



State-space models for building control: how deep should you go?

Baptiste Schubnel , Rafael E. Carrillo , Paolo Taddeo , Lluç Canal Casals ,
Jaume Salom , Yves Stauffer & Pierre-Jean Alet

To cite this article: Baptiste Schubnel , Rafael E. Carrillo , Paolo Taddeo , Lluç Canal Casals ,
Jaume Salom , Yves Stauffer & Pierre-Jean Alet (2020) State-space models for building control:
how deep should you go?, Journal of Building Performance Simulation, 13:6, 707-719, DOI:
[10.1080/19401493.2020.1817149](https://doi.org/10.1080/19401493.2020.1817149)

To link to this article: <https://doi.org/10.1080/19401493.2020.1817149>



Published online: 14 Sep 2020.



Submit your article to this journal [↗](#)



View related articles [↗](#)



View Crossmark data [↗](#)



State-space models for building control: how deep should you go?

Baptiste Schubnel^a, Rafael E. Carrillo^a, Paolo Taddeo^b, Lluç Canal Casals^{b,c}, Jaume Salom^b, Yves Stauffer^a and Pierre-Jean Alet^a

^aCentre Suisse d'Electronique et de Microtechnique (CSEM), Neuchâtel, Switzerland; ^bCatalonia Institute for Energy Research (IREC), Barcelona, Spain; ^cDepartment of Project and Construction Engineering, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

ABSTRACT

Power consumption in buildings show nonlinear behaviours that linear models cannot capture, whereas recurrent neural networks (RNNs) can. This ability makes RNNs attractive alternatives for the model-predictive control (MPC) of buildings. However, RNNs are nonlinear and non-smooth functions which makes their use challenging in optimization problems. Therefore, this work systematically investigates whether using RNNs for building control provides net gains in MPC. It compares over 2 months of simulated operation the representation power and control performance of two architectures: an RNN architecture and a linear state-space (LSS) model with a nonlinear regressor to estimate energy consumption. The results show that RNNs yield an identification error 69% lower than LSS, but the LSS models yield control laws that achieve 10% lower objective function with a computational time three times lower than the RNNs. Thus, on balance, well-designed LSS models with nonlinear regressors are best in most cases of MPC.

ARTICLE HISTORY

Received 12 June 2020
Accepted 22 August 2020

KEYWORDS

Building modelling; model predictive control; linear state-space models; neural networks; optimization

1. Introduction

Following the success of deep learning for imaging, text, and audio processing, architectures with (recurrent) neural networks are increasingly popular to model dynamic systems (Chung et al. 2014; Ogunmolu et al. 2016; Gonzalez and Yu 2018). Indeed, they provide a gain in prediction accuracy over linear systems model that is evidenced by both theoretical and experimental studies. For example, Funahashi and Nakamura (1993) demonstrated universal approximation properties of recurrent neural networks (RNNs). Several groups have sought to take advantage of this gain by using RNNs as predictive models within model predictive control (MPC), an optimal control procedure for dynamic systems (Camacho and Bordons 2007; Peng, Nakano, and Shioya 2007; Mayne 2014; Drgoña et al. 2018; Lanzetti et al. 2019; Bieker et al. 2020).

However, these models lead to optimization problems that are not guaranteed to be linear or convex. Convexity of the function encoded by an RNN can be enforced, for example, by imposing non-negative constraints on the weights and using rectified linear unit activations (ReLU). Such approaches show promising results for control applications (Amos, Xu, and Kolter 2017; Chen, Shi, and Zhang 2019, 2020). However, these approaches come at the expense of the universal approximation capabilities of RNNs for non-convex system dynamics. Because of

this trade-off, it is, therefore, an open question whether net gains in control performance can be obtained by using RNNs as predictors rather than linear state-space models.

Energy management in buildings is a representative illustration of this question. Indeed, buildings are dynamic systems for which the common practice is to use rule-based controllers and state of the art is MPC based on linear state-space models (Sourbron, Verhelst, and Helsen 2013; Sturzenegger et al. 2016). These models are adequate and reliable for room temperature (Gorecki 2017; Péan, Salom, and Costa-Castelló 2019) but cannot capture some nonlinear phenomena due to shading and solar gain, low-level control loops, or physical characteristics of technical systems such as heat pumps (Valenzuela et al. 2020). The authors of Chen, Shi, and Zhang (2017) used 'long short-term memory' (LSTM), an RNN architecture, to model a complex simulated building and to minimize energy consumption through optimal control. They claimed some improvements over linear state-space models based on a limited amount of validation data. Therefore, this paper sets about evaluating whether there are net, systematic gains in the performance of building control by using neural networks within MPC as compared to state-of-the-art linear state-space models.

In order to isolate the impact of control from random variations in, for example, weather and occupancy, and to run long experiments in identical conditions, we chose to apply the different control methods on a detailed physical model of a single building. We trained and used for control five instances for each of two types of models: (1) linear state-space models of the building envelope combined with a nonlinear regressor to estimate energy consumption, and (2) fully nonlinear, recurrent neural network models. In addition, we implemented rule-based controllers as benchmarks. We evaluated the performance of the controllers over 2 months of operations. To ensure a fair comparison, we used exactly the same optimization constraints and the same optimization algorithm (sequential quadratic programming) for both model architectures.

Based on this experimental approach, the paper evaluates the relative strengths and weaknesses of linear state-space models and RNNs in terms of:

- representation capabilities by training the different instances under several strategies and datasets;
- control performance in a constrained optimization problem consisting in minimizing the power exchanged between the grid and a building with photovoltaic (PV) production under the constraint of thermal comfort in the building;
- computing time.

None of the two approaches outperforms the other on all criteria. Our investigation confirmed the superior representation capability of RNNs. However, on balance, linear state-space models appear better suited to model-predictive control, except if the application emphasizes minimizing constraint violations.

The rest of the paper is organized as follows. Section 2 describes the simulated building. Section 3 presents the two types of models used in this study. Section 4 explains the optimization strategy used to solve the non-convex problems within the MPC as well as the heuristics used to improve the performance of the objective minimization. Sections 5 and 6, respectively, present the results of system identification and control performance results in Section 6. Section 7 concludes the paper with a discussion and directions for future work. The appendix provides details on the hyperparameters of the models and the training parameters.

2. Building test case

A building simulated with EnergyPlus (Crawley et al. 2000) was used to carry out the systematic performance analysis. The simulation time resolution was equal to 3 min,

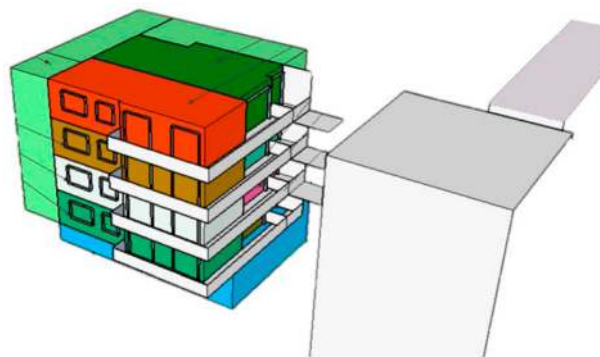


Figure 1. Building envelope overview with the four floors. There are two thermal zones per apartment.

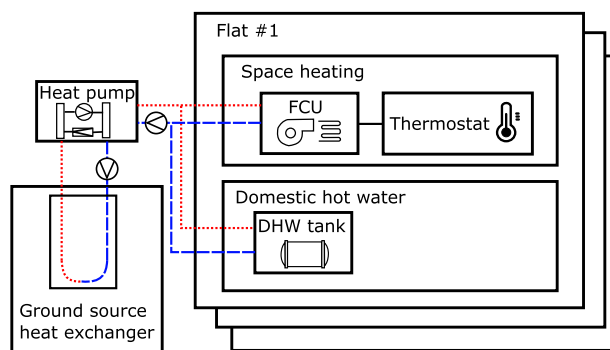


Figure 2. Schematic view of the heating and distribution system.

but control variables were updated by the MPC every 15 min. The simulated building is a residential building with four apartments and eight thermal zones, located in Barcelona, Spain (Figure 1). It has two major technical systems: a centralized geothermal heat pump and a photovoltaic installation that has a peak power of 10.8 kW.

The centralized water-to-water geothermal heat pump system, which extracts heat from the ground through a vertical ground heat exchanger, provides hot water for the indoor fan coil units (two units per apartment) and the domestic hot water (DHW) system. The domestic hot water system is composed of four storage tanks, one for each household. The heating loop circuit from the heat pump is connected to the bottom half part of the tanks and electrical heaters are placed on the top part acting as auxiliary heating systems to meet domestic hot water demand. The heat pump is connected to four tanks for hot water consumption and to a heating loop for rooms thermal regulation; see Figure 2.

Room temperature control is carried out with thermostats and by a low-level control loop with an EnergyPlus Program Manager that implements a hysteresis control based on thermostats setpoints (SP) and bottom tank temperature setpoints. It triggers the on/off cycles

Table 1. Setpoints ranges.

Controlled setpoints	Range (°C)
Heat pump supply temp.	[40, 55]
Tanks bottom temp. (4 SP)	[35, 50]
Zone temp. (4 SP)	[19, 25]

of the heat pump based on setpoints hysteresis violations. The low-level control loop coupled with the heat pump is responsible for nonlinear behaviour of the power consumption as a function of the room thermostat temperatures and the pump supply temperature. The heat pump properties have been carefully calibrated to match the behaviour of a real geothermal heat pump in the IREC laboratory. See Taddeo et al. (2020) for further details of the EnergyPlus model where its thermal envelope was used with a real heat pump in the IREC laboratory.

Daily schedule patterns for occupancy, hot water, light and appliances consumption are used to model activity in the building. The domestic hot water (DHW) profiles used in our study are based on the European standard EN 16147 (European Standard 2011). This standard provides energy consumption for different daily DHW usage and tapping profiles. In the present study, the ‘medium’ tapping cycle and an equivalent DHW temperature of 60°C have been selected and adapted to the occupancy and simulation time step. Control variables are the room thermostats (4 setpoints, one per apartment), the tanks bottom temperatures (4 setpoints) and the heat pump supply temperature. Restrictions are imposed on the range of allowed values as displayed on Table 1. ° Moreover, the bottom temperatures of the tanks have to stay 5°C lower than the heat pump supply temperature.

The photovoltaic (PV) installation is simulated as an array of PV modules with inverter. The PV array has an active surface area of 58 m² and is oriented South with an inclination of 40°. The full geometric model for solar radiation is used, including shading and reflections, to determine the incident solar resource. The EnergyPlus PV model was parametrized to match the real PV system installed on the IREC laboratory roof in Tarragona (Taddeo et al. 2020).

3. Linear and nonlinear state-space models

Two different model architectures were used to estimate rooms temperatures and electrical energy consumption:

- The first model is a linear state-space model that is used to estimate room temperatures (see, e.g. Van Overschee and De Moor 2012). A nonlinear regressor is used to estimate electrical energy consumption by the thermal system using state-space model predictions.

Model instances built with this architecture are called LSS-NL in the rest of the paper.

- The second model is a fully nonlinear state-space model based on an encoder-decoder architecture with LSTMs (Hochreiter and Schmidhuber 1997). It simultaneously estimates temperature in the rooms and electrical energy consumption by the thermal systems. Model instances built with this architecture are called ENC-DEC in the rest of the paper.

3.1. Linear state-space models with nonlinear regression

A linear state-space model follows the set of equations:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + \varepsilon_k, \\ y_k &= Cx_k + Du_k + \varepsilon'_k, \end{aligned} \quad (1)$$

where k is the discrete timestep, $x_k \in \mathbb{R}^h$ is the state variable at time k , $u_k \in \mathbb{R}^{ex}$ are the inputs at time k (or exogenous variables), $y_k \in \mathbb{R}^{end}$ are the modelled system outputs (endogenous variables) at time k and $\varepsilon_k, \varepsilon'_k$ are white random vectors with zero mean. Initially, internal state variables x_k are estimated from a sequence of observations y_1, \dots, y_n using the forward innovation Kalman equation (Van Overschee and De Moor 2012). For the building test case, u_k in Equation (1) is a concatenation of the control commands and the external disturbances at time step k . The control commands are the temperature setpoints for the zone’s thermostats, tanks and heat pump forward temperature. The external disturbances are the solar irradiance, outdoor temperature and humidity. The output variables y_k in Equation (1) are the rooms temperatures. Even if the state variable x_k in Equation (1) has a priori no physical meaning, correspondence between RC models and linear state-space models for buildings leads to the interpretation of x_k as representing effective dynamics of internal components, e.g. walls and sublayers temperatures.

Electrical energy consumption in buildings may exhibit nonlinear behaviour, in particular if systems like heat pumps and HVACs are used for heating and/or cooling the building. The nonlinear behaviour may also arise from low-level controllers that cannot be directly controlled by the MPC controller. A kernel regression with radial basis functions was chosen to model the electric energy consumption. More specifically, at any time k , nonlinear variables $z_k \in \mathbb{R}^{nl}$ (here the electric energy consumption) are modelled by

$$z_k = \sum_{i=1}^N \alpha_i \varphi(w_i, w_k), \quad (2)$$

where $\varphi(x, y) = \exp(-\gamma \|y - x\|^2)$ is the kernel function, $w_k := (u_k, y_k)$ is the concatenation of the inputs and output variables, and N is the number of points in the training set. The right-hand side in Equation (2) is a closed-form expression which describes a smooth function. These two properties are advantages of kernel regression over other nonlinear regression methods (see, e.g. Papadopoulos et al. 2018; Chakraborty and Elzarka 2019) if the optimization techniques used for the model predictive controller require gradient or hessian estimates of f , as discussed below in Section 4.

3.2. Nonlinear state-space model

The nonlinear state-space model is based on an encoder-decoder architecture where both the encoder and the decoder are long short-term memory (LSTM) cells (Hochreiter and Schmidhuber 1997). The encoder is an LSTM-network that is used instead of a Kalman filter to initialize the state of the physical system. The decoder is another LSTM network followed by a multilayer perceptron (MLP) that is used to predict the model outputs step by step. Let $n \in \mathbb{N}$ be the number of encoder steps used to initialize the model. A schematic view of the neural architecture is given on Figure 3.

The encoder cell state and output at time t are vectors of size p respectively denoted by $c_{\text{enc}}(t) \in \mathbb{R}^p$ and $h_{\text{enc}}(t) \in \mathbb{R}^p$. The concatenation $(c_{\text{enc}}, h_{\text{enc}})$ plays a similar role as the state variable x in linear state-space models; see Equation (1). Like the linear Kalman filter,

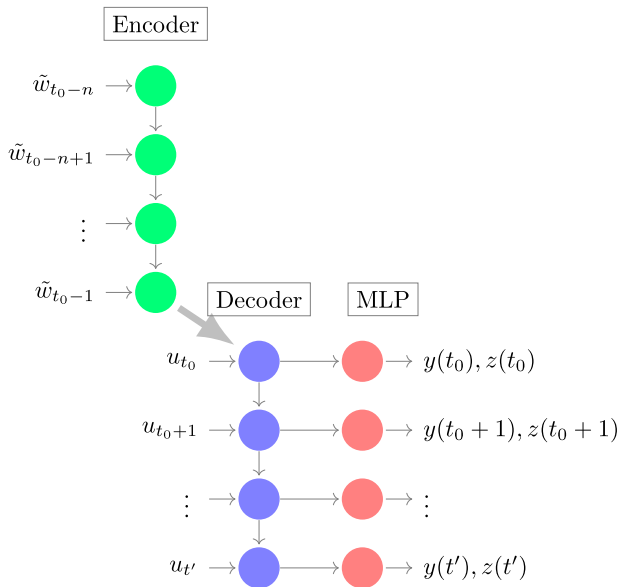


Figure 3. Structure of a nonlinear state-space model based on an encoder-decoder architecture.

the LSTM encoder is used to generate the representation of the initial system state at time t_0 , given the past n values. Setting $\tilde{w} := (u, y, z)$ as a concatenation of exogenous variables, room temperature variables and energy variables, the encoder can be viewed as an iterative application of the map $f_{\text{enc}} : \mathbb{R}^{\text{ex}+\text{end}+nI+2p} \rightarrow \mathbb{R}^{2p}$ such that the tuple $(c_{\text{enc}}(t), h_{\text{enc}}(t))$ is a representation of the initial state of the system for $t = t_0 - 1$ provided that Equation 3 has been repeated n times from $t_0 - n$ to $t_0 - 1$.

$$(c_{\text{enc}}(t), h_{\text{enc}}(t)) = f_{\text{enc}}(\tilde{w}(t), c_{\text{enc}}(t-1), h_{\text{enc}}(t-1)). \quad (3)$$

For $t' \geq t_0$, the decoder LSTM is a map $f_{\text{dec}} : \mathbb{R}^{\text{ex}+2p} \rightarrow \mathbb{R}^{2p}$ that only depends on the commands and external parameters (but not on previous system observables y nor on energy variables z):

$$(c_{\text{dec}}(t'), h_{\text{dec}}(t')) = f_{\text{dec}}(u(t'), c_{\text{dec}}(t'-1), h_{\text{dec}}(t'-1)), \quad (4)$$

where $c_{\text{dec}}(t_0 - 1) = c_{\text{enc}}(t_0 - 1)$ and $h_{\text{dec}}(t_0 - 1) = h_{\text{enc}}(t_0 - 1)$.

The output $h_{\text{dec}}(t')$ is fed to a multilayer perceptron network, giving rise to the estimated output $y(t'), z(t')$ at time t' :

$$y(t'), z(t') = f_{\text{MLP}}(h_{\text{dec}}(t')). \quad (5)$$

Both architectures were trained by minimizing the mean square error between observations and predictions. For both architectures, the electric power production from photovoltaic panels was estimated independently using a simple physical model from the PVLIB library (Stein et al. 2016) since the geometry of the PV system at the IREC site was simple. The physical model has as input solar irradiance forecasts.

3.3. Training method

The architectures have different sampling efficiency for system identification and different datasets were therefore used to identify the building models.

Linear state-space models identification techniques are well covered in the literature; see, e.g. Ljung (1999) and Van Overschee and De Moor (2012). Linear state-space system identification is very sample efficient, theoretically founded, and a few days of observations are sufficient for rooms temperature identification if signal excitations can be sent to the building. In this paper, the N4SID method is used to carry out rooms' dynamics identification; see Van Overschee and De Moor (2012). Nonlinear

variables identification with kernel ridge regression is sample-efficient as well.

On the other hand, the encoder–decoder architecture is sample inefficient and tends to be overparametrized (see, e.g. Nakkiran et al. 2020 for an interesting empirical study of over-parametrization in Neural Networks architecture and its effect on model accuracy). It needs to ingest data under various conditions and setpoints to avoid model overfit, and a period of at least a year of data is needed to perform good identification (to cover the four seasons). It is trained by stochastic gradient descent, with no optimal fitting guarantee because of the non-convexity of the architecture.

For the linear state-space model with nonlinear outputs, between 7 and 40 days of data with random multisine signals excitation (see, e.g. Schoukens and Ljung 2019) in winter times were used to capture room dynamics and electrical power from thermal sources. The distinct numbers of days considered as well as the random character of the excitations were used to build five models with distinct dynamic matrices A , B , C and D . The models were kept constant over the 2 months evaluation phase, as degradation in models predictions were observed in the case where refit of the matrices A , B , C , D was carried out online with collected data from the MPC. The encoder–decoder model instances were trained on 7 years of data with piecewise constant setpoints updated at a random frequency and with random amplitudes lying in an acceptable physical range. Prediction range were arbitrarily sampled between 1 h and 1 day ahead during training, to ensure that the model was able to capture short- and long-range dynamics. As for the linear state-space models, five different models were built using stochasticity of the training by mini-batch gradient descent, as well as by varying the number of training steps. Models hyperparameters are detailed for all models in Appendix 1. The size of the training sets are based on the intrinsic requirements of each architecture. A large training data set is necessary for the encoder-decoder architecture, because of the high number of parameters of the model and because it is trained with stochastic gradient descent. On the other hand, the linear state-space model is defined by a small number of parameters (A in Equation (1) is an 8×8 matrix); the identification results in Tables A5 and A6 show that even within the reported range of training periods, which are significantly shorter than for the RNN architecture, accuracy decreases with increasing the size of the training set. Additional experiments conducted by the authors (not reported here) with 6 months of training data for linear state-space models with nonlinear output actually showed a worsening of the identification accuracy.

4. Formulation of the model predictive controller

The objective of the model predictive controller is to minimize the power exchanged between the grid and the building without violating room temperature comfort constraints. The problem horizon is denoted by H and the time resolution by Δ . Splitting the exogenous variables $u \equiv (u_c, u_{nc})$ into controllable (setpoint inputs) and non-controllable variables (exterior parameters) and introducing the *model function* $m : \mathbb{R}^{ex \times H} \rightarrow \mathbb{R}^{(end+nl) \times H}$ that predicts room temperatures and nonlinear variables (electrical energy consumption), the optimization problem at every discrete optimization starting time t takes the form

$$\begin{aligned} \min_{\underline{u}_c \in \mathbb{R}^{ex_c \times H}} \quad & f(m(\underline{u})) \\ \text{s.t.} \quad & g(m(\underline{u})) \leq 0, \end{aligned} \quad (6)$$

where f denotes the objective function and g the constraint vector. The initialization phase is neglected here to simplify notations.

In the present case, the objective function f is the mean absolute difference between the electric power consumed by all equipment, $P_{el}(t)$, and the PV power produced, $P_{prod}(t)$, i.e.

$$f(m(\underline{u})) = \sum_{i=1}^H |P_{el}(t + i\Delta) - P_{prod}(t + i\Delta)|, \quad (7)$$

for $\underline{u} = (u(t + \Delta), \dots, u(t + H\Delta))$. The equipment comprises appliances, lighting system and thermal systems, i.e. heat pump, auxiliary pumps and associated fans. The components of the constraint vector g are the room constraints $T_r - 24^\circ\text{C}$ and $19^\circ\text{C} - T_r$, that apply to all eight rooms, as well as the setpoint constraints in Table 1. As commonly done in MPC, problem (6) is solved at discrete optimization time t over the entire horizon H but only the first setpoint $u(t + \Delta)$ is applied, and a new problem is formulated starting at time $t + \Delta$ in a receding horizon fashion.

The problem in Equation (6) is non-convex for both models and was solved using sequential quadratic programming (SQP); see e.g. Gill and Wong (2012). SQP involves the minimization of a quadratic approximation of the objective function $f(m(\underline{u}))$ under a linear approximation of the constraint vector $g(m(\underline{u}))$. It solves iteratively the problem starting from an initial trajectory $\underline{u}^{(0)}$, making local quadratic expansion around each solution $\underline{u}^{(k)}$, $k \geq 0$, and solving the quadratic subproblem in Equation 8 to obtain the next value $\underline{u}^{(k+1)}$. Iterations stop when the difference between two subsequent val-

ues becomes negligible or when a maximum number of iterations (e.g. 12) is reached.

$$\begin{aligned} \min_{\underline{u}_c \in \mathbb{R}^{exc \times H}} \quad & f(m(\underline{u}^{(k)})) + \nabla(f \circ m)_{|\underline{u}^{(k)}}^T (\underline{u}_c - \underline{u}^{(k)}) \\ & + \frac{1}{2} (\underline{u}_c - \underline{u}^{(k)})^T Q(\underline{u}^{(k)}) (\underline{u}_c - \underline{u}^{(k)}) \\ \text{s.t.} \quad & g(m(\underline{u}^{(k)})) + \nabla(g \circ m)_{|\underline{u}^{(k)}}^T (\underline{u}_c - \underline{u}^{(k)}) \leq 0, \end{aligned} \quad (8)$$

for all $k \geq 0$. The matrix Q denotes the Hessian of the objective $f \circ m$ of the original problem. In practical algorithms, important speed up and accuracy gains arise if the gradients of the objective and constraint functions can be computed without using finite difference methods. In the present work, for the linear state-space method with kernel regressor, the explicit expression of the nonlinear output is a net advantage in terms of speed and numerical stability as it can be differentiated easily – see Sections 6.1 and 6.2. The nonlinear state-space model can use the automatic differentiation capabilities of the deep learning libraries like Tensorflow or Pytorch (Abadi et al. 2015) to compute the model Jacobians. In that case, estimating gradients with Graphics Processing Unit (GPU) provides better and faster estimates than finite difference methods. Hessians are estimated using the LBFGS secant method (Fletcher 1987).

Since the problem in Equation (6) is non-convex, a key point to get good solutions with SQP is the choice of the initial solution trajectory $\underline{u}^{(0)}$. To stabilize the optimization process with the fully nonlinear state-space model, we used a shifted and smoothed version of the solution at the previous optimization step as initial trajectory. The resulting trajectories are presented in Section 6.1.

5. Identification results

To assess the representation capabilities of each architecture, models were evaluated on a test set of setpoints and external parameters not seen during the identification phase. The synthetic test set is made of weather data observed in Barcelona from January to end of April. The setpoints signals in this synthetic set (heat pump, tanks and thermostats setpoints) are similar to excitation signals used for system identification in real buildings. They consist of a random profile of sinusoidal, square and triangular setpoints, with varying frequencies over time. These signals are used to check that the models adequately reproduce the dynamic of the building simulation under various excitations. The evaluation windows were shifted on a fixed basis and prediction accuracy over short and long range were computed. At time t , both LSS-NL and encoder-decoder models were used to predict m

timesteps ahead until $t + m$ given the last observed data points. Then the windows were shifted m steps ahead and evaluation was performed starting at $t + m$, and so on. Results are displayed for $m = 4$ (1-h-ahead prediction) and $m = 96$ (1-day-ahead prediction) on Figure 4(a,b). In Appendix 2, all the numerical values of the mean absolute error (MAE) and symmetric mean relative absolute error (sMRAE) are tabulated. They are defined for N predictions $\hat{y}(t_i)$ and observations $y(t_i)$ by

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}(t_i) - y(t_i)|, \quad (9)$$

$$\text{sMRAE} = \frac{2}{N} \sum_{i=1}^N \frac{|\hat{y}(t_i) - y(t_i)|}{|\hat{y}(t_i)| + |y(t_i)|}. \quad (10)$$

Encoder–decoder architectures (the five models labelled by ENC-DEC1, . . . , ENC-DEC5) exhibit significantly lower prediction errors both on 1-h and 1-day-ahead predictions. For 1-h-ahead forecast, encoder–decoder architectures have a mean absolute error between 0.19 and 0.24°C for temperature predictions and between 1.90 kW and 2.50 kW for power predictions, whereas linear state-space models (LSS-NL1 to 5) have errors on temperature and power between 0.64 and 0.77°C and between 3.24 kW and 4.32 kW, respectively (see Table A6 in Appendix 2). For comparison, the peak power generation by the PV system is 10.8 kW and the average power consumption under rule-based control is between 5 and 6 kW. For-day-ahead forecasts, errors grow but exhibit similar ratios between encoder-decoder models and linear state-space models (see Table A5 in Appendix 2). Figure 5 shows the temperature error distributions for 1-day-ahead predictions for the best model fit of both architectures. Power error distributions show similar patterns, with the error distribution being broader for the linear state-space models than the encoder–decoder architectures.

6. MPC results

6.1. Control performance

Four metrics are used to estimate the control performance of the model predictive controllers. The two first metrics are directly related to the optimization objective and the constraints. The first metric is the mean power exchanged with the grid, given by

$$\bar{P} = \frac{1}{N} \sum_{n=1}^N |P_{e_i}(t_n) - P_{\text{prod}}(t_n)|. \quad (11)$$

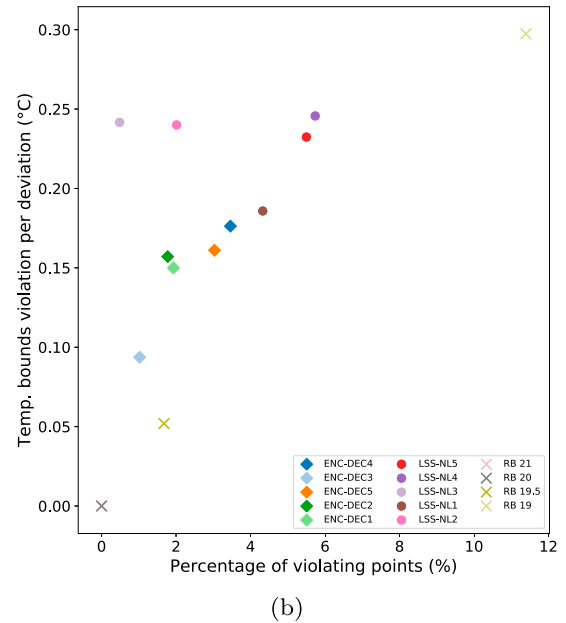
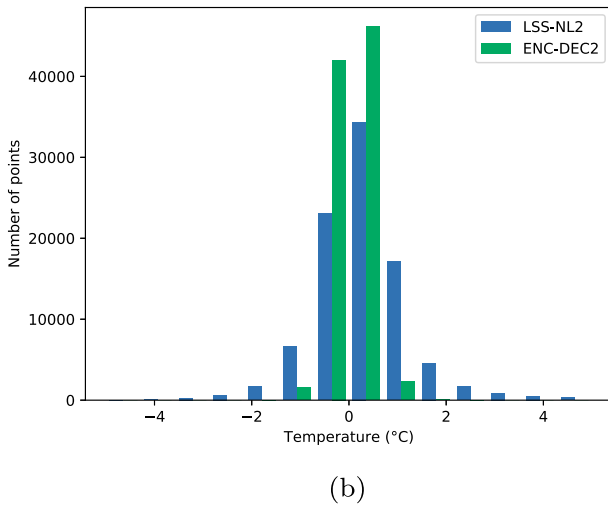
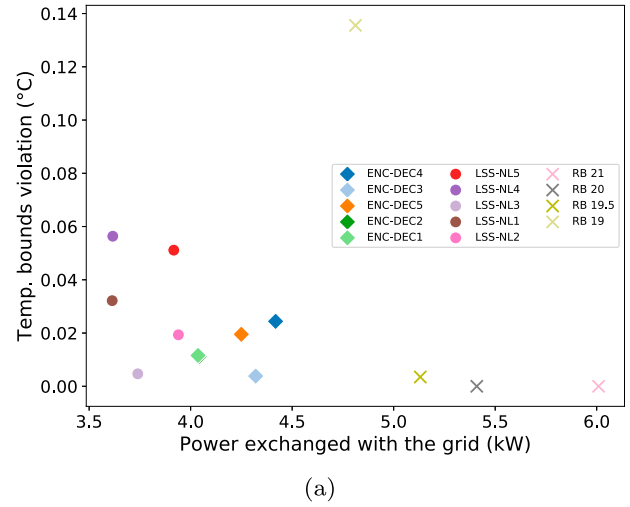
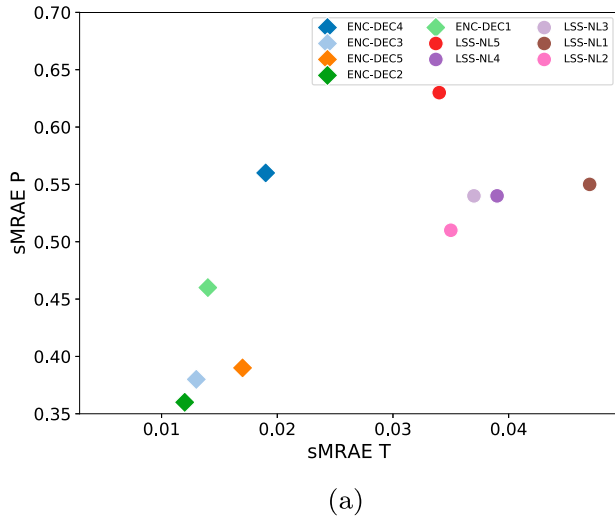


Figure 4. (a) sMRAE for temperature and power predictions at 1 day ahead, for the five instances of each model architectures. ENC-DEC denotes encoder–decoder model instances, and LSS-NL linear state-space model instances with nonlinear power prediction; see Section 3. (b) Temperature error distribution for the best two model instances of each architecture at 1 day ahead.

The second metrics is used to evaluate mean comfort bounds violation, and is defined by

$$\bar{C} := \frac{1}{8N} \sum_{n=1}^N \sum_{r=1}^8 (\chi(T_r(t_n)) > 24)(T_r(t_n) - 24) + \chi(T_r(t_n) < 19)(19 - T_r(t_n)), \quad (12)$$

where T_r is the measured temperature of room r and χ is the Heaviside step function. The metric \bar{C} has the weakness to be relatively insensible to the magnitude of the temperature violations, as the factor N^{-1} provides an average over the entire evaluation period. To overcome this issue the third metric, $\bar{\Delta}$, estimates the mean

Figure 5. (a) Mean self-consumed power \bar{P} and temperature deviation \bar{C} for the model predictive controllers, as compared to rule-based controllers. ENC-DEC denotes encoder–decoder model instances, and LSS-NL linear state-space model instances with nonlinear power prediction; see Section 3. (b) Percentage of deviating points versus mean temperature violation per deviation.

temperature deviation per violating points, i.e.

$$\bar{\Delta} = \frac{8N}{N_{\text{out}}} \bar{C}, \quad (13)$$

where N_{out} is the number of points lying outside the interval $[19, 24]^\circ\text{C}$. Finally, the mean power electric consumption by the thermal systems are also reported to make sure that optimizers do not globally consume more than rule-based controllers:

$$\bar{P}_{\text{th}} = \frac{1}{N} \sum_{n=1}^N P_{\text{th}}(t_n). \quad (14)$$

Table 2. Key performance indicators for rule-based controllers and MPC on all model instances.

Model	\bar{P} (kW)	\bar{P}_{th} (kW)	\bar{C} ($^{\circ}\text{C}$)	$\bar{\Delta}$ ($^{\circ}\text{C}$)
RB 19 $^{\circ}\text{C}$	4.81	4.46	1.35E-1	2.97E-1
RB 19.5 $^{\circ}\text{C}$	5.13	4.91	3.48E-3	5.19E-2
RB 20 $^{\circ}\text{C}$	5.41	5.32	0	0
RB 21 $^{\circ}\text{C}$	6.01	6.22	0	0
LSS-NL1	3.61	3.41	3.21E-2	1.86E-1
LSS-NL2	3.94	3.95	1.93E-2	2.40E-1
LSS-NL3	3.74	3.63	4.7E-3	2.42E-1
LSS-NL4	3.62	3.56	1.05E-1	2.46E-1
LSS-NL5	3.92	3.62	5.11E-2	2.23E-1
ENC-DEC1	4.03	3.92	1.16E-2	1.50E-1
ENC-DEC2	4.04	3.91	1.11E-2	1.57E-1
ENC-DEC3	4.32	4.08	3.84E-3	9.37E-2
ENC-DEC4	4.42	4.30	2.44E-2	1.76E-1
ENC-DEC5	4.25	4.07	1.95E-2	1.61E-1

All metrics are evaluated over 2 months in Barcelona, in February and March. The models were evaluated under the hypothesis of perfect weather predictions.¹ The MPC controllers were compared against four rule-based controllers (abbreviated RB in Table 2), that tried to keep the zones temperature close to the following constant setpoints: 19, 19.5, 20 and 21 $^{\circ}\text{C}$. Results are reported on Table 2 and Figure 5(a,b). Properties of rule-based controllers can be found in Appendix 1, where rule-based commands for thermostats, tanks and heat pump are given.

From Table 2 and Figure 5(a,b), the following key points can be observed:

- All optimizers, even the worst ones, consistently outperforms the implemented rule-based controllers in terms of self-consumption and power consumption from thermal devices.
- The linear state-space models show better results in power consumption metrics than the encoder-decoder models. However, as displayed clearly on Figure 5(b), they come with the drawback of violating the constraints in a stronger way than the full nonlinear models. On the trajectory displayed on Figure 8, the linear state-space MPC is clearly violating the comfort rules at low temperature (light green curve), indicating that the model has not caught heating saturation when the heat pump temperature is too low. The nonlinear MPC, however, only does a very minor violation. Scatter plots on Figure 6(b) clearly show the superiority of encoder-decoder models for comfort, as well as the low temperature excursions of the controllers with linear state-space models.
- This distinct behaviour at low exterior temperature is due to the limited identification capabilities of linear state-space models. Due to the low-level hysteresis control loop (and such loops are often encountered

in real buildings), the room temperature evolution at 15 min timescale follows two distinct regimes: It either oscillates between the control band if the supply temperature of the pump is high enough, or it decreases because not enough heat is supplied to maintain the temperature of the room close to the setpoint (which is even enhanced by the similar temperature drops happening simultaneously in all other rooms of the building). The linear state-space model has to find a linear compromise between the two regimes, that will necessarily miss some of either the saturation or the oscillation in the dead band interval. Since saturated regimes occurs at a very low external temperature, that is rare in the considered geographical

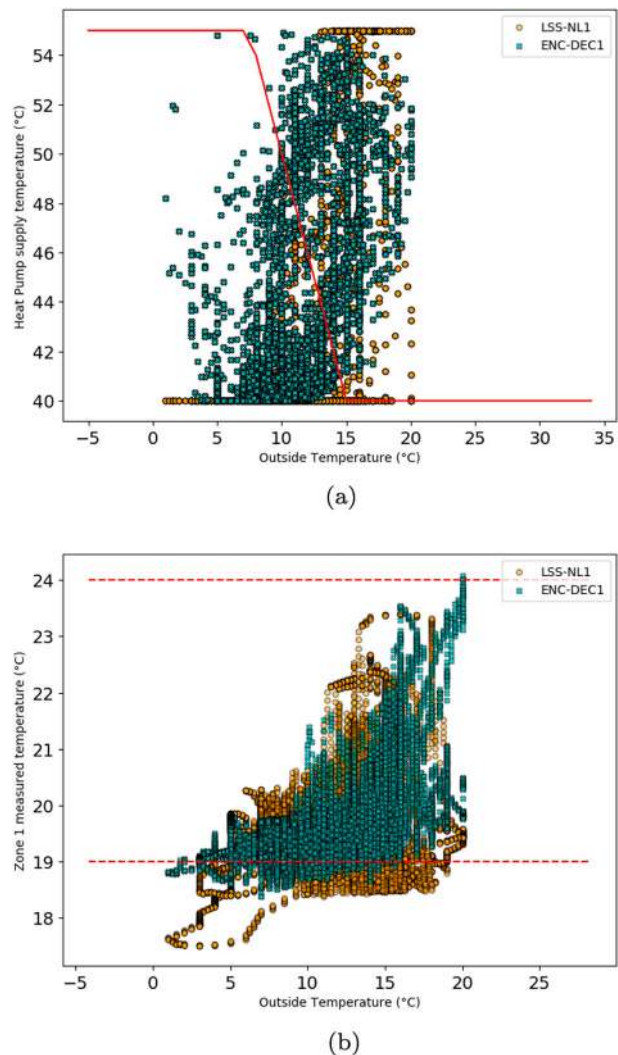


Figure 6. (a) Heating curve for the rule-based controllers (full curve), as compared to MPC Heat pump temperature supply scatter plots for two of the best performing models. (b) Zone 1 room temperature measured: Scatter plots for two of the best performing models. Linear state-space controllers exhibit clear deviations at low outside temperatures.

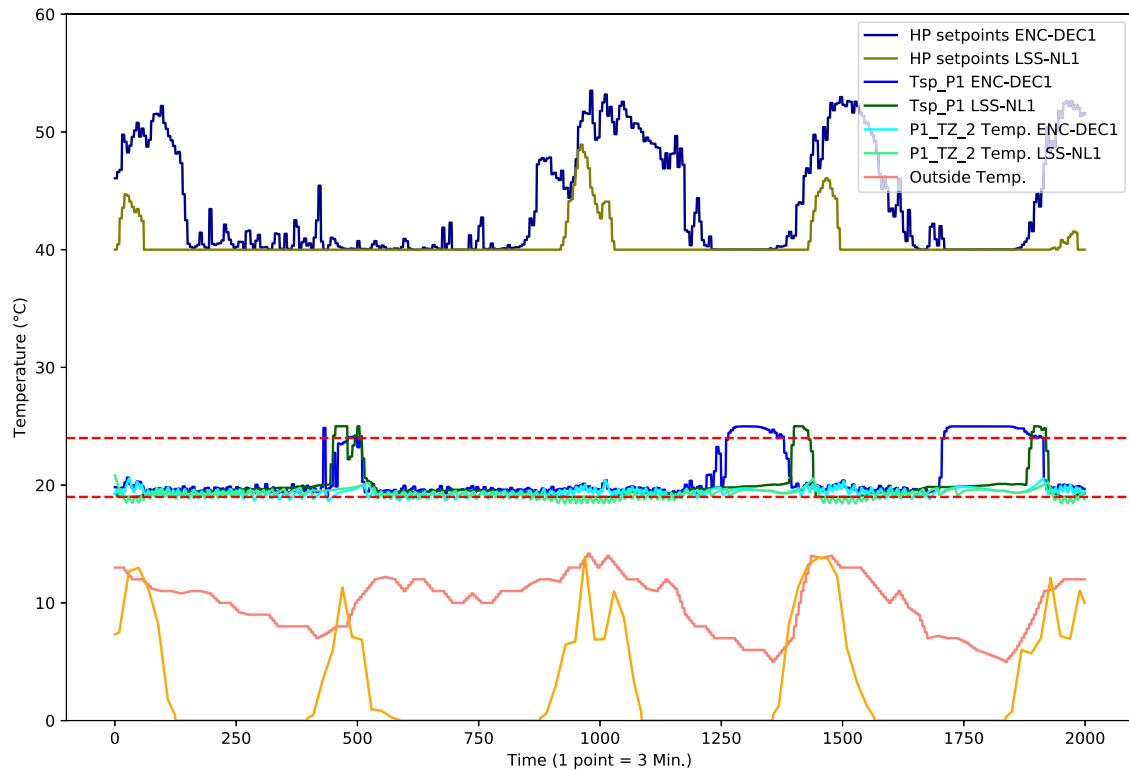


Figure 7. Example of trajectories for two of the best performing models of each architecture. Dashed horizontal curves show the limit range of 19 and 24°C. Normalized irradiance data are displayed at the very bottom of the figure.

region, the saturated regime will not be well reproduced by the LSS model (whatever the amount of data used).

- To take advantage of the heat pump COP and the power generated by the photovoltaic panels, the two MPC architectures use a completely different strategy than typical heating curves used for rule-based controllers. At low outside temperature, both controllers lower the heat pump temperature to get a higher COP (Figure 6(a)).
- Hours with higher ambient temperatures are correlated with higher irradiance, and thus a high probability of having an excess of PV. So the controllers tend to increase the room setpoint temperatures as a strategy to preheat the buildings and reduce power exchange with the grid; see Figure 6(b) as well as Figure 7.

These results show the difficulty of optimizing nonlinear state-space models with local optimization techniques. Local methods like SQP find it hard to escape a minimum and are quite sensitive to the initial trajectory solution. In the fully nonlinear case, due to the high number of parameters and the non-convexity of the model, the objective hypersurface will present many local minima. On the contrary, for linear state-space models with smooth nonlinear outputs (like the kernel regression

used in this paper), fewer local minima are expected. This effect is well illustrated on the heat pump trajectories on Figure 7. The heat pump setpoints for the nonlinear MPC (ENC-DEC1, in dark blue) slowly react to external conditions, and the optimizer takes time, at low temperature, to reach setpoints close to 40°C. On the contrary, transitions from the linear state-space model controller (LSS-NL1, in olive) are much faster, and setpoints quickly make the transition from high temperatures heat pump supply to the lower bound of 40°C. Having a low supply temperature increases the COP of the heat pump, and therefore reduces energy consumption, especially at night. The fact that the SQP with the nonlinear models takes more time to achieve the temperature transition lowers its energy performance.

6.2. Execution time

Model identification and controller evaluations were done for both architectures on the same workstation. This workstation is equipped with an Intel i9-9960X CPU with a nominal frequency of 3.10 GHz and 16 physical cores, and an Nvidia RTX 2080 TI Graphics Processing Unit (GPU). The GPU was used to speed up computations with the neural network architectures, in particular the gradient evaluations. The use of the CPUs was intensified through

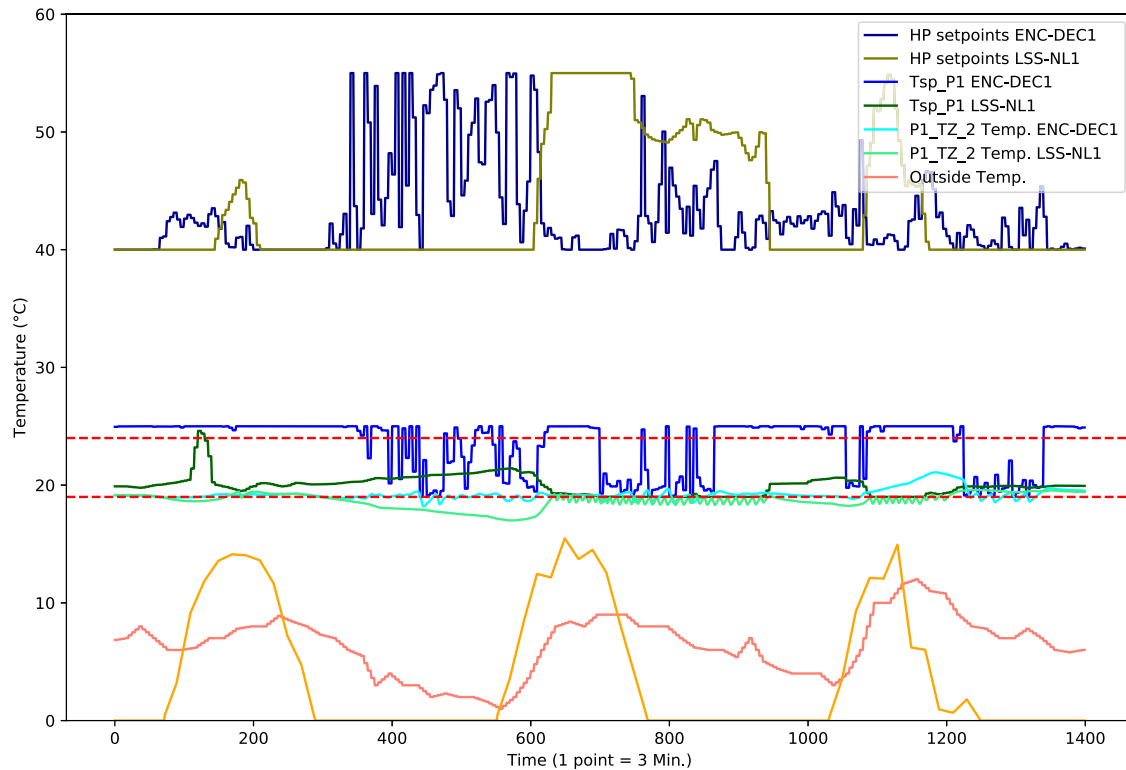


Figure 8. Example of trajectories for two of the best performing models of each architecture. Dashed horizontal curves show the limit range of 19 and 24°C. Normalized irradiance data are displayed at the very bottom of the figure.

multiprocessing to speed up the gradient analytical computations in the optimization of the linear state-space models with kernel outputs.

The encoder–decoder architecture required about one hour for training (depending on the number of iterations) and the linear state-space model about 1 min. In both cases, this training time is negligible over the lifetime of the system in comparison with the computation time required to solve the optimal control problems.

For both architectures, the execution speed of the controllers was optimized. Jacobians in the SQP optimization were evaluated with Tensorflow 1.14 on the GPU for speed up, providing better time execution than numerical approximation with finite differences. Optimization coupled with the encoder–decoder models took around 4.5 min per control step. Analytic computations of the Jacobians of the linear state-space model with nonlinear outputs were carried out using multiprocessing python package Numba (Lam, Pitrou, and Seibert 2015) and using all the cores at disposal. Optimization coupled with the linear state-space models with nonlinear outputs took around 1.3 min per control step. The two methods, therefore, require a computation time in the same order of magnitude at each control step but the linear state-space model results in an optimization more than three times as fast as the encoder-decoder architecture.

Table 3. Comparison summary of the architectures capabilities.

Properties	LSS-NL	ENC-DEC
Sample efficiency Sys. Id.	+	-
Accuracy System Id.	-	+
Objective minimization	+	-
Respect of constraints	-	+
Computation time	+	-

7. Conclusion

A comparison of two model architectures for MPC for building control has been presented: a linear state-space model combined with a nonlinear regressor, and a fully nonlinear architecture based on recurrent neural networks. Both architectures are capable of taking nonlinearities into account. They were used within a local optimization procedure to minimize the power exchanged between a building and the grid without degrading comfort.

Table 3 presents a qualitative comparison of the two models used in the study. None comes on top of the other on all criteria. The second architecture based on RNNs is more data intensive but also more accurate. In general, the higher representation capability of RNN architectures does not translate into strong enough control performance to justify the increased computational demand as

compared to MPC based on well-designed linear state-space models. The exception to this statement would be cases where constraint violations have disproportionate importance or for systems with more extreme nonlinearities than the typical building used in this study.

In future work, the sample efficiency of the RNN architecture could be improved using pre-training and transfer learning techniques for buildings that share the same inputs-outputs features. In cases where extreme nonlinearities strengthen the value of this architecture research is needed to alleviate some of the drawbacks of the recurrent architectures outlined in Table 3. For instance, as suggested in Maddalena et al. (2019), it could be possible to reduce the computing costs by changing the encoder-decoder architecture to directly predict the quadratic form to be minimized in Equation (8).

Through this systematic comparison in the representative example of a building, we were able to demonstrate that, in their current form, RNN architectures do provide improved accuracy on the modelling of nonlinear systems but have limited value in the optimal control of such systems. Well-designed linear state-space models with nonlinear regressors are the solution of choice in most cases for model-predictive control.

Note

1. Actual weather data was used in order to benchmark only the models and not errors from the predictions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no 731211, project SABINA.

ORCID

Baptiste Schubnel  <http://orcid.org/0000-0001-5029-6450>

References

- Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, et al. 2015. "TensorFlow: Large-scale Machine Learning on Heterogeneous Systems." Software available from tensorflow.org.
- Amos, B., L. Xu, and J. Z. Kolter. 2017. "Input Convex Neural Networks". Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 146–155.
- Bieker, K., S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz. 2020. "Deep Model Predictive Flow Control with Limited Sensor Data and Online Learning." *Theoretical and Computational Fluid Dynamics* 34: 577–591.
- Camacho, E. F., and C. Bordons. 2007. *Model Predictive Control*. London: Springer-Verlag.
- Chakraborty, D., and H. Elzarka. 2019. "Advanced Machine Learning Techniques for Building Performance Simulation: A Comparative Analysis." *Journal of Building Performance Simulation* 12 (2): 193–207.
- Chen, Y., Y. Shi, and B. Zhang. 2017. "Modeling and Optimization of Complex Building Energy Systems with Deep neural Networks." In *51st Asilomar Conference on Signals, Systems, and Computers, ACSSC 2017*, Pacific Grove, CA, USA, October 29–November 1, 2017, 1368–1373.
- Chen, Y., Y. Shi, and B. Zhang. 2019. "Optimal Control via Neural Networks: A Convex Approach." Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, USA.
- Chen, Y., Y. Shi, and B. Zhang. 2020. "Input Convex Neural Networks for Optimal Voltage Regulation." Preprint, arXiv:2002.08684.
- Chung, J., C. Gulcehre, K. Cho, and Y. Bengio. 2014, December. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." Proceedings of the NIPS 2014 Workshop on Deep Learning and Representation Learning, Montreal, Canada.
- Crawley, D. B., L. K. Lawrie, C. O. Pedersen, and F. C. Winkelmann. 2000. "Energy Plus: Energy Simulation Program." *ASHRAE Journal* 42 (4): 49–56.
- Drgoña, J., D. Picard, M. Kvasnica, and L. Helsen. 2018. "Approximate Model Predictive Building Control Via Machine Learning." *Applied Energy* 218 (23): 199–216.
- European Standard. 2011. "Heat Pumps with Electrically Driven Compressors-Testing and Requirement for Marking of Domestic Hot Water Units." *EN 16147*, January 2011.
- Fletcher, R. 1987. *Practical Methods of Optimization*. 2nd ed. New York: John Wiley & Sons.
- Funahashi, K., and Y. Nakamura. 1993. "Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks." *Neural Networks* 6 (6): 801–806.
- Gill, P. E., and E. Wong. 2012. "Sequential Quadratic Programming Methods." In *Mixed Integer Nonlinear Programming*, 147–224. New York, NY: Springer
- Gonzalez, J., and W. Yu. 2018. "Non-Linear System Modeling Using Lstm Neural Networks." *IFAC-PapersOnLine* 51 (13): 485–489. 2nd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2018.
- Gorecki, T. 2017. "Predictive Control methods for Building Control and Demand Response." PhD thesis, Ecole Polytechnique Fédérale de Lausanne.
- Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–1780.
- Lam, S. K., A. Pitrou, and S. Seibert. 2015. "Numba: A LLVM-Based Python Jit Compiler." Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, San Jose, CA, USA, 1–6.
- Lanzetti, N., Y. Z. Lian, A. Cortinovis, L. Dominguez, M. Mercangöz, and C. Jones. 2019, June. "Recurrent Neural Network Based MPC for Process Industries." 18th European Control Conference (ECC), Naples, Italy, 1005–1010.

- Ljung, L. 1999. "System Identification." *Wiley Encyclopedia of Electrical and Electronics Engineering*. Hoboken, NJ: Wiley.
- Maddalena, E. T., C. G. da S. Moraes, G. Waltrich, and C. N. Jones. 2019. "A Neural Network Architecture to Learn Explicit MPC Controllers from Data." Preprint, arXiv:1911.10789.
- Mayne, D. Q. 2014. "Model Predictive Control: Recent Developments and Future Promise." *Automatica* 50 (12): 2967–2986.
- Nakkiran, P., G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. 2020. "Deep Double Descent: Where Bigger Models and More Data Hurt." Proceedings of the International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia.
- Ogunmolu, O., X. Gu, S. Jiang, and N. Gans. 2016. "Nonlinear Systems Identification Using Deep Dynamic Neural Networks." In *Proceedings of the of the IEEE International Conference on Information and Automation (ICIA)*, August 2016, Yunnan, China, 8–10.
- Papadopoulos, S., E. Azar, Wei-Lee Woon, and C. E. Kontokosta. 2018. "Evaluation of Tree-Based Ensemble Learning Algorithms for Building Energy Performance Estimation." *Journal of Building Performance Simulation* 11 (3): 322–332.
- Péan, T. Q., J. Salom, and R. Costa-Castelló. 2019. "Review of Control Strategies for Improving the Energy Flexibility Provided by Heat Pump Systems in Buildings." *Journal of Process Control* 74: 35–49.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas. (Oct) 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–2830.
- Peng, H., K. Nakano, and H. Shioya. 2007. "Nonlinear Predictive Control Using Neural Nets-based Local Linearization Arx Model—Stability and Industrial Application." *IEEE Transactions on Control Systems Technology* 15 (1): 130–143.
- Schoukens, J., and L. Ljung. 2019. "Nonlinear System Identification: A User-Oriented Roadmap." *IEEE Control Systems Magazine* 39 (6): 28–99.
- Sourbron, M., C. Verhelst, and L. Helsen. 2013. "Building Models for Model Predictive Control of Office Buildings with Concrete Core Activation." *Journal of Building Performance Simulation* 6 (3): 175–198.
- Stein, J., W. Holmgren, J. Forbess, and C. W. Hansen. 2016. "Pvlib: Open Source Photovoltaic Performance Modeling Functions for Matlab and Python." Proceedings of the 2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC), Portland, USA, 3425–3430.
- Sturzenegger, D., D. Gyalistras, M. Morari, and R. S. Smith. 2016. "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.
- Taddeo, P., A. Colet, R. E. Carrillo, L. C. Canals, B. Schubnel, Y. Stauffer, I. Bellanco, C. C. Garcia, and J. Salom. January 2020. "Management and Activation of Energy Flexibility At Building and Market Level: A Residential Case Study." *Energies* 13 (5): 1188.
- Valenzuela, P. E., A. Ebadat, N. Everitt, and A. Parisio. 2020. "Closed-Loop Identification for Model Predictive Control of HVAC Systems: From Input Design to Controller Synthesis." *IEEE Transactions on Control Systems Technology* 28 (5): 1681–1695.
- Van Overschee, P., and B. L. De Moor. 2012. *Subspace Identification For Linear Systems: Theory–Implementation–Applications*. New York, NY: Springer Science & Business Media.

- Zhang, J., T. He, S. Sra, and A. Jadbabaie. 2019. "Analysis of Gradient Clipping and Adaptive Scaling with A Relaxed Smoothness Condition." Preprint, arXiv:1905.11881.

Appendices

Appendix 1. Controller architectures

A.1 Rule-based controllers

Rule-based controllers use fixed thermostat temperature setpoints for the rooms and a heating curve for the heat pump supply temperature. The heat pump temperature supply lies within 40–55°C; see Figure 1. The heating strategy follows typical heating strategies in buildings, where supply temperature varies as a function of the outside temperature, providing more heat at a lower temperature such as to ensure comfort bounds will not be violated. The heat pump supply and tank setpoints are the same for all models. A temperature of 40°C has been applied to the bottom temperature of the tanks to ensure that the heat pump supply temperature was always higher than the tank temperature. This prevents the heat pump from cooling down the tanks. The rule-based controller names indicate the fixed setpoints for the room thermostats: e.g. RB 19 is a rule-based controller with 19°C temperature setpoint for the room thermostats. The following setpoints are used in the rule-based controllers:

The hysteresis control around these setpoints is realized at a lower level. The low-level hysteresis control loop turns on the fan coil units in each room if the temperature is lower than the room temperature setpoint minus 0.5°C and it ensures the heat pump is on. The pump and coils stay on until all room temperature have reached their setpoints minus 0.1°C (to take inertia into account and avoid going above the desired temperature). A similar rule with a higher deadband (2°C) is applied to the tanks.

A.2 Linear state-space models

Linear state-space models have been fitted using N4SID method. The models were fitted on data from the simulation

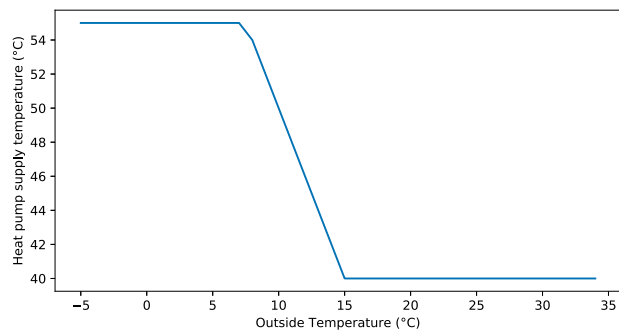


Figure A9. Heating curve for heat pump supply temperature.

Table A1. Rule-based controllers setpoints.

Controllers	Room sp. (°C)	Tank sp. (°C)
RB 19	19.0	40.0
RB 19.5	19.5	40.0
RB 20	20.0	40.0
RB 21	21.0	40.0

Table A2. Linear state-space models with nonlinear outputs: number of data points used for model identification, resolution of 15 min.

Models	Data points for training
LSS-NL1	4000
LSS-NL2	4000
LSS-NL3	2000
LSS-NL4	2000
LSS-NL5	750

Table A3. Encoder-decoder sizing parameters.

Layers	Hidden size
LSTM encoder	512
LSTM decoder	512
layer1	512
layer2	256
layer3	128

obtained with random multisine excitations (Schoukens and Ljung 2019). The number of data points were varying between the models, as well as the excitations and building response.

The linear state-space models have eight hidden degrees of freedom i.e. x in Equation (1) belongs to \mathbb{R}^8 . Power energy consumption has been estimated with kernel ridge regression with rbf kernel with parameters $\gamma = 0.1$ and $\alpha = 1.0$ (see Pedregosa et al. (2011)).

A.3 Encoder-decoder models

Encoder-decoder models use LSTM and multi-layer perceptrons as described on Figure 3. All models have been identified with the same dataset but with distinct numbers of training iterations. Architecture parameters and number of training iterations per model instances are given in Tables A3 and A4. For training, the following strategies were used:

- Batch size of 200.
- Adam optimizer with default tensorflow Adam parameters and learning rate of 1.10^{-4} .
- Decoding length randomly chosen at each iteration in the set {2, 4, 6, 8, 10, 16, 24, 32, 64, 88, 144}.
- Gradient clipping with norm 1 (see Zhang et al. (2019) for an interesting analysis of gradient clipping and convergence speed up).

Table A4. Encoder-decoder training iterations per model instances (batch size of 200 sequences per iteration).

Models	Training iterations
ENC-DEC1	8×10^4
ENC-DEC2	8×10^4
ENC-DEC3	6×10^4
ENC-DEC4	3×10^4
ENC-DEC5	6×10^4

Appendix 2. Error models summary

Models accuracy metrics on the evaluation set are given below for one hour ahead and 1 day ahead forecasts.

Table A5. Error metrics for sliding windows of 1-day-ahead forecast, averaged over 4 months.

Models	MAE T	MAE P	sMRAE T	sMRAE P
LSS-NL1	0.97	3.57	0.047	0.55
LSS-NL2	0.73	3.24	0.035	0.51
LSS-NL3	0.77	3.52	0.037	0.54
LSS-NL4	0.81	3.43	0.039	0.54
LSS-NL5	0.71	4.33	0.034	0.63
ENC-DEC1	0.29	2.25	0.014	0.46
ENC-DEC2	0.26	1.91	0.012	0.36
ENC-DEC3	0.28	1.93	0.013	0.38
ENC-DEC4	0.39	2.50	0.019	0.56
ENC-DEC5	0.36	2.04	0.017	0.39

Values corresponding to the best-performing model are in bold.

Table A6. Error metrics for sliding windows of 1-h-ahead, averaged over 4 months.

Models	MAE T	MAE P	sMRAE T	sMRAE P
LSS-NL1	0.64	3.43	0.030	0.54
LSS-NL2	0.67	3.24	0.031	0.51
LSS-NL3	0.66	3.55	0.031	0.55
LSS-NL4	0.77	3.41	0.037	0.54
LSS-NL5	0.62	4.32	0.029	0.63
ENC-DEC1	0.22	2.05	0.010	0.40
ENC-DEC2	0.19	1.90	0.009	0.37
ENC-DEC3	0.19	1.93	0.009	0.38
ENC-DEC4	0.24	2.50	0.011	0.54
ENC-DEC5	0.20	1.91	0.009	0.37

Values corresponding to the best-performing model are in bold.