

Learning Hybrid Models with Guarded Transitions

Pedro Santana*, Spencer Lane*, Eric Timmons*, Brian Williams*, Carlos Forster⁺

*Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory, MERS
32 Vassar St. Room 32-224, Cambridge, MA 02139, {psantana,slane,etimmoms,williams}@mit.edu

⁺Instituto Tecnológico de Aeronáutica

Pç. Mal. Eduardo Gomes, 50. Vl. das Acácias, 12228-900-São José dos Campos, SP, Brazil, forster@ita.br

Abstract

Innovative methods have been developed for diagnosis, activity monitoring, and state estimation that achieve high accuracy through the use of stochastic models involving hybrid discrete and continuous behaviors. A key bottleneck is the automated acquisition of these hybrid models, and recent methods have focused predominantly on Jump Markov processes and piecewise autoregressive models. In this paper, we present a novel algorithm capable of performing unsupervised learning of guarded Probabilistic Hybrid Automata (PHA) models, which extends prior work by allowing stochastic discrete mode transitions in a hybrid system to have a functional dependence on its continuous state. Our experiments indicate that guarded PHA models can yield significant performance improvements when used by hybrid state estimators, particularly when diagnosing the true discrete mode of the system, without any noticeable impact on their real-time performance.

1 Introduction

In the field of dynamical systems, we refer to a model as being hybrid if its behavior combines features of continuous and discrete state dynamics (van der Schaft and Schumacher 2000; Goebel, Sanfelice, and Teel 2009). For example, modern electronic circuits switch transistors on and off in order to control the flow of current, an inherently continuous quantity, across different components. Estimation and control involving hybrid models have proven valuable for a range of tasks involving complex behavior, in which accuracy is often paramount. Such real-world examples include target tracking (Blom and Bar-Shalom 1988), monitoring of aircraft at airports (Seah and Hwang 2009), fault diagnosis for space explorers (Hofbaur and Williams 2002; Blackmore et al. 2007), and control in degraded modes (Santana, Borges, and Ishihara 2010), among many others.

Probabilistic Hybrid Automata (PHA) (Hofbaur and Williams 2002; 2004) constitute a hybrid model class that can represent a large number of engineered systems. In a PHA, one refers to the discrete state as the *mode* of the automaton, which is usually hidden. Depending on its mode, the dynamics of the continuous portion of the automaton's

state can be governed by potentially distinct sets of difference or differential equations, therefore giving rise to a multiple-model system. Another feature of PHA is that discrete mode transitions are guarded, i.e., the probability of their occurrence depends on the continuous states and inputs to the system. For example, the switched RC circuit in Figure 1 can be modeled as a simple PHA with guarded transitions. The discrete mode represents whether the circuit is charging or discharging; the continuous state is the output voltage; and the input to system comes from the voltage source. Our results in Section 5 indicate that being able to handle stochastic guarded transitions is essential to improve the accuracy of hybrid inference algorithms, for these guards shed additional light on hidden mode transitions.

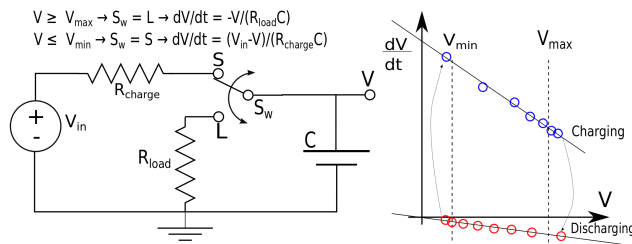


Figure 1: Switched RC circuit.

Effective methods have been developed to perform inference on Jump Markov Linear Systems (JMLS) (Blom and Bar-Shalom 1988; Mazor et al. 1998; Doucet, Gordon, and Krishnamurthy 2001; Hwang, Balakrishnan, and Tomlin 2006; Seah and Hwang 2009), a particular type of PHA with unguarded, purely stochastic mode transitions. There has been significant effort in the scientific community to automate the acquisition of JMLS (Logothetis and Krishnamurthy 1999; Henry 2002; Balakrishnan et al. 2004; Blackmore et al. 2007) and Piecewise Affine Autoregressive with Exogenous input (PWARX) models (Nakada, Takaba, and Katayama 2005; Juloski, Weiland, and Heemels 2005; Paoletti et al. 2007; Baptista, Ishihara, and Borges 2011), for the manual specification of such models can be quite a challenging and time-consuming task for even small-scale systems. Both types of model follow a similar high-level procedure when being learned from data. The first step is to determine which discrete mode was responsible for gen-

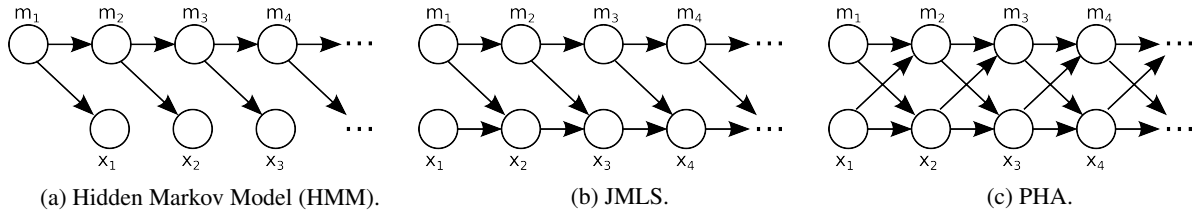


Figure 2: Different hybrid models considered. Discrete and continuous states are represented by variables m_i and x_i , $i \in \{1, \dots, n\}$, respectively. The deterministic inputs $u_{1:n}$ and distribution parameters have been omitted for the sake of clarity.

erating each one of the observations. Then, model parameters are found by means of a linear or nonlinear regression step within each class. A commonly used strategy in the JMLS community is to apply the widely-known Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) to iteratively refine model parameters so as to maximize the likelihood of the observed data. For PWARX models, on the other hand, the procedure usually involves an initial clustering step to determine observations pertaining to the same model, followed by an in-class parameter optimization procedure, and a full partitioning of the state and input spaces so as to isolate the different models, which is usually done by Support Vector Machines (SVM) (Cortes and Vapnik 1995). Generalizing to guarded transitions, the authors in (Seah and Hwang 2009) focused on the special case of guarded transition probabilities being Gaussian, which allows for closed-form expressions in filtering equations. However, learning PHA models with more expressive types of guard conditions remains a challenge.

We address this need by developing a variant of EM for PHA with guard conditions of arbitrary shape. Our key insight is that complex guard conditions can be efficiently represented as classifiers, and we show that these can be trained iteratively during the parameter optimization process in a completely unsupervised manner. Conditioning discrete mode transitions on the activation of guard conditions can significantly reduce the uncertainty about discrete mode transitions at any given point in time, therefore allowing hybrid diagnostic engines to better explore the space of possible diagnoses. In order to support this claim, we implemented the well-known Interacting Multiple Models (IMM) (Blom and Bar-Shalom 1988) filter for hybrid systems and found significant performance improvements when it operates with a PHA model, as opposed to a JMLS model for the same hybrid system.

This paper is organized as follows. Section 2 formally introduces the type of PHA addressed in this work, followed by our problem formulation in Section 3. Our algorithm is presented in Section 4, along with numerical results in Section 5. We offer our concluding remarks in Section 6.

2 Probabilistic Hybrid Automata

This paper focus on learning PHA models of the form

$$\begin{aligned} x_i &= A_{m_{i-1}}x_{i-1} + B_{m_{i-1}}u_{i-1} + w_{i-1}, \\ y_i &= C_{m_i}x_i + D_{m_i}u_i + v_i, \end{aligned} \quad (1)$$

$$\begin{aligned} p(m_{0:n}|x_{0:n}, u_{1:n}) &= \\ p(m_0) \prod_{i=1}^n p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1})). \end{aligned}$$

where $m_i \in \mathbb{M} = \{1, 2, \dots, M\}$ is the PHA’s discrete mode at the i -th time step; $x_i \in \mathbb{R}^{d_x}$, $u_i \in \mathbb{R}^{d_u}$, $y_i \in \mathbb{R}^{d_y}$ are the continuous state, input, and output vectors of the PHA, whose mode-dependent dynamics can switch among M different linear models; $w_{i-1} \sim \mathcal{N}(0, Q_{m_{i-1}})$, $v_i \sim \mathcal{N}(0, R_{m_i})$ are uncorrelated, white Gaussian noise; and $g_{m_i}(x_i, u_i) \in \mathbb{M}$ is the guard function for mode m_i . In order to be coherent with the PHA definition in (Hofbauer and Williams 2002), we say that the guard condition for the transition $m_i \rightarrow m$ is true iff $g_{m_i}(\cdot) = m$. Figure 2 shows a comparison in the form of graphical models between PHA and other common forms of hybrid models with probabilistic mode transitions.

It is important to gain more insight about the importance of folding guard conditions into hybrid models, specially when these models are being used for real-time tracking and diagnosis. Intuitively, a guard condition is a function of the PHA’s state (including the discrete mode m_i , continuous state x_i , and input u_i) that influences our confidence about what the next discrete mode of the PHA should be. More formally, guard conditions act as parameters in a family of mode transition priors $p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1}))$, each one reflecting our confidence about transitions $m_{i-1} \rightarrow m_i$ for different regions of the state and input. In JMLS models, on the other hand, unguarded stochastic mode transitions are represented by a single Transition Probability Matrix (TPM), which usually has to be hand-tuned after model learning in order to achieve acceptable performance in terms of peak estimation errors during mode transitions and steady-state errors, as described in (Li and Jilkov 2003). This problem is particularly pronounced for hybrid systems that dwell on discrete states for extended periods of time, for this increases the self-transition probabilities in the TPM during learning and adds a lot of “inertia” to the discrete mode transitions.

It is worthwhile to mention that no special form for the guard conditions is assumed in this paper, except that they must return an index of one of the models (m_i, \dots, m_k). In this work, we decided to use multi-class Support Vector Machines (SVM’s) as our guard functions for reasons that will later become apparent, but our learning algorithm is in no way dependent on this choice of a classifier.

3 Problem Formulation

Let θ be the PHA model parameters in (1). We frame our problem as an instance of data likelihood optimization, whose goal is to find the set of parameters θ^* such that

$$\theta^* = \arg \max_{\theta'} f(\theta') = \log p(y_{1:n}|u_{1:n}; \theta'), \quad (2)$$

where $y_{1:n}$ is a series of observations. Direct maximization of (2) is hard in general due to the presence of hidden states $x_{0:n}$ and modes $m_{0:n}$. However, if we somehow had access to the true values of $\bar{x}_{0:n}$ and $\bar{m}_{0:n}$, we could write

$$f(\theta') = \sum_{m_{0:n}; x_{0:n}} \int \delta(x_{0:n} - \bar{x}_{0:n}, m_{0:n} - \bar{m}_{0:n}) L(\theta') dx_{0:n} \quad (3)$$

where $\delta(\cdot)$ is the Dirac delta function and

$$\begin{aligned} L(\theta') &= \log p(y_{1:n}, x_{0:n}, m_{0:n}|u_{1:n}; \theta') \quad (4) \\ &= \log p(x_0, m_0) + \sum_{i=1}^n \log p(y_i|x_i, m_i, u_i; \theta') \\ &\quad + \log p(x_i|x_{i-1}, m_{i-1}, u_{i-1}; \theta') \\ &\quad + \log p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1}); \theta') \end{aligned}$$

Evaluating (4) would be possible if one had access to the Dirac delta function that effectively selects the correct hybrid states for (3). In order to circumvent this limitation, the approach in EM (Dempster, Laird, and Rubin 1977) is to iteratively maximize a lower bound for (3) given by

$$h(\theta') = \sum_{m_{0:n}; x_{0:n}} \int p(x_{0:n}, m_{0:n}|y_{1:n}, u_{1:n}) L(\theta') dx_{0:n} \quad (5)$$

$$= E_{p(x_{0:n}, m_{0:n}|y_{1:n}, u_{1:n})} [L(\theta')]. \quad (6)$$

In order to have an intuitive understanding of why (6) is a sensible choice, a quick comparison between (3) and (6) is sufficient. Instead of using a hard indicator, (6) replaces the Dirac delta by the posterior probability of the hidden state given the measurements and inputs, which is how much we believe that a particular sequence of hidden states is the true one. The EM algorithm then iteratively maximizes the lower bound (6) by following a two step procedure:

1. Use parameters θ^k for (1) from the k -th iteration to compute posterior probabilities $p^{k+1}(x_{0:n}, m_{0:n}|y_{1:n}, u_{1:n})$. This is the *E-step*;
2. Use $p^{k+1}(x_{0:n}, m_{0:n}|y_{1:n}, u_{1:n})$ to find new parameters θ^{k+1} that maximize (6). This is the *M-step*. Repeat until successive evaluations of (6) converge.

By following this simple procedure, EM is guaranteed to converge to a local optimum of (3), the true likelihood function. A short and useful review of EM is given in (Blackmore et al. 2007), along with more thorough and pedagogical descriptions available at other sources, such as (Bishop and Nasrabadi 2006).

Computing $p(x_{0:n}, m_{0:n}|y_{1:n}, u_{1:n})$, however, is still an intractable problem due to the large number of possible discrete mode trajectories, which is exponential in n . The authors in (Blackmore et al. 2007) use the PHA

mode trajectory enumerator developed in (Hofbauer and Williams 2002) to compute a set of N likely mode trajectories and, for each one of them, use a forward-backward Kalman filter recursion to compute smoothed estimates of $p(x_{0:n}|m_{0:n}, y_{1:n}, u_{1:n})$. For linear models corrupted by Gaussian noise, these smoothed estimates $p(x_{0:n}|m_{0:n}, y_{1:n}, u_{1:n})$ will also be Gaussian distributions. Then, integration over $x_{0:n}$ in (6) causes the hidden state x_i in (4) to be replaced by $E_{p(x_{0:n}|m_{0:n}, y_{1:n}, u_{1:n})}[x_i]$.

Following the practice in PWARX model learning, we develop our algorithm assuming that the continuous state vector can be directly measured, i.e., only the discrete mode sequence $m_{0:n}$ is hidden. In this case, our objective function becomes

$$Q(\theta') = \sum_{m_{0:n}} p(m_{0:n}|x_{0:n}, u_{1:n}) \tilde{L}(\theta') \quad (7)$$

$$= E_{p(m_{0:n}|x_{0:n}, u_{1:n})} [\tilde{L}(\theta')]. \quad (8)$$

where $\tilde{L}(\theta') = \log p(x_{0:n}, m_{0:n}|u_{1:n}; \theta')$. In Section 4, we develop the equations for the *E* and *M* steps that maximize (8) iteratively by improving the parameters of the model (1), which includes the computation of guard conditions. Should the continuous state not be directly observable, we could then resort to mode trajectory enumeration (Hofbauer and Williams 2002) or sampling-based methods (Doucet, Gordon, and Krishnamurthy 2001) in order to focus the optimization on a small subset of likely trajectories.

4 Learning PHA Models with EM

In this section, we derive a novel extension of EM to estimate maximum likelihood parameters of guarded PHA models from data. For the sake of clarity and space, some intermediate manipulations have been omitted.

4.1 E Step

Let θ be the model parameters that we are trying to estimate. The log-likelihood of the data in (8) is given by

$$\begin{aligned} \tilde{L}(\theta) &= \log p(x_{0:n}, m_{0:n}|u_{1:n}; \theta) \\ &= \log p(x_0, m_0) + \sum_{i=1}^n \log p(x_i|x_{i-1}, m_{i-1}, u_{i-1}; \theta) \\ &\quad + \sum_{i=1}^n \log p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1}); \theta) \quad (9) \end{aligned}$$

Given the posterior joint probability of the discrete modes $p(m_{0:n}|x_{0:n}, u_{1:n}; \theta)$, the lower bound (8) is given by

$$\begin{aligned} Q(\theta) &= E_{p(m_{0:n}|x_{0:n}, u_{1:n})} [\tilde{L}(\theta)] \\ &= \sum_{i=1}^n \left(\sum_{m_{i-1}} \gamma(m_{i-1}) \log p(x_i|x_{i-1}, m_{i-1}, u_{i-1}) \right. \\ &\quad \left. + \sum_{m_{i-1}, m_i} \xi(m_{i-1}, m_i) \log p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1})) \right) \\ &\quad + \sum_{m_0} \gamma(m_0) \log p(x_0, m_0), \quad (10) \end{aligned}$$

where

$$\begin{aligned}\gamma(m_{i-1}) &= p(m_{i-1}|x_{0:n}, u_{1:n}; \theta), \\ \xi(m_{i-1}, m_i) &= p(m_{i-1}, m_i|x_{0:n}, u_{1:n}; \theta),\end{aligned}\quad (11)$$

are posterior marginal mode probabilities. Below we derive a Forward-Backward algorithm capable of computing (11) for the PHA model in Figure 2c. Let

$$\begin{aligned}\alpha_g(m_{i-1}) &= p(x_1, \dots, x_{i-1}, m_{i-1}|u_{1:n}; \theta), \\ \beta_g(m_{i-1}) &= p(x_n, \dots, x_i|m_{i-1}, x_{i-1}, u_{1:n}; \theta).\end{aligned}\quad (12)$$

For simplicity, we drop the explicit dependency of (12) on θ . We can rewrite (11) in terms of (12) as

$$\gamma(m_{i-1}) = \frac{\alpha_g(m_{i-1})\beta_g(m_{i-1})}{\sum_{m'_{i-1}} \gamma(m'_{i-1})}, \quad (13)$$

$$\xi(m_{i-1}, m_i) = \frac{\alpha_g(m_{i-1})d(x_i, m_{i-1})t(m_{i-1}, m_i)\beta_g(m_i)}{\sum_{m'_{i-1}, m'_i} \xi(m'_{i-1}, m'_i)},$$

where

$$\begin{aligned}d(x_i, m_{i-1}) &= p(x_i|x_{i-1}, m_{i-1}, u_{i-1}), \\ t(m_{i-1}, m_i) &= p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1})).\end{aligned}\quad (14)$$

The probabilities (14) are computed from the dynamics (1) and the guarded transitions. The terms in (12) can be computed recursively as follows:

$$\alpha_g(m_{i-1}) = \sum_{m_{i-2}} \alpha_g(m_{i-2})d(x_{i-1}, m_{i-2})t(m_{i-2}, m_{i-1}), \quad (15)$$

$$\beta_g(m_{i-1}) = \sum_{m_i} \beta_g(m_i)d(x_i, m_{i-1})t(m_{i-1}, m_i), \quad (16)$$

$$\alpha_g(m_0) = p(x_0, m_0), \beta_g(m_n) = 1. \text{ (given).}$$

4.2 M Step

Our goal now is to maximize (10) given the posterior probabilities (13) computed in the E-step. For each parameter in the model (1), we present its corresponding M step.

Linear models From (1), we have

$$p(x_i|x_{i-1}, m_{i-1}, u_{i-1}; \theta) = N(x_i; A(m_{i-1})x_{i-1} + B(m_{i-1})u_{i-1}, Q) \quad (17)$$

Assuming that $Q = \sigma^2 I$ is fixed, we have

$$\log p(x_i|x_{i-1}, m_{i-1}, u_{i-1}; \theta) = -\frac{1}{2\sigma^2} \|x_i - \mu_{i-1}\|^2 + c, \quad (18)$$

where $\mu_{i-1} = A(m_{i-1})x_{i-1} + B(m_{i-1})u_{i-1}$ and c is a constant. Substituting (18) into the first summation in (10), the parameters that should be optimized are the matrices A_k and B_k for each one of the possible M assignments to m_{i-1} . Choosing A_k and B_k so as to maximize (4) requires us to minimize a quadratic matrix term of the form

$$(Y - \Phi b(k))^T W(k) (Y - \Phi b(k)), \quad (19)$$

which is an instance of Weighted Least Squares (WLS) with the weights being the posterior probabilities (13). In terms of the data, the matrices in (19) are given by

$$\begin{aligned}\Phi &= \begin{bmatrix} x_0^T & u_0^T \\ \vdots & \vdots \\ x_{n-1}^T & u_{n-1}^T \end{bmatrix}, Y = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}, b(k) = \begin{bmatrix} A_k^T \\ B_k^T \end{bmatrix} \\ W(k) &= \text{diag}(\gamma(m_0)=k, \dots, \gamma(m_{n-1})=k),\end{aligned}\quad (20)$$

whose closed form solution is

$$\hat{b}(k) = \begin{bmatrix} \hat{A}_k^T \\ \hat{B}_k^T \end{bmatrix} = (\Phi^T W(k) \Phi)^{-1} \Phi^T W(k) Y. \quad (21)$$

Initial Probability For the initial mode probability, we use

$$\hat{p}(m_0 = k) = \frac{\gamma(m_0 = k)}{\sum_{k'} \gamma(m_0 = k')} \quad (22)$$

Guarded Transition Probabilities The guarded transition probabilities are given by

$$\begin{aligned}\hat{p}(m_i=k|m_{i-1}=k', g_{m_{i-1}=k}(x_{i-1}, u_{i-1})=m) &= \\ & \frac{\sum_{\{x_{j-1}, u_{j-1}\}: g_{m_{j-1}=k}(x_{j-1}, u_{j-1})=m} \xi(m_{j-1}=k', m_j=k)}{\sum_{m_j=k''} \hat{p}(m_j=k''|m_{j-1}=k', g_{m_{j-1}=k}(x_{j-1}, u_{j-1})=m)}\end{aligned}\quad (23)$$

Guard Functions Different from the other parameters, whose update equations can be readily obtained via standard maximization of (10) using Lagrange multipliers, finding the optimal parameters for the guard conditions in (1) is more challenging. For the guard function with origin at some particular m_{i-1} , the only affected portion of (10) is

$$\sum_{i=1}^n \sum_{m_i} \xi(m_{i-1}, m_i) \log p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1})). \quad (24)$$

The values of $\xi(m_{i-1}, m_i)$ have been previously computed in the E-step and we assume that we are using the values for $p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1}))$ from the previous iteration of EM. In the ideal case where we can choose any value for $g_{m_{i-1}}(x_{i-1}, u_{i-1})$ for each sample x_{i-1} and u_{i-1} , the optimization of (24) is easy. In this case, for sample $i-1$, we choose $g_{m_{i-1}}(x_{i-1}, u_{i-1}) = m^*$ such that

$$m^* = \arg \max_m \sum_{m_i} \xi(m_{i-1}, m_i) \log p(m_i|m_{i-1}, g_{m_{i-1}}(x_{i-1}, u_{i-1})). \quad (25)$$

Equation (25) is creating a correspondence between the input (x_{i-1}, u_{i-1}) and the value $m \in \mathbb{M} = 1, 2, \dots, M$ that maximizes (10). Therefore, an obvious approach would be to create a table, one row per distinct pair (x_{i-1}, u_{i-1}) , and its corresponding label $m \in \mathbb{M}$ according to (25). However, that would not generalize well to unseen examples of x and u , since it is very unlikely that we would observe during operation the same pairs of inputs that were used to learn the model. Hence, we need a better approach.

In this work, we chose to represent the correspondence between the input space (x, u) and discrete values in the set \mathbb{M} using multi-class SVM’s. A complete discussion of SVM’s is out of the scope of this paper, but extensive literature on the subject exists and we refer the reader to (Burges 1998; Bishop and Nasrabadi 2006) as examples of good references. Among its many good features, SVM’s usually generalize well to unseen examples; they can potentially represent transition boundaries of arbitrary shape; and the optimization problem used to determine its parameters can be solved quickly for moderately-sized datasets. However, we stress that our PHA learning algorithm does not rely on SVM’s in order to represent guard conditions.

The M-step for the guard function then consists of computing the mapping $(x_{i-1}, u_{i-1}) \rightarrow m \in \mathbb{M}$ by means of (25) and using this “labeled” data as the input to a multi-class SVM aiming at increasing the value of (24).

5 Numerical Results

Our algorithm was implemented in Python and multi-class SVM’s were trained using wrappers for *LIBSVM* (Chang and Lin 2011) in *Scikit-learn* (Pedregosa et al. 2011). In order to evaluate the potential gains of using guarded PHA models in hybrid estimation, we implemented the widely-known IMM filter (Blom and Bar-Shalom 1988) and evaluated its performance with PHA and JMLS models for the same underlying hybrid systems in two different target-tracking domains. State estimation was always performed with the same parameters, the only difference being the type of hybrid model being used. It is worthwhile to mention that IMM was originally designed as a state estimator for JMLS.

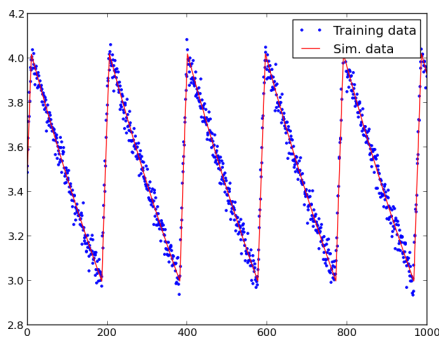


Figure 3: PHA model fit to the RC circuit data.

We started with the pedagogical switched RC circuit example shown in Figure 1, given that its simplicity allowed us to verify that all portions of the PHA model were being correctly learned from data. We initialized EM with 40% misclassified modes and learned a PHA model using 1000 data points. The minimum and maximum output voltages were set to $V_{\min}=3.0V$ and $V_{\max}=4.0V$, respectively. The input voltage was $V_{in}=10V$. Our SVMs were trained using linear feature vectors and were allowed slack with very high penalty for misclassified points. Figure 3 shows the PHA model fit to the RC circuit data, while Figure 4 shows the following transitions rules learned as guard conditions by

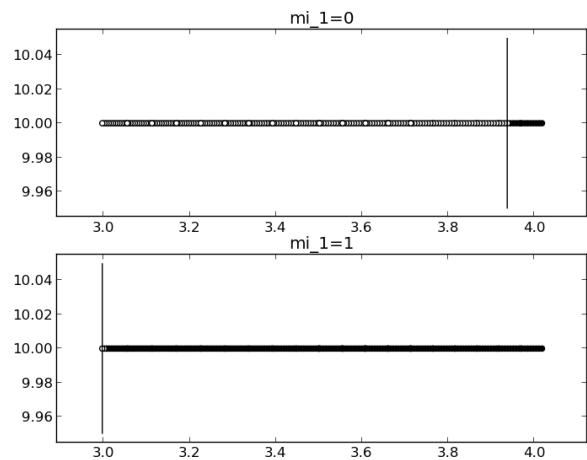


Figure 4: Learned guard conditions for the switched RC Circuit model. In the top plot, all voltage samples below 3.94V are classified as 0 (charging) and shown as white dots. Black dots are in the active transition to mode 1 (discharging). The bottom plot shows the 3.0V boundary for the discharging \rightarrow charging transition.

our algorithm:

$$g_0(x_{i-1}, u_{i-1}) = \begin{cases} 1 \text{ (discharging)} & \text{if } x_{i-1} > 3.94 \\ 0 & \text{otherwise.} \end{cases}$$

$$g_1(x_{i-1}, u_{i-1}) = \begin{cases} 0 \text{ (charging)} & \text{if } x_{i-1} < 3.00 \\ 1 & \text{otherwise.} \end{cases} \quad (26)$$

Once we determined that our algorithm was capable of learning correct parameters for PHA models, we evaluated the impact of a guarded PHA model on two instances of target tracking, the domain for which IMM was originally developed and a classical benchmark for hybrid filtering. Our models were based on (Seah and Hwang 2009) and describe an airplane that can travel straight or turn in either direction with constant speed. In the first target tracking domain, the airplane follows a “lawnmower” pattern, which involves traveling back and forth over a bounding box in a systematic fashion (Figure 5a). This type of flight pattern is widely used in a number of applications ranging from search and rescue (Goodrich et al. 2008), agriculture (Bryson et al. 2010), and underwater exploration (Ferri, Jakuba, and Yoerger 2008). In the second target-tracking domain, the airplane performs a random Markovian transition to one of the modes (straight, turn left, or turn right) with equal probability, dwells in it for 50 time steps, and repeats the process (Figure 5d). For each one of these domains, we ran IMM 32 times, discarding the best and worst results, and scaled the RMSE so that it is always 1 for PHA. The average results are summarized in Tables 1 and 2. Selected IMM performance plots using PHA and JMLS models are shown for both domains in Figure 5.

The plots in Figures 5a-5c and the results in Table 1 show a significant performance improvement for IMM when performing target tracking with a PHA model. The excessive “inertia” associated with self-transition probabilities in

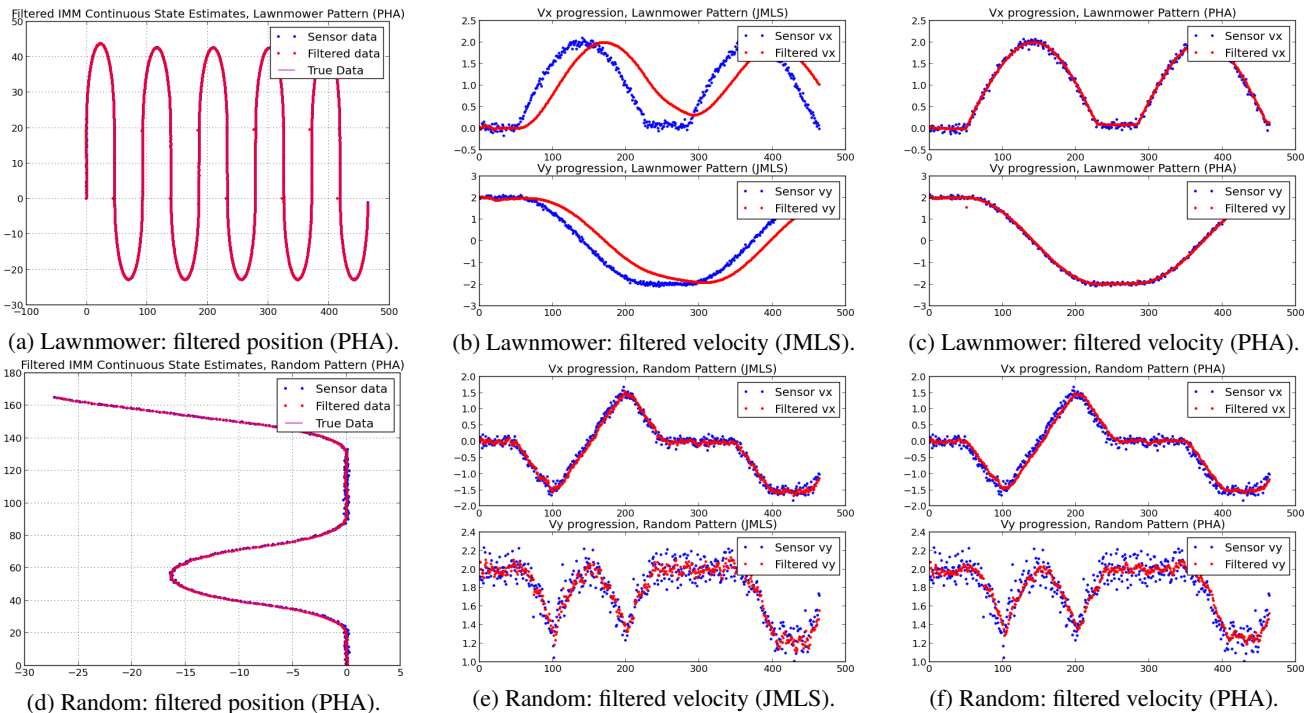


Figure 5: Typical IMM filtering performance on target-tracking domains for different types of hybrid models.

	Pos. RMSE	Vel. RMSE	% Misclas. modes
JMLS	2.284	14.44	53.93
PHA	1	1	3.33

Table 1: Average IMM performance in lawnmower domain.

	Pos. RMSE	Vel. RMSE	% Misclas. modes
JMLS	0.985	1.002	58.91
PHA	1	1	63.2

Table 2: Average IMM performance in random trajectory domain.

the JMLS model prevented IMM to correctly detect mode changes in the lawnmower pattern, causing it to poorly track the evolution of the airplane’s velocity. Tracking with the PHA model, on the other hand, did not have this issue, since the PHA model automatically adjusts discrete mode transition probabilities according to the airplane’s two-dimensional positional and velocity vector. Another evidence of the inability of the JMLS model to adapt to this fast-changing domain can be seen in the significant fraction of misclassified discrete modes across the trajectory.

Our second testing domain, target tracking with a Markovian trajectory, was used in order to provide a fair comparison of the two models in a situation where JMLS is expected to perform well. Also, PHA is a strict superset of JMLS, so we would expect the performance of IMM with a PHA model to be basically the same as with a JMLS one. The average results in Table 2 confirm our expectations and show

that, when there is no structure relating the discrete mode transitions to the system’s discrete state, PHA and JMLS perform equally well. In fact, one can also notice that they perform equally poorly in terms of diagnosing the true discrete state of the system due to the probability “inertia” issue previously mentioned for the lawnmower domain. This is confirmed by the fact that, if we replace all discrete mode transition probabilities by uniform distributions (probability 1/3 of transitioning from any mode to any other mode), IMM is capable of reconstructing the correct mode sequence for both models.

6 Conclusions

This work presented a novel method of learning PHA models that takes determination of guard conditions explicitly into account. We frame the parameter estimation problem as an instance of maximum likelihood optimization and develop the equations for a novel variant of EM that is capable of learning PHA models from experimental data with hidden mode transitions. Our numerical results show that not only our algorithms is capable of correctly extracting PHA model parameters from unlabeled data, but also that PHA models can boost the performance of hybrid state estimators by incorporating continuous state information into stochastic discrete mode transition models.

Acknowledgments

This research was partially funded by AFOSR grant 6926079 and the Mitsubishi Electric Corporation.

References

- Balakrishnan, H.; Hwang, I.; Jang, J. S.; and Tomlin, C. J. 2004. Inference methods for autonomous stochastic linear hybrid systems. In *Hybrid Systems: Computation and Control*. Springer. 64–79.
- Baptista, R.; Ishihara, J.; and Borges, G. 2011. Split and merge algorithm for identification of piecewise affine systems. In *American Control Conference (ACC), 2011*, 2018–2023.
- Bishop, C. M., and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 1. Springer New York.
- Blackmore, L.; Gil, S.; Chung, S.; and Williams, B. 2007. Model learning for switching linear systems with autonomous mode transitions. In *Decision and Control, 2007 46th IEEE Conference on*, 4648–4655. IEEE.
- Blom, H. A., and Bar-Shalom, Y. 1988. The interacting multiple model algorithm for systems with markovian switching coefficients. *Automatic Control, IEEE Transactions on* 33(8):780–783.
- Bryson, M.; Reid, A.; Ramos, F.; and Sukkarieh, S. 2010. Airborne vision-based mapping and classification of large farmland environments. *Journal of Field Robotics* 27(5):632–655.
- Burges, C. J. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 2(2):121–167.
- Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27.
- Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.
- Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 1–38.
- Doucet, A.; Gordon, N. J.; and Krishnamurthy, V. 2001. Particle filters for state estimation of jump markov linear systems. *Signal Processing, IEEE Transactions on* 49(3):613–624.
- Ferri, G.; Jakuba, M. V.; and Yoerger, D. R. 2008. A novel method for hydrothermal vents prospecting using an autonomous underwater robot. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 1055–1060. IEEE.
- Goebel, R.; Sanfelice, R. G.; and Teel, A. 2009. Hybrid dynamical systems. *Control Systems, IEEE* 29(2):28–93.
- Goodrich, M. A.; Morse, B. S.; Gerhardt, D.; Cooper, J. L.; Quigley, M.; Adams, J. A.; and Humphrey, C. 2008. Supporting wilderness search and rescue using a camera-equipped mini uav. *Journal of Field Robotics* 25(1-2):89–110.
- Henry, M. M. 2002. Model-based estimation of probabilistic hybrid automata. Master’s thesis, Massachusetts Institute of Technology.
- Hofbauer, M. W., and Williams, B. C. 2002. Mode estimation of probabilistic hybrid systems. In *Hybrid Systems: Computation and Control*. Springer. 253–266.
- Hofbauer, M. W., and Williams, B. C. 2004. Hybrid estimation of complex systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 34(5):2178–2191.
- Hwang, I.; Balakrishnan, H.; and Tomlin, C. 2006. State estimation for hybrid systems: applications to aircraft tracking. *IEE Proceedings-Control Theory and Applications* 153(5):556–566.
- Juloski, A. L.; Weiland, S.; and Heemels, W. 2005. A bayesian approach to identification of hybrid systems. *Automatic Control, IEEE Transactions on* 50(10):1520–1533.
- Li, X. R., and Jilkov, V. P. 2003. A survey of maneuvering target tracking part v: Multiple-model methods. In *Conference on Signal and Data Processing of Small Targets*, volume 4473, 559–581.
- Logothetis, A., and Krishnamurthy, V. 1999. Expectation maximization algorithms for map estimation of jump markov linear systems. *Signal Processing, IEEE Transactions on* 47(8):2139–2156.
- Mazor, E.; Averbuch, A.; Bar-Shalom, Y.; and Dayan, J. 1998. Interacting multiple model methods in target tracking: a survey. *Aerospace and Electronic Systems, IEEE Transactions on* 34(1):103–123.
- Nakada, H.; Takaba, K.; and Katayama, T. 2005. Identification of piecewise affine systems based on statistical clustering technique. *Automatica* 41(5):905–913.
- Paoletti, S.; Juloski, A. L.; Ferrari-Trecate, G.; and Vidal, R. 2007. Identification of hybrid systems a tutorial. *European Journal of Control* 13(2):242–260.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.
- Santana, P.; Borges, G.; and Ishihara, J. Y. 2010. Hybrid data fusion for 3d localization under heavy disturbances. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2425–2430. IEEE.
- Seah, C. E., and Hwang, I. 2009. State estimation for stochastic linear hybrid systems with continuous-state-dependent transitions: an IMM approach. *Aerospace and Electronic Systems, IEEE Transactions on* 45(1):376–392.
- van der Schaft, A. J., and Schumacher, J. M. 2000. *An introduction to hybrid dynamical systems*, volume 251. Springer London.