Beyond local optimality: An improved approach to hybrid model learning

Citation: Gil, S., and B. Williams. "Beyond local optimality: An improved approach to hybrid model learning." Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on. 2009. 3938-3945. © 2010 Institute of Electrical and Electronics Engineers.

As Published: http://dx.doi.org/10.1109/CDC.2009.5400529

Publisher: Institute of Electrical and Electronics Engineers

Persistent URL: http://hdl.handle.net/1721.1/59343

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Joint 48th IEEE Conference on Decision and Control and
28th Chinese Control Conference
Shanghai, P.R. China, December 16-18, 2009

Beyond Local Optimality: An Improved Approach to Hybrid Model Learning

Stephanie Gil and Brian Williams

Abstract— **Local convergence is a limitation of many optimization approaches for multimodal functions. For hybrid model learning, this can mean a compromise in accuracy. We develop an approach for learning the model parameters of hybrid discrete-continuous systems that avoids getting stuck in locally optimal solutions. We present an algorithm that implements this approach that 1) iteratively learns the locations and shapes of explored local maxima of the likelihood function, and 2) focuses the search away from these areas of the solution space, toward undiscovered maxima that are *a priori* likely to be optimal solutions. We evaluate the algorithm on Autonomous Underwater Vehicle (AUV) data. Our aggregate results show reduction in distance to the global maximum by 16% in 10 iterations, averaged over 100 trials, and iterative increase in log-liklihood value of learned model parameters, demonstrating the ability of the algorithm to guide the search toward increasingly better optima of the likelihood function, avoiding local convergence.**

I. Introduction

Stochastic continuous and hybrid discrete-continuous systems have earned wide usage in a range of fields including finance, machine vision, and autonomous underwater vehicles [1], [2], [3]. These models accurately represent a general scope of problems while explicitly accounting for the uncertainty inherent to the environment they work in. Recent research using hybrid models has focused on estimation and control [4], [5], [6], [7]. However, the effectiveness of these methods hinges on the accuracy of the underlying model. Specifying these models manually is often challenging and inadequate, given that many systems change or degrade over time, obviating the validity of the original model. We must therefore determine hybrid system models from observed data [8], [9], [4], [10]. Since the discrete and continuous dynamics of such models are coupled, partially observable and stochastic, this is a very challenging problem.

One widely-used technique that has been adapted to learn the model parameters for these systems is Expectation Maximization [11], [12], [13], [10], [14]. Expectation Maximization (EM) is well-suited to this problem because it provides Maximum Likelihood parameter estimates from data where some variables may be unobservable, or hidden. EM-based model learning techniques provide methods that converge to a locally optimal set of model parameters. Local convergence, however, is a limitation of many optimization approaches for multi-modal functions, including EM. This is the case for hybrid parameter estimation where the likelihood function is multi-modal. An EM-based parameter estimation technique can get stuck at the top of a local maximum hill of the likelihood function, far from the best parameter estimate.

A standard technique for jumping out of these local maxima is a restart method that samples from a distribution over the model parameters, often a uniform distribution, to find a new initialization point for the local optimization algorithm. Restart algorithms can be effective in general, if there is a high probability that the restart method will discover a new maximum of the likelihood function that has a greater value than previously discovered maxima. Global optimization techniques such as Simulated Annealing and Genetic Algorithms also employ the idea of sampling other areas of the search space [15], [16]. Unfortunately, in many cases, as for the case of hybrid model learning, the posterior distribution over the hidden state cannot be obtained in closed form and must be approximated [10]. This makes it difficult to use global or stochastic optimization techniques [17], [18], [19] that often rely on the ability to sample from the likelihood or its gradient. Previous work employs an approximation of the posterior distribution over the hidden state and uses local optimization techniques to learn the model parameters of hybrid systems [10], [14]. We extend [14] by avoiding convergence to a single local optima via a guided restart method over the parameter space; the algorithm aims to find globally or near-globally optimal solutions to the hybrid model learning problem.

We propose an approach that identifies explored areas of the solution space and guides the search outside of these areas. The high-dimensional nature of the search space in model learning problems make memory-based techniques that record and avoid explored regions, like TABU search [20], less well-suited. Instead, we learn a mixture of Gaussians representing the searched area of model parameters. We make this representational choice because we are able to use results of Gaussian distributions in the derivation of our approach, and because we wish to capture specific properties of the EM algorithm. EM guarantees that within the base of a local maxima, the optimization always moves in the direction of increasing likelihood [21]. While in the general case this may lead to undesirable solutions resulting from poor initialization, this is precisely the property of EM that we exploit in designing the kMeans-EM algorithm. For continuous functions whose maxima can be approximated by Gaussian shapes, and where an initialization point within the base of a maximum θ^* converges toward θ^*, we wish to use the smallest number of strategically chosen initializations to find the optimal solutions of the likelihood function.

The kMeans-EM algorithm learns the shapes and locations

This work is supported by the Bell Labs Graduate Fellowship
Stephanie Gil is an SM student, MIT. sgil@mit.edu
Brian Williams is a Professor, MIT. williams@mit.edu

of discovered local maxima of the likelihood function, that is combined with a prior, to construct a density over the solution space. This density is maximized to find initializations that are most likely to be optimal solutions *a priori* and lie outside of explored areas of the solution space. We target the learning of continuous model parameters, θ_c, that are optimal or near-optimal solutions to the maximization of the likelihood function.

In Section II we define the models that we are interested in learning, as well as the problem statement for KMeans-EM model learning, in Section III we provide an overview of the kMeans-EM algorithm, in Section IV we briefly review EM for Hybrid Model Learning, and describe the key components of the kMeans-EM algorithm in Sections V and VI. Finally in section VII we provide simulation results.

II. Problem Statement

Prior work defined the Linear Probabilistic Hybrid Automaton (LPHA) and presented an EM-based method for learning the continuous and discrete model parameters for these systems [22], [14]. The continuous dynamics for a LPHA are given by:

$$\begin{aligned} \mathbf{x}_{t+1} &= A(\mathbf{m}_t)\mathbf{x}_t + B(\mathbf{m}_t)\mathbf{u}_t + \omega_t \\ \mathbf{y}_{c,t+1} &= C(\mathbf{m}_t)\mathbf{x}_{t+1} + D(\mathbf{m}_t)\mathbf{u}_t + \nu_t, \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is the continuous state and $\mathbf{m} \in \mathcal{X}_d$ is the discrete state, where $\mathcal{X}_d$ is the set $\{1, .., M\}$ and M is the number of allowed discrete states. The evolution of the discrete state is often determined by a set of switching probabilities as in [14], however, we do not include the learning of discrete model parameters in our current problem statement. We use the subscript notation $\mathbf{x}_t$ to denote the value of variable $\mathbf{x}$ at time t, and use $\mathbf{x}_{t_1}^{t_2}$ to denote the sequence $\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_2}$. The variables ω and ν are process and observation noise respectively, which we restrict to be zero-mean, Gaussian white noise with covariance $Q(\mathbf{m}_t)$ and $R(\mathbf{m}_t)$, respectively. The initial distribution $p(\mathbf{x}_0, \mathbf{m}_0)$ is a sum-of-Gaussians, $p(\mathbf{x}_0|\mathbf{m}_0)$ is a Gaussian with mean $\mu(\mathbf{m}_0)$ and covariance $V(\mathbf{m}_0)$.

The goal of the [14] paper was to present an algorithm to learn a set of locally optimal model parameters for the dynamics equation in (1). However, the accuracy of the learned parameters are subject to the quality of the initialization parameters of the algorithm. In fact, sensitivity to initialization is common for many local optimization techniques [23]. The goal of this paper extends previous work to learning the optimal or near-optimal continuous model parameters in (1) and thus mitigating the effect of poor algorithm initialization.

The *continuous model parameters* $\theta_c(\mathbf{m})$ are defined for each mode $\mathbf{m} \in \mathcal{X}_d$ of a LPHA as the set $[A(\mathbf{m}), B(\mathbf{m}), C(\mathbf{m}), D(\mathbf{m}), Q(\mathbf{m}), R(\mathbf{m}), V(\mathbf{m}), \mu(\mathbf{m})]$.

Our model learning problem statement is the following:

> *Given a finite sequence of observations $\mathbf{y}_1^{T+1}$, and a finite sequence of control inputs $\mathbf{u}_0^T$, iteratively learn the shapes and locations of discovered local maxima of the log-likelihood objective function $g(\theta_c) = \log p(\mathbf{y}_1^{T+1}|\theta_c)$. Use this map to guide the search toward new maxima that are a priori more likely to be optimal solutions to*

$$\theta_c^* = \arg\max_{\theta} \{p(\mathbf{y}_1^{T+1}|\theta)\} \tag{2}$$

III. Overview of Approach

We propose an approach that allows for a thorough search over the space of model parameters and a systematic method of forcing Expectation Maximization for hybrid model learning to jump away from local maxima of the log-likelihood function. Intuitively, we wish to find initializations that have high probability of a) converging to optimal solutions while b) avoiding convergence to discovered maxima of the log-likelihood function, $g(\theta)$. We can learn the probability that an arbitrarily chosen point in the parameter space will converge to any explored maxima of the likelihood function. Combining this probability with a prior favoring areas of the search space that are likely to be optimal solutions of $g(\theta)$, we define a function $s(\theta)$ whose maximization with respect to θ results in a set of initialization parameters, $\widetilde{\theta}_0$, that achieves a and b. We present an instance of this approach via an algorithm that combines EM and K-Means Clustering, that we call the kMeans-EM algorithm.

1) **Initialization Phase.** Set k=1. Run Hybrid EM on many different initializations to label, or map, each initialization with a corresponding converged set of parameters θ_f
   $\theta_{0_1} \rightarrow \theta_{f_1}$
   $\theta_{0_2} \rightarrow \theta_{f_2}$
   $\vdots$
   $\theta_{0_n} \rightarrow \theta_{f_n}$
2) **Clustering Phase.** Learn a mixture of Gaussians model over all labeled sets of parameters $\{\theta_{f_i}\}_{i=1}^n$ via a clustering algorithm where k is the number of existing clusters and each cluster has a centroid c_j and a covariance Λ_j.
3) **Optimization Phase.** Generate a new guess of parameters $\widetilde{\theta}_0$ that is likely to converge to a new local maxima of $g(\theta)$ by maximizing the objective function $s(\theta)$
4) If $\widetilde{\theta}_0$ has a low probability of belonging to any of the existing k clusters, go to step 5, else return to step 3.
5) **Labeling Phase.** Run Hybrid EM using $\widetilde{\theta}_0$ as initial guess of parameters to find $\widetilde{\theta}_f$. Set $n = n+1$. If $\widetilde{\theta}_f$ has a low probability of belonging to an existing cluster, set $k = k + 1$. Go to step 2.

Fig. 1. The kMeans-EM Algorithm

A mixture of Gaussians is learned over the parameter space, where peaks of each Gaussian correspond to the approximate locations of discovered local maxima of $g(\theta)$, and the respective covariances approximate the width and shape of the bases of each maxima. We choose a mixture of Gaussians because we are interested in smooth, bounded

variance likelihood functions whose maxima can be approximated by Gaussians, and because there exists many well-known results for these distributions that we can exploit. Also, because EM is guaranteed to move in the direction of increasing likelihood within the base of a local maximum [21], we must define a distribution over the solution space that will reflect this behavior. A Gaussian distribution will assign higher probability to the event of a set of model parameters θ_0 converging to the maximum θ_1^* if it is within the base of the hill corresponding to θ_1^*. This density is found during the *Clustering* phase of the kMeans-EM algorithm described in Section V and can be integrated over to find the probability of an arbitrary θ converging to any identified maxima of $g(\theta)$.

We consider the kMeans-EM algorithm to have reached completion either when the search space has been exhausted, or when a certain measure of performance has been met; such as a bound on squared error, or a maximum with a desired $g(\theta)$ value has been found.

IV. Review of EM-Based Hybrid Model Learning

This section of the paper provides a brief review of EM for hybrid model learning. For a more in-depth exposition of EM for hybrid model learning, including all relevant derivations, we refer the reader to [14]. A description of general can be found in [24], [21], and EM for linear non-switching parameter estimation in [13], [11].

The Hybrid Model Learning for Linear Probabilistic Hybrid Automata (HML-LPHA) algorithm that is reviewed here can be thought of as a function that takes as input a set of model parameters θ_0 and returns a converged set of model parameters θ_f. We refer to this as labeling. This step is important for the *Initialization* and *Labeling* phases of the kMeans-EM algorithm. We run the HML-LPHA algorithm either for a predetermined number of iterations, or until convergence of the lower bound defined in (5).

A. Expectation Maximization

Expectation Maximization is an iterative approach for finding Maximum Likelihood parameter estimates from data that includes latent, or hidden, variables. More specifically, we wish to find the set of model parameters θ that maximizes the likelihood, or equivalently the log-likelihood

$$g(\theta) = \log p(\mathbf{Y}|\theta). \tag{3}$$

The likelihood $p(\mathbf{Y}|\theta)$ however, is not readily evaluated due to the presence of latent variables so that only the distribution $p(\mathbf{Y}, \mathbf{X}|\theta)$ is known explicitly. We can express the log-likelihood $g(\theta)$ as

$$g(\theta) = \log \int_{\mathbf{X}} p(\mathbf{Y}, \mathbf{X}|\theta) d\mathbf{X}. \tag{4}$$

The key difficulty in maximizing $g(\theta)$ is that it involves a logarithm over an integral (or a large summation), which is difficult to deal with in many cases [25]. It is, however, possible to create a lower bound to $g(\theta)$ that instead involves an integral or sum of logarithms, which *is* tractable. In EM, Jensen's inequality is used to give the lower bound:

$$\begin{aligned} g(\theta) &= \log \int_{\mathbf{X}} p(\mathbf{Y}, \mathbf{X}|\theta) d\mathbf{X} \\ &\geq \int_{\mathbf{X}} p(\mathbf{X}|\mathbf{Y}, \theta^r) \log \frac{p(\mathbf{Y}, \mathbf{X}|\theta)}{p(\mathbf{X}|\mathbf{Y}, \theta^r)} d\mathbf{X} := h(\theta|\theta^r), \end{aligned} \tag{5}$$

where θ^r is a guess for the value for the parameters θ at iteration r of the EM algorithm. This bound can be written in terms of an expectation over the hidden state X, and an 'entropy' term denoted $\mathcal{H}$ that does not depend on θ.

$$\begin{aligned} h(\theta|\theta^r) &= \int_{\mathbf{X}} p(\mathbf{X}|\mathbf{Y}, \theta^r) \log \frac{p(\mathbf{Y}, \mathbf{X}|\theta)}{p(\mathbf{X}|\mathbf{Y}, \theta^r)} d\mathbf{X} \\ &= E_{\mathbf{X}|\mathbf{Y}, \theta^r}\left[\log p(\mathbf{Y}, \mathbf{X}|\theta)\right] + \mathcal{H}. \end{aligned} \tag{6}$$

The key iteration in the general Expectation-Maximization algorithm first computes the lower bound, $h(\theta|\theta^r)$, to the likelihood function and then maximizes this lower bound to find a new set of parameters θ^{r+1}.

1) **Initialization:** Set $r = 1$. Initialize θ^r to initial guess.
2) **Expectation Step:** Given θ^r, calculate bound $h(\theta|\theta^r)$.
3) **Maximization Step:** Set θ^{r+1} to value of θ that maximizes bound $h(\theta|\theta^r)$.
4) **Convergence Check:** Evaluate $g(\theta^{r+1})$. If $g(\theta)$ has converged, stop. Otherwise set $r = r + 1$ and go to 2).

For the hybrid case, the posterior distribution over the hidden states $p(\mathbf{X}|\mathbf{Y}, \theta^r)$ is intractable and thus we must formulate an approximate EM method for finding a locally optimal set of model parameters θ. We now discuss the approximate EM method that makes up the HML-LPHA algorithm.

B. Approximate Expectation Maximization

Recall that we are interested in learning the continuous model parameters $\theta_c(\mathbf{m})$. We must therefore maximize the lower bound $h(\theta|\theta^r)$ with respect to $\theta_c(\mathbf{m})$ at each iteration r of EM. In the case of hybrid model learning, the hidden data X comprises both the hidden continuous state sequence $\mathbf{x}_0^{T+1}$ and the hidden discrete mode sequence $\mathbf{m}_0^T$. The observed data consists of the observation sequence $\mathbf{y}_1^{T+1}$. It is important to note that the distribution over the discrete state is an approximate distribution as denoted by the tilde, $\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1}, \theta^r)$. This approximation arises because the number of possible state sequences is exponential in time and thus not all discrete state sequences are used in the distribution calculation; instead, a subset $\mathcal{S}$ of state sequences are used. A discussion of how $\mathcal{S}$ is chosen and detailed computation of $\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1}, \theta^r)$ can be found in [14]. We move relevant results on how to compute parameter values for $\theta_c(\mathbf{m})^{r+1}$ to the Appendix.

V. Clustering the Model Parameters

In this section we discuss fitting k Gaussian clusters to the set of converged, or labeled, model parameters $\{\theta_{f_i}\}_{i=1}^n$. We begin by giving a brief description of K-Means clustering, we refer the reader to [26] for a more complete exposition on K-Means clustering.

K-Means Clustering Problem Statement: *Given a set of n data points in d-dimensional space $\mathbb{R}^d$, and an integer k, determine a set of k centroid points c_j, $j \in [1,\ldots,k]$ in $\mathbb{R}^d$ so as to minimize the mean squared distance from each data point in a set $\mathcal{G}_j$ to its nearest center c_j where $j \in \{1,\ldots,k\}$.*
More formally, K-Means clustering aims to minimize the cost function J:

$$J = \sum_{j=1}^{k} \sum_{i \in G_j} |x_i - c_j|^2 \tag{7}$$

Assuming that the data points are generated independently from k different multivariate distributions in $\mathbb{R}^d$, let $\{\mathbf{X}\} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}]$ denote a random vector from this distribution. We can group this data into k clusters each with mean μ_j and covariance Λ_j. Furthermore, we can evaluate the pdf of this distribution for any data point x_n:

$$p(x_n) = \frac{1}{\sqrt{2\pi^d |\Lambda_j|}} \exp(\tfrac{-1}{2}(x_n - \mu_j)^T \Lambda_j^{-1}(x_n - \mu_j))$$

The value of the pdf at x_n gives us an idea of how likely x_n is to belong to the Gaussian cluster with mean μ_j and covariance Λ_j. This is an important observation and is key for our application of k-means clustering to jumping out of local maxima in EM.

A. Clustering the $\{\theta_{f_i}\}_{i=1}^n$

The goal in this section is to provide a characterization of a cluster defined over sets of labeled parameters $\{\theta_{f_i}\}_{i=1}^n$.

Definition of a Cluster over a Set of Parameters: *We define a cluster G_j over a set of parameters to be a Gaussian distribution with a mean μ_j and a covariance Λ_j such that*
$\mu_j = c_j$= *centroid for cluster j*
Λ_j = *empirical covariance for cluster j*
The mean of each cluster is equivalent to the centroid of each cluster computed using the standard k-means clustering algorithm [26]. We now discuss the calculation of the empirical covariance.

In Section II we defined the set of parameters θ to be composed of various matrices A, B, C, D, Q, R, V, μ. In this section, we must define a *vector* of parameters $\vec{\theta}$ that decomposes all matrices component-wise so that we can compute the empirical variance of each cluster. Recall that $x \in \mathbb{R}^{n_x}$ and $y \in \mathbb{R}^{n_y}$

Definition $\vec{\theta}$: *We define the vector of parameters $\vec{\theta}$ such that*

$$\begin{aligned}\vec{\theta} =& [A_{11}, A_{12}, \ldots, A_{n_x n_x}, B_{11}, \ldots, B_{1 n_x},\\ & C_{11}, C_{12}, \ldots, C_{n_y n_y}, D_{1 n_y}, \ldots, D_{1 n_y},\\ & Q_{11}, Q_{12}, \ldots, Q_{n_x n_x}, R_{11}, R_{12}, \ldots, R_{n_y n_y},\\ & V_{11}, \ldots, V_{n_x n_x}, \mu_{11}, \ldots, \mu_{1 n_x}]^T\end{aligned} \tag{8}$$

If we view each set of parameters θ as a vector $\vec{\theta}$ and we have n such vectors, we now have n observations of each parameter value and can compute the empirical variance. This calculation becomes

$$\Lambda_j = \frac{1}{n-1}\left(\sum_{i=1}^{n}\left(\vec{\theta_{f_i}} - \vec{\mu_j}\right)\left(\vec{\theta_{f_i}} - \vec{\mu_j}\right)^T\right)$$

Knowing the mean and covariance of each Gaussian cluster, we can now evaluate the pdf of a given cluster G_j for any vector set of parameters $\vec{\theta}$.

$$\begin{aligned}p_j\left(\vec{\theta}\right) =& \frac{1}{\sqrt{(2\pi)^d |\Lambda_j|}}\\ & * exp\left(-\frac{1}{2}\left(\vec{\theta} - \vec{\mu_j}\right)^T (\Lambda_j)^{-1}\left(\vec{\theta} - \vec{\mu_j}\right)\right)\end{aligned}$$

We can use these pdfs to evaluate the probability of a set of parameters belonging to any characterized local maximum as derived in Section VI. We choose the number of clusters, k, to be the number of identified maxima of the likelihood function, and declare a new cluster when the probability of a converged set of parameters belonging to any discovered local maxima is below an arbitrary threshold. One technique for determining cluster validity is to monitor the cluster silhouette values [27]. Figure 5 demonstrates the average normalized separation between clusters as obtained by the silhouette value. This can be used to adjust the number of clusters k to improve clustering performance.

VI. Finding An Optimal Restart: $\widetilde{\theta}_0$

In the last section we discussed how to cluster sets of parameters into Gaussians whose means and covariances approximate the location of, and shape of, the local maxima of the log-likelihood function respectively. In the present section we design a heuristic function, $s(\theta)$ that incorporates this cluster information and can be optimized to yield a new initialization set of parameters, $\widetilde{\theta}_0$, that 1) minimizes the probability of belonging to any discovered maximum of $g(\theta)$, and 2) maximizes the probability of being an *a priori* optimal set of parameters.

We define the probability that a certain set of model parameters θ belongs to cluster j as $q_j(\theta)$:

$$q_j(\theta) = P\{\theta \in G_j(\mu_j, \Lambda_j)\} \tag{9}$$

where G_j denotes cluster j with mean μ_j and covariance Λ_j

If we denote our prior distribution over parameters, $p_0(\theta)$, our maximization becomes

$$\begin{aligned}& s(\theta) = p_0(\theta)\prod_{j=1}^{k}(1 - q_j(\theta))\\ & \widetilde{\theta}_0 = \arg\max_{\theta} s(\theta)\\ & \text{where}\\ & p_0(\theta) \sim \mathcal{N}(\mu, \Lambda)\end{aligned} \tag{10}$$

The effect of the prior on the optimization can be mitigated arbitrarily. We use a basic Gaussian distribution that does not assume previous knowledge of where valid model parameters

lie and prevents solutions that are always at the edge of the search space. This prior is updated at each iteration of the kMeans-EM algorithm, where the mean is an average over all converged sets of model parameters labeled by the Approximate EM algorithm. We assign a large covariance to this prior that reflects its non-informativeness. Increasing the covariance can be compared to that of using the least-informative prior, *i.e.* a uniform distribution. If more *a priori* information is available about where valid model parameters can be found in the search space, this can be incorporated into the prior and used to focus the search. Figure 2 demonstrates the effect of the prior on $s(\theta)$.

In order to perform the optimization in (10) we must find $q_j(\theta)$, the probability of a set of parameters, θ, belonging to cluster j. Letting $\mathcal{E}$ denote the ellipse defined by the covariance of p_j, and passing through θ, we define $q_j(\theta)$ mathematically as

$$q_j(\theta) = \int_{\mathcal{E}} p_j(\theta)$$
$$\text{where}$$
$$p_j(\theta) \sim \mathcal{N}(\mu_j, \Lambda_j)$$

Integrating over Gaussians cannot generally be done in closed form, however, integrating a Gaussian over the area of a disc or ellipse (for elliptical Gaussians) *is* possible in closed form. We take advantage of this to compute $q_j(\theta)$ in closed form. For the case of a zero-mean Gaussian, $\mu_j = 0$:

$$\begin{aligned} &P\{\theta \text{ being outside of cluster } j\} = \int_{\mathcal{E}} p_j(x)dx \\ &p_j(x) = \frac{1}{\sqrt{(2\pi)^d det(\Lambda_j)}} \exp(\frac{-1}{2} x^T \Lambda_j^{-1} x) \\ &\Lambda = U^T \Sigma U \\ &\mathcal{E} = \left\{x : x^T \Lambda^{-1} x \leq \theta^T \Lambda^{-1} \theta\right\} \\ &\text{where } \Sigma \text{ is diagonal, } U \text{ is an orthogonal matrix,} \\ &\text{and } d \text{ is the dimension of } x \end{aligned} \tag{11}$$

We present the result of this integration here and place its derivation in the appendix for the interested reader. Generalizing the formulation in (11) to the non zero-mean case where $\mu_j \neq 0$,

$$q_j(\theta) = \exp\left(\frac{-1}{2}\left[(\theta - \mu_j)^T \Lambda_j^{-1} (\theta - \mu_j)\right]^2\right) \tag{12}$$

Thus we find a closed form solution for the probability of a set of parameters θ being in cluster j. We can now use this result in our objective function expression.

$$\begin{aligned} &s(\theta) = p_0(\theta) \prod_{i=1}^{k} (1 - \exp(\frac{-1}{2}\left[(\theta - \mu_i)^T \Lambda^{-1} (\theta - \mu_i)\right]^2)) \\ &\widetilde{\theta}_0 = \arg\max_{\theta} s(\theta) \end{aligned} \tag{13}$$

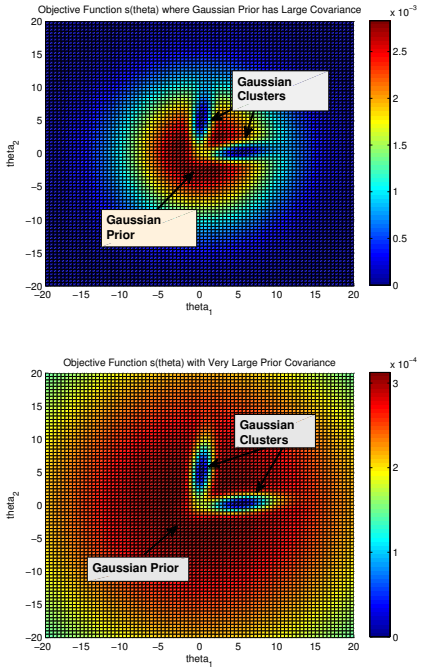

Fig. 2. Example of two-dimensional plot of $s(\theta)$ using a prior centered at $\theta_1 = 0, \theta_2 = 0$ and two elliptical Gaussian clusters. The covariance of the prior is varied, where a smaller covariance focuses the search to areas of high p_0 value, and a larger covariance mitigates the effect of the prior and allows a more uniform treatment of the search space.

Our objective function is of the form of a Gaussian prior multiplying a product of exponential functions. The form of this objective function makes an analytical solution very challenging. Although a closed form solution is not available, an off-the-shelf numerical optimization method can optimize $s(\theta)$. We emphasize that there are other heuristic functions that embody the same principles that we used to derive $s(\theta)$ and may be easier to optimize in certain cases. We derive two such heuristics in [28]: one in the form of a rational function of polynomials, and one in the form of a quadratic objective function whose constraints may be linearized and solved using a quadratic program. We avoid presenting these alternatives in the present paper due to space limitations.

VII. Simulation Results

We evaluate the kMeans-EM algorithm implemented on Autonomous Underwater Vehicle (AUV) data from the Monterey Bay Aquarium Research Institute (MBARI) in California, USA. The linearized AUV discrete time longitudinal dynamics are used for this simulation. We refer the refer the reader to [29] for more information on the AUV dynamics model. We consider the hybrid model where $\mathbf{m} \in \{\text{nominal, fail1}\}$, where *fail1* is a failure mode where the response to elevator angle is reduced. We contrast our results to a Random Restarts baseline method. Like kMeans-EM, the Random Restarts method can be implemented on approximate EM, but chooses new initializations in a random non-strategic fashion. Other global optimization methods that carefully choose initializations, like standard Simulation Annealing and Genetic algorithms, are not well-suited for

problems with hidden state and intractable likelihood functions. Our empirical analysis demonstrates 1) the result of optimizing the objective function $s(\theta)$ to find a new initialization point, 2) aggregate results demonstrating iteratively increasing performance over Random Restarts in the areas of improved likelihood of learned model parameters and decreasing Euclidean distance to true model parameters, and 3) aggregate statistics on clustering efficiency.

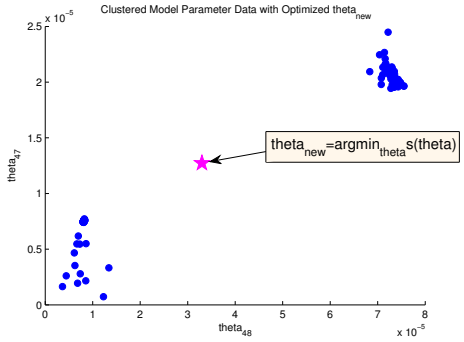

Fig. 3. Plot of two clusters with the optimal initialization point, found via a maximization of $s(\theta)$, shown as the purple star.

Our empirical results on the improvement of the log-likelihood in Fig. 4 are averaged over 20 trials and demonstrate that in 50 iterations the Random Restarts method shows a maximum improvement of only 1% in $g(\theta)$, indicating inefficient rediscovery of the same local maxima hills or discovery of non-optimal local maxima. KMeans-EM finds converged sets of model parameters with iteratively improving $g(\theta)$ and a clear trend of discovering new, more optimal maxima without remaining stuck in a single non-optimal maximum. We note that the plot for the log-likelihood value was found by running kMeans-EM on the non-hybrid case such that the actual value of $g(\theta)$ can be computed as in [11] and no approximation of the posterior density over hidden states is necessary. Figure 4, showing Euclidean distance between the learned model parameters and the true model parameters, further supports the trend of the kMeans-EM algorithm iteratively finding better solutions closer to the optimal solution. This plot shows that in 10 iterations, the kMeans-EM algorithm reduces distance to the global maximum by 16% on average, calculated over 100 trials. Each trial includes one full run of Hybrid Model Learning which takes up the bulk of the computation time, which, for our problem with 2 discrete modes and 4 continuous states takes about 6 minutes per trial on a standard Intel Core 2 Duo T7500 2.2GHz processor.

We find that the normalized separation between clusters fluctuates around the average value of 0.6, Fig. 5, which indicates that the clusters may not be currently separated in an optimal manner. Severe mis-classification of clusters would negatively influence the accuracy of our objective function $s(\theta)$ leading to reduced performance. Much work has been dedicated to improving the clustering efficiency of the K-Means algorithm [23], although we do not currently make use of these methods. Our aggregate results demonstrate however, that the kMeans-EM algorithm outperforms a Random Restarts method for our application, even if the clustering is non-optimal.

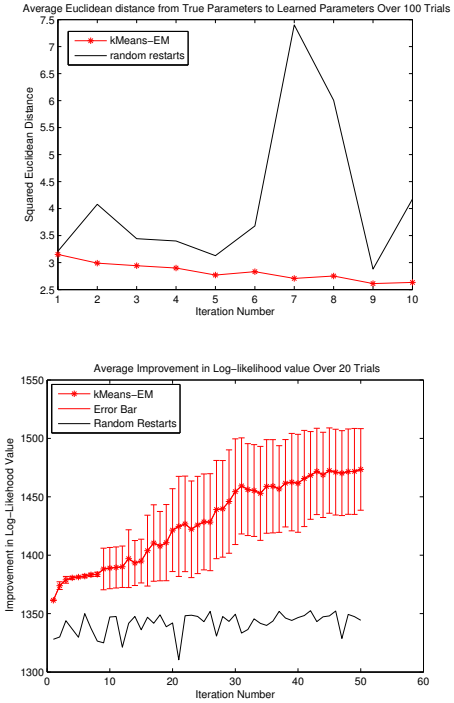

Fig. 4. Euclidean distance between best learned model and true model, averaged over 100 trials followed by a plot of the average relative improvement in the likelihood value of the best learned set of model parameters versus iteration number, for 50 clusters averaged over 20 trails. The improved log-likelihood value is taken with respect to the $g(\theta)$ value of the initialization set of parameters.

VIII. Conclusion

We have presented an approach to learning the continuous model parameters θ_c of hybrid systems that avoids getting stuck in locally optimal solutions that are subject to the quality of the initialization. This approach does not rely on the ability to evaluate the likelihood or its gradient directly, which is intractable for the hybrid case where the posterior distribution over the hidden state cannot be obtained in closed form. Thus we believe that this approach can be generalized to other problems of interest with multi-modal objective functions that have intractable forms and whose local maxima can be reasonably approximated by Gaussians.

We evaluate the kMeans-EM algorithm on MBARI AUV data. Our aggregate results averaged over as many as 100 trials demonstrate decreasing Euclidean distance between the best learned model parameters and true model parameters, as well as iterative improvement in log-likelihood value. We found that our algorithm increasingly outperforms a standard random restarts method that expends much computational resources re-discovering locally optimal solutions, even in the cases where the K-Means clustering is suboptimal as shown by monitoring the silhouette values of the clusters.

Directions for future work include incorporating better methods for determining the number of clusters such as

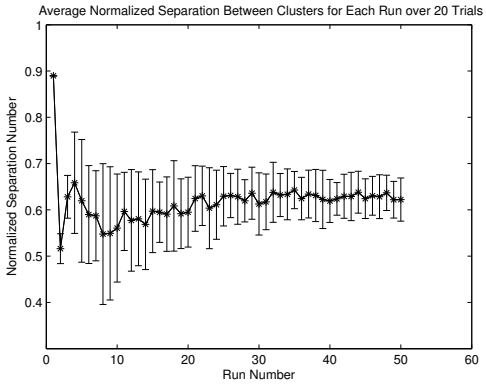

Fig. 5. Normalized separation between clusters, or mean silhouette value, averaged over 50 clustering iterations and 20 trials of the kMeans-EM algorithm.

an adaptive method or one of the many investigated for k-means clustering as in [23]. Other directions include improvement of the hybrid model learning step by reducing the dimensionality of the problem in similar spirit to [30], or by replacing the hybrid model learning step with a stochastic approach adjusted to not rely on explicit calculation of the likelihood or its gradient. Compatibility of the algorithm presented here with other identification techniques such as [31] would also be interesting.

ACKNOWLEDGEMENTS

We would like to thank the Bell Labs Graduate Research Fellowship for funding this work, as well as various people who have provided valuable feedback including L.P. Kaelbling, L. Blackmore, E. Abbe, and the CDC reviewers.

IX. APPENDIX

A. Approximate EM Results

The optimal values for $A(\mathbf{m})$ and $B(\mathbf{m})$ are found by summing the LTI results from [12] over the mode sequences in $\mathcal{S}$ to give the following equations:

$$\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}P_{t+1,t}(\mathbf{m}_0^T)\right)=$$
$$A^*(\mathbf{m})\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}P_t(\mathbf{m}_0^T)\right)$$
$$+B^*(\mathbf{m})\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\mathbf{u}_t\hat{\mathbf{x}}_t{}'(\mathbf{m}_0^T)\right) \tag{14}$$

$$\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\hat{\mathbf{x}}_{t+1}\mathbf{u}_t{}'\right)=$$
$$A^*(\mathbf{m})\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\hat{\mathbf{x}}_t(\mathbf{m}_0^T)\mathbf{u}_t'\right)$$
$$+B^*(\mathbf{m})\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\mathbf{u}_t\mathbf{u}_t{}'\right), \tag{15}$$

where $\mathcal{F}(\mathbf{m}_0^T)$ is the set of time steps in the sequence $\mathbf{m}_0^T$ for which the mode is $\mathbf{m}$. Members of $\mathcal{F}(\mathbf{m}_0^T)$ are integers in the range $[0,T]$. Solving the set of linear equations (14), (15) yields the optimal values for $A(\mathbf{m})$ and $B(\mathbf{m})$. Similarly the optimal values for $C(\mathbf{m})$ and $D(\mathbf{m})$ are found by performing a weighted sum over the LTI results from [12] to give the system of linear equations:

$$\sum_{\mathbf{m}_0^T}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\mathbf{y}_{t+1}\hat{\mathbf{x}}_{t+1}{}'(\mathbf{m}_0^T)\right)=$$
$$C^*(\mathbf{m})\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}P_{t+1}(\mathbf{m}_0^T)\right)$$
$$+D^*(\mathbf{m})\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\mathbf{u}_t\hat{\mathbf{x}}_{t+1}{}'(\mathbf{m}_0^T)\right) \tag{16}$$

$$\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_1^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\mathbf{y}_{t+1}\mathbf{u}_t{}'\right)=$$
$$C^*(\mathbf{m})\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\hat{\mathbf{x}}_{t+1}(\mathbf{m}_0^T)\mathbf{u}_t{}'\right)$$
$$+D^*(\mathbf{m})\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\mathbf{u}_t\mathbf{u}_t{}'\right). \tag{17}$$

Using the optimal values for $A(\mathbf{m})$, $B(\mathbf{m})$, $C(\mathbf{m})$ and $D(\mathbf{m})$ we obtain the optimal covariance matrices for the noise processes:

$$Q^*(\mathbf{m})=\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\frac{\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)}{|\mathcal{F}(\mathbf{m}_0^T)|}\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\Big(P_{t+1}(\mathbf{m}_0^T)\right.$$
$$\left.-A^*(\mathbf{m})P_{t,t+1}(\mathbf{m}_0^T)-B^*(\mathbf{m})\mathbf{u}_t\hat{\mathbf{x}}_{t+1}{}'(\mathbf{m}_0^T)\Big)\right) \tag{18}$$

$$R^*(\mathbf{m})=\sum_{\mathbf{m}_0^T\in\mathcal{S}}\left(\frac{\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)}{|\mathcal{F}(\mathbf{m}_0^T)|}\sum_{t\in\mathcal{F}(\mathbf{m}_0^T)}\Big(\mathbf{y}_{t+1}\right.$$
$$\left.-C^*(\mathbf{m})\hat{\mathbf{x}}_{t+1}(\mathbf{m}_0^T)-D^*(\mathbf{m})\mathbf{u}_t\Big)\mathbf{y}_{t+1}{}'\right). \tag{19}$$

Finally, the optimal parameters for the initial continuous distribution are given by:

$$\mu^*(\mathbf{m})=\sum_{\{\mathbf{m}_0^T|\mathbf{m}_0=\mathbf{m}\}}\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)\hat{\mathbf{x}}_0{}'(\mathbf{m}_0^T)$$
$$V^*(\mathbf{m})=\sum_{\{\mathbf{m}_0^T|\mathbf{m}_0=\mathbf{m}\}}\tilde{p}(\mathbf{m}_0^T|\mathbf{y}_1^{T+1},\theta^r)P_{0,0}(\mathbf{m}_0^T), \tag{20}$$

where the summation is over the discrete mode sequences $\mathbf{m}_0^T$ for which the initial mode $\mathbf{m}_0$ is the same as $\mathbf{m}$.
In (14) through (20) we use the following definitions:

$$\hat{\mathbf{x}}_t(\mathbf{m}_0^T)=E\big[\mathbf{x}_t|\mathbf{m}_0^T,\mathbf{y}_1^{T+1},\theta^r\big]$$
$$P_{t_1,t_2}(\mathbf{m}_0^T)=E\big[\mathbf{x}_{t_1}\mathbf{x}_{t_2}{}'|\mathbf{m}_0^T,\mathbf{y}_1^{T+1},\theta^r\big]. \tag{21}$$

The mean and covariance $\hat{\mathbf{x}}_t(\mathbf{m}_0^T)$ and $P_{t_1,t_2}(\mathbf{m}_0^T)$ are found in the E-step of the algorithm via a Kalman Smoother. The reader is referred to [14] for more comprehensive explanation of the distributions over the hidden state and relevant derivations.

B. Computation of $q_j(\theta)$

We make the following substitutions into equation (11).

$$\begin{aligned} & x^T\Lambda^{-1}x \\ &= x^TU^T\Sigma^{-1}Ux \\ &= y^T\Sigma^{-1}y \\ &= y_1^2\lambda_1^{-1} + y_2^2\lambda_2^{-1} \text{ for the two-dimensional case} \end{aligned}$$

Our integral becomes

$$\int_{y^T\Sigma^{-1}y \le \theta^T\Lambda_j^{-1}\theta} \frac{1}{\sqrt{(2\pi)^d det(\Sigma)}} \exp(\frac{-1}{2}y^T\Sigma^{-1}y)dy_1\cdots dy_n$$

Expanding out the above integral for the two-dimensional case, our integral becomes:

$$\int_{y_1^2\lambda_1^{-1}+y_2^2\lambda_2^{-1}\le\alpha} \frac{1}{\sqrt{2\pi\lambda_1}}\exp(\frac{-y_1^2}{2\lambda_1})\frac{1}{\sqrt{2\pi\lambda_2}}\exp(\frac{-y_2^2}{2\lambda_2})dy_1dy_2 \tag{22}$$

where

$\alpha = \theta^T\Lambda^{-1}\theta$ as before

Completing the integral in polar coordinates and generalizing to the multivariate Gaussian case with non-zero mean μ_j we find the probability of any set of parameters θ belonging to cluster j, which we denote G_j.

$$\begin{aligned} P\left\{\theta \in G_j(\mu_j,\Lambda_j)\right\} =& \exp(\frac{-1}{2}\left[(\theta-\mu_j)^T\Lambda_j^{-1}(\theta-\mu_j)\right]^2) \\ =& q_j(\theta) \end{aligned} \tag{23}$$

REFERENCES

[1] P. Peursum, S. Venkatesh, and G. West, "Observation-switching linear dynamic systems for tracking humans through unexpected partial occlusions by scene objects," *IEEE 18th Intl. Conf. on Pattern Recognition*, vol. 4, pp. 929–934, 2006.

[2] V. Pavlovic, J. M. Rehg, and J. Maccormick, "Learning switching linear models of human motion," 2000, pp. 981–987.

[3] C. J. Kim, "Dynamic linear models with Markov-switching," *Journal of Econometrics*, vol. 60, no. 1–2, pp. 1–22, 1992.

[4] H. Balakrishnan, I. Hwang, J. Jang, and C. Tomlin, "Inference methods for autonomous stochastic linear hybrid systems," in *Hybrid Systems: Computation and Control, HSCC 2007*, ser. Lecture Notes in Computer Science, G. Goos, J. Hartmanis, and J. van Leeuwen, Eds. Springer Verlag, 2004, vol. 2993, pp. 64–79.

[5] M. Henry, "Model-based estimation of probabilistic hybrid automata," Master's thesis, Massachusetts Institute of Technology, Cambridge, 2002.

[6] L. Blackmore, A. Bektassov, M. Ono, and B. C. Williams, "Robust, optimal predictive control of jump markov linear systems using particles," in *Hybrid Systems: Computation and Control, HSCC 2007*, ser. Lecture Notes in Computer Science, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. Springer Verlag, 2007, vol. 4416, pp. 104–117.

[7] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump markov linear systems," 2000. [Online]. Available: citeseer.ist.psu.edu/doucet00stochastic.html

[8] J.Roll, "Local and piecewise affine approaches to system identification," Ph.D. dissertation, Linkoping University, Sweden, 2003.

[9] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems: a tutoiral," *European Journal of Control*, 2007.

[10] Z. Ghahramani and G. E. Hinton, "Variational learning for switching state-space models," *Neur. Comp.*, vol. 12, no. 4, pp. 831–864, 2000.

[11] Ghahramani and Hinton, "Parameter estimation for linear dynamical systems," 1996.

[12] S. Cheng and P. N. Sabes, "Modeling sensorimotor learning with linear dynamical systems," *Neural Computation*, vol. 18, no. 4, pp. 760–793, 2006.

[13] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the em algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[14] L. Blackmore, S. Gil, S. Chung, and B.Williams, "Model learning for switching linear systems with autonomous mode transitions," *Decision and Control, 2007 46th IEEE Conference on*, pp. 4648–4655, Dec. 2007.

[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[16] M. Mitchell, *An introduction to genetic algorithms*. Cambridge, MA, USA: MIT Press, 1996.

[17] J. C. Spall and S. Member, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, pp. 332–341, 1992.

[18] G. C. G. Wei and M. A. Tanner, "A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms," *Journal of the American Statistical Association*, no. 411, pp. 699–704.

[19] C. Biernacki, G. Celeux, and G. Govarert, "Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models," *Comput. Stat. Data Anal.*, 2003.

[20] F.Glover and M.Laguna, *TABU Search*. Springer Press, 1997.

[21] T. Minka, "Expectation-maximization as lower bound maximization," 1998. [Online]. Available: citeseer.ist.psu.edu/minka98expectationmaximization.html

[22] M. Hofbaur and B. Williams, "Mode estimation of probabilistic hybrid systems," *Lecture Notes in Computer Science*, vol. 2289, pp. 253–266, 2002.

[23] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*. New York, NY, USA: ACM, 2002, pp. 600–607.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the e m algorithm," *J. Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 2006.

[25] F. Dellaert, "The expectation maximization algorithm," College of Computing, Georgia Institute of Technology, Tech. Rep. GIT-GVU-02-20, 2002.

[26] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.

[27] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.

[28] S. Gil, "Robust learning of probabilisitic hybrid models," Master's thesis, Massachusetts Institute of Technology, Cambridge, 2008.

[29] R. McEwen, "Modeling and control of a variable-length auv," Tech. Rep., 2006.

[30] D. Koller and S. Mehran, "Toward optimal feature selection," Stanford, Tech. Rep., 1996.

[31] A. L. Juloski, S. Weiland, and W. Heemels, "A bayesian approach to identification of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, 2005.