


Comparing Machine Learning based Methods to standard Regression Methods for MPC on a virtual Testbed

Conference Paper**Author(s):**

Bünning, Felix; Pfister, Corentin; Aboudonia, Ahmed; Heer, Philipp; [Lygeros, John](#) 

Publication date:

2021

Permanent link:

<https://doi.org/10.3929/ethz-b-000524933>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Comparing Machine Learning based Methods to standard Regression Methods for MPC on a virtual Testbed

Felix Bünning^{1,2}, Corentin Pfister^{1,2}, Ahmed Aboudonia², Philipp Heer¹, John Lygeros²

¹Empa, Urban Energy Systems Laboratory, Switzerland

²Automatic Control Laboratory, ETH Zürich, Switzerland

Abstract

Data Predictive Control has emerged as a promising way to control buildings optimally with the help of data-driven models. Besides conventional system identification methods, also Machine Learning based methods can be the basis of such models. While these methods have been validated in individual simulations or experiments, there is a lack of comparability due to changing experimental conditions or mismatch between simulation cases. Here, we present a comparison of three different data enabled building control methods one of the test cases of the virtual building controller testbed BOPTEST: a conventional ARX model with a one-hot encoded solar model, a Random Forest model with linear control inputs, and an Input Convex Neural Network model. Our results suggest that the ARX model outperforms the other models in most of the relevant criteria in the one-zone hydronic test case.

Key Innovations

A conventional system identification method is compared to machine learning methods for building MPC on a standardized test bed.

Practical Implications

Conventional ARX models with a one-hot encoded solar model can constitute a competitive alternative to novel machine learning methods for applications of Data Predictive Control in buildings.

Introduction

Model predictive control (MPC) can significantly reduce the amount of energy consumed for heating and cooling of buildings (Sturzenegger et al., 2016). However, generating and maintaining physics-based models for the thermal behaviour of buildings can be difficult, time consuming and thus expensive. This may hinder the wide-spread application of MPC in buildings.

To address this issue, the use of data-driven models, which are generated purely from historical measurement data has been proposed (Maddalena et al.,

2020). The use of such models in a receding horizon optimal control similar to MPC is often referred to as Data Predictive Control (DPC). In (Bünning et al., 2020, 2021; Schalbetter, 2020) we have applied different modelling techniques, based on Random Forests (RF), Input Convex Neural Networks (ICNN), and ARX models in experimental case studies on a real apartment unit. All model types showed satisfactory control performance in terms of comfort constraint violations and reduction of the heating/cooling input energy compared to standard thermostat-based control. However, the exact comparison of the performance of different controllers in experiment is difficult due to changing ambient conditions or occupant behaviour. Other authors (Smarra et al., 2018; Wang et al., 2019) use individual simulation cases to investigate DPC behaviour. However, as the simulation environments differ between studies, direct comparison is also difficult.

The Building Optimization Testing (BOPTEST) framework (Blum et al., 2019) is a benchmarking framework to address this problem. It intends to feature a variety of simulation cases from the building domain with varying complexity and heating, cooling and air conditioning technologies. Similar to the OpenAI Gym (Brockman et al., 2016) in the Machine Learning community, it encourages researchers to benchmark their controllers on the same test cases. Besides MPC-based controllers, also imitation or reinforcement learning based controllers could potentially be tested.

In this work, we apply the three DPC strategies based on RF, ICNN and ARX to a test case of the BOPTEST framework featuring a hydronic heating system coupled to a single-zone building model. We compare the controller performance with respect to energy consumption and comfort constraint violation and investigate the training sample efficiency of the different models on training data sets of varying length. Here, we find that the ARX model outperforms the other methods in most relevant criteria.

In the remainder of the article we explain the concept of Model Predictive control and introduce the

different modelling strategies. We then present the BOPTTEST test case and discuss the generation of training data as well as the set up of the individual controllers. Afterwards, we discuss the results of the comparison study and conclude the article.

Methodology

In the following section, we define the concept of Model Predictive Control and introduce the different data-driven modelling techniques used in this study.

Model Predictive Control

MPC is a scheme for optimal control, where an optimization scheme is solved over a receding horizon. At discrete time instants, the current state of the system x_0 is measured and the optimisation problem

$$\min_{u,x,\epsilon} \sum_{k=0}^{N-1} J_k(x_{k+1}, \epsilon_{k+1}, u_k) \quad (1a)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k, d_k) \quad (1b)$$

$$(x_{k+1}, u_k) \in (\mathcal{X}_{k+1} \oplus \epsilon_{k+1}, \mathcal{U}_k) \quad (1c)$$

$$\forall k \in [0, \dots, N-1],$$

is solved. Here, the variables x , u and d denote the states, inputs and disturbances respectively, J is the cost function, k denotes the time step in the horizon N and f is the function describing the system dynamics, which depends on the state, input and disturbance in the last time step. The sets $\mathcal{X}_k \oplus \epsilon_{k+1}$ and \mathcal{U}_k describe constraints for states and inputs. The variable ϵ describes a slack variable for the state constraints. After solving the problem, the controller applies the first element of the optimal input sequence, u_0^* , to the plant (here the building) and the process is repeated. To solve problem (1) to optimum efficiently and reliably, it is beneficial for the optimisation problem to be convex in the decision variables u , x and ϵ . This is ensured for all the models presented in this study.

Random forests and linear regression

The first model to describe the dynamics f is based on Random Forests and linear regressions (Smarra et al., 2018; Bünning et al., 2020). We briefly discuss the model generation for completeness.

We assume given a data set of historical measurements. This set consists of tuples of measured states (which are room temperatures in this application), control inputs (such as valve positions for example), and disturbances which could have an influence on the state (ambient temperature, solar irradiation, etc.). The tuples also include autoregressive terms of these variables. The goal is to build N single state predictors, one for each x_{k+j} in problem (1), where x_{k+j} depends on these tuples, which are sequences of previous measured states, disturbances and control inputs up to the time of optimization (i.e. up to time $k=0$),

and future (i.e. between time $k=1$ and $k=N-1$) disturbances and control inputs.

First, we build Random Forests on the basis of all variables that are non-controllable: previous states, previous and future disturbances and previous control inputs. Random Forests are ensembles of regression trees, which approximate a function with constant values for certain partitions of the function input. These constant values lie at the end of the tree and are called leaves. After training a tree (or a forests) a given input tuple (previous states, previous and future disturbances and previous control inputs) will lead to a leaf (or a set of leaves). We note that the resulting optimization will remain convex, although the model has switching dynamics, because the trees are not built on the basis of states between time $k=0$ and $k=N-1$.

For each of these leaves, there exists a corresponding set of tuples of future control inputs that is not part of the model yet. We therefore fit a linear model in each leaf i to approximate x_{k+j} with these inputs. For the prediction of the future state x_{k+j} , this gives rise to

$$x_{k+j} = \beta_{ji,0} + \beta_{ji,1} \sum_{n=1}^j u_{k+n-1} + e, \quad (2)$$

in which $\beta_{ji,0}$ and $\beta_{ji,1}$ denote the fitted coefficients and e the model error. The state (i.e. room temperature) is therefore an affine function of the sum of all control inputs from the time instant k , when the prediction is made, to the predicted time instant $k+j$. Note, that the β coefficients are causal and implicitly capture the influence of all non-controllable model inputs and the prediction step j , because they are different for each leaf and each tree. We use the sum in (2) instead of individual coefficients, because we found in previous studies (Bünning et al., 2020) that it benefits sample-efficiency. This usually outweighed the disadvantage of a less physical model in previous studies. As we use random forests and not single regression trees, the average of all relevant β_{ji} from the individual trees is taken, giving rise to β_j . For a more detailed and mathematical description of the model, please refer to the original sources (Smarra et al., 2018; Bünning et al., 2020).

Input Convex Neural Networks

The second model is based on ANN, which are generally non-convex. However, Amos et al. (2017) presented a set of constraints on the network weights and a specific structure for feed-forward Artificial Neural Networks. These enforce convexity of the network output with respect to a subset of the inputs for single step predictions. In (Bünning et al., 2021), we have extended this approach to multi-step predictions, which makes the resulting model suitable for approximating f in problem (1).

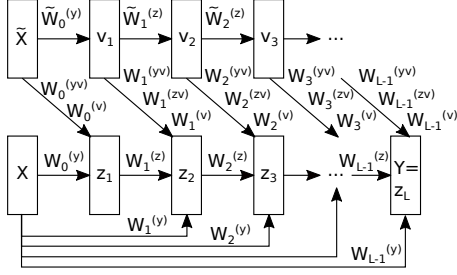


Figure 1: Structure of a Partially Input Convex Neural Network.

Figure 1 shows the structure of a Partially Input Convex Neural Network (PICNN), where the model output Y is convex in the model input X , but not necessarily in the input \tilde{X} . Here, the output of the layers v and z follow

$$\begin{aligned}
 v_{i+1} &= \tilde{g}_i(\tilde{W}_i v_i + \tilde{b}_i) \\
 z_{i+1} &= g_i \left[W_i^{(zv)} \left(z_i \circ g_i^{(zv)} [W_i^{(zv)} v_i + b_i^{(z)}] \right) \right. \\
 &\quad + W_i^{(yv)} \left(y \circ g_i^{(yv)} [W_i^{(yv)} v_i + b_i^{(y)}] \right) \\
 &\quad \left. + W_i^{(v)} v_i + b_i \right], \quad (3)
 \end{aligned}$$

with $z_0, W_0^{(z)} = 0$ and $v_0 = \tilde{X}$, where $\tilde{W}_i, W_i^{(z)}, W_i^{(zv)}, W_i^{(y)}, W_i^{(yv)}, W_i^{(v)}$ are input weights, $\tilde{b}_i, b_i, b_i^{(z)}, b_i^{(y)}$ denote constant biases, \tilde{g}_i and g_i are activation functions, and \circ denotes the Hadamard product. The composition of two PICNN $f_2(\tilde{x}_2, x_2, f_1(\tilde{x}_1, x_1))$ is convex with respect to x_1 , if all weights $W_i^{(z)}$ and $W_i^{(y)}$ are non-negative, all functions $g_i^{(zv)}$ and $g_i^{(yv)}$ map to a non-negative value and the function g_i is convex and non-decreasing (Bünning et al., 2021).

If the network inputs are chosen as $\tilde{X} = d_k$ and $X = (x_k, u_k)$, PICNN can be used for approximating f in problem (1), while maintaining a quasi-convex problem structure (Bünning et al., 2021). Accordingly to standard feed-forward ANN, PICNN can be trained with stochastic gradient decent methods.

ARX with solar model

In an Autoregressive model with Exogenous Variables (ARX), the output is a linear function of autoregressive terms of the output and of inputs and disturbances (Ljung, 1998). To act as a model for the dynamics f in problem (1), the ARX model can be written as a one step predictor

$$x_{k+1} = \Theta [x_k \dots x_{k-\delta} \ u_k \dots u_{k-\delta} \ d_k \dots d_{k-\delta}]^T, \quad (4)$$

where x, u, d are again states, control inputs and disturbances respectively, and δ determines the number of autoregressive terms. The regression coefficients Θ can be found with least squares linear regression from measurement data.

A building-modelling specific problem is to model the disturbance through solar irradiation gains through windows I_{win} from data, especially with a linear model. These gains follow

$$I_{win} = A_{win} \sin(\alpha - \alpha_0) \frac{\cos(\beta)}{\sin(\beta)} I_{hor}, \quad (5)$$

where A_{win} denotes the surface area of the window, α and β denote the azimuth and elevation of the sun respectively, α_0 denotes the orientation of the window surface and I_{hor} denotes the horizontal global irradiation. The gains are a highly non-linear function of the time t (which influences α and β) and more importantly of the window orientation. To address this problem, we assume that $I_{vert} = \frac{\cos(\beta)}{\sin(\beta)} I_{hor}$, which describes the irradiation through a vertical surface following the sun, is given as a forecast, as it is only time and location dependent and we learn the coefficient $A_{win} \sin(\alpha - \alpha_0)$ through regression, by adding corresponding parameters to Θ in (4). To account for the time-dependence of α , we perform a one-hot encoding (Brownlee, 2018) of I_{vert} with respect to discrete time-periods t_1, \dots, t_T :

$$I_{vert, t_i} = \begin{cases} I_{vert(t)}, & \text{if } t_i \geq t \geq t_{i+1} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

This creates T input variables per autoregressive step for (4), which are zero most of the time, but equal to I_{vert} for fixed periods of the day. For example, if T is chosen to be 6, $I_{vert,1}$ will attain I_{vert} for the first four hours of the day, and zero otherwise, $I_{vert,2}$ will attain I_{vert} for the from 4 am to 8 am, and zero otherwise, etc. The variables $I_{vert,1}, \dots, I_{vert,T}$ are added to the set d in equation (4). Although one-hot encoding introduces a piecewise function, the problem remains convex, because the choice of the piece of the function is only dependent on disturbances, but not on decision variables.

Case study

The three models presented in the previous section are tested and compared to each other in the setting of predictive building control within the BOPTTEST¹ framework (Blum et al., 2019). It provides standardized simulation test cases for advanced building controllers and thus encourages the benchmarking of different control methods.

Test case

In this study, we use the test case *bestest-hydronic*, which is an implementation of the BESTEST 900, a validated high-fidelity model of a single zone building according to the ASHRAE 140 standard. As depicted in Fig. 2, the building has a rectangular floor plan of 6 by 8 m, a ceiling height of 2.7 m, and a south-facing

¹commit [bd22bf2edca2b31a51e64f67038adf236c1dafa9](https://github.com/BOPTTEST/bd22bf2edca2b31a51e64f67038adf236c1dafa9)

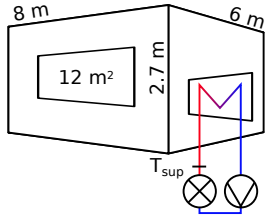


Figure 2: Schematic of the used BOPTEST test case.

window of 12 m^2 . The exact specification of the wall materials can be found in the model documentation (Blum et al., 2019). The zone is occupied by a single occupant before 7 am and after 8 pm during weekdays and all day during weekends. For the heating system, the model comprises a boiler with a maximum power of 5 kW and an efficiency calculated on the basis of a polynomial curve, a circulation pump with a fixed efficiency of 49 % and a radiator that transfers the heat to the thermal zone (Blum et al., 2019).

We control the system by modulating the heating system supply temperature T_{sup} to either $60 \text{ }^\circ\text{C}$ or $20 \text{ }^\circ\text{C}$ (which effectively sets it equal to the return temperature as the boiler can only heat but not cool), with pulse-width-modulation (PWM). The controlled variable is the zone temperature x . For the disturbances d , we assume to have access to the ambient temperature and the global horizontal irradiation forecasts, which are exact in the current version of BOPTEST, and to the current time. There is no measurement or forecast of occupancy assumed², although the models could implicitly learn occupancy patterns through time-related inputs.

Training data generation

As all of the used models are data-driven, training data for outputs, control inputs and disturbances need to be generated first. For this, we use a discrete P-controller

$$u_{k+1} = k_p(x_{set} - x_k), \quad (7)$$

where the set point x_{set} is varied between $22.5 \text{ }^\circ\text{C}$ and $23.5 \text{ }^\circ\text{C}$ every hour. The sampling time is 15 minutes and the control input is applied through PWM of the heating supply temperature with a single switch per sampling step, as described before. The proportional controller gain k_p is chosen such that the resulting trajectory resembles one created with a hysteresis controller, which is commonly used in many buildings and also resembles the baseline controller we work with in experiments on real buildings. For training the model, we use three different time periods, as described in Table 1.

We note that closed-loop identification (i.e. identification with training data generated by a feedback

²Although an exact occupancy forecast is available from BOPTEST, we chose to not use it in this study, as such forecasts are often difficult to obtain in real life.

controller) is generally difficult, because closed-loop trajectories are often not informative, i.e. persistently exciting (Ljung, 1998). However, as open-loop experiments are expensive and generally not desirable in the context of building control, it is interesting to investigate how the different models cope with the closed-loop generated training data.

Model set up

The models in the Methodology Section perform differently with different types of model inputs and outputs. In earlier studies involving experiments on a real building, we have optimized these (Bünning et al., 2020, 2021; Schalbetter, 2020). We inherit these settings here.

In the case of the RF model, we use as the model output the room temperature difference between two time steps, thus the model predicts Δx instead of x . As model inputs, autoregressive terms of Δx , of the ambient temperature, of the horizontal solar irradiation, of the time of the day encoded as a sine function, and the control input are used. The choice of the inputs is a result of feature engineering for a specific experimental test case from a previous study (Bünning et al., 2020). For the ICNN model, also Δx is used as the model output. As convex inputs, autoregressive terms of Δx , of the temperature difference between room and ambient, and the control input are used. Autoregressive terms of the horizontal solar irradiation and the sine encoded time of the day are used as non-convex inputs because they are not decision variables in the optimization. Also in this case, the input choice is a result of a previous study (Bünning et al., 2021). In the case of the ARX model, we directly predict x as an output. The model inputs are autoregressive terms of x , of the ambient temperature, of the control input, and of the one-hot encoded solar irradiation through a vertical surface as described in the Methodology section. We do not use the sine encoded time feature in the ARX model because we assume that the main time-dependent disturbance, solar irradiation, is already modelled by (5) and (6), which is not the case for RF and ICNN. It can also be noted that we use absolute temperatures for the ARX models, while temperature differences are used for RF and ICNN. While temperature differences have shown to produce more accurate predictions in the cases of RF and ICNN in previous studies, for linear models, the models are equivalent because the only difference between both approaches is an offset in Φ , which can be exactly obtained from the data. However, the implementation is more efficient if x is used directly.

The number of autoregressive input terms in the three models will be set using cross validation in the next section.

Controller set up

The ARX, RF and ICNN model are embedded as $f(x_k, u_k, d_k)$ into an MPC control scheme that solves

the optimization problem

$$\min_{u,x,\epsilon} \sum_{k=0}^{N-1} (u_k R u_k + \lambda \epsilon_{k+1}) \quad (8a)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k, d_k) \quad (8b)$$

$$\mathcal{X}_{min} - \epsilon_{k+1} \leq x_{k+1} \leq \mathcal{X}_{max} + \epsilon_{k+1} \quad (8c)$$

$$\epsilon_{k+1} \geq 0 \quad (8d)$$

$$\mathcal{U}_{min} \leq u_k \leq \mathcal{U}_{max} \quad (8e)$$

$$\forall k \in [0, \dots, N-1],$$

in a receding horizon, where a quadratic cost for the control input $R = 1$, and a linear cost $\lambda = 100$ for the comfort slack variable is used. The sampling time is 15 minutes and the control horizon $N = 12$, which amounts to three hours. All of these values were chosen after preliminary simulations. The comfort constraints \mathcal{X}_{min} and \mathcal{X}_{max} are time varying and will be obvious in the result section. The limits for the control input u_k are $\mathcal{U}_{min} = 0$ and $\mathcal{U}_{max} = 1$, which follows from the applied PWM of the heating supply temperature: u_k denotes the fraction of time in which the supply temperature is set to 60 °C. In the case of the ARX and RF models, the resulting problem is a Quadratic Program, which we solve with the QP solver of CVXOPT (Andersen et al., 2004) in Python 3. In the case of the ICNN, the resulting problem is a quasi-convex problem, which we solve it with the COBYLA (Powell, 1994) solver of SciPy (Virtanen et al., 2020) in Python 3.

Results and Discussion

Sample efficiency and open-loop accuracy

Table 1: Training data periods.

Training set	Simulation period
Full training	11 January - 11 March 2009
Medium training	11 January - 01 February 2009
Short training	11 January - 21 January 2009

In a first study, we compare the open-loop accuracy of the models trained with three training data sets of varying length, as defined in Table 1, on a testing set from 4 January - 11 January 2009. The testing set is obtained with the same P-controller as used for the training data generation and the P-controller outputs are used as model inputs for the predictions by the three models. We compare three different prediction horizons of $N=2$ (30 minutes), $N=6$ (1.5 hours) and $N=12$ (3 hours).

However, for this, the number of autoregressive terms needs to be chosen for each model first. We choose this parameter by training the models on the Full training set and comparing the coefficient of determination³ R^2 for the $N=12$ open-loop predictions in the testing set.

³The coefficient of determination ranges from -inf to 1, with

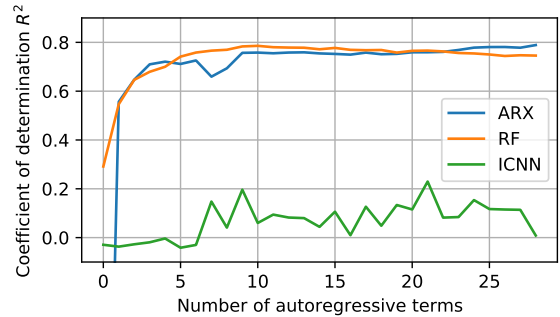


Figure 3: Analysis of model history for training.

The results of the analysis are shown in Fig. 3. For the ARX and RF models, we chose the number of auto-regressive terms based on where R^2 first crosses 0.75, which is at $\delta = 9$ for the ARX model, and at 6 for the RF model. It is already apparent that the ICNN is less accurate compared to the other two models. For the ICNN, we chose $\delta = 7$, which is the point where R^2 first crosses 0.1.

We note that, while the training of the ARX model is a convex problem leading to the same model every time the training is performed on the same training data, this is not the case for the RF and ICNN. Here, every time a model is trained, a different model may be obtained. This effect is less prominent for the RF, as it is built by averaging many regression trees trained with boot-strapped data, but very prominent for the ICNN, where every model is trained with different initial values of the networks' weights and biases. This can be noticed by the non-smooth curve for ICNN in Fig. 3. For both RF and ICNN, we took into account the first trained model with $R^2 > -0.1$ on the testing set for each analyzed number of autoregressive terms

The results for the open-loop prediction with varying training sets and prediction horizons are shown in Table 2. It can be seen that the ARX model clearly outperforms the other models in terms of sample efficiency. It shows high accuracy for all prediction horizons, even when trained with the smallest training data set of ten days. For the full training data set, the RF model shows comparable accuracy and for the medium training data set it still performs well. When the smallest training data set is used with the RF model, the resulting prediction accuracy is bad for the large horizon of $N=12$. The ICNN is clearly outperformed by the other two models. While it is accurate for $N=2$ in all cases, the prediction accuracy for higher horizons is not satisfactory in any case.

Control performance

The models were then deployed within an MPC framework to the BOPTTEST test case in the test-

1 being a perfect forecast, 0 denoting a forecast that is as good as taking the average of the predicted value, and lower values denoting worse prediction.

Table 2: Results of open-loop prediction with varying training size in R^2 .

Experiment	ARX full	ARX med	ARX short	RF full	RF med	RF short	ICNN full	ICNN med	ICNN short
N=2	0.99	0.99	0.99	0.98	0.96	0.86	0.94	0.92	0.77
N=6	0.93	0.94	0.92	0.91	0.87	0.66	0.29	0.05	-0.27
N=12	0.79	0.79	0.71	0.75	0.67	0.20	0.13	0.12	0.03

Table 3: Energy consumption and comfort constraint violations in closed-loop experiments.

Experiment	ARX full	ARX med	ARX short	RF full	RF med	RF short	ICNN full	ICNN med	ICNN short
Constr. viol. in $^{\circ}\text{C min}$	193.12	201.59	241.52	1363.38	859.74	398.48	9366.86	6341.79	538.50
Energy in kWh	363.84	363.83	363.82	360.87	363.11	365.05	338.80	348.10	366.38

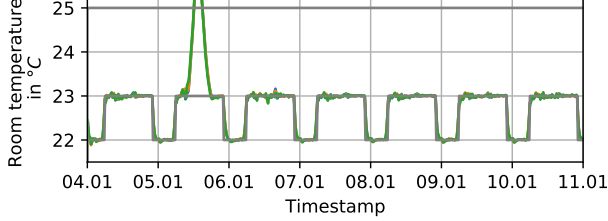


Figure 4: Room temperature trajectories in the MPC experiment with the ARX model with Full training set (blue), Medium training set (orange), and Short training set (green).

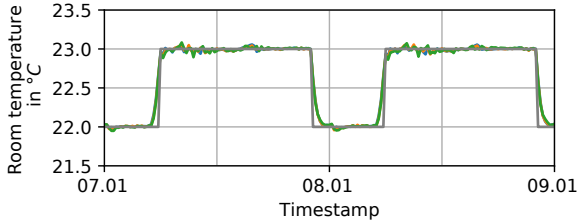


Figure 5: Details of room temperature trajectories in the MPC experiment with the ARX models.

ing period of 4 January - 11 January 2009. During this period, the ambient temperature varies between -0.5°C and 7°C and the maximum daily solar irradiation varies between 70 W/m^2 and 220 W/m^2 , which constitutes interesting varying ambient conditions for a heating period.

Figure 4 shows the room temperature trajectory of MPC with ARX for the full experiment. The time-varying comfort constraints are depicted in grey. The controllers behave very similar to each other (which is why only the green trajectory is really visible) and lead to smooth temperature trajectories. Note, that during the day of 5 January, the temperature is driven up due to strong solar irradiation. The controller has no authority here, as it can only heat, but not cool. Figure 5 shows a detailed view of the trajectories for the days 07 to 09 January. Here, it can be seen that the controller starts to heat up the room early enough on both days to meet the higher comfort constraint at 06:00 in the morning. It can also be noticed that a horizon of $N=6$ would be sufficient to anticipate the 1°C step of the comfort constraint in time.

Figure 6 shows the trajectories of the RF model for the whole experiment. The general trend of following the lower comfort constraint in order to minimize en-

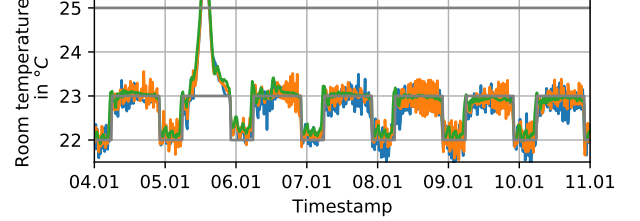


Figure 6: Room temperature trajectories in the MPC experiment with the RF models.

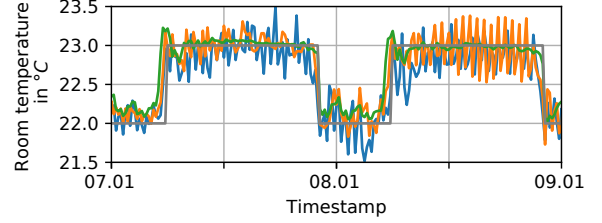


Figure 7: Details of room temperature trajectories in the MPC experiment with the RF models.

ergy is apparent. However, the controllers cause visible oscillations in the temperature trajectory. This is especially apparent in the detailed Figure 7. Here, it can also be seen that all controllers start heating early enough to meet the constraint at 06:00. Moreover, it is surprising to see that the model with the smallest amount of training data (green) leads to the smoothest control performance.

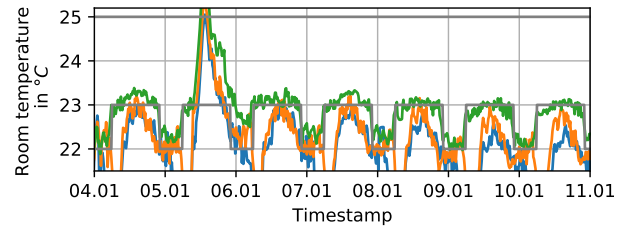


Figure 8: Room temperature trajectories in the MPC experiment with the ICNN models.

This effect of the short training set performing best, is even more prominent in the case of ICNN, as can be seen in Figures 8 and 9. While the controllers using the Full and Medium training set completely fail to meet the comfort constraints, the controller using the ICNN model based on the Short training set creates a trajectory that can be expected for a predictive controller.

The reason for this behaviour of the RF and ICNN

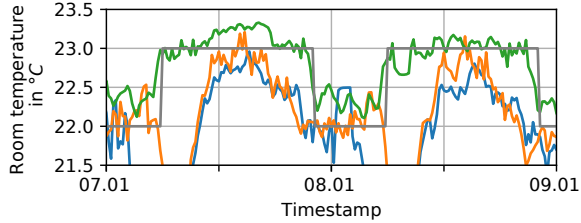


Figure 9: Details of room temperature trajectories in the MPC experiment with the ICNN models.

model is part of ongoing research. Possible explanations could be related to the non-convex nature of the training process, overfitting, or the training data being generated by a feed-back regulated system. It should also be noted that for both cases, RF and ICNN, the open-loop predictions are more accurate for longer training sets than for short ones, as discussed in the previous subsection. Similarity between training and testing data sets does not seem to play a role, as the same behaviour can be seen when the controllers are tested on a period in March.⁴

It can be assumed that functioning models for the longer training data periods could be obtained through extensive hyper parameter tuning. However, the given result demonstrates that the RF and ICNN hyper parameters cannot simply be transferred from case to case (i.e. from the experimental studies (Bünning et al., 2020, 2021) to the simulation study presented here), which is a disadvantage in the domain of building control, where every building is individual. Moreover, the results indicate that good controller behaviour cannot be deduced from open-loop accuracy for RF and ICNN.

Table 3 shows the cumulative comfort constraint violation and the energy consumption for all experiments. It can be seen that the consumed energy is similar for all cases except the two ICNN experiments that failed, which are written in italic. The comfort constraint violation results confirm the impressions from the room temperature trajectories. However, here it is more apparent that the ARX model benefits from a larger training set while the other models do not.

Computational resources

Besides controller performance, computational requirements are an important aspect of choosing a prediction model, as the MPC scheme should be able to run on micro controllers or cloud platforms outside the laboratory environment. Table 4 summarises the computational requirements for the three controllers. While data processing times and memory usage are in a similar order of magnitude for all models with a slight advantage for the ARX model, ARX and RF clearly outperform the ICNN in optimization problem solving time, which is due to the nature of the problem type (QP vs. quasi-convex problem with-

⁴Data not shown here.

out access to analytical gradients). However, as sampling times in building control related problems usually range from ten to thirty minutes, also the solving time for the ICNN model is well within acceptable range, which is also shown by the BOPTTEST time ratio KPI. The optimizations were conducted on a Intel(R) Core(TM) i7-7500U CPU with 2.7 GHz.

Table 4: Computational resources for MPC controllers with varying models.

Model	Data processing	Solving time	Time ratio (BOPTTEST)	Memory usage
ARX	<0.3 s	<0.01 s	0.00079	119 MB
RF	2.8 s	<0.02 s	0.00428	150 MB
ICNN	1.5 s	20-80 s	0.05348	135 MB

Conclusion

In this study, we have compared the performance of ARX models with a one-hot encoded solar model to models based on Random Forests and Input Convex Neural Networks for Data Predictive Control in buildings within the virtual test bed BOPTTEST. From the results, it is apparent that the ARX model with the one-hot encoded solar model outperforms both RF and ICNN in terms of open-loop accuracy, sample efficiency, closed-loop performance and computational resources. It should be noted that both ARX and RF models require little hyper parameter tuning (number of autoregressive terms, discretization of I_{vert} for the ARX model, and number of samples per leaves/tree depth, number of trees per forest for the RF model), while the ICNN model depends on many hyper parameters (number of layers, number of nodes per layer, training rate, training epochs, batch size, etc.). The hyper parameters of the ARX model can be chosen with some knowledge about the physics of the system, for example whether the building is a light or heavy construction. The sensitivity of RF and ANN regarding hyper parameters is discussed in (Ahmad et al., 2017) and (Rodriguez-Galiano et al., 2015), where RF are found to be more robust towards parameter selection.

Also, while the ARX model can rely on a broad background of theory for linear systems and system identification, the other models cannot. As the RF and ICNN generally allow to approximate more general sets of functions than the ARX model does, the question arises, why these models do not replicate the behaviour of the ARX model. For the ICNN this can be explained with the amount of parameters to be fit. Many processes in buildings are indeed linear, for example energy balances or heat conduction. The many degrees of freedom of the ICNN thus do not add to the model quality, but likely only to overfitting or fitting of fast modes of the system, which are not relevant for modelling the governing building dynamics but potentially increase the forecasting error for large horizons. In the case of the RF, the sequential (and thus non-convex) training process likely leads to not

finding the linear solution that the ARX models find. These observations match the ones we find in ongoing experiments on real systems.

The chosen test case is relatively simple, but the results are relevant as the case is similar to many buildings, i.e. residential buildings with hydronic systems. It is to be expected that the RF and ICNN models would perform better for more complex systems with more non-linearities. This will be tested as soon as BOPTTEST offers such cases.

Acknowledgement

We would like to sincerely thank Benjamin Huber and Adrian Schalbetter for previous work on the RF and ICNN models. We also thank Roy Smith and Mathias Hudoba de Badyn for fruitful discussions. This research project is financially supported by the Swiss Innovation Agency Innosuisse and is part of the Swiss Competence Center for Energy Research SC-CER FEEB&D. This work emerged from the IBPSA Project 1, an international project conducted under the umbrella of the International Building Performance Simulation Association (IBPSA).

References

- Ahmad, M. W., M. Mourshed, and Y. Rezgui (2017, jul). Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy and Buildings* 147, 77–89.
- Amos, B., L. Xu, and J. Z. Kolter (2017, jul). Input Convex Neural Networks. In *34th International Conference on Machine Learning, ICML 2017*, Volume 1, pp. 192–206. PMLR.
- Andersen, M. S., J. Dahl, and L. Vandenberghe (2004). CVXOPT.
- Blum, D., F. Jorissen, S. Huang, Y. Chen, J. Arroyo, K. Benne, Y. Li, V. Gavan, L. Rivalin, L. Helsen, D. Vrabie, M. Wetter, and M. Sofos (2019). Prototyping the BOPTTEST Framework for Simulation-Based Testing of Advanced Control Strategies in Buildings. In *Proceedings of the 16th IBPSA Conference*, pp. 2737–2744.
- Brockman, G., V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba (2016, jun). OpenAI Gym.
- Brownlee, J. (2018). Why One-Hot Encode Data in Machine Learning?
- Bünning, F., B. Huber, P. Heer, A. AbouDonia, and J. Lygeros (2020, mar). Experimental demonstration of data predictive control for energy optimization and thermal comfort in buildings. *Energy and Buildings* 211, 109792.
- Bünning, F., A. Schalbetter, A. AbouDonia, M. H. de Badyn, P. Heer, and J. Lygeros (2021). Input Convex Neural Networks for Building MPC. *presented at L4DC 2021*.
- Ljung, L. (1998). *System Identification: Theory for the User* (2 ed.). Pearson.
- Maddalena, E. T., Y. Lian, and C. N. Jones (2020, feb). Data-driven methods for building control — A review and promising future directions. *Control Engineering Practice* 95, 104211.
- Powell, M. J. D. (1994). A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In *Advances in Optimization and Numerical Analysis*, pp. 51–67. Springer Netherlands.
- Rodriguez-Galiano, V., M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas (2015, aug). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews* 71, 804–818.
- Schalbetter, A. (2020). Input Convex Neural Networks for Energy Optimization in an occupied Apartment.
- Smarra, F., A. Jain, T. de Rubeis, D. Ambrosini, A. D’Innocenzo, and R. Mangharam (2018). Data-driven model predictive control using random forests for building energy optimization and climate control. *Applied Energy* 226, 1252–1272.
- Sturzenegger, D., D. Gyalistras, M. Morari, and R. S. Smith (2016, jan). Model Predictive Climate Control of a Swiss Office Building: Implementation, Results, and Cost–Benefit Analysis. *IEEE Transactions on Control Systems Technology* 24(1), 1–12.
- Todorov, E., T. Erez, and Y. Tassa (2012). MuJoCo: A physics engine for model-based control. In *IEEE International Conference on Intelligent Robots and Systems*, pp. 5026–5033.
- Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, and T. Reddy (2020, mar). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 17(3), 261–272.
- Wang, J., S. Li, H. Chen, Y. Yuan, and Y. Huang (2019, aug). Data-driven model predictive control for building climate control: Three case studies on different buildings. *Building and Environment* 160, 106204.