CYBER-PHYSICAL SYSTEMS
2019, VOL. 5, NO. 1, 41–64
https://doi.org/10.1080/23335777.2018.1541028

ARTICLE

Data-driven online learning and reachability analysis of stochastic hybrid systems for smart buildings

Hamzah Abdel-Aziz and Xenofon Koutsoukos

Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA

ABSTRACT

Analysis and design of cyber physical systems (CPS) relies typically on detailed dynamical models. Identifying parametric models of complex CPS such as smart buildings is very hard because of the heterogeneity and complexity of components as well as uncertainty and variability. Alternatively, data availability can potentially support the use of machine learning techniques to develop nonparametric models, which can be used for prediction, analysis, and control. In this paper, we present a data-driven methodology to learn a nonparametric stochastic hybrid system from the observed data in an online fashion. The model uses Gaussian processes and periodic Markov chains to represent the coupled continuous and discrete dynamics respectively. Moreover, we propose a reachability analysis algorithm that represents the reachable states for a receding finite horizon using mixtures of Gaussian processes. The reachability analysis algorithm provides an efficient multi-step prediction for SHS, which can be used to analyse the system's control policy, and/or its safety. Finally, we demonstrate the proposed approach to predict the thermal behaviour of smart buildings. The results show that the model can adapt to the system uncertainty and variability and predict the reachable states efficiently.

ARTICLE HISTORY
Received 15 October 2018
Accepted 19 October 2018

KEYWORDS
Stochastic hybrid systems; Gaussian processes; reachability analysis; smart buildings

1. Introduction

Identifying parametric models of complex cyber physical systems (CPS) such as smart buildings is very hard because of the heterogeneity and complexity of components as well as uncertainty and variability. On the other hand, recent technological advances enable the utilisation of sensory data to build robust and detailed models. Such models can represent the complex interactions between the cyber and the physical parts and can be used for developing methods for advanced analysis and control. Therefore, developing data-driven methodologies for learning nonparametric models of CPS is a promising alternative. Nonparametric modelling necessitates the use of machine learning

CONTACT Hamzah Abdel-Aziz hamzah.abdelaziz@vanderbilt.edu Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37235, USA

techniques because of their ability to extract information about systems and the environment from sensory data.

Smart buildings are a practical example of modern CPS where the integration of sensing and control systems is used for energy efficiency and user comfort. Buildings have complex stochastic nonlinear thermal dynamics, which depend on several factors such as the environment and the applied thermal load. Forecasting models of the environments such as the ambient temperature can be used to increase the prediction accuracy.

On the other hand, modelling the behaviour of the thermal load is very challenging especially if it cannot be measured in a cost effective manner. For example, the sensory data does not necessary provide the occupancy level in office buildings. Moreover, building parameters may change over time as the buildings age or may change abruptly due to events (e.g. opening/closing windows). Such challenges can be addressed using nonparametric modelling approaches based on online model learning.

Stochastic hybrid systems (SHS) are models that include coupled continuous and discrete dynamics and can be used to represent complex stochastic systems [1]. SHS can be used to model many CPS with multi-modal behaviours. Additionally, SHS provide an efficient modelling paradigm for complex CPS such as smart buildings. However, there are many research challenges that arise for using machine learning techniques to learn data-driven SHS models. Model learning for SHS aims to identify the continuous and the discrete dynamics from sensory data. However, the sensory data in many CPS do not necessary include explicit information about the discrete dynamics. For instance, thermal dynamics of data-centre buildings depend on the thermal load because of the utilisation of the servers and the IT equipment. However, measuring the utilisation level of the equipment explicitly may be expensive and not feasible.

Reachability analysis for SHS aims to predict the probability of the coupled continuous/discrete states to stay within a certain target region (e.g. the zone air temperature in buildings staying within the user comfort thresholds). This analysis is essential in many applications for designing the control policy. However, such prediction presents a major challenge because of the coupled stochastic discrete/continuous dynamics. Typical reachability analysis algorithms are based on Monte Carlo simulation to predict the system trajectories for a finite receding horizon [2]. However, algorithms based on Monte Carlo simulation are computationally demanding, and therefore, they may not be suitable for many modern CPS.

In our previous work, we developed online model learning framework for SHS with latent discrete state [3]. The developed framework supports one-step ahead prediction of the continuous state within the current estimated discrete state of the system. The main limitation of the learning framework is that it cannot be used to develop multi-step prediction algorithms because it does not learn the switching dynamics of the discrete state (i.e. the discrete dynamics). Furthermore, we developed a preliminary reachability analysis method based on a mixture of Gaussian processes; however, the developed approach assumes SHS with deterministic and known discrete dynamics [4]. In this paper, we propose a new learning framework which additionally learn the discrete dynamics for SHS. Such addition facilitates the development of the proposed multi-step prediction algorithm, which we use to solve the reachability analysis problem for SHS with stochastic and unknown discrete dynamics. The contributions of this paper are as follows:

(1) We present a non-parametric SHS model based on coupled Gaussian processes and periodic Markov chains. Gaussian processes capture the stochastic nonlinear continuous dynamics for different discrete states and the periodic Markov chain captures the periodic transitions of the discrete states.
(2) We propose an online clustering-based learning methodology, which can be used to learn SHS when the discrete states are not measured explicitly. In the proposed methodology, we use the K-means clustering algorithm to identify the discrete states and label the training data. Thus, we can learn the transition probabilities of the Markov chain and segment the data for each corresponding discrete state. Each segment is used to learn the continuous dynamics using a distinct GP. The proposed learning methodology is efficient and can run in an online fashion, so that the model adapts to the system variability.
(3) We propose an algorithm for finite horizon reachability analysis, which estimates the probability distribution of the reachable states based on mixtures of Gaussian processes.
(4) Finally, we demonstrate the proposed approach to learn a detailed model for smart buildings when the buildings thermal load is not measured (e.g. occupancy). We learn a model of a multi-zone office building using data generated by EnergyPlus (high-fidelity building simulator) and a stochastic occupancy simulator [5,6]. Also, we evaluate the multi-step prediction of the building's zones temperature and the thermal load behaviour.

This paper is organised as follows: Section 2 presents a brief background for model learning and prediction of Gaussian processes (GPs). Section 3 formalises the nonparametric SHS model. Section 4 discusses the model learning and the reachability analysis problem for SHS. Section 5 illustrates the proposed data-driven online learning and reachability analysis approach to approximate the reachable states for a finite horizon. Section 6 discusses the implementation and the evaluation of the proposed approach in the context of smart buildings applications. Finally, Section 7 presents a brief review of the related work.

2. Background

Gaussian processes (GPs) are nonparametric probabilistic models that utilise observed data to represent the behaviour of the underlying system [7]. GPs are identified by a mean $m(\mathbf{x})$ and a covariance $k(\mathbf{x},\mathbf{x})$ functions. The function modelled by a GP can be written as: $f(x) \sim GP(m(x), k(x,x))$. Typically, a zero mean function and squared exponential (SE) covariance function are used for their expressiveness [7].

We can learn GP models by identifying their hyperparameters Θ to best represent the training data ($\mathcal{D} = \{(\mathbf{x}_i, y_i)|i = 1, ..., n\}$). The learning process can be expressed as an optimisation problem, where the optimal hyperparameters ($\hat{\Theta}$) maximise the marginal likelihood, that is:

$$\hat{\Theta} = \arg\max_{\Theta} \log p(\mathbf{y}|\Theta, \mathcal{D})$$

Optimisation algorithms based-on conjugate gradients can be utilised to optimise the hyperparameters [7,8].

Typically, we are interested in the GP posterior distribution given some test inputs and observations (training data $\mathcal{D}$). We define the set of test inputs for which we want to predict the function value as $\mathbf{X}_*$. Hence, the posterior distribution $p(\mathbf{y}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y})$ is a conditional Gaussian distribution with a mean and a covariance defined as:

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

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

The posterior distribution shown in (1) is a prediction model for a given deterministic test input $\mathbf{X}_*$. In multi-step prediction, the $\mathbf{X}_*$ can be defined by a probability distribution (i.e. $p(\mathbf{X}_*) \sim \mathcal{N}(\boldsymbol{\mu}_*, \sum_*)$), therefore, in this case, the GP posterior distribution is obtained by:

$$p(\mathbf{y}_*) = \int\int p((\mathbf{y}_*|\mathbf{X}_*) p(\mathbf{X}_*) d\mathbf{y}_* d\mathbf{X}_*. \tag{2}$$

Equation (2) is analytically intractable [9] and several approximation algorithms have been developed to represent the posterior distribution in (2) as Gaussian (i.e. $p(\mathbf{y}_*) \sim \mathcal{N}(\boldsymbol{\mu}_y, \sum_y)$) [9,10]. In this paper, we approximate the posterior distribution in (2) by linearising the posterior GP mean function [11]. Hence, the mean and variance of the predictive distribution is obtained by:

$$\begin{aligned}\boldsymbol{\mu}_y &= \mathbb{E}[\mathbf{y}_*|\boldsymbol{\mu}_*] \\ \textstyle\sum_y &= Var[\mathbf{y}_*|\boldsymbol{\mu}_*] + \mathbf{V}\textstyle\sum_*\mathbf{V}^T\end{aligned} \tag{3}$$

where $\mathbb{E}[\mathbf{y}_*|\boldsymbol{\mu}_*]$ and $var[\mathbf{y}_*|\boldsymbol{\mu}_*]$ is the mean and the covariance of the GP posterior calculated at the mean $\boldsymbol{\mu}_*$ of the input distribution as shown in (1), and $\mathbf{V}$ is defined as:

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

3. Stochastic hybrid systems

Let's denote $\mathcal{Q}$ the set of discrete states and $\mathbb{R}^D$ the continuous state space. The system hybrid state space is defined as $\mathcal{S} = \mathcal{Q} \times \mathbb{R}^D$. The continuous dynamics evolves according to a stochastic process modelled by a GP which depends on the current discrete mode ($q \in \mathcal{Q}$). The discrete state also evolves based on a stochastic process $\delta : \mathcal{Q} \times \mathcal{Q} \to [0, 1]$ represented by a periodic Markov chain (MC). Periodic MCs can represent periodic system behaviour (e.g. based on hour-of-day, day-of-week, or seasonal effects in buildings applications). In this paper, we consider systems with two inputs: Control inputs and external uncontrolled inputs (disturbances) from the environment. The control inputs 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 system dynamics and are modelled using a time-series model ($E : \mathbb{N} \to \mathcal{V}$).

The SHS model is formalised as follows:

Definition 3.1. (Non-parametric SHS). A nonparametric SHS model is defined as a tuple $\mathcal{H} = (\mathcal{Q}, X, Init, \mathcal{U}, \mathcal{V}, \mathrm{A}, \delta)$:

- $\mathcal{Q} := \{q_1, q_2, \cdots, q_m\}$, 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 $\mathcal{S} := \mathcal{Q} \times \mathbb{R}^D$.
- $\mathcal{U} \subset \mathbb{R}^E$, 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.
- $\mathcal{A}$ assigns to each discrete state $q \in \mathcal{Q}$ a function $(x_{k+1} = f_q(x_k, u_k, v_k))$ modelled by a GP which represents the evolution of the continuous state given the predecessor continuous state $x_k \in \mathbb{R}^D$, a control input $u_k \in \mathcal{U}$ and an external uncontrolled input $v_k \in \mathcal{V}$.

- δ is a stochastic process of the discrete state $\{q_k : k \geq 0, q_k \in \mathcal{Q}\}$ represented by a periodic MC such that $p(q_k|q_{k-1}, q_{k-2}, \cdots q_0) = p(q_k = i|q_{k-1} = j) = p_{ij}(k), \forall i, j \in \mathcal{Q}$.

For a finite time horizon $[0, N]$, an execution of $\mathcal{H}$ is a trajectory denoted by $\{s(k) = (q(k), x(k)), k \in [0, N]\}$, with a control policy $\pi(\mathcal{S})$ and a time-series disturbance model $E(k)$. A trajectory can be easily obtained by simulating the model (i.e. calculate the control input, forecast the external input, and evaluate the continuous/discrete states) for the required horizon.

In this paper, we use $\mathcal{H}$ to model the thermal behaviour of smart buildings when the thermal load is not measured. In the context of smart buildings, the continuous state ($\mathbf{x} \in \mathbb{R}^D$) represents the building's zones air temperatures, the discrete state q represents the thermal load level, the uncontrolled input ($v \in \mathbb{R}$) represents the ambient temperature, the control input ($u \in \mathbb{R}$) represents the HVAC cooling/heating rate, and δ models the periodic behaviour of the thermal load levels. Therefore, the predictive distribution of the building's zone air temperature can be represented as

$$\mathbf{x}_{k+1} \sim \sum_i p(q_k = i|q_{k-1}) f_i(\mathbf{x}_k, u_k, v_k)$$

4. Problem formulation

Online learning and reachability analysis necessitate updating the system model $\mathcal{H}$ and predicting the system states iteratively and efficiently after receiving new measurements.

Learning the SHS model $\mathcal{H}$ requires identifying the discrete dynamics (i.e. the discrete state space $\mathcal{Q} := \{q_1, q_2, \cdots, q_m\}$ and the discrete transition function δ), and the continuous dynamics (i.e. $GP(.)$ for all $q \in \mathcal{Q}$) from the observed data ($\mathcal{D}$) collected from the physical system and the environment. At each time step k, the observed data include measurements of the continuous state $\mathbf{x}_k$, the control inputs u_k, and the external disturbances v_k. Thus, the training dataset is defined as:

$$\mathcal{D} := \{(\hat{\mathbf{x}}_k, \mathbf{y}_k) : k = T_s, \cdots, T_e\}$$

where $\mathbf{y}_k$ is the successor continuous state (i.e. $\mathbf{y}_k = \mathbf{x}_{k+1}$), $\hat{\mathbf{x}}_k$ is the tuple $(\mathbf{x}_k, u_k, v_k)$, and $[T_s, T_e]$ is the time period at which the data are collected. In general, learning $\mathcal{H}$ using $\mathcal{D}$ is a challenging task because the sensory data do not necessary include explicit measurements of the discrete state (e.g. thermal load). Moreover, the model must capture the uncertainty and the variability of the system and the environmental disturbances.

Reachability analysis aims at predicting the probabilities of the reachable states for a finite time horizon given an initial state $s(0)$. This multi-step prediction is usefully in many applications to analyse the system design. For instance, prediction the zone air temperature in buildings is important to ensure the user comfort for a given control policy. Therefore, our objective is to calculate $P(s(k)|s(0)), \forall k \in [1, T]$ for the finite receding horizon $[1, T]$, given the SHS model $\mathcal{H}$, the control policy $\pi(\mathcal{S})$ and the time-series disturbance model $E(k)$. Prediction of reachable states can be performed as an iterative process since $s(k)$ depends on $s(k-1)$. However, this is a challenging task because: (1) Predicting the continuous state $x(k)$ distribution requires prediction at an uncertain input because $x(k)$ depends on the distribution of $x(k-1)$ and it also evolves differently for each discrete state $q(k)$; (2) the discrete mode $q(k)$ is a discrete random distribution given the probability distribution of $q(k-1)$; and (3) the system trajectory depends on the switching times between the discrete states.

5. Online learning and reachability analysis

To overcome these challenges, we propose a novel online learning and reachability analysis framework depicted in Figure 1 . The proposed framework comprises the following steps: (1) Collect data from the system and update the training data, (2) learn both the continuous and the discrete dynamics of the SHS model $\mathcal{H}$, as well as the time-series disturbance model $E(k)$; and (3) solve the reachability analysis problem for a finite horizon based on Mixture of Gaussian processes. These steps are repeated iteratively in an online fashion to adapt to the system variability.

Initially, we collect training dataset ($\mathcal{D}$) which consists of the continuous state, the control input, and the external disturbance. This dataset is used to initialise the system model. As new measurements are collected, we update the training dataset using a moving window technique based on first-in-first-out (FIFO) policy. The FIFO policy maintains a fixed size of the training dataset that is then used to learn the system model as described in the next section.

5.1. Model learning

We first identify the discrete modes of the SHS model using a clustering algorithm. Then, we segment and label the training data based on the identified modes. This allows us to learn the continuous dynamics for each discrete state using a distinct GP model and learn the discrete dynamics using a periodic MC model.

5.1.1. Feature extraction

Feature extraction is a technique used to transform the training data into a set of features. A set of features contains the useful data needed for the clustering

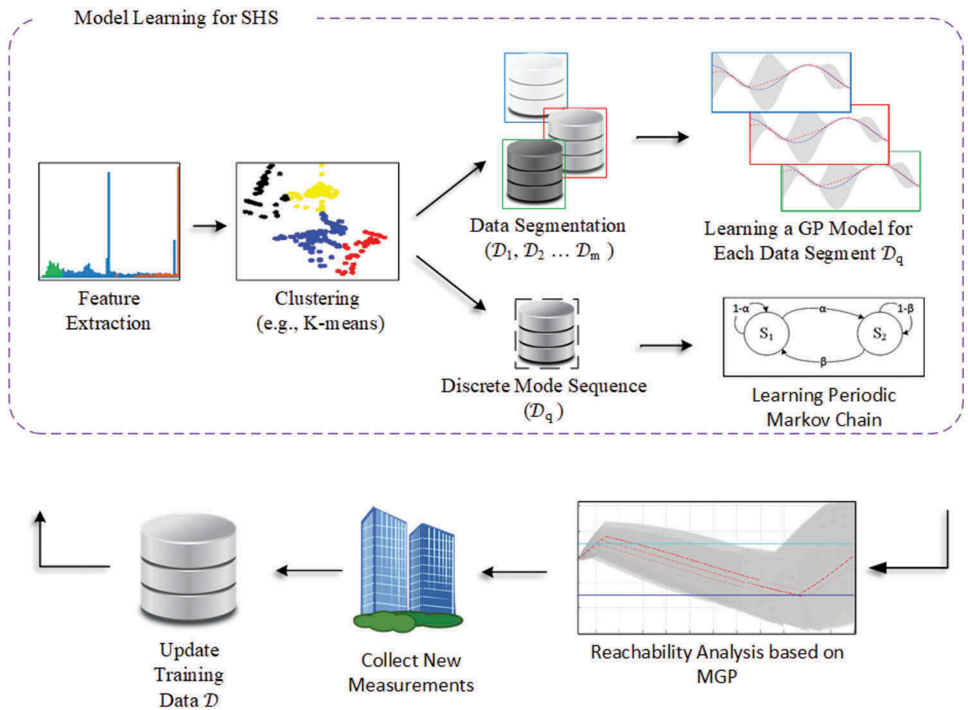

Figure 1. Block diagram of the proposed approach.

stage. Generally, the feature vector can be computed based on time domain features or frequency domain analysis. In our work, we compute the feature vector based on time-domain features (e.g. mean, root-mean-square), such that:

$$\mathbf{z} = g(\mathbf{x})$$

where $\mathbf{x} \in \mathbb{R}^c$ is a vector of the raw data with dimension c, $\mathbf{z} \in \mathbb{R}^d$ is the feature vector with dimension d, and $g(.)$ is a function that calculate the time-domain features. For thermal models of buildings, the time-domain features are the average cooling rate (i.e. $u(k)$) and the zone air temperature difference (i.e. $\Delta x = x_k - x_{k-1}$).

5.1.2. Data clustering for discrete mode identification

The goal of data clustering is to associate each data point with a discrete state. Various clustering algorithms can be used such as K-means, Gaussian Mixture Model (GMM), or hierarchical clustering [12,13]. The choice of the algorithm depends on the application and the collected data. We use the K-means clustering algorithm which calculates the cluster centroids ($\hat{\mathcal{C}}$) for K clusters, so that $\hat{\mathcal{C}}$ minimises the following potential function:

$$\hat{\mathcal{C}} \quad = \arg\min_{\mathcal{C}} \sum_{g(x) \in \chi} \min_{c \in \mathcal{C}} \parallel g(\hat{\mathbf{x}}) - c \|^2 \tag{4}$$

where $\chi \in \mathbb{R}^{n \times d}$ is the feature matrix extracted from the training data ($\mathcal{D}$) with size n and feature dimension d, $g(\hat{\mathbf{x}}) \in \mathbb{R}^d$ is the feature vector for the data point $\hat{\mathbf{x}} \in \mathcal{D}$, and $\hat{\mathcal{C}} \in \mathbb{R}^{K \times d}$ represents the K cluster centroids.

The approach requires the number of clusters (i.e. the number of discrete states $m \in \mathbb{N}$) to be known a priori. We identify the number of discrete states m using a heuristic algorithm known as Silhouette analysis method [13]. Silhouette analysis determines the number of clusters ($\hat{K}$) which results in the best clustering consistency. The clustering consistency for a given K is represented by a Silhouette scoring coefficient for each clustered data point. The Silhouette scoring coefficient has a range of $[-1, 1]$ where scores near $+1$ are assigned to data points that lie far from the neighbouring clusters. On the other hand, scores near zero are assigned to data points that lie very close to the boundary between their cluster and a neighbouring one. Negative scores are assigned to data points, which might have been allocated to the wrong cluster. Therefore, clusters with a higher average Silhouette score have a better consistency than clusters with a lower average Silhouette score. Formally, the Silhouette score $s(i)$ for a given data point i can be obtained using the following formula:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{5}$$

where $a(i)$ is the average distance from the data point i and the other points in its cluster, and $b(i)$ is the minimum, minimised over clusters, average distance between the data point i and other points in a different cluster. Silhouette analysis maximises the average Silhouette score:

$$m = \hat{K} \quad = \arg\max_{K} \bar{s}(i), \; K \in G \tag{6}$$

where G is a finite set of potential value of m.

5.1.3. *Learning discrete dynamics*

The discrete dynamics (i.e. $\delta(.)$) represents the stochastic transitions between the discrete modes. We represent $\delta(.)$ using a periodic MC and we consider SHS whose discrete state transitions are independent of the control signals and the continuous state. A typical MC has a stationary matrix that does not capture any periodic or time-dependent behaviours explicitly. However, the discrete dynamics of SHS models for buildings exhibit periodic behaviour (e.g. occupancy patterns depend on the time of the day). Therefore, we represent the discrete dynamics using a periodic MC with non-stationary transition probabilities. The transition matrix is used to calculate the probability of the discrete modes as follows:

$$p(q_{k+1}) = \delta(q_k) = p(q_k)A_{h(k)}$$

where $h(k) : k \to \{1, 2, \cdots, H\}$ maps the time-step k to the time of the day (e.g. hour of the day $\{1, 2, \cdots, 24\}$), and $A_{h(k)}$ is the associated transition matrix. A graphical representation of the periodic MC with 24 transition matrices (i.e. $H = 24$) and two states is depicted in Figure 2.

We learn the periodic MC by identifying the parameters of the transition matrices (i.e. transition probabilities). We use the sequence of labels of the training dataset $\mathcal{D}_q$ from the previous step (clustering):

$$\mathcal{D}_q := \{q_k : q_k = \arg\min_{q' \in \mathcal{Q}} \| g(\hat{\mathbf{x}}_k) - c_{q'} \|^2, \\ \forall (\hat{\mathbf{x}}, \mathbf{y})_i \in \mathcal{D}, k = \{0, 1, \cdots, M\}\}$$

where $c_{q'}$ is the cluster centroid of the discrete mode q', $g(\hat{x})$ is the feature vector of the data point x and M is the size of the training dataset. $\mathcal{D}_q$ is used

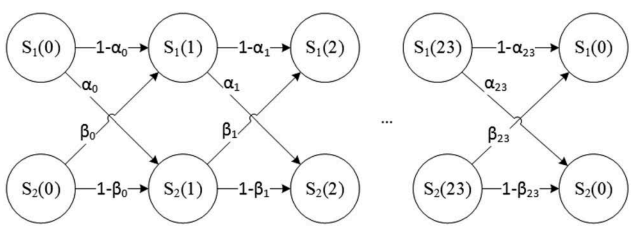

Figure 2. State-transitions diagram of a two-state periodic MC model.

to learn/update the model parameters (i.e. the transition probabilities) for each transition matrix $A_i, i \in \{1, 2, \cdots H\}$. Each transition matrix A_i is identified independently by counting all the distinct sequences $(q_k, q_{k+1}) \in \mathcal{D}_q$ such that $h(k) = i$. So, its transition probabilities can be computed by:

$$a_{ab} = \frac{\text{total number of } q(a)\, q(b) \text{ occurrence}}{\text{total number of } q(a) \text{ occurrence}} \tag{7}$$

5.1.4. Data segmentation and learning GP models

Data clustering enables us to identify the discrete states ($\{q_1, q_2, \cdots, q_m\}$) and to label each data point in $\mathcal{D}$ with the corresponding discrete state. The labelled data are used to segment the training data into m datasets:

$$\forall j \in \mathcal{Q}, \mathcal{D}_j := \{(\hat{\mathbf{x}}, \mathbf{y})_i : j = \arg\min_{q' \in \mathcal{Q}} \| g(\hat{\mathbf{x}}_k) - c_{q'} \|^2, \forall (\hat{\mathbf{x}}, \mathbf{y})_i \in \mathcal{D}\}$$

where $c_{q'}$ is the cluster centroid of the discrete mode q' and $g(\hat{\mathbf{x}})$ is the feature vector of the data point $\hat{\mathbf{x}}$. We use each data segment $\mathcal{D}_j$ to learn a distinct GP model for the continuous dynamics. As discussed in Section 2, learning a GP model is an optimisation process that calculates the optimal hyperparameters $\hat{\Theta}_j$ of the GP in order to maximise the log likelihood function:

$$\hat{\Theta}_j = \arg\max_{\Theta_j} \log p(\mathbf{y}|\Theta_j, \mathcal{D}_j) \tag{8}$$

Finally, we model the uncontrolled external input v (i.e. ambient temperature) using a time-series model $E(k)$. We learn this model using a single Gaussian processes model independently from the SHS model-learning algorithm. Formally, the time-series model $E(k)$ of an observed time-dependent variable v_k is defined as: $v_k = f(k) \sim GO(m(k), K(k, k'))$.

5.2. Reachability analysis

We represent the reachable states of $\mathcal{H}$ using mixtures of Gaussian processes (MGP) [14]. 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 used to calculate the system output at a given input. The MGP model is expressed as:

$$P(y|x) = \sum_{i=1}^{z} P(z = i\,|\,x) GP_i(m_i(x), k_i(x, x)) \tag{9}$$

where y is the output, x is the input, z is the discrete latent variable with Z states and GP_i is the GP function corresponding to the discrete state $z = i$. Our goal is to predict the probability of the continuous state $\mathbf{x}(k+1)$ given the probability of the hybrid state $s(k) = (\mathbf{x}(k), q(k))$. Thus, the one-step state prediction of the SHS can be defined as:

$$P(\mathbf{x}(k+1)\Big|s(k)) = \sum_{i=1}^{m} P(q(k+1)=i\Big|q(k))f_i(\hat{\mathbf{x}}(k)) \tag{10}$$

where $f_i(\hat{\mathbf{x}}(K)) \sim GP_i(m_i(\hat{\mathbf{x}}), \mathrm{K_i}(\hat{\mathbf{x}}, \hat{\mathbf{x}}))$ with $\hat{\mathbf{x}}$ defined as the tuple $(\mathbf{x}, u, v)$ and $P(q(k+1)=i|q(k)) \sim \delta(q(k))$ is the probability distribution of the discrete state at time-step $k+1$. In order to predict the probability distribution of $s(k), \forall k \in [1, T]$ for the finite-horizon T, we can apply (10) iteratively. However, this equation depends on $p(\mathbf{x}(k))$ which is represented by a Gaussian mixture model from the previous iteration (i.e. $p(\mathbf{x}(k)|s(k-1))$). Formally, let's define $p(\mathbf{x}(k))$ as:

$$p(\mathbf{x}(k)) = \sum_{j=1}^{C} w_j \mathcal{N}(\boldsymbol{\mu}_j, \textstyle\sum_j) \tag{11}$$

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 $\boldsymbol{\mu}_i, \sum_i$ is the mean and the variance of i^{th} Gaussian component respectively. Calculating $P(x(k+1)|s(k))$ iteratively from Equations (10) and (11) is analytically intractable because the input of the MGP model in (10) is uncertain (represented by a mixture of Gaussian probability distributions as illustrated in Equation (11)). To overcome this limitation, we approximate the predictive distribution by propagating every Gaussian component in (11) independently. Therefore, the predictive distribution can be obtained as:

$$P(\mathbf{x}(k+1)) = \sum_{i=1}^{Q} \sum_{j=1}^{C} w_j \quad P(q(k+1)=i|q_j(k))\tilde{f}_i(\mathbf{x}_j(k), u_j(k), v(k)) \tag{12}$$

where $\mathbf{x}_j$ is the j^{th} Gaussian component of $p(x(k))$ with weight w_j, mean $\boldsymbol{\mu}_j$ and variance $\sum_j$, $q_j(k)$ is the discrete mode of the j^{th} Gaussian component and $\tilde{f}(.)$ is the approximation of GP posterior f_i defined in Equation (3). Algorithm 1 illustrates the prediction of the reachable states of the SHS iteratively based on Equation (12).

Algorithm 1 Discrete-time SHS state prediction

Input: $s(0)$, T
Output: $p(s(k))$ for $k \in [1, T]$
Load GP function $f_{q(0)} \sim GP_{q(0)}$
k = 0
while $k < T$ **do**
▹ Forecast the external input at time k
$v(k) \leftarrow E(k)$

for each $s_c(k) \in s(k)$ **do**
  for each $q \in \mathcal{Q}$ **do**
    ▹ Calculate the probability of the discrete state q
    $p(q_c(k+1) = q|q_c(k)) \leftarrow \delta(q_c(k))$
    ▹ Calculate the new weight
    $w_c(k+1) \leftarrow p(q(k+1) = q_c(k+1)|q_c(k)) \times w_c(k)$
    if $w_c(k+1) > \delta_w$ **then** ▹ ignore components with small probabilities
      ▹ Calculate the control input
      $u_c(k) \leftarrow \pi(x_c(k), q_c(k+1))$
      ▹ Predict the continuous state
      $x_c(k+1) \leftarrow \tilde{f}_{q_c(k+1)}(x_c(k), u_c(k), v(k))$
      $s_c(k+1) \leftarrow [x_c(k+1), w_c(k+1), q_c(k+1)]$
      add $s_c(k+1)$ to $s(k+1)$
    end if
  end for
end for
end while

The prediction algorithm approximates the probability distribution of the discrete state by ignoring the discrete transitions which have probabilities less than δ_w. Moreover, it approximates the prediction of the continuous state by linearising the posterior GP mean function. The accuracy of the prediction algorithm can be increased by tuning the threshold δ_w and/or by approximating the GP posterior using the exact moments [11]. The prediction algorithm is efficient and can run in an online fashion. The most expensive part is computing the inverse covariance matrix which requires $\mathcal{O}(n^3)$ time where n is the size of the data. Many studies have been conducted to improve GP complexity using different approximation algorithms. For instance, sparse Gaussian processes are developed to approximate the inverse of the covariance matrix K with a low rank matrix approximation of dimension $m \times m$, (where $m << n$) [15].

6. Evaluation

In this section, we demonstrate the efficacy of the proposed method on multi-zone buildings. We have implemented the approach using MATLAB® and the Statistics and Machine Learning Toolbox Release 2017a [16]. We evaluate the approach for an office building with five zones. The physical system is simulated using EnergyPlus [5]. EnergyPlus is an open-source cross-platform building energy simulator funded by the U.S. Department of Energy, and Building Technologies

Office, and managed by the National Renewable Energy Laboratory. EnergyPlus is used for high-fidelity simulation of buildings using (1) the ambient temperature and the environment data and (2) the building description. The building description defines its structure and layout, the construction materials, the thermal zones with their dimensions and area, the HVAC system, the control strategies and so on. It also defines the building thermal loads with their schedules such as occupancy, lights, and electrical equipment. These detailed descriptions are used to construct several models (e.g. airflow network model, pollution model, on-site power model) based on which EnergyPlus simulates the thermal behaviour. Although such models can be used for high-fidelity simulation, multi-step prediction is only possible using Monte Carlo techniques.

We generate a dataset for a single-story office building with five zones. We simulate the office building based on a realistic occupant schedule for office spaces to evaluate the ability of our model to capture the stochastic discrete dynamics. The occupancy schedule is generated using the occupancy simulator developed at Lawrence Berkeley National Laboratory [6,17]. This stochastic occupancy schedule is then used by EnergyPlus to simulate the thermal behaviour of the office building and generate data for training. The major thermal sources for all the five zones are the HVAC unit heating and cooling supply air, the office lights and equipment, and the office occupancy. The dataset measures the building thermal behaviour hourly for one year using weather data from San Francisco, CA. The measurements consist of the ambient temperature, zone air temperatures, cooling/heating rate from the HVAC unit, and heating rate from the thermal load (lights, occupancy, and office equipment) aggregated and averaged for every hour.

We set the window size of the training data to four Weeks (i.e. 672 data points). The goal is to predict the system behaviour for the next day (i.e. horizon $T = 24$). Initially, we train the model using the first four weeks of the simulation data, then apply the proposed online approach to predict the system behaviour for the next day. We collect new data for the predicted day and update the training dataset to re-learn/update the model and we repeat these steps.

For learning the SHS model, we identify the model discrete states using the K-mean clustering algorithm. Data clustering starts by extracting the time-domain features which are the average heating/cooling rate (i.e. $u(k)$) and the zone air temperature difference (i.e. $\Delta x = x_k - x_{k-1}$). The number of the discrete states is estimated using the Silhouette analysis method such that, the number of the discrete states for the core-zone, south-zone and east-zone is three, for the north-zone is two, and for the west-zone is five. Furthermore, we clustered the training dataset to label each data point with its corresponding discrete mode and used the labels sequence to learn the periodic MC for the discrete dynamics. Finally, we segment the data for each discrete mode and use them to learn distinct GP models for each mode. For reachability analysis, we use the proposed algorithm to generate a distribution of the reachable states. The prediction distribution for both the discrete mode (i.e. estimated thermal load level) and the continuous state (i.e. zone air temperature) of three days are shown in Figure 3 for the west zone.

6.1. Performance

We evaluate the performance of the reachability analysis using weighted root mean square error (RMSE) and mean relative square error (MRSE) error metrics, defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N} e(k)^2}, \qquad MRSE = \sqrt{\frac{\sum_{k=1}^{N} e(k)^2}{\sum_{k=1}^{N} y(k)^2}} \tag{13}$$

where $y(k)$ is the system output and $e(k) = \sum_c w_c(k)(\hat{y}_c(k) - y(k))$ is the weighted prediction error of the i^{th} time step, evaluated for each Gaussian components using the component mean $\hat{y}_c$ and the component weight w_c. Moreover, we evaluate the predicted distribution of the reachable states using negative log predictive density (LD), defined as:

$$LD = \sum_{k=1:T} -\log p(y(k)) \tag{14}$$

where $p(y(k))$ is the prediction density function (i.e. mixture of Gaussian) of the system state, and T is the prediction horizon (i.e. 24).

GP models have been proposed in the literature to provide thermal models for buildings [18]. We compare the prediction of our SHS approach against the prediction obtained using two typical GP models: a unimodal single GP model and a full GP model. The unimodal GP model does not have discrete states associated with the thermal load level and it is defined by: $\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k, v_k)$ where $\mathbf{x}_k$ is the zone air temperature, u_k is the heating/cooling rate from the HVAC unit, and v_k is the ambient temperature. The full GP model assumes that thermal load is measured and it is defined by: $\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k, l_k, v_k)$ where l_k is the thermal load.

Reachability analysis using the unimodal model is a typical multi-step prediction for a single GP model as illustrated in section (2). The same approach can also be used for the full GP model, however additional time-series model should be learned to predict the thermal load for the finite-receding horizon. In this experiment, we use a time-series GP such that: $l_k \sim GP(m(k), K(k, k'))$. Figures 4 and 5 show one day prediction of the west-zone using the unimodal GP and the full GP model; respectively.

The performance statistics for the SHS model compared against the unimodal GP and the full GP models are shown in Table 1. These results indicate

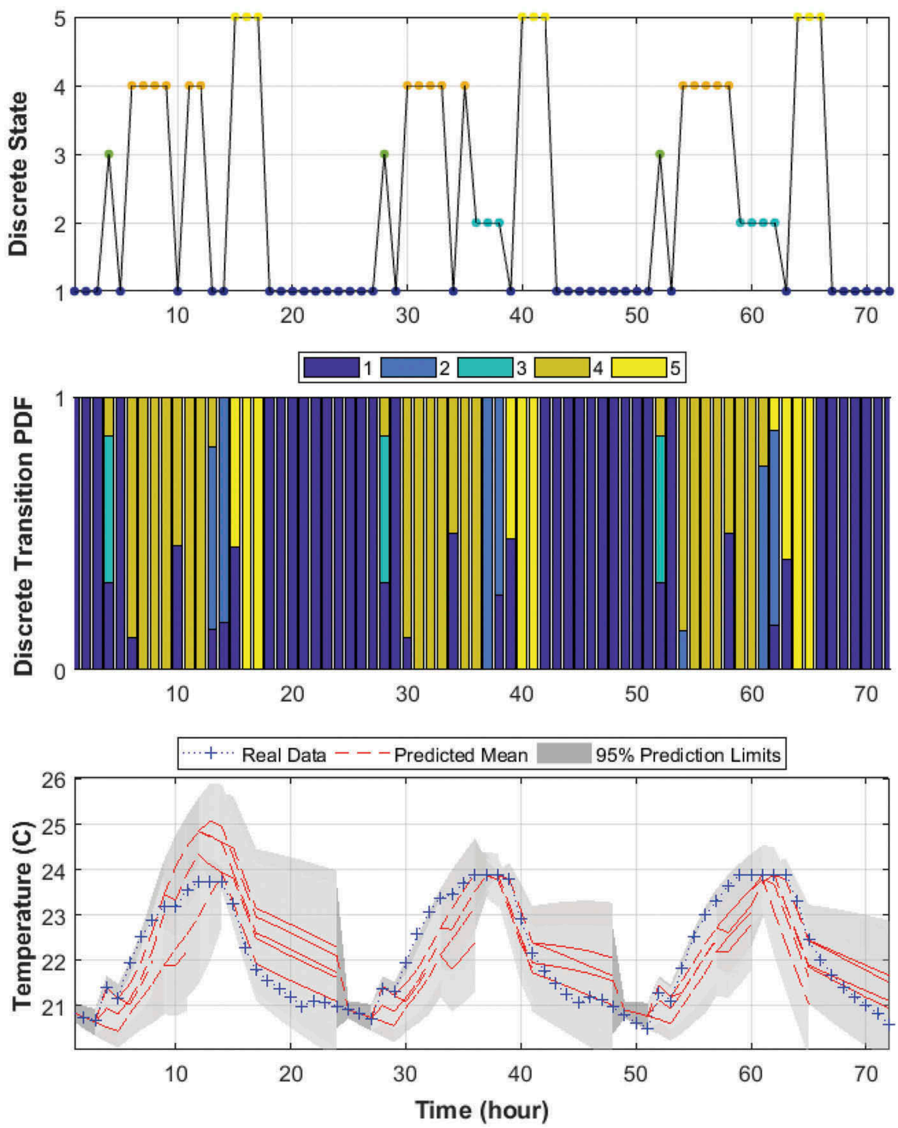

Figure 3. Prediction distribution of the hybrid states for the west-zone using the proposed approach.

that the developed approach outperforms both typical GP models since it takes into consideration the prediction of the thermal load level and its periodic patterns. In comparison to the SHS prediction, the unimodal GP lacks the prediction of the thermal load pattern, and therefore, the prediction distribution averages over the thermal load levels with high uncertainty, measured by the predictive density (LD) metric as shown in Table 2. The full GP model has a better performance than the unimodal GP since it considers the applied thermal load, assuming that we can measure it. However, the

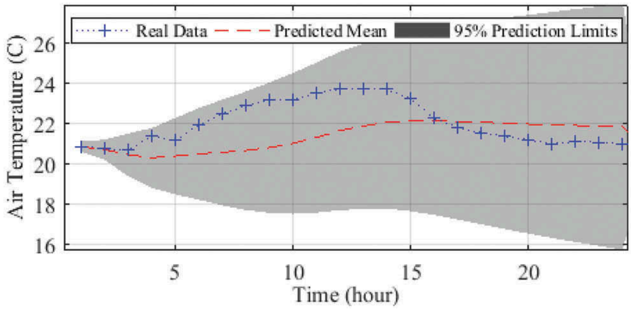

Figure 4. Prediction distribution of the west-zone air temperature using a unimodal GP model.

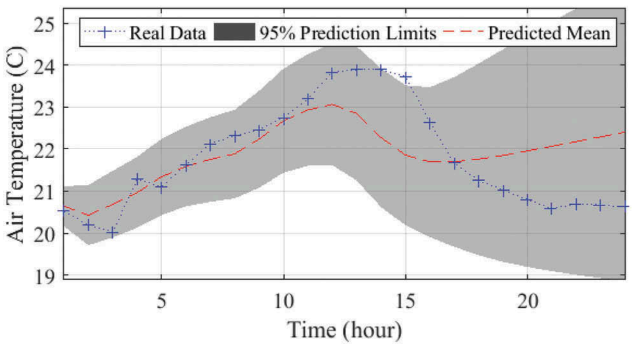

Figure 5. Prediction distribution of the west-zone air temperature using the full GP model.

Table 1. Performance statistics of the reachability analysis prediction with a comparison against the full GP and the unimodal GP models.

	RMSE (MRSE)		
	Full GP	SHS	Unimodal GP
Core zone	1.10 (0.05)	0.92 (0.04)	2.28 (0.10)
South zone	1.93 (0.09)	1.09 (0.07)	2.35 (0.10)
East zone	4.08 (0.18)	1.17 (0.05)	3.82 (0.16)
North zone	1.46 (0.07)	1.02 (0.04)	3.00 (0.13)
West zone	1.95 (0.09)	1.28 (0.05)	2.15 (0.09)

predictive density of the full GP is lower than the proposed SHS model since the uncertainties in the predicted thermal load increase the uncertainty in the prediction of the zone air temperature.

Further, we compare the proposed online-learned model against an offline model, which is learned once, in order to evaluate the improvement of the online learning approach. In this experiment, we consider changes in the ambient temperature due to seasons as an example to represent the variability in the system. One year of weather data from San Francisco, CA is used to represent real ambient

Table 2. Average negative log predictive density of the proposed approach, the unimodal GP, and the full GP.

	SHS	Unimodal GP	Full GP
Core zone	44.6	78.6	78.5
South zone	41.9	51.5	52.2
East zone	37.6	63.2	48.2
North zone	36.5	66.1	51.4
West zone	53.3	48.6	43.7

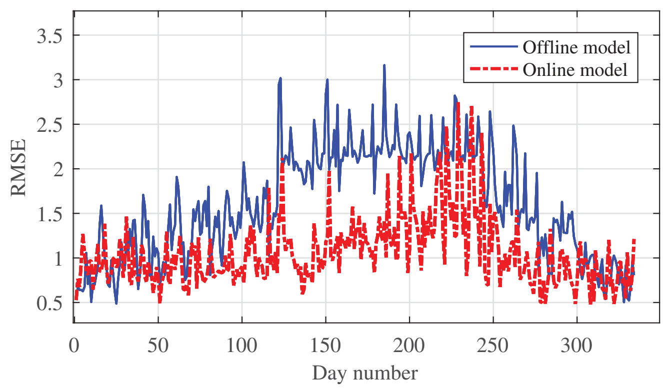

Figure 6. Error metric (RMSE) of the air temperature prediction for both the online and the offline learned SHS model, averaged over all zones.

temperature. Additionally, we use a stochastic occupancy schedule to add uncertainty in the system behaviour. Figure 6 shows the RMSE statistics for both models calculated for each day of the year. The results indicate that online learning allows our model to adapt to the variation in the system with a good performance. On the other hand, the offline model fails to adapt to these variations.

6.2. *Efficiency*

Despite the computation demand of machine learning algorithms, specifically GPs, the proposed methodology is computationally efficient and can run in an online fashion with acceptable performance. The most expensive part is computing the inverse covariance matrix of GPs which requires $\mathcal{O}(n^3)$ time where n is the size of the training data for each GP model. GP learning becomes more computationally expensive when the dimension of the model and/or the training dataset increases. The computation time because of the GP is evaluated using different experiments of the proposed approach with different sizes of the training dataset. Figure 7 shows the variation of the model learning and the reachability analysis average running time for five-zone and two-zone buildings with different dataset sizes. As indicated from these results, the running time becomes a major factor in the GP

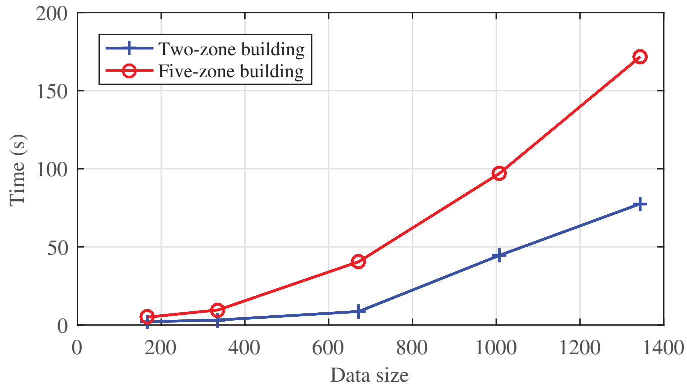

Figure 7. Average running time of the proposed framework using different sizes of the training dataset sizes for two buildings examples.

performance as we increase the training dataset size. However, the performance of the model prediction is still adequate for the relatively small dataset sizes since we segment this data and distribute the computation for each discrete mode independently. On the other hand, the single GP models (i.e. unimodal GP and Full GP) lacks the advanced of segmenting the dataset, and therefore, they are more computationally expensive as shown in Figure 8.

Furthermore, we evaluate the proposed methodology performance for different sampling periods of 15, 30, 45, and 60 min. For all cases, we fixed the duration of the window size for the training data (2 weeks). However, the dataset sizes increase as the sampling periods decreases. In addition, the prediction horizon is fixed as one day ahead for all sampling periods and therefore the number of time-steps increase as we decrease the sampling periods. Figure 9 shows the running times of the model learning and the reachability analysis, respectively,

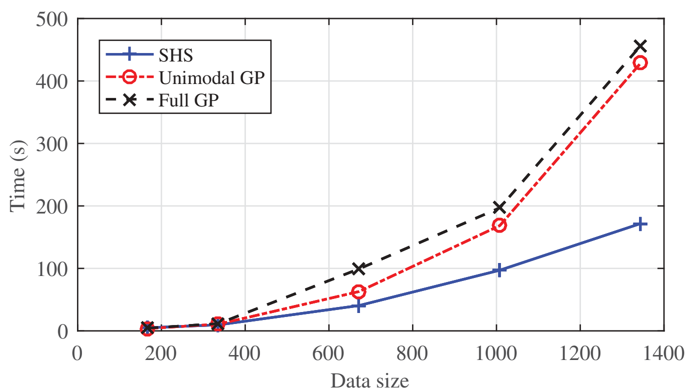

Figure 8. Average running time of different training dataset sizes for the five-zone office building.

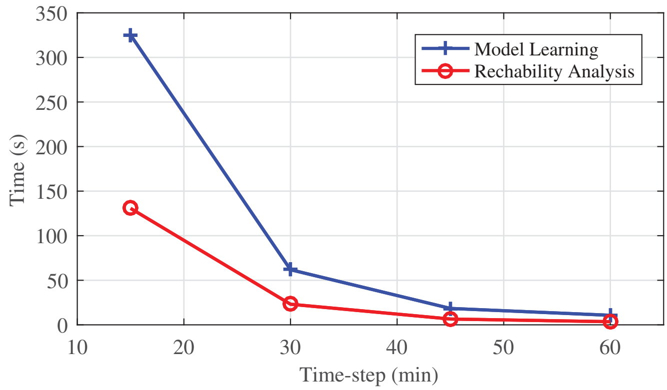

Figure 9. Average running times using different time-step size and training dataset sizes for model learning and reachability analysis.

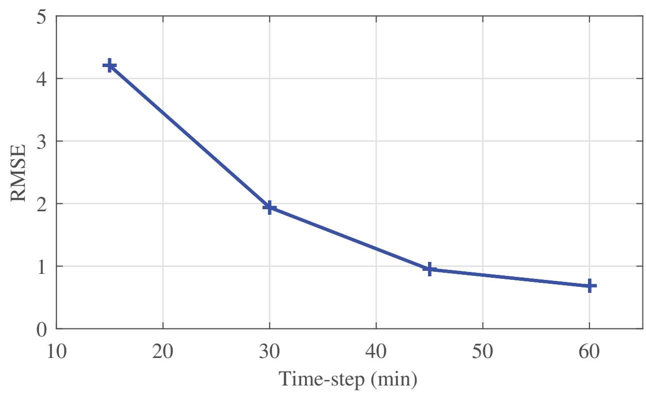

Figure 10. Error metric (RMSE) of the predicted zone-air temperature with various time-step size, the RMSE metric is averaged over all zones for the five-zone office building.

using different sampling periods. The results show an exponential increase of the running time as the sampling period is decreased. The proposed approach runs efficiently with an accepted performance for sampling periods between 30 to 60 min. The RMSE, averaged over all zones, performance metrics depicted in Figure 10 show that, the reachability analysis performance increases as we increase the sample rate. This is expected since the prediction time-steps increases as we decrease the sampling rate, and therefore, the prediction uncertainty increases. We conducted the above experiments in Intel core i5 PC with 8 GB memory. The results indicate that the proposed approach is applicable for smart buildings applications in real time.

7. Related work

GPs have shown a great success in learning many nonlinear and stochastic systems because of its attractive features [19]. Recently, many studies used GPs to learn time-series models for short-term and multi-step forecasting [9,20]. Additionally, GPs shown a great success in developing data-efficient learning framework for systems represented by state-space model with control inputs [11,21].

Multi-step prediction for stochastic system modelled by GPs is a challenging task because it requires propagating the uncertainty of the predicted state at each step. Monte-Carlo simulation is used to predict the system trajectory and represent the predicted trajectories by samples (particles) [22]. In spite of that, Monte Carlo simulation is limited to particle-based control and analysis methodologies. Moreover, it can be computationally demanding, especially when a large amount of samples are used to obtain good accuracy. To overcome these limitations, other methods are developed to approximate the predictive distribution analytically. For instance, methods based on moment-matching and linearisation of the predictive distribution approximate the predictive distribution as Gaussian distribution [11,23]. To gain more approximation accuracy, authors in [24] propose a multi-step prediction algorithm which uses the kurtosis metric to measure the non-Gaussianity of the predictive distribution, and based on these measurement, the algorithm adaptively split the input distribution into a sum of Gaussian and approximate the predictive (output) distribution by a Gaussian mixture model.

GPs have been used in many recent studies to learn and to predict stochastic nonlinear systems. However, they typically do not consider systems with coupled discrete/continuous dynamics as the case in SHS. Learning SHS is more challenging because of the uncertainty in the model behaviour along with the coupled continuous/discrete dynamics. Simulation-based learning methods use simulated trajectories for parameters identification based on randomised optimisation techniques (e.g. Genetic Algorithm) [25]. Expectation-maximisation(EM) algorithm is also used to identify the parameters of SHS [26]. Additionally, a kernel-based approach is developed for a popular class of hybrid systems, known as piecewise affine systems, where GPs are used to model the impulse response of each sub-model of the model [27].

Typically, reachability analysis is a problem in SHS, which necessitates to predict the reachable states for a finite-receding horizon. Several methods have been proposed in the literature to estimate the reachable states for SHS [2]. Analytical estimations methods have been used to solve the reachability problem for SHS via quadratic forms known as Dirichlet forms [28]. Other approximation methods based on numerical estimations are also used such as Markov Chain approximations [29,30], and dynamic programming [31]. Probabilistic methods have been considered to solve the reachability problem based on randomising algorithms such as Monte-Carlo methods [32] and multilevel splitting (MLS) variance reduction [33]. Finally, statistical methods are an area of active research, which aim to leverage available data to approximate the reachable state [2]. In this context, a data-driven Bayesian framework is developed to learn and to verify complex physical systems via reachability analysis [34].

8. Conclusions

In this paper, we propose an online data-driven approach for learning and reachability analysis of SHS model. The proposed approach can be applied to many modern CPS with a multi-modal behaviour when a parametric model is hard to obtain. As a practical example, we illustrate the efficacy of the approach on smart buildings. The results indicate that our approach runs efficiently in an online fashion and provides a statistical distribution of the reachable state with a good performance.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Air Force Office of Scientific Research [DDDAS FA9550-18-1-0126]; National Science Foundation [CNS-1238959].

ORCID

Hamzah Abdel-Aziz http://orcid.org/0000-0002-8991-0451
Xenofon Koutsoukos http://orcid.org/0000-0002-0923-6293

References

[1] Hu J, Lygeros J, Sastry S. Towards a theory of stochastic hybrid systems. In: HSCC. Vol. 1790; Berlin, Heidelberg: Springer; 2000. p. 160–173.

[2] Bujorianu LM. Stochastic reachability analysis of hybrid systems. London: Springer Science & Business Media; 2012.

[3] Abdel-Aziz H, Koutsoukos X. Online model learning of buildings using stochastic hybrid systems based on Gaussian processes. J Control Sci Eng. 2017;2017.

[4] Abdel-Aziz H, Koutsoukos X. Learning and reachability analysis for stochastic hybrid systems using mixtures of Gaussian processes. In: 24th Mediterranean Conference on Control and Automation (MED), 2016. Athens, Greece: IEEE; 2016. p. 332–337.

[5] Crawley DB, Lawrie LK, Winkelmann FC, et al. Energyplus: creating a new-generation building energy simulation program. Energy Build. 2001;33(4):319–331.

[6] Occupancy simulator [Lawrence berkeley national laboratory]. 2016. Available from: http://occupancysimulator.lbl.gov/.(accessed Oct 2017).

[7] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Cambridge, Massachusetts, USA: The MIT Press; 2006.
[8] MacKay DJ. Information theory, inference and learning algorithms. Cambridge, UK: Cambridge university press; 2003.
[9] Girard A, Rasmussen CE, Candela JQ, et al. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In: Advances in neural information processing systems; Vancouver, British Columbia, Canada; 2003. p. 545–552.
[10] Candela JQ, Girard A, Larsen J, et al. Propagation of uncertainty in Bayesian kernel models-application to multiple-step ahead forecasting. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). Vol. 2; Hong Kong, China: IEEE; 2003. p. II–701.
[11] Deisenroth M, Fox D, Rasmussen C. Gaussian processes for data-efficient learning in robotics and control. IEEE Trans Pattern Anal Mach Intell. 2015 Feb;37(2):408–423.
[12] McLachlan G, Peel D. Finite mixture models. John Wiley & Sons; 2004.
[13] Kaufman L, Rousseeuw PJ. Finding groups in data: an introduction to cluster analysis. Vol. 344. John Wiley & Sons; 2009.
[14] Tresp V. Mixtures of Gaussian processes. In: NIPS; Vancouver, British Columbia, Canada; 2000. p. 654–660.
[15] Quiñonero-Candela J, Rasmussen CE. A unifying view of sparse approximate Gaussian process regression. J Machine Learn Res. 2005;6(Dec):1939–1959.
[16] MATLAB and Statistics Toolbox Release 2017a. 2017.
[17] Feng X, Yan D, Hong T. Simulation of occupancy in buildings. Energy Build. 2015;87:348–359.
[18] Gray FM, Schmidt M. Thermal building modelling using Gaussian processes. Energy Build. 2016;119:119–128.
[19] Kocijan J Gaussian process models for systems identification. In: Proceeding of 9th International PhD Workshop on Systems and Control; Vancouver, British Columbia, Canada; 2008. p. 8–15.
[20] Roberts S, Osborne M, Ebden M, et al. Gaussian processes for time-series modelling. Phil Trans R Soc A. 2013;371(1984):20110550.
[21] Turner RD, Deisenroth MP, Rasmussen CE. State-space inference and learning with Gaussian processes. In: AISTATS; Sardinia, Italy; 2010. p. 868–875.
[22] Kupcsik AG, Deisenroth MP, Peters J, et al. Data-efficient generalization of robot skills with contextual policy search. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013; Bellevue, Washington USA; 2013. p. 1401–1407.
[23] Girard A, Murray-Smith R. Gaussian processes: prediction at a noisy input and application to iterative multiple-step ahead forecasting of time-series. In: Switching and learning in feedback systems. Berlin, Heidelberg: Springer; 2005. p. 158–184.
[24] Hardy J, Havlak F, Campbell M. Multi-step prediction of nonlinear Gaussian process dynamics models with adaptive Gaussian mixtures. Int J Rob Res. 2015;34(9):1211–1227.
[25] Koutroumpas K, Cinquemani E, Kouretas P, et al. Parameter identification for stochastic hybrid systems using randomized optimization: a case study on subtilin production by bacillus subtilis. Nonlinear Anal: Hybrid Syst. 2008;2(3):786–802.
[26] Santana PH, Lane S, Timmons E, et al. Learning hybrid models with guarded transitions. In: AAAI; Austin Texas, USA; 2015. p. 1847–1853.
[27] Pillonetto G. A new kernel-based approach to hybrid system identification. Automatica. 2016;70:21–31.

[28] Bujorianu ML. Extended stochastic hybrid systems and their reachability problem. In: Hybrid systems: computation and control. Berlin, Heidelberg: Springer; 2004. p. 234–249.
[29] Kushner HJ. Probability methods for approximations in stochastic control and for elliptic equations. Vol. 129. New York, NY: Academic Press; 1977.
[30] Prandini M, Hu J A stochastic approximation method for reachability computations. In: Stochastic hybrid systems. Berlin, Heidelberg: Springer; 2006. p. 107–139.
[31] Abate A, Prandini M, Lygeros J, et al. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. Automatica. 2008;44(11):2724–2734.
[32] Krystul J, Blom HA. Sequential Monte Carlo simulation of rare event probability in stochastic hybrid systems. In: Proceedings of the 16th IFAC World Congress. Prague, Czech Republic: Citeseer; 2005. p. 4–8.
[33] Riley D, Koutsoukos X, Riley K. Reachability analysis for stochastic hybrid systems using multilevel splitting. In: Hybrid systems: computation and control. Berlin, Heidelberg: Springer; 2009. p. 460–464.
[34] Haesaert S, Van Den Hof PM, Abate A. Data-driven and model-based verification via Bayesian identification and reachability analysis. Automatica. 2017;79:115–126.