Learning and Control using Gaussian Processes

Towards bridging machine learning and controls for physical systems

Achin Jain*,§, Truong X. Nghiem*,§,‡, Manfred Morari§, Rahul Mangharam§

§*University of Pennsylvania, Philadelphia, PA 19104, USA*

‡*Northern Arizona University, Flagstaff, AZ 86011, USA*

**These authors contributed equally to this work.*

***Abstract*—Building physics-based models of complex physical systems like buildings and chemical plants is extremely cost and time prohibitive for applications such as real-time optimal control, production planning and supply chain logistics. Machine learning algorithms can reduce this cost and time complexity, and are, consequently, more scalable for large-scale physical systems. However, there are many practical challenges that must be addressed before employing machine learning for closed-loop control. This paper proposes the use of Gaussian Processes (GP) for learning control-oriented models: (1) We develop methods for the optimal experiment design (OED) of functional tests to learn models of a physical system, subject to stringent operational constraints and limited availability of the system. Using a Bayesian approach with GP, our methods seek to select the most informative data for optimally updating an existing model. (2) We also show that black-box GP models can be used for receding horizon optimal control with probabilistic guarantees on constraint satisfaction through chance constraints. (3) We further propose an online method for continously improving the GP model in closed loop with a real-time controller. Our methods are demonstrated and validated in a case study of building energy control and Demand Response. We show that OED achieves faster learning rate than random sampling, reducing the duration of functional tests by upto 50%. The GP controller provides the desired curtailment perfectly with maximum 1.7% prediction error.**

***Index Terms*—Machine learning, Gaussian Processes, optimal experiment design, receding horizon control, active learning**

I. Introduction

Machine learning and control theory are two foundational but disjoint communities. Machine learning requires data to produce models, and control systems require models to provide stability, safety or other performance guarantees. Machine learning is widely used for regression or classification, but thus far data-driven models have not been suitable for closed-loop control of physical plants. The current challenge with data-driven approaches is to close the loop for real-time control and decision making.

Consider a dynamical system subject to external disturbances. The first and foremost requirement for making any control decision is to obtain the underlying control-oriented predictive model of the system. With a reasonable forecast of the external disturbances, these models should predict the state of the system in the future and thus a predictive controller based on Model Predictive Control (MPC) can act preemptively to provide a desired behavior. In particular, MPC has been proven to be very powerful for multivariable systems in the presence of input and output constraints, and forecast of the disturbances. The caveat is that MPC requires a reasonably accurate physical representation of the system. This makes MPC unsuitable for control of complex plants such as natural gas processing, oil refineries, boilers, manufacturing plants, and buildings where the user expertise, time, and associated sensor costs required to develop a model are very high. It is concluded in [1] that the required effort for model development and engineering appears to be too high to justify the deployment of MPC in everyday building projects.

There are three main reasons for modeling complexity.

(1) Model capture using only historical data is not suitable for control. Historical data, as large as it may be, does not capture the full model dynamics as the control setpoints are based on rule-based strategies, thus lack in input excitation. Therefore, we need functional tests to excite the system with wide range of control inputs. However, in practice, functional tests are permitted only for a few hours in a month.

(2) Change in model properties over time - even if the model is identified once via an expensive route as in [1], as the model changes with time, the system identification must be repeated to update the model. Thus, model adaptability or adaptive control is desirable for such systems.

(3) Model heterogeneity further prohibits the use of model-based control. For example, unlike the automobile or the aircraft industry, each building is designed and used in a different way. Therefore, this modeling process must be repeated for every new building.

Due to aforementioned reasons, the control strategies in such systems are often limited to fixed, sometimes ad-hoc, rules that are based on best practices. The key question now is: can we employ data-driven techniques to reduce the cost of modeling, and still exploit the benefits that MPC has to offer? We therefore look for automatic data-driven approaches to control that are also adaptive, scalable and interpretable. We solve this problem by bridging the gap between Machine Learning and Predictive Control.

A. Challenges in bridging machine learning and controls

It is important to note that the standard machine learning regression used for prediction is fundamentally different from using machine learning for control synthesis. In the former, all the inputs to the model (also called regressors or features) are known, while in the latter some of the inputs that are the control variables must be optimized in real-time for desired performance. We next discuss the practical challenges in using machine learning algorithms for control.

(1) Data quality: Most of the historical data that are available from complex systems like buildings are based on some rule-based controllers. Therefore, the data may not be sufficient to explain the relationship between the inputs and the outputs. To obtain richer data with enough excitation in the inputs, new experiments must be done either by exciting the inputs randomly or by a procedure for optimal experiment design (OED) [2], [3]. This paper proposes a procedure for OED using Gaussian Processes (GP) to recommend control strategies.
(2) Computational complexity: Depending upon the learning algorithm, the output from a learned model is a non-linear, non-convex and sometimes non-differentiable (eg. Random Forests [4]) function of the inputs with no closed-form expression. Using such models for control synthesis where some of the inputs must be optimized can lead to computationally intractable optimization. Our previous work uses an adaptation of Random Forests which overcomes this problem by separation of variables to derive a linearized input-output mapping at each time step [5]. This paper uses GPs for receding horizon control where the output mean and variance are analytical functions of the inputs, albeit non-convex.
(3) Performance guarantees and robustness: A desired characteristic for closed-loop control is to provide performance guarantees. This becomes hard when a black-box is used to replace a physical model. However, it is possible to provide probabilistic guarantees with a learning algorithm based on Gaussian Processes. GPs allow us to define chance constraints or account for model uncertainty in the cost while solving the optimization problem. This helps bound the performance errors with high confidence. Handling disturbance uncertainties or robustness to sensor failures in this framework is part of our on-going work and is thus excluded from this paper.
(4) Model adaptability: It is often the case that the model properties change with time, and thus, the learned model must also be updated when required. The traditional mode of system identification, done repeatedly, can be time and cost prohibitive, especially in the case of buildings. In this paper, we show how GPs can be updated online to account for changes in the properties of the system.

B. Overcoming practical challenges

To address these challenges, we can take two different approaches based on how machine learning is used to learn the models.
(1) Mix of black-box and physics-based models: In this approach, we use machine learning to learn only the dynamics of a sub-system or to model uncertainties in the dynamics. An example of former is in the use of machine learning for perception and model-based control for low-level control in autonomous cars [6]. Examples on learning uncertainties in the models include [7], [8].
(2) Fully black-box models: The full dynamical model can also be obtained using only machine learning algorithms. This deviates from the traditional notion of system identification where a physics-based structure is assumed to begin with. An example would be fully autonomous control using camera where control actions are mapped to raw image pixels [9].

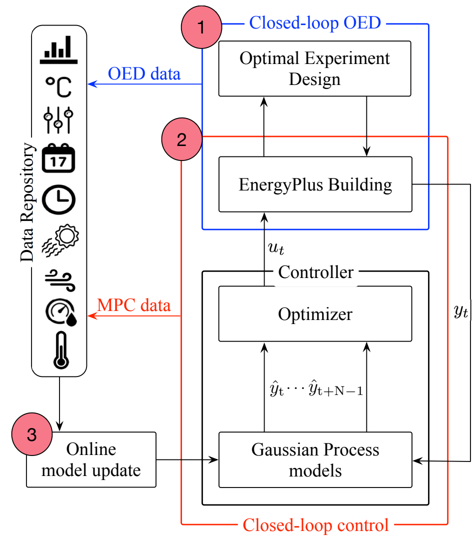

Fig. 1: Paper contributions: (1) Optimal experiment design sequentially samples the inputs and applies to the system to generate training data, (2) Model Predictive Controller uses a Gaussian Process model learned on the OED data for receding horizon control, (3) the new data generated by controller is used to update the model online.

For the application to building control in context of Demand Response, where there is a massive cost to physical modeling, this paper explores the latter route to bypass the modeling difficulties as summarized before. The data-driven approach allows to scale this methodology to multi-building campus and in general to many more applications like control of autonomous systems.

C. Contributions

This paper addresses the aforementioned practical challenges using fully black-box models based on Gaussian Processes. In particular, this paper has the following contributions.
(1) Optimal experiment design: We develop a procedure for optimal experiment design with the dynamical system in a closed-loop by exploiting the variance in the predictions from a GP model. We show that under limited system availability and operation constraints, OED can provide faster learning rate than uniform random sampling or pseudo binary random sampling. In a related work [10], GPs are used for sensor placement to capture maximum information, whereas our method sequentially recommends control strategies to generate more informative training data, reducing the duration of functional tests by upto 50%.
(2) Stochastic Model Predictive Control: We show that the dynamical GP model can be used for real-time closed-loop finite horizon receding horizon control with probabilistic guarantees on constraint satisfaction. We again use the variance in the predictions from a GP model to make decisions where the model is most confident. In the case of Demand Response, we show that GP controller provides the necessary curtaiment with maximum 1.7% prediction error.
(3) Online model update: We propose an online method to update the GP model as new data are generated by running the GP-based controller in a closed-loop with the physical system. Our method maximizes the information gain to select

the best subset of data to update the model, thereby reducing the need for repetitive functional tests as systems properties change with time.

An overview of the organization is shown in Fig. 1. We apply all three methods to large-scale buildings in EnergyPlus [11], high fidelity building simulation software. In the context of load curtailment for Demand Response, we apply OED to recommend control strategies to learn a model, fast and accurately. We show that MPC with GPs can provide the desired load curtailment with high confidence. After running the controller for a few weeks, we update the GP model with newly collected data, thus avoiding the need for a functional test in a new season.

D. Related Work

A broad range of data-driven modeling, assessment, and control methods for DR with buildings have been investigated in the literature. Regression trees were used in our previous work [12], [13], [5] to model and compute setpoint schedules of buildings for DR. Regression and GP models were investigated for forecasting short- and long-term building energy consumption in [14]. Simulation studies with EnergyPlus and regression models were used in [15] to quantify the flexibility of buildings for DR using setpoint change rules. The authors of [16] compared four data-driven methods for building energy predictions and concluded that the Gaussian approaches were accurate and highly flexible, and the uncertainty measures could be helpful for certain applications involving risks.

II. Gaussian Processes

In this section, we briefly introduce modeling with Gaussian Process (GP) and its applications in control. More details can be found in [17] and [18].

Definition 1 ([17]): A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

Consider noisy observations y of an underlying function $f : \mathbb{R}^n \mapsto \mathbb{R}$ through a Gaussian noise model: $y = f(x) + \mathcal{N}\left(0, \sigma_n^2\right)$, $x \in \mathbb{R}^n$. A GP of y is fully specified by its mean function $\mu(x)$ and covariance function $k(x, x')$,

$$\mu(x;\theta) = \mathbb{E}[f(x)] \tag{1}$$

$$k(x, x';\theta) = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))] + \sigma_n^2 \delta(x, x')$$

where $\delta(x, x')$ is the Kronecker delta function. The hyperparameter vector θ parameterizes the mean and covariance functions. This GP is denoted by $y \sim \mathcal{GP}(\mu, k; \theta)$.

Given the regression vectors $X = [x_1, \ldots, x_N]^T$ and the corresponding observed outputs $Y = [y_1, \ldots, y_N]^T$, we define training data by $\mathcal{D} = (X, Y)$. The distribution of the output $y_\star$ corresponding to a new input vector $x_\star$ is a Gaussian distribution $\mathcal{N}\left(\bar{y}_\star, \sigma_\star^2\right)$, with mean and variance given by

$$\bar{y}_\star = g_{\mathrm{m}}(x_\star) := \mu(x_\star) + K_\star K^{-1}(Y - \mu(X)) \tag{2a}$$

$$\sigma_\star^2 = g_{\mathrm{v}}(x_\star) := K_{\star\star} - K_\star K^{-1} K_\star^T, \tag{2b}$$

where $K_\star = [k(x_\star, x_1), \ldots, k(x_\star, x_N)]$, $K_{\star\star} = k(x_\star, x_\star)$, and K is the covariance matrix with elements $K_{ij} = k(x_i, x_j)$.

Note that the mean and covariance functions are parameterized by the hyperparameters θ, which can be learned by

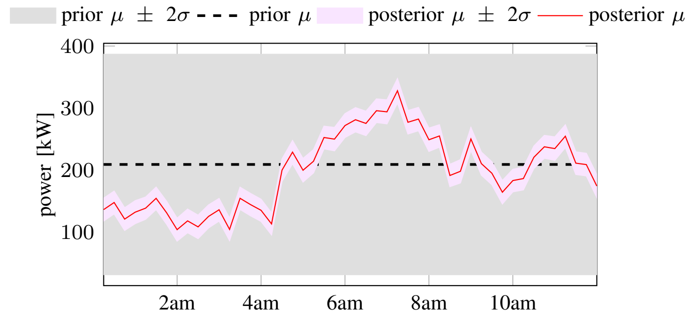

Fig. 2: Example of priors calculated using (1) and posteriors using (2) for predicting power consumption of a building for 12 hrs. Initially the mean is constant because $\mu(x)$ is constant, and we observe a high variance. The posterior agrees with the actual power consumption with high confidence.

maximizing the likelihood: $\arg\max_\theta \Pr(Y|X, \theta)$. The covariance function $k(x, x')$ indicates how correlated the outputs are at x and x', with the intuition that the output at an input is influenced more by the outputs of nearby inputs in the training data $\mathcal{D}$. In other words, a GP model specifies the structure of the covariance matrix of, or the relationship between, the input variables rather than a fixed structural input–output relationship. It is therefore highly flexible and can capture complex behavior with fewer parameters. An example of GP prior and posterior is shown in Fig. 2. We use a constant mean function and a combination of squared exponential kernel and rational quadratic kernel as described in Sec. VI-B. There exists a wide range of covariance functions and combinations to choose from [17].

GPs offer several advantages over other machine learning algorithms that make them more suitable for identification of dynamical systems.

1) GPs provide an estimate of uncertainty or confidence in the predictions through the predictive variance. While the predictive mean is often used as the best guess of the output, the full distribution can be used in a meaningful way. For example, we can estimate a 95% confidence bound for the predictions which can be used to measure control performance.
2) GPs work well with small data sets. This ability is generally useful for any learning application.
3) GPs allow including prior knowledge of the system behavior by defining priors on the hyperparameters or constructing a particular structure of the covariance function. This feature enables incorporating domain knowledge into the GP model to improve its accuracy.

A. Gaussian Processes for Dynamical Systems

GPs can be used for modeling nonlinear dynamical systems, by feeding autoregressive, or time-delayed, input and output signals back to the model as regressors [18]. Specifically, in control systems, it is common to use an autoregressive GP to model a dynamical system represented by the nonlinear function $y_t = f(x_t)$ where

$$x_t = [y_{t-l}, \ldots, y_{t-1}, u_{t-m}, \ldots, u_t, w_{t-p}, \ldots, w_{t-1}, w_t].$$

Here, t denotes the time step, u the control input, w the exogenous disturbance input, y the (past) output, and l, m, and

p are respectively the lags for autoregressive outputs, control inputs, and disturbances. Note that u_t and w_t are the current control and disturbance inputs. The vector of all autoregressive inputs can be thought of as the current state of the model. A dynamical GP can then be trained from data in the same way as any other GPs.

When a GP is used for control or optimization, it is usually necessary to simulate the model over a finite number of future steps and predict its multistep-ahead behavior. Because the output of a GP is a distribution rather than a point estimate, the autoregressive outputs fed to the model beyond the first step are random variables, resulting in more and more complex output distributions as we go further. Therefore, a multistep simulation of a GP involves the propagation of uncertainty through the model. There exist several methods for uncertainty propagation in GPs [18].

We use the *zero-variance method* that does not propagate uncertainty. At each step, the autoregressive outputs are replaced by their corresponding expected values. Obviously, this method will underestimate the variances of the output distributions. However, its computational simplicity is attractive, especially in optimization applications where the GP must be simulated for many time steps. In such cases, if the prediction error caused by not propagating uncertainty is insignificant, the zero-variance method can and should be used. For more detailed discussions on this topic, see [18].

III. Optimal Experiment Design

In this section, we address the practical challenge of "Data quality" listed in Sec. I-A.

In general, the more data we have, the better we can learn a model using machine learning algorithms. These data are often obtained by running tests, called *functional tests*, on the real system. However, in many applications, the amount of training data we can practically obtain is usually limited due to many factors, such as a short permitted duration for functional tests and operational or safety constraints of the physical system. For example, in the context of buildings as we will discuss in Sec. VI, a functional test typically involves changing various set-points of the building energy control system in order to excite the different components and operation modes of the building, so that the obtained data will reflect their behaviors. It is often the case that a functional test in a building is limited by the short time window during which the set-points are allowed to change, and by the maximum allowable rates of change of these set-points. Subject to these constraints, it is desirable to optimally design the functional tests so that the data quality is maximized, in the sense that the model obtained from the data with a specific learning technique likely has the best quality possible. This practice is known as *optimal experiment design* (OED).

A. Information theoretic approach to OED

In this section, we present an *information theoretic* approach for OED to incrementally design or select the best data points for explaining the behavior of the underlying physical system with GP. This is achieved by exploiting the predictive variance in GP regression (2). The goal here is to update the hyperparameters θ in the model $y \sim \mathcal{GP}(\mu(x), k(x); \theta)$ as new samples are observed sequentially. One popular method for selecting the next sample is the point of Maximum Variance (MV), which is also widely used for Bayesian Optimization using GPs [19]. Since we can calculate the variance in y for any x, OED based on MV can be directly computed using (2). However, another approach which has been shown to result in better samples for learning the hyperparameters θ is maximizing the Information Gain (IG) [10]. In Sec. VI, we will compare both approaches in a case study.

The IG approach selects the sample which adds the maximum information to the model, i.e. which reduces the maximum uncertainty in θ. If we denote the existing data before sampling by $\mathcal{D}$, then the goal is to select x that maximizes the information gain defined as

$$\arg\max_x H(\theta|\mathcal{D}) - \mathbb{E}_{y\sim\mathcal{N}(\bar{y}(x),\sigma^2(x))} H(\theta|\mathcal{D}, x, y), \tag{3}$$

where H is the Shannon's Entropy given by

$$H(\theta|\mathcal{D}) = -\int p(\theta|\mathcal{D}) \log(p(\theta|\mathcal{D}))d\theta. \tag{4}$$

Since $y|x \sim \mathcal{N}\left(\bar{y}(x), \sigma^2(x); \theta\right)$, we need to take an expectation over y. When the dimension of θ is large, computing entropies is typically computationally intractable. Using the equality $H(\theta) - H(\theta|y) = H(y) - H(y|\theta)$, we can rewrite (3) equivalently as

$$\arg\max_x H(y|x, \mathcal{D}) - \mathbb{E}_{\theta\sim p(\theta|\mathcal{D})} H(y|x, \theta). \tag{5}$$

In this case, as the expectation is defined over θ, (5) is much easier to compute because y is a scaler. For further details, we refer the reader to [20]. The first term in (5) can be calculated by marginalizing over the distribution of $\theta|\mathcal{D}$:

$$\begin{aligned} p(y|x, \mathcal{D}) &= \mathbb{E}_{\theta\sim p(\theta|\mathcal{D})} p(y|x, \theta, \mathcal{D}) \\ &= \int p(y|x, \theta, \mathcal{D}) p(\theta|\mathcal{D}) d\theta \end{aligned} \tag{6}$$

for which the exact solution is difficult to compute. We therefore use an approximation described in [21]. It is shown that for $\theta|\mathcal{D} \sim \mathcal{N}\left(\bar{\theta}, \Sigma\right)$, we can find a linear approximation to $\bar{y}(x) = a^T(x)\theta + b(x)$ such that

$$p(y|x, \mathcal{D}) \sim \mathcal{N}\left(a^T\bar{\theta} + b, \sigma^2 + a^T\Sigma a\right) \tag{7}$$

in the neighborhood of $\bar{\theta}$. Under the same approximation, the variance $p(y|x, \mathcal{D})$ is approximated to be

$$\begin{aligned} \tilde{\sigma}^2(x) = \frac{4}{3}\sigma^2(x) + \frac{\partial \bar{y}(x)}{\partial\theta}^T \Sigma \frac{\partial \bar{y}(x)}{\partial\theta} + \\ \frac{1}{3\sigma^2(x)} \frac{\partial \sigma^2(x)}{\partial\theta}^T \Sigma \frac{\partial \sigma^2(x)}{\partial\theta} \end{aligned} \tag{8}$$

evaluated at $\bar{\theta}$ while the second term in (5) can be written as $H(y|x, \bar{\theta})$. Finally, maximizing the information gain in (3) is equivalent to maximizing $\tilde{\sigma}^2(x)/\sigma^2(x)$. Next, we apply this result for sequential optimal experiment design.

B. Sequential experiment design with Gaussian Processes

Our goal is to update the hyperparameters θ of the GP efficiently as new data is observed. To begin the experiment design, we assume that we only know about which features x have an influence on the output y. This is often known in practice. For example, for the case study in Sec. VI, the output

Algorithm 1 Sequential sampling for OED based on IG

procedure INITIALIZATION
  if initial $\mathcal{D} := (X, Y)$ **then**
    Compute $\theta_{\text{MLE}} = \arg\max_{\theta^{\text{MLE}}} \Pr(Y|X, \theta)$
    Assign priors $\theta_0 \sim \mathcal{N}\left(\theta_{\text{MLE}}, \sigma^2_{\text{init}}\right)$
  else
    Assign priors $\theta_0 \sim \mathcal{N}\left(\mu_{\text{init}}, \sigma^2_{\text{init}}\right)$
  end if
end procedure
procedure SAMPLING
  while $t < t_{\max}$ **do**
    Calculate features x_t in (9) as a function of u_t
    Solve (10) to calculate optimal u_t^*
    Apply u_t^* to the system and measure y_t
    $\mathcal{D} = \mathcal{D} \cup (x_t, y_t)$
    Update $\theta_{\text{t}} = \arg\max_{\theta^{\text{MAP}}} \Pr(Y|X, \theta_{\text{t}-1})$
  end while
end procedure

of interest is the building power consumption, and the features we consider include outside air temperature and humidity, time of day to account for occupancy, control setpoints and lagged terms for the output. Then a covariance structure of GP must be selected. For the example above, we chose a squared exponential kernel. If samples $\mathcal{D} := (X, Y)$ are available, we can assign the prior distribution on θ based on the MLE estimate $\arg\max_\theta \Pr(Y|X, \theta)$, i.e. $\theta_0 \sim \mathcal{N}\left(\theta_{\text{MLE}}, \sigma^2_{\text{init}}\right)$ where a suitable value of σ^2_{init} is chosen. Otherwise, the Gaussian priors $\theta_0 \sim \mathcal{N}\left(\mu_{\text{init}}, \sigma^2_{\text{init}}\right)$ are initialized manually.

Now, consider a dynamical GP model introduced in Sec. II, $y_t = f(x_t; \theta)$ where

$$x_t = [y_{t-l}, \ldots, y_{t-1}, u_{t-m}, \ldots, u_t, w_{t-p}, \ldots, w_{t-1}, w_t]. \quad (9)$$

At time t, the current disturbance, and the lagged terms of the output, the control inputs and the disturbance are all known. The current control input $u_t \in \mathbb{R}^u$ is the only unknown feature for experiment design, which we aim to select optimally. For physical systems, very often, we must operate under strict actuation or operation constraints. Therefore, the new sampled inputs must lie within these constraints. To this end, we solve the following optimization problem to compute optimal control set point recommendations u_t^* for experiment design

$$\begin{aligned} \text{maximize}_{u_t} \quad & \tilde{\sigma}^2(x_t)/\sigma^2(x_t) \\ \text{subject to} \quad & u_t \in \mathcal{U} \end{aligned} \quad (10)$$

where x_t is a function of u_t. The new control input u_t^* is applied to the physical system to generate the output y_t, update the parameters θ using maximum a posteriori (MAP) estimate [21], and we proceed to time $t+1$. The algorithm for OED is summarized in Algo. 1.

As an example, in Sec. VI where we learn a dynamical model of a building, the proposed OED method is used to optimally sample the chilled water temperature, supply air temperature and zone-level cooling set-points, subject to operation constraints on the chiller system. The result is illustrated in Fig. 3, which shows the changes in model accuracy between several experiment design methods, for various durations of functional tests. For short functional test durations, our OED methods achieve much more accurate models compared to random sampling methods. The random sampling requires more than 300 hours to reach the same accuracy. Since the historical data is unperturbed, it would require even longer to provide similar accuracy.

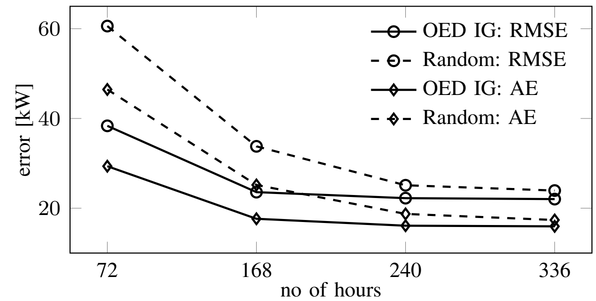

Fig. 3: Error in prediction of power consumption on test data set. RMSE: root mean square error. AE: Absolute error. The errors are much lower with optimal experiment design based on information gain. The random sampling requires more than 300 hours to reach the same accuracy

IV. MODEL PREDICTIVE CONTROL

This section addresses the challenge of "Computational complexity" and "Performance guarantees" listed in Sec. I-A.

Consider a black-box model given by $x_{t+1} = f(x_t, u_t, d_t)$, where x, u, d represent state, input and disturbance, respectively. Depending upon the learning algorithm, f is typically nonlinear, nonconvex and sometimes nondifferentiable (as is the case with regression trees and random forests) with no closed-form expression. Such functional representations learned through black-box modeling may not be directly suitable for control and optimization as the optimization problem can be computationally intractable, or due to nondifferentiabilities we may have to settle with a sub-optimal solution using evolutionary algorithms. In our previous work, we use *separation of variables* that allows us to approximate the Random Forests as affine models in the neighborhood of a given disturbance [5]. The main drawback of this approach is that these models lead to a non-smooth input behavior. Gaussian Processes overcome this problem. Further, GPs can generalize well using only a few samples while also providing an estimate for uncertainty in the predictions. We exploit this property in the optimization to generate input trajectories with high confidence.

Given a GP model of the plant, the *zero-variance* method is used to predict the plant's outputs in a horizon of N time steps starting from the current time t, for $\tau \in \{0, \ldots, N-1\}$:

$$\begin{aligned} y_{t+\tau} &\sim \mathcal{N}\left(\bar{y}_{t+\tau} = g_{\text{m}}(x_{t+\tau}), \sigma^2_{t+\tau} = g_{\text{v}}(x_{t+\tau})\right), \quad (11) \\ x_{t+\tau} &= [\bar{y}_{t+\tau-l}, \ldots, \bar{y}_{t+\tau-1}, u_{t+\tau-m}, \ldots, u_{t+\tau}, \\ &\qquad w_{t+\tau-p}, \ldots, w_{t+\tau-1}, w_{t+\tau}]. \end{aligned}$$

The output at step $t+\tau$ depends upon the control inputs $u_{t+\tau-m}, \ldots, u_{t+\tau}$. We are interested in the following optimization problem with quadratic cost with $R \succ 0$

$$\begin{aligned}
\text{minimize} \quad & \sum_{\tau=0}^{N-1} (\bar{y}_{t+\tau} - y_{\text{ref}})^2 + {u_{t+\tau}}^T R u_{t+\tau} + \lambda \sigma^2_{y,t+\tau} \qquad (12)\\
\text{subject to} \quad & \bar{y}_{t+\tau} = \mu(x_{t+\tau}) + K_\star K^{-1}(Y - \mu(X))\\
& \sigma^2_{y,t+\tau} = K_{\star\star} - K_\star K^{-1} K_\star^T\\
& u_{t+\tau} \in \mathcal{U}\\
& \Pr(y_{t+\tau} \in \mathcal{Y}) \geq 1 - \epsilon
\end{aligned}$$

where the constraints hold for all $\tau \in \{0, \dots, N-1\}$. Here, $K_\star = [k(x_{t+\tau}, x_1), \dots, k(x_{t+\tau}, x_N)]$, $K_{\star\star} = k(x_{t+\tau}, x_{t+\tau})$. The last constraint is a chance constraint, which keeps the plant's output inside a given set $\mathcal{Y}$ with a given probability of at least $1 - \epsilon$. The hyperparameters θ of the mean function μ and the covariance function k are optimized while training GPs as described in Sec. II or by experiment design in Sec. III. We solve (12) to compute optimal $u_t^*, \dots, u_{t+N-1}^*$, apply u_t^* to the system and proceed to time $t+1$.

Although we have an analytical expressions for all the constraints in the optimization, depending upon the choice of mean and covariance functions, the optimization can be computationally hard to solve. For the case study in Sec. VI we choose a combination of a squared exponential and rational quadratic kernel which results in nonconvex problem. We solve (12) using IPOPT [22] and CasADi [23]. Our future work focuses on developing MPC based on sparse GPs which are more scalable to large scale systems.

V. Evolving Gaussian Processes

In this section, we discuss the challenge of "Model adaptability" listed in Sec. I-A.

As the system properties change with time, the learned model must actively update itself so that it best reflects the current behavior of the system. For example, the same GP model may not be suitable to control a building in both Summer and Winter seasons. As we generate more data with time with the controller in the loop, it is intuitive to incorporate the new data into the existing model to improve its accuracy. However, we may not want to use the full new data set for model update for multiple reasons. First, because not all data are created equal, especially in closed loop with a controller, we should select only the most informative subset of data that best explain the system dynamics at the time. Second, since the computational complexity of training and predicting with Gaussian Processes is $\mathcal{O}(n^3)$, where n is number of training samples, the learning and control problems become computationally hard as the size of data increases. Therefore, obtaining the best GP model with the least amount data is highly desired. The solution to this problem lies in *selecting the optimal subset of data*, from the available data, that best explains the system behavior or dynamics. Towards this goal, we extend the result from Sec. III-A.

A. Optimal subset of data selection: selecting the most informative data for periodic model update

Out goal is to filter the most informative subset of data that best explain the dynamics. In this section, we outline a systematic procedure that aims to select the best k samples

Algorithm 2 Optimal subset of data selection

procedure Initialization
  Sample with replacement k integers $\in \{1, \dots, n\}$
  Compute $\theta_{\text{MLE}} = \arg\max_{\theta^{\text{MLE}}} \Pr(Y|X, \theta)$
  Assign priors $\theta_0 \sim \mathcal{N}\left(\theta_{\text{MLE}}, \sigma^2_{\text{init}}\right)$
end procedure
Define $\mathcal{S} = \varnothing$
procedure Sampling
  while $j \leq k$ **do**
    Solve (13) for optimal $x_j | (x_j, y_j) \in \mathcal{D} \setminus \mathcal{S}$
    $\mathcal{S} = \mathcal{S} \cup (x_j, y_j)$
    Update $\theta_{\text{j}} = \arg\max_{\theta^{\text{MAP}}} \Pr(Y|X, \theta_{\text{j}-1})$
  end while
end procedure

from a given set $\mathcal{D}$ of n observations. The main differences between the problem of selecting the best or the most informative subset of data and the sequential sampling for OED described in Sec. III-B are that in the former, (1) all the features x must be optimized as opposed to only control variables u, and (2) the decision has to be made only from the available data rather than sampling.

We begin by selecting k samples randomly, then assign the priors of the hyperparameters θ based on the MLE estimate obtained by learning a GP on the drawn set. Starting with an empty set of samples $\mathcal{S}$, we loop through the full data set $\mathcal{D}$ to identify which sample maximizes the information gain. In this setup, we solve the following optimization problem

$$\underset{x_j | (x_j, y_j) \in \mathcal{D} \setminus \mathcal{S}}{\text{maximize}} \quad \tilde{\sigma}^2(x_j)/\sigma^2(x_j) \qquad (13)$$

Then, we add this sample to $\mathcal{S}$, update θ and proceed until $|\mathcal{S}| = k$. This algorithm is summarized in Algo. 2.

The proposed method is used in a case study in Sec. VI-E to update the learned model from time to time as a controller runs in a closed loop and we generate more data. Fig. 4 shows the improvement in mean prediction error and prediction variance obtained after optimal selection, starting with a model trained on uniformly random sampled data.

VI. Case Study

In January 2014, the east coast electricity grid, managed by PJM, experienced an 86-fold increase in the price of electricity from \$31/MWh to \$2,680/MWh in a matter of 10 minutes. Similarly, the price spiked 32 times from an average of \$25/MWh to \$800/MWh in July of 2015. This extreme price volatility has become the new norm in our electric grids. Building additional peak generation capacity is not environmentally or economically sustainable. Furthermore, the traditional view of energy efficiency does not address this need for *Energy Flexibility*. A promising solution lies with Demand Response (DR) from the customer side – curtailing demand during peak capacity for financial incentives. However, it is a very hard problem for commercial, industrial and institutional plants – the largest electricity consumers – to decide which knobs to turn to achieve the required curtailment, due to the large scale and high complexity of these systems. Therefore, the problem of energy management during a DR event makes an ideal case for our proposed approach of combining machine learning and

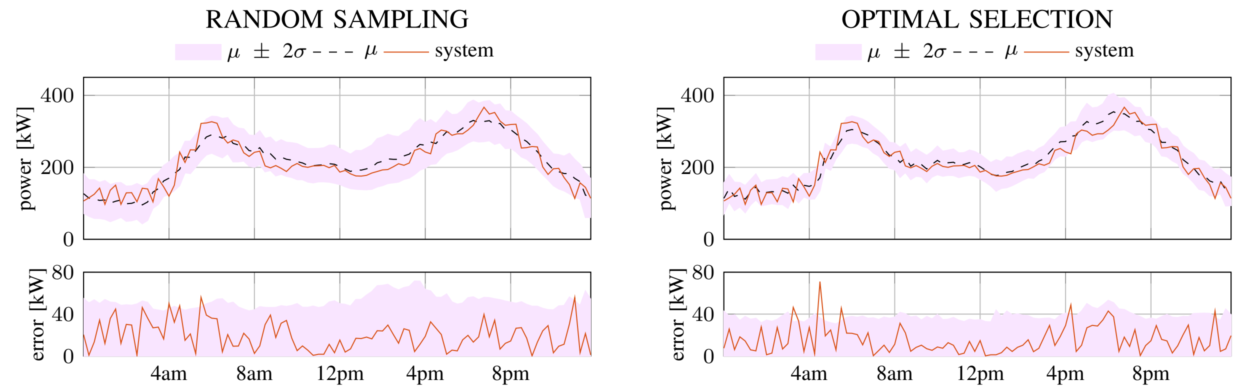

Fig. 4: Left: Selection using random sampling. Right: Optimal subset of data selection. Starting with the random sampling, we use the model parameters and apply Algo. 2 to improve the model accuracy. Both the mean prediction error and the prediction variance are lower for optimal selection based on information gain.

control. In this section, we apply optimal experiment design, receding horizon control based on GPs, and evolving GPs on large scale EnergyPlus models to demonstrate the effectiveness of our approach.

A. Building Description

We use two different U.S. Department of Energy's Commercial Reference Buildings (DoE CRB) simulated in EnergyPlus [11] as the virtual test-bed buildings. The first is a 6-story hotel consisting of 22 zones with a total area of 120,122 sq.ft, with a peak load of about 400 kW. The second building is a large 12-story office building consisting of 19 zones with a total area of 498,588 sq.ft. Under peak load conditions the office can consume up to 1.4 MW of power. Developing a high fidelity physics-based model for these buildings would require massive cost and effort. Leveraging machine learning algorithms, we can now do both prediction and control with high confidence at a very low cost.

We use the following data to validate our results. We limit ourselves to data which can be measured directly from installed sensors like thermostats, multimeters and weather forecasts, thus making it scalable to any other building or a campus of buildings.

- *Weather variables d^w:* outside temperature and humidity – these features are derived from historical weather data.
- *Proxy features d^p:* time of day, day of week – these features are indicators of occupancy and periodic trends.
- *Control variables u:* cooling, supply air temperature and chilled water setpoints – these will be optimized in the MPC problem.
- *Output variable y:* total power consumption – this is the output of interest which we will predict using all the above features in the GP model.

The time step for modeling and control is 15 minutes.

B. Gaussian Process Models

We learn a single GP model of the building and use the *zero-variance method* to predict the outputs y at the future time steps following the current time step. For each prediction step $t+\tau$, where t is the current time and $\tau \geq 0$, the output $y_{t+\tau}$ is a Gaussian random variable given by (11).

$$
\begin{aligned}
y_{t+\tau} &\sim \mathcal{N}\left(\bar{y}_{t+\tau} = g_{\mathrm{m}}(x_{t+\tau}), \sigma^2_{y,t+\tau} = g_{\mathrm{v}}(x_{t+\tau})\right), \qquad (14)\\
x_{t+\tau} &= [\bar{y}_{t+\tau-l}, \ldots, \bar{y}_{t+\tau-1}, u_{t+\tau-m}, \ldots, u_{t+\tau},\\
&\qquad w_{t+\tau-p}, \ldots, w_{t+\tau-1}, w_{t+\tau}].
\end{aligned}
$$

where $w := [d^w, d^p]$. We assume that at time t, $w_{t+\tau}$ are available $\forall \tau$ from forecasts or fixed rules as applicable.

As for the mean and covariance functions of the GP, we use a constant mean μ and a kernel function $k(x, x')$ which is a mixture of constant kernel $k_1(x, x')$, squared exponential kernel $k_2(x, x')$ and rational quadratic kernel $k_3(x, x')$ as

$$
\begin{aligned}
k_1(x, x') &= k,\\
k_2(x, x') &= \sigma^2_{f_2} \exp\left(-\tfrac{1}{2} \textstyle\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\lambda_d^2}\right),\\
k_3(x, x') &= \sigma^2_{f_3}\left(1 + \tfrac{1}{2\alpha} \textstyle\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\lambda^2}\right)^{-\alpha},\\
k(x, x') &= (k_1(x, x') + k_2(x, x')) * k_3(x, x'). \qquad (15)
\end{aligned}
$$

Here, D is the dimension of x, $k_3(x, x')$ is applied to only temporal features like time of day and day of week, while $k_1(x, x')$ and $k_2(x, x')$ are applied to all the remaining features as proposed in [24]. We optimize the hyperparameters θ of the model in (14) using GPML [25].

C. Optimal Experiment Design

OED is powerful when limited data are available for training. To demonstrate this, using Algo. 1, we begin the experiment by assigning $\mathcal{N}(0, 1)$ priors to the kernel hyperparameters. For OED, we only consider the one-step-ahead model with $\tau = 0$ in (14). The goal at time t is to determine what should be the optimal cooling set-point $u_{\mathrm{clg},t}$, supply air temperature set-point $u_{\mathrm{sat},t}$, and chilled water temperature set-point $u_{\mathrm{chw},t}$ which, when applied to the building, will require power consumption y_t such that (x_t, y_t) can be used to learn θ as efficiently as possible. We use the lagged terms of the power consumption, proxy variables, weather variables and their lagged terms to define $x_t(u_{\mathrm{clg,t}}, u_{\mathrm{sat,t}}, u_{\mathrm{chw,t}})$. We assume a practical operational constraint that the chilled water temperature set-point cannot be changed faster than 0.13^oC

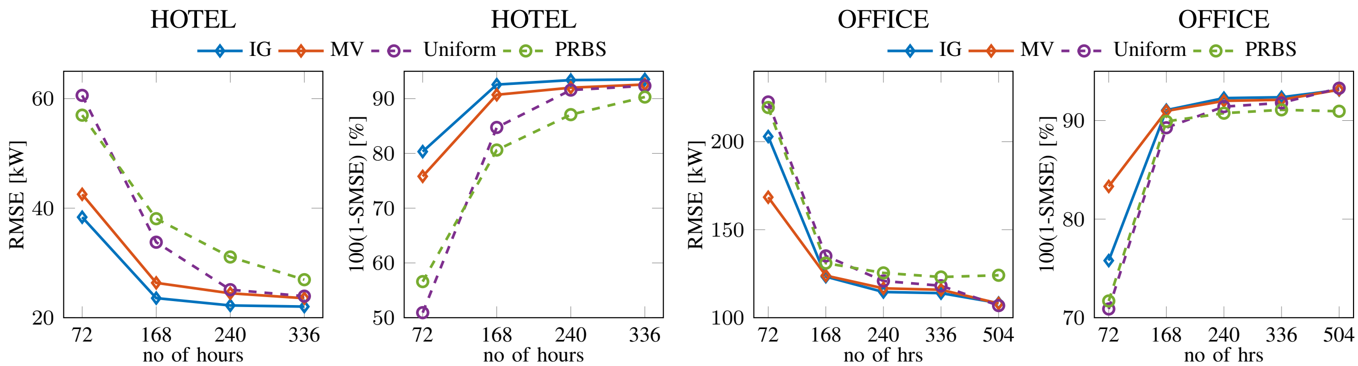

Fig. 5: Comparison of model accuracies for different experiments: OED based on information gain (IG), OED based on maximum variance (MV), uniform random sampling (Uniform) and pseudo random binary sampling (PRBS) for two buildings: hotel (left) and office (right). RMSE denotes Root Mean Square Error and SMSE means Standardized Mean Square Error; lower RMSE and higher 1-SMSE indicate better prediction accuracy.

every minutes. Keeping this constraint and thermal comfort constraints into consideration, we consider the following operational constraints:

$$\begin{aligned} 22^o\mathrm{C} &\leq u_{\mathrm{clg,t}} \leq 27^o\mathrm{C}, \\ 12^o\mathrm{C} &\leq u_{\mathrm{sat,t}} \leq 14^o\mathrm{C}, \\ 3.7^o\mathrm{C} &\leq u_{\mathrm{chw,t}} \leq 9.7^o\mathrm{C}, \\ |u_{\mathrm{chw},t} &- u_{\mathrm{chw},t-1}| \leq 2^o\mathrm{C}. \end{aligned} \tag{16}$$

Finally, we solve the optimization (13), subject to the operational constraints (16), every $15\mathrm{min}$ to calculate optimal inputs for OED.

The results for experiment design in closed-loop with the EnergyPlus building models described in Sec. VI-A are shown in Fig. 5. We compare 4 different methods: OED based on maximum information gain (IG), OED based on maximum variance (MV), uniform random sampling (Uniform) and pseudo random binary sampling (PRBS). The inputs $u_{\mathrm{clg,t}}, u_{\mathrm{sat,t}}, u_{\mathrm{chw,t}}$ generated via OED or random sampling are applied to the building every $15\mathrm{min}$. We repeat OED/random sampling continuously for 14 days and learn a model at the end of each day using the data generated until that time. For example, at the end day of day 3 we have 3×96 samples, at day 7 we have 7×96 samples and so on. As the days progress, we add more training samples and therefore the model accuracy is expected to increase with time. This is visible in both metrics Root Mean Square Error (RMSE) and Standardized Mean Square Error (SMSE) for both buildings.

For OED based on information gain as well as maximum variance, the learning rate is much faster than any random sampling. For the hotel building on the left, the IG method is the best in terms of accuracy. Uniform random sampling and PRBS are far worse in both metrics for approx. 200 hrs. For the same performance, OED reduces the duration of functional tests by over 50%. For the office building on the right, IG is marginally better than MV in terms of SMSE for all days, while MV shows faster learning rate with lower RMSE. Thus for the office building, OED based on IG and MV are comparable. With the random sampling, we observe the same trend as before. Random sampling, both uniform and PRBS require more than 200 hrs for functional tests to achieve the same RMSE and model accuracy.

We have shown that OED can be used to learn a model very fast. In practice, due to operational constraints, the functional tests cannot be performed for sufficiently long time. They are permitted only in a small window, during non-business hours for only a few hours in a month. Even short periodic tests based on OED can provide far better models due to its ability to capture more information in the same amount of time. Thus, OED can drastically reduce the duration for functional tests.

D. Power Reference Tracking Control

This section formulates an MPC approach for the following demand tracking problem. Consider a building, which responds to various setpoints resulting in power demand variations, and a battery, whose state of charge (SoC) can be measured and whose charge/discharge power can be controlled. Given a power reference trajectory, for example a curtailed demand trajectory from the nominal energy consumption profile (the *baseline*), our objective is to control the building and the battery to track the reference trajectory as closely as possible without violating the operational constraints. The building's response to the setpoint changes is modeled by a GP. The battery helps improve the tracking quality by absorbing the prediction uncertainty of the GP. An MPC based on the GP model computes the setpoints for the building and the power of the battery to optimally track the reference demand signal.

For simplicity, we assume an ideal lossless battery model

$$s_{t+1} = s_t + Tb_t \tag{17}$$

where b_t is the battery's power at time step t and s is the battery's SoC. Here, b is positive if the battery is charging and negative if discharging. The battery is subject to power and SoC constraints: $b_{\min} \leq b_t \leq b_{\max}$, and $s_{\min} \leq s_t \leq s_{\max}$ where $s_{\max}$ is the fully-charged level and $s_{\min}$ is the lowest safe discharged level.

The building and the battery are linked via the power tracking constraint which states that $p_t = y_t + b_t$ should track the reference r_t at any time t. Therefore, our objective is to minimize $\delta_t = r_t - p_t$. In this way, the battery helps reject the uncertainty of the GP and acts as an energy buffer to increase the tracking capability of the system. The controller tries to keep $\delta_t = 0$, however when exact tracking is impossible, it will maintain the operational safety of the system while keeping δ_t as small as possible. The bounds on the battery's power and SoC lead to corresponding chance constraints. We wish to guarantee that at each time step, the power and SoC constraints are satisfied with probability at least $(1-\epsilon_p)$ and at least $(1-\epsilon_s)$, respectively, where $0<\epsilon_p,\epsilon_s\leq\frac{1}{2}$ are given constants. Specifically, for each τ in the horizon,

$$\Pr\left(b_{\min}\leq b_{\tau+t}\leq b_{\max}\right)\geq 1-\epsilon_p \tag{18}$$
$$\Pr\left(s_{\min}\leq s_{\tau+t}\leq s_{\max}\right)\geq 1-\epsilon_s \tag{19}$$

where $b_{t+\tau}$ and $s_{t+\tau}$ are Gaussian random variables whose mean and variance are given by

$$\bar{b}_{t+\tau}=r_t-\delta_{t+\tau}-\bar{y}_{t+\tau},\quad \sigma^2_{b,t+\tau}=\sigma^2_{y,t+\tau}, \tag{20}$$
$$\bar{s}_{t+\tau+1}=s_t+T\sum_{k=t}^{t+\tau}\bar{b}_k,\ \sigma^2_{s,\tau+1}=T^2\sum_{k=t}^{t+\tau}\sigma^2_{y,k}. \tag{21}$$

For further details on modeling we refer the reader to our previous work [24]. To track a given reference power signal, we solve the following stochastic optimization problem to optimize $\delta_{\tau+t},u_{\text{clg},\tau+t},u_{\text{sat},\tau+t},u_{\text{chw},\tau+t}\ \forall\tau\in\{0,\ldots,N-1\}$

$$\begin{aligned}\underset{\delta,u}{\text{minimize}}\quad & \sum_{\tau=0}^{N-1}(\delta_{\tau+t})^2+\lambda\sigma^2_{y,\tau+t} \\ \text{subject to}\quad & \text{dynamics constraints (14), (18)} - \text{(21)} \\ & \text{operation constraints (16).}\end{aligned} \tag{22}$$

The term $\sigma^2_{y,\tau+t}$ in the objective functions ensures control setpoints where model is more confident. At time t, we solve for $u_t^*,\ldots,u_{t+N-1}^*$, apply the first input u_t^* to the building, and proceed to the next time step.

The office building has a large HVAC system, so for this building we consider the following Demand Response scenario. Due to price volatility, the office receives a request from the aggregator to shed 90 kW load between 2-4pm. Now, the goal of the operators is to decide setpoints that would guarantee this curtailment while following stringent operation and thermal comfort constraints. Rule-based strategies do not guarantee this curtailment and hence pose a huge financial risk. Using our data-driven approach for control, we can synthesize optimal setpoint recommendations. Fig. 6 shows the load shedding between 2-4pm. The baseline power consumption indicates the usage if there was no DR event, or in other words if the building would have continued to operate under normal conditions. The reference for tracking differs from baseline by 90 kW during 2-4pm. The mean prediction denoted by μ is the output $\bar{y}_t$ which follows the reference signal closely as the input constraints are never active. The actual (system) building power consumption differs only marginally from the reference as shown in Fig. 7. The maximum prediction error during the DR event is 22.5 kW (1.7%) and the mean absolute error is 7.9 kW (0.6%). While tracking the reference signal, the battery power compensates for this error to provide near perfect tracking. The optimal setpoints are shown in Fig. 8. The controller has a prediction horizon of 1 hr. It kicks in at 1:15pm and increases the cooling, chilled water temperature and supply air temperature setpoints to meet the requirement of 90 kW. After 4pm, we continue to follow the baseline signal for the next one hour to reduce the effect of the kickback.

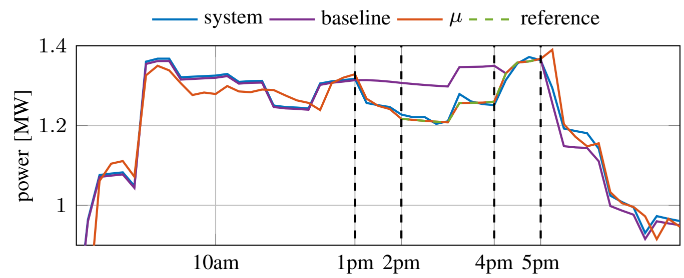

Fig. 6: The reference power signal is closely tracked by GP model providing sustained curtailment of 90 kW (with respect to the baseline) during the Demand Response event 2-4pm. Due to 1hr horizon in the control problem, the curtailment starts at 1:15pm, and the controller is further active until 5pm to reduce kickback.

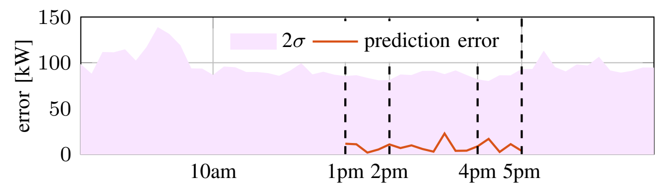

Fig. 7: The prediction error during the DR event is always less 22.5 kW (1.7%) and the mean absolute error is 7.9 kW (0.6%). This error is compensated by the battery.

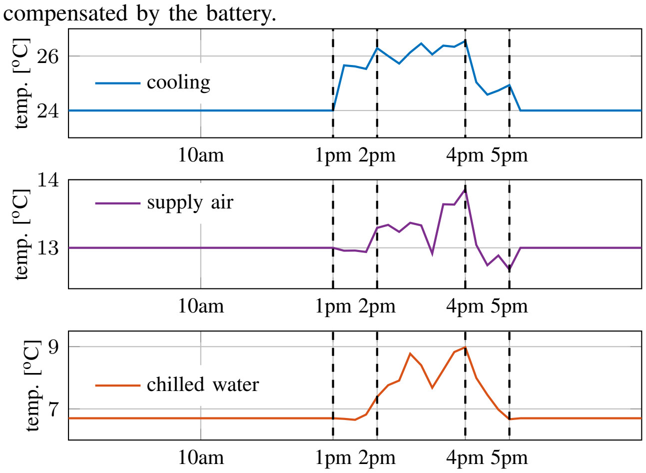

Fig. 8: Optimal set points obtained after solving optimization (22).

E. Online Model Update

The GP model used for control in Sec. VI-D is trained on the data set $\mathcal{D}$ generated from the OED procedure in Sec. VI-C. We run the controller in a closed loop with the building for two weeks and collect the new data set $\mathcal{D}'$ generated in the process. $\mathcal{D}'$ contains useful and current information about the dynamics of the system, that is beneficial for updating the GP model to improve its accuracy. This can be achieved by re-training the model on the combined data set $D\cup D'$. However, due to the fast growth of computational complexity of GPs with the size of the training data set $(\mathcal{O}(|D\cup D'|^3))$, it is not recommended to re-train the model on $\mathcal{D}\cup\mathcal{D}'$, especially when $|\mathcal{D}|$ and $|\mathcal{D}'|$ are large. Therefore, we select the most informative subset of data $\mathcal{S}\subset\{\mathcal{D}\cup\mathcal{D}'\}$ to update the GP model.

We consider two different GP models learned using OED for 14 days and 21 days in the month of June. In the first case, we

TABLE I: Comparison of Root Mean Square Error (RMSE) in kW between Updated GP and Outdated GP models for different months.

	July		August		September	
	14-day	21-day	14-day	21-day	14-day	21-day
Outdated GP	65.2	63.8	91.81	93.2	103.2	101.4
Updated GP	58.9	59.4	86.4	85.7	97.7	94.9
% improved	9.6%	6.9%	5.9%	7.9%	5.3%	6.4%

have 14×96 samples and in the second case 21×96 samples for training. We run each GP model in closed-loop with the building in a different month for further 14 days. To update the model for evolving GP, we use our algorithm in Algo. 2 for optimal subset of data selection to choose the most informative 14×96 samples in the first case and 21×96 samples in the second case. These models are denoted by "Updated GP" in Tab. I. We compare the performance of these models against the original model, referred to as an "Outdated GP" in Tab. I since this model does not include the most up-to-date data about the current system which has evolved due to seasonal and operational changes. We repeat this for the months of July, August and September. Finally, we test the prediction accuracy (RMSE) of both models on the remaining 14days of the respective months. For example, when the outdated GP model is used for control from Aug 1 to Aug 14, we calculate the prediction error from Aug 15 to Aug 28. For the office building, our results show that the updated GP model is better in all the cases with lower RMSE, decreasing the model errors by at least 5%.

VII. Conclusion

Learning black-box models for real-time control reduces the cost and time required to model complex physical systems like buildings and chemical plants *by an order of magnitude*. This paper addresses the various challenges associated in bridging machine learning and controls with application to load curtailment for Demand Response. (1) We propose a method for optimal experiment design using Gaussian Processes to recommend strategies for functional test (in closed-loop with the plant) when limited data are available. We show that under operational constraints, data generated by the proposed OED method based on maximizing information gain or maximizing variance provides much faster learning rate than uniform random sampling or pseudo random binary sampling. OED drastically reduces the duration of required functional tests by upto (50%), which, in practice, are permitted for only a few hours in a month due to operation constraints. (2) We exploit the variance in predictions from GPs to formulate a stochastic optimization problem to design an MPC controller to provide the desired load curtailment with perfect tracking and maximum 1.7% prediction error during a DR event. (3) Finally, we extend the OED approach to update the GP model as new data is generated by running the controller in a closed-loop with the building, reducing the repetitive need for functional tests as the system properties change with time.

While we can do functional tests more efficiently, perform closed-loop control with high confidence and update the model online with Gaussian Processes, our future work will focus on scaling the approach to even more complex systems like a network of buildings in a district.

References

[1] David Sturzenegger, Dimitrios Gyalistras, Manfred Morari, and Roy S Smith. Model predictive climate control of a swiss office building: Implementation, results, and cost–benefit analysis. *IEEE Transactions on Control Systems Technology*, 24(1):1–12, 2016.

[2] AF Emery and Aleksey V Nenarokomov. Optimal experiment design. *Measurement Science and Technology*, 9(6):864, 1998.

[3] Valerii Fedorov. Optimal experimental design. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):581–589, 2010.

[4] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[5] Achin Jain, Francesco Smarra, and Rahul Mangharam. Data Predictive Control using Regression Trees and Ensemble Learning. In *Proceedings of the 2017 Conference on Decision and Control*. IEEE, 2017.

[6] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[7] Felix Berkenkamp and Angela P Schoellig. Safe and robust learning control with gaussian processes. In *Control Conference (ECC), 2015 European*, pages 2496–2501. IEEE, 2015.

[8] Vishnu R Desaraju and Nathan Michael. Experience-driven predictive control. *Robot Learning and Planning (RLP 2016)*, page 29, 2016.

[9] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[10] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Machine Learning Research*, 9(Feb):235–284, 2008.

[11] Michael Deru, Kristin Field, Daniel Studer, Kyle Benne, Brent Griffith, Paul Torcellini, Bing Liu, Mark Halverson, Dave Winiarski, Michael Rosenberg, et al. US Department of Energy commercial reference building models of the national building stock, 2011.

[12] Madhur Behl, Truong X. Nghiem, and Rahul Mangharam. DR-Advisor: A data driven demand response recommender system. In *Proceedings of CISBAT*, 2015.

[13] Achin Jain, Madhur Behl, and Rahul Mangharam. Data Predictive Control for building energy management. In *Proceedings of the 2017 American Control Conference*. IEEE, 2017.

[14] Hae Young Noh and Ram Rajagopal. Data-driven forecasting algorithms for building energy consumption. In *Proc. of SPIE*, 2013.

[15] Rongxin Yin, Emre C. Kara, Yaping Li, et al. Quantifying flexibility of commercial and residential loads for demand response using setpoint changes. *Applied Energy*, 2016.

[16] Yuna Zhang, Zheng O'Neill, Bing Dong, and Godfried Augenbroe. Comparisons of inverse modeling approaches for predicting building energy performance. *Building and Environment*, 86:177–190, 2015.

[17] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[18] Juš Kocijan. *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

[19] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[20] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[21] Roman Garnett, Michael A Osborne, and Philipp Hennig. Active learning of linear embeddings for gaussian processes. *arXiv preprint arXiv:1310.6740*, 2013.

[22] A Wächter and L Biegler. Ipopt-an interior point optimizer, 2009.

[23] Joel Andersson. *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, KU Leuven, 2013.

[24] Truong X. Nghiem and Colin N. Jones. Data-driven demand response modeling and control of buildings with gaussian processes. In *American Control Conference (ACC)*, 2017.

[25] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11(Nov):3011–3015, 2010.