Data-driven Switched Affine Modeling for Model Predictive Control [⋆]

Francesco Smarra [∗,∗∗] **Achin Jain** [∗∗] **Rahul Mangharam** [∗∗]
Alessandro D'Innocenzo [∗]

[∗] *Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Via Vetoio, 67100 L'Aquila, Italy (e-mail: [francesco.smarra,alessandro.dinnocenzo]@univaq.it).*
[∗∗] *Department of Electrical and Systems Engineering, University of Pennsylvania, 200 South 33rd Street, 19104 Philadelphia, PA, USA (e-mail: [fsmarra,achinj,rahulm]@seas.upenn.edu)*

Abstract: Model Predictive Control (MPC) is a well consolidated technique to design optimal control strategies, leveraging the capability of a mathematical model to predict the system's behavior over a predictive horizon. However, building physics-based models for large scale systems, such as buildings and process control, can be cost and time prohibitive. To overcome this problem we propose in this paper a methodology to exploit machine learning techniques (i.e. regression trees and random forests) in order to build a state-space switched affine dynamical model of a large scale system only using historical data. Finite Receding Horizon Control (RHC) setup using control-oriented data-driven models based on regression trees and random forests is presented as well. A comparison with an optimal MPC benchmark and a related methodology is provided on an energy management system to show the performance of the proposed modeling framework. Simulation results show that the proposed approach is very close to the optimum and provides better performance with respect to the related methodology in terms of cost function optimization.

Keywords: Data-driven modeling, data-driven model predictive control, machine learning, switched systems.

---

1. INTRODUCTION

Model Predictive Control (MPC) is a well known control strategy used to design optimal control actions to optimize desired system performance while guaranteeing a desired system behavior. To provide such an optimal control strategy, MPC uses mathematical models to predict system behavior over a horizon. MPC has been widely applied in past years to control a large variety of systems, as for example energy systems such as smart buildings, smart grids and power systems Ma et al. (2015); Oldewurtel et al. (2012); Maasoumy et al. (2014); Iovine et al. (2017); Venkat et al. (2008); Kennel et al. (2013). However, creating a physics-based models for large-scale systems, as the ones mentioned above, can be cost and time prohibitive Žáčeková et al. (2014); Sturzenegger et al. (2015). To overcome this issue, a possibility is to use machine learning algorithms to create models using only historical data available to the system. Several works, Macarulla et al. (2017); Afram et al. (2017); Ferreira et al. (2012) and others, deal with this problem and use machine learning algorithms to construct data-driven models to be used for control. Nevertheless, none of these works addressed the problem of creating data-driven state models only using

data, that could be used to predict the state evolution over a horizon in an MPC problem formulation. To the best of authors' knowledge, such problem has been addressed for the first time in Behl et al. (2016), where data-driven models, built using regression trees-based algorithms, have been developed to enable one-step lookahead closed-loop predictive control for the demand-response problem in buildings. This approach has been extended in Jain et al. (2017), where the authors proposed a regression trees and random forests-based Data Predictive Control (DPC) strategy, that implements finite Receding Horizon Control (RHC) over an horizon of arbitrary length. However, both the aforementioned approaches make use of data-driven *static* models, where the input-output relation is represented by affine functions. As a consequence, such modeling framework does not take into account the presence of an internal state evolution and loose the information, over the prediction horizon, of the past inputs applied to the system. This translate in a lack of internal consistency of the model and in a loss of control performance, as we will show in the simulations. Furthermore, due to the lack of an internal state, system's properties, such as stability, structural properties, etc., cannot be studied.

Main Contribution. The goal of this paper is to provide a new methodology to create a state-space switched affine dynamical model of a system using only historical data, leveraging regression trees and random forests, and without any knowledge about its physics-based modeling. We then use this model to setup an MPC problem to optimally control the system's behavior. The idea is to bridge ma-

chine learning and control to build control-oriented models that can be used to provide system guarantees. This modeling technique has the advantage to keep the simplicity of the model identification methodology used to create data-driven models for DPC, while guaranteeing the presence of an internal state typical of state-space models. More precisely, we derive a Switched Affine (SA) data-driven model using regression trees and random forests algorithms. One of the main reasons for choosing regression trees resides in the fact that, other than providing good model accuracy, they are by design highly interpretable, which is a fundamental desirable quality in any model. We then use this technique to setup an MPC problem both for the model obtained using Regression Trees (SAMPC-RT) and for the model obtained using Random Forests (SAMPC-RF). To show the validity of our approach, we compare our methodology with DPC using a bilinear model of a building with 12 states, 4 inputs and 8 disturbances, whose parameters were identified using experiments on a building in Switzerland (Oldewurtel (2011)). As an optimal reference to compare our approach to, we consider a benchmark MPC controller with perfect knowledge of the model. We show that the proposed approach outperform DPC in terms of cost function minimization, providing results that are closer to the optimum. Although we said this approach fits well for large-scale systems, where obtaining a mathematical model can be prohibitive, we specifically consider in this paper the aforementioned model, which we have a formulation of, otherwise a comparison with the optimal solution would not be possible. The application to more complex systems is part of our future work. In Section 2 we briefly recall the DPC formulation. In Section 3, as the main contribution of this paper, we present a new methodology to derive, starting from a set of data, switched affine models to capture system's dynamics, and setup a Model Predictive Control problem that uses such data-driven model. In Section 4 we provide a comparison of the proposed methodology to DPC and to the MPC benchmark in a building automation case study. A more complete version of this paper can be found in the technical report Smarra et al. (2018b). We denote by I and $\mathbf{0}$ respectively the identity matrix and a matrix with all the entries equal to 0 of appropriate dimensions, by $det(A)$ the determinant of matrix A, by $\biguplus$ the disjoint union, and by $|S|$ the cardinality of the set S.

2. DATA PREDICTIVE CONTROL

In this section we briefly recall first the concept of regression trees partitioning, and then the concept of DPC provided in Jain et al. (2017). This will be useful to both better understand and compare the approach we want to propose. The main idea is to create system models using machine learning algorithms (in the specific case, regression trees and random forests) starting from data, that can be used in a receding horizon control scheme. Let a dataset $(\mathcal{X}, \mathcal{Y})$, where $\mathcal{X} = \{s_1^x, \ldots, s_{|\mathcal{X}|}^x\}$ is the set of predictor variables (or features) samples and $\mathcal{Y} = \{s_1^y, \ldots, s_{|\mathcal{X}|}^y\}$ is the set of response variables (system outputs) samples, be given. Each sample in the dataset corresponds to a measurement over time of system's variables. The regression trees algorithm creates a tree structure $\mathcal{T}$ by partitioning the set $\mathcal{X}$ into smaller regions, the *leaves* of the tree, following specific rules Breiman et al. (1984). Each leaf i contains a certain number of samples from $\mathcal{X}$. In particular, let $\ell_i \subset \mathcal{X}$, with $i = 1, \ldots, p$, be the set of predictor variable samples contained in the ith leaf of $\mathcal{T}$. The leaves of $\mathcal{T}$ form a partition of $\mathcal{X}$: $\mathcal{X} = \biguplus_{i=1}^p \ell_i$, $(\ell_\alpha \cap \ell_\beta = \varnothing, \forall \alpha \neq \beta)$. Then, the algorithm associates to each leaf ℓ_i a prediction

$\hat{y}_i$ as the average of the response values associated to each sample in ℓ_i. This algorithm is known as CART (see Breiman et al. (1984) for more details). However, since the prediction provided by the tree is an averaged value, the learning procedure described above needs to be modified to be applied in a RHC problem setup. To this aim, the following approach has been addressed in Jain et al. (2017).

Let us consider $\mathcal{X}$ as the set containing control inputs $u \in \mathbb{R}^m$, disturbances $d \in \mathbb{R}^r$ and state variables $x \in \mathbb{R}^n$, and $\mathcal{Y}$ as the set containing the output variable $y \in \mathbb{R}$. Without any loss of generality and for the sake of simplicity, we consider only a single output, but the discussion can be generalized considering different trees for different outputs, as shown in Smarra et al. (2018a), or multi-output trees, as shown in Jain et al. (2018). Each sample in the dataset is a vector containing the measured values of the variables at instant k. The set $\mathcal{X} = \{\mathcal{X}_c, \mathcal{X}_d\}$ is partitioned into set $\mathcal{X}_c = \{s_1^c, \ldots, s_{|\mathcal{X}|}^c\}$, of data associated to the m control variables, i.e. $s_k^c = [u_1(k), \ldots, u_m(k)]$, and set $\mathcal{X}_d = \{s_1^d, \ldots, s_{|\mathcal{X}|}^d\}$, of data associated to the $r + n$ disturbance and state variables, i.e. $s_k^d = [d_1(k), \ldots, d_r(k), x_1(k), \ldots, x_n(k)]$. For $\mathcal{Y} = \{s_1^y, \ldots, s_{|\mathcal{X}|}^y\}$, we have $s_k^y = y(k)$. The training process to grow a tree $\mathcal{T}$ is divided in 2 steps: 1) the tree is trained only using $\mathcal{X}_d$, instead of $\mathcal{X}$. It is important to note that besides external disturbances, $\mathcal{X}_d$ can also contain past terms of the output $\mathcal{Y}$; 2) affine function models are fit only as a function of variables in $\mathcal{X}_c$ for each leaf ℓ_i, i.e. only using samples $s_k^c, s_k^y \in \ell_i$, as we will show in Equation (1). This process is illustrated in the left side of Figure 1. The same methodology can be applied to Random Forests. The basic idea of the random forests algorithm (Breiman (2001)) is to grow multiple trees, indeed a forest, considering different random subsets of the dataset to train each tree. The prediction is given by averaging the response of all the trees in the forest, Breiman (2001). This reduces the overall variance in the prediction and mitigates the effect of the overfitting. The price to pay for the identification accuracy improvement is an increase in the computational complexity. In this paper both approaches are considered. To obtain a model that can be used in a RHC problem with predictive horizon of arbitrary length N, this procedure is used to grow multiple trees $\mathcal{T}_1, \ldots, \mathcal{T}_N$. Each tree $\mathcal{T}_j$ is used to predict system's response at the jth step of the horizon. We will discuss this more in detail in the following sections for both Regression Trees and Random Forests.

2.1 DPC-RT: DPC with Regression Trees

Without any loss of generality, and for the sake of simplicity, we consider $n = 1$, but the results can be extended to multiple responses considering multiple trees, i.e. one tree per state. The idea is to predict output y over an horizon of length N given the state measurement and the disturbance forecast at the current step. Applying the methodology illustrated above, it is possible to build N regression trees $\mathcal{T}_j, j = 1, \ldots, N$, and associate to each leaf i_j of each tree the following affine static model:

$$y(k + j) = \beta_{i_j}[1 \ u(k) \cdots u(k + j)]^\top, \forall i_j, \forall j, \qquad (1)$$

where $\beta_{i_j} \in \mathbb{R}^{jm+1}$ is obtained by fitting data in the leaf. Once these models are created, given the current state measurement and the prediction of the disturbance, i.e. the values of the variables in $\mathcal{X}_d$, in run-time, at each instant k, we can narrow down to a leaf of each tree $\mathcal{T}_j$ to
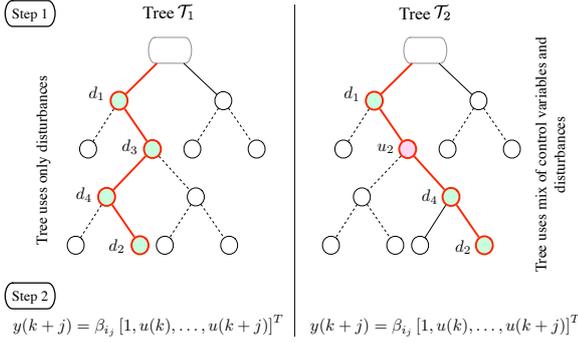
Fig. 1. Separation of variables. *Step 1:* Tree $\mathcal{T}_1$ is trained only on the disturbances $\mathcal{X}_d$ as the features. Tree $\mathcal{T}_2$ uses both the disturbances $\mathcal{X}_d$ and the control variables $\mathcal{X}_c$ for splitting and is thus not computationally suitable for control. *Step 2:* In the leaf ℓ_{i_j} of trees $\mathcal{T}_j$, a linear regression model parametrized by β_{i_j} is defined as a function only of the control variables.

find coefficients β_{i_j}, for $j = 1, \ldots, N$, to obtain the linear model (1) for each step of the horizon (see Algorithm 1 in Jain et al. (2017) for details). This is used to solve the following RHC problem

Problem 1.

$$\begin{aligned}
\underset{u_k}{\text{minimize}} \quad & \sum_{j=0}^{N} y_{k+j}^2 Q + u_{k+j}^\top R u_{k+j} + \lambda \varepsilon_j \\
\text{subject to} \quad & y_{k+j} = \beta_{i_j}[1 \ u_k \cdots u_{k+j}]^\top \\
& u_{k+j} \in \mathcal{U} \\
& |y_{k+j}| \leq \bar{y} + \varepsilon_j \\
& \varepsilon_j \geq 0, \ j = 0, \ldots, N,
\end{aligned}$$

where slack variables ε_j ensure recursive feasibility of the algorithm. This problem is solved as in classical MPC. At each time step the optimal $u_k^*, \ldots, u_{k+N}^*$ are computed and only the first one is applied as control input, i.e. $u(k) = u_k^*$.

2.2 DPC-RF: DPC with Random Forests

The goal in DPC-RF is to replace model (1) in Problem 1 with the following model obtained using Random Forests:

$$y(k+j) = \Theta_{i_j}[1 \ u(k) \cdots u(k+j)]^\top, \ \forall i_j, \ \forall j, \quad (2)$$

where $\Theta_{i_j} \in \mathbb{R}^{jm+1}$ is obtained by simply averaging out all the coefficients from all the trees in the forest j. Both DPC-RT and DPC-RF will be used in Section 4 to compare the methodology we propose in the next section.

3. DATA-DRIVEN SWITCHED AFFINE MODEL

The DPC modeling framework, although simple, is characterized by two main drawbacks: 1) the models in the leaves are affine functions that provide input-output *static* relations as described in (1); 2) the current state of the system is not considered by the model equation to predict the system's evolution. Basically, in the DPC, system modeling does not take into account an internal state evolution, which is a fundamental characteristic in control systems theory. The scope of this section is to address this issue providing a data-driven state-space modeling framework using regression trees and random forests. To this aim we leverage the concept introduced in Section 2, i.e. splitting

the training set, and replace the affine static models (1) in the leaves with LTI models, obtaining the following switched affine system

$$x(k+1) = A_{\sigma_k} x(k) + B_{\sigma_k} u(k) + f_{\sigma_k}, \quad (3)$$

where $\sigma_k : \mathbb{N} \to \{i_1, \ldots, i_N\}$ is an exogenous signal, that depends on measured values of variables in $\mathcal{X}_d$, and drives the switching rule among the leaves. We will see in Section 3.2 that this task is not trivial, since replacing the affine functions in (1) with LTI models is not enough to guarantee an internal state evolution, so we propose an extension of the state-space to overcome this problem. More precisely, the rest of this section is organized in 3 steps: 1) in Section 3.1 we derive, for each leaf of each tree, a model that provides the prediction $y(k+j)$ as an affine function of the internal state at time k and of the inputs at time $k, \ldots, k+j$, using regression trees; 2) in Section 3.2 we derive, from the model above, an equivalent switched affine system as in (3), and setup an MPC problem that will be used in Section 4 to show the performance of the proposed approach with respect to DPC and to the optimal MPC benchmark; 3) in Section 3.3 we extend the results obtained in Section 3.2 to the case where matrices of (3) are estimated using random forests. In this paper we consider the prediction of the state at next steps as the response variable with its regression terms, i.e. $x(k+j+1) = [y(k+j), y(k+j-1), \ldots, y(k+j-\nu)]^\top$.

3.1 Model generation

Let us consider a prediction horizon equal to N. We create N trees $\mathcal{T}_j$, $j = 1, \ldots, N$, each one built using disturbance $d(k+\psi-\delta_d), \ldots, d(k+\psi)$, $\psi \leq j$, and the current state with regression terms $y(k-1), \ldots, x(k-1-\delta_y)$ data from $\mathcal{X}_d$. Following the same idea of Section 2, we associate to each leaf ℓ_{i_j} of each tree $\mathcal{T}_j$, an LTI model of the form:

$$x(k+j) = A'_{i_j} x(k) + \sum_{\alpha=1}^{j} B'_{i_j, \alpha} u(k+\alpha-1) + f'_{i_j}. \quad (4)$$

The reason for including in the identification matrices $B'_{i_j, \alpha}$ that multiply past inputs, is to enforce the prediction accuracy that regression trees can provide. Matrices A'_{i_j}, $B'_{i_j, \alpha}$ and f'_{i_j} are identified using the least square method defined in Problem 2 below. To this aim we consider the experiments associated to the samples $s_{k_1}, \ldots, s_{k_\epsilon}$ in the leaf ℓ_{i_j} at time instants $k_1, \ldots, k_\epsilon$, and their past ν values. In particular, for each leaf ℓ_{i_j}, let us define

$$\Lambda_{i_j} = \begin{bmatrix} 1 & \cdots & 1 \\ x(k_1) & \cdots & x(k_\epsilon) \\ \vdots & & \vdots \\ x(k_1 - \nu) & \cdots & x(k_\epsilon - \nu) \\ u_1(k_1) & \cdots & u_1(k_\epsilon) \\ \vdots & & \vdots \\ u_m(k_1) & \cdots & u_m(k_\epsilon) \\ \vdots & & \vdots \\ u_1(k_1 + j) & \cdots & u_1(k_\epsilon + j) \\ \vdots & & \vdots \\ u_m(k_1 + j) & \cdots & u_m(k_\epsilon + j) \end{bmatrix}^\top \quad \xi_{i_j} = \begin{bmatrix} f \\ a_1 \\ \vdots \\ a_\nu \\ b_{1,1} \\ \vdots \\ b_{m,1} \\ \vdots \\ b_{1,j} \\ \vdots \\ b_{m,j} \end{bmatrix} \quad (5)$$

$$\lambda_{i_j} = \begin{bmatrix} x(k_1 + j + 1) & \cdots & x(k_\epsilon + j + 1) \end{bmatrix}^\top \quad (6)$$

We use this setup to formalize the following problem to estimate matrices in (4).

Algorithm 1 switched LTI model

1: DESIGN TIME (OFFLINE)
2: **procedure** TRAINING LTI MODELS IN LEAVES
3:     Set $\mathcal{X}_c \leftarrow$ manipulated features
4:     Set $\mathcal{X}_d \leftarrow$ non-manipulated features
5:     Build N predictive trees $\mathcal{T}_j$ using $(\mathcal{Y}, \mathcal{X}_d)$
6:     **for all** trees $\mathcal{T}_j$ **do**
7:         **for all** leaves ℓ_{i_j} of $\mathcal{T}_j$ **do**
8:             Solve Problem 2
9:             Create A'_{i_j}, $B'_{i_j,\iota}$ and f'_{i_j} as in (9)
10:         **end for**
11:     **end for**
12: **end procedure**

Problem 2.

$$\begin{aligned}
\underset{\xi_{i_j}}{\text{minimize}} \quad & \| \Lambda_{i_j} \xi_{i_j} - \lambda_{i_j} \|_2^2 \\
\text{subject to} \quad & \Gamma_{eq} \xi_{i_j} = \gamma_{eq} \qquad (7) \\
& \Gamma_{diseq} \xi_{i_j} = \gamma_{diseq} \qquad (8)
\end{aligned}$$

where (7) and (8) are used to constraint elements in ξ_{i_j} due to practical constraints of the plant. Elements of ξ_{i_j}, obtained solving Problem 2, are used to build matrices A'_{i_j}, $B'_{i_j,\iota}$ and f'_{i_j} in (4) as follows

$$A'_{i_j} = \begin{bmatrix} a_1 & a_2 & \cdots & a_\nu \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad B'_{i_j,\iota} = \begin{bmatrix} b_{1,\iota} & \cdots & b_{m,\iota} \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ & \vdots & \\ 0 & \cdots & 0 \end{bmatrix} \quad f'_{i_j} = \begin{bmatrix} f \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$
$$(9)$$

The procedure to compute matrices in (9) is summarized in Algorithm 1. In the following section we use such matrices to build our switched affine state-space model.

3.2 SAMPC-RT: data-driven Switched Affine MPC with Regression Trees

The scope of this section is to derive a state-space model formulation starting from the dynamical model (4), that can be used, other than to setup an MPC problem, to apply classical results in control theory, in particular for switched affine systems. In Smarra et al. (2018b), we show that this framework can not be obtained using model (3). As a consequence, to translate the model built as in Equation (4) into a switched affine dynamical system as in Equation (3), we define the following extended state

$$x_e = \begin{bmatrix} \bar{x} \\ u_- \end{bmatrix}, \qquad (10)$$

composed by the system state $\bar{x}$ and input dummy variables $u_- = [u_{-N+1} \cdots u_{-2} \ u_{-1}]^\top$, which represent the past values of the inputs applied to the system during the previous $N-1$ steps. Defining the following state-space representation

$$x_e(k+1) = A^e_{\sigma_k} x_e(k) + B^e_{\sigma_k} u(k) + f^e_{\sigma_k}, \qquad (11)$$

where

$$A^e_{\sigma_k} = \begin{bmatrix} A_{\sigma_k} & \bar{B}_{\sigma_k,-N+1} & \bar{B}_{\sigma_k,-N+2} & \cdots & \bar{B}_{\sigma_k,-1} \\ \mathbf{0} & \mathbf{0} & I & \cdots & \mathbf{0} \\ & & & \ddots & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & I \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \qquad (12)$$

$$B^e_{\sigma_k} = \begin{bmatrix} \bar{B}^\top_{\sigma_k,0} & \mathbf{0} & \cdots & \mathbf{0} & I \end{bmatrix}^\top \qquad (13)$$

$$f^e_{\sigma_k} = \begin{bmatrix} \bar{f}^\top_{\sigma_k} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}^\top \qquad (14)$$

$$\sigma_k \in \{i_1, \ldots, i_N\} \ \forall k, \qquad (15)$$

we can state the following proposition.

Proposition 1. Let A'_{i_j}, $B'_{i_j,\iota}$ and f'_{i_j}, $j = 1, \ldots, N$, $\iota = 1, \ldots, j$, be given as output of Algorithm 1. If A'_{i_j} is invertible for $j = 2, \ldots, N-1$, then there exist A_{σ_k}, $\bar{B}_{\sigma_k,\mu}$ and $\bar{f}_{\sigma_k}$, $\forall \mu = 1, \ldots, N-1$, such that, for any initial condition x_0, if $\bar{x}(k) = x(k) = x_0$, then $\bar{x}(k+j) = x(k+j)$, $\forall j = 1, \ldots, N$.

As derived in the proof of Proposition 1 in Smarra et al. (2018b), matrices A_{σ_k}, $\bar{B}_{\sigma_k,\mu}$ and $\bar{f}_{\sigma_k}$ are as follows:

$$\bar{f}_{i_j} = f'_{i_j} - A'_{i_j} A'^{-1}_{i_{j-1}} f'_{i_{j-1}} \qquad (16)$$

$$A_{i_j} = \begin{cases} A_{i_1} & \text{if } j = 1 \\ A'_{i_j} A'^{-1}_{i_{j-1}} & \text{if } j > 1 \end{cases} \qquad (17)$$

$$\bar{B}_{i_j,-\mu} = \begin{cases} 0 & \text{if } \mu \geq j \\ B'_{i_j,j-\mu} - A'_{i_j} A'^{-1}_{i_{j-1}} B'_{i_{j-1},j-\mu} & \text{if } 0 < \mu < j \\ B'_{i_j,j} & \text{if } \mu = 0 \end{cases} \qquad (18)$$

The switched affine model (11) can be now used to formalize the following MPC problem.

Problem 3.

$$\underset{u}{\text{minimize}} \quad x^\top_{e,k+N} P_N x_{e,k+N} + \sum_{j=0}^{N-1} x^\top_{e,k+j} Q x_{e,k+j} + u^\top_{k+j} R u_{k+j}$$

$$\begin{aligned}
\text{subject to} \quad & x_{k+j+1} = A^e_{\sigma_k} x_{k+j} + B^e_{\sigma_k} u_{k+j} + f^e_{\sigma_k} \\
& x_{k+j} \in \mathcal{X}, \ u_{k+j} \in \mathcal{U} \\
& x_{k+N} \in \mathcal{X}_f \\
& x_k = x(k), \ j = 1, \ldots, N-1.
\end{aligned}$$

The model obtained in (11) with our methodology is a switched affine dynamical system. Hence several results available in literature can be used to investigate systems' properties such as stability, controllability, stability and recursive feasibility for Problem 3 (see Smarra et al. (2018b) and references therein).

3.3 SAMPC-RF: data-driven Switched Affine MPC with Random Forests

As for Section 2.2, the idea here is to estimate matrices A'_{i_j}, B'_{i_j} and f'_{i_j} using Random Forests instead of Regression Trees. To this aim, let us consider N Random Forests $\mathcal{F}_\tau$, $\tau = 1, \ldots, N$. For each tree $\mathcal{T}_j$ of the forest $\mathcal{F}_\tau$, we can estimate a vector of parameters $\xi^\tau_{i_j}$ for each leaf ℓ_{i_j}, solving Problem 2 considering $\xi^\tau_{i_j}$ instead of ξ_{i_j}. With a small abuse of notation, let us indicate by $|\mathcal{F}_\tau|$ the number of trees in the forest τ. Then, parameters to build matrices in (9) can be obtained by averaging parameters in $\xi^\tau_{i_j}$, $\forall j = 1, \ldots, |\mathcal{F}_\tau|$. Thus we obtain the following switched affine system using Random Forests

$$x_e(k+1) = A^e_{f,\sigma_k} x_e(k) + B^e_{f,\sigma_k} u(k) + f^e_{f,\sigma_k}. \qquad (19)$$

SAMPC-RF is setup replacing model (11) in Problem 3 with model (19).

4. CASE STUDY

For the comparison of our approach with respect to DPC, we consider a bilinear building model developed at the Automatic Control Laboratory of 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 Gyalistras and Gwerder (2010). As we mentioned above our methodology fits well for large-scale systems, where building a mathematical model can be prohibitive. In order to provide a comparison with an optimal MPC benchmark, in this paper we validate our methodology considering a system for which a dynamical model is available. We build our switched affine dynamical model from the output data of Monte Carlo simulations of such model. For our future work we will consider complex building models, whose data can be obtained using EnergyPlus^{TM} (2000).

Model description. The bilinear model has 12 internal states including the inside zone temperature $\mathsf{T_{in}}$, the slab temperatures $\mathsf{T_{sb}}$, the inner wall $\mathsf{T_{iw}}$ and the outside wall temperature $\mathsf{T_{ow}}$. The state vector is defined as $x := [\mathsf{T_{in}}, \mathsf{T_{sb}^{(1:5)}}, \mathsf{T_{ef}^{(1:3)}}, \mathsf{T_{in}^{(1:3)}}]^T$. There are 4 control inputs including the blind position B, the gains due to electric lighting L, the evaporative cooling usage factor C, and the heat from the radiator H such that $u := [\mathsf{B}, \mathsf{L}, \mathsf{H}, \mathsf{C}]^T$. B and L affect both room illuminance and temperature due to heat transfer, whereas C and H affect only the temperature. The model is subject to 5 weather disturbances: solar gains with fully closed blinds $\mathsf{Q_{sc}}$ and with open blinds $\mathsf{Q_{so}}$, daylight illuminance with open blinds $\mathsf{I_o}$, external dry-bulb temperature $\mathsf{T_{db}}$ and external wet-bulb temperature $\mathsf{T_{wb}}$. The hourly weather forecast, provided by MeteoSwiss, was updated every 12 hrs. Therefore, to improve the forecast, an autoregressive model of the uncertainty was considered. Other disturbances come from the internal gains due to occupancy $\mathsf{Q_{io}}$ and due to equipments $\mathsf{Q_{ie}}$ which were assumed as per the Swiss standards (Merkblatt (2006)). We define $d := [\mathsf{Q_{sc}}, \mathsf{Q_{so}}, \mathsf{I_o}, \mathsf{Q_{io}}, \mathsf{Q_{ie}}, \mathsf{T_{db}}, \mathsf{T_{wb}}]^T$. For further details, we refer the reader to Oldewurtel (2011). The model dynamics are given below. The bilinearity is present in both input-state, and input-disturbance.

$$x(k+1) = Ax(k) + (B_u + B_{xu}[x_k] + B_{du}[d_k])u(k) + B_d d(k) \quad (20)$$
$$B_{xu}[x_k] = [B_{xu,1}x(k), B_{xu,2}x(k), \ldots, B_{xu,4}x(k)] \quad (21)$$
$$B_{du}[d_k] = [B_{du,1}d(k), B_{du,2}d(k), \ldots, B_{du,4}d(k)], \quad (22)$$

with $B_{xu,i} \in \mathbb{R}^{12 \times 12}$ and $B_{du,i} \in \mathbb{R}^{12 \times 8}$, $\forall i = 1, 2, 3, 4$. For this study, we assume that the disturbances are precisely known. We use MPC with this model to compare our approach to. The solution obtained from MPC sets the optimal benchmark, since it uses the exact knowledge of the plant dynamics and thus the associated control strategy is indeed the optimal strategy for the plant.

Goal. We want to minimize the energy usage, i.e. $c^\top u$, while maintaining a desired level of thermal comfort. At time step k, we solve the following continuously linearized MPC problem to determine the optimal sequence of inputs u^*.

Problem 4.

$$\underset{u}{\text{minimize}} \quad \sum_{j=1}^{N} \left(x_{k+j} - x_{\text{ref}}\right)^\top Q_j \left(x_{k+j} - x_{\text{ref}}\right) + c^\top u_{k+j-1}^2 + \lambda \varepsilon_j$$

$$\begin{aligned}
\text{subject to} \quad & x_{k+j} = Ax_{k+j-1} + Bu_{k+j-1} + B_d d_{k+j-1} \\
& B = B_u + B_{xu}[x_k] + B_{du}[d_{k+j-1}] \\
& x_{k+j} \in [x_{\min} - \varepsilon_j, x_{\max} + \varepsilon_j] \\
& u_{k+j-1} \in [u_{\min}, u_{\max}] \\
& x_k = x(k) \\
& \varepsilon_j \geq 0, j = 1, \ldots, N-1,
\end{aligned}$$

$$(23)$$

where $Q \in \mathbb{R}^{12 \times 12}$ has all zero entries except for the element $(1, 1)$ associated to the zone temperature, $c^\top \in \mathbb{R}^{1 \times 4}$ is proportional to the cost of using each actuator and λ penalizes state bound violations. At each time step only the first optimal input of the sequence is applied to the system.

Training. As the output variable of the system for the training, we consider the first component of x, i.e. $y(k+j) := x^1(k+j+1)$. To train the trees $\mathcal{T}_j$ and the forests $\mathcal{F}_j$, we consider weather disturbances, external disturbances due to occupancy and equipments, and autoregressive terms of the room temperature, i.e.

$$s_k^d = [d(k+j-N), \ldots, d(k+j-1), x^1(k), \ldots, x^1(k-\delta)] \in \mathcal{X}_d, \quad (24)$$

where δ is the order of auto-regression. Finally, the inputs are used to train both, the affine static models for DPC and the LTI models for the switched MPC. The training data was generated by simulating the bilinear model with rule-based strategies for 10 months in 2007. May was deliberately excluded for testing the DPC implementation. For the simulation we chose $\delta = 6$ and $\nu = 1$.

Validation. We compare the prediction for the first time step $x^1(k+1)$ and the 6-hour ahead prediction $x^1(k+6)$ for a week in the month of May. The quantitative summary of model accuracies is given in Table 1. We can see that at step $k+1$ the quality of the identification of the different approaches is quite similar, except for the affine model obtained with regression trees. Instead at step $k+6$ the switched affine model shows better approximation with respect to the other approaches, although it is still close to the others. As mention in Section 2, the models based on regression trees show worse performance than the ones obtained with random forests due to high variance. The quality of the proposed approach will be more evident in the closed-loop simulations, since, due to the presence of the internal state, the switched affine model has memory of the past inputs applied.

Closed-loop simulations. We simulated the data-driven Switched Affine MPC Problem 3, with both model (11) (SAMPC-RT) and model (19) (SAMPC-RF), and DPC Problem 1, with both model (1) (DPC-RT) and model (2) (DPC-RF) using the same cost function and constraints of Problem 4. We compare the results against the MPC benchmark setup in Problem 4. The results are shown in Fig. 2. The performance is compared for 3 days in winter, i.e. January 28-31 and 3 days in summer, i.e. May 1-3. The sampling time in the simulations is 1 hour. The control horizon N is 6 hours. 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 winter and

Model	Prediction step	NRMSE	RMSE
Tree	$k+1$	0.027	0.79
Tree	$k+6$	0.034	0.99
SA-tree	$k+1$	0.013	0.39
SA-tree	$k+6$	0.035	1.02
Forest	$k+1$	0.012	0.36
Forest	$k+6$	0.033	0.97
SA-forest	$k+1$	0.013	0.40
SA-forest	$k+6$	0.030	0.88

Table 1. Quantitative comparison of NRMSE and RMSE for the different models: affine static models and switched affine LTI models identified with trees and forests for different prediction steps.
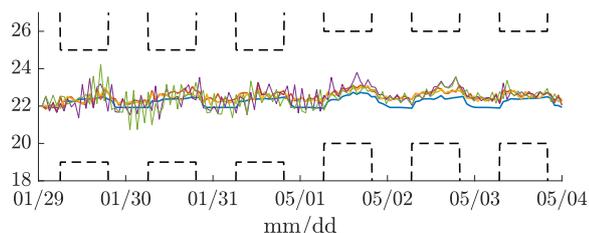
(a) Room temperature. Random forests-based approaches track the reference temperature better than regression trees-based ones and provide smoother behaviour.
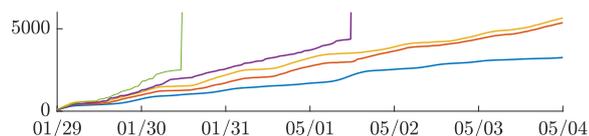
(b) Cumulative optimal cost after solving optimization. SAMPC-RF shows best performance after the MPC.

Fig. 2. Comparison of optimal performance for 3 days in January and 3 days in May.

$[20, 26]$ °C during the summer. The optimization is solved in MATLAB using CPLEX. The reference temperature is chosen to be 22 °C. The optimized cost function is shown in Figure 2(b). Results show that the SAMPC-RF is the closest to the optimum, outperforming all the other methods. SAMPC-RT also behave better than its counterpart in DPC, but provides worse performace than DPC-RF due to the high variance problem regression trees are subject to. For the same reason, that brings to model inaccuracy, the cost for regression trees model based controllers blows up as one of the slack variables is non-zero. The cumulative cost goes extremely high since the parameter λ that weights bounds violation is setup to 10^3. The room temperature profile is shown in Figure 2(a). We can see also here that random forests based controllers provide less spiked trajectories that are closer to the optimum with respect to regression trees based controllers. More details about disturbances and optimal inputs can be found in Smarra et al. (2018b).

5. CONCLUSIONS AND FUTURE WORK

In this paper, we provide a methodology to construct a data-driven state-space switched affine model of a system using regression trees and random forests using only historical data. We setup an MPC problem and compare the performance of our approach against another MPC controller, designed using the actual physics-based model, and against DPC controller, that uses static data-driven

models. We compare the results of the approaches on a building energy management problem. We show that our strategy, without using any physical model of the system, is comparable in terms of optimal cost minimization to MPC which requires a physics-based model, thus bypassing the need for expensive physical modeling in the complex systems like buildings. This new approach for data-driven state-space switched affine model is also better than our previous work on control with regression trees and random forests that uses static models. This work is a staring point for the following future directions. Future work is focused on applying the proposed methodology on a more complex and realistic EnergyPlus model, for which building a model predictive controller is time and cost prohibitive. We want also to consider problems where stability is important, i.e. frequency control in a microgrid, and address the problem of providing guarantees on the system and on system properties.

REFERENCES

Afram, A., Janabi-Sharifi, F., Fung, A., and Raahemifar, K. (2017). 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 and Buildings*, 141, 96–113.

Behl, M., Smarra, F., and Mangharam, R. (2016). Dr-advisor: A data-driven demand response recommender system. *Applied Energy*, 170, 30 – 46.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5–32.

Breiman, L., Friedman, J., Stone, C.J., and Olshen, R.A. (1984). *Classification and regression trees*. CRC press.

EnergyPlus^{TM} (2000). Energyplus. https://energyplus.net/.

Ferreira, P., Ruano, A., Silva, S., and Conceiçao, E. (2012). Neural networks based predictive control for thermal comfort and energy savings in public buildings. *Energy and Buildings*, 55, 238–251.

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

Iovine, A., Damm, G., Santis, E.D., and Benedetto, M.D. (2017). Management controller for a DC microgrid integrating renewables and storages. In *2017 IFAC World Congress*.

Jain, A., Smarra, F., Behl, M., and Mangharam, R. (2018). Data-driven model predictive control with regression trees-an application to building energy management. *ACM Transactions on Cyber-Physical Systems*, 2(1), 1:21.

Jain, A., Smarra, F., and Mangharam, R. (2017). Data predictive control using regression trees and ensemble learning. In *Proceedings of the 56th IEEE Conference on Decision and Control*.

Kennel, F., Görges, D., and Liu, S. (2013). Energy management for smart grids with electric vehicles based on hierarchical mpc. *IEEE Transactions on Industrial Informatics*, 9(3), 1528 – 1537.

Ma, Y., Matuško, J., and Borrelli, F. (2015). Stochastic model predictive control for building hvac systems: Complexity and conservatism. *IEEE Transactions on Control Systems Technology*, 23(1), 101 – 116.

Maasoumy, M., Razmara, M., Shahbakhti, M., and Vincentelli, A.S. (2014). Handling model uncertainty in model predictive control for energy efficient buildings. *Energy and Buildings*, 77, 377 – 392.

Macarulla, M., Casals, M., Forcada, N., and Gangolells, M. (2017). Implementation of predictive control in a commercial building energy management system using neural networks. *Energy and Buildings*, 151, 511–519.

Merkblatt, S. (2006). 2024: Standard-nutzungsbedingungen für die energie-und gebäudetechnik. *Zürich: Swiss Society of Engineers and Architects*.

Oldewurtel, F., Parisio, A., Jones, C.N., Gyalistras, D., Gwerder, M., Stauch, V., Lehmann, B., and Morari, M. (2012). Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45, 15 – 27.

Oldewurtel, F. (2011). *Stochastic Model Predictive Control for energy efficient building climate control*. Ph.D. thesis, ETH Zurich.

Smarra, F., Jain, A., de Rubeis, T., Ambrosini, D., D'Innocenzo, A., and Mangharam, R. (2018a). Data-driven model predictive control using random forests for building energy optimization and climate control. *Applied Energy*, To Appear.

Smarra, F., Jain, A., Mangharam, R., and D'Innocenzo, A. (2018b). Data-driven switched affine modeling for model predictive control. In *IFAC Conference on Analysis and Design of Hybrid Systems*. Technical report available on arXiv.

Sturzenegger, D., Gyalistras, D., Morari, M., and Smith, R.S. (2015). Model predictive climate control of a swiss office building: Implementation, results, and costbenefit analysis. *IEEE Transactions on Control Systems Technology*, 24(1), 1 – 12.

Venkat, A., Hiskens, I., Rawlings, J., and Wright, S. (2008). Distributed mpc strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6), 1192 – 1206.

Žáčeková, E., Váňa, Z., and Cigler, J. (2014). Towards the real-life implementation of mpc for an office building: Identification issues. *Applied Energy*, 135, 53 – 62.