_k is the n dimensional state of the system at time t_k , and θ_k is the mode switching process. We assume that θ_k is a discrete time finite state process taking values in $\{1, \dots, d\}$. For any $1 \leq r \leq d$, $\mathbf{A}(r)$ is a square n dimensional matrix governing the dynamics associated to mode r . The noise process \mathbf{w}_k is a n' dimensional Gaussian vector consisting of n' independent Gaussian random variables with zero mean and unit covariance matrix, while $\mathbf{Q}(r)$ is a mode dependent $n \times n'$ positive semidefinite matrix. We assume that the pair (\mathbf{x}_k, θ_k) is a homogeneous Markov chain with transition matrix P^Δ defined as

$$\begin{aligned} P^\Delta(\mathbf{x}, j, \mathbf{y}, i) &:= P(\mathbf{x}_k = \mathbf{x}, \theta_k = j | \mathbf{x}_{k-1} = \mathbf{y}, \theta_{k-1} = i) \\ &= P(\theta_k = j | \theta_{k-1} = i, \mathbf{x}_k = \mathbf{x}) \cdot \\ &\quad \cdot P(\mathbf{x}_k = \mathbf{x} | \mathbf{x}_{k-1} = \mathbf{y}, \theta_{k-1} = i) \end{aligned} \quad (2)$$

and prior distribution on the chain

$$\begin{aligned} \pi(\mathbf{u}, j) &:= P(\mathbf{x}_0 = \mathbf{u}, \theta_0 = j) \\ &= P(\mathbf{x}_0 = \mathbf{u})P(\theta_0 = j | \mathbf{x}_0 = \mathbf{u}) \end{aligned} \quad (3)$$

where the initial density $P(\mathbf{x}_0 = \mathbf{u})$ is assumed to be Gaussian with mean μ_0 and covariance matrix Σ_0 . It is clear from Eq. (2) and (3) that the mode switching process θ_k is statistically dependent on the state \mathbf{x}_k . In the special case that it is not, then it becomes a Markov chain and we recover the mode dependent state transition probabilities as in classical Markov jump linear systems.

We next discuss the structure of the observation model. The measurement \mathbf{z}_k produced at time t_k is described by the following equation

$$\mathbf{z}_k = \mathbf{H}(\theta_{k-1})\mathbf{x}_k + \mathbf{R}(\theta_{k-1})\mathbf{v}_k \quad (4)$$

The observation matrix \mathbf{H} is a mode dependent $m \times n$ matrix. The measurement noise \mathbf{v}_k is a m' dimensional vector consisting of m' independent Gaussian random variables with zero mean and unit covariance matrix, while \mathbf{R} is a mode dependent $m \times m'$ matrix. For notation simplicity, both the design matrix \mathbf{H} and the transition matrix \mathbf{A} are assumed to be time invariant.

The main objective is to develop an accurate methodology which approximates the unnormalized posterior density

$$p_{k,k}^l(\mathbf{x}) := P(\mathbf{x}_k = \mathbf{x}, \theta_{k-1} = l, \mathbf{Z}^k), l = 1, \dots, d \quad (5)$$

where $\mathbf{Z}^k = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$ denotes the set of observations received up to and including sampling time t_k . Equation (5) then allows to obtain easily the joint state-mode posterior hybrid density, and the posterior density of the state through normalization.

III. EXACT FILTER

A. Notation and terminology

The following notation will be used henceforth:

- $\mathcal{N}(x, \mu, \sigma^2)$: Gaussian r.v. with mean μ and variance σ^2
- $n(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$: multivariate Gaussian density with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
- $\mathbf{Z}^k = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$: observations up to time t_k
- $\lambda^{i,j}(\mathbf{y}) = P(\theta_k = j | \theta_{k-1} = i, \mathbf{x}_k = \mathbf{y})$: mode switching probabilities
- $\lambda^j(\mathbf{y}) = P(\theta_0 = j | \mathbf{x}_0 = \mathbf{y})$: prior mode probability
- $p(\mathbf{x} | \mathbf{y}, l) = P(\mathbf{x}_k = \mathbf{x} | \theta_{k-1} = l, \mathbf{x}_{k-1} = \mathbf{y})$: mode dependent transition density
- $p(\mathbf{x}) = P(\mathbf{x}_0 = \mathbf{x})$: the initial density on \mathbf{x}_0
- $p_{k,k}^l(\mathbf{x})$: unnormalized posterior joint-state mode probability density
- $p_{k|k}(\mathbf{x}) = P(\mathbf{x}_k = \mathbf{x} | \mathbf{Z}^k)$: the posterior density
- $L_{k|k}^l(\mathbf{x}) = p(\mathbf{z}_k | \mathbf{x}_k = \mathbf{x}, \theta_{k-1} = l)$: the mode conditioned measurement likelihood
- $E_f[g] = \int_{\mathbb{R}^n} f(\mathbf{w})g(\mathbf{w})d\mathbf{w}$: the expectation of g with respect to the density f .

B. Exact Bayesian Filter

We use a recursive expression for the unnormalized posterior density, instead of working directly with its normalized counterpart following the approach presented by Blom in [4].

Such density is obtained through an interaction of d Bayesian filters, with each filter being an unnormalized posterior density

$$p_{k,k}^l(\mathbf{x}) := P(\mathbf{x}_k = \mathbf{x}, \theta_{k-1} = l, \mathbf{Z}^k), l = 1, \dots, d \quad (6)$$

The unnormalized prediction density is defined as

$$p_{k,k-1}^l(\mathbf{x}) := P(\mathbf{x}_k = \mathbf{x}, \theta_{k-1} = l, \mathbf{Z}^{k-1}) \quad (7)$$

and can be developed as

$$\begin{aligned} \int_{\mathbb{R}^n} \sum_{r=1}^d P(\mathbf{x}_k = \mathbf{x}, \mathbf{x}_{k-1} = \mathbf{y}, \theta_{k-1} = l, \theta_{k-2} = r, \mathbf{Z}^{k-1}) d\mathbf{y} \\ = \sum_{r=1}^d \int_{\mathbb{R}^n} \lambda^{r,l}(\mathbf{y}) p(\mathbf{x} | \mathbf{y}, l) p_{k-1,k-1}^r(\mathbf{y}) d\mathbf{y} \end{aligned} \quad (8)$$

where $p(\mathbf{x} | \mathbf{y}, l)$ is a Gaussian density with mean $\mathbf{A}(l)\mathbf{y}$ and covariance $\mathbf{Q}(l)\mathbf{Q}(l)'$. The above decomposition steps above follow from straightforward application of Bayes rule. The unnormalized posterior joint state-mode probability density function may then be obtained as

$$\begin{aligned} p_{k,k}^l(\mathbf{x}) &= L_{k|k}^l(\mathbf{x}) p_{k,k-1}^l(\mathbf{x}) \quad k \geq 2 \\ p_{1,1}^l(\mathbf{x}) &= L_{1|1}^l(\mathbf{x}) \int_{\mathbb{R}^n} p(\mathbf{x} | \mathbf{u}, l) \pi(\mathbf{u}, l) p(\mathbf{u}) d\mathbf{u} \end{aligned} \quad (9)$$

where the correction term $L_{k|k}^l(\mathbf{x})$ is a Gaussian density with mean $\mathbf{H}\mathbf{x}$ and covariance $\mathbf{R}(l)\mathbf{R}(l)'$ due to the structure of the observation equation (4).

From the above derivations, we can see that the filters interact with each other according to fundamental Bayesian rules, leading to the linear recursive expression given in Eq. (9) for the unnormalized density. The posterior density may

then be obtained from the interacting bayesian filters through normalization as

$$p_{k|k}(\mathbf{x}) = \frac{\sum_{l=1}^d p_{k,k}^l(\mathbf{x})}{\int_{\mathbb{R}^n} \sum_{l=1}^d p_{k,k}^l(\mathbf{x})} \quad (10)$$

IV. THE FILTER APPROXIMATION SCHEME

Although an exact filter has been derived in Subsection III-B, it is not amenable to an efficient implementation. First of all, the recursive expression (9) shows that an exponentially increasing number of filters have to interact to obtain the unnormalized density at time t_k . Additionally, Eq. (8) involves the evaluation of non Gaussian integrals due to the appearance of the terms $\lambda^{j,l}$ in $p_{k,k-1}^l(\mathbf{x})$, and such integrals may in general be computationally expensive to evaluate. In order to deal with these computational issues, we propose an approach which at every step k approximates the unnormalized density using a restricted set of Gaussian densities, selected from a prescribed base set. The number of densities and the weight of each density are recovered as the solution of a convex *second order cone* programming problem.

A. The approximation method

Let $p(\mathbf{x})$ be the density which we wish to approximate. Let I and J be two set of indices. We construct a *base set* of multivariate Gaussian densities

$$B = \{n_{i,j}(\mathbf{x})\}_{i \in I, j \in J} \quad (11)$$

where $n_{i,j}(\mathbf{x})$ stands for the multivariate Gaussian density with mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_j$. Moreover, we require

$$\begin{aligned} \boldsymbol{\mu}_{i_1} &\neq \boldsymbol{\mu}_{i_2}, & \forall i_1 \neq i_2 \\ \boldsymbol{\Sigma}_{j_1} &\neq \boldsymbol{\Sigma}_{j_2}, & \forall j_1 \neq j_2 \end{aligned} \quad (12)$$

meaning that the means and covariances of the Gaussian densities in B are all different. We choose a *training set*

$$\mathcal{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q) \quad (13)$$

of size q containing vectors in \mathbb{R}^n . Let us define the matrix

$$\Phi(\mathcal{X}) = \begin{pmatrix} \mathbf{n}_1(\mathbf{x}_1) & \mathbf{n}_2(\mathbf{x}_1) & \dots & \mathbf{n}_{|I|}(\mathbf{x}_1) \\ \mathbf{n}_1(\mathbf{x}_2) & \mathbf{n}_2(\mathbf{x}_2) & \dots & \mathbf{n}_{|I|}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{n}_1(\mathbf{x}_q) & \mathbf{n}_2(\mathbf{x}_q) & \dots & \mathbf{n}_{|I|}(\mathbf{x}_q) \end{pmatrix} \quad (14)$$

where $\mathbf{n}_i(\mathbf{x}_l) = (n_{i,1}(\mathbf{x}_l), n_{i,2}(\mathbf{x}_l), \dots, n_{i,|J|}(\mathbf{x}_l))$, i.e. a row vector whose j -th entry is the multivariate Gaussian density in B with mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_j$ evaluated at \mathbf{x}_l .

Moreover, we assume that $q < |I| \times |J|$, i.e. the size of the training set is strictly smaller than the cardinality of the base set B . Let $\mathbf{p} = (p(\mathbf{x}_1), \dots, p(\mathbf{x}_q))'$. The linear system

$$\mathbf{p} = \Phi \mathbf{z} \quad (15)$$

is solvable and overdetermined being $q < |I| \times |J|$. Although we could solve the system and then approximate the density $p(\mathbf{x})$ with $\phi \mathbf{z}(\mathbf{x})$, we notice that such approach would require

to propagate a number of Gaussian densities equal to the size q of the training set, and therefore it would scale linearly with the size of the training set, making a real time implementation computationally intensive. Our goal is to approximate \mathbf{p} using a short linear combination of Gaussian densities and at the same time reduce the introduced approximation error. Therefore, we look for the sparsest representation of $\mathbf{p}(\mathbf{x})$ in the following sense:

$$\min \|\mathbf{v}\|_0 \quad \text{subject to } \|\mathbf{p} - \Phi \mathbf{v}\|_2 \leq \iota \quad (P1)$$

(P1) is a mathematical programming problem with decision variables \mathbf{v} , and $\|\mathbf{v}\|_0$ denotes the number of non-zero entries of the vector \mathbf{v} , i.e.

$$\|\mathbf{v}\|_0 = |\{(i, j) : v_{(i,j)} \neq 0\}| \quad (16)$$

where $v_{(i,j)}$ is the entry of the $|I| \times |J|$ dimensional vector \mathbf{v} multiplying the Gaussian density $n_{i,j}$. If \mathbf{v}^* is the solution of (P1), we would approximate $\mathbf{p}(\mathbf{x})$ with $\phi \mathbf{v}^*$. However, this is of little practical use, since the optimization problem (P1) is non-convex and generally impossible to solve as its solution usually requires an intractable combinatorial search. To this purpose, we look for the convex penalty function which is as close as possible to $\|\mathbf{v}\|_0$. This turns out to be the l_1 norm which thus leads to the following optimization problem

$$\min \|\boldsymbol{\omega}\|_1 \quad \text{subject to } \|\mathbf{p} - \Phi \boldsymbol{\omega}\|_2 \leq \epsilon \quad (P2)$$

with decision variable $\boldsymbol{\omega}$. The problems (P1) and (P2) differ only in the choice of the objective function, with the latter using an l_1 norm as a proxy for the sparsity count. However, unlike (P1), (P2) is a convex *second order cone* programming problem and can be solved efficiently using standard optimization algorithms, see Boyd (2004). Therefore, we will approximate the actual density \mathbf{p} as

$$\mathbf{p}(\mathbf{x}) \approx \Phi \boldsymbol{\omega}(\mathbf{x}) \quad (17)$$

B. The filter approximation

We describe how to construct an approximation to the actual unnormalized density $p_{k,k}^l(\mathbf{x})$ at time t_k from an existing set of approximations to the unnormalized density $\{p_{k-1,k-1}^r(\mathbf{y})\}_{r=1,\dots,d}$ available from time t_{k-1} . For any $r = 1, \dots, d$, let us denote by $\hat{p}_{k-1,k-1}^r(\mathbf{y})$ such an approximation which would have been computed in the previous step of the approximation procedure. Let

$$\Omega_k^l = \{(i, j) \in I \times J : [\boldsymbol{\omega}_k^l]_{(i,j)} \neq 0\} \quad (18)$$

where $\boldsymbol{\omega}_k^l$ is the solution of (P2) obtained when approximating $p_{k,k}^l(\mathbf{x})$, i.e.

$$\hat{p}_{k-1,k-1}^r(\mathbf{y}) \approx \sum_{(i,j) \in \Omega_{k-1}^r} [\boldsymbol{\omega}_{k-1}^r]_{(i,j)} n_{i,j}(\mathbf{y}) \quad (19)$$

Notice that we are only propagating from time t_{k-1} to time t_k the Gaussian densities in B with nonzero weight. Our goal is to approximate the actual filter density $p_{k,k}^l(\mathbf{x})$ at time t_k given by Eq. (9). This is done in two separate steps.

The output of the first step is an approximate density $\hat{p}_{k,k}^l(\mathbf{x})$ computed as follows:

$$\begin{aligned} p_{k,k}^l(\mathbf{x}) &= L_{k|k}^l(\mathbf{x}) \sum_{r=1}^d \int_{\mathbb{R}^n} \lambda^{r,l}(\mathbf{y}) p(\mathbf{x}|\mathbf{y}, l) p_{k-1,k-1}^r(\mathbf{y}) d\mathbf{y} \\ &\approx L_{k|k}^l(\mathbf{x}) \sum_{r=1}^d \int_{\mathbb{R}^n} \lambda^{r,l}(\mathbf{y}) p(\mathbf{x}|\mathbf{y}, l) \hat{p}_{k-1,k-1}^r(\mathbf{y}) d\mathbf{y} \quad (20) \\ &:= \check{p}_{k,k}^l(\mathbf{x}) \end{aligned}$$

where the approximation step consists in replacing each term $p_{k-1,k-1}^r(\mathbf{y})$ with its previously computed approximation $\hat{p}_{k-1,k-1}^r(\mathbf{y})$. We can write out explicitly each term in the approximation, thus obtaining

$$\begin{aligned} \check{p}_{k,k}^l(\mathbf{x}) &= L_{k|k}^l(\mathbf{x}) \sum_{r=1}^d \sum_{(i,j) \in \Omega_{k-1}^r} [\boldsymbol{\varpi}_{k-1}^r]_{(i,j)} \\ &\quad \cdot \int_{\mathbb{R}^n} \lambda^{r,l}(\mathbf{y}) p(\mathbf{x}|\mathbf{y}, l) n_{i,j}(\mathbf{y}) d\mathbf{y} \quad (21) \end{aligned}$$

Eq. (21) shows that we have obtained an approximation density $\check{p}_{k,k}^l(\mathbf{x})$ for $p_{k,k}^l(\mathbf{x})$ consisting of $d \cdot \sum_{r=1}^d |\Omega_{k-1}^r|$ components. Moreover, $\check{p}_{k,k}^l(\mathbf{x})$ is no longer a mixture of Gaussian due to the appearance of the state-dependent mode transition term $\lambda^{r,l}(\mathbf{y})$. To this purpose, we approximate $\check{p}_{k,k}^l(\mathbf{x})$ further before propagating it to the next sampling time t_{k+1} . This is the second step of our methodology, which is described next. We first numerically compute the mean $\check{\mathbf{x}}_{k,k}^l$ and covariance matrix $\check{\boldsymbol{\Sigma}}_{k,k}^l$ of the unnormalized density $\check{p}_{k,k}^l(\mathbf{x})$. We next choose an n -dimensional ellipsoid of uncertainty $G \in \mathbb{R}^n$ defined as

$$(\mathbf{x} - \check{\mathbf{x}}_{k,k}^l)' (\check{\boldsymbol{\Sigma}}_{k,k}^l)^{-1} (\mathbf{x} - \check{\mathbf{x}}_{k,k}^l) \quad (22)$$

Both the set \mathcal{X} of training points and the means of the Gaussian densities in the base set B are chosen in the region G . The covariance matrices of the Gaussian densities in B are instead assumed to be the same at all stages. Further implementation details for a specific example will be presented in Section V.

We next evaluate the density $\check{p}_{k,k}^l(\mathbf{x})$ on the set \mathcal{X} of training points, thus obtaining a vector

$$\check{\mathbf{p}}_{k,k}^l = (\check{p}_{k,k}^l(\mathbf{x}_1), \dots, \check{p}_{k,k}^l(\mathbf{x}_q)) \quad (23)$$

Let ϵ_k^l be the smallest component of the vector $\check{\mathbf{p}}_{k,k}^l$. Then we solve the optimization problem (P2) as follows:

$$\min \|\boldsymbol{\varpi}\|_1 \quad \text{subject to} \quad \|\check{\mathbf{p}}_{k,k}^l - \Phi \boldsymbol{\varpi}\|_2 \leq \epsilon_k^l \quad (\text{P2})$$

If $\boldsymbol{\varpi}_k^{l,*}$ denotes the optimal solution to (P2), the approximation density at step k , which is propagated to step $k+1$ is

$$\hat{p}_{k,k}^l(\mathbf{x}) = \sum_{(i,j) \in \Omega_k^l} [\boldsymbol{\varpi}_k^{l,*}]_{(i,j)} n_{i,j}(\mathbf{x}) \quad (24)$$

The total variation distance between the optimal unnormalized filter density $p_{k,k}^l(\mathbf{x})$ and its computed approximation $\hat{p}_{k,k}^l(\mathbf{x})$ may be controlled through an analytical bound provided in [6]. We summarize the all procedure with a block diagram of the estimator in Figure 1.

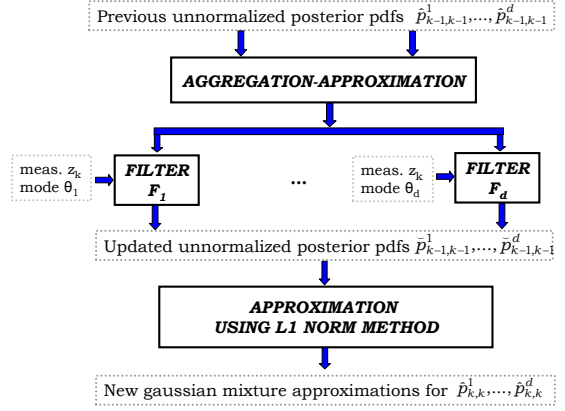


Fig. 1. One cycle of the estimator.

C. Computational requirements

We compare the computational complexity of our methodology with the one of the IMM estimator. The overall complexity of the IMM algorithm is $O(d^2 + n^3)$. Our approach has higher computational load which comes from the amount of time needed to set up and solve the second order cone programming problem (P2). Before solving the optimization problem, it is required to compute $\check{p}_{k,k}^l$ for any mode l and this has complexity $O(d^2 |I| |J| q)$. This is because $\check{p}_{k,k}^l$ in Eq. (21) may be written as

$$\check{p}_{k,k}^l(\mathbf{x}) = \sum_{r=1}^d \sum_{(i,j) \in \Omega_k^r} [\boldsymbol{\varpi}_{k-1}^r]_{(i,j)} \mathbb{E}_{n_{i,j}} [L_{k|k}^l(\mathbf{x}) \lambda^{r,l}(\mathbf{Y}) p(\mathbf{x}|\mathbf{Y}, l)] \quad (25)$$

where \mathbf{Y} is a gaussian random vector with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_j$. For each \mathbf{x} in the training set it is required to compute $d|I||J|$ Gaussian integrals, each of which computable in constant time using Hermite quadrature. This has to be repeated for each mode, thus the total complexity is $O(d^2 q |I| |J|)$. Then the optimization problem needs to be solved for each mode, and this can be done in time $O((|I||J|)^2 q)$ using the algorithm proposed in [9].

V. APPLICATION TO TARGET TRACKING

We evaluate the prediction accuracy and the computational power of our scheme on a target tracking problem. The considered scenario consists of an aircraft flying in the (x^1, x^2) plane. The planned flight path consists of travelling along the vertical direction and execute a 90° right turn (with turn rate of $\frac{3^\circ}{s}$) when the trajectory change point (TCP) is reached, see Figure 2 for an illustration. The sensor sampling period is 10 seconds. After the turn, the aircraft continues straight along the horizontal direction. However, in practice the aircraft can turn earlier at a difference trajectory change point (TCA) and we assume that the distance TCA - TCP at which the trajectory change point occurs is Gaussian with mean zero and standard deviation $\sqrt{10}m$. Therefore, the further the aircraft is from the expected TCP, the less likely it is to change its trajectory.

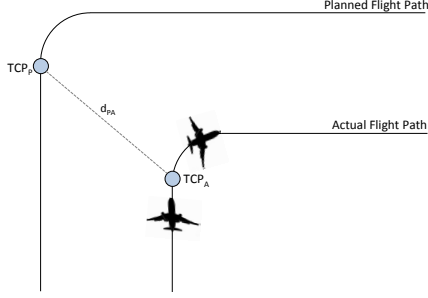


Fig. 2. Actual trajectory versus expected trajectory of the aircraft

This simple scenario is challenging because the deviation from non-maneuvering motion is very large, while the duration of the maneuver is extremely short. The second order system for this motion, with position only measurements is:

$$\begin{aligned} \mathbf{x}_{k+1} &= \text{diag}(\mathbf{A}, \mathbf{A})\mathbf{x}_k + \text{diag}(\mathbf{\Gamma}, \mathbf{\Gamma})\mathbf{w}_k \\ \mathbf{z}_k &= \text{diag}(\mathbf{H}, \mathbf{H})\mathbf{x}_k + \mathbf{v}_k \end{aligned} \quad (26)$$

where

$$\begin{aligned} \mathbf{x}_k &= [x_k^1 \ x_k^1 \ x_k^2 \ \dot{x}_k^2]' & \mathbf{z}_k &= [z_k^1 \ z_k^2]' \\ \mathbf{A} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} & \mathbf{\Gamma} &= \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} & \mathbf{H} &= [1 \ 0] \end{aligned}$$

The probability of switching from the straight mode to the maneuvering mode, here denoted as $\lambda^{1,2}(\mathbf{x})$, is defined as

$$\begin{aligned} \lambda^{1,2}(\mathbf{x}) &:= \lambda^{1,2}(x^1, x^2) \\ &= \alpha_{1,2} + \beta_{1,2}n([x^1, x^2], [0, L], \mathbf{M}) \end{aligned} \quad (27)$$

In Eq. (27) we want to emphasize that the probability of turning is only dependent on the (x^1, x^2) aircraft position and it is independent of the other state components. For this experiment, we set $\alpha_{1,2} = 0.03$, $\beta_{1,2} = 16\pi$, and

$$\mathbf{M} = \begin{pmatrix} 10m & 0 \\ 0 & 10m \end{pmatrix} \quad (28)$$

Here α is modeling the probability that the aircraft turns at a different point than TCP, while $\beta_{1,2}$ models the probability that the aircraft turns at the expected TCP.

The two process noise sequences are zero-mean and only differ in the choice of the process noise covariance which is set to a two dimensional diagonal matrix with equal entries. For the straight mode (also referred to as mode 1) such entries are set to $(0.3 \frac{m}{s^2})^2$ and for the maneuvering mode (also referred to as mode 2) such entries are set to $(6 \frac{m}{s^2})^2$. The covariance matrix of the measurement noise is a two dimensional matrix with diagonal entries equal to $(100m)^2$.

For the true dynamics, the process noise covariance is assumed to be a two-dimensional diagonal matrix having both entries equal to $(0.4 \frac{m}{s^2})^2$. The maneuver is obtained by changing the mean of the process noise $\mathbb{E}[\mathbf{w}_k]$ from time $k = 7$ to $k = 9$ as

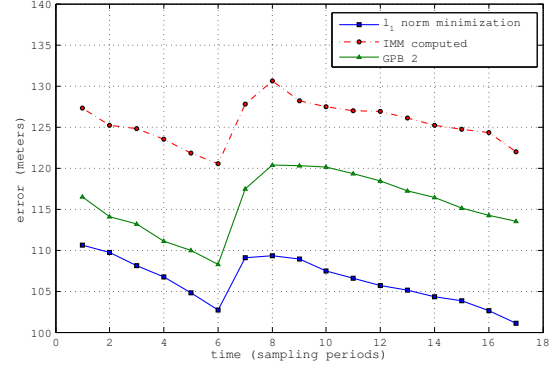


Fig. 3. Coordinate-combined position estimation error

$\mathbb{E}[\mathbf{w}_7] = \mathbb{E}[\mathbf{w}_9] = \begin{bmatrix} 6 \frac{m}{s^2} \\ -1.61 \frac{m}{s^2} \end{bmatrix}$ $\mathbb{E}[\mathbf{w}_8] = \begin{bmatrix} 4.39 \frac{m}{s^2} \\ -4.39 \frac{m}{s^2} \end{bmatrix}$
At all other times $\mathbb{E}[\mathbf{w}_k] = 0$. We carry out one-hundred Monte-Carlo simulations by generating aircraft trajectories randomly distributed around the planned flight path according to a Gaussian distribution with zero mean and standard deviation of 5 meters. The set of training points in \mathcal{X} are chosen in such a way that any two adjacent points in the uncertainty ellipsoid G in Eq. (22) are equidistant. The means of the Gaussian densities in B are chosen to coincide with the training points in \mathcal{X} . The covariance matrices are chosen as follows. We specify the standard deviation of each state space component σ_i , $i \in 1, \dots, n$, and the correlation factors between them, $\rho_{i,j}$, $i, j \in 1, \dots, n$. The values for σ_i in each covariance matrix is assumed to be identical for all i . Those values decay exponentially from a value σ_{\max} for the first covariance matrix to a value σ_{\min} for the last covariance matrix. The correlation factors $\rho_{i,j}$ are equally spaced in the interval $(-1, 1)$. The combination of all these parameters resulted in a base set B of about 700 Gaussian densities and a training set \mathcal{X} of about 49 training points when averaged out across all sampling times.

We compare our method with the IMM and GPB2 estimator and the reported results are obtained through an average of one-hundred Monte-Carlo runs. The transition probability function used within IMM and GPB2 is

$$\lambda(\mathbb{E}[\mathbf{x}_k | \mathbf{Z}^{k-1}]) \quad (29)$$

i.e. the function λ in Eq. (27) is evaluated at the state predicted at time k on the basis of the the observations \mathbf{Z}^{k-1} .

Figures 3 and 4 report the comparison of our filtering scheme with the IMM estimator on one-hundred Monte-Carlo simulations and show that our approximation scheme improves the tracking accuracy with respect to both IMM and GPB2. This is partly due to the fact that our method always propagates more than two densities at each sampling step, whereas the IMM always propagates only two of them per time step. Our method also appears to exhibit a better tracking performance during the maneuvering stage which may be explained by the fact that it is considering actual state dependent transitions (using Eq. (27)), whereas IMM and GPB2 consider estimated state dependent transitions (using Eq. (29)). Figure 5 reports the number of components

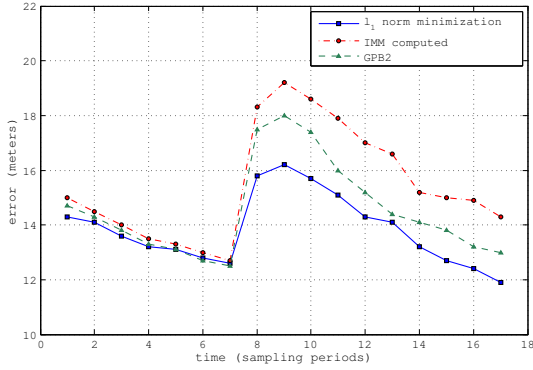


Fig. 4. Coordinate-combined velocity estimation error

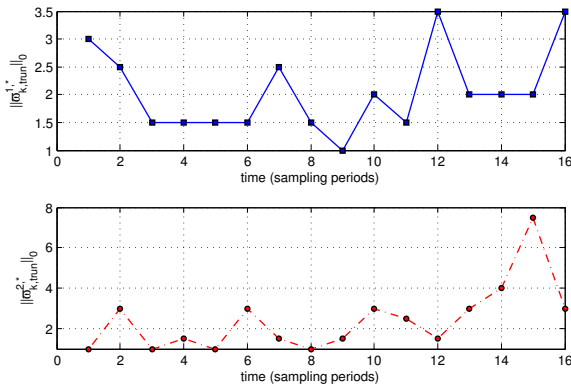


Fig. 5. The number of non-zero Gaussian components used in the density approximations $\hat{p}_{k,k}^1(x)$ and $\hat{p}_{k,k}^2(x)$

of the vector $\omega_k^{l,*}$ which are significantly different from zero in each sampling time t_k . More specifically, at each sampling time, we observe that there are few entries of $\omega_k^{l,*}$ with a large order of magnitude and many remaining entries which are smaller than those by a factor larger than 10^{-6} . Our computational scheme does not propagate these components, thus they are not counted in the l_0 norms of $\omega_k^{1,*}$ and $\omega_k^{2,*}$, which are reported in Figure 5. We denote these truncated l_0 norms respectively by $\|\omega_{k, trunc}^{1,*}\|_0$ and $\|\omega_{k, trunc}^{2,*}\|_0$.

This evidences a feature of our method, the fact that it does not put any hard constraint on the number of densities to propagate, but it is adaptive to the situation at hand. In cases when the situation is “easy”, then it does not use many of them, but if the situation is “hard” it compensates by propagating more densities. It also emerges clearly from Figure 5 that the number of Gaussian densities propagated is always reasonably small (it never exceeds 10), thereby showing that the recovered solution is sufficiently sparse.

With respect to the computational time, our method performs worse than the IMM and GPB2 estimator. For

each Monte-Carlo run, we took the average time needed to compute the state estimate over all sampling intervals. We then averaged out those times across all Monte-Carlo runs. Times are measured on a 2.20GHz processor with 4Gb of RAM. The times required by IMM, GPB2 and by the proposed method are respectively 5.0 msec, 10 msec and 150 msec, showing that our method is slower than IMM by a factor of 30 and than GPB2 by a factor of 15.

VI. CONCLUSIONS

In this paper we have presented a new approximation scheme to compute the filter density in jump linear systems with state dependent transition probabilities between modes. Such scheme obtains an approximation density for each unnormalized posterior density by solving a second-order cone programming problem which trades off sparsity and accuracy. By means of a target tracking study case, we have shown that our method can outperform the IMM and GPB2 estimator, although this occurs at the expenses of an increase in computational time. In the future we plan to compare our method with particle filtering and investigate which families of mode switching probabilities lead to filter densities close to a low dimensional Gaussian mixture, thus allowing sparse gaussian mixture approximations computable using the proposed methodology.

REFERENCES

- [1] Bar-Shalom, Y. and Li, X.-R. Estimation and Tracking: Principles, Techniques and Software, Artech House, Boston- London, 1993.
- [2] Basseville, M. and Nikiforov I. Detection of Abrupt Changes: Theory and Application. Prentice Hall, 1993.
- [3] Blom, H. and Bar-Shalom, Y. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transactions on Automatic Control*, vol. 33, 1988, pp. 780–783.
- [4] Blom, H. and Bloem, E. Exact Bayesian and Particle Filtering of Stochastic Hybrid Systems. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, 2007, pp. 55–70.
- [5] Boyd, S. and Vandenberghe, L. Convex Optimization. Cambridge University Press, 2004.
- [6] Capponi, A. A Convex Optimization Approach to Filtering in Jump Systems with State Dependent Transitions. Accepted in *Automatica*, 2009.
- [7] Capponi, A. and Cvitanic, J. Credit Risk Modeling with Misreporting and Incomplete Information. *International Journal of Theoretical and Applied Finance*, vol. 12, 2009.
- [8] Doucet, A., Gordon, N. and Krishnamurthy, V. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, vol. 49, 2001, pp. 613–624.
- [9] Lobo, M., Vanderberghe, L., Boyd, S. and Lebret H. *Linear Algebra and its Applications*, vol. 284, 1998, pp. 193–228.
- [10] Mazor, E., Averbuch, A., Bar-Shalom, Y. and Dayan, J. Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, 1998, pp. 103–123.
- [11] Rong Li, X. and Jilkov, V.P. A Survey of Maneuvering Target Tracking: Multiple-Model Methods. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, 2005, pp. 1255–1321.
- [12] Rozowskii, B., Petrov, A., and Blazek, R. Interacting Banks of Bayesian Matched Filters. In *Proceedings of Signal and Data Processing of Small Targets (SPIE)*, vol. 4048, 2000, pp. 122–133.
- [13] Salmond D. Mixture reduction algorithms for target tracking in clutter. In *Proceedings of Signal and Data Processing of Small Targets (SPIE)*, 1990, vol. 1305, pp. 434–445.