24th Mediterranean Conference on Control and Automation (MED)
June 21-24, 2016, Athens, Greece

Learning and Reachability Analysis for Stochastic Hybrid Systems using Mixtures of Gaussian Processes

Hamzah Abdel-Aziz and Xenofon Koutsoukos
Institute for Software Integrated Systems
Department of Electrical Engineering and Computer Science
Vanderbilt University, Nashville, TN, 37235, USA
Email: {hamzah.abdelaziz, xenofon.koutsoukos}@vanderbilt.edu

***Abstract*— Robust and efficient modeling and reachability analysis of stochastic hybrid systems for control and decision is very demanding and challenging task. In this paper, we develop a novel methodology which provides a model of stochastic hybrid systems based on Gaussian Processes. This model uses observed data to update the model in an online fashion. In addition, we provide an efficient reachability analysis methodology that utilizes mixtures of Gaussian Processes to predict the reachable states for a finite horizon. We demonstrate the efficiency of the proposed approach using a multi-room heating system. Despite dynamic changes in the system parameters, the results show that the model can adapt and efficiently predict the reachable states.**

I. Introduction

Modeling and analyzing complex systems for control and decision is very important task with several challenges, especially for systems that exhibit stochastic and hybrid dynamics known as Stochastic Hybrid Systems (SHS). Parametric modeling of many of these complex systems is very difficult. For instance, the heat exchange parameters in buildings between several zones and the environment cannot be easily identified. Further, such systems have continuous and discrete dynamics coupled with uncertain behavior that may result in completely different system trajectories, and therefore, prediction of the system behavior is a difficult task.

Reachability analysis is a typical problem in SHS where given the initial states, it is necessary to predict the reachable states for some finite time horizon. Reachability analysis is used typically to verify system safety and stability. To that end, different approximation methods have been proposed to estimate the reachable states for such as polygonal flow-pipe approximation [1], and ellipsoidal approximation [2]. Optimization techniques such as face lifting [3] have been also used. Reachability analysis for SHS has additional level of complexity because of the presence of uncertainty in its behavior. This problem has been investigated in the literature through several methods [4]. Methods based on analytical estimation have been used to solve the reachability problem, for instance a quadratic form, called Dirichlet forms, associated with a right-Markov process is used in [5]. Other approximation methods based on numerical estimations are also used such as Markov Chain approximations [6] [7], and dynamic programming [8]. Probabilistic methods based on randomize algorithms have been considered such as Monte-Carlo methods [9] and multilevel splitting (MLS) variance reduction [10]. Finally, statistical methods aim to leverage available data and are an area of active research [4].

The contribution of this paper is twofold: First, we present a non-parametric SHS model by utilizing Gaussian Processes [11] to represent the continuous dynamics and learn the model from data. Moreover, the model can be updated efficiently online during the system execution, and therefore, it can adapt to dynamic changes in the system parameters. Second, we propose a methodology for reachability analysis of SHS using a statistical method based on Mixtures of Gaussian Processes [12] to represent the reachable states for a finite horizon. We demonstrate the efficiency of the proposed approach using a multi-room heating system. Despite the dynamic changes, the results show that the model can adapt to these changes and predicts efficiently the reachable states.

The paper is organized as follows: Section II summarizes Gaussian Process (GP) focusing on model learning and prediction. We describe in section III our proposed SHS model. Then, in section IV, we illustrate our online learning algorithm. Also, we discuss the SHS reachability analysis problem and illustrate our proposed statistical approximation to solve it. Finally in section V, we discuss the implementation and evaluation of our method using a multi-room heating system as an illustrating example.

II. Background

A. Gaussian process model

Gaussian Process (GP) is a non-parametric model which uses the observed data to model the system behavior [11]. A GP is identified by its mean and covariance functions. The mean function represents the expected value before observing any data and the covariance function (also called kernel) identifies the expected correlation between the observed data. For a function $y = f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^D$, the mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ are defined as:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \end{aligned} \tag{1}$$

Thus, the function modeled by the GP can be written as:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

Typically, we use a zero mean function for simplicity and squared exponential (SE) covariance kernel for its expressiveness combined with a noise kernel. Therefore, the mean and covariance functions are expressed as:

$$\begin{aligned} m(\mathbf{x}) &= 0, \\ k(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 exp[-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^T \Lambda^{-1}(\mathbf{x}-\mathbf{x}')] \\ &+ \delta_{\mathbf{x},\mathbf{x}'}\sigma_\omega^2 \end{aligned} \tag{2}$$

where σ_f is the kernel signal variance, $\Lambda := diag([l_1^2, \cdots, l_D^2])$ is the characteristic length-scales matrix, δ is the Kronecker delta, and σ_ω is the noise variance. The above GP model builds a probability distribution over the modeled function by mapping n-samples $\mathbf{X}$ of a continuous variable $\mathbf{x}$ to a random vector $\mathbf{y}$ with a Gaussian joint distribution, such that:

$$p(\mathbf{y}) \sim \mathcal{N}(0, \mathbf{K}(\mathbf{X}, \mathbf{X})) \tag{3}$$

where $\mathbf{K}$ is $nD \times nD$ covariance matrix generated by (2).

B. GP Model Learning

We define a set of n observations as $\mathcal{D} = \{(\mathbf{x}_i, y_i)|i = 1, ..., n\}$ and $\Theta := (\sigma_f, l_1^2, \cdots, l_D^2, \sigma_\omega)$ as the GP hyperparameters. The learning objective is to identify the model hyperparameters Θ such that they fit the observed data $\mathcal{D}$. The model hyperparameters can be learned by evidence maximization [11], [13], where the hyperparameters Θ are selected such that the following marginal likelihood (evidence) is maximized.

$$\hat{\Theta} = \arg\max_{\Theta} \log p(\mathbf{y}|\Theta, \mathcal{D})$$
$$\log p(\mathbf{y}|\Theta, \mathcal{D}) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}\mathbf{y} - \frac{1}{2}log(|\mathbf{K}|) - \frac{n}{2}log(2\pi)$$

C. GP prediction at certain input

We are interested in the posterior distribution $p(\mathbf{y}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y})$ given test inputs $\mathbf{X}_*$ and training data $(\mathbf{X}, \mathbf{y})$. After observing data $\mathcal{D}$, and according to (3), the joint distribution of the known $\mathbf{y}$ and the unknown $\mathbf{y}_*$ is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X}_*) \\ K(\mathbf{X}_*,\mathbf{X}) & K(\mathbf{X}_*,\mathbf{X}_*) \end{bmatrix}\right).$$

Therefore, the posterior distribution $p(\mathbf{y}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y})$ is also a conditional Gaussian distribution with a mean and a covariance given by:

$$\begin{aligned} \mathbb{E}[\mathbf{y}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*] &= \mathbf{K}_*^T\boldsymbol{\beta} \\ Var[\mathbf{y}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*] &= \mathbf{K}_{**} - \mathbf{K}_*^T(\mathbf{K} + \sigma_\omega^2\boldsymbol{I})^{-1}\mathbf{K}_* \end{aligned} \tag{4}$$

where $\mathbf{K}_* := k(\mathbf{X}, \mathbf{X}_*)$, $\mathbf{K}_{**} := k(\mathbf{X}_*, \mathbf{X}_*)$, $\mathbf{K} := k(\mathbf{X}, \mathbf{X})$ and $\boldsymbol{\beta} := (\mathbf{K} + \sigma_\omega^2\boldsymbol{I})^{-1}\mathbf{y}$.

D. GP prediction at uncertain input

The posterior distribution shown in (4) is a prediction model for a given test input $\mathbf{X}_*$. However, this prediction model is not valid if $\mathbf{X}_*$ is defined by a probability distribution. For instance, if the test input is defined as a Gaussian joint distribution (i.e. $p(\mathbf{X}_*) \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$), the GP posterior distribution must be calculated by:

$$p(\mathbf{y}_*) = \int\int p((\mathbf{y}_*|\mathbf{X}_*)p(\mathbf{X}_*)d\mathbf{y}_*d\mathbf{X}_*. \tag{5}$$

The prediction distribution shown in (5) is analytically intractable [14] but we can approximate it as Gaussian (i.e., $p(\mathbf{y}_*) \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$) using an approximation method introduced in [14], such that:

$$\begin{aligned} \boldsymbol{\mu}_y &= \mathbb{E}[\mathbf{y}_*|\mu_*] \\ \boldsymbol{\Sigma}_y &= Var[\mathbf{y}_*|\mu_*] + \mathbf{V}\boldsymbol{\Sigma}_*\mathbf{V}^T + cov[\mathbf{y}_*, \mathbf{X}_*] + cov[\mathbf{X}_*, \mathbf{y}_*] \end{aligned} \tag{6}$$

where $\mathbb{E}[\mathbf{y}_*|\boldsymbol{\mu}_*]$ and $var[\mathbf{y}_*|\boldsymbol{\mu}_*]$ is the mean and covariance of the GP posterior calculated at the mean $\boldsymbol{\mu}_*$ of the input distribution as in (4) and $cov[\mathbf{X}_*, \mathbf{y}_*]$ is the cross-covariance between the input and output and it is given by $\boldsymbol{\Sigma}_*\mathbf{V}$ where $\mathbf{V}$ is defined by:

$$\mathbf{V} = \frac{\partial \boldsymbol{\mu}_y}{\partial \boldsymbol{\mu}_*} = \beta^T \frac{\partial k(\mathbf{X}, \boldsymbol{\mu}_*)}{\partial \boldsymbol{\mu}_*}$$

III. Stochastic Hybrid Systems

Stochastic hybrid systems exhibit continuous, discrete, and probabilistic behavior. In order to model these systems, we need to identify both the continuous and discrete components as well as the interaction between them, and abstract their dynamics using a stochastic process. In this work, we use Gaussian Processes to model the continuous dynamics and we consider systems with deterministic discrete dynamics. To formalize our model, we define $\mathcal{Q}$ as the set of discrete states and denote the continuous state space by $\mathbb{R}^D$. Thus, the system hybrid state space is defined as

$$\mathcal{S} := \cup_{q\in\mathcal{Q}}\{q\} \times \mathbb{R}^D.$$

The continuous dynamics evolves according to a stochastic process modeled by a GP which depends on the current discrete mode $q \in \mathcal{Q}$. However, the discrete state can change based on logical conditions described by guards. Furthermore, we consider systems with two inputs: Control inputs and external uncontrolled inputs from the system environment. The control inputs usually govern the transitions between the discrete states using a control policy $\pi(\mathcal{S}) : \mathcal{S} \to \mathcal{U}$ which maps the hybrid state space $\mathcal{S}$ into the control input space $\mathcal{U}$. The external inputs $v \in \mathcal{V}$ affect the continuous state evolution,and unlike the controlled inputs, there is no policy that can determine the external input. Thus, we model the external inputs as time-series model $E : \mathbb{N} \to \mathcal{V}$ in order to forecast the external inputs value over time.

A non-parametric SHS model is defined as a tuple $\mathcal{H} = (\mathcal{Q}, X, Init, \mathcal{U}, \mathcal{V}, \mathrm{A}, \mathrm{E})$

- $\mathcal{Q} := \{q_1, q_2, \cdots, q_m\}$, for some $m \in \mathbb{N}$, represents the discrete state space.
- X is a set of continuous variables in the Euclidean space $\mathbb{R}^D$.
- Init: $\mathcal{B}(\mathcal{S}) \to [0, 1]$ is an initial probability measure on the Borel space $\mathcal{B}(\mathcal{S})$ where $\mathcal{S} := \cup_{q\in\mathcal{Q}}\{q\} \times \mathbb{R}^D$.

- $\mathcal{U} \subset \mathbb{R}^E$, for some $E \in \mathbb{N}$, represents the control input space.
- $\mathcal{V} \subset \mathbb{R}^F$, for some $F \in \mathbb{N}$, represents the external uncontrolled input space.
- A assigns to each discrete state $q \in \mathcal{Q}$ a function $f_q(x, v) = f_q(\hat{\boldsymbol{x}}) \sim \mathcal{GP}_q(m(\hat{\boldsymbol{x}}), k(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}))$ modeled by a GP which defines a probability distribution given a continuous state $x \in \mathbb{R}^D$ and an external uncontrolled input $v \in \mathcal{V}$.
- E $\subset \mathcal{Q} \times \mathcal{Q}$ is a finite set of edges that represent the discrete transitions. Each discrete transition is a function of the current discrete mode q and the control input value u such that $q' = \delta(u, q)$ where $q' \in \mathcal{Q}$ is the new discrete mode.

For a finite time horizon $[0, N]$, a system trajectory is denoted by $\{s(k) = (q(k), x(k)), k \in [0, N]\}$ which is an execution of $\mathcal{H}$, with a control policy $\pi(\mathcal{S})$. A discrete-time execution algorithm of $\mathcal{H}$ is shown in Algorithm 1.

Algorithm 1 Discrete-time SHS execution algorithm

State Initialization: $s(0) = (q(0), x(0)) \in Init$
k $\leftarrow$ 0
while $k < N$ **do**
  ▷ Calculate the control input $u(k)$:
  $u(k) \leftarrow \pi(s(k))$
  ▷ Forecast the external input $v(k)$:
  $v(k) \leftarrow E(k)$
  ▷ Update the discrete mode $q(k+1)$:
  $q(k+1) \leftarrow \delta(q(k), u(k))$
  ▷ Update the continuous state $x(k+1)$:
  $x(k+1) \leftarrow f_{q(k+1)}(x(k), v(k)) \sim \mathcal{GP}_{q(k+1)}$
  k $\leftarrow$ k +1
end while

IV. Online Learning and Reachability Analysis

Reachability analysis aims at predicting the set of reachable states for a finite time horizon. Prediction can be performed as an iterative process since $s(k+1)$ depends on $s(k)$. However, it is a challenging task because: (1) Predicting the continuous state $x(k+1)$ distribution requires prediction at an uncertain input since $x(k)$ is identified by a probability distribution. Also, the continuous state $x(k+1)$ evolves differently for each discrete state $q(k)$; (2) the discrete mode $q(k+1)$ is calculated as discrete random distribution given the probability distribution of $s(k)$; and (3) the system trajectory depends on switching times between discrete states.

The proposed approach consists of three steps: (1) Collect data from the system; (2) Learn the system model $\mathcal{H}$ and the external input (forecast) model $E(k)$ using evidence maximization; and (3) Perform reachability analysis algorithm to predict the system behavior using the updated models. These steps are repeated periodically in an online fashion as shown in figure (1).

The observed data consist of the hybrid state, the control input, and the external input. Since the SHS model has a

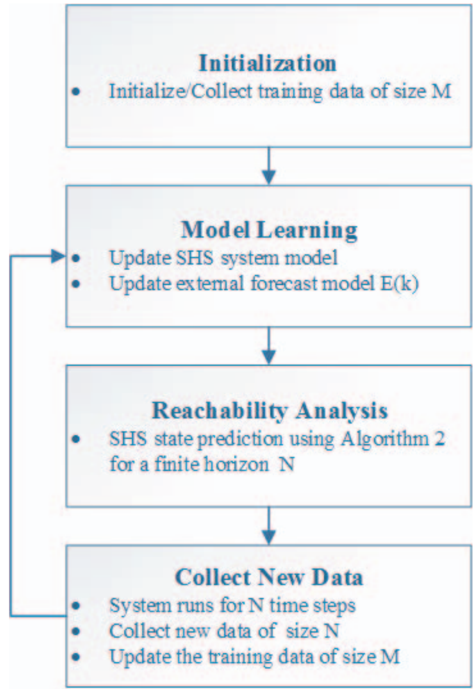

Fig. 1. Flow chart of the proposed approach

distinct GP model for each discrete state $q \in \mathcal{Q}$, the data is separated for each discrete mode such that:

$$\forall q \in \mathcal{Q}, \mathcal{D}_q = \{(\hat{\boldsymbol{x}}_i, \mathbf{y}_i) : i = 1, \cdots, m_q\}$$

where

$$\begin{aligned} \hat{\boldsymbol{x}} &= [\mathbf{x}(k), v(k)], \\ \mathbf{y} &= \mathbf{x}(k+1), \\ q(k+1) &= q \end{aligned}$$

During the system execution, we keep all data sets to be of size M such that $\sum_q m_q = M$. The data set associated with the forecast model $E(k)$ is defined as $\mathcal{D}_E = \{(k, v(k)) : k = 1, \cdots, M\}$ with also a size M. Upon collecting new data, we updated the data sets $\{\mathcal{D}_q : q \in \mathcal{Q}\}$ and $\mathcal{D}_E$, then we re-learn the SHS model $\mathcal{H}$ and the forecast model $E(k)$ respectively.

A. Mixtures of Gaussian Process for Reachability Analysis

We represent the reachable states of $\mathcal{H}$ using Mixtures of Gaussian Processes (MGP) [12]. An MGP consists of a latent discrete variable, typically called gating network, and a set of GP functions, called the experts. The state of the discrete variable specifies the GP function which used to calculate the system output at a given input. The MGP model is expressed as:

$$P(y|x) = \sum_{i=1}^{L} P(z = i|x)\mathcal{GP}_i(m_i(x), k_i(x, x)) \quad (7)$$

where y is the output, x is the input, z is the discrete latent variable with L states and $\mathcal{GP}_i$ is the GP function corresponding to the discrete state $z = i$. In our problem, we want to predict the probability of the continuous state $\mathbf{x}(k+1)$

given the probability of the hybrid state $s(k) = (\mathbf{x}(k), q(k))$. Hence, the MGP latent discrete variable represents the discrete state of the SHS model and the experts are the GP functions for each discrete state. Therefore, one-step state prediction model can be written as:

$$P(\mathbf{x}(k+1)|s(k)) = \sum_{i=1}^{m} P(q(k+1) = i|s(k)) f_i(\mathbf{x}(k), v(k)) \tag{8}$$

where $f_i(\mathbf{x}(k), v(k)) \sim \mathcal{GP}_i(m_i(\hat{\mathbf{x}}), k_i(\hat{\mathbf{x}}, \hat{\mathbf{x}}))$ and $\hat{\mathbf{x}}$ is the tuple $(\mathbf{x}, v)$. Reachability analysis requires to perform a multi-step prediction. To do so, we need to apply (8) iteratively. However, this equation depends on $p(\mathbf{x}(k))$ calculated from the previous iteration where it was calculated as Gaussian mixture model:

$$p(\mathbf{x}(k)) = \sum_{j=1}^{C} w_j \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{9}$$

where C is the number of Gaussian distribution components in the mixture, w_i is the weight of i^{th} Gaussian component with $\sum_{i=1}^{C} w_i = 1$, and $\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i$ is the mean and the variance of i^{th} Gaussian component respectively.

Therefore, calculating $P(\mathbf{x}(k+1)|s(k))$ iteratively from equation (8) is analytically intractable because the input of the MGP prediction model in (8) is uncertain (i.e. represented by Gaussian mixture model) as shown in (9). Therefore, we instead approximate the predictive model by propagate each mixture component in $p(\mathbf{x}(k))$ independently. The weight of each of those components along with its probability of switching to/staying in a discrete mode are used to calculate the new discrete state probability, hence, the approximated predictive distribution is defined as:

$$P(\mathbf{x}(k+1)|s(k)) = \sum_{i=1}^{Q}\sum_{j=1}^{C} w_j P(q(k+1) = i|\mathbf{x}_c(k), q_c(k)) \tilde{f}_i(\mathbf{x}_c(k), v(k)) \tag{10}$$

where $\mathbf{x}_c$ is the j^{th} Gaussian component of $p(\mathbf{x}(k))$ with weight w_j, mean $\boldsymbol{\mu}_j$ and variance $\boldsymbol{\Sigma}_j$, $q_c(k)$ is the discrete mode of the j^{th} Gaussian component, and $\tilde{f}(.)$ is its approximation GP function f_i defined in (6). Algorithm 2 illustrates the prediction of the reachable states of the SHS iteratively based on (10).

The prediction algorithm is computationally efficient and can be performed online. The most expensive part is computing the inverse covariance matrix which requires $O(n^3)$ time where n is the size of the data.

Algorithm 2 Discrete-time SHS state prediction

Input: $s(0)$, N
Output: $p(s(k))$ for $k \in [1, N]$

```
k ← 0
while k < N do
    ▷ Forecast the external input at time k
    v(k) ← E(k)
    for each s_c(k) ∈ s(k) do
        for each q ∈ Q do
            ▷ Calculate the discrete probability distribution
            p(q(k+1) = q|x_c(k)) ← p(π(s_c(k)) ∈ G_e)
            ▷ Calculate the new weight
            Wnew ← p(q(k+1) = q|x_c(k)) × w_c
            if Wnew > δ_w then    ▷ To ignore component with small probability
                x_c(k+1) ← f̃_q(x_c(k), v(k))
                add [x_c(k+1), Wnew, q] to x(k+1)
            end if
        end for
    end for
    k ← k +1
end while
```

V. Benchmark Example: Multi-room heating system

This section illustrates the implementation of the proposed model and methodology using a multi-room heating system that has been proposed as benchmark for reachability analysis [15]. The multi-room heating system comprises h rooms where each room has its own heater and user setting. Additionally, each room is affected by its adjacent rooms and the ambient temperature. The discrete state represents the heater mode $q_i = \{\texttt{ON}, \texttt{OFF}\}$ for each room and the continuous state is the rooms temperature $x_i : i \in h$. The continuous state x_i for each room evolves according to the following stochastic difference equation [8]:

$$\begin{aligned} x_i(k+1) = x_i(k) &+ b_i(x_a(k) - x_i(k)) \\ &+ \sum_{i \neq j} a_{ij}(x_j(k) - x_i(k)) + c_i \mathbb{I}_{Q_i}(q_i(k)) \\ &+ \omega_i(k) \end{aligned} \tag{11}$$

where $x_a(k)$ is the ambient temperature at time k, $\mathbb{I}_{Q_i}(\cdot)$ is the indicator function of set $Q_i = \{(q_1, \cdots, q_h) \in \mathcal{Q} : q_i = \texttt{ON}\}$, b_i is non-negative constant representing the average heat transfer rate from room i and the ambient x_a, a_{ij} is non-negative constant representing the average heat transfer rate from room i and room j, c_i is non-negative constant representing the heat rate supplied to room i by the heater ω_i is a Gaussian noise disturbance in room i.

The discrete transition function represents the heater operation using a typical controller where each room is controlled independently from the other rooms. The controller switches the heater on if the temperature gets below a lower threshold xl, and switches the heater off if the temperature exceeds upper threshold xu. Formally, the control policy can be described by:

$$\pi(s(k)) = \begin{cases} 0 & if\ q(k) = \texttt{ON}\ \&\ x(k) >= xu \\ 1 & if\ q(k) = \texttt{OFF}\ \&\ x(k) <= xl \end{cases} \tag{12}$$

The parameters (i.e. x_a, b_i, a_{ij} and c_i) in the multi-room heating system are hard to model for several reasons. They differ from building to building (e.g., different geometry and materials) and they may change during the system operation either abruptly (e.g., open window or door) or slowly because of aging. As a result, it is very hard to identify a parametric model of the system.

Our proposed method uses sensory data to learn a non-parametric SHS model of the system. Also, the system model can be updated online in a periodic fashion, and then, used to analyze the behavior by computing the reachable states. We consider the ambient temperature as the external uncontrolled input of the system $v(k) = x_a(k) \in \mathbb{R}$, the controller as the control policy $u(k) = \pi(s(k)) \in \{0, 1\}$, the room temperature vector as the continuous variable $x(k) \in \mathbb{R}^h$, and the heater state as the discrete state $\mathcal{Q} = \{q_1, q_2, \cdots, q_h\} = \{\texttt{ON}, \texttt{OFF}\}^h$.

To evaluate our approach, we implemented a system with two rooms (i.e. $h = 2$), and therefore, four discrete states. For each discrete state $q \in \mathcal{Q}$, the continuous state evolves according to

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{x}(k) + \Delta\mathbf{x}(k) \\ \Delta\mathbf{x}(k) &= f_q(\mathbf{x}(k), x_a(k)) \end{aligned} \tag{13}$$

where each mode q function $f_q(.) \sim \mathcal{GP}_q$ is modeled as a GP described in (1 -3). The control input $\mathbf{u}(k) = \pi(s(k))$ defines the guards for the discrete transitions. The uncontrolled external input is defined as $v(k) = x_a(k) = E(k)$, where $E(k)$ is a time-series model of the ambient temperature. We model $E(k)$ using a GP model introduced in [16] to forecast $v(k)$.

We have implemented the approach using Matlab. We use the parametric model shown in equation (11) with the following parameters: $b_1 = 0.4, b_2 = 0.45, a_{12} = a_{21} = 0.5, c_1 = 25, c_2 = 27$, and $\omega_i \sim \mathcal{N}(0, 5)$ to represent the physical system and to gather data from. We generate data with a time-step of 1 min and we collect data for 6 hours to learn our models (i.e., $M = 360$). Our goal is to predict the system behavior every hour for the next one hour (i.e., $N = 60$). We train the model using the first 6 hours simulation data, then apply the online approach to predict the system behavior for the next hour. Next, we generate the data for the predicted hour and re-learn the model. We repeat these periodic 1-hour predict/learn steps for 10 hours. To emulate changes in the system parameters, we change the heat transfer rate with the ambient (i.e. b_i parameters) in the second hour to be $b_1 = 0.8$ and $b_2 = 0.6$. In order to evaluate the advantages of online learning and reachability analysis, we also implement the prediction without updating the models online. In other words, we learn the model once offline using the initial training data of six hours only.

To evaluate the prediction accuracy, we use the following weighted mean absolute error (WMAE) metric:

$$WMAE = \frac{1}{N} \sum_k \Big(\frac{1}{C(k)} \sum_{c=1}^{C(k)} (|\mathbf{x}_m(k) - \boldsymbol{\mu}_c(k)| \times w_c(k)) \Big)$$

where $C(k)$ is the number of Gaussian components at time k, $\mathbf{x}_m(k)$ is the real system measurement at time k, and $\boldsymbol{\mu}_c(k)$ and $w_c(k)$ is the mean and weight of Gaussian component c at time k respectively.

The proposed approach is able to generate a statistical distribution of the reachable SHS states. Figure (2) shows the prediction distribution of the discrete mode, room 1 temperature, and room 2 temperature for the fourth hour of the system operation. On the other hand, Figure (3) shows also the prediction distribution using the model learned offline. The results in (2) show that the online algorithm tracks the system changes (i.e. the increase in heat leak) and manage to predict that the system requires more time to heat the room as the first discrete transition occurs around $t = 15min$. On the other hand, the offline approach was not able to adapt to this change and predicted the first discrete transition occurs around early (i.e. $t = 10min.$).

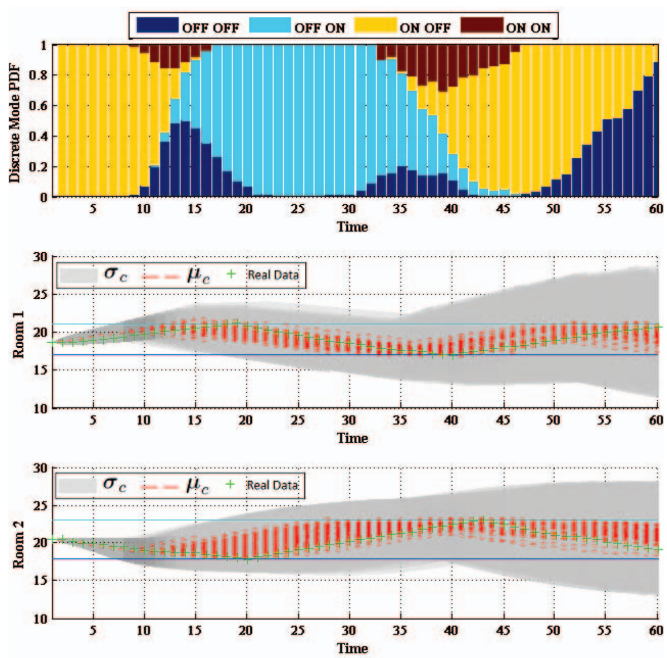

Fig. 2. Prediction distribuiot of hybrid state of the fourth hour using online learned model

The WMAE error for both online and offline models is shown in figure (4) for the ten hours of system operation. The results show that the error is keep decreasing as the model aggregates more data.

The average execution time of the learning and reachability analysis algorithms are 7.1 and 4.7 sec respectively. Finally, Figure (5) illustrates the accuracy of our reachability analysis MGP approximation by comparing with sample trajectories generated by Monte-Carlo simulation from the parametric model (11).

VI. Conclusion

In this paper, we introduced a data-driven SHS model and present an online learning approach using Gaussian Process. Also, we infer a statistical distribution of the model reachability states for a finite-horizon using Mixtures of Gaussian Processes. We illustrate the approach using a multi-room heating system benchmark. The results show the capability of our approach to adapt to system changes and to approximate

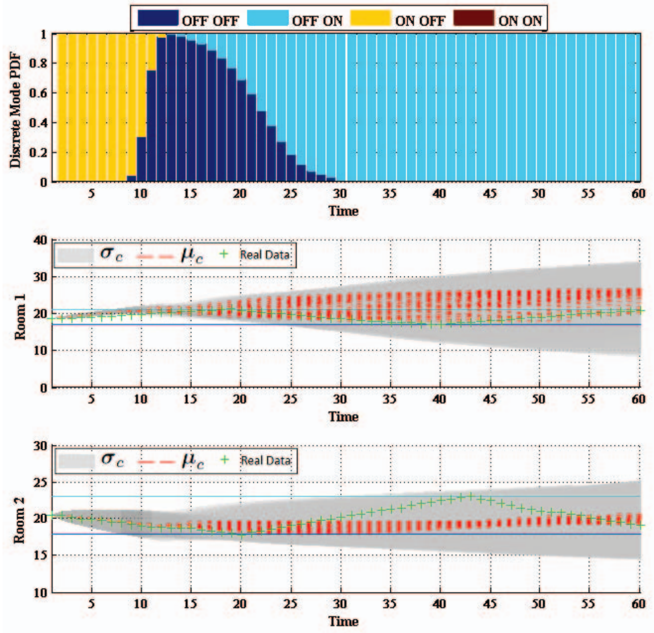

Fig. 3. Prediction distribuiot of hybrid state of the fourth hour using offline learned model

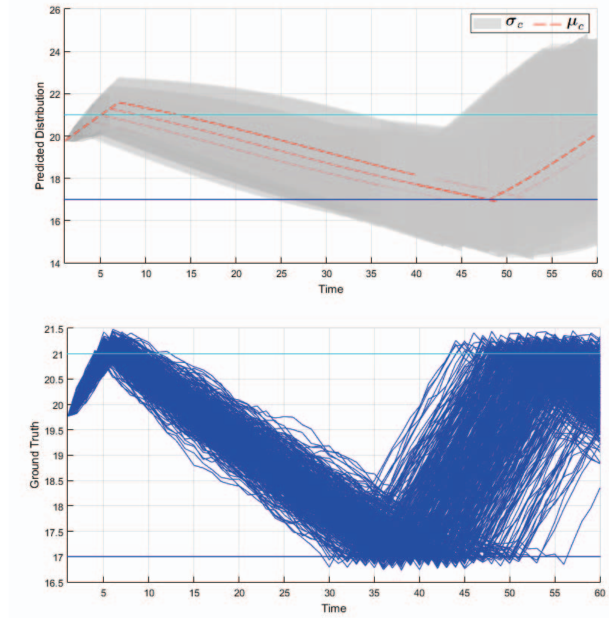

Fig. 5. Prediction Distribution of reachable state vs ground truth

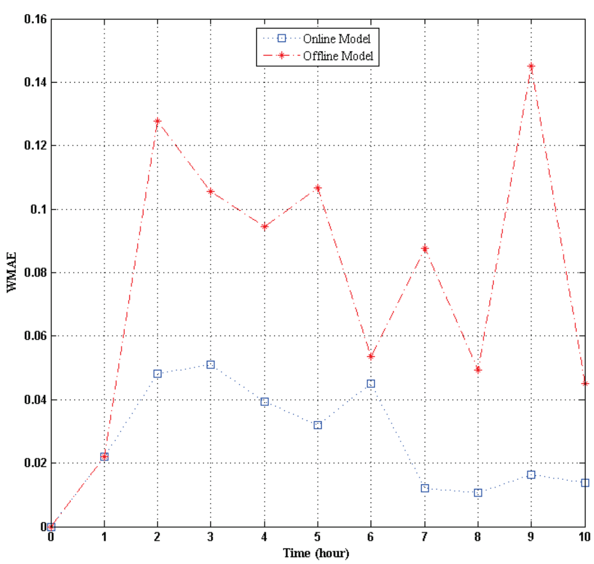

Fig. 4. WMAE Error for the 10 hour prediction for both Online and Offline learned model

the reachable set with a good accuracy and efficiency. Our future work focuses on extending the method of decision making based on optimal control.

Acknowledgment

This work is supported in part by AFOSR DDDAS FA9550-13-1-0227 and the National Science Foundation under Award (CNS-1238959).

References

[1] A. Chutinan and B. H. Krogh, "Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, F. W. Vaandrager and J. H. v. Schuppen, Eds. Springer Berlin Heidelberg, no. 1569, pp. 76–90, DOI: 10.1007/3-540-48983-5_10.

[2] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis: internal approximation," *Systems & control letters*, vol. 41, no. 3, pp. 201–211, 2000.

[3] T. Dang and O. Maler, "Reachability analysis via face lifting," in *Hybrid Systems: Computation and Control*. Springer, 1998, pp. 96–109.

[4] L. M. Bujorianu, *Stochastic reachability analysis of hybrid systems*. Springer Science & Business Media, 2012.

[5] M. L. Bujorianu, "Extended stochastic hybrid systems and their reachability problem," in *Hybrid Systems: Computation and Control*. Springer, 2004, pp. 234–249.

[6] H. J. Kushner, *Probability methods for approximations in stochastic control and for elliptic equations*. Academic Press, 1977, vol. 129.

[7] M. Prandini and J. Hu, "A stochastic approximation method for reachability computations," in *Stochastic Hybrid Systems*. Springer, 2006, pp. 107–139.

[8] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, 2008.

[9] J. Krystul and H. A. Blom, "Sequential monte carlo simulation of rare event probability in stochastic hybrid systems," in *Proceedings of the 16th IFAC World Congress*. Citeseer, 2005, pp. 4–8.

[10] D. Riley, X. Koutsoukos, and K. Riley, "Reachability analysis for stochastic hybrid systems using multilevel splitting," in *Hybrid Systems: Computation and Control*. Springer, 2009, pp. 460–464.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[12] V. Tresp, "Mixtures of gaussian processes," in *NIPS*, 2000, pp. 654–660.

[13] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[14] A. Girard, C. E. Rasmussen, J. Quinonero-Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs? application to multiple-step ahead time series forecasting," 2003.

[15] A. Fehnker and F. Ivančić, "Benchmarks for hybrid systems verification," in *Hybrid Systems: Computation and Control*. Springer, 2004, pp. 326–341.

[16] A. Rogers, S. Maleki, S. Ghosh, and J. Nicholas R, "Adaptive home heating control through gaussian process prediction and mathematical programming," in *Second International Workshop on Agent Technology for Energy Systems (ATES 2011)*, May 2011, pp. 71–78, event Dates: May 2011.