

Data Predictive Control using Regression Trees and Ensemble Learning

Achin Jain¹, Francesco Smarra², Rahul Mangharam¹

Abstract—Decisions on how to best operate large complex plants such as natural gas processing, oil refineries, and energy efficient buildings are becoming ever so complex that model-based predictive control (MPC) algorithms must play an important role. However, a key factor prohibiting the widespread adoption of MPC, is the cost, time, and effort associated with learning first-principles dynamical models of the underlying physical system. An alternative approach is to employ learning algorithms to build black-box models which rely only on real-time data from the sensors. Machine learning is widely used for regression and classification, but thus far data-driven models have not been used for closed-loop control. We present novel Data Predictive Control (DPC) algorithms that use Regression Trees and Random Forests for receding horizon control. We demonstrate the strength of our approach with a case study on a bilinear building model identified using real weather data and sensor measurements. In a one-to-one comparison, we show that DPC explains 70% variation in the MPC controller. We further apply DPC to a large scale multi-story EnergyPlus building model to curtail total power consumption in a Demand Response setting. In such cases, when the model-based controllers fail due to modeling cost, complexity and scalability, our results show that DPC curtails the desired power usage with high confidence.

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 and performance guarantees to plant operations. 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 challenge now, with using data-driven approaches, is to close the loop for real-time control and decision making.

Consider a multivariable dynamical system subject to external disturbances. The first and foremost requirement for making any 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 [17], [18].

There are two main reasons for model complexity. (1) The prime contributor is the change in model properties over time. Even if the model is identified once via an expensive route, 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. (2) A secondary reason is the model heterogeneity which 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 fuzzy logic rules that are based on best practices.

The 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 and data-driven approaches to control that are also adaptive, scalable and interpretable. We solve this problem with *Data Predictive Control (DPC)* by bridging the gap between Machine Learning and Predictive Control.

In our previous work [9], [10], we introduced the concept of DPC for receding horizon control. This work has the following contributions. (1) We first formally present two underlying algorithms: (i) DPC with regression trees, and (ii) DPC with random forests, which also ensure recursive feasibility in receding horizon control. (2) Using a bilinear building model whose parameters were identified using experiments on a building in Switzerland, we demonstrate the strength of DPC for receding horizon control via one-to-one comparison against a benchmark MPC controller. We show DPC captures 70% variance in MPC and offers a comparable performance. (3) We present a practical application of DPC for Demand Response, where we apply DPC to a 6 story 22 zone building model in EnergyPlus [3] for which model-based control is not economical and practical due to extreme complexity. We show scalability and efficiency of DPC in providing financial incentives to the end-customers bypassing the need for high fidelity models. We observe that DPC provides the desired power reduction with an average error of 3%.

II. DATA PREDICTIVE CONTROL

The central idea behind DPC is to obtain control-oriented models using machine learning or black-box modeling, and formulate the control problem in a way that receding horizon

¹Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA {achinj, rahulm}@seas.upenn.edu

²Department of Information Engineering, Computer Science and Mathematics, Center of Excellence DEWS, University of L'Aquila, L'Aquila 67100, Italy francesco.smarra@univaq.it

This work was supported partially by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and by the Italian Government under Cipe resolution n.135 (Dec. 21, 2012), project INnovating City Planning through Information and Communication Technologies (INCIPICT).

control (RHC) can still be applied and the optimization problem can be solved efficiently.

Consider a black-box model given by $x_{k+1} = f(x_k, u_k, d_k)$, where x, u, d represent states, inputs and disturbances, 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 [11]. These problems can be eliminated by decomposing

$$f(x_k, u_k, d_k) = g(d_k, x_k, h(u_k)), \quad (1)$$

where both g and h are learned using the data, and $h(u_k)$ is convex and differentiable, and thus suitable for optimization. DPC uses this functional decomposition or *separation of variables* to overcome the aforementioned challenges with black-box optimization.

A. Separation of Variables

We distinguish between two sets of variables: control (or manipulated) variables $X^c \in \mathbb{R}^c$ and disturbance (or non-manipulated) variables $X^d \in \mathbb{R}^d$. The union of the two sets forms the full feature set for training, i.e. $X \equiv X^c \cup X^d \in \mathbb{R}^{c+d}$. Our goal is to replace a model-based controller with a data-driven controller, where the latter depends only on the historical sensor data. These measurements could directly represent one or more states in the model-based control framework. We denote these as outputs $Y \in \mathbb{R}$ for training, i.e. a Y represents a particular output and we can have separate models for multiple outputs. We define the number of training samples by $|(\mathbf{X}, \mathbf{Y})| = n$.

Using separation of variables, the training process is divided into two steps. *Step 1*: The trees and the ensembles are trained only on X^d , which eases the computational complexity. It is important to note that besides external disturbances, X^d also contains autoregressive terms of the output Y which is the main reason for the state space explosion. *Step 2*: Linear regression models are trained in the leaves (or terminal nodes) of the trees which are function of only X^c . We have validated this linear model assumption in [10]. As we shall see in Sec. II-B and II-C, the second step reduces the run-time control problem into a convex program. This process is illustrated in Fig. 1.

B. DPC-RT: DPC with Regression Trees

When the data has lots of features, which interact in complicated, nonlinear ways, assembling a single global model such as linear or polynomial regression can be difficult, and can lead to poor response predictions. An approach to nonlinear regression is to partition the data space into smaller regions, where the interactions are more manageable. This partition is repeated recursively until finally we get to small chunks of the data space where we can fit simple (eg. linear parametric) models. Therefore, in (1), the global model f

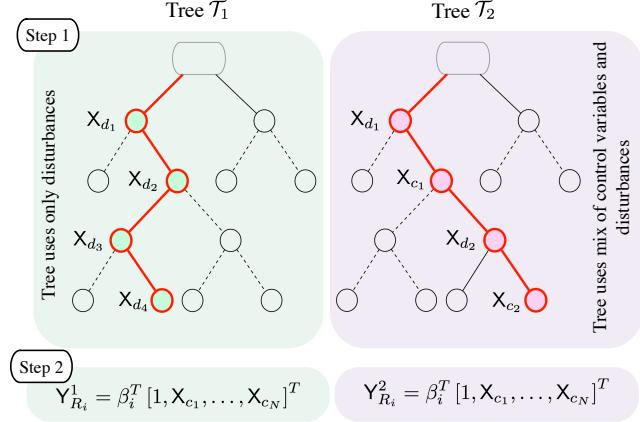


Fig. 1: Separation of variables. *Step 1*: Tree \mathcal{T}_1 is trained only on the disturbances X_d as the features. Tree \mathcal{T}_2 uses both the disturbances X_d and the control variables X_c for splitting and is thus not computationally suitable for control. *Step 2*: In the leaf R_i of the trees, a linear regression model parametrized by β_i is defined as a function only of the control variables.

has two parts: the recursive partition g , and a linear (and convex) model h for each cell of the partition.

Now, our goal is to predict the state Y at time k for next N time steps, i.e. $Y_{k+1|k}, \dots, Y_{k+N|k}$, where N is the control horizon. Applying the separation of variables, we build N regression trees using CART procedure [2] such that the output $Y_{k+j|k}$ of the j^{th} tree depends upon the previous N disturbances:

$$Y_{k+j|k} = f_{\text{tree}} \left(X_{k+j-N|k}^d, \dots, X_{k+j-1|k}^d \right), \quad (2)$$

$$X_{k+j-l|k}^d \in \mathbb{R}^d \quad \forall l, j = 1, \dots, N.$$

Then, the linear models as functions of X^c in each leaf of the tree \mathcal{T}_j are defined as

$$Y_{k+j|k} = \beta_j^T [1, X_{k|k}^c, \dots, X_{k+j-1|k}^c]^T, \quad (3)$$

$$X_{k+j-l|k}^c \in \mathbb{R}^c \quad \forall l, j = 1, \dots, N.$$

Note that the coefficients β_j would be different for each leaf. Eq. (3) implies that the prediction of output Y_{k+j} at time k is an affine combination of control inputs from time k to $k+j-1$. Thus, we have managed to linearize the original model dynamics via black-box modeling. This two-step training is done offline. In run-time, given the disturbances $X_{k|k}^d$ at time k , we can narrow down to a leaf of each tree in (2) to retrieve the linear models in (3).

In run-time, when a new control action is to be determined, each tree (prediction step) contributes to a linear constraint in the optimization as a replacement for the state dynamics in the case of MPC. Thus, the RHC optimization problem with a quadratic cost ($\mathcal{Q} \geq 0, \mathcal{R} \succeq 0$) can be formulated as:

$$\begin{aligned} \min & \sum_{j=1}^N (Y_{k+j|k})^2 \mathcal{Q} + X_{k+j-1|k}^c \mathcal{R} X_{k+j-1|k}^c + \lambda \epsilon_j \\ \text{s. t. } & Y_{k+j|k} = \beta_j^T [1, X_{k|k}^c, \dots, X_{k+j-1|k}^c]^T \\ & X^c \leq X_{k+j-1|k}^c \leq \bar{X}^c \\ & \underline{Y} - \epsilon_j \leq Y_{k+j|k} \leq \bar{Y} + \epsilon_j \\ & \epsilon_j \geq 0, \quad j = 1, \dots, N. \end{aligned} \quad (4)$$

Here, $\mathcal{Q} \in \mathbb{R}$ and $\mathcal{R} \in \mathbb{R}^{c \times c}$, and the slack variables ϵ_j ensure recursive feasibility since the equality constraint on \mathbf{Y} is relaxed. Of course, a different cost function can be chosen depending upon the application. In the current formulation, the data-driven control problem is reduced to a convex program which is much easier to solve than running an optimization directly on a black-box model trained on \mathbf{X}^c as features. We solve this optimization in the same manner as MPC to determine the optimal sequence of inputs $[\mathbf{X}_{k|k}^c, \dots, \mathbf{X}_{k+N-1|k}^c]$, apply the first control input $\mathbf{X}_{k|k}^c$ and proceed to the next time step $k + 1$. The pseudo code for DPC-RT is given in Alg. 1.

C. DPC-En: DPC with Ensemble Methods

Regression trees obtain good predictive accuracy in many domains. However, the models used in their leaves have some limitations regarding the kind of functions they are able to approximate. The problem with trees is their high variance and that they can overfit the data easily. A small change δ in the data can result in a different series of splits and thus violate the acceptable accuracy ε , i.e. $\exists \hat{\mathbf{X}}^d \mid \|\mathbf{X}^d - \hat{\mathbf{X}}^d\| < \delta$ & $\|\mathbf{Y} - \mathbf{Y}_{\text{true}}\| > \varepsilon$. This is the price to be paid for estimating a tree-based structure from the data.

We use ensemble methods [6] to combine the predictions of several independent regression trees in order to improve generalizability and robustness over a single estimator. The essential idea is to average many noisy trees to reduce the overall variance in prediction. We inject randomness into the tree construction in two ways. First, we randomize the features used to define splitting in each tree. Second, we build each tree using a bootstrapped or sub-sampled data set. In this way, each tree in the forest is trained on different data, which introduces differences between the trees. More explicitly, training features $\mathbf{X}^d \in \mathbb{R}^p$ with $p < d$ and the in-bag samples (in-bag samples correspond to the data samples on which the tree was trained) are different for each tree in the forest i.e $|(\mathbf{X}, \mathbf{Y})| < n$.

The goal with DPC-En is to replace each tree in Alg. 1 by a forest

$$\begin{aligned} \mathbf{Y}_{k+j|k} &= f_{\text{forest}} \left(\mathbf{X}_{k+j-N|k}^d, \dots, \mathbf{X}_{k+j-1|k}^d \right), \quad (5) \\ \mathbf{X}_{k+j-l|k}^d &\in \mathbb{R}^p \quad \forall l, j = 1, \dots, N, \end{aligned}$$

which, again, is trained only on \mathbf{X}^d , but $\mathbf{X}^d \in \mathbb{R}^p \subset \mathbb{R}^d$ for each tree, and then fit a linear regression model using \mathbf{X}^c in every leaf of every tree. We build N such forests for N prediction steps such that the leaf R_i of forest \mathcal{R}_j uses a linear model

$$\begin{aligned} \mathbf{Y}_{k+j|k} &= \Theta_{ij}^T [1, \mathbf{X}_{k|k}^c, \dots, \mathbf{X}_{k+j-1|k}^c]^T, \quad (6) \\ \mathbf{X}_{k+j-l|k}^c &\in \mathbb{R}^c \quad \forall l, j = 1, \dots, N. \end{aligned}$$

Here $(\mathbf{X}^c, \mathbf{Y})$ correspond to the in-bag samples for the trees.

While the offline training burden in DPC-En is slightly increased compared to DPC-RT, in the control step we exploit the better accuracy, and lower variance properties of the random forest. If a forest has t number of trees, given the forecast of disturbances, we have t sets of linear coefficients. We simply average out all the coefficients from all the trees to get one linear model represented by $\bar{\Theta}_j$ for each forest. Note

Algorithm 1 Data Predictive Control with Regression Trees

```

1: DESIGN TIME
2: procedure MODEL TRAINING USING SEPARATION OF VARS
3:   Set  $\mathbf{X}^c \leftarrow$  manipulated features
4:   Set  $\mathbf{X}^d \leftarrow$  non-manipulated features
5:   Build  $N$  predictive trees with  $(\mathbf{Y}, \mathbf{X}^d)$  defined in (2)
6:   for all trees  $\mathcal{T}_j$  do
7:     for all regions  $R_i$  at the leaves of  $\mathcal{T}_j$  do
8:       Fit  $\mathbf{Y}_{k+j|k} = \beta_j^T [1, \mathbf{X}_{k|k}^c, \dots, \mathbf{X}_{k+j-1|k}^c]^T$  as in (3)
9:     end for
10:   end for
11: end procedure
12: RUN TIME
13: procedure PREDICTIVE CONTROL
14:   while  $k < k_{\text{stop}}$  do
15:     for all trees  $\mathcal{T}_j$  do
16:       Determine the leaf  $R_i$  using  $\mathbf{X}^d$  as in (2)
17:       Obtain the linear model at  $R_i$  trained in (3)
18:     end for
19:     Solve optimization in (4) to determine optimal
20:     control actions  $[\mathbf{X}_{k|k}^c, \dots, \mathbf{X}_{k+N-1|k}^c]$ 
21:     Apply the first input  $\mathbf{X}_{k|k}^c$ 
22:   end while
23: end procedure

```

that the averaging step can only be done in run-time because the leaf of each tree can be narrowed down only when the \mathbf{X}^d is known. Thus, for N forests, we again have exactly N linear equality constraints in the optimization problem below:

$$\begin{aligned} \min \sum_{j=1}^N (\mathbf{Y}_{k+j|k})^2 \mathcal{Q} + \mathbf{X}_{k+j-1|k}^c \mathcal{R} \mathbf{X}_{k+j-1|k}^c + \lambda \epsilon_j \\ \text{s. t.} \quad \mathbf{Y}_{k+j|k} = \hat{\Theta}_j^T [1, \mathbf{X}_{k|k}^c, \dots, \mathbf{X}_{k+j-1|k}^c]^T \quad (7) \\ \underline{\mathbf{X}}^c \leq \mathbf{X}_{k+j-1|k}^c \leq \bar{\mathbf{X}}^c \\ \underline{\mathbf{Y}} - \epsilon_j \leq \mathbf{Y}_{k+j|k} \leq \bar{\mathbf{Y}} + \epsilon_j \\ \epsilon_j \geq 0, \quad j = 1, \dots, N. \end{aligned}$$

DPC-En is graphically described in Fig. 2. The ensemble data predictive control (DPC-En) is the first such method to bridge the gap between ensemble predictive models (such as random forests) and receding horizon control. In the next section, we compare DPC-RT and DPC-En to MPC for a building model.

III. COMPARISON WITH MPC

We consider a bilinear building model developed at Automatic Control Laboratory, ETH Zurich. It captures the essential dynamics governing the zone-level operation while considering the external and the internal thermal disturbances. By Swiss standards, the model used for this study is of a heavyweight construction with a high window area fraction on one facade and high internal gains due to occupancy and equipments [7].

The bilinear model is a standard building model used for practical considerations [12], [15], [16] as it is detailed enough and suitable for model-based control unlike the ones obtained from simulation software like EnergyPlus. We specifically consider this model to show a comparison against MPC. MPC of EnergyPlus models can be cost and time prohibitive, making them unsuitable for control. In Sec. IV, we show how DPC scales easily to such large scale models.

A. Bilinear Model

The bilinear model has 12 internal states including the inside zone temperature T_{in} , the slab temperatures T_{sb} , the inner wall T_{iw} and the outside wall temperature T_{ow} . The state vector is defined as $x := [T_{in}, T_{sb}^{(1:5)}, T_{ef}^{(1:3)}, T_{in}^{(1:3)}]^T$.

There are 4 control inputs including the blind position B , the gains due to electric lighting L , the evaporative cooling usage factor C , and the heat from the radiator H such that $u := [B, L, H, C]^T$. B and L affect both room illuminance and temperature due to heat transfer whereas C and H affect only temperature.

The model is subject to 5 weather disturbances: solar gains with fully closed blinds Q_{sc} and with open blinds Q_{so} , daylight illuminance with open blinds I_o , external dry-bulb temperature T_{db} and external wet-bulb temperature T_{wb} . The hourly weather forecast, provided by MeteoSwiss, was updated every 12 hrs. Therefore, to improve the forecast, an autoregressive model of the uncertainty was considered. Other disturbances come from the internal gains due to occupancy Q_{io} and due to equipments Q_{ie} which were assumed as per the Swiss standards [14]. We define $d := [Q_{sc}, Q_{so}, I_o, Q_{io}, Q_{ie}, T_{db}, T_{wb}]^T$. For further details, we refer the reader to [15].

The model dynamics is given below. The bilinearity is present in both input-state, and input-disturbance.

$$\begin{aligned} x_{k+1} &= Ax_k + (B_u + B_{xu}[x_k] + B_{du}[d_k])u_k + B_dd_k \quad (8) \\ x_k &\in \mathbb{R}^{12}, u_k \in \mathbb{R}^4, d_k \in \mathbb{R}^8 \quad \forall k = 0, \dots, T, \end{aligned}$$

where, the matrices B_{xu} and B_{du} are defined as

$$\begin{aligned} B_{xu}[x_k] &= [B_{xu,1}[x_k], B_{xu,2}[x_k], \dots, B_{xu,4}[x_k]] \in \mathbb{R}^{12 \times 4}, \\ B_{du}[d_k] &= [B_{du,1}[d_k], B_{du,2}[d_k], \dots, B_{du,4}[d_k]] \in \mathbb{R}^{12 \times 4}, \\ B_{xu,i} &\in \mathbb{R}^{12 \times 12}, B_{du,i} \in \mathbb{R}^{12 \times 8} \quad \forall i = 1, 2, 3, 4. \end{aligned}$$

For this study, we assume that the disturbances are precisely known to MPC as well as DPC controller. In our future work, we will account for the uncertainties in the disturbances with an extension to Scenario approach [1] for DPC.

B. Model Predictive Control

We use an MPC controller with a quadratic and a linear cost for comparison. The finite RHC approach involves optimizing a cost function subject to the dynamics of the system and the constraints, over a finite horizon of time [13]. After an optimal sequence of control inputs are computed, the first input is applied, then at the next step the optimization is solved again.

The objective of the controller is to minimize the energy usage $c^T u$ while maintaining a desired level of thermal comfort. Therefore, at time step k , we solve a continuously linearized MPC problem to determine the optimal sequence

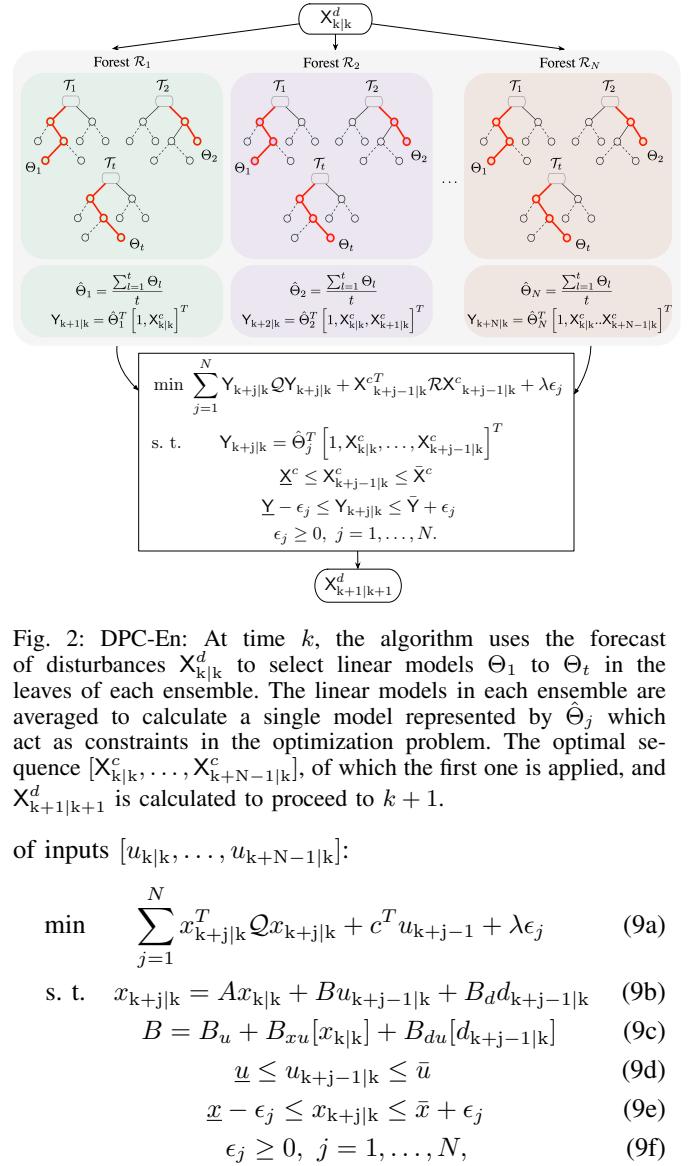


Fig. 2: DPC-En: At time k , the algorithm uses the forecast of disturbances $X_{k|k}^d$ to select linear models Θ_1 to Θ_t in the leaves of each ensemble. The linear models in each ensemble are averaged to calculate a single model represented by $\hat{\Theta}_j$ which act as constraints in the optimization problem. The optimal sequence $[X_{k|k}^c, \dots, X_{k+N-1|k}^c]$, of which the first one is applied, and $X_{k+1|k+1}^d$ is calculated to proceed to $k+1$.

of inputs $[u_{k|k}, \dots, u_{k+N-1|k}]$:

$$\min \sum_{j=1}^N x_{k+j|k}^T Q x_{k+j|k} + c^T u_{k+j-1} + \lambda \epsilon_j \quad (9a)$$

$$\text{s. t. } x_{k+j|k} = Ax_{k|k} + Bu_{k+j-1|k} + B_dd_{k+j-1|k} \quad (9b)$$

$$B = B_u + B_{xu}[x_{k|k}] + B_{du}[d_{k+j-1|k}] \quad (9c)$$

$$\underline{u} \leq u_{k+j-1|k} \leq \bar{u} \quad (9d)$$

$$\underline{x} - \epsilon_j \leq x_{k+j|k} \leq \bar{x} + \epsilon_j \quad (9e)$$

$$\epsilon_j \geq 0, j = 1, \dots, N, \quad (9f)$$

where $Q \in \mathbb{R}^{12 \times 12}$ has all zeros except at $Q^{(1,1)}$ corresponding to the zone temperature, $c \in \mathbb{R}^4$ is proportional to cost of using each actuator and λ penalizes the slack variables.

C. Data Predictive Control

In this section, we explain how DPC can be applied to this case study. We begin with a description of features X and output Y used for training.

1) *Training Data:* The fundamental reason why DPC is suitable for such a problem is that when the complexity rises, there is a huge cost to model all the states given by the dynamical system (8). For example, states in the bilinear model also include slab temperatures which require modeling of structural and material properties in detail and often we also need to install new sensors to capture additional states. Thus, DPC is based solely on one state of the model i.e. the zone temperature that can be easily measured with a thermostat. This serves as the output variable Y of interest for which we build N trees and N forests as described in Sec. II-B and II-C, respectively. Therefore, $Y_{k+j|k} := x_{k+j|k}^1$,

TABLE I: Quantitative comparison of root mean square error (RMSE), R^2 score, and explained variance (EV) for trees and forests for different predictions steps.

	RMSE	R^2 score	EV
tree- $\mathbf{Y}_{k+1 k}$	0.42	0.75	0.76
tree- $\mathbf{Y}_{k+6 k}$	0.64	0.41	0.42
forest- $\mathbf{Y}_{k+1 k}$	0.29	0.87	0.88
forest- $\mathbf{Y}_{k+6 k}$	0.38	0.78	0.80

where x^1 is the first component of x . Next, we define the non-manipulated features $\mathbf{X}_{k|k}^d$. At time k , for the tree \mathcal{T}_j and the forest \mathcal{R}_j , we base these features to include weather disturbances, external disturbances due to occupancy and equipments, and autoregressive terms of the room temperature, i.e. $\mathbf{X}_{k|k}^d := [d_{k+j-N|k}, \dots, d_{k+j-1|k}, x_{k|k}^1, \dots, x_{k-\delta|k}^1]$, where δ is the order of autoregression. Finally, the inputs in DPC are exactly same as in MPC. i.e. $\mathbf{X}_{k+j-1|k}^c := u_{k+j-1|k}$.

The training data in the above format was generated by simulating the bilinear model with rule-based strategies for 10 months in 2007. January and May were deliberately excluded for testing the DPC implementation.

2) *Optimization*: For a fair comparison with MPC, we cast DPC optimization problem as follows:

$$\min \sum_{j=1}^N \mathbf{Y}_{k+j|k} Q^{(1,1)} \mathbf{Y}_{k+j|k} + c^T \mathbf{X}_{k+j-1|k}^c + \lambda \epsilon_j \quad (10a)$$

$$\text{s. t. } \mathbf{Y}_{k+j|k} = \alpha_j^T [1, \mathbf{X}_{k|k}^c, \dots, \mathbf{X}_{k+j-1|k}^c]^T \quad (10b)$$

$$\underline{\mathbf{X}} \leq \mathbf{X}_{k+j-1|k}^c \leq \bar{\mathbf{X}} \quad (10c)$$

$$\underline{\mathbf{Y}} - \epsilon_j \leq \mathbf{Y}_{k+j|k} \leq \bar{\mathbf{Y}} + \epsilon_j \quad (10d)$$

$$\epsilon_j \geq 0, j = 1, \dots, N. \quad (10e)$$

Here $\alpha = \beta$ for DPC-RT and $\alpha = \hat{\Theta}$ for DPC-En. Note that, (10) is DPC analog of (9). The only difference is the state dynamics (9b) and (9c) are now replaced with (10b).

3) *Validation*: We compare the prediction for the first time step $\mathbf{Y}_{k+1|k}$ and the 6-hour ahead prediction $\mathbf{Y}_{k+6|k}$ for a week in the month of May in Fig. 3. It is visible how trees have a high variance, and the forests are more accurate. Note that data from January and May was not used for training. The quantitative summary of the accuracy is given in Tab. I. We can see that the random forests are better in all respects.

D. Comparison

We compare the performance of DPC (10) against an equivalent MPC formulation (9). The solution obtained from MPC sets the benchmark that we compare to. Note that the MPC implementation uses the exact knowledge of the plant dynamics. Therefore, the associated control strategy is indeed the optimal strategy for the plant.

The performance is compared for 3 days in winter, i.e. January 28-31 and 3 days in summer, i.e. May 1-3. These are shown on the same plots in Fig. 4. The sampling time in the simulations is 1 hr. The control horizon N and the order of autoregression are both 6 hrs. The training procedure required a few minutes in the case of trees and 2 hrs for forests on a Win 10 machine with an i7 processor and 8GB memory. The cooling usage factor C is constrained in $[0, 1]$,

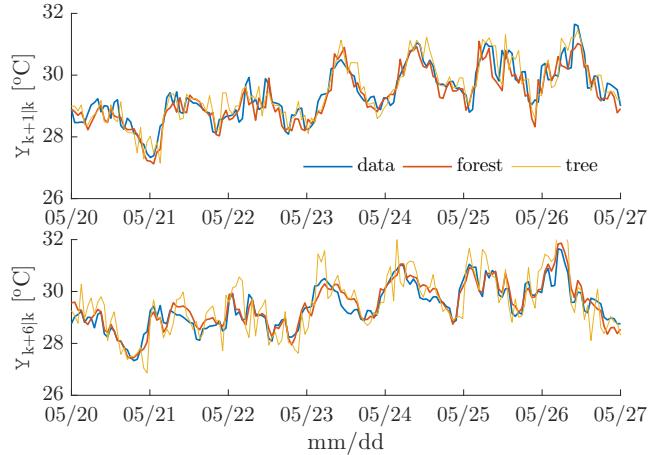


Fig. 3: Temperature predictions from a tree and a forest for first step prediction (top) and the 6-hour ahead prediction (bottom). Ensemble method shows a relatively higher accuracy.

the heat input in $[0, 23]$ W/m 2 , and the room temperature in $[19, 25]$ °C during the winters and $[20, 26]$ °C during the summers. The optimization is solved using CPLEX [8].

The external disturbances - solar gains, internal gain due to equipment and dry-bulb temperature during the chosen periods are shown in Fig. 4(a). The internal gain due to occupancy was proportional to the gain due to equipment. The reference temperature is chosen to be 22 °C. Due to cold weather, which is evident from the dry-bulb temperature, the heater is switched on during the night to maintain the thermal comfort requirements. When the building is occupied during the day, due to excessive internal gains, the building requires cooling. The lighting in the building is adjusted to meet the minimum light requirements. The optimal cooling usage factor and the radiator power for MPC, DPC-En and DPC-RT are shown in Fig. 4(b) and Fig. 4(c), respectively. The control strategy with DPC-En shows a remarkable similarity to MPC, switching on/off the equipments at the same time with similar usage. However, the performance with DPC-RT is much different and worse. DPC-RT inherently suffers from high variance which is also evident in the control strategy, thus making it unsuitable for practical purposes. Although it seems like that adding the rate constraints to DPC-En would smoothen its behavior, this was avoided because the sampling time of the system is 1 hr which is already too high. The room temperature profile in Fig. 4(d) is close to the reference in the case of DPC-En as well as MPC. Fig. 4(e) shows that the cumulative cost of the objective function is, as expected, minimum for MPC, and a bit higher for DPC-En. The cost for DPC-RT blows up around 12 noon on 30th January as one of the slack variables is non-zero, which happens due to high model inaccuracy.

The quantitative performance comparison is shown in Tab. II. MPC tracks the reference more closely at the expense of higher input costs in comparison to DPC-En. The higher cost of the inputs in MPC is also due to lighting. DPC-En explains 70.1% variation in the optimal control strategies obtained from MPC while DPC-RT explains only 1.8%. The mean optimal cost of DPC-En is more than MPC, and is maximum for DPC-RT due to a constraint violation.

Thus, we have shown that DPC-En provides a comparable

TABLE II: Quantitative comparison of explained variance, mean value of objective function, mean input cost $c^T u$ and mean deviance from the reference temperature $|T - T_{ref}|$.

	explained variance [-]	mean objective value [-]	mean input cost [-]	mean deviance [°C]
MPC	—	22.60	17.16	0.26
DPC-En	70.1%	39.26	15.12	0.48
DPC-RT	1.8%	204.55	16.84	0.57

performance to MPC without using the physical model. However, one major limitation of the bilinear model is that the information about the building power consumption is not available. Much nonlinearities in the system are due to equipment efficiencies which are not considered in the bilinear case but are very important for practical purposes.

Therefore, our next goal is to apply DPC-En on even more complex and realistic EnergyPlus model for which building a model predictive controller is time and cost prohibitive [17]. This is because we would need to model intricate details like the geometry and construction layouts, the equipment design and layout plans, material properties, equipment and operational schedules etc.

IV. APPLICATION: DEMAND RESPONSE

In January 2014, the east coast (PJM) electricity grid experienced an 86x increase in the price of electricity from \$31/MWh to \$2,680/MWh in a matter of 10 minutes. Similarly, the price spiked 32x 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*. The solution lies with Demand Response (DR) from the customer side - curtailing demand during peak capacity for financial incentives. However, this is a very hard problem for commercial, industrial and institutional plants, the largest electricity consumers.

Thus, the problem of energy management during a DR event makes an ideal case for DPC. In the following sections, we apply DPC-En to a large scale EnergyPlus model to show how effectively DPC can provide a desired power curtailment as well as a desired thermal comfort. DPC builds predictive models of a building based on historical weather, schedule, set-points and electricity consumption data, while also learning from the actions of the building operator. These models are then used for synthesizing recommendations about the control actions that the operator needs to take, during a DR event, to obtain a given load curtailment while providing guarantees on occupant comfort and operations.

A. EnergyPlus Model

We use the DoE Commercial Reference Building (DoE CRB) simulated in EnergyPlus [5] as the virtual test-bed building. This is a large 6 story hotel building consisting of 22 zones with a total area of 122,120 sq.ft. During peak load conditions the building can consume up to 400 kW of power. For the simulation of the DoE CRB building we use actual meteorological year data from Chicago for the years 2012 and 2013.

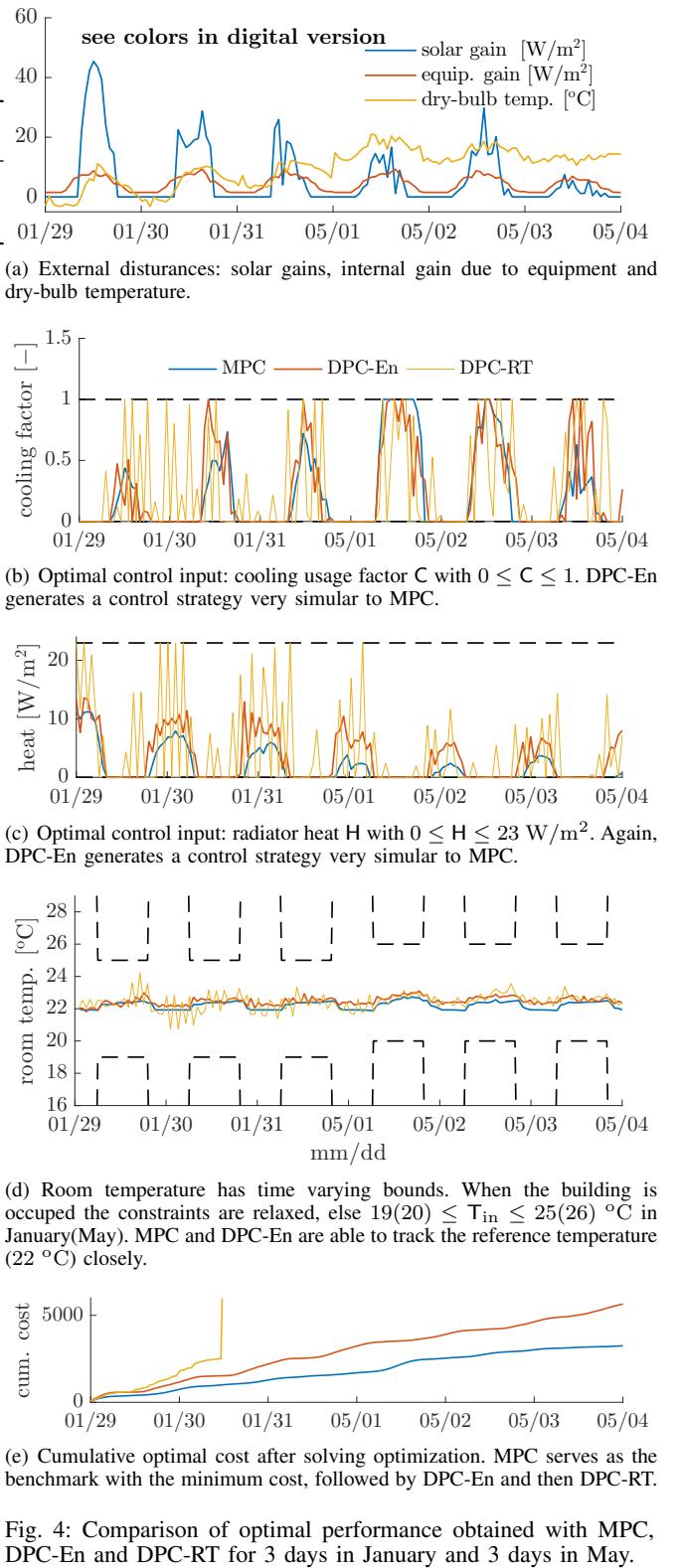


Fig. 4: Comparison of optimal performance obtained with MPC, DPC-En and DPC-RT for 3 days in January and 3 days in May.

B. Model training for DPC

In the following simulations, we consider a long DR event from 7am - 2pm when the end-users are expected to follow/track the reference power signal sent by the utility. This is indeed common in Demand Tracking Control. During offline training, we sample data every 15 min to learn 2 kinds of forests. (1) Power forests are built using output as

the total building power consumption, and (2) Temperature forests with output as temperature of one of the 22 zones.

The training data set contains the following types of features. (1) The *weather data* which includes measurements of the outside air temperature and relative humidity. Since we are interested in predicting the power consumption or the zone temperature for a finite horizon, we include the weather forecast of the complete horizon in the training features. (2) The *schedule data* includes the proxy variables which correlate with repeated patterns of electricity consumption e.g., due to occupancy or equipment schedules. Day of Week is a categorical predictor which takes values from 1-7 depending on the day of the week. This variable can capture any power consumption patterns which occur on specific days of the week. Likewise, Time of Day is quite an important predictor of power consumption as it can adequately capture daily patterns of occupancy, lighting and appliance use without directly measuring any one of them. Besides using proxy schedule predictors, actual building equipment schedules can also be used as training data for building the trees. (3) The *building data* include (i) cooling set points for the guest rooms, kitchen and corridors, (ii) supply air temperature, and (iii) chilled water temperature.

For the following simulations, we use five control variables (i) cooling set point for corridors ClgSP, (ii) cooling set point for guest rooms GuestSP, (iii) cooling set point for kitchen KitchenSP, (iv) chilled water supply temperature ChwSP, and (v) supply air temperature SupplyAirSP, so $X^c = [ClgSP, GuestClgSP, KitchenClgSP, SupplyAirSP, ChwSP]$. The power forest \mathcal{R}_p is built using the total building power consumption P. Its features X^d include the weather variables, their lag terms and their forecast over the horizon, the schedule variables, and finally the lag terms of the power consumption. The temperature forest \mathcal{R}_t is built with zone temperature T as the output. Except for the lag terms corresponding to the same zone temperature, all other features are same in X^d .

Fig. 5 shows the prediction accuracy for the power forest, and also explains the two level training approach introduced in Sec. II-A. During S1, the forests are trained using only disturbances as the features. Then in S2, the local effects of the control variables are accounted for by the linear models in the leaves. We observe how the accuracy is drastically improved after including the linear models in the predictions.

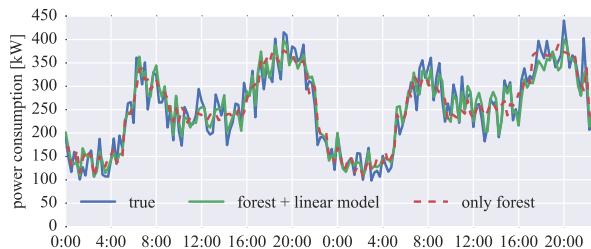
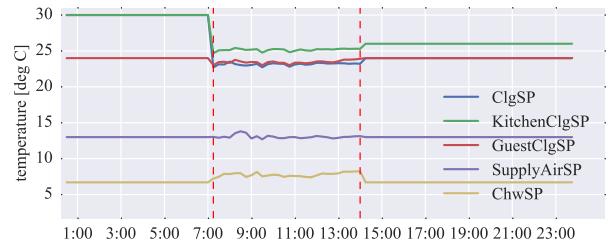
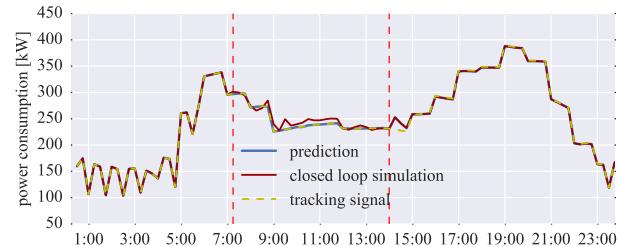


Fig. 5: Model accuracy during training: The prediction made by forest using only X^d (red) captures the effect due to disturbances. The linear models in the leaves capture the local effects (green) due to the control inputs X^c and improve the model accuracy.



(a) Optimal inputs calculated by DPC-En. At first, the inputs are changed rapidly because of a significant difference between the desired and the actual power consumption. Then gradual adjustments are made to follow the desired reference.



(b) Power tracking by DPC-En at 1.1 MW: The difference in closed-loop simulation and prediction is due to model mismatch.

Fig. 6: Power management using DPC. The controller is active between 7am - 2pm. This region is marked in dashed red lines.

C. Power Management

Typically, the end customer receives a notification to curtail the power by some fraction. In this example on power management, we show how DPC can generate optimal inputs to track a desired power signal within a small allowance while maintaining the zone level thermal comfort. It may not be possible to have the same thermal comfort level in all the zones due to power curtailment, so we choose one zone (for example CEO's office) where the constraints must be met. This is done by solving the following optimization problem with control variables $X^c = [ClgSP, GuestClgSP, KitchenClgSP, SupplyAirSP, ChwSP]$ as defined before:

$$\begin{aligned} \min & \sum_{j=1}^N (P_{k+j|k} - P_{ref})^2 + \lambda \epsilon_j + \nu \delta_j \\ \text{s. t. } & P_{k+j|k} = \hat{\Theta}_{P_j}^T [1, X_{k|k}^c, \dots, X_{k+j-1|k}^c]^T \\ & T_{k+j|k} = \hat{\Theta}_{T_j}^T [1, X_{k|k}^c, \dots, X_{k+j-1|k}^c]^T \\ & P - \epsilon_j \leq P_{k+j|k} \leq \bar{P} + \epsilon_j \\ & T - \delta_j \leq T_{k+j|k} \leq \bar{T} + \delta_j \\ & X^c \leq X_{k+j-1|k}^c \leq \bar{X}^c \\ & \epsilon_j \geq 0, \delta_j \geq 0, j = 1, \dots, N. \end{aligned} \quad (11)$$

Here, the temperature forests are used to enforce thermal constraints in the zone of interest. The setup of optimization problem is flexible to include even other variables in the cost or the constraints. For example, we are currently looking at including the dynamic pricing of electricity in the cost since the customers can more directly relate to the financial incentives.

The results are shown in Fig. 6. The DPC controller is active between 7am - 2pm. Before 7am and after 2pm, the

building is using a predefined rule-based control strategy. The optimal control inputs from DPC-En are shown in Fig. 6(a). It is observed that, with the optimal inputs generated by DPC, we can track the reference power consumption signal closely. In fact, the average tracking error between 7am - 2pm is 3%. The difference between the predicted power consumption and that in the closed-loop simulation in Fig. 6(b) is due to model mismatch between the EnergyPlus model and the power forest used in the optimization (11). Due to this inaccuracy, the actual power consumption is on an average 7 kW higher.

Thus, DPC-En successfully tracks a given power reference signal with an average $\sim 3\%$ error for such a complex building which would require several years of efforts to develop a physics based model.

D. Practical Challenges and Future Work

Data Availability: The main practical challenge for DPC lies in the availability of data for training and we require answers to questions like how much data (functional testing) is required, and how should the sampling be done? Therefore, the procedure for optimal experiment design, and model improvement with estimation of variance in predictions is one of the main focus of our ongoing work.

Stability: While the buildings are inherently stable, many other applications require stability guarantees. In our ongoing work, we are working towards proving asymptotic stability to origin with DPC-RT and DPC-En by using concept of switched LTI systems. This will make DPC useful for systems with faster dynamics.

Robustness: Another direction of work is on handling uncertainties in the DPC framework, namely an extension to Scenario DPC to account for the disturbance uncertainty. This will help us in quantifying the robustness of DPC.

V. CONCLUSION

We present two algorithms based on trees and random forests for receding horizon control with data-driven models. We compare the performance of our Data Predictive Control to MPC on a multivariable bilinear building model. We establish that DPC with random forests shows a remarkable similarity to MPC in the optimal control strategies explaining 70% variance. On the other hand, DPC with regression trees suffers from practical limitations due to model overfitting. We further apply DPC with random forests to a large scale 6 story EnergyPlus model with 22 zones for which the traditional model-based control is largely unsuitable due to complex dynamics and the cost of model identification. We show that DPC, relying only on the sensor data, can provide significant energy savings while maintaining thermal comfort. Our results demonstrate that even for such complex system, DPC tracks a reference signal with a mean error of 3%.

DPC has applications which go beyond buildings and energy systems, to industrial process control, and controlling large critical infrastructures like water networks, district heating & cooling. DPC is immensely valuable in situations where first principles based modeling cost is extremely high.

ACKNOWLEDGMENT

The authors would like to thank Xiaojing Zhang, a Post-Doctoral Researcher at the University of California, Berkeley for providing the building model, and Manfred Morari for his feedback on DPC.

REFERENCES

- [1] D. Bernardini and A. Bemporad. Scenario-based model predictive control of stochastic constrained linear systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 6333–6338. IEEE, 2009.
- [2] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
- [3] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, et al. Energyplus: Creating a new-generation building energy simulation program. *Energy and buildings*, 33(4):319–331, 2001.
- [4] D. Davis. Lighting the Way to Demand Response Lighting the Way to Demand Response. Technical report, CEC, 2011.
- [5] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, M. Rosenberg, et al. Us department of energy commercial reference building models of the national building stock. 2011.
- [6] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.
- [7] D. Gyalistras and M. Gwerder. Use of weather and occupancy forecasts for optimal building climate control (opticontrol): Two years progress report main report. *Terrestrial Systems Ecology ETH Zurich R&D HVAC Products, Building Technologies Division, Siemens Switzerland Ltd, Zug, Switzerland*, 2010.
- [8] I. ILOG. IBM ILOG CPLEX Optimizer-Highperformance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming, 2012.
- [9] A. Jain, M. Behl, and R. Mangharam. Data Predictive Control for building energy management. In *Proceedings of the 2017 American Control Conference*. IEEE, 2017.
- [10] A. Jain, R. Mangharam, and M. Behl. Data Predictive Control for peak power reduction. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pages 109–118. ACM, 2016.
- [11] A. Kusiak, Z. Song, and H. Zheng. Anticipatory control of wind turbines with data-driven predictive models. *IEEE Transactions on Energy Conversion*, 24(3):766–774, 2009.
- [12] Y. Ma, J. Matuško, and F. Borrelli. Stochastic Model Predictive Control for building hvac systems: Complexity and conservatism. *IEEE Transactions on Control Systems Technology*, 23(1):101–116, 2015.
- [13] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [14] S. Merkblatt. 2024: Standard-nutzungsbedingungen für die energie- und gebäudetechnik. Zürich: Swiss Society of Engineers and Architects, 2006.
- [15] F. Oldewurtel. *Stochastic Model Predictive Control for energy efficient building climate control*. PhD thesis, ETH Zurich, 2011.
- [16] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari. Use of Model Predictive Control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45:15–27, 2012.
- [17] D. Sturzenegger, D. Gyalistras, M. Morari, and R. 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.
- [18] E. Žáčeková, Z. Váňa, and J. Cigler. Towards the real-life implementation of MPC for an office building: Identification issues. *Applied Energy*, 135:53–62, 2014.