

# Inference Methods for Autonomous Stochastic Linear Hybrid Systems <sup>\*</sup>

Hamsa Balakrishnan, Inseok Hwang, Jung Soon Jang, and Claire J. Tomlin

Hybrid Systems Laboratory  
Department of Aeronautics and Astronautics  
Stanford University, Stanford, CA 94305, U.S.A.  
hamsa,ishwang,jsjang,tomlin@stanford.edu

**Abstract.** We present a parameter inference algorithm for autonomous stochastic linear hybrid systems, which computes a maximum-likelihood model, given only a set of continuous output data of the system. We overcome the potentially intractable problem of identifying the sequence of discrete modes by using dynamic programming; we then compute the maximum-likelihood continuous models using an Expectation Maximization technique. This allows us to find a maximum-likelihood model in time that is polynomial in the number of discrete modes as well as in the length of the data series. We prove local convergence of the algorithm. We also propose a novel initialization technique to derive good initial conditions for the model parameters. Finally, we demonstrate our algorithm on some examples - two simple one-dimensional examples with simulated data, and an application to real flight test data from a dual-vehicle demonstration of the Stanford DragonFly Unmanned Aerial Vehicles.

## 1 Introduction

The modeling of systems as stochastic hybrid systems has applications in fields such as target-tracking, the statistical analysis of time-series data, and systems biology. These systems frequently exhibit behavior that is a combination of discrete switches and continuous evolution; in addition, the data available in these applications is usually corrupted by noise. Most target-tracking algorithms for maneuvering targets, as well as estimators for hybrid systems, depend on the prior knowledge of a good model for the plant dynamics and noise characteristics, as well as knowledge of the transition probabilities between the discrete modes [1, 2]. In this paper, we formulate an algorithm that finds the maximum-likelihood values of parameters for both the continuous dynamics in each mode, and the transition probabilities between the modes. We draw broadly on several techniques from data association and target tracking [3], motion-capture and synthesis methods in computer graphics [4, 5] and statistical time series

---

<sup>\*</sup> This work is supported by ONR under MURI contract N00014-02-1-0720, by DARPA under Software Enabled Control (AFRL contract F33615-99-C-3014) and by an NSF Career Award. H. Balakrishnan is supported by a Stanford Graduate Fellowship.

analysis [6]. While these techniques are related to classical methods of system identification for continuous- and discrete-time systems [7, 8], we use them to develop a method to identify the parameters of a hybrid system. Given only the continuous- or discrete-time output of the system, our algorithm iteratively computes the maximum-likelihood parameters for the discrete and continuous models, and converges to a local maximum-likelihood autonomous stochastic linear hybrid system model. We propose methods to derive good initial conditions, so that the local maximum converged to is a suitable model for tracking the future behavior of the system.

## 2 Model Structure

We consider a class of hybrid systems, with linear stochastic dynamics in each discrete mode. An autonomous discrete-time stochastic linear hybrid system [9] is defined to be:

$$H : \begin{cases} x(k+1) = A_i x(k) + w_i(k) \\ y(k) = Cx(k) + v_i(k) \end{cases}, k \in \mathbb{N} \quad (1)$$

where  $x \in \mathbb{R}^n$  and  $y \in \mathbb{R}^p$  are the continuous state and output variables respectively. The index  $i \in \{1, 2, \dots, N\}$  represents the discrete state, where  $N$  is the (unknown, but finite) number of discrete modes in the model. The system matrices are  $A_i \in \mathbb{R}^{n \times n}$  for  $i \in \{1, 2, \dots, N\}$  (assumed unknown), and  $C \in \mathbb{R}^{p \times n}$  is the measurement matrix ( $n$  and  $C$  can be determined using a Singular Value Decomposition (SVD) on the output data, and are therefore assumed to be known). We denote the covariance of the initial state  $x(k_0)$  as  $\pi_0 \in \mathbb{R}^N$ , and assume that the process noise  $w_i(k)$  and the measurement noise  $v_i(k)$  are uncorrelated, zero-mean white Gaussian sequences with the unknown covariance matrices  $\mathbb{E}[w_i(k)w_i(k)'] = Q_i \in \mathbb{R}^{n \times n}$  and  $\mathbb{E}[v_i(k)v_i(k)'] = R_i \in \mathbb{R}^{p \times p}$  respectively, where  $\mathbb{E}[\cdot]$  and  $(\cdot)'$  denote expectation and matrix transpose. It is assumed that  $w_i(k)$  and  $v_i(k)$  are both uncorrelated with the initial state, i.e.,  $\mathbb{E}[x(k_0)w_i(k)'] = \mathbb{E}[x(k_0)v_i(k)'] = 0$ .  $Y_{1:T} \in \mathbb{R}^{p \times T}$  is used to represent the given data series, that is, the sequence of  $p$ -vectors  $[y(1), y(2), \dots, y(T)]$ . We use similar notation for other sequences of vectors; for example, the sequence of state vectors  $[x(1), x(2), \dots, x(t)]$  is denoted by  $X_{1:t} \in \mathbb{R}^{n \times t}$ . We denote the set of parameters which defines the dynamics for each discrete mode  $i$  by  $\theta_i = \{A_i, Q_i, R_i\}$  and the entire continuous model by  $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$ .

Given only  $Y_{1:T}$ , we would like to find a model that maximizes the likelihood ( $\mathcal{L}$ ) that the data was generated by this model. To do this, we segment  $[1, T]$  into the *best*  $N_S$  ( $\geq N$ ) segments, such that each segment corresponds to a single discrete mode, allowing for modes to be repeated in the data sequence. We label segment  $k$  as  $l_k$ , representing its discrete mode, ( $l_k \in \{1 \dots N\}, k = 1 \dots N_S$ ). We impose the condition that the system stays in a mode for a given minimum *dwell time*,  $T_d$ : this constraint reflects the observation that physical systems do not exhibit infinitely fast switching; additionally, we have shown in [10] that a minimum dwell time is necessary to estimate the state of a hybrid system,

once the model is given. We also compute the switching times between modes, denoted  $s_k$ , such that segment  $k$  spans the time interval  $[s_k, s_{k+1} - 1]$ , and in this segment, the system is in mode  $l_k$ . The minimum dwell time constraint can be expressed as  $s_{k+1} - s_k \geq T_d$ . We denote the switching time sequence by  $S = \{s_1, s_2, \dots, s_{N_S}\}$ , and the labeling sequence by  $L = \{l_1, l_2, \dots, l_{N_S}\}$  (see Figure 1). We assume in this paper that the discrete transitions are independent of the continuous state of the system, the relaxation of this assumption will be the subject of future work. We also assume that the discrete transitions are Markovian, and we define the Markovian switching matrix  $M$ , whose elements are  $M_{ij} = \text{prob}(l_k = j | l_{k-1} = i)$ .  $M$  gives the probability of transition to any mode at the switching time. The Markovian assumption is reasonable, since systems frequently exhibit probabilistic patterns in their switching behavior - for example, a civilian aircraft is more likely to transition from a turn maneuver mode to a straight mode than to another maneuver mode. The optimal segmentation of the data sequence into  $N_S$  segments can also provide us with the maximum likelihood value of  $M$ , so we represent our discrete model by  $D = \{S, L, M\}$ . We use  $\Omega_{(\cdot)}$  to denote the parameter space of  $(\cdot)$ , for example,  $\Omega_y$  represents the parameter space of  $y$ , and is equal to  $\mathbb{R}^p$ . Finally, we define the function  $\delta(\text{statement})$  to be 1, if *statement* is true; 0, otherwise. In summary, given a data series (of length  $T$ ) denoted by  $Y_{1:T}$ , and knowing the measurement matrix ( $C$ ), we would like to find the system parameters, continuous ( $\{A_i, Q_i, R_i\}, i = 1 \dots N$ ) as well as discrete ( $\{S, L, M\}$ ), such that the resultant model maximizes the likelihood that the data series was generated by the model.

For a stochastic linear hybrid system, there are an infinite number of continuous models that can realize a given output sequence. Since the main problem of interest to us is the design of hybrid estimators, we restrict ourselves to the class of systems which satisfy the following conditions: (1) We consider the innovations form of the model [7, 11], given by  $\hat{x}(k+1) = A_i \hat{x}(k) + K_i(k) e_i(k)$ ;  $y(k) = C_i \hat{x}(k) + e_i(k)$ , where the innovations  $e_i(k) = y(k) - C_i \hat{x}(k)$ ,  $K_i(k)$  is the Kalman filter gain and  $\hat{x}$  is the state estimate, and we assume  $A_i - K_i C_i$  is stable [11]. (2) We require that the stochastic linear hybrid system be *identifiable*. This condition, while not overly restrictive, roughly ensures that the models are distinct enough so as to be distinguished from each other using their continuous outputs. This would also give us a class of transformations for the continuous model which would explain the given output. The conditions for identifiability for a deterministic linear hybrid system have been derived in [12], these can be extended quite naturally to stochastic linear hybrid systems.

### 3 Parameter Inference Algorithm for Stochastic Linear Hybrid Systems

In this section, we propose an algorithm for hybrid system model inference, assess its complexity, and prove its convergence to a local optimum. The structure of the algorithm (for one iteration) is given in Figure 2.

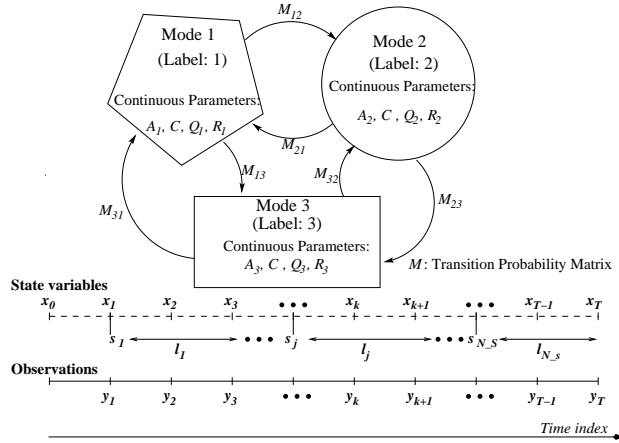


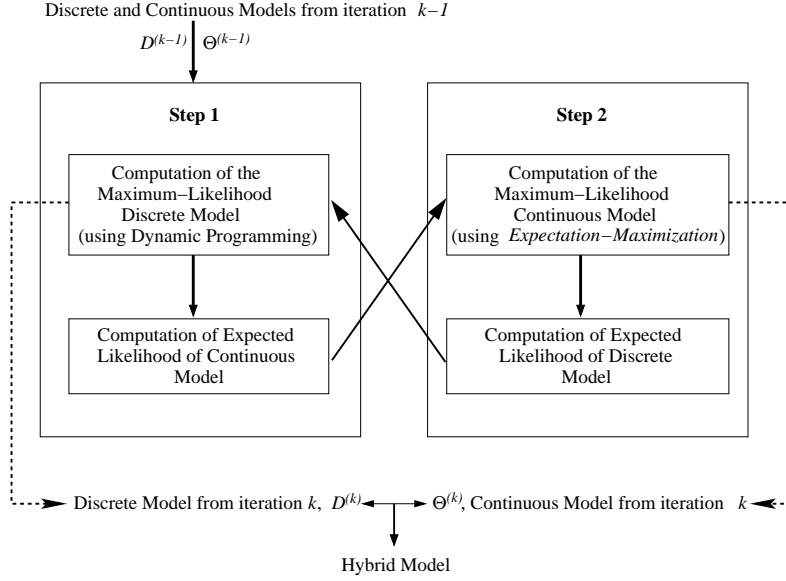
Fig. 1. Example of a three mode hybrid model, showing model parameters.

### 3.1 Expectation Maximization

For the sake of clarity, we first briefly describe the traditional Expectation-Maximization (EM) algorithm [13]. Given general observed data  $Y \in \Omega_Y$ , we could postulate probability density functions (pdfs)  $g(Y|\psi)$ , which depend on parameters  $\psi$ . The Expectation-Maximization algorithm is a method of finding a valuation of  $\psi$  that maximizes  $g(Y|\psi)$  given the observed  $Y$ . Suppose we knew that the observations were in fact *incomplete* data, for which the underlying *complete* data  $Z \in \Omega_Z$  had corresponding pdfs  $f(Z|\psi)$ . (For example, in the case of a stochastic linear system, the observations form the incomplete data, while the observations together with the state variables form the complete data). Denoting the vector of parameters by  $\Psi = (\psi_1, \dots, \psi_d)$ , if the mapping from  $\Omega_Z \mapsto \Omega_Y$  were many to one, we could write

$$g(Y|\Psi) = \int_{\Omega_Z(y)} f(Z|\Psi) dZ \quad (2)$$

To find the model that is most likely to have generated the data, the EM algorithm iteratively computes  $\log \mathcal{L}(\Psi) \triangleq \log f(Z|\Psi)$ , updating the parameter fit ( $\Psi$ ) at every step. We compute  $\log \mathcal{L}(\Psi)$  instead of  $\mathcal{L}(\Psi)$ , because for many exponential families, including systems which are Gaussian, the logarithm of the likelihood is easier to compute than the likelihood itself; and since the logarithm is a monotonic function, maximizing the log-likelihood is equivalent to maximizing the likelihood. However the complete data  $Z$  is not available in practice (we only have the data we actually measure), and so we use the expectation of the log-likelihood, *i.e.*,  $E_\Psi \{\log \mathcal{L}(\Psi)|Y\}$ , derived using the current fit for  $\Psi$ . For example, in the case of stochastic linear systems, these expectations can be computed using Kalman recursions. Therefore it is the expectation of the likelihood that is maximized; hence the name *Expectation-Maximization*.



**Fig. 2.** Structure of the Parameter Inference Algorithm for Stochastic Linear Hybrid Systems.

**Algorithm 1 (Classical (G)EM Algorithm [13])**

Let  $\Psi^{(0)}$  be some initial value of  $\Psi$ .

**Repeat:**

1. **E- Step:** Compute

$$Q(\Psi|\Psi^{(k)}) \triangleq E_{\Psi^{(k)}} \{ \log \mathcal{L}(\Psi) | Y \} = E_{\Psi^{(k)}} \{ \log f(Z|\Psi) | Y \}, \quad (3)$$

where  $E_{\Psi}(\cdot)$  denotes the expectation derived using the parameter set  $\Psi$ .

2. **M- Step:** Choose  $\Psi^{(k+1)}$  to be any value of  $\Psi \in \Omega_{\Psi}$  which maximizes  $Q(\Psi|\Psi^{(k)})$ , i.e.,

$$Q(\Psi^{(k+1)}|\Psi^{(k)}) \geq Q(\Psi|\Psi^{(k)}), \text{ for all } \Psi \in \Omega_{\Psi} \text{ (denoted EM)}; \quad (4)$$

or any value that satisfies the less restrictive condition:

$$Q(\Psi^{(k+1)}|\Psi^{(k)}) \geq Q(\Psi^{(k)}|\Psi^{(k)}) \text{ (denoted Generalized EM (GEM))}. \quad (5)$$

**until**  $\mathcal{L}(\Psi^{(k+1)}) - \mathcal{L}(\Psi^{(k)})$  converges to  $\epsilon$ , where  $\epsilon \in \mathbb{R}$  is arbitrarily small.

The EM algorithm is a special case of the GEM algorithm (5). The EM algorithm maximizes the conditional expectation at every iteration, while the GEM only ensures its increase. Under fairly general conditions, such as the boundedness of the likelihood functions, several properties of the EM and GEM algorithms, including convergence, have been proved in [13, 14]. In particular, it can be shown

that a sequence of likelihoods obtained using the GEM algorithm will converge to a local optimum [14]. We will use this property to prove the convergence of the algorithm that we propose for stochastic linear hybrid systems, which have Gaussian and bounded likelihoods.

The two main issues with the implementation of any version of the Expectation-Maximization algorithm are: (1) the pdf  $f(Z|\Psi)$  (or its conditional expectation) is sometimes difficult to compute; (2) since the convergence is to a local maximum, a good choice of initial conditions is necessary. We develop methods to alleviate both of these problems in the application of EM to the inference of stochastic linear hybrid systems.

### 3.2 Parameter Inference Algorithm for Stochastic Hybrid Systems

In order to fit into the form of the standard EM algorithm [13], we note that we wish to find the maximum likelihood solution to our model parameters, *i.e.*,

$$\{\bar{D}, \bar{\Theta}\} = \arg \max_{\{D, \Theta\}} \mathcal{L}(Y_{1:T}|D, \Theta) \quad (6)$$

$\mathcal{L}(Y_{1:T}|D, \Theta)$  is analogous to the function  $g(Y|\psi)$  in (2). As in the case of the classical EM algorithm, rather than work with this density function, we define the *complete* data as the combination of the sequence of state variables ( $X$ ) and the observations ( $Y$ ), from system (1), and denote it by  $Z$ . In other words,  $z(k) = \{x(k), y(k)\} \in \mathbb{R}^n \times \mathbb{R}^p$ . The algorithm can be written as follows:

**Algorithm 2 (Parameter Inference Algorithm for Stochastic Linear Hybrid Systems)**

*Assume an initial continuous model  $\Theta^{(0)}$  and an initial discrete model  $D^{(0)}$ . Iterate the following until the convergence of the likelihood to a local maximum:*

1. **Step 1:** *Find the globally optimal segmentation points ( $S$ ) and their respective labels ( $L$ ), assuming the model parameters of the current iteration ( $k$ ). Then, update the switching probability matrix  $M$ , using:  
 $M_{ij} = \sum_{k=2}^{N_S} \delta(l_{k-1} = i)\delta(l_k = j)$ , normalized such that  $\sum_{j=1}^{N_S} M_{ij} = 1$ .  
This gives us the maximum-likelihood discrete model,  $D^{(k+1)}$ .*
2. **Step 2:** *Fit new maximum-likelihood models into the segmented time sequences; *i.e.*, for the computed  $\{S, L\}$  fit the best  $\Theta^{(k+1)}$ .*

Drawing an analogy to the classical EM algorithm (Algorithm 1):  $\Psi = \{D, \Theta\}$ . Then, we would like to maximize the expectation

$$\mathcal{Q}(\Psi|\Psi^{(k)}) = \mathbb{E}_{\Psi^{(k)}} \{\log \mathcal{L}(\Psi)|Y\} = \mathbb{E}_{D^{(k)}, \Theta^{(k)}} \{\log f(Z|D, \Theta)|Y\}.$$

We may rewrite Algorithm 2 in terms of the different conditional likelihoods maximized in each step. The algorithm begins with an initialization of  $D^{(0)}$  and  $\Theta^{(0)}$ . Then, we iterate the following until the convergence of  $\mathbb{E}_{D^{(k)}, \Theta^{(k)}} \{\log f(Z|\Psi)|Y\}$  to a local maximum:

In Step 1, we maximize  $E_{D^{(k)}, \Theta^{(k)}} \{\log f(Z|D, \Theta^{(k)})|Y\}$ ; this allows us to compute the conditional expectation  $E_{D^{(k+1)}, \Theta^{(k)}} \{\log f(Z|\Theta, D^{(k+1)})|Y\}$ .

In Step 2, we maximize  $E_{D^{(k+1)}, \Theta^{(k)}} \{\log f(Z|\Theta, D^{(k+1)})|Y\}$ , and compute the conditional expectation  $E_{D^{(k+1)}, \Theta^{(k+1)}} \{\log f(Z|D, \Theta^{(k+1)})|Y\}$ .

Algorithm 2 no longer falls into the classical EM framework, since we do not compute the expectation in one step and maximize it in the next. In Algorithm 2, in Step 1, we assume a continuous model and compute the best discrete model for this continuous model. In Step 2, we assume this discrete model, and compute the maximum-likelihood continuous model. While both steps correspond to either an E-step or an M-step, the expectation computed in one step is maximized in the next, and vice versa. Since the likelihood function is changed at every step, it is not clear that the convergence properties of classical EM or GEM hold in this case, and thus the convergence of Algorithm 2 must be analyzed.

**Theorem 1.** *Algorithm 2 iteratively generates a sequence of models, whose likelihoods satisfy the model-likelihood sequence conditions (5) of the Generalized Expectation-Maximization Algorithm [13, 14]. Therefore the algorithm is guaranteed to converge to a local maximum.*

*Proof.* Let us consider the likelihood sequence generated. For iteration  $k + 1$ , we would like to prove that the model parameters  $\{D^{(k+1)}, \Theta^{(k+1)}\}$  satisfy (5), *i.e.*,

$$\mathcal{Q}(D^{(k+1)}, \Theta^{(k+1)}|D^{(k)}, \Theta^{(k)}) \geq \mathcal{Q}(D^{(k)}, \Theta^{(k)}|D^{(k)}, \Theta^{(k)}). \quad (7)$$

We can rewrite Algorithm 2 as follows:

**Step 1:** Maximize  $E_{D^{(k)}, \Theta^{(k)}} \{\log f(z|D, \Theta^{(k)})|Y_{1:T}\}$ : compute  $D^{(k+1)}$  such that

$$\mathcal{Q}(D^{(k+1)}, \Theta^{(k)}|D^{(k)}, \Theta^{(k)}) \geq \mathcal{Q}(D, \Theta^{(k)}|D^{(k)}, \Theta^{(k)}), \text{ for all } D \in \Omega_D. \quad (8)$$

**Step 2:** Maximize  $E_{D^{(k+1)}, \Theta^{(k)}} \{\log f(z|\Theta, D^{(k+1)})|y\}$ : compute  $\Theta^{(k+1)}$  such that

$$\mathcal{Q}(D^{(k+1)}, \Theta^{(k+1)}|D^{(k)}, \Theta^{(k)}) \geq \mathcal{Q}(D^{(k+1)}, \Theta|D^{(k)}, \Theta^{(k)}), \text{ for all } \Theta \in \Omega_\Theta \quad (9)$$

Combining (8) and (9), we get

$$\begin{aligned} \mathcal{Q}(D^{(k+1)}, \Theta^{(k+1)}|D^{(k)}, \Theta^{(k)}) &\geq \mathcal{Q}(D^{(k+1)}, \Theta|D^{(k)}, \Theta^{(k)}), \text{ for all } \Theta \in \Omega_\Theta \\ &\geq \mathcal{Q}(D^{(k+1)}, \Theta^{(k)}|D^{(k)}, \Theta^{(k)}) \\ &\geq \mathcal{Q}(D^{(k)}, \Theta^{(k)}|D^{(k)}, \Theta^{(k)}), \end{aligned} \quad (10)$$

thus proving (7). Since in the case of stochastic models with Gaussian noise, the likelihood functions are bounded, the sequence will converge to either a stationary point (saddle surface) or a local maximum. Such a convergence to a saddle surface can only occur in the continuous step (Step 2). However, in the case of linear Gaussian systems, we can show that the Hessian is always negative definite, ruling out convergence to a stationary point. Therefore the sequence converges to a local maximum, proving the convergence of the parameter inference algorithm for hybrid systems. (Note that the convergence criterion is the convergence of the likelihood, and not the model parameters [14]. However, we prevent oscillations between different discrete models with identical likelihoods by updating the discrete model only when the likelihood increases.) ■

## 4 Implementation of Parameter Inference Algorithm for Hybrid Systems

Algorithm 2 is of little use if we cannot efficiently compute the likelihoods. In this section, we describe the actual likelihood functions chosen, as well as the procedures used to maximize them. The problem of maximizing the likelihood in Step 1 is potentially intractable, since we need to find the maximum likelihood hypothesis from  $\mathcal{O}(N^T)$  potential segmentations. However, it is fortunately possible in this case to compute the solution, by formulating a dynamic program of polynomial complexity [4]. In Step 2, we need to iteratively compute the model parameters to maximize the likelihood of the dynamics in each of the modes. We use an algorithm which iteratively maximizes this likelihood using a form of the EM algorithm [15].

### 4.1 Step 1: Maximizing the likelihood of the discrete model

Suppose we have initial values of  $\Theta$  and  $D$  (we leave the discussion of details of the initialization for Section 4.3). We want to find globally optimal segmentation points ( $S$ ) and labels ( $L$ ). To achieve this purpose, we employ the following **dynamic programming** algorithm [4]:

Let us define a reward function  $\mathcal{L}max_n(t)$  as the maximum value of likelihood that can be derived from dividing  $Y_{1:t}$  into  $n$  segments.  $\mathcal{L}max_n(t)$  is achieved by the optimal segmentation of  $Y_{1:t}$  into  $n$  parts. Let us define  $LastMode_n(t)$  and  $LastStart_n(t)$  to be the label (mode) and start time of the final segment in this optimal segmentation.

Clearly, the optimal segmentation is the one that maximizes the likelihood of the entire data set, *i.e.*, has likelihood  $\max_{1 \leq n \leq \lfloor \frac{T}{T_d} \rfloor} \mathcal{L}max_n(T)$ , and  $N_S$  is the number of segments in this “optimal” segmentation. Then, for  $T_d \leq t \leq T$ ,

$$\mathcal{L}max_1(t) = \max_{1 \leq i \leq N} \mathcal{L}(Y_{1:T}|\theta_i), \text{ and } LastMode_1(t) = \arg \max_i \mathcal{L}(Y_{1:T}|\theta_i) \quad (11)$$

Also, while  $1 \leq n \leq \lfloor \frac{T}{T_d} \rfloor$ , and  $nT_d \leq t \leq T$ ,

$$\mathcal{L}max_n(t) = \max_{\substack{1 \leq i \leq N \\ (n-1)T_d < b \leq t - T_d}} [\mathcal{L}max_{n-1}(b-1)M_{li}\mathcal{L}(Y_{b:t}|\theta_i)]$$

$$[LastMode_n(t), LastStart_n(t)] = \arg \max_{i,b} [\mathcal{L}max_{n-1}(b-1)M_{li}\mathcal{L}(Y_{b:t}|\theta_i)] \quad (12)$$

where  $l = LastMode_{n-1}(b-1)$ . In other words, the maximum likelihood derived from segmenting a sequence into  $n$  parts is the maximum product of the likelihood resulting from segmenting a smaller sequence into  $n-1$  parts in the best manner, multiplied by the likelihood of the new sequence, multiplied by the probability of the corresponding mode switch. A schematic representation of this



dynamic programming algorithm is given in Figure 3. Therefore, the required optimal solution to the segmentation is

$$\mathcal{L}max(T) = \max_{1 \leq n \leq \lfloor \frac{T}{T_d} \rfloor} \mathcal{L}max_n(T) \quad (13)$$

and the optimal number of segments is  $N_S = \arg \max_n \mathcal{L}max(T)$ . We can also find the optimal segmentation  $(\{S, L\})$  corresponding to this solution. We note that

$$s_1 = 1; s_{N_S+1} = T + 1; s_n = LastStart_n(s_{n+1} - 1), \text{ for } N_S \geq n > 1 \quad (14)$$

$$l_{N_S} = LastMode_{N_S}(T); l_{n-1} = LastMode_{n-1}(s_n - 1), \text{ for } N_S \geq n > 1 \quad (15)$$

We now also update the switching matrix, using the labels from the optimal segmentation

$$M_{ij} = \sum_{k=2}^{N_S} \delta(l_{k-1} = i) \delta(l_k = j), \text{ normalized such that } \sum_{j=1}^{N_S} M_{ij} = 1.$$

This algorithm solves the optimal segmentation problem with a complexity of  $\mathcal{O}(NT^3)$ , if we know the likelihood function  $\mathcal{L}(Y_{1:t}|\theta_i)$ . In the case of stochastic linear systems with Gaussian noise, it is possible to simply express the likelihood function in terms of the residuals of a Kalman filter. Given  $\theta_i$ , we can design an optimal estimator for the continuous dynamics in the form of a Kalman filter. Given the predictions for the continuous state variables  $E(x(k)|Y_{1:k-1}) = x(k|k-1)$ , and their covariances  $P(k|k-1)$ , we can express the likelihood as :

$$\log \mathcal{L}(Y_{1:t}|\theta_i) = -\frac{1}{2} \sum_{k=1}^t \log |\Sigma_{ki}| - \frac{1}{2} \sum_{k=1}^t (r_i(k))' \Sigma_{ki}^{-1} r_i(k) \quad (16)$$

where  $r_i(k) = y(k) - C_i x(k|k-1)$  is the residual at time  $k$ , with a covariance  $\Sigma_{ki} = C_i P(k|k-1) C_i' + R_i$ . It can be shown ([13, 14]) that maximizing

$$\mathcal{L}max_n(t) = \max_{i,b} \{ \mathcal{L}max_{n-1}(b-1) M_{li} \mathcal{L}(Y(b:t) | \theta_i) \}$$

The diagram shows a horizontal axis representing time from  $s_l$  to  $T$ . A vertical tick mark at  $s_l$  marks the start of a segment. A vertical tick mark at  $b$  marks the end of a segment. A vertical tick mark at  $t$  marks the end of a sub-segment. A vertical tick mark at  $T$  marks the end of the entire time interval. A bracket above the axis from  $s_l$  to  $b$  is labeled  $\mathcal{L}max_{n-1}(b-1)$ . A bracket above the axis from  $b$  to  $t$  is labeled  $\mathcal{L}(Y(b:t) | \theta_i)$ . A curved arrow labeled  $M_{li}$  points from the  $b$  tick mark to the  $t$  tick mark. Below the axis, a bracket from  $s_l$  to  $b$  is labeled  $l = LastMode_{n-1}(b-1)$ . A bracket from  $b$  to  $t$  is labeled  $i$ .

**Fig. 3.** Dynamic Programming Algorithm for Step 1.

$\mathcal{L}(Y_{1:t}|\Theta, M)$  over  $\{\Theta, M\}$  is equivalent to maximizing  $E[\log \mathcal{L}(Z|\{\Theta, M\})|Y]$ , where  $Z$  is the “complete” data, *i.e.*, the joint likelihood of the observed variables ( $Y_{1:t}$ , and the state variables  $X_{1:t}$ ). Given the optimal segmentation  $(S, L)$ , the function we would like to maximize is the sum of the conditional likelihood functions for the data in each segment. To do this, we update the models in Step 2 to be the maximum (conditional) likelihood values for the data of each of the segments.

## 4.2 Step 2: Finding the maximum likelihood continuous model

For each mode, we fit the maximum likelihood model, using a form of the EM algorithm proposed in [15, 6]. We demonstrate this for one of the segments which we (in Step 1) have labeled as mode  $i$ . Suppose the data output of this sequence is  $Y_{b+1:b+n}$ , where  $b+1 < b+n < T$ ,  $n > T_d$ . Then, the desired log-likelihood for this segment can be written as

$$\begin{aligned} \log \mathcal{L}(Z_{b+1:b+n}|\theta_i) = & -\frac{1}{2} \log |\Sigma| - \frac{1}{2} (x(b) - \mu)' \Sigma^{-1} (x(b) - \mu) - \frac{n}{2} \log |Q_i| \\ & - \frac{1}{2} \sum_{k=b+1}^{b+n} (x(k) - A_i x(k-1))' Q_i^{-1} (x(k) - A_i x(k-1)) \\ & - \frac{n}{2} \log |R_i| - \frac{1}{2} \sum_{k=b+1}^{b+n} (y(k) - C_i x(k))' R_i^{-1} (y(k) - C_i x(k)) \end{aligned} \quad (17)$$

where, as before,  $Z_{b+1:b+n}$  is the joint (“complete”) data, namely, the observed variables  $Y_{b+1:b+n}$ , and the continuous state variables,  $X_{b+1:b+n}$ ;  $\mu$  and  $\Sigma$  are the mean and covariance of the initial values in that segment of the continuous state variable, *i.e.*,  $x(b)$ . Then, as explained earlier, the maximum likelihood solution is the one that maximizes the function

$$\mathcal{Q}(\mu, \Sigma, A_i, Q_i, R_i) = \mathbb{E}[\log \mathcal{L}(Z_{b+1:b+n-1}|\theta_i)|Y_{b+1:b+n}] \quad (18)$$

In computing the conditional expectation in (17), we need to compute the following conditional means and covariances, which are easily obtained using Kalman recursions [16]:  $x^b(k|s) = \mathbb{E}[x(k)|Y_{b+1:b+s}]$ ,  $P^b(k|s) = \text{cov}(x(k)|Y_{b+1:b+s})$ , and  $P^b(k, k-1|s) = \text{cov}(x(k), x(k-1)|Y_{b+1:b+s})$ . We define the following quantities:

$$\begin{aligned} \mathcal{A} & \triangleq \sum_{k=b+1}^{b+n} (P^b(k-1|n) + x^b(k-1|n)x^b(k-1|n)') \\ \mathcal{B} & \triangleq \sum_{k=b+1}^{b+n} (P^b(k, k-1|n) + x^b(k|n)x^b(k-1|n)') \\ \mathcal{C} & \triangleq \sum_{k=b+1}^{b+n} (P^b(k|n) + x^b(k|n)x^b(k|n)') \end{aligned}$$

As shown in [15], taking conditional expectations on (17), we get:

$$\begin{aligned} \mathcal{Q}(\mu, \Sigma, A_i, Q_i, R_i) = & -\frac{1}{2} \log |\Sigma| - \frac{1}{2} \text{tr} \left\{ \Sigma^{-1} (P^b(b|n) + (x(b) - \mu)(x(b) - \mu)') \right\} \\ & - \frac{n}{2} \log |Q_i| - \frac{1}{2} \text{tr} \left\{ Q_i^{-1} (\mathcal{C} - \mathcal{B}A_i' - A_i\mathcal{B}' + A_i\mathcal{A}A_i') \right\} \\ & - \frac{n}{2} \log |R_i| \\ & - \frac{1}{2} \text{tr} \left\{ R_i^{-1} \sum_{k=b+1}^{b+n} \left[ (y(k) - C_i x^b(k|n))(y(k) - C_i x^b(k|n))' \right. \right. \\ & \quad \left. \left. + C_i P^b(k|n) C_i' \right] \right\} \end{aligned}$$

where  $\text{tr}$  denotes the trace of a matrix. It can be shown, as in ([15]), that at iteration  $r$ , the choice of parameters that maximizes  $\mathcal{Q}(A_i, Q_i, R_i)$  is:

$$A_i(r+1) = \mathcal{B}A^{-1} \quad (19)$$

$$Q_i(r+1) = \frac{1}{n} \{ \mathcal{C} - \mathcal{B}A^{-1}\mathcal{B}' \} \quad (20)$$

$$R_i(r+1) = \frac{1}{n} \sum_{k=b+1}^{b+n} \left[ (y(k) - C_i x^b(k|n))(y(k) - C_i x^b(k|n))' + C_i P^b(k|n) C_i' \right] \quad (21)$$

Therefore Step 2 consists of the following EM algorithm: For each of the models, for the corresponding optimal segment(s) from the E-step, we repeat the following until the estimates and the log-likelihood converge to a local maximum.

1. Compute the means and covariances of the continuous state estimates using a Kalman filter.
2. Find the new iterates of parameters using (19), (20) and (21).

The converged values of the (continuous) model parameters ( $\Theta$ ) are the maximum likelihood model parameter fits to the segmented data sequence.

As shown in [17, 7], for a large number of samples, the difference between the estimate and the true model (local maximum it converges to) tends in distribution to a zero-mean Gaussian with a covariance given by  $-\left[\frac{\partial^2 \mathcal{L}}{\partial \Theta^2}\right]^{-1}$ . Therefore, the inverse of the Hessian of the likelihood function gives us a measure of the uncertainty of our parameter estimates.

### 4.3 Initialization

We have already mentioned that the EM algorithm is only guaranteed to converge to a local minimum. The optimal solution is a function of how “good” the initial guesses of the parameters are. We present here a novel method of initializing the continuous parameters. Since we know that the system stays in a mode for at least a time  $T_d$ , we fit in the maximum likelihood model for mode 1 by applying the method of the Step 2 described earlier, to  $Y_{1:T_d}$ . Given a state estimate and measurement for the current time step, and a model for the dynamics, the Kalman filter gives us a prediction for the measurement at the next time step, as well as a region of uncertainty around it, in the form of a covariance matrix. We call the ellipsoidal region defined by the predicted estimate and the covariance, the validation-gate. We iteratively proceed, one time step at a time, and compare the measurement at every time step with the validation-gate from the Kalman filter prediction from the previous time step. If the measurement falls outside the validation gates of all previously identified modes propagated from the previous time step, we initiate a new mode into the system. Doing this repeatedly, we estimate the number of modes ( $N$ ), the initial segmentation ( $S, L$ ), and the initial  $M$ . If the stochastic hybrid system is identifiable, then the Kalman filter validation gates will be distinguishable, and the algorithm will find the correct number of modes; if not, it will only give us a possible model that would explain the output data.

## 5 Examples

**Examples 1 and 2:** We first present two illustrative 1-D examples to explain the proposed algorithm. We use simulated data generated from the stochastic linear hybrid system  $x(k+1) = a_i x(k) + v(k); y(k) = c_i x(k) + w(k)$  [9] where  $i \in \{1, 2, 3\}$ ,  $a_1 = 0.5$ ,  $a_2 = 1.0$ ,  $a_3 = 1.1$ , and  $c_i = 1$ , for  $i = 1, 2, 3$ . The noise covariances are 1.0 and 0.1 respectively. The switching times in the

original system are at 39s., 56s., and 79s. The results of the parameter inference algorithm for hybrid systems for this 1-D example are given in Figure 4. The algorithm correctly identifies three discrete modes, and correctly detects the switching times. It also converges to a model consistent with the actual model. This is illustrated by the following extracts from the algorithm output:

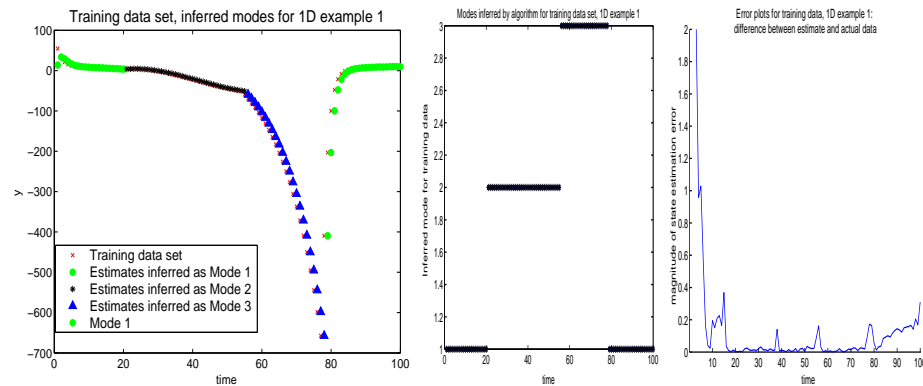
Parameter	$a_1$	$a_2$	$a_3$	$Q_1$	$Q_2$	$Q_3$	$R_1$	$R_2$	$R_3$	Transitions
True Value	0.5	1.0	1.1	1.0	1.0	1.0	0.1	0.1	0.1	[39 56 79]
Estimate	0.5014	1.0564	1.1023	1.547	1.2547	1.436	0.0939	0.0455	0.156	[39 56 79]

We realize that the estimates of the dynamical parameters ( $a_i$ ) are more critical to this method than the noise covariances, which are statistical, and therefore whose values will depend on the number of trials of data available. Clearly, there is a very good match in the actual and inferred values of the system dynamics and switching times. This is further illustrated by another example, this time with data generated by a system with parameters  $a_1 = 1.3$ ,  $a_2 = 1.5$ ,  $a_3 = 0.6666$ , and  $c_i = 1$ , for  $i = 1 \dots 3$ . In this case, with the same switching times as before, the system parameters are identified correctly.

System Parameter	$a_1$	$a_2$	$a_3$	Switching Times
True Value	1.3	1.5	0.6666	[39 56 79]
Estimated Value	1.3	1.5	0.6666	[39 56 79]

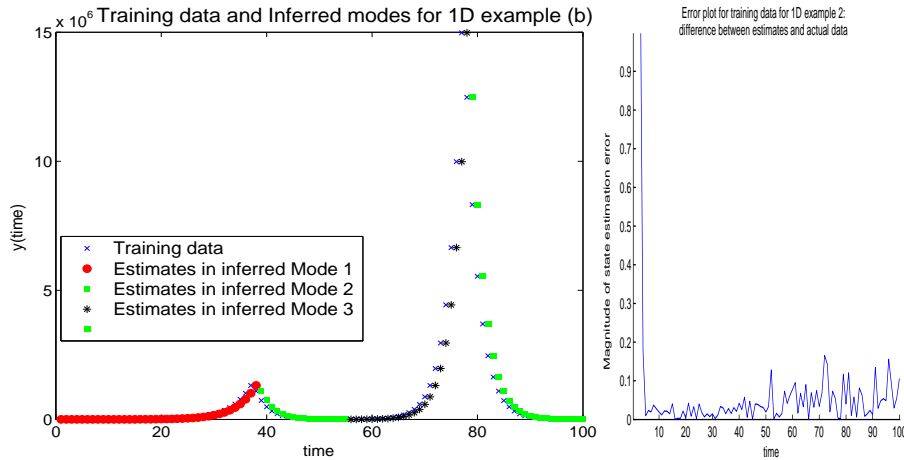
The training data and the results of the mode-detection are shown in Figure 5, along with the error plots. The error is the difference between the estimate of the continuous state from the inferred model, and the actual continuous state.

**Example 3:** We apply the algorithm to data obtained from the DragonFly



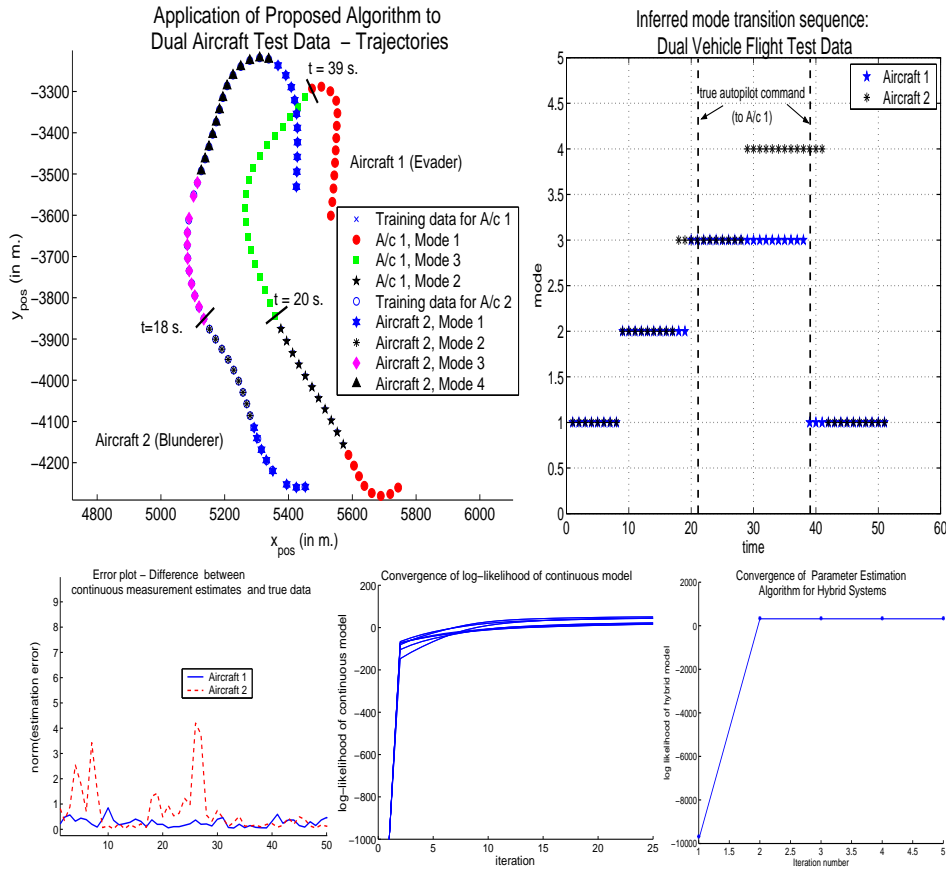
**Fig. 4.** Training data, with output of proposed algorithm for hybrid systems, for 1D Example (1). Left: Data sets, as a function of time. The different markers correspond to different identified discrete modes. Center: Mode transitions inferred from data set. Right: Estimation error plot.

Testbed at the Hybrid Systems Laboratory, at Stanford University. These are Unmanned Aerial Vehicles (UAVs), and the data we use are the position and



**Fig. 5.** Left: Training data, with output of Algorithm 2, for 1D Example (2). The different markers correspond to different identified discrete modes. Right: Estimation error plot.

velocity estimates. The data shown in this paper corresponds to a Dual-vehicle Flight Test. Aircraft 1 is referred to as the *evader*, and Aircraft 2 as the *blunderer*. The purpose of this experiment that produced this data was to demonstrate algorithms for provably safe closely-spaced parallel approaches [18]. However, we now use this data for a different purpose: given only the  $x$ - and  $y$ - position and velocity measurements, *i.e.*, no discrete or continuous state information, we build a model of the system which estimates the different modes of flight and the continuous states. While applying Algorithm 2 to Aircraft 1, we identify three discrete modes, with four distinct segments in the data. The switching times are estimated to be at  $t = 9, 20$  and  $39$  seconds. Comparing this to the times when the autopilot actions were initiated, we find that the system mode-transition commands were issued at  $t = 21$  and  $t = 39$  seconds. This further validates the performance of the algorithm we have proposed. We are also able to identify the other segments (and their modes) in the data sequence. In both aircraft trajectories, the initial segment (Mode 1) corresponds to the localizer-capture segment in which the aircraft first moves into autopilot, and attempts to capture the localizer trajectory (straight line). Mode 2 corresponds to the localizer-tracking segment. Now that the two aircraft are flying parallel to each other, at  $t = 18$  seconds, the blunderer (Aircraft 2) transitions to a mode that causes a potential conflict with the evader. Detecting this potential conflict, at  $t = 21$  seconds, the evader initiates a maneuver command to avoid the blunderer. At  $t = 39$  seconds, the manual pilot takes over from the autopilot and flies the aircraft back. The trajectories are plotted in Figure 5, along with the norm error of the smoothed estimates compared to the true data, the convergence plots, and the mode transition sequences.



**Fig. 6.** Parameter Inference Algorithm for Stochastic Linear Hybrid Systems, applied to Dual-Vehicle Flight Test data. Top: (right) Training data, inferred modes, estimates. (left) Mode transition sequence. Bottom (left) Error plots. (center) Convergence of continuous model. (right) Convergence of hybrid model.

## 6 Conclusions

We have proposed an algorithm that determines a maximum-likelihood hybrid system model, given only the continuous output of the system. With an intelligent choice of initial conditions, we also compute physically realistic models. We have applied these methods to the analysis of flight data from our UAV testbed (DragonFly) and obtained models that allow us to track the vehicles. These methods will have a wide-range of applications ranging from system identification for the many modes of an aircraft, to the problems of modeling biological systems as hybrid systems for the purpose of analysis. These techniques also provide a means of determining the parameters of a system that are required for the efficient design of hybrid estimators, as well as algorithms for multiple-target tracking and identity management [2].

## 7 Acknowledgments

The authors would like to thank Rodney Teo for providing the data from the CSPA algorithm verification, as well as for invaluable discussions on the same.

## References

1. Y. Bar-Shalom and T.F. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
2. I. Hwang, H. Balakrishnan, K. Roy, and C.J. Tomlin. Multiple-target Tracking and Identity Management in Clutter for Air Traffic Control. Submitted to the American Control Conference, 2004.
3. Y. Bar-Shalom and X.R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Boston, 1993.
4. Y. Li, T. Wang, and H.-Y. Shum. Motion texture: A two-level statistical model for character motion synthesis. In *ACM SIGGRAPH2002*, pages 465–472, 2002.
5. S. Soatto, G. Doretto, and Y.-N. Wu. Dynamic textures. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 439–446, July 2001.
6. R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications*. Springer-Verlag, New York, 2000.
7. L. Ljung. *System Identification: Theory for the User*. Prentice-Hall Inc., New Jersey, 1987.
8. B. De Moor and P. Van Overschee. Numerical algorithms for subspace state space system identification. In A. Isidori, editor, *Trends in Control: A European Perspective, The European Control Conference*, pages 385–422, 1995.
9. H.A.P. Blom and Y. Bar-Shalom. The Interacting Multiple Model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.
10. I. Hwang, H. Balakrishnan, and C.J. Tomlin. Observability criteria and estimator design for stochastic linear hybrid systems. In *Proceedings of the European Control Conference*, September 2003.
11. T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, New Jersey, 2000.
12. R. Vidal, A. Chiuso, and S. Soatto. Observability and identifiability of jump linear systems. In *Proceedings of the 41<sup>th</sup> IEEE Conference on Decision and Control*, December 2002.
13. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:185–197, 1977.
14. C.F.J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.
15. R.H. Shumway and D.S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.
16. A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.
17. E.J. Hannan and M. Deistler. *The Statistical Theory of Linear Systems*. Wiley Series in Probability and Mathematical Statistics. John Wiley and Sons, 1988.
18. R. Teo and C.J. Tomlin. Computing danger zones for provably safe parallel approaches. *Journal of Guidance, Control and Dynamics*, 26(3):434–442, May-June 2003.