Applied Energy 226 (2018) 1252–1272

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Data-driven model predictive control using random forests for building energy optimization and climate control

Francesco Smarra[a,b,*,1], Achin Jain[b,1], Tullio de Rubeis[c,1], Dario Ambrosini[c], Alessandro D'Innocenzo[a], Rahul Mangharam[b]

[a] Department of Information Engineering, Computer Science and Mathematics, Università degli Studi dell'Aquila, L'Aquila, Italy
[b] Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, USA
[c] Department of Industrial and Information Engineering and Economics, Università degli Studi dell'Aquila, L'Aquila, Italy

HIGHLIGHTS

- A novel approach to data-driven predictive control (DPC) using Random Forests.
- Accuracy, scalability & robustness of the algorithm are verified with three studies.
- Case Study I: DPC shows comparable performance to a physics-based MPC controller.
- Case Study II: DPC provides Demand Response curtailment for an EnergyPlus building.
- Case Study III: DPC provides up to 50% energy savings in a real off-grid house.

ARTICLE INFO

Keywords:
Building control
Energy optimization
Demand response
Machine learning
Random forests
Receding horizon control

ABSTRACT

Model Predictive Control (MPC) is a model-based technique widely and successfully used over the past years to improve control systems performance. A key factor prohibiting the widespread adoption of MPC for complex systems such as buildings is related to the difficulties (cost, time and effort) associated with the identification of a predictive model of a building. To overcome this problem, we introduce a novel idea for predictive control based on historical building data leveraging machine learning algorithms like regression trees and random forests. We call this approach Data-driven model Predictive Control (DPC), and we apply it to three different case studies to demonstrate its *performance*, *scalability* and *robustness*. In the first case study we consider a benchmark MPC controller using a bilinear building model, then we apply DPC to a data-set simulated from such bilinear model and derive a controller based only on the data. Our results demonstrate that DPC can provide comparable performance with respect to MPC applied to a perfectly known mathematical model. In the second case study we apply DPC to a 6 story 22 zone building model in EnergyPlus, for which model-based control is not economical and practical due to extreme complexity, and address a Demand Response problem. Our results demonstrate scalability and efficiency of DPC showing that DPC provides the desired power curtailment with an average error of 3%. In the third case study we implement and test DPC on real data from an off-grid house located in L'Aquila, Italy. We compare the total amount of energy saved with respect to the classical bang-bang controller, showing that we can perform an energy saving up to 49.2%. Our results demonstrate robustness of our method to uncertainties both in real data acquisition and weather forecast.

1. Introduction

Control-oriented models of energy system's dynamics and energy consumption, are needed for understanding and improving the overall energy efficiency and operating costs of a building. With a reasonably accurate forecast of future weather and building operating conditions, dynamical models can be used to predict the energy needs of the building over a prediction horizon, and use them to determine optimal

[*] Corresponding author at: Department of Information Engineering, Computer Science and Mathematics, Università degli Studi dell'Aquila, L'Aquila, Italy.
E-mail addresses: francesco.smarra@univaq.it, fsmarra@seas.upenn.edu (F. Smarra), achinj@seas.upenn.edu (A. Jain), tullio.derubeis@graduate.univaq.it (T. de Rubeis), dario.ambrosini@univaq.it (D. Ambrosini), alessandro.dinnocenzo@univaq.it (A. D'Innocenzo), rahulm@seas.upenn.edu (R. Mangharam).
[1] Equal contribution.

https://doi.org/10.1016/j.apenergy.2018.02.126
Received 30 September 2017; Received in revised form 24 January 2018; Accepted 17 February 2018
Available online 21 April 2018

control actions to save energy and guarantee thermal comfort, as is the case with Model Predictive Control (MPC) [1]. However, a major challenge with MPC is in (accurately) modeling the dynamics of the underlying physical system. The task is much more complicated and time consuming in the case of a large buildings and often times, it can be even more complex and involved than the controller design itself. After several years of work on using first principles based models for peak power reduction, energy optimization and thermal comfort for buildings, multiple authors [1,2] have concluded that the biggest hurdle to mass adoption of MPC in intelligent building control is the cost, time and effort required to capture accurate dynamical models of the buildings. The user expertise, time, and associated sensor costs required to develop a model of a single building is very high. Thus, the payback period for the upfront hardware and software installation is expected to be too high, making MPC an uneconomical choice for energy management. This is probably the main reason why rule-based control strategy have been widely used so far. Indeed, there are several reasons why physics-based modeling is hard for complex systems like buildings:

1. **Model capture** – a building modeling domain expert typically uses a software tool to create a model to reproduce the geometry of a building from the building design and equipment layout plans, and add detailed information about material properties, equipment and operational schedules. However, there is always a gap between the modeled and the real building, and the domain expert must then manually tune the model to match the measured data [3]. Moreover, the modeling process also varies from building to building with the construction and types of installed equipment. Another major downside with physics-based modeling is that enough data is not easily available, so guesses for parameter values have to be made, which also requires expert know-how.
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.

In Section 2 we will present a detailed technical example to better illustrate how data-driven approaches can address the above issues and thus reduce the cost of modeling buildings. In practice, 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 alternative is to use black-box, or completely data-driven, modeling approaches, to obtain a realization of the system's input-output behavior. The primary advantage of using data-driven methods is that it has the potential to eliminate the time and effort required to build white and grey box building models. Listening to real data, from existing systems and interfaces, is far cheaper than unleashing hoards of on-site engineers to physically measure and model the building. Improved building technology and better sensing is fundamentally redefining the opportunities around smart buildings. Unprecedented amounts of data from millions of smart meters and thermostats installed in recent years has opened the door for systems engineers and data scientists to analyze and use the insights that data can provide, about the dynamics and power consumption patterns of these systems.

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 for control, that are also adaptive, scalable and interpretable. We solve this problem by bridging Machine Learning and Predictive Control. In this paper, we present a method based on Random Forests which uses historical data for receding horizon control. We begin with a discussion on the related literature and novelty of our contribution.

1.1. Related work

A vast literature exists in building energy applications that deals with Demand Response, peak power reduction, energy saving, thermal comfort, and related topics. Among them, we selected the ones that we believe are more related to our work.

All these approaches can be classified based on two characteristics:

1. the type of system model:
   - model-based, such as "white-box" and "grey-box" approaches: [9,6,10,15,7,11,12,8,27,14];
   - data-based, i.e. "black-box" approaches, mainly done using Neural Networks: [16–20,14,15];
   - simulation tool-based, such as EnergyPlus [28] and TRNSYS [29]: [4,5];
2. the purpose these models are created for:
   - only model identification: [16–18,6,15,7,8,19,14,20];
   - model identification and control, mainly Predictive Control: [9,10,4,11–13].

These references are summarized in Table 1 highlighting the key differences. We also emphasize the case studies the results are applied to, and whether the authors used experimental data to simulate their algorithms. Only in three cases the algorithms are tested on real systems (see Table 1: RI – Real Implementation). We observe that, except for the last six cases [21–26], which we discuss in detail, either model-based approaches or only tools are considered with/without control, or data-driven modeling approaches are considered *only without* control. The last six papers of Table 1 are more related to the methodology presented in this paper, since they address both data-driven modeling and control. In particular, the authors in [21] proposed a predictive control strategy based on Neural Networks, for boilers control in buildings, to decide the optimal time to switch-on the plant to guarantee energy savings and

Table 1
References ordered considering: case study they are applied to; whether they use experimental data, other than simulated data, and if they do real implementation (RI), i.e. implement the methodologies on real systems; if they use simulative tools; the type of the model considered, i.e. Model-Based or Data-Driven or both; if the models are used for control.

Ref.	Case study	Exp.	Tool	MB/DD	Control
[4]	Commercial Building	Yes	E+	None	Yes
[5]	Commercial building	Yes	E+	None	Yes
[6]	Commercial building	Yes	E+	MB	No
[7]	2 office buildings and 1 residential building	Yes	None	MB	No
[8]	2 commercial buildings	n/a	E+	MB	No
[9]	Residential area	No	None	MB	Yes
[10]	2 residential buildings	Yes	E+	MB	Yes
[11]	3 residential buildings	No	E+	MB	Yes
[12]	6 commercial buildings	Yes	E+	MB	Yes
[13]	Residential building	Yes	None	MB	Yes
[14]	Commercial building	No	E+	MB-DD	No
[15]	2 commercial buildings	Yes	E+	MB-DD	No
[16]	Office building	Yes	None	DD	No
[17]	Office building	Yes	E+	DD	No
[18]	Residential house	Yes	TRANSYS	DD	No
[19]	Residential building	Yes	None	DD	No
[20]	Office building	No	E+	DD	No
[21]	Commercial building	Yes + RI	None	DD	Yes
[22]	Living lab (1 room)	Yes + RI	None	DD	Yes
[23]	Commercial building	Yes + RI	None	DD	Yes
[24]	Residential house	Yes	None	DD	Yes
[25]	9 commercial buildings	No	E+	DD	Yes
[26]	Commercial building	No	E+	DD	Yes

thermal comfort. However, the approach is not easily scalable to different types of plants and does not use optimization in the closed-loop scheme. In [22], an approach based on reinforcement learning, called Model-Assisted Batch Reinforcement Learning, is considered to provide data-driven control for the demand response problem in HVAC systems. Reinforcement Learning is a model-free methodology and is an alternative approach to MPC [30] (with pros and cons). The authors in [23] considered a data-driven predictive control based on Neural Networks to guarantee energy saving and thermal comfort in public buildings. Neural Networks are used in the closed-loop control scheme to determine a thermal comfort index based on parameters that can be measured or estimated, but no Neural Networks-based system state dynamics are included into the optimization problem. [24] uses a Neural Network based data-driven state model as a plant simulator in the MPC closed-loop optimization. More papers related to this topic can be found in the literature. Unfortunately, since Neural Networks models are nonlinear, the MPC based on such models is also nonlinear. This means that a global optimal solution cannot be guaranteed and solving the optimization problem becomes computationally harder due to nonlinearties when the complexity of the neural network is high.

To overcome this complexity, we introduced the regression trees-based approach in [25,26]. In particular, in [25], we used regression trees to setup an MPC problem and apply it to the problem of Demand-Response. However, this approach can be used for optimal control with only one-step lookahead prediction. Hence, it is not possible to use it to control the system considering a prediction over an horizon of arbitrary length. This limitation was addressed in [26], where multi-output regression trees are used instead of single output trees. The different outputs correspond to different steps of the prediction horizon which allows us to setup an MPC problem with a finite horizon. However, modeling accuracy using single trees is strongly affected by overfitting and high variance. On the other hand, such approach has the advantage to be extremely simple from the complexity point of view, although the range of applications is limited. In the conference papers [31,32], we took a different approach that improves the system's identification accuracy – instead of considering a single tree with multiple output, we considered multiple trees and forests with single output. Each tree/forest provides the prediction of the system's behavior for a different time steps of the horizon. However, the results are based on simulated data and we did not account for inaccuracies in the weather forecast and data acquisition.

1.2. Main contribution

In this paper, we provide a new methodology based on random forests that overcomes the drawbacks of all our previous works, and more precisely: we obtain better performance and scalability when compared to other approaches (both optimal and rule-based), and we provide and validate robustness with respect to uncertainties due to real data acquisition and weather forecast inaccuracies. More in detail, the paper contribution is organised as follows.

1. In Section 3, we formally describe our Data-driven model Predictive Control approach, i.e. Data Predictive Control (DPC). For clarity, we first modify the regression trees algorithm to apply DPC, and then we introduce DPC based on random forests.
2. In Section 4, we demonstrate that DPC can provide comparable *performance* when compared to MPC (using a physical model) applied to a perfectly known mathematical model. More precisely, we first consider a benchmark MPC controller using a bilinear building model whose parameters were identified using experiments on a building in Switzerland, then we apply DPC to a dataset simulated from such bilinear model and derive a controller based only on the data. We show that DPC captures 70% variance in MPC and offers a comparable performance.
3. In Section 5, we demonstrate scalability and efficiency of DPC in a Demand Response problem with the aim of enabling financial incentives for the end-customers bypassing the need for expensive high fidelity models. We apply DPC to a 6 story 22 zone building model in EnergyPlus for which model-based control is not economical and practical due to extreme complexity. We observe that DPC provides the desired power curtailment with an average error of 3%.
4. In Section 6, we demonstrate the robustness of our method to uncertainties due to real data acquisition and weather forecast inaccuracies by implementing and testing DPC on historical data from an off-grid house located in L'Aquila, Italy. We derive a predictive model on such real data and design the optimal ON/OFF scheduling for the heating system in order to save energy while guaranteeing thermal comfort for the occupants. We compare the total amount of energy saved with respect to the classical bang-bang controller (widely used in houses for temperature control) using an EnergyPlus model built specifically for the house. We show that we can perform an energy saving that ranges from 25.4% (if we guarantee thermal comfort i.e. strictly respect the desired temperature range in the rooms) to 49.2% (if we allow small violation in the desired temperature range). Finally, we test the robustness of our method to uncertainties in data acquisition and weather forecast.

2. Complexity issues in modeling buildings: physics-based vs data-driven

In this section we provide a detailed example to emphasize the differences in terms of complexity when modeling a building using physical laws versus machine learning. We show that the data-driven modeling eliminates several drawbacks that occur using physics-based models such as the need to have good knowledge of the building structure and the material properties, time required to build a model and limited availability of sensors. The example is based on a real building taken from [1], which is located in Allschwil (Switzerland). The building consists of 6 floors, with a total air conditioned floor area of around 6000 m^2. We will first illustrate the physics-based modeling approach in [1], then we will emphasise some drawbacks and we finally illustrate how data-driven approaches can overcome such issues.

2.1. Physics-based modeling

We describe the physics-based approach based on an RC network outlined in [1], which derives a bilinear model (constructed from physical principles) of the second floor of the building. The model of the whole building is derived in [1] assuming that such floor is identical to all other floors. The modeling process of the second floor consists of three steps:

1. Building geometry and construction data are used together with first-principles to derive the following linear model for the building's thermal dynamics:

$$\dot{x}(t) = A_x x(t) + B_q q(t). \tag{1}$$

This model describes the behavior of the zone, wall, floor and ceiling temperatures. Walls, floors and ceilings are considered as divided into 3 layers with different features. Therefore, each zone was described with an RC network model (see Fig. 3–10 in [33]), where the capacitances represent the states of the layers and the resistances represent the thermal resistance of the layers. The heat exchange between two adjacent layers, i.e. layer "a" and layer "b", is modeled to be proportional to the temperature difference of the two layers and the corresponding thermal resistance R, and is given by

$$\begin{aligned} C_a\dot{x}_a &= \frac{x_b(t)-x_a(t)}{R},\\ C_b\dot{x}_b &= \frac{x_a(t)-x_b(t)}{R}, \end{aligned} \tag{2}$$

where C_a and C_b are the heat capacitances of the layers. This is done for each layer of each zone, obtaining the compact representation given in (1). The thermal parameters are derived from zones geometry and material properties.

2. External heat fluxes, modeled as a bilinear model, affect the building directly as well as indirectly through zones:

$$q(t) = A_q x(t) + B_{q,u} u(t) + B_{q,d} d(t) + \sum_{i=1}^{n_u} [(B_{q,du,i} d(t) + D_{q,xu,i} x(t)) u_i(t)], \tag{3}$$

   where u are the inputs and d the disturbances of the system. Eq. (3) for the heat flux is obtained by modeling:
   - heat exchange associated with the building hull (except for windows), both conductive and radiative part;
   - heat flux to each thermally activated building system (TABS), i.e. pipes buried in the concrete slabs of the floors carrying hot/cold water;
   - heat flux through the windows in three different parts: radiation due to elements directly in contact with the zone air, conduction through the window, and absorption of the solar radiation through the window;
   - convection due to internal gains due to occupants, appliances and lighting;
   - the effects of the AHU.
3. The resulting system (1) is discretized with a sampling time of 15 min. This model has approximately 300 states, that include temperature of the zones, walls and floors on the second floor. The outputs of the system are the zone temperatures. Since the performance of the state estimation, needed to compute the optimal control inputs using MPC, badly scales with the number of states, an approximated model with fewer states is derived in [33]. In particular, although the rooms are equipped with temperature sensors, different averaged temperatures of the building facades (North, South, West, East) and of the zones are considered, obtaining a coarser model with only 35 internal variables. Among these 35 internal variables only 5 are measurable output variables, i.e. the averaged room temperature for each group of zones (North, South, West, East, Center). The system has 18 input variables: TABS heating heat flux, TABS cooling heat flux, averaged transmitted solar heat flux for each group of zones (North, South, West, East) that is estimated using blinds position measurements, air mass-flow through the energy recovering mode, air mass-flow bypassing the energy recovering mode, air massflow through the air cooler, AHU heat coil heat flux, lighting power for the offices for each group of zones (North, South, West, East), and radiator heat flux in the corner offices (North, South, West, East). Finally, 7 disturbance signals are considered: internal gains in the offices and internal gains in non-office zones, which are predicted using a standard schedule, ambient temperature and solar radiation on facade (North, South, West, East) whose values were obtained through Kalman filtering using measurements from the weather station placed on the roof of the building. This filtering is needed to take into account the shadowing of the neighboring buildings. This approximate model is then considered “suitable for MPC” (see Section 3.31.4 in [33]).

To identify model parameters of matrices $A_x, B_q, A_q, B_{q,u}, B_{q,d}, B_{q,du,i}, D_{q,xu,i}$ in Eqs. (1) and (2), the authors built an EnergyPlus model of the building to get geometry and materials data. This was a choice of the authors, but if necessary data are available, real geometry and materials data can also be used to estimate the model parameters. For this particular building, 24 parameters are estimated/taken form a datasheet/computed for the considered zone model. Although some of the parameters are in common among different zones, the others are found independently for each zone. As already discussed, in [33] the parameters of all the other floors are assumed to be identical to the second floor, which potentially introduces substantial modeling uncertainties. To derive the EnergyPlus model using the available measurements and to use the same control implemented in the building, the EnergyPlus model is coupled with MATLAB using BCVTB [34].

The physics-based approach described above, although detailed and accurate (in some cases), is clearly cost and time prohibitive, and as the building characteristics changes with time, the system identification must be repeated to update the model. Moreover, such expensive, time-consuming and complex modeling procedure is unique (and not repeatable) for other buildings. For all these reasons, physics-based modeling suffers practical challenges when the objective is applying MPC to large scale buildings.

2.2. Data-driven modeling

The goal with the data-driven modeling is to learn, based on historical measurable data and without modeling physical details of a building, a function map

$$y(k) = f(x(k),\ldots,x(k-\delta_x),u(k),\ldots,u(k-\delta_u),d(k),\ldots,d(k-\delta_d)), \tag{4}$$

where y represents the variables we wish to predict, x the measurable variables of the system, u the measurable inputs and d the measurable disturbances, with $\delta_x,\delta_u,\delta_d$ memory indices of the variables which capture the dynamical behavior. Compared to the variables in the example presented above in Section 2.1, all variables y,x,u,d in (4) only include variables that are directly measurable through already installed sensors like thermostats and multimeters. Therefore, many internal states like the temperatures of different layers – interior, middle and exterior – of the walls, the floors and the ceilings are not required for black-box modeling, which reduces the order of complexity. We use machine learning to learn such black-box models with the objective of finding the hyperparameters (for a given structure like Random Forests) of a model that best explain the input-output relationship within the measured variables, compensating the effect of the internal unmeasurable variables. The data-driven modeling addresses the following drawbacks of the physical counterpart.

1. During physical modeling in (2), in order to keep the model simple, the heat exchange between layers is assumed to exhibit a linear behavior. In the case of heat flux in (3), again, a bilinear model ignores complex nonlinearities. Hence, many nonlinearities are neglected to simplify the modeling complexities. We avoid such assumptions in data-driven modeling where a nonlinear function such as Random Forest, that represents the dynamics of zone temperatures or power consumption, is learned rather efficiently and accurately. For example, if y is the zone temperature in Eq. (4), we learn a nonlinear function f which depends on current and previous inputs and disturbances. The hyperparameters of f are trained automatically depending upon the learning algorithm.
2. Physical modeling neglects different geometries and different materials of the floors assuming they are the same for each floor, which is never true in reality. Moreover, in many cases, the details of the construction layout and equipments are not even available, so many parameters have to be guessed making physical modeling a difficult choice. On the other hand, data-driven modeling automatically captures the interaction with the environment while optimizing the hyperparameters. Thus, the data from building's construction/materials/equipments are not required explicitly.
3. Tuning physical parameters in (1)–(3) requires expert know-how, which adds to the cost and time of modeling. Data-driven approach reduces both cost and time by an order of magnitude as we directly work with the senor data without explicitly modeling internal states. Further, for a different building, given the historical data from the building, data-driven method is scalable as the same process can be repeated to identify a control-oriented model for MPC.
4. Physical modeling has large number of states and variables. Thus, many more measurements are needed to use the physical model to predict the system's behavior. This can be expensive due to necessary

new sensor installations. Moreover, when some measurements are not directly available from the sensors, observers are needed for state estimations. However, observability problems can limit the construction of such observers [35]. On the other hand, data-driven modeling obviates the need to model the internal states of the walls, floors and ceilings so we do not have as many states. It relies only on direct measurements from the sensors such as thermostats and multimeters, and weather data, reducing the need for new installations and hence the cost of modeling. The behavior of the internal/unobservable/missing states is captured in the parameters of function *f*.

To summarize, the data-driven approaches have the potential to simplify the modeling of buildings to a large extent reducing the overall cost and time investment, while avoiding several assumptions that are generally made in standard physics-based modeling procedures and that reduce the accuracy. Also, unlike physics-based modeling, the parameters (or hyperparamters) of the model are automatically determined by the learning algorithm to compensate for the effect of unobservable variables on the input-output relation.

Given the advantages of data-driven modeling, the challenge now lies in using such models for optimal predictive control. This is exactly the focus of this paper. In the next sections, we will demonstrate how the training algorithm for Random Forests can be modified to develop control oriented models that enable Model Predictive Control, and provide better performance, when compared to other approaches, and robustness with respect to disturbance and to real data acquisition uncertainties.

3. Data predictive control

The central idea behind DPC is to obtain control-oriented models using machine learning on historical datasets of buildings and formulate the optimal control problem in a way that Receding Horizon Control (RHC) can be solved efficiently. Let an historical dataset $(\mathscr{X},\mathscr{Y})$ be given. We define $\mathscr{X} = \{(x(k),u(k),d(k))\}_{k=1}^{n}$ the set of predictor variables (or features), i.e. the set of samples $(x(k),u(k),d(k))$ measured a time instants $k = 1,...,n$, where $x(k) \in \mathbb{R}^{n_x}$ is the vector of the *measurable* state variables (e.g. the rooms temperatures, power consumption of the building, and others), $u(k) \in \mathbb{R}^{n_u}$ is the vector of the *measurable* input variables (e.g. the set-points and schedule information of the building) and $d(k) \in \mathbb{R}^{n_d}$ is the vector of the *measurable and predictable* disturbance variables (e.g. weather historical data). We define $\mathscr{Y} = \{y(k)\}_{k=1}^{n}$ the set of response variables, i.e. the set of measured samples $y(k) \in \mathbb{R}^{n_y}$ representing the variables we wish to predict with our model. In the most general case, when we want to predict all measured variables, the response variables are represented by the state variables at the next time step, i.e. $y(k) = x(k+1)$. However, in general, we only wish to predict a subset $y(k) = \bar{x}(k+1) \subset \mathbb{R}^{n_x}$ of variables, e.g. only room temperatures and power consumption. Clearly, $|\mathscr{X}| = |\mathscr{Y}| = n$ is the number of time samples of the dataset.

We remark that, as explained in Section 2, the dataset $(\mathscr{X},\mathscr{Y})$ does not contain non-measurable states such as temperature of wall layers, ceiling, floor, etc.: nevertheless, our simulated and experimental results show that our data-driven methodology is able to compensate the absence of such variables while providing excellent prediction accuracy.

Our goal is to learn data-driven models, using Regression Trees and Random Forests, that relate the value of the response variables (possibly for a certain future time horizon) with the value of the predictor variables and can be used to set up an MPC problem. To this aim, in the simplest example of one time-step prediction, we need to derive a model with a closed-form expression of the following form

$$y(k) = x(k+1) = f(x(k),u(k),d(k)). \tag{5}$$

However classical Regression Tree and Random Forest algorithms do not provide a closed-form expression for *f*, hence they cannot be used for MPC.

In the next section we provide a new methodology (DPC) that adapts the classical Regression Tree and Random Forest algorithms to determine a closed-form expression for *f* that is efficiently applicable to MPC. For simplicity of presentation we first describe DPC using Regression Trees and then DPC using Random Forests.

3.1. DPC-RT: DPC with Regression Trees

In order to have a model that can be used for prediction in an MPC problem with a future horizon of arbitrary length *N* we need to predict, at time *k*, the response *y* for the next *N* time steps, i.e. $y(k),...,y(k+N)$. For the sake of simplicity and without loss of generality we consider in this section only a scalar response variable, i.e. $y(k) \in \mathbb{R}$ $(n_y = 1)$. We will show in Sections 5 and 6 that multiple trees (or forests) can be easily built to account for the case when $n_y > 1$.

Assume that, at time *k*, we want to predict the response variable at time $k+j$, i.e. $y(k+j)$, $j \in \{1,...,N\}$: when the data have lots of features interacting 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.

As discussed before an approach to non-linear regression is to partition the dataset into smaller regions where the interactions are more manageable. This partition can be obtained by recursively splitting the dataset via an adaptation we propose to the Regression Tree algorithm (see Appendix A for details), and is repeated recursively until we finally get to small chunks of the dataset (i.e. the leaves of the regression tree) where we can fit simple (e.g. linear or affine) parametric models.

Our modification of the Regression Tree algorithm first partitions the features set $\mathscr{X}$ into the sets $\mathscr{X}^c = \{u(k)\}_{k=1}^{n} \subset \mathscr{X}$, containing the control (or manipulated) variables, and $\mathscr{X}^d = \{(x(k),d(k))\}_{k=1}^{n} \subset \mathscr{X}$, containing the disturbance and state (or non-manipulated) variables. The union of the two non-intersecting sets forms the full feature set of training $\mathscr{X} \equiv \mathscr{X}^c \cup \mathscr{X}^d$. Then the training process is divided into the following two steps, which generate as output a tree $\mathscr{T}_j$ as illustrated in Fig. 1 (left):

1. the splitting of the dataset only partitions $\mathscr{X}^d$ (see Appendix A and [36] for technical details): this choice is necessary to make our model suitable for control, as we will clarify later on, and also reduces the computational complexity. To each leaf ℓ_i will correspond an equivalence class of data samples $\mathscr{X}_i^d$ of the partition of $\mathscr{X}^d$, with $\mathscr{X}_i^d \subset \mathscr{X}^d$: as a consequence at any time *k*, given δ_d autoregressive terms of the disturbances and δ_x autoregressive terms of the state, we can associate $(d(k-\delta_d),...,d(k+j),x(k-\delta_x),...,x(k))$, to the corresponding leaf ℓ_i

$$\ell_i = g_{\mathscr{T}_j}(d(k-\delta_d),...,d(k+j),x(k-\delta_x),...,x(k)) \tag{6}$$

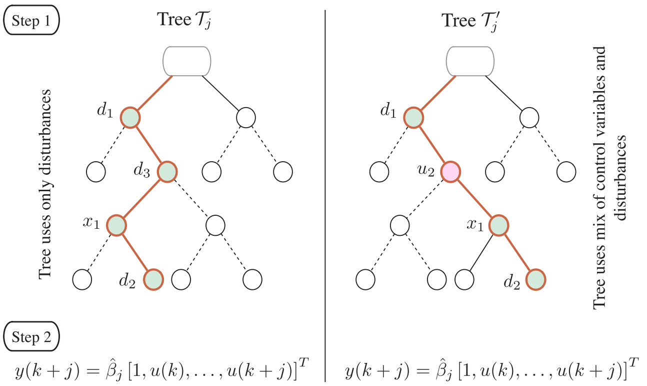

Step 1
Tree $\mathcal{T}_j$
Tree $\mathcal{T}_j'$
Tree uses only disturbances
d_1
d_3
x_1
d_2
d_1
u_2
x_1
d_2
Tree uses mix of control variables and disturbances
Step 2
$y(k+j) = \hat{\beta}_j\,[1, u(k), \ldots, u(k+j)]^T$
$y(k+j) = \hat{\beta}_j\,[1, u(k), \ldots, u(k+j)]^T$

Fig. 1. *Step 1:* Tree $\mathscr{T}_j$ trained only with variables in $\mathscr{X}^d$ using adapted RT algorithm. Tree $\mathscr{T}_j'$ trained with variables in $\mathscr{X}^d$ and control variables in $\mathscr{X}^c$ using classical RT algorithm. *Step 2:* In the leaf ℓ_i of the tree an affine model β_{ij} is defined as a function only of the control variables dataset corresponding to leaf ℓ_i.

if and only if $(d(k-\delta_d),\ldots,d(k+j),x(k-\delta_x),\ldots,x(k)) \in \mathscr{X}_i^d$.

2. in each leaf ℓ_i of the tree $\mathscr{T}_j$ we derive, solving a convex program over the data samples in the partition $\mathscr{X}_i^d$ associated to leaf ℓ_i, an affine function, with coefficients β_{ij}, that relates the response $y(k+j)$ to the previous control inputs in $\mathscr{X}^c$:

$$y(k+j) = \beta_{ij}[1,u(k),\ldots,u(k+j)]^T \tag{7}$$

Clearly the coefficients β_{ij} are different for each leaf ℓ_i as they are derived from different sets of samples.

As an example, the tree $\mathscr{T}_0$ defines the function f in (5) to predict $y(k)$ as follows: assuming for simplicity that $\delta_x = \delta_d = 0$, given the measurement of the variables $(x(k),d(k))$ at time k, it is possible to determine by (6) the corresponding leaf ℓ_i of $\mathscr{T}_0$ and the associated β_{i0}. The prediction of $y(k)$ is provided by (7) as an affine function of the control variable $u(k)$.

Applying the above procedure for $j = 0,\ldots,N$, we build N regression trees $\mathscr{T}_0,\ldots,\mathscr{T}_N$: thus, we have managed to linearize the original model dynamics via black-box modeling.

Our two-steps training procedure, described by the pseudo code of lines 1–11 of Algorithm 1, can be computed off-line: this is an important advantage because the time required to create the model does not affect the control execution in run-time.

The next problem now is: *how to use this modeling framework to set up an MPC problem?* We setup the MPC optimization problem in the general case of multiple responses, i.e. $y(k) \in \mathbb{R}^{n_y}$, $n_y \geqslant 1$, as follows:

Problem 1.

$$\begin{aligned}
\underset{u_{k+j},\epsilon_j}{\text{minimize}} \quad & \sum_{j=0}^{N} y_{k+j}^{\top} Q y_{k+j} + u_{k+j}^{\top} R u_{k+j} + \lambda\, \epsilon_j \\
\text{subject to} \quad & y_{k+j} = \widehat{\beta}_j [1,u_k^{\top},\ldots,u_{k+j}^{\top}]^{\top} \\
& u_{k+j} \in \mathscr{U} \\
& |y_{k+j}| \leqslant \overline{y}_{k+j} + \epsilon_j \\
& \epsilon_j \geqslant 0 \\
& j = 0,\ldots,N,
\end{aligned} \tag{8}$$

where $\widehat{\beta}_j$ is defined later on. Here, $Q \succeq 0 \in \mathbb{R}^{n_y \times n_y}$ and $R \succeq 0 \in \mathbb{R}^{n_u \times n_u}$ are weight matrices used to trade-off the importance we provide in the minimization to y versus u. The slack variables ϵ_j are added to ensure recursive feasibility: we relax the equality constraint on y allowing violation up to ϵ_j to guarantee that Problem 1 can provide a solution at each step $k \geqslant 0$. The weight λ is then used to tradeoff the importance we provide in the optimal solution to the constraints violation versus the quadratic term. For example in the following sections we will use ϵ_j and λ to define our tolerance to the violation of prescribed bounds by the rooms' temperature.

Clearly different cost functions can be chosen depending upon the application, i.e. they can be linear, nonlinear, etc., obviously changing the complexity of the problem. In the current formulation, the data-driven control problem, is reduced to a convex program which is very easy and efficient to solve in C, C+ +, Python and Matlab, and thus is easily integrable in SCADA systems. Indeed, Problem 1 is solved as in the classical MPC formulation, i.e. at each time step $k = 1,2,\ldots$ the optimal control sequence $u_k^*,\ldots,u_{k+N}^*$ is computed, and only the first input of the sequence is applied to the system: $u(k) = u_k^*$.

Note that each tree $\mathscr{T}_j$ contributes to Problem 1 with the linear constraint $y_{k+j} = \widehat{\beta}_j[1,u_k^{\top},\ldots,u_{k+j}^{\top}]^{\top}$ as a replacement for the state dynamics in the classical MPC formulation. As a consequence, when solving Problem 1 at time k, we need to determine the affine functions parameters $\widehat{\beta}_j$: it follows by Eq. (6) that, to determine $\beta_{ij} j = 0,\ldots,N$ at time k, the knowledge of the state and disturbance measurements $(d(k-\delta_d),\ldots,d(k+N),x(k-\delta_x),\ldots,x(k))$ is needed. However, the values of $d(k+1),\ldots,d(k+N)$ are not available at time k, so we need to use disturbance forecast $\widetilde{d}(k+1),\ldots,\widetilde{d}(k+N)$. Using this information we can narrow down to a leaf in each tree using (6) and thus retrieve the affine model with β_{ij} in (7) for each step $j = 0,\ldots,N$ of the horizon, and associate $\widehat{\beta}_j := \beta_{ij}$. The run-time solution of Problem 1 illustrated above is described by the pseudo code of lines 12–23 of Algorithm 1.

Remark 1. The authors in [37] investigate the effect of uncertainty in the weather forecast on the performance of MPC in building systems operations through a large-scale simulation study, and compare against a rule-based strategy. They consider 48 different scenarios of uncertainties for 72 h weather forecast. With such a long horizon, results have shown that (with a few exceptions) MPC outperforms the rule-based controller in providing energy savings, and is in general quite close to the perfect forecast case despite the uncertainty in weather forecast. The length of horizon for the DPC algorithm presented in this paper is usually much shorter, for example 6 h in Section 4, 7 h in Section 5, and 40 min in Section 6. As a consequence, we can reasonably presume that our approach will be robust to weather forecast inaccuracies. Indeed, in Section 6 we test the robustness of our approach to noisy weather forecast and show that the control performance is very close to the ideal case.

Remark 2. It is now easy to understand that using the classical Regression Tree algorithm, e.g. using also the input variable u in the data splitting procedure to create the trees as in the right side of Fig. 1, the resulting model would not be suitable for control. Indeed, since u is the variable we want to optimize in Problem 1, at time k we have still not chosen its value at times $k,\ldots,k+N$: as a consequence the affine functions *parameters*β_{ij} needed to set up Problem 1 cannot be determined at time k.

The pseudo code for the whole DPC-RT procedure (i.e. Off-line and Run-time) is given in Algorithm 1. Our procedure is also graphically described in Fig. 2 for the Random Forest case, providing a good intuition also for the Regression Tree case.

Algorithm 1. Data Predictive Control with Regression Trees

```
 1: DESIGN TIME (OFF-LINE)
 2: procedure MODEL TRAINING USING DATASET SPLITTING
 3:   Set $\mathscr{X}^c$ ← manipulated features
 4:   Set $\mathscr{X}^d$ ← non-manipulated features
 5:   Build $N$ predictive trees with $(\mathscr{X}^d,\mathscr{Y})$ using Regression Trees
      algorithm
 6:   for all trees $\mathscr{T}_j$ do
 7:     for all leaves $\ell_i$ of $\mathscr{T}_j$ do
 8:       Compute parameters $\widehat{\beta}_j := \beta_{ij}$ in (7) using convex
      programming
 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 $\mathscr{T}_j$ do
16:       Determine the leaf $\ell_i$ using $\mathscr{X}^d$ as in (6)
17:       Obtain the linear model at $\ell_i$ trained in (7)
18:     end for
19:     Solve Problem 1 to determine optimal
20:     control actions $u_k^*,\ldots,u_{k+j}^*$
21:     Apply the first input $u(k) = u_k^*$
22:   end while
23: end procedure
```

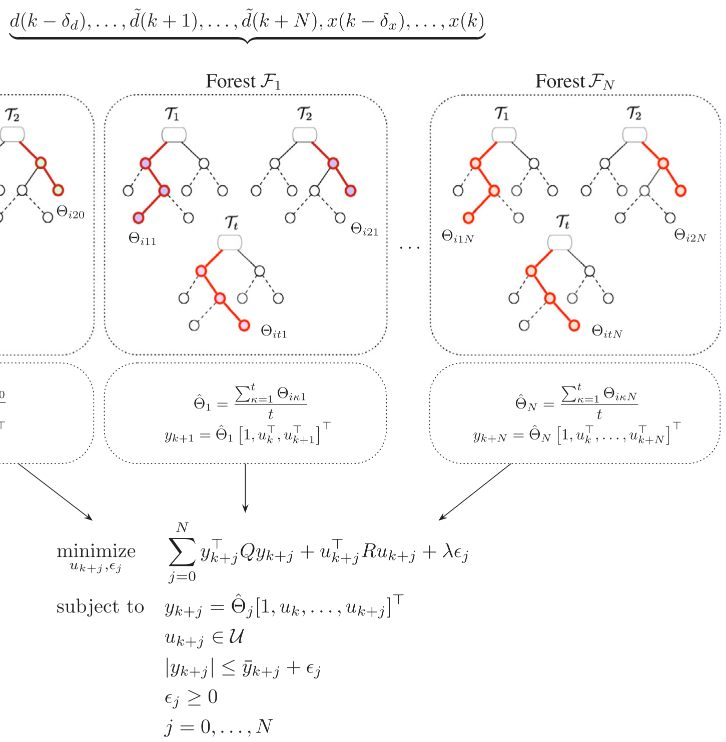

$\underbrace{d(k-\delta_d),\ldots,\tilde{d}(k+1),\ldots,\tilde{d}(k+N),x(k-\delta_x),\ldots,x(k)}$
Forest $\mathcal{F}_0$
Forest $\mathcal{F}_1$
Forest $\mathcal{F}_N$
$\mathcal{T}_1$
$\mathcal{T}_2$
$\mathcal{T}_t$
Θ_{i10}
Θ_{i20}
Θ_{it0}
Θ_{i11}
Θ_{i21}
Θ_{it1}
Θ_{i1N}
Θ_{i2N}
Θ_{itN}
. . .
$\hat{\Theta}_0 = \frac{\sum_{\kappa=1}^{t}\Theta_{i\kappa 0}}{t}$
$y_k = \hat{\Theta}_0\,[1, u_k^\top]^\top$
$\hat{\Theta}_1 = \frac{\sum_{\kappa=1}^{t}\Theta_{i\kappa 1}}{t}$
$y_{k+1} = \hat{\Theta}_1\,[1, u_k^\top, u_{k+1}^\top]^\top$
$\hat{\Theta}_N = \frac{\sum_{\kappa=1}^{t}\Theta_{i\kappa N}}{t}$
$y_{k+N} = \hat{\Theta}_N\,[1, u_k^\top, \ldots, u_{k+N}^\top]^\top$
$\underset{u_{k+j},\epsilon_j}{\text{minimize}} \quad \sum_{j=0}^{N} y_{k+j}^\top Q y_{k+j} + u_{k+j}^\top R u_{k+j} + \lambda\epsilon_j$
subject to $y_{k+j} = \hat{\Theta}_j[1, u_k, \ldots, u_{k+j}]^\top$
$u_{k+j} \in \mathcal{U}$
$|y_{k+j}| \le \bar{y}_{k+j} + \epsilon_j$
$\epsilon_j \ge 0$
$j = 0, \ldots, N$

Fig. 2. Graphical description for solving Problem 2 at each time k to compute u_k^*.

3.2. 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, and a small change in the data can result in a different series of splits and thus affect the prediction accuracy. This is the price to be paid for estimating a tree-based structure from the data.

To address the above problems we use ensemble methods [38], in particular Random Forests, 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. As a consequence each tree in the forest is trained on different data, which introduces differences between the predictive models of the trees.

More precisely, in DPC-En we replace each tree in Algorithm 1 by a forest $\mathscr{F}_j$ of t trees $\mathscr{T}_{1j},\ldots,\mathscr{T}_{tj}$. Each tree $\mathscr{T}_{\varkappa j}$ is trained on the basis of a different subset of features $\mathscr{X}_{\varkappa j}^{d} \subset \mathscr{X}^d$. As discussed in the previous section, at any time k, given δ_d autoregressive terms of the disturbances and δ_x autoregressive terms of the state, we can associate $(d(k-\delta_d),\ldots,d(k+j),x(k-\delta_x),\ldots,x(k))$, to the corresponding leaf ℓ_i of $\mathscr{T}_{\varkappa j}$

$$\ell_i = g_{\mathscr{T}_{\varkappa j}}(d(k-\delta_d),\ldots,d(k+j),x(k-\delta_x),\ldots,x(k)). \tag{9}$$

Also, in each leaf ℓ_i of tree $\mathscr{T}_{\varkappa j}$ we derive an affine function $\Theta_{i\varkappa j}$, that relates the response $y(k+j)$ to the previous control inputs in $\mathscr{X}^c$:

$$y(k+j) = \Theta_{i\varkappa j}[1,u(k),\ldots,u(k+j)]^T. \tag{10}$$

We can now set up the MPC problem for DPC-En as follows:

Problem 2.

$$\begin{aligned}
\underset{u_{k+j},\epsilon_j}{\text{minimize}} \quad & \sum_{j=0}^{N} y_{k+j}^\top Q y_{k+j} + u_{k+j}^\top R u_{k+j} + \lambda\,\epsilon_j\\
\text{subject to} \quad & y_{k+j} = \hat{\Theta}_j[1,u_k,\ldots,u_{k+j}]^\top\\
& u_{k+j} \in \mathscr{U}\\
& |y_{k+j}| \leqslant \bar{y}_{k+j} + \epsilon_j\\
& \epsilon_j \geqslant 0\\
& j = 0,\ldots,N,
\end{aligned} \tag{11}$$

where $\hat{\Theta}_j$ is defined later on. Note that, as in the DPC-RT, the crucial problem when solving Problem 2 is determining all parameters $\hat{\Theta}_j$ at time k. We first proceed similarly to DPC-RT by using forecast of disturbances and the state measurements to determine for each tree $\mathscr{T}_{\varkappa j}$ the corresponding leaf ℓ_i and thus the affine model $\Theta_{i\varkappa j}$, with $\varkappa = 1,\ldots,t$ and $j = 0,\ldots,N$. In DPC-En, however, we have t affine models for each prediction step j, thus we define $\hat{\Theta}_j = \frac{1}{t}\sum_{\varkappa=1}^{t}\Theta_{i\varkappa j}$ averaging the affine models associated to all trees of forest $\mathscr{F}_j$. Once we are able to determine at each time k the parameters $\hat{\Theta}_j$ of the constraints of Problem 2, the optimal input u_k^* can be computed as described in the previous section. The overall procedure is sketched 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. Reasoning on complexity, we remark that the off-line training computation in DPC-En is increased compared to DPC-RT as we need to train t trees for each of the N prediction steps. The run-time computation is also slightly increased as we need to derive the average $\hat{\Theta}_j = \frac{1}{t}\sum_{\varkappa=1}^{t}\Theta_{i\varkappa j}$ for each of the N prediction steps. However, as shown in the following sections, it's worth the price as we obtain much better accuracy and lower variance properties with a marginal increase of the run-time computation time (the increase of off-line computation time is not a big issue as it must be run very seldom, i.e. only when a new model of the building needs to be created).

4. 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 [39].

The bilinear model is a standard building model used for practical considerations [40–42] 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 Section 5, we show how DPC scales easily to such large scale models.

4.1. 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 $x := [\mathsf{T}_{\mathrm{in}},\mathsf{T}_{\mathrm{sb}}^{(1:5)},\mathsf{T}_{\mathrm{ef}}^{(1:3)},\mathsf{T}_{\mathrm{in}}^{(1:3)}]^\top$.

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 $u := [\mathsf{B},\mathsf{L},\mathsf{H},\mathsf{C}]^\top$. 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 h. 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 [43]. We define $d := [\mathsf{Q}_{\mathrm{sc}},\mathsf{Q}_{\mathrm{so}},\mathsf{I}_{\mathrm{o}},\mathsf{Q}_{\mathrm{io}},\mathsf{Q}_{\mathrm{ie}},\mathsf{T}_{\mathrm{db}},\mathsf{T}_{\mathrm{wb}}]^\top$. For further details, we refer the reader to [41].

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

$$x(k+1) = Ax(k) + (B_u + B_{xu}[x(k)] + B_{du}[d(k)])u(k) + B_d d(k), \tag{12}$$

where $x_k \in \mathbb{R}^{12}, u_k \in \mathbb{R}^4, d_k \in \mathbb{R}^8 \quad \forall\, k = 0,\ldots,T$, and the matrices B_{xu} and B_{du} are defined as

$$\begin{aligned} B_{xu}[x(k)] &= [B_{xu,1}x(k),B_{xu,2}x(k),B_{xu,3}x(k),B_{xu,4}x(k)] \in \mathbb{R}^{12\times 4},\\ B_{du}[d(k)] &= [B_{du,1}d(k),B_{du,2}d(k),B_{xu,3}x(k),B_{du,4}d(k)] \in \mathbb{R}^{12\times 4}, \end{aligned} \tag{13}$$

with $B_{xu,i} \in \mathbb{R}^{12\times 12}, B_{du,i} \in \mathbb{R}^{12\times 8} \quad \forall\, i = 1,2,3,4$. For this study, we assume that the disturbances are precisely known to MPC as well as DPC controller.

4.2. 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 [44]. After an optimal sequence of control inputs is 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 $c^\top u$ while maintaining a desired level of thermal comfort x_{ref} (or $\mathsf{T}_{\mathrm{ref}}$). Therefore, at time step k, we solve a continuously linearized MPC problem to determine the optimal sequence of inputs $u_k^*,\ldots,u_{k+N-1}^*$:

$$\underset{u_{k+j-1},\epsilon_j}{\text{minimize}} \quad \sum_{j=1}^{N+1} (x_{k+j}-x_{ref})^\top Q(x_{k+j}-x_{ref}) + c^\top u_{k+j-1} + \lambda\, \epsilon_j \tag{14a}$$

$$\text{subject to} \quad x_{k+j} = Ax_{k+j-1} + Bu_{k+j-1} + B_d d_{k+j-1} \tag{14b}$$

$$B = B_u + B_{xu}[x_k] + B_{du}[d_{k+j-1}] \tag{14c}$$

$$\underline{u} \le u_{k+j-1} \le \overline{u} \tag{14d}$$

$$\underline{x}-\epsilon_j \le x_{k+j} \le \overline{x} + \epsilon_j \tag{14e}$$

$$x_k = x(k) \tag{14f}$$

$$\epsilon_j \ge 0,\ j = 1,\ldots,N+1, \tag{14g}$$

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.

4.3. Data Predictive Control

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

4.3.1. 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 (12). 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 in $\mathscr{Y}$ of interest for which we build N trees and N forests as described in Sections 3.1 and 3.2, respectively. Therefore, $y(k+j) := x(k+j+1)^1$, where x^1 is the first component of x. Next, we define the non-manipulated features in $\mathscr{X}^{\mathrm{d}}$. At time k, for the tree $\mathscr{T}_j$ and the forest $\mathscr{F}_j$, we base these features to include weather disturbances, external disturbances due to occupancy and equipments, and autoregressive terms of the room temperature, i.e. $\mathscr{X}^{\mathrm{d}} := \{(d(k+j-N),\ldots,d(k+j-1),x(k)^1,\ldots,x(k-\delta)^1)\}$, $\forall\, k = 1,\ldots,n$, where δ is the order of state autoregression. Finally, the inputs in DPC are exactly same as in MPC, i.e. $\mathscr{X}^{\mathrm{c}} := \{u(k)\}$, $\forall\, k = 1,\ldots,n$. 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.

4.3.2. Optimization

For a fair comparison with MPC, we cast DPC optimization problem as follows:

$$\underset{u_{k+j},\epsilon_j}{\text{minimize}} \quad \sum_{j=0}^{N} (y_{k+j}-x_{ref})\mathscr{Q}^{(1,1)}(y_{k+j}-x_{ref}) + c^\top u_{k+j} + \lambda\, \epsilon_j \tag{15a}$$

$$\text{subject to} \quad y_{k+j} = \alpha_j[1,u_k^\top,\ldots,u_{k+j}^\top]^\top \tag{15b}$$

$$\underline{u} \le u_{k+j} \le \overline{u} \tag{15c}$$

$$\underline{y}-\epsilon_j \le y_{k+j} \le \overline{y} + \epsilon_j \tag{15d}$$

$$\epsilon_j \ge 0,\ j = 0,\ldots,N, \tag{15e}$$

where $\alpha_j = \widehat{\beta}_j$ for DPC-RT and $\alpha_j = \widehat{\Theta}_j$ for DPC-En. Note that, (15) is DPC analog of (14). The only difference is the state dynamics (14b) and (14c) are now replaced with (15b).

4.3.3. Validation

We compare the prediction for the first time step $y(k+1)$ and the

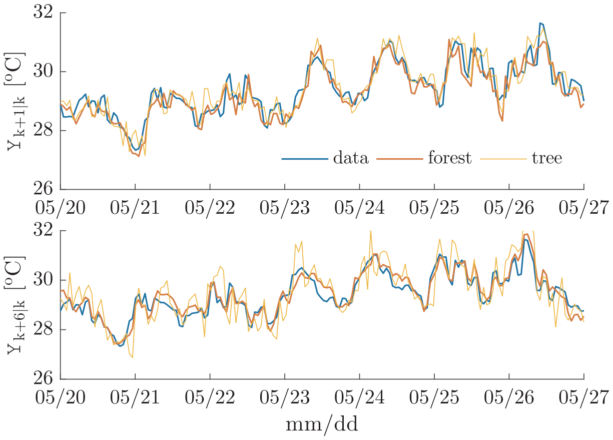


Fig. 3. Temperature predictions from a tree and a forest for first step prediction (top) and the 6-h ahead prediction (bottom). Ensemble method shows a relatively higher accuracy.

6 h ahead prediction $y(k + 6)$, given 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 were not used for training. The quantitative summary of the accuracy is given in Table 2. We can see that the random forests are better in all respects.

4.4. Comparison

We compare the performance of DPC (15) against an equivalent MPC formulation (14). 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. 5. The sampling time in the simulations is 1 h. The control horizon N and the order of autoregression are both 6 h. The training procedure required a few minutes in the case of trees and 2 h for forests on a Win 10 machine with an i7 processor and 8 GB memory. The cooling usage factor C is constrained in [0,1], 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 [45].

The external disturbances – solar gain, internal gain due to equipment and dry-bulb temperature during the chosen periods are shown in Fig. 4. 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 heating system is switched on during the night to maintain the thermal comfort requirements. When the building is occupied during the day, due to excessive internal gain, 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. 5(a) and (b), 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 h which is already too high. The room temperature profile in Fig. 5(c) is close to the reference in the case of DPC-En as well as MPC. Fig. 5(d) shows that the cumulative cost of the

Table 2
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 – $y(k + 1)$	0.42	0.75	0.76
tree – $y(k + 6)$	0.64	0.41	0.42
forest – $y(k + 1)$	0.29	0.87	0.88
forest – $y(k + 6)$	0.38	0.78	0.80

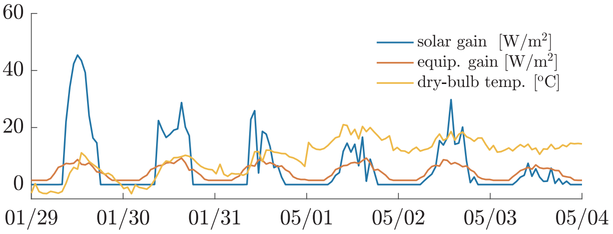
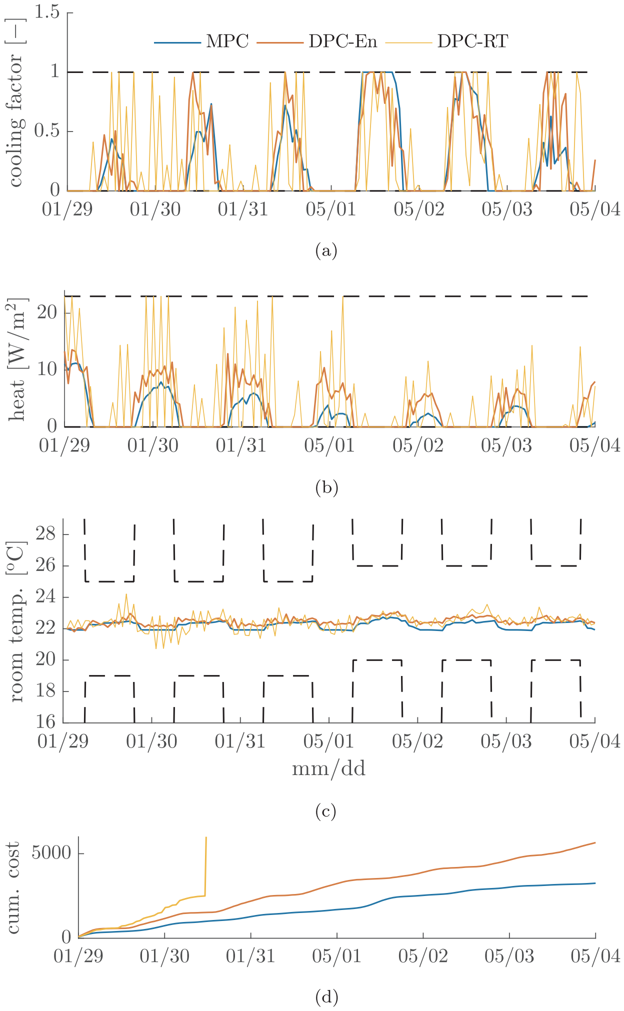


Fig. 4. External disturbances: solar gain, internal gain due to equipment and dry-bulb temperature.



(a)

(b)

(c)

(d)

Fig. 5. Comparison of optimal performance obtained with MPC, DPC-En and DPC-RT for 3 days in January and 3 days in May. (a) Optimal cooling factor C. DPC-En control strategy very similar to MPC. (b) Optimal radiator heat H. DPC-En control strategy very similar to MPC. (c) Room temp. has time varying bounds. When building is occupied, constraints are relaxed. MPC and DPC-En track the ref. temp. (22 °C) closely. (d) Cumulative optimal cost after solving optimization. MPC serves as the benchmark with the minimum cost, followed by DPC-En and then DPC-RT.

Table 3
Quantitative comparison of explained variance, mean value of objective function, mean input cost $c^{\top}u$ and mean deviance from the reference temperature $|\mathsf{T}-\mathsf{T}_{\mathrm{ref}}|$.

	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

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 Table 3. 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 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 [1]. 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.

5. Application to demand response

In January 2014, the east coast (PJM) electricity grid experienced an 86x increase in the price of electricity from $31/MW h to $2680/MW h in a matter of 10 min. Similarly, the price spiked 32x from an average of $25/MW h to $800/MW h 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 synthesising 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.

5.1. EnergyPlus model

We use the DoE Commercial Reference Building (DoE CRB) simulated in EnergyPlus [46] 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.

5.2. Model training for DPC

In the following simulations, we consider a long DR event from 7 am to 2 pm 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 to 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 $\mathsf{GuestSP}$, (iii) cooling set point for kitchen $\mathsf{KitchenSP}$, (iv) chilled water supply temperature ChwSP, and (v) supply air temperature $\mathsf{SupplyAirSP}$, so $u := [\mathsf{ClgSP},\mathsf{GuestClgSP},\mathsf{KitchenClgSP},\mathsf{SupplyAirSP},\mathsf{ChwSP}]^{\top}$. The power forest $\mathscr{F}_p$ is built using the total building power consumption P. Its features in $\mathscr{X}^d$ 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 $\mathscr{F}_t$ is built with zone temperature T as the output. Except for the autoregressive terms corresponding to the same zone temperature, all other features are same in $\mathscr{X}^d$. Fig. 6 shows the prediction accuracy for the power forest, and also explains the two-steps training approach introduced in Section 3. 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.

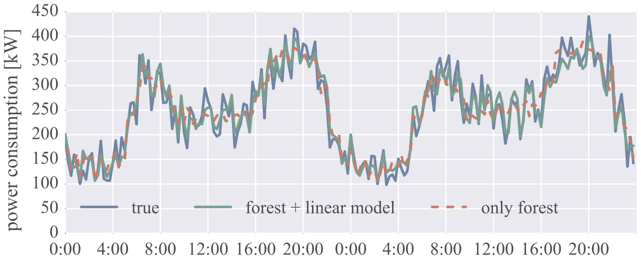


Fig. 6. Model accuracy during training: the prediction made by forest using only $\mathscr{X}^d$ (red) captures the effect due to disturbances. The linear models in the leaves capture the local effects (green) due to the control inputs in $\mathscr{X}^c$ and improve the model accuracy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

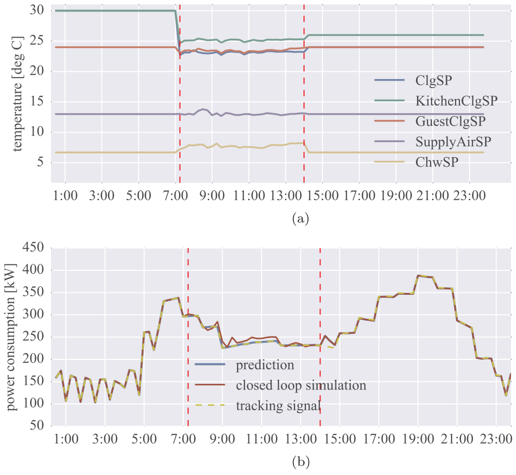
5.3. Power management

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 defined before:

$$\begin{aligned}
\underset{u_{k+j-1},\epsilon_j,\delta_j}{\text{minimize}}\quad & \sum_{j=1}^{N} (\mathsf{P}_{k+j}-\mathsf{P}_{ref})^2 + \lambda\,\epsilon_j + \nu\delta_j\\
\text{subject to}\quad & \mathsf{P}_{k+j} = \widehat{\Theta}_{\mathsf{P}_j}[1,u_k^\top,\ldots,u_{k+j-1}^\top]^\top\\
& \mathsf{T}_{k+j} = \widehat{\Theta}_{\mathsf{T}_j}[1,u_k^\top,\ldots,u_{k+j-1}^\top]^\top\\
& \underline{\mathsf{P}}-\epsilon_j \leqslant \mathsf{P}_{k+j} \leqslant \overline{\mathsf{P}} + \epsilon_j\\
& \underline{\mathsf{T}}-\delta_j \leqslant \mathsf{T}_{k+j} \leqslant \overline{\mathsf{T}} + \delta_j\\
& \underline{u} \leqslant u_{k+j-1} \leqslant \overline{u}\\
& \epsilon_j \geqslant 0,\delta_j \geqslant 0,\ j = 1,\ldots,N.
\end{aligned} \tag{16}$$

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. 7. The DPC controller is active between 7 am and 2 pm. Before 7 am and after 2 pm, the building is using a predefined rule-based control strategy. The optimal control inputs from DPC-En are shown in Fig. 7(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 7 am and 2 pm is 3%. The results from the closed-loop simulation are shown in Fig. 7(b). Marked in blue is the response when the optimal input is applied to the power predictive model of the Random Forest (16) and in red is the response when the same input is applied to the power predictive model of EnergyPlus. Since the optimal input is computed using the power predictive model of the Random Forest the blue trajectory perfectly follows the tracking signal. The red trajectory, as expected, is characterised by a (small) tracking error because of the model mismatch between the predictive model of the Random Forest (used to compute the input) and that of EnergyPlus (used to simulate the closed-loop system). Due to this inaccuracy, the actual power consumption is on an average 7 kW higher.



(a)

(b)

Fig. 7. Power management using DPC. The controller is active between 7 am and 2 pm. This region is marked in dashed red lines. (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. In blue is the response when the optimal input is applied to the power predictive model of the Random Forest and in red is the response when the same input is applied to the power predictive model of EnergyPlus). The difference in closed-loop simulation and prediction is due to model mismatch. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

6. Application to optimal heating system scheduling

In this section we show an application of DPC to a real house located in L'Aquila, Italy. Random forest models for DPC built using historical data from this house are described in Section 6.3. In Section 6.2, we present the thermal energy model of the building in EnergyPlus, built using historical data, construction layout and materials after spending ~ 3 months of efforts. In Section 6.4, we set up the DPC optimization problem and describe the bang-bang control strategy. Finally, in Section 6.5, the performance of the two controllers are compared in terms of energy savings using the EnergyPlus model. In particular, in Section 6.5.1, we show that DPC provides up to 49.2% energy savings with respect to the bang-bang controller while guaranteeing thermal comfort for the occupants, considering the perfect knowledge of the weather forecast. In Section 6.5.2, we show that DPC is robust with respect to imperfect weather forecast.

The contribution of this section, graphically shown in Fig. 8, is threefold.

1. Differently from Section 5, where we used data simulated from an EnergyPlus model, we consider here real data from an occupied house, hence subject to real world imperfections such as random (non predictable) occupancy schedules, open/close windows, random (non predictable) light on/off switch, etc.

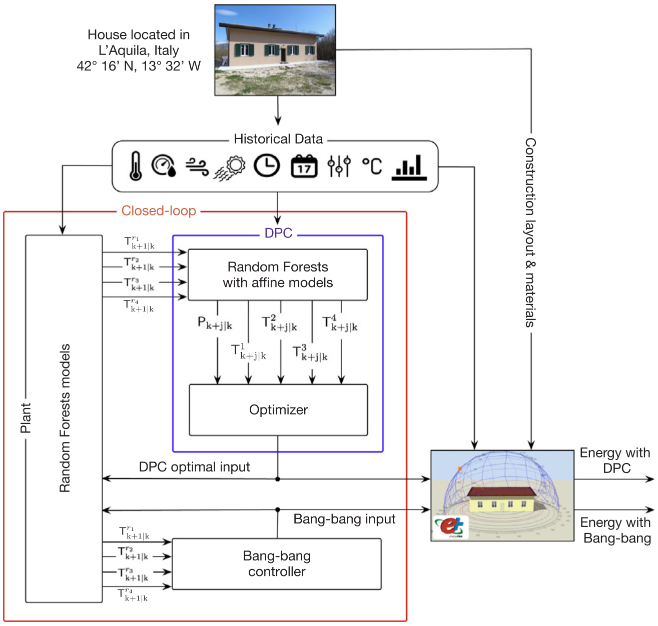


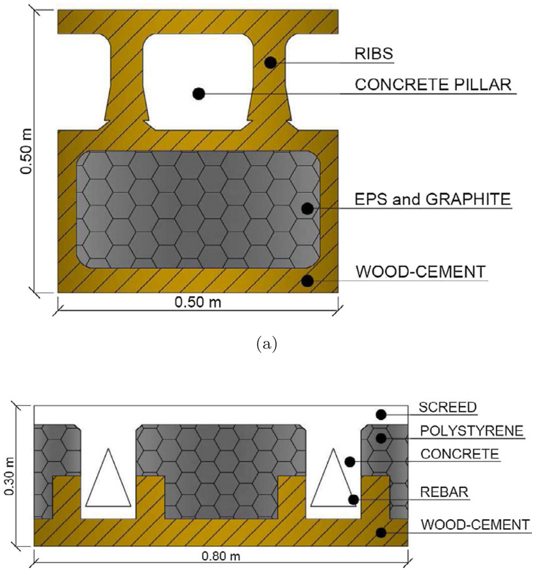
Fig. 8. Random Forest trained on all features including the control variable are used as the plant in the closed-loop simulations with DPC and bang-bang controller. DPC optimization uses Random Forests with affine functions as predictive models. Controllers' strategies are evaluated using the EnergyPlus model to compare the actual energy consumption.

Fig. 9. Off-grid residential house.

2. While in Section 5 we follow a reference DR signal while maintaining thermal comfort, in this section we show the adaptability of DPC to different problems where we minimize the energy consumption required to keep the room temperatures within a specified range of comfort. In this way, we show the potential of DPC in providing energy savings for a long period.
3. We consider uncertainties in the weather forecast, showing that DPC is robust with respect to prediction inaccuracies.

6.1. *Description of the house*

The chosen case study is a detached off-grid two-story residential house, located in the outskirt of L'Aquila (coordinates 42°16′ latitude and 13°32′ longitude), Italy and is shown in Fig. 9. The building, inhabited by the two owners, has a main north-south orientation and it is composed by a heated ground floor and an attic without heating system. Therefore, although the gross area of the house is equal to $209.5\,\text{m}^2$, the heated gross area is equal to $112.4\,\text{m}^2$. Thanks to the off-grid characteristic, the technological plants guarantee the complete energy self-sufficiency of the building. The house is equipped with a biomass boiler, a solar thermal plant, a stand-alone photovoltaic system, black water and rainwater reuse systems and a well for water supply, and has complete independence from the utilities. The bearing structure of the house is made of reinforced concrete and EPS (expanded polystyrene) insulation, while the building envelope is composed by prefabricated wood-cement blocks, shown in Fig. 10, with EPS and graphite insulation, that allow low thermal fluxes. The thermal performance of a wood-cement block, with similar geometry, was investigated in a previous work [47], both with experimental and numerical approach. The heating system of the building, that supplies the thermal energy required in winter season, is a hydronic system consisting of a vegetable biomass boiler, with a manual ON/OFF, a constant-flow pump station, and tubular steel radiators as shown in Fig. 11. The standard biomass boiler has an efficiency of 83.5% with 16.5 kW of thermal power transferred to the water, without gas-flame modulation. The heat transfer fluid distribution is realized through a manifold circuit, that supplies the radiators placed in the various rooms of the house. The energy needs for the domestic hot water (DHW) are covered by the same boiler, coupled with a solar thermal plant. It is worth noting that, in this work, the thermal energy needed for the domestic hot water production is neglected. For a thorough knowledge of the building thermal behavior, an in-situ analysis was carried out. Temperature measuring devices, completely self-produced at the G. Parolini Lab of the University of L'Aquila [48], mainly based on an $ATmega2560$ microcontroller and $DS18B20$ temperature probes (temperature range from −55.0 °C to 125.0 °C with an accuracy of ± 0.5 °C), were employed to acquire the ambient temperatures of 4 differently oriented rooms of the house as shown in Fig. 12 (displayed with red dots $A1$,$A2$,$A3$ and $A4$). The probes positions were chosen to optimize the data acquisition and to minimize the discomfort of the occupants in the house. Furthermore, as can be seen in Figs. 11 and 12, a commercial heat meter (temperature range from 10.0 °C to 90.0 °C with an accuracy of ± 0.05 °C, displayed with orange dot $G1$) was installed downstream of the biomass boiler to measure the produced thermal energy. The turbine flowmeter was installed on the return pipe, while the two thermocouples were placed inside the delivery and return pipes, respectively. All the measuring devices have been set with a data acquisition rate equal to 10 min, from March 11th 2016 to May 15th 2016.



(a)

(b)

Fig. 10. Cross-sections of the wood-cement blocks. (a) Walls. (b) Floor and roof.

The collected data are used in the following to create both an EnergyPlus and a random forest model for the energy consumption assessment of the house. These models will be used for performance comparison of DPC with respect to a classical bang-bang controller. In particular DPC will be set up to provide an optimal ON/OFF scheduling policy for the heating system in order to save energy while guaranteeing thermal comfort for the occupants. To this aim, we also create random forest models for power consumption and room temperatures to be used in the closed-loop simulations.

6.2. *EnergyPlus model*

In this section we create an EnergyPlus model of the thermal energy consumption of the house that will be used in Section 6.5 to compare the quality of the DPC with respect to a classical bang-bang controller.

To assess energy performance of the use case heating system, the EnergyPlus [28] together with DesignBuilder modeling environment [49] has been employed. The model has been created based on L'Aquila weather data, shown in Fig. 13, provided by the CETEMPS Centre of Excellence [50]. According to the Köppen-Geiger climate classification, Italy is classified in Csa, Cfa and Csb climate zones, with a warm temperate climate [51].

Because of the complex morphology of the blocks shown in Fig. 10, some simplified assumptions were made, in order to create the EnergyPlus virtual model. Considering the thermal properties of the wood-cement blocks that compose the walls shown in Fig. 10(a), the thermal effects of the ribs were neglected. For the block employed for floor and roof shown in Fig. 10(b), a less complex equivalent block, with only three layers, was evaluated in order to consider only one equivalent

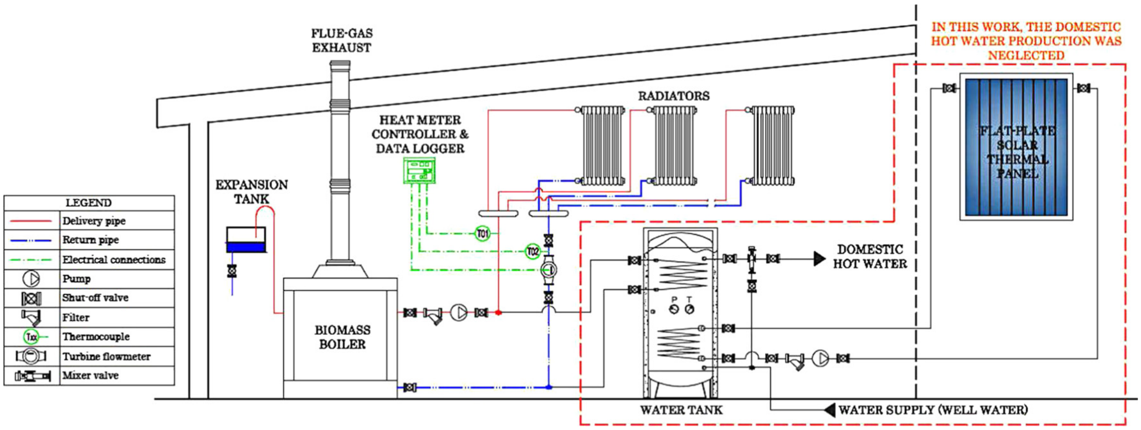
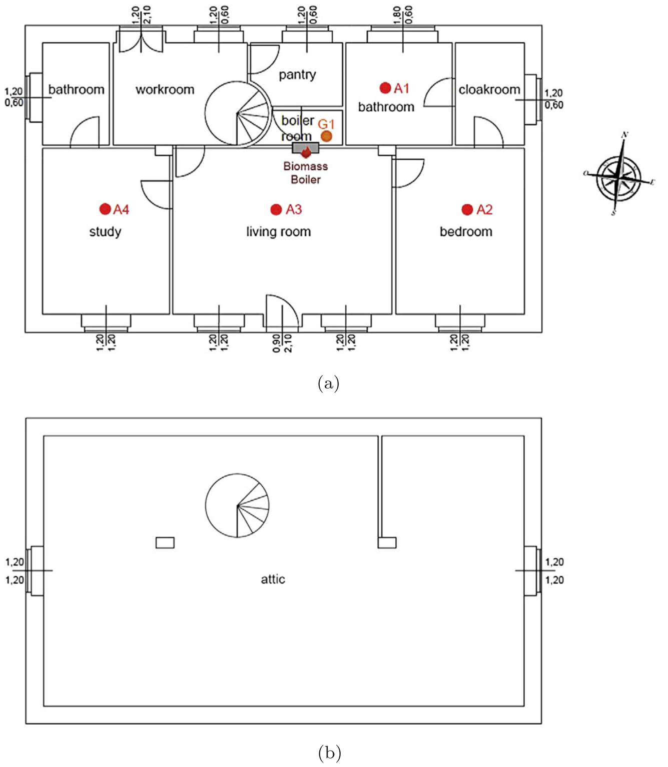


Fig. 11. Technological plant scheme of the use case.



(a)

(b)

Fig. 12. Layout of the house and probes placement. Legend: red circle for ambient temperature; orange circle for heat meter. (a) Ground floor. (b) Attic. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

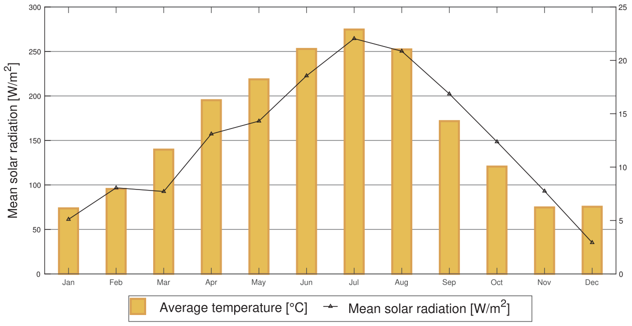


Fig. 13. Weather data of L'Aquila, Italy.

Table 4
Wood-cement blocks properties.

Structural member	Layer description (from inside to outside)	Thermal resistance [(m^2 K)/W]	Total thickness [m]	Total U-value [W/(m^2 K)]
Wall	Wood-cement Concrete EPS and graphite	0.308 0.096 6.774	0.50	0.12
Floor	Wood-cement Polystyrene Screed	0.308 6.000 0.027	0.23	0.28
Pitched roof	Wood-cement Polystyrene Screed Polyurethane resins and polyisocianurate foams	0.308 6.000 0.027 4.000	0.31	0.13

thermal transmittance value. The properties of the blocks used for the simulation model are listed in Table 4.

Therefore, the dynamic simulation model of the use case, shown in Fig. 14, has been created by analyzing the fundamental characteristics of the building (orientation, geometry, structural members, heating system components, air changes with natural ventilation, activity, internal gains, air leakages) and the weather file specifically created for L'Aquila. The comprehensive method was chosen for modeling the heating system, once checked all the characteristics of the components. In order to calibrate the EnergyPlus model, a comparison between simulated and measured thermal energy consumption was performed.

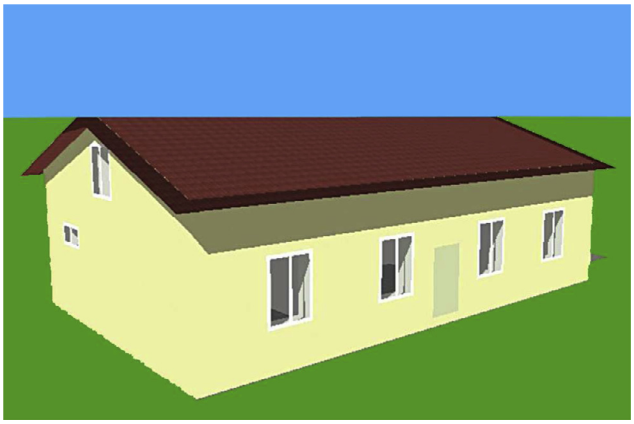
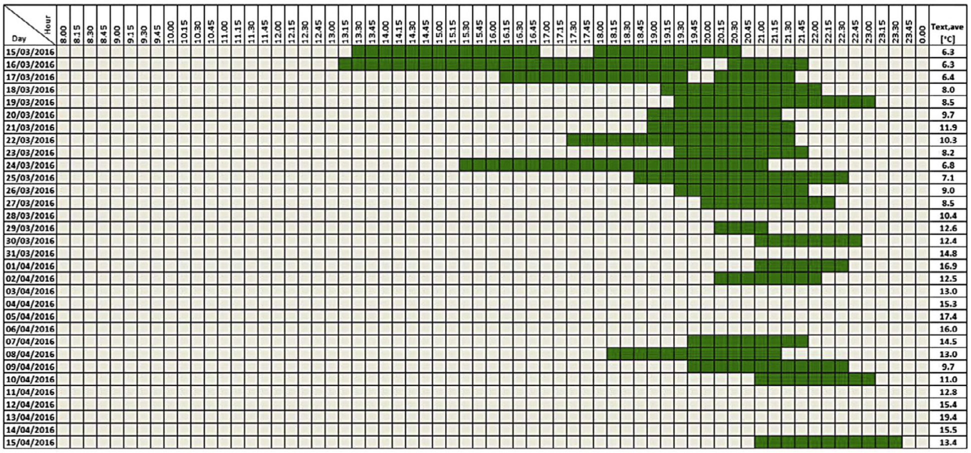


Fig. 14. Virtual model of the use case.



Day	Text,ave [°C]
15/03/2016	6.3
16/03/2016	6.3
17/03/2016	6.4
18/03/2016	8.0
19/03/2016	8.5
20/03/2016	9.7
21/03/2016	11.9
22/03/2016	10.3
23/03/2016	8.2
24/03/2016	6.8
25/03/2016	7.1
26/03/2016	9.0
27/03/2016	8.5
28/03/2016	10.4
29/03/2016	12.6
30/03/2016	12.4
31/03/2016	14.8
01/04/2016	16.9
02/04/2016	12.5
03/04/2016	13.0
04/04/2016	15.3
05/04/2016	17.4
06/04/2016	16.0
07/04/2016	14.5
08/04/2016	13.0
09/04/2016	9.7
10/04/2016	11.0
11/04/2016	12.8
12/04/2016	15.4
13/04/2016	19.4
14/04/2016	15.5
15/04/2016	13.4

Fig. 15. Actual scheduling of the heating system, where white color means switched OFF and green color switched ON, and ($T_{ext,ave}$) is the external average temperature. From midnight to 8*am* the heating system is always switched off. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Considering the off-grid characteristic of the house, the installation of a heat meter was necessary for acquiring the actual energy consumption. Moreover, the heat meter installation has allowed a detailed knowledge of the actual heating system scheduling, shown in Fig. 15, faithfully reproduced in the simulated model. The heat meter, that consists of two thermocouples for flow and return thermal fluid temperatures, a turbine flowmeter and a controller, employed Eq. (17) to calculate the real thermal energy consumption $\dot{Q}$ of the house.

$$\dot{Q} = 0.2777698 \times 10^{-3} \times \rho \times \Delta V \times c_p \times \Delta T \tag{17}$$

In (17) $\dot{Q}$ is the thermal energy consumption [W h], ρ is the water density [kg/m^3], ΔV is the water volume variation [m^3] detected by the turbine flowmeter, $c_p = 4.186$ [kJ/(kg K)] is the water specific heat at constant pressure, ΔT is the difference between flow and return temperatures of the water [K], 0.2777698103 is a dimensionless conversion factor. The considered period goes from March 15, 2016 to April 15, 2016. Following the hourly calibration proposed by the M&V guidelines of ASHRAE [52], also applied in [53,54], a simulated model is calibrated when the mean bias error (MBE) and the coefficient of variation of the root mean square error [CV(RMSE)] are less than acceptable tolerances, respectively equal to ± 10.0% and 30.0%. MBE and CV (RMSE) have been calculated by using Eqs. (18) and (19).

$$MBE\,(\%) = \frac{\sum_{Period}(S{-}M)_{Interval}}{\sum_{Period} M_{Interval}} \times 100 \tag{18}$$

where M is the measured kW h and S is the simulated kW h.

$$\begin{aligned} CV\,(RMSE_{Period}) &= \frac{RMSE_{Period}}{A_{Period}} \times 100 \\ &= \sqrt{\frac{\sum(S-M)^2_{Interval}}{N_{Interval}}} \times \frac{1}{A_{Period}} \times 100 \end{aligned} \tag{19}$$

where A_{Period} is the mean of the measured data for the period, Eq. (20), and $N_{Interval} = 4563$ is the number of time intervals in the monitoring period.

$$A_{Period} = \frac{\sum_{Period} M_{Interval}}{N_{Interval}} \tag{20}$$

The comparison between numerical and experimental data is shown in Fig. 16 and shows a quite good agreement. Therefore, with a MBE equal to 7.38% and a CV(RMSE) equal to 8.37%, the EnergyPlus model of the use case can be considered well calibrated.

6.3. Random forest models

For the closed-loop simulations with DPC we create 2 different sets of models, $S_1 = \{\mathsf{T}^{r_1}_{k+j}, \mathsf{T}^{r_2}_{k+j}, \mathsf{T}^{r_3}_{k+j}, \mathsf{T}^{r_4}_{k+j}\}$ and $S_2 = \{\mathsf{P}_{k+j}, \mathsf{T}^1_{k+j}, \mathsf{T}^2_{k+j}, \mathsf{T}^3_{k+j}, \mathsf{T}^4_{k+j}, j = 1,...,N\}$, using random forests. In each set we have 4 models that describe the room temperature evolution (T^{r_i} and $\mathsf{T}^i, i = 1,2,3,4$) in each of the 4 rooms equipped with temperature sensors. In S_2 we have a model for the power consumption of the house (P). Models in S_1 are created using the classical random forests algorithm with all the features and are used as plant simulator of the house. For this reason, they are computed to give prediction only for time step $k + 1$ and not over the whole horizon N. Models in S_2 are used as predictors over a horizon N in the DPC algorithm and are then created using the methodology provided in Section 3.2. The non-manipulated features in $\mathscr{X}^d$ are the disturbance data (relative humidity, atmospheric pressure, outside air temperature, solar radiation, wind, time of the day and day of the week) and the states (temperature of the 4 rooms). The manipulated feature in $\mathscr{X}^c$ is the flow rate ([m^3/h]). All this features are used to create models in S_1 and the power model in S_2, while

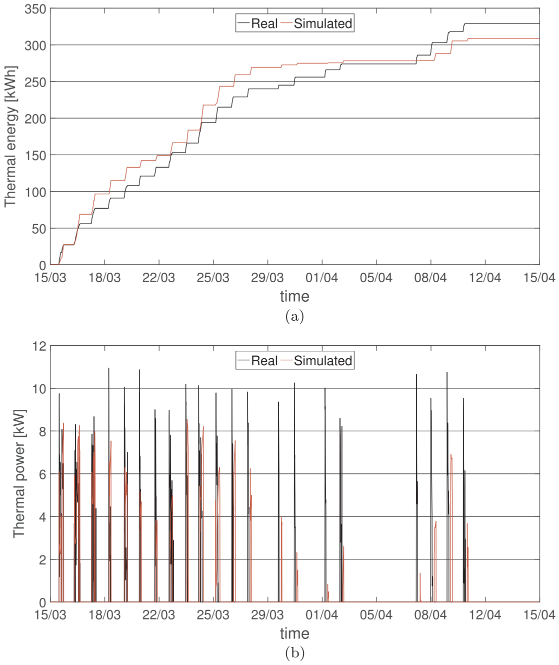


Fig. 16. Comparison between numerical and experimental data. (a) Thermal energy comparison. The accuracy error is 7:38% with MBE definition and 8:37% with CV(RMSE) definition. (b) Thermal power comparison.

Table 5
Models accuracy for power and temperatures models in S_1 and S_2 expressed as 1−NRMSE (%).

Set	Power	T. room 1	T. room 2	T. room 3	T. room 4
S_1		96.58	96.99	97.21	96.62
S_2	92.21	97.38	97.29	96.93	96.81

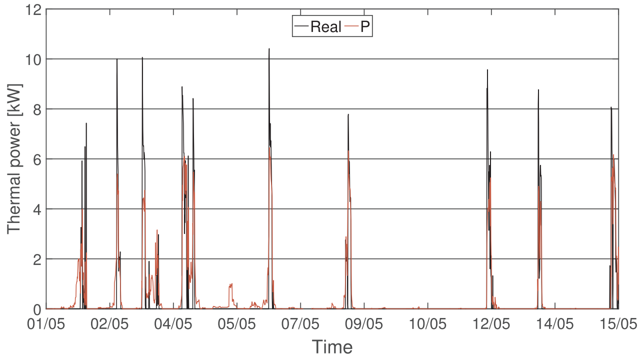


Fig. 17. Power consumption model accuracy validation. The accuracy over the testing period expressed as 1−NRMSE (%) is 92.1%.

disturbance data, state temperature of room i only and flow rate are used to identify $\widehat{\Theta}_{\mathrm{T}^i_j}$ for temperature models in S_2. All the models have been trained on the data from March 11, 2016 to April 26, 2016 and validated on the data from May 1, 2016 to May 15, 2016. The accuracy of these models with respect to real data is shown in Table 5, based on the definition of Normalized Root Mean Square Error (NRMSE).

A graphical comparison is shown in Fig. 17 for the power consumption and Fig. 18 for the temperature of room 1. The plots for the other rooms are omitted since they are very similar.

6.4. DPC and bang-bang controllers

We set up 2 different controllers, DPC and bang-bang controller, to obtain a scheduling policy to switch the radiators ON and OFF in order to keep the temperature of room 3, i.e. the living room, within a comfort range. Since the heating system serves all of the 4 rooms simultaneously, without giving the possibility to control the temperature of each room independently, we setup the problem defining the comfort range only for one room. Other rooms temperatures will follow the scheduling policy. Room 3 has been chosen randomly. In the following we describe the 2 controllers.

DPC. We want to optimize the ON/OFF heating system schedule in order to minimize power consumption of the house while keeping temperature of room 3 within a comfort range. We also allow violations $\epsilon^{min},\epsilon^{max}$ of temperature bounds to guarantee feasibility of the algorithm. We include these violations in the objective function to be minimized.

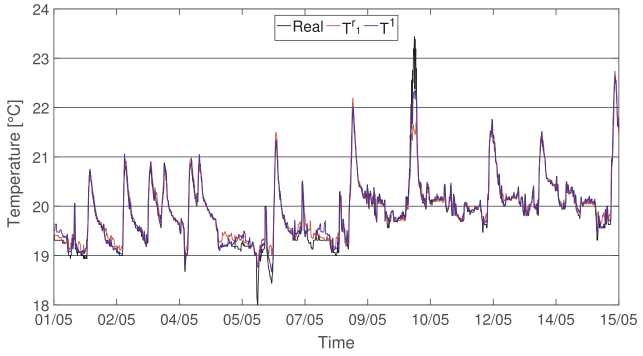


Fig. 18. Temperature model accuracy validation for room 1. The accuracy over the testing period expressed as 1−NRMSE (%) is 96.58%, for the temperature model T^{r_1} in S_1 and 97.3% for the temperature model T^1 in S_2. The accuracy for the other rooms is very similar as can be seen in Table 5.

The problem is set up as follows:

Problem 3.

$$\begin{aligned}
\underset{u_{k+j-1},\epsilon_j^{min},\epsilon_j^{max}}{\text{minimize}}\quad & \sum_{j=1}^{N} Q\mathrm{P}^2_{k+j} + \lambda_{min}\|\epsilon^{min}\|_2 + \lambda_{max}\|\epsilon^{max}\|_2\\
\text{subject to}\quad & \mathrm{P}_{k+j} = \widehat{\Theta}_{\mathrm{P}j}[1,u_k,\ldots,u_{k+j-1}]^{\mathrm{T}}\\
& \mathrm{T}^1_{k+j} = \widehat{\Theta}_{\mathrm{T}^1_j}[1,u_k,\ldots,u_{k+j-1}]^{\mathrm{T}}\\
& \mathrm{T}^2_{k+j} = \widehat{\Theta}_{\mathrm{T}^2_j}[1,u_k,\ldots,u_{k+j-1}]^{\mathrm{T}}\\
& \mathrm{T}^3_{k+j} = \widehat{\Theta}_{\mathrm{T}^3_j}[1,u_k,\ldots,u_{k+j-1}]^{\mathrm{T}}\\
& \mathrm{T}^4_{k+j} = \widehat{\Theta}_{\mathrm{T}^4_j}[1,u_k,\ldots,u_{k+j-1}]^{\mathrm{T}}\\
& \underline{\mathrm{T}}_{k+j}-\epsilon_j^{min} \leqslant \mathrm{T}^3_{k+j} \leqslant \overline{\mathrm{T}}_{k+j} + \epsilon_j^{max}\\
& u_{k+j-1} = \underline{u} \vee u_{k+j-1} = \overline{u}\\
& \epsilon_j^{min},\epsilon_j^{max} \geqslant 0,\ j = 1,\ldots,N.
\end{aligned}\tag{21}$$

The choice of different weights Q,λ_{min} and λ_{max} allows the designer to give more importance to energy consumption rather than temperature comfort and vice versa. In Section 6.5 we will show different performance results considering different weights. Parameters $\underline{u}$ and $\overline{u}$ are respectively minimum and maximum values the heating system can actuate, while $\underline{\mathrm{T}}_{k+j-1}$ and $\overline{\mathrm{T}}_{k+j-1}$ are respectively time varying lower and upper bounds to keep the temperature in a desired range of comfort. Due to the integer variable constraint for input u, the problem is a Mixed Integer Quadratic Programming. For the implementation we use in Section 6.5 Gurobi solver [55] through CVX [56,57].

Bang-bang controller. This is the classical controller widely used in private houses to keep temperature within a comfort range. It switches the heating system ON when the temperature goes under the temperature lower bound and switches it OFF when the temperature goes over the temperature upper bound. The advantage in using this controller is that it is very simple to set up. On the other hand it uses more energy than actually needed to achieve the task.

6.5. Simulation results

We simulated DPC in (21) and the bang-bang controller, in closed-loop with the house models $\mathrm{T}^{r_i}_{k+1}$, $i = 1,\ldots,4$ in S_1. We considered a sampling time of 10 min and chose $N = 4$ as a predictive horizon, i.e. 40 min. From historical data we got that $\overline{u} = 0.35\ \mathrm{m}^3/\mathrm{h}$ when the heating system is ON and obviously $\underline{u} = 0\ \mathrm{m}^3/\mathrm{h}$ when the heating system is OFF. For the temperature comfort range we set a constant upper bound $\overline{\mathrm{T}}_k = 22.5$ °C and a variable lower bound that is $\underline{\mathrm{T}}_k = 21$ °C from 7 am to 9 am when people in the house wake up and go out for work, and from 6 pm to midnight when people come back from work and go to sleep. During other hours when people are either not at home or asleep, we set $\underline{\mathrm{T}}_k = 20$ °C.

We explained in Section 6.1 that the fuel used in the house is vegetable biomass. Therefore, it is not possible to have ON/OFF switching phases too close to each other, differently from a traditional gas boiler, due to the burning process and heat exchange. For this reason we set up both control problems with the constraint that when the heating system is activated it must stay active for at least 20 min. This operating period can be obviously adapted depending on the fuel flow rate.

We ran DPC with 3 different sets of parameters Q,λ_{min} and λ_{max}. Each set allows a different level of temperature bounds violation. In particular, we considered a small ($Q = 100,\lambda_{min} = 3000$ and $\lambda_{max} = 100$), a medium ($Q = 100,\lambda_{min} = 1000$ and $\lambda_{max} = 100$) and a large ($Q = 100,\lambda_{min} = 100$ and $\lambda_{max} = 100$) violation configuration. The simulation period is of 15 days, from 00 h on May 1, 2016 to 00 h on May 15, 2016.

We considered two different simulation conditions to show the robustness of our approach.

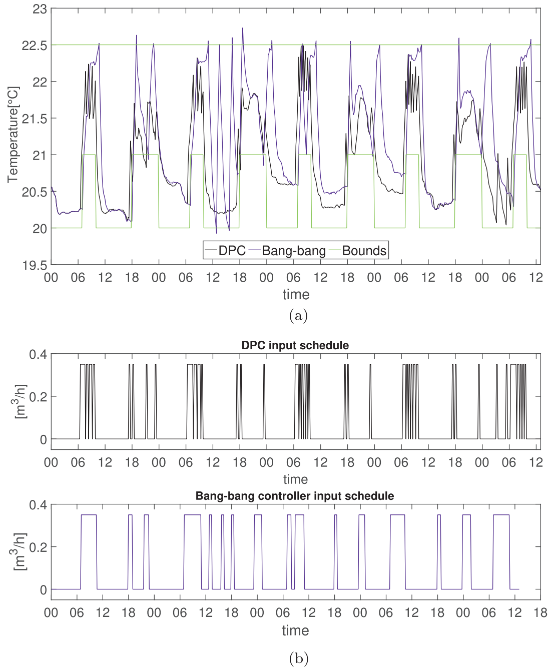
1. In Section 6.5.1, we provide the results considering perfect knowledge of the weather forecast and only focus on robustness with respect to uncertainties due to real data acquisition.
2. In Section 6.5.2, we consider weather forecast subject to uncertainties. We modify Problem 3 by adding Gaussian noise to the perfect forecast and show that the results are close to the perfect weather forecast case.

6.5.1. *Perfect knowledge of the weather forecast*

In this section, we ran the simulations considering perfect knowledge of the disturbance over the horizon, obtaining the following results.

Result 1. The comparison for temperature and input schedule obtained allowing small bound violations in DPC is shown in Fig. 19. For sake of the plot's clarity, the shown period is restricted to 4 days and a half, from 00 h on May 1, 2016 to 13 h on May 5, 2016. The whole period will be used in "Result 2" to provide the bounds violation errors, and in "Result 3" for the energy consumption comparison. We can see that the temperature controlled with DPC does not violate the bounds and if it does then the violation is approximately 0.1 °C. Bang-bang control also presents small bounds violations due to its working principle. We can see how the DPC control law requires the heating system to be ON for less time than the bang-bang one to keep the temperature in the comfort range. We will see in Fig. 21 that this translates to significant energy saving.

Result 2. A comparison of temperature regulation obtained running the DPC with small, medium and large violations is shown in Fig. 20. The results show that with the large violation configuration, that gives more importance to the power consumption minimization than to keep temperature within the bounds, temperature is almost always outside the lower bound during the period when the range is tighter. However the maximum violation is still lower than 1 °C. In Table 6, MBE and CV (RMSE) (expressed in %, and computed over the whole simulative period, i.e. 15 days) violation errors are reported to quantify the bounds violation of DPC, in each of the 3 configurations, and of bang-bang controller. We can see that if we allow very small violations, DPC outperforms bang-bang controller in terms of comfort guarantees.



(a)

(b)

Fig. 19. Comparison of DPC and bang-bang control performance over 5 days of the testing period for room 3. (a) Temperature variation obtained with DPC and bang-bang controller. DPC controller allows almost no violation, so guaranteeing better comfort than bang-bang controller. (b) Input schedules obtained from DPC and bang-bang controller. DPC keeps the heating system ON for less time than bang-bang controller, hence saving energy, and guarantees better thermal comfort.

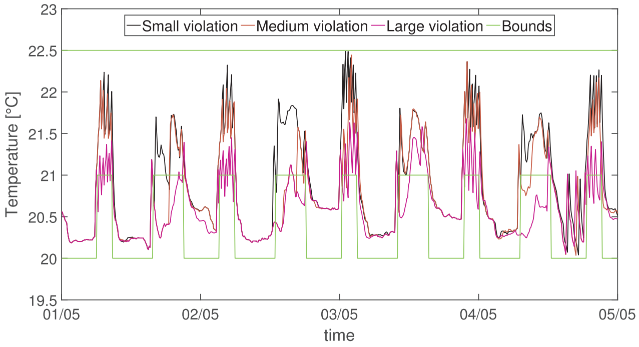


Fig. 20. Comparison, in terms of comfort, of DPC control performance simulated over 5 days of the testing period with 3 different violation configurations: small, medium and large. With the large violation configuration, the temperature is almost always outside of the lower bound when this is tighter. However the maximum bound violation is less than 1 °C. With the small violation configuration, the temperature is always within the bound with very few exceptions. However, when it happens, the violation is lower than 0.1 °C. The medium violation configuration gives a temperature bound violation that is in the middle with respect to the small and the large ones.

Result 3. In Fig. 21, using the thermal energy consumption model derived in Section 6.2 using EnergyPlus, we show how DPC outperforms the bang-bang controller also in terms of energy consumption and how the bounds violations allow us to save more energy. In this case, since the plot is clear and we are interested in showing energy saving on a long period, we ran the simulations over the whole testing period, i.e. 15 days of May, from May 1, 2016 to May 15, 2016. We observe that the energy consumption associated to the bang-bang control strategy is approximately equal to 177 kW h. If we want to keep the temperature within a comfort range, only allowing small violations, the use of DPC produce an energy saving of 45 kW h, that corresponds to the 25.4% over a period of 15 days. In case of medium violations get an energy saving of 57 kW h, that is 32.2%. Instead if we allow a large violation over the same period, the energy saving is of 87 kW h, that corresponds to the 49.2%.

6.5.2. *Weather forecast subject to uncertainty*

In this section we study the robustness of our approach to noisy weather forecast. To this aim, we add noise with a zero-mean Gaussian distribution to the prediction variables $\widetilde{d}(k+1),\ldots,\widetilde{d}(k+N)$ used to obtain the parameters $\widehat{\Theta}_{\cdot,j}, j = 0,\ldots,N$ in the DPC Problem (21) (i.e. the disturbance in input in the blue rectangle in Fig. 8), while we use the

Table 6

Lower Bound Violation (LBV) and Upper Bound Violation (UBV) errors expressed as MBE% and CV(RMSE)% for DPC Small Violation (DPC-SV), DPC Medium Violation (DPC-MV), DPC large Violation (DPC-LV) and bang-bang controller.

CONTROLLER	LBV MBE	LBV RMSE	UBV MBE	UBV RMSE
DPC−SV	0.013	0.092	0	0
DPC−MV	0.165	0.479	0	0
DPC−LV	0.410	0.733	0	0
Bang−bang	0.0485	0.265	0.0063	0.040

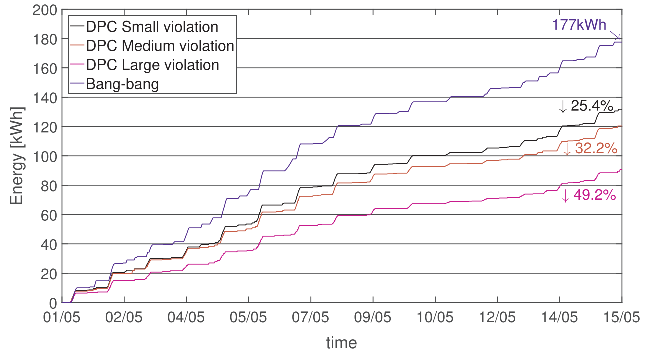


Fig. 21. Comparison of DPC and bang-bang controller performance over 15 days of the testing period with different violation configurations, in terms of thermal energy saving using EnergyPlus model. Using bang-bang controller the house energy consumption after 15 days is of 177 kW h. DPC with small, medium and large violation configurations allows an energy saving of 25.4%, 32.3% and 49.2% respectively.

true values to simulate the process (i.e. the disturbance in input in the box “Plant” in the red rectangle in Fig. 8).

We summarize the mean and the standard deviation of the Gaussian noises that we add to each variable in Table 7. For example, a standard deviation of 0.5 °C on the outside air temperature means that the error on the predicted temperature lies within the range of ± 1.5 °C with a probability of 99%. Since “Time of the day” and “Day of the week” are perfectly known, we clearly do not add any noise to these variables. Table 7 also provides the range of values of the variables in the historical disturbances dataset to emphasise that the standard deviation we choose adds significant error to the forecast.

The following paragraphs provide comments on the change in performance due to inaccurate forecast on bounds violation and on energy consumption.

Result 1. Fig. 22 shows a comparison of the DPC results in terms of temperature control for thermal comfort between the perfect and the noisy forecast. For the sake of clarity, we show only $4\frac{1}{2}$ days of the 15 days simulation period. We split the plot into 3 sub-figures referring to the small, medium and large violation cases. We see that, except for an isolated case in the medium violation case (slightly before “02/05”), we have a small performance deterioration in terms of bounds violation. The effect is more prominent in the large violation case. However, the maximum violation is still lower than 1 °C, as in the case of perfect forecast. We report the violation errors in terms of MBE and CV(RMSE) (in %) computed for the whole simulative period (15 days) in Table 8. Comparing them with the values obtained earlier in Table 6, we see that the performance of DPC is robust to noisy predictions despite a large error in the weather forecast.

Result 2. In Fig. 23, we show the difference in terms of energy consumption considering the perfect weather forecast (solid line) and the noisy forecast (dashed line). We first recall that our optimization function is a weighted sum of energy and thermal comfort: when the weather forecast is affected by inaccuracies, the MPC solution (and thus the control input) changes depending upon the prediction errors.

Table 7
Mean and standard deviation of the Gaussian noise added on the weather forecast data, and the range of the variables in the historical dataset.

Variable	Range	Mean	Deviation
Outside temperature	[0, 31]	0	0.5
Wind	[0, 5]	0	0.25
Atmospheric pressure	[990, 1030]	0	50
Relative Humidity	[20, 91]	0	5
Solar Radiation	[0, 1000]	0	50
Time of the day	[0, 23]	0	0
Day of the week	[1, 7]	0	0

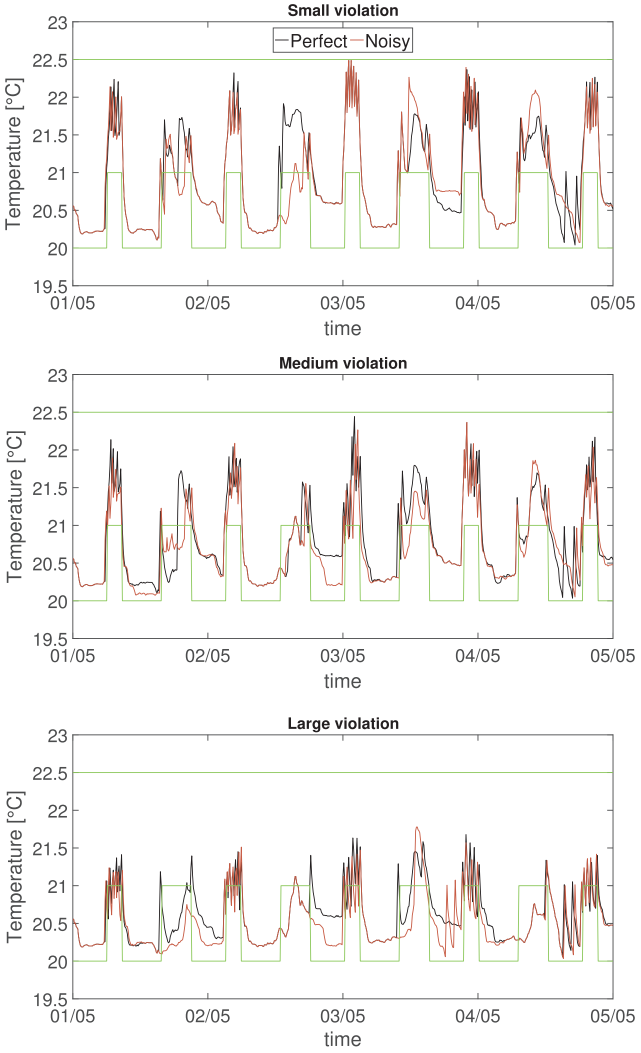


Fig. 22. Comparison of DPC considering perfect and noisy weather forecast. Small, medium and large violation cases are compared.

Table 8
Lower Bound Violation (LBV) and Upper Bound Violation (UBV) errors expressed as MBE% and CV(RMSE)% for DPC Small Violation (DPC-SV), DPC Medium Violation (DPC-MV), DPC large Violation (DPC-LV) and bang-bang controller considering noisy weather forecast.

CONTROLLER	LBV MBE	LBV RMSE	UBV MBE	UBV RMSE
DPC−SV	0.034	0.142	0	0
DPC−MV	0.178	0.453	0	0
DPC−LV	0.478	0.891	0	0

We immediately observe that the energy usage in the ideal and perturbed cases is very close, especially in the small violation case. Our interpretation for this is that the DPC, in the small violation case, keeps the temperature within the bounds with an extremely small error in bounds violation. The only exception can be seen before May 3rd (03/05) where, due to a large prediction error, the temperature violates the bounds: as a consequence the controller uses more energy (with respect to the perfect forecast knowledge case) to bring the temperature in the bounds, as evidenced by the blacked lines of Fig. 23. In the medium and large violation cases, instead, when the temperature violates the bounds the DPC does not strongly increase the energy usage as the violation tolerance is more relaxed: this produces a general reduction of energy

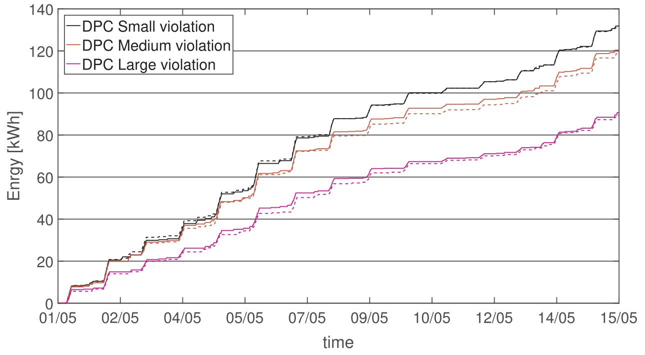


Fig. 23. Comparison of DPC in terms of thermal energy savings, considering perfect (full) and imperfect (dashed) weather forecast, over 15 days of the testing period with different violation settings. The performance is similar, showing the robustness of DPC.

usage in the ”noisy” simulations, as evidenced by the purple and red lines of Fig. 23.

The above results show the robustness of the DPC with respect to errors in the weather forecast.

7. Conclusion

To overcome the difficulties associated with the model identification in Model Predictive Control (MPC), we introduce a novel idea for predictive control using data: Data-driven model Predictive Control (DPC). Data-driven control is based on non-physical (black-box) models, therefore they cannot be integrated with most of the classical control approaches. The goal is to create data-driven models that are suitable for receding horizon control. To this aim, we present two algorithms, based on trees and random forests, to create control-oriented models for DPC. We then apply DPC to three different case studies to demonstrate its strength.

1. ***Comparison with MPC.*** We compare the performance of our DPC 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.
2. ***Application to Demand Response.*** 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%.
3. ***Application to optimal heating system scheduling.*** We demonstrate robustness of our method to uncertainties due to real data acquisition and weather forecast inaccuracies by implementing and testing DPC on historical data from an off-grid house located in L'Aquila, Italy. We derive a predictive model on such real data and design the optimal ON/OFF scheduling for the heating system in order to save energy while guaranteeing thermal comfort for the occupants. We compare the total amount of energy saved with respect to the classical bang-bang controller (widely used in houses for temperature control) using an EnergyPlus model built specifically for the house. We show that we can perform an energy saving that ranges from 25.4% (if we guarantee thermal comfort i.e. strictly respect the desired temperature range in the rooms) to 49.2% (if we allow small violation in the desired temperature range). Finally, we test the robustness of our method to uncertainties in the real data acquisition and weather forecast.

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. In general, DPC is immensely valuable in situations where first principles based modeling cost is extremely high.

7.1. *Practical challenges and future work*

1. ***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 is required, and how should the sampling (functional testing) 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 [58].
2. ***Stability:*** While the buildings are inherently stable, many other applications, such as power networks, 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.

Acknowledgment

This work was supported partially by the Italian Government under Cipe resolution n. 135 (Dec. 21, 2012), project *INnovating City Planning through Information and Communication Technologies* (INCIPICT).

The authors would like to thank Xiaojing Zhang, a PostDoctoral Researcher at the University of California, Berkeley, for providing the bilinear building model, Iole Nardi, a PostDoctoral Researcher at the University of L'Aquila, for the help in obtaining the real data from the house, and Manfred Morari, for his feedback on DPC.

Appendix A. Regression trees

In this appendix we explain how Regression Trees are built using an example adapted from [36]. Tree-based methods partition the feature space into a set of rectangles (more formally, hyper-rectangles) and then fit a simple model in each one. They are conceptually simple yet powerful. Let us consider a regression problem with continuous response $\mathscr{Y} = \{Y\}$ and 2 predictors $\mathscr{X} = \{X_1,X_2\}$, each taking values in the unit interval. The top left plot of Fig. 24 shows a partition of the feature space by lines that are parallel to the coordinate axes. In each partition element, we can model Y with a different constant. However, there is a problem: although each partitioning line has a simple description like $X_1 = k$, some of the resulting regions are complicated to describe. To simplify things, we can restrict ourselves to only consider recursive binary partitions, like the ones shown in the top right plot of Fig. 24. We first split the space into two regions, and model the response by the mean of Y in each region. We choose the variable and split-point to achieve the best prediction for Y. Then one or both of these regions are split into two more regions, and this process is continued, until some stopping rule is applied. This is the ”recursive partitioning” part of the algorithm. For example, in the top right plot of Fig. 24, we first split at $X_1 = t_1$. Then the region $X_1 \leqslant t_1$ is split at $X_2 = t_2$ and the region $X_1 > t_1$ is split at $X_1 = t_3$. Finally, the region $X_1 > t_3$ is split at $X_2 = t_4$. The result of this process is a partition of the data-space into the five regions (or leaves) $R_1,R_2,...,R_5$. The corresponding regression tree model, $\mathscr{T}$, predicts Y with a constant, c_i, in region R_i i.e.,

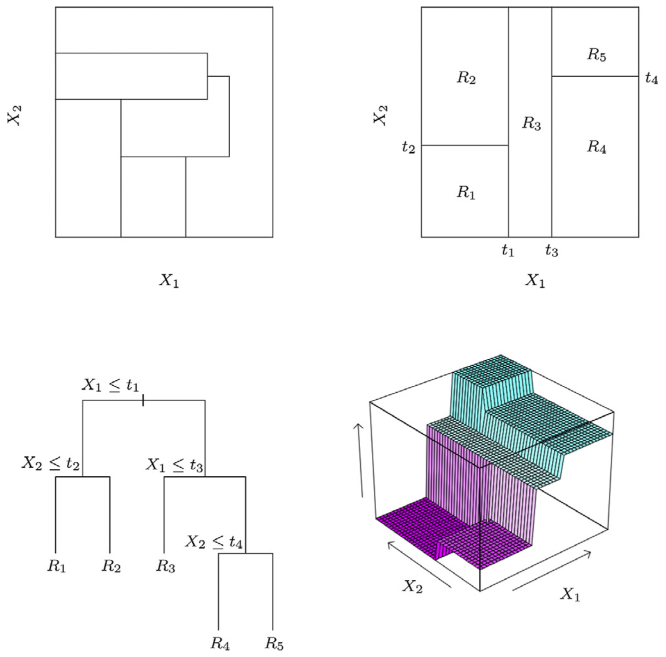


Fig. 24. Top right: 2D feature space by recursive binary splitting. Top left: partition that cannot be obtained from recursive binary splitting. Bottom left: tree corresponding to the partition. Bottom right: perspective plot of the prediction surface.

$$f_{\mathscr{T}}(X_1,X_2) = \sum_{i=1}^{5} c_i I\{(X_1,X_2) \in R_i\}, \tag{A.1}$$

where $I\{X \in R\}$ is a function that is equal to 1, if $X \in R$, and 0, otherwise. This same model can be represented by the binary tree shown in the bottom left of Fig. 24. The full data-set sits at the top of the tree. Observations satisfying the condition at each node are assigned to the left branch, and the others to the right branch. The terminal nodes or leaves of the tree correspond to the regions $R_1,R_2,\ldots,R_5$. The bottom right plot of Fig. 24, shows the perspective plot of the regression surface obtained as a result of building a regression tree with the 5 constants c_i, $i = 1,2,3,4,5$.

Node splitting criteria. Now, a first question is: *how to grow a regression tree?* Suppose our dataset, $(\mathscr{X},\mathscr{Y})$, consisting of p features, i.e. $\mathscr{X} = \{X_1,X_2,\ldots,X_p\}$, and one response variable, i.e. $\mathscr{Y} = \{Y\}$. Suppose we have $|(\mathscr{X},\mathscr{Y})| = n$ observations (samples): $(x_i,y_i), i = 1,2,\ldots,n$, with $x_i = (x_{i1},x_{i2},\ldots,x_{ip})$. For regression trees we adopt the sum of squares as our splitting criteria, i.e. a variable at a node will be split if it minimizes the following sum of squares between the predicted response and the actual output variable:

$$\sum_i (y_i - f_{\mathscr{T}}(x_i))^2. \tag{A.2}$$

The best response c_i (from Eq. (A.1) for the partition R_i), is just the average of output samples in the region R_i, i.e.

$$c_i = avg(y_i | x_i \in R_i). \tag{A.3}$$

Finding the best binary partition in terms of minimum sum of squares is generally computationally infeasible. A greedy algorithm is used instead. Starting with all of the data, consider a splitting variable j and split point s, and define the following pair of left (R_L) and right (R_R) half-planes

$$\begin{aligned} R_L(j,s) &= \{X | X_j \leqslant s\}, \\ R_R(j,s) &= \{X | X_j > s\} \end{aligned} \tag{A.4}$$

The splitting variable j and the split point s is obtained by solving the following minimization:

$$\min_{j,s}\left[\min_{c_L} \sum_{x_i \in R_L(j,s)} (y_i - c_L)^2 + \min_{c_R} \sum_{x_i \in R_R(j,s)} (y_i - c_R)^2\right] \tag{A.5}$$

where, for any choice of j and s, the inner minimization in Eq. (A.5) is solved using

$$\begin{aligned} c_L &= \mathrm{avg}(y_i | x_i \in R_L(j,s)), \\ c_R &= \mathrm{avg}(y_i | x_i \in R_R(j,s)). \end{aligned} \tag{A.6}$$

For each splitting variable X_j, the determination of the split point s can be done very quickly and hence by scanning through all of the inputs (X_i's), the determination of the best pair (j,s) is feasible. Having found the best split, we partition the data into the two resulting regions and repeat the splitting process on each of the two regions. Then this process is repeated on all of the resulting regions.

Rather than splitting each node into just two regions at each stage, we might consider multiway splits into more than two groups. While this can sometimes be useful, it is not a good general strategy. The problem is that multiway splits fragment the data too quickly, leaving insufficient data at the next level down. Hence we would want to use such splits only when needed. Also multiway splits can be achieved by a series of binary splits.

Stopping criteria and pruning. At this point, the second question is: *How large should we grow the tree?* Every recursive algorithm needs to know when it's done, i.e. it requires a stopping criteria. For regression trees this means when to stop splitting the nodes. A very large tree might over fit the data, while a small tree might not capture the important structure. Tree size is a tuning parameter governing the models complexity, and the optimal tree size should be adaptively chosen from the data. One approach is to split tree nodes only if the decrease in sum-of-squares due to the split exceeds some threshold. However, this strategy is myopic, since a seemingly worthless split might lead to a very good split below it. A preferred strategy is to grow a large tree, stopping the splitting process only when some minimum number of data points at a node (`MinLeaf`) is reached. Then this large tree is pruned using cost-complexity pruning methods.

Define a subtree $\mathscr{T}_{sub} \subset \mathscr{T}$ to be any tree that can be obtained by pruning $\mathscr{T}$, i.e. collapsing any number of its non-terminal nodes. Let node i corresponding to the partition R_i. $|\mathscr{T}_{sub}|$ denotes the number of terminal nodes in $\mathscr{T}_{sub}$ Define,

$$\begin{aligned} N_i &= \#\{x_i \in R_i\},\\ c_i &= \frac{1}{N_i}\sum_{x_i \in R_i} y_i,\\ Q_i(T) &= \frac{1}{N_i}\sum_{x_i \in R_i} (y_i - c_i)^2 \end{aligned} \tag{A.7}$$

where N_i is the number of samples in the partition R_i,c_i is the estimate of Y within R_i and $Q_i(T)$ is the mean square error of the estimate c_i. The cost complexity criteria is then defined as:

$$C_\alpha(\mathscr{T}_{sub}) = \sum_{i=1}^{|\mathscr{T}_{sub}|} N_i Q_i(T) + \alpha|\mathscr{T}_{sub}| \tag{A.8}$$

The goal is to find, for each α, the subtree $\mathscr{T}_\alpha \subset \mathscr{T}$ to minimize $C_\alpha(\mathscr{T}_{sub})$. The tuning parameter $\alpha \geqslant 0$ governs the trade off between tree size and its goodness of fit to the data. For each α one can show that there is a unique smallest subtree $\mathscr{T}_\alpha$ that minimizes $C_\alpha(\mathscr{T}_{sub})$ [59]. Estimation of α is achieved by cross-validation.

References

[1] Sturzenegger D, Gyalistras D, Morari M, Smith RS. Model predictive climate control of a swiss office building: implementation, results, and cost–benefit analysis. IEEE Trans Control Syst Technol 2016;24(1):1–12.

[2] Žáčeková E, Váňa Z, Cigler J. Towards the real-life implementation of MPC for an office building: identification issues. Appl Energy 2014;135:53–62.

[3] New JR, Sanyal J, Bhandari M, Shrestha S. Autotune E+ building energy models. Proc 5th Natl SimBuild IBPSA-USA 2012:1–3.

[4] Yin R, Kiliccote S, Piette M. Linking measurements and models in commercial buildings: a case study for model calibration and demand response strategy evaluation. Energy Build 2016;124:222–35.

[5] Christantoni D, Oxizidis S, Flynn D, Finn D. Implementation of demand response strategies in a multi-purpose commercial building using a whole-building simulation model approach. Energy Build 2016;131:76–86.

[6] Li X, Wen J. Building energy consumption on-line forecasting using physics based system identification. Energy Build 2014;82:1–12.

[7] Harb H, Boyanov N, Hernandez L, Streblow R, Müller D. Development and validation of grey-box models for forecasting the thermal response of occupied buildings. Energy Build 2016;117:199–207.

[8] Li X, Wen J. System identification and data fusion for on-line adaptive energy forecasting in virtual and real commercial buildings. Energy Build 2016;129:227–37.

[9] Shakouri G, Kazemi A. Multi-objective cost-load optimization for demand side management of a residential area in smart grids. Sustain Cities Soc 2017;32:171–80.

[10] Yoon J, Bladick R, Novoselac A. Demand response for residential buildings based on dynamic price of electricity. Energy Build 2014;80:531–41.

[11] Salakij S, Yu N, Paolucci S, Antsaklis P. Model-based predictive control for building energy management. I: energy modeling and optimal control. Energy Build 2016;133:345–58.

[12] Li X, Malkawi A. Multi-objective optimization for thermal mass model predictive control in small and medium size commercial buildings under summer weather conditions. Energy 2016;112:1194–206.

[13] Hu M, Xiao F, Wang L. Investigation of demand response potentials of residential air conditioners in smart grids using grey-box room thermal model. Appl Energy 2017;207:324–35.

[14] Cecconi F, Manfren M, Tagliabue L, Ciribini A, Angelis ED. Probabilistic behavioral modeling in building performance simulation: a Monte Carlo approach. Energy Build 2017;148:128–41.

[15] Li X, Wen J, Bai E-W. Developing a whole building cooling energy forecasting model for on-line operation optimization using proactive system identification. Appl Energy 2016;164:69–88.

[16] Safa M, Allen J, Shahi A, Haas C. Improving sustainable office building operation by using historical data and linear models to predict energy usage. Sustain Cities Soc 2017;29:107–17.

[17] Neto A, Fiorelli F. Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. Energy Build 2008;40:2169–76.

[18] Magnier L, Haghighat F. Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and artificial neural network. Build Environ 2010;45:739–46.

[19] Candanedo L, Feldheim V, Deramaix D. Data driven prediction models of energy use of appliances in a low-energy house. Energy Build 2017;140:81–97.

[20] Ascione F, Bianco N, Stasio CD, Mauro G, Vanoli G. Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: a novel approach. Energy 2017;118:999–1017.

[21] Macarulla M, Casals M, Forcada N, Gangolells M. Implementation of predictive control in a commercial building energy management system using neural networks. Energy Build 2017;151:511–9.

[22] Costanzo G, Iacovella S, Ruelens F, Leurs T, Claessens B. Experimental analysis of data-driven control for a building heating system. Sustain Energy Grids Netw 2016;6:81–90.

[23] Ferreira P, Ruano A, Silva S, ao EC. Neural networks based predictive control for thermal comfort and energy savings in public buildings. Energy Build 2012;55:238–51.

[24] Afram A, Janabi-Sharifi F, Fung A, Raahemifar K. Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: a state of the art review and case study of a residential HVAC system. Energy Build 2017;141:96–113.

[25] Behl M, Smarra F, Mangharam R. DR-advisor: a data-driven demand response recommender system. Appl Energy 2016;170:30–46.

[26] Jain A, Smarra F, Behl M, Mangharam R. Data-driven model predictive control with regression trees-an application to building energy management. ACM Trans Cyber-Phys Syst 2018;2(1):1–21.

[27] Hou Z-S, Wang Z. From model-based control to data-driven control: survey, classification and perspective. Inf Sci 2013;235:3–35.

[28] EnergyPlus™. EnergyPlus. < https://energyplus.net/ > .

[29] Trnsys, A transient system simulation program. University of Wisconsin.

[30] Ernst D, Glavic M, Capitanescu F, Wehenkel L. Reinforcement learning versus model predictive control: a comparison on a power system problem. IEEE Trans Syst Man Cybern Part B (Cybern) 2009;39(2):517–29.

[31] Jain A, Behl M, Mangharam R. Data predictive control for building energy management. Proceedings of the 2017 american control conference. IEEE; 2017.

[32] Jain A, Smarra F, Mangharam R. Data predictive control using regression trees and ensemble learning. Proceedings of the 2017 conference on decision and control. IEEE; 2017.

[33] Gwerder M, Gyalistras D, Sagerschnig C, Smith R, Sturzenegger D. Final report: use of weather and occupancy forecasts for optimal building climate control part I: demonstration (opticontrol-I). Tech. Rep. Zürich, Switzerland: Autom. Control Lab., Dept. Elect. Eng. ETH Zürich; 2013. < http://www.opticontrol.ethz.ch/Lit/Gwer_13_Rep-OptiCtrl2FinalRep.pdf > .

[34] Wetter M, Nouidui T, Haves P. Building control virtual testbed (BCVTB)-v1. 5.0. Lawrence Berkeley National Laboratory; 2015.

[35] Dorf R, Bishop RH. Modern control systems. Pearson; 2011.

[36] Hastie T, Tibshirani R, Friedman J, Hastie T, Friedman J, Tibshirani R. The elements of statistical learning vol. 2. Springer; 2009.

[37] Petersen S, Bundgaard K. The effect of weather forecast uncertainty on a predictive control concept for building systems operation. Appl Energy 2014;116:311–21.

[38] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning vol. 1. Berlin: Springer Series in Statistics Springer; 2001.

[39] Gyalistras D, Gwerder M. Use of weather and occupancy forecasts for optimal building climate control (opticontrol): two years progress report – main report. Zug, Switzerland: Terrestrial Systems Ecology ETH Zurich R&D HVAC Products, Building Technologies Division, Siemens Switzerland Ltd; 2010.

[40] Ma Y, Matuško J, Borrelli F. Stochastic model predictive control for building HVAC systems: complexity and conservatism. IEEE Trans Control Syst Technol 2015;23(1):101–16.

[41] Oldewurtel F. Stochastic model predictive control for energy efficient building climate control [Ph.D. thesis]. ETH Zurich; 2011.

[42] Oldewurtel F, Parisio A, Jones CN, Gyalistras D, Gwerder M, Stauch V, et al. Use of model predictive control and weather forecasts for energy efficient building climate control. Energy Build 2012;45:15–27.

[43] Merkblatt S. 2024: standard-nutzungsbedingungen für die energie-und gebäudetechnik. Zürich: Swiss Society of Engineers and Architects; 2006.

[44] Mayne DQ, Rawlings JB, Rao CV, Scokaert PO. Constrained model predictive control: stability and optimality. Automatica 2000;36(6):789–814.

[45] Ilog I. IBM ILOG CPLEX optimizer-highperformance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming; 2012.

[46] Deru M, Field K, Studer D, Benne K, Griffith B, Torcellini P, et al. US department of energy commercial reference building models of the national building stock; 2011.

[47] Nardi I, de Rubeis T, Buzzi E, Sfarra S, Ambrosini D, Paoletti D. Modeling and optimization of the thermal performance of a wood-cement block in a low-energy house construction. Energies 2016;9(9):677–94.

[48] Pantoli L, Muttillo M, de Rubeis T, Ferri G, Stornelli V, Nardi I. Digital multi-probe temperature monitoring system for long-term on field measurements. In: Eurosensor conference; 2017.

[49] DesignBuilder Software Ltd. DesignBuilder. < https://www.designbuilder.co.uk/ > .

[50] CETEMPS. Center of excellence. < http://cetemps.aquila.infn.it/ > [accessed july 19, 2017].

[51] Peel M, Finlayson B, McMahon T. Updated world map of the Köppen-Geiger climate classification. Hydrol Earth Syst Sci Discuss 2007;4(2):439–73.

[52] U.S. department of energy, federal energy management program. M&V guidelines: measurements and verification for performance-based contacts version 4.0; 2015.

[53] Mustafaraj G, Marini D, Costa A, Keane M. Model calibration for building energy efficiency simulation. Appl Energy 2014;130:72–85.

[54] Raftery P, Keane M, O'Donnell J. Calibrating whole building energy models: an evidence-based methodology. Energy Build 2011;43(9):2356–64.

[55] I. Gurobi Optimization. Gurobi optimizer reference manual; 2015. < http://www.gurobi.com > .

[56] I. CVX Research. CVX: matlab software for disciplined convex programming, version 2.0; Aug. 2012. < http://cvxr.com/cvx > .

[57] Grant M, Boyd S. Graph implementations for nonsmooth convex programs. In: Blondel V, Boyd S, Kimura H, editors. Recent advances in learning and control, lecture notes in control and information sciences. Springer-Verlag Limited; 2008. p. 95–110 < http://stanford.edu/boyd/graph_dcp.html > .

[58] Jain A, Nghiem TX, Morari M, Mangharam R. Learning and control using Gaussian processes. Proceedings of the 9th international conference on cyber-physical systems (ICCPS). ACM/IEEE; 2018.

[59] Ripley BD. Pattern recognition and neural networks. Cambridge: Cambridge University Press; 1996.