BS 2023 4-6 SEPT SHANGHAI
International Building Performance Simulation Association

Combined reinforcement learning (RL) and model predictive control (MPC) for optimal building energy use

Marius Bagle[1,*] and Francesco Goia[2]

[1] *Norwegian University of Science and Technology, Trondheim, Norway*
[1] *SINTEF Community, Oslo, Norway*
[*] *Corresponding author: Marius Bagle, mariuseb@stud.ntnu.no*

Abstract

A well-known control strategy from the process industry, model predictive control (MPC), is regarded as a promising avenue towards achieving smart operation of building HVAC systems, by enabling energy flexibility through the thermal mass of the building. Given sufficient models of the building envelope and HVAC-system, alongside predictions of future disturbances, it is possible to optimize the building control in a receding horizon fashion, thus achieving near-optimal control. However, a significant effort is required to deliver sufficient control models for building MPC. To this end, we will apply moving horizon estimation (MHE), in combination with reinforcement learning (RL), to deliver closed-loop state and parameter estimation, relaxing the need for exhaustive system identification (SID) prior to control deployment.

Highlights

- Automated, closed-loop, moving horizon system identification for building MPC
- MPC as function approximator in reinforcement learning scheme
- Reinforcement learning of uncertain policy parameters subject to large uncertainties (e.g. model parameter adaption inertia, noise covariances)

Introduction

MPC is a well-established control strategy for constrained optimization, which enables energy flexibility by exploiting energy storage capabilities and optimization of renewable on-site generation. In addition, incorporation of forecast models (i.e., internal gains, weather), and user inputs (e.g. comfort ranges, electric vehicle charging needs) allows anticipation of energy needs that can be optimized for flexible energy sources (Serale et al. (2018)).
MPC has been successfully demonstrated on building systems in several previous works (Drgoňa et al. (2020)). However, control deployment is usually preceded by a time-consuming SID-phase, which increases the threshold for widespread building MPC implementation. In Rockett and Hathway (2017), it is argued that the most important feature of a building controller model is its ability to predict state trajectories within the control horizon, and that it need not accurately reflect year-round behavior of the building. In this context, we introduce MHE, an optimization-based state estimation strategy, where the current system state is inferred based on a finite sequence of past measurements, considered to be the natural counterpart to MPC (Rawlings et al. (2017)). In addition, MHE can be extended to deliver parameter estimates (Kühl et al. (2011)). As MHE increases the computational burden compared to traditional eanstimation techniques, it is generally considered to be more suitable for systems with slower dynamics and greater computational resources, which fits well with building systems (Drgoňa et al. (2020)).

Reinforcement learning (RL) is a branch of machine learning, concerned with how intelligent agents should take actions in an environment, in order to maximize the notion of cumulative reward (Sutton and Barto (2020)). For each state, the optimal action is computed either directly as a policy, or indirectly as through value-based methods (Zanon and Gros (2021)). Recent achievements of RL in general include the large language model chatGPT (OpenAI (2023)), as well as achieving superhuman performance in games such as Go (Silver et al. (2016)) and Dota (OpenAI et al. (2019)). In the context of control, one is often concerned with constraint satisfaction, which RL typically does not guarantee. This is especially the case when using deep neural networks (DNNs) as function approximators, which is typically the case in state of the art applications (Zanon and Gros (2021)). In addition, there is the issue of sampling efficiency: it may take an inordinate amount of interactions with the environment to arrive at RL policies capable of achievements such as those mentioned. In the case of buildings, such pre-training must also take place in simulated environments. A fairly recent example from the building literature, highlighting the issue, is Zhang and Lam (2018), where an RL-agent is trained for 47.5 years in simulation in order achieve the tracking performance

https://doi.org/10.26868/25222708.2023.1578

of a P-controller.
The novelty of this work is two-fold: (i) first, it investigates the potential of automated system identification through MHE for building control problems under realistic conditions, i.e. under significant plant-model mismatch. MHE is investigated in the same terms for a simple 3R3C controller model-emulator setup (i.e. no structural model mismatch) in Maree et al. (2021), where parameters are found to converge to their true values. In the present work, we instead emulate the real process with a white-box building simulation model, while keeping the grey-box controller model structure.
(ii) It introduces the notion of combined RL and MPC by viewing the MPC optimization problem as a differentiable implicit function in a building context. Attempts at combining RL and MPC for building applications has been made in e.g. Arroyo et al. (2022), where a value function is pre-trained on a simplified simulation environment, and the optimal policy is defined as a superposition of the one-step solution to the MPC problem and the pre-trained value function, which is adapted on-line during control deployment. Hence, the MPC and RL policy parameterizations are kept separate. In Chen et al. (2020), a differentiable MPC (black-box) policy in combination with imitation, end-to-end learning is used, requiring extensive pre-training of agents. In this work, on the other hand, the learning is done in a purely online fashion, i.e. value function parameter updates are delivered during control of the building.

Methods

In the following, the methodology is presented. In essence, the methodology boils down to the formulation of three distinct optimization problems, in addition to a learning scheme (delivered through RL); one to be solved once in the "traditional" MPC approach (system identification), and two to be solved along a receding horizon: one looking forward in time (MPC), and one looking backward in time (MHE). A brief description of the building emulator will also be given. The order of presentation is then: (i) building emulator (ii) system identification problem (iii) model predictive control formulation (iv) moving horizon estimation formulation (v) MHE-MPC as function approximator in RL (vi) NLP-sensitivities/KKT.

Building emulator

For the purpose of representing the building, i.e. the map from control action a to the next state s_+, we formally define our control task as a discrete Markov decision process (MDP) with the state transition dynamics:

$$\mathbb{P}[x_+|x,u] \tag{1}$$

where x, u is some state-action pair, and x_+ the state after applying action u (Gros and Zanon, 2020). The virtual control bench-marking framework BOPTEST (Blum et al., 2021) is used as a proxy for the real system (1). The framework is extended by implementing a one-zone model of the ZEB Living Lab (Goia et al., 2015), based on components from the Buildings library (Wetter et al., 2014). The envelope model is based on the shoe-box geometry of the building, and has not yet been validated. The HVAC-system and control system consists of an on-off controller, modulating an electric heater. The heat flow of the heater is split into 70 % concective and 30 % radiative heat gains, as per standard assumptions Strachan et al. (2016). To ensure that equation 1 holds, ideal control of the heater is assumed, i.e. the baseline control can be overwritten by providing the exact heat flow as desired from the calculation of the MPC block. Thus, when running the MPC, the on-off controller is bypassed. Some simplifications have been made to the emulator model. There is no (i) forecast uncertainty, and no (ii) ventilation.

System identification problem

In the following, a technique for obtaining a reduced-order model suitable for controlling the system (1), represented by the proxy model shown in figure 1. It is assumed that a suitable model structure, i.e. a structure that allows for sufficient accuracy without over-fitting, is at hand. From Yu et al. (2019), one obtains that a 2R2C model structure is sufficient to represent one thermal zone of a building envelope. This is in contrast to the approach in e.g. Bacher and Madsen (2011), where model structures are iterated on in increasing order of complexity with a maximum likelihood based approach, until a p-value larger than 0.05 is reached. A reduced-order system formulation suitable for system identification is:

$$x_{k+1} = f(x_k, u_k, p) + w_k \tag{2a}$$
$$y_k = h(x_k, p) + v_k \tag{2b}$$

with $x_k \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$, $u_k \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$. Furthermore, Gaussian white-noise is assumed for the additive disturbances w_k, v_k, i.e. $v_k \sim \mathcal{N}(0, R)$, $w_k \sim \mathcal{N}(0, Q)$. The parameter estimation problem to be solved is

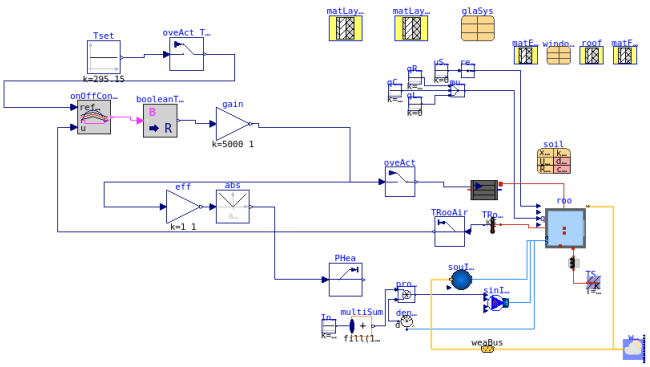

Figure 1: Building emulator.

https://doi.org/10.26868/25222708.2023.1578

then:

$$\min_{\mathbf{p},\mathbf{v},\mathbf{w}} V_N(\mathbf{u}_N, \mathbf{y}_N) \tag{3a}$$
$$x_{k+1} = f(x_k, u_k, p) + w_k, \quad \forall k \in \mathbb{I}_{1:N-} \tag{3b}$$
$$y_k = h(x_k, p) + v_k, \quad \forall k \in \mathbb{I}_{1:N} \tag{3c}$$
$$x_k \in \mathbb{X}, \quad \forall k \in \mathbb{I}_{1:N-} \tag{3d}$$
$$\mathbf{p} \in \mathbb{P} \tag{3e}$$

where $\mathbf{u}_N = [u_1, ..., u_N]$ defines both controllable inputs and disturbances (i.e. heating, solar irradiation, ambient temperature etc.) and $\mathbf{y}_N = [y_1, ..., y_N]$ is a sequence of measurements. The parameter estimation value function V_N is defined as a sum of stage costs over the estimation horizon N:

$$V_N(\mathbf{u}_N, \mathbf{y}_N) := \sum_{k=1}^{N} l(v_j, w_j) \tag{4}$$

where the stage cost on each time step is defined as follows:

$$l(w_j, v_j) = \begin{cases} \|v_j\|^2_{R_\theta^{-1}} + \|w_j\|^2_{Q_\theta^{-1}}, & \forall j \in \mathbb{I}_{0:N-2} \\ \|v_j\|^2_{R_\theta^{-1}}, & j = N-1 \end{cases} \tag{5}$$

where covariance matrices Q_θ, R_θ are considered fixed and known a priori. The objective then is to minimize the prediction error over the measurement history, subject to these covariance matrices.

Model predictive control

We will now describe the model predictive control strategy, which is used to control the emulator of the real system (1) in real-time, assuming we have obtained the reduced-order model (2). First, we define $N := k + K - 1$, with K the horizon length. Using the assumption of zero-order hold (Rawlings et al. (2017)), the MPC problem is to be solved for the control vector $\mathbf{u}_{k:N} = [u_k, \ldots, u_N]$. The optimization problem to be solved over a receding horizon is then:

$$\min_{\mathbf{u}} V_N(\mathbf{d}_{k:N}) \tag{6a}$$
$$x_{j+1} = f(x_j, u_j, d_j), \quad \forall j \in \mathbb{I}_{k:N} \tag{6b}$$
$$u_j \in \mathbb{U}, \quad \forall j \in \mathbb{I}_{k:N} \tag{6c}$$
$$x_j \in \mathbb{X}, \quad \forall j \in \mathbb{I}_{k:N} \tag{6d}$$
$$x_k = \hat{x}_{t_k} \tag{6e}$$

with the estimation of the state $\hat{x}_{t_k}$ provided by some state estimation scheme at time $t = t_k$, e.g. a Kalman filter or a moving horizon estimator. In the work presented here, a simple minimum energy formulation will be used. The corresponding value function for this formulation is:

$$V_N(\mathbf{d}_{k:N}) = \sum_{j=k}^{N} l(u_j) \tag{7}$$

with stage cost:

$$l(u_j) = u_j^2, \quad \forall j \in \mathbb{I}_{k:N} \tag{8}$$

The implicit MPC control law is defined as taking the first optimal control move, i.e.:

$$\pi(x_k) := u_k^*. \tag{9}$$

Moving horizon estimation

The reduced model (2) cannot completely capture the dynamics of the real system (1), since the true model structure is unknown (Hjalmarsson (2009)). A strategy to accommodate this structural uncertainty is to allow a subset of the parameters p, i.e. $\overline{p} \subseteq \mathbb{P}$, in addition to the state trajectory $\mathbf{x}$ to be updated as new data arrives from closed-loop operation. This is known is dual state-parameter moving horizon estimation (MHE) (Kühl et al. (2011)). Analogous to the MPC case, we define a horizon into the past of M time steps, and let $L := k - M + 1$. Assuming $L > 0$, which implies $\mathbf{u}^*_{L:k} = \boldsymbol{\pi}_{L:k} = [\pi_L, \ldots, \pi_k]$, $\mathbf{d}_{L:k}$, $\mathbf{y}_{L:k}$ available for the estimation problem. We make the following adaptions to the system (2) (parameter estimation DAE):

$$x_{j+1} = f(x_j, \pi_j, p) + w_j, \quad \forall j \in \mathbb{I}_{L:k-1} \tag{10a}$$
$$y_j = h(x_j, p) + v_j, \quad \forall j \in \mathbb{I}_{L:k} \tag{10b}$$

to emphasize the dependence of previous optimal control actions $\mathbf{u}^*_{L:k} = \boldsymbol{\pi}_{L:k}$. The MHE value function is defined as a least-squares stage cost:

$$V_M(\boldsymbol{\pi}_{L:k}, \mathbf{y}_{L:k}, \mathbf{d}_{L:k}) = \gamma^{M-1} V_L(\hat{x}_L, \hat{p}_L) + \sum_{j=L-1}^{k} \gamma^j l(x_j, w_j, v_j) \tag{11}$$

with γ a discount factor $\in [0, 1)$ introduced to give a higher relevance to recent data, and the stage cost $l(w_j, v_j)$ given by equation (5). The term $V_L(\hat{x}_L, \hat{p}_L)$ is known as the *arrival cost* (Rawlings et al., 2017), which is needed to avoid rendering the estimation problem computationally intractable. It can be seen that it represents an inertia for departing from previous estimations $\hat{x}_L$, $\hat{p}_L$.

One common choice of parameterization for this term is a quadratic approximation Kühl et al. (2011), based on the covariance of parameter and state estimates:

$$V_L(\hat{x}_L, \hat{p}_L) = \left\| z_L - \overline{z}_L \right\|_{P_\theta^{-1}} \tag{12}$$

with the *costate* z defined as:

$$z = \begin{bmatrix} p_k \\ x_L \end{bmatrix} \tag{13}$$

and $\overline{z}$ the a-priori most likely values of z, which we take to be the result from the estimation at $t = t_{k-1}$, i.e.:

$$\overline{z}_L = \begin{bmatrix} p_{k-1} \\ x_{L|k-1} \end{bmatrix} \tag{14}$$

and finally: P_θ a positive semi-definite matrix, whose inverse represents the confidence associated with our previous estimate $\bar{z}$ (that is, both of the parameters and states). With the value function defined, we describe the MHE problem of interest as:

$$\min_{\mathbf{w},\mathbf{v},\mathbf{p}} V_M(\boldsymbol{\pi}_{L:k}, \mathbf{y}_{L:k}, \mathbf{d}_{L:k}) \tag{15a}$$

s.t.

eq. (10)

$$p_j \in \mathbb{P}, \qquad \forall j \in \mathbb{I}_{L:k-1} \tag{15b}$$

$$w_j \in \mathbb{W}, \qquad \forall j \in \mathbb{I}_{L:k-1} \tag{15c}$$

$$v_j \in \mathbb{V}, \qquad \forall j \in \mathbb{I}_{L:k} \tag{15d}$$

with all entities previously defined. MHE defined in this manner reduces to solving the problem (3) on a receding horizon, with the observed closed-loop response when applying the control law (9).

Controller model structure

For the purpose of solving problems given by eqs. (3), some a priori defined, grey-box type model structures need to provided to the parameter estimation algorithms. For the simulation studies in this work, we provide the model structure given by figure 2.

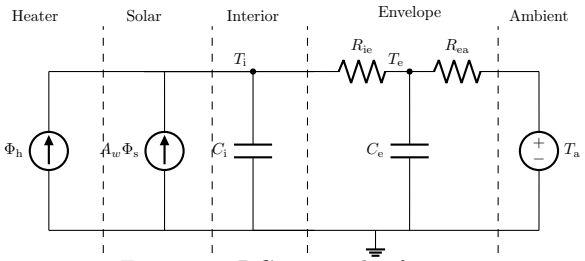

Figure 2: RC-network of T_iT_e.

.

The system of ordinary differential equations (ODE's) for this system can be written out by considering the energy balances of the nodes:

$$\frac{dT_\mathrm{i}}{dt} = \frac{1}{R_\mathrm{ie}C_\mathrm{i}}(T_\mathrm{e} - T_\mathrm{i}) + \frac{A_\mathrm{w}}{C_\mathrm{i}}\phi_\mathrm{s} + \frac{1}{C_\mathrm{i}}\phi_\mathrm{h} \tag{16a}$$

$$\frac{dT_\mathrm{e}}{dt} = \frac{1}{R_\mathrm{ie}C_\mathrm{e}}(T_\mathrm{i} - T_\mathrm{e}) + \frac{1}{R_\mathrm{ie}C_\mathrm{e}}(T_\mathrm{a} - T_\mathrm{e}) \tag{16b}$$

with all entities of the model having a natural physical representation.

Reinforcement learning with MPC/MHE

It is well known that the assumption of normally distributed noise may not hold in non-ideal cases, where factors such as model mismatch (parametric) and unmodelled dynamics (structural) may lead to difficulty in tuning covariance matrices R_θ, Q_θ, which determine the weighting between model and measurement. Furthermore, the problem of both parameterization of the arrival cost in general, and optimally updating the particular parameterization represented by the weighting P_θ, remains an open problem in the realm of cybernetics (Rawlings et al. (2017)). To handle these uncertain parameters, we turn to recent efforts made to combine the fields of model predictive control and reinforcement learning, leveraging strengths of both fields Gros and Zanon (2020). From classical reinforcement learning theory, one obtains the Bellman equations, Sutton and Barto (2020):

$$\pi_\theta(s) = \arg\min_a Q_\theta(s,a) \tag{17}$$

$$V_\theta(s) = \min_a Q_\theta(s,a) \tag{18}$$

describing respectively the action a to be taken in a given state s from the minimization of the action-value function $Q_\theta(s,a)$, and the value of being in a given state s through the value function $V_\theta(s)$. Following the theory presented in Gros and Zanon (2020), we note that an RL policy is implicitly given by the the solution to the optimization problems described by eqs. (6) and (15), i.e.:

$$Q_\theta(s,a) = \min_{\mathbf{u},\mathbf{x}} \quad \text{(6a)} \tag{19a}$$

$$\text{s.t.} \quad \text{(6b)} - \text{(6e)} \tag{19b}$$

$$a = u_k, s = x_k \tag{19c}$$

with the implicit function given by MHE-MPC scheme acting as a *function approximator*, a role traditionally played by linear functions and neural networks in the context of reinforcement learning (Sutton and Barto (2020)). Taking this view, we can shape the values of the uncertain parameters Q_θ, R_θ, P_θ through interactions with the environment/building, while maintaining reasonable confidence in performance of the control policy.

Let us consider a TD(0)-algorithm, SARSA, as a candidate to deliver on-line adjustments of uncertain parameters θ. The SARSA algorithm with the notation defined here is defined as follows ((Sutton and Barto, 2020)):

$$\theta_{k+1} = \theta_k + \alpha\delta_k \nabla_\theta Q_\theta(x_k, u_k) \tag{20}$$

$$\delta_k = R_{k+1} + \gamma Q_\theta(x_{k+1}, u_{k+1}) - Q_\theta(x_k, u_k) \tag{21}$$

with the temporal difference driven by the $(1-\epsilon)$-greedy update, instead of the predicted action-value of the optimal move (on-policy), in contrast to Q-learning approaches. In our implementation, we let ϵ decay over time by the update rule:

$$\epsilon_{k+1} = 0.99\epsilon_k \tag{22}$$

to gradually encourage exploitation in favour of exploration as more information of environment is built up. To obtain the gradient $\nabla_\theta Q_\theta(x_k, u_k)$, one needs to differentiate *through* the policy $Q_\theta(x_k, u_k)$ by use of the chain rule. Since the MHE scheme indirectly acts on the Q_θ through estimation of x_k, p_k, and those in turn are functions of θ, i.e.:

$$(x_k, p_k) = \arg\min_{x_k, p_k} V_{M,k}(\theta) \tag{23}$$

where the subscript k is added to emphasize the solution of the MHE problem in timestep $t = t_k$. the total derivative of the policy reads as:

$$\frac{dQ_\theta}{d\theta} = \frac{\partial Q_\theta}{\partial x_k}\frac{\partial \hat{x}_k}{\partial \theta} + \frac{\partial Q_\theta}{\partial p_k}\frac{\partial \hat{p}_k}{\partial \theta} \tag{24}$$

since we consider a policy parameterization where parameters θ only appear in the MHE problem.

NLP sensitivities

To obtain the sensitivities above, needed for constructing the gradient in the TD(0)-scheme, we first define a general parametric non-linear program as:

$$\min_{\mathbf{w}} \quad \phi(\mathbf{w}) \tag{25a}$$
$$\text{s.t.}$$
$$\mathbf{g}(\mathbf{w}, \theta) = 0 \tag{25b}$$
$$\mathbf{h}(\mathbf{w}, \theta) \leq 0 \tag{25c}$$

where $\phi(\cdot) \in \mathbb{R}$ is the objective function, $\mathbf{h}$ and $\mathbf{g}$ yield the inequality and equality constraints, respectively, and the set of variables $\mathbf{w}$ are labelled the decision variables. Assuming the problems (6) and (15) have been discretized to yield problems on the form (25), we can define the Lagrangians of the respective optimization problems as $\mathcal{L}$ and $\hat{\mathcal{L}}$, respectively, as follows:

$$\mathcal{L} = \Phi(\tilde{\mathbf{w}}) + \tilde{\lambda}^T \tilde{\mathbf{g}}(\tilde{\mathbf{w}}) + \tilde{\mu}_{\mathbb{A}}^T \tilde{\mathbf{h}}_{\mathbb{A}}(\tilde{\mathbf{w}}) \tag{26}$$
$$\hat{\mathcal{L}} = \hat{\Phi}(\hat{\mathbf{w}}) + \hat{\lambda}^T \hat{\mathbf{g}}(\hat{\mathbf{w}}) + \mu_{\mathbb{A}}^T \hat{\mathbf{h}}_{\mathbb{A}}(\hat{\mathbf{w}}) \tag{27}$$

where $\mathbb{A}$ denotes the active set at any optimal NLP solution. Let us now define the primal-dual pairs of solutions to the MPC and MHE problems of interest be denoted as $\tilde{\mathbf{z}} = \{\tilde{\mathbf{w}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}\}$ and $\hat{\mathbf{z}} = \{\hat{\mathbf{w}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}}'\}$. Sensitivities are obtained by:

$$\frac{\partial Q_\theta}{\partial \theta} = \frac{\partial \mathcal{L}(x_k, p_k, \tilde{\mathbf{z}}^*)}{\partial \theta} \tag{28}$$
$$\frac{\partial Q_\theta}{\partial x_k} = \frac{\partial \mathcal{L}(x_k, p_k, \tilde{\mathbf{z}}^*)}{\partial x_k}, \tag{29}$$
$$\frac{\partial Q_\theta}{\partial p_k} = \frac{\partial \mathcal{L}(x_k, p_k, \tilde{\mathbf{z}}^*)}{\partial p_k}, \tag{30}$$

where $\tilde{\mathbf{z}}^*$ denotes the optimal solution to problem (6). To obtain the rest of the terms, we can use (with linear indepence constraint qualification, LICQ, and second-order sufficient conditions, SOSC) the implicit function theorem to obtain (Büskens and Maurer (2001)):

$$\frac{\partial \hat{\mathbf{z}}^*}{\partial \theta} = -\left(\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{z}}}\right)^{-1} \frac{\partial \hat{\mathbf{R}}}{\partial \theta} \tag{31}$$

where

$$\hat{\mathbf{R}}(\hat{\mathbf{z}}, \theta) = \begin{bmatrix} \nabla_{\hat{\mathbf{w}}} \hat{\mathcal{L}}(\hat{\mathbf{w}}, \theta) \\ \hat{\mathbf{g}}(\hat{\mathbf{w}}, \theta) \\ \hat{\mathbf{h}}_{\mathbb{A}}(\hat{\mathbf{w}}, \theta) \end{bmatrix} \tag{32}$$

so one ends up needing to obtain the KKT-matrix:

$$\left(\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{z}}}\right)^{-1} = \begin{bmatrix} \nabla_{\hat{\mathbf{w}},\hat{\mathbf{w}}} \hat{\mathcal{L}} & \nabla_{\hat{\mathbf{w}}} \hat{\mathbf{g}} & \nabla_{\hat{\mathbf{w}}} \hat{\mathbf{h}}_{\mathbb{A}} \\ \nabla_{\hat{\mathbf{w}}} \hat{\mathbf{g}}^T & 0 & 0 \\ \nabla_{\hat{\mathbf{w}}} \hat{\mathbf{h}}_{\mathbb{A}}^T & 0 & 0 \end{bmatrix}^{-1} \tag{33}$$

which can either be obtained directly from the solver (at the optimal solution of the NLP in question, in this case arising from the MHE problem), or calculated fairly cheaply (compared to NLP-solutions) in post-processing. Note that one also needs to obtain:

$$\frac{\partial \hat{\mathbf{R}}(\hat{\mathbf{z}}, \theta)}{\partial \theta} = \begin{bmatrix} \nabla_{\hat{\mathbf{w}},\theta} \hat{\mathcal{L}}_\theta \\ \nabla_\theta \hat{\mathbf{g}} \\ \nabla_\theta \hat{\mathbf{h}}_{\mathbb{A}} \end{bmatrix} \tag{34}$$

As x_k, p_k are both part of $\hat{\mathbf{w}}$, $\frac{\partial \hat{x}_k}{\partial \theta}$, $\frac{\partial \hat{p}_k}{\partial \theta}$ can be obtained from the expression given by equation (31).
Using the computations described above, we can summarize the algorithm with the pseudo-code given by Algorithm 1. We have assumed that the number of time steps have evolved past the MHE horizon length, i.e. $\overline{k} \geq M - 1$ and $\overline{k} = k + M$, such that the measurement window is filled up.

Algorithm 1 MHE+RL MPC

Require: α, θ_0, x_0, p_0
while True **do**
- $p_k, x_k \leftarrow$ Solve (15) ▷ MHE solve
- $\frac{\partial \hat{x}_k}{\partial \theta}, \frac{\partial \hat{p}_k}{\partial \theta} \leftarrow$ Solve (31) ▷ MHE sensitivity
- $\pi_\theta(x_k), V_\theta(x_k) \leftarrow$ Solve (6) ▷ MPC solve
- Apply $\pi_\theta(x_k) = u_0^*$ to real plant (1)
- Measure y_k, R_{k+1} from real plant (1)
- $\frac{\partial Q_\theta}{\partial x_k} \leftarrow$ Solve (29)
- $\frac{\partial Q_\theta}{\partial p_k} \leftarrow$ Solve (30)
- $\frac{\partial Q_\theta}{\partial \theta} \leftarrow$ Solve (28)
- $\frac{dQ_\theta}{d\theta} \leftarrow$ Assemble from (24) ▷ MPC sensitivities
- $\pi_\theta(x_{k+1}), V_\theta(x_{k+1}) \leftarrow$ Solve (6) ▷ Resolve MPC
- $\Delta\theta \leftarrow$ Solve (21)
- $\theta_{k+1} \leftarrow \theta_k + \Delta\theta$ ▷ RL update
- $k \leftarrow k + 1$

end while

Results and Discussion

To validate the proposed framework, three numerical experiments have been set up. All three use the emulator described in the methods section (depicted in figure 1) as a representation for the real building, and the controller model structure depicted in figure 2. All cases also use the same MPC formulation, given by equations (6)-(9), with a minimum energy formulation. The difference between the cases lies in how the parameters of the controller model are identified. In the baseline case / experiment 1, we run an SID-phase prior to control deployment, with a PRBS-signal providing excitation of the building

Table 1: Experiment description

Name	**SID**	**Observer**	**RL**
Exp. 1	y	KF	n
Exp. 2	n	MHE	n
Exp. 3	n	MHE	y

thermal mass according to the guidelines in Madsen et al. (2015). The PRBS-signal is run for 6 days. The observer (state estimator) used in this case is a classical Kalman filter.

In experiment 2, no SID-phase is run prior to control deployment, and MHE without arrival cost is used for receding horizon system identification and state estimation. In experiment 3, the learning scheme described in the last two subsections of the Methods section is deployed to update uncertain parameters of the MHE scheme, i.e. the arrival cost and noise covariances.

Table 2: Parameters found by SID (prior to Exp. 1)

Name	**Value**	**Unit**
R_{ie}	2.00	$[\frac{K}{kW}]$
R_{ea}	12.27	$[\frac{K}{kW}]$
C_i	0.41	$[\frac{kWh}{K}]$
C_e	2.05	$[\frac{kWh}{K}]$
A_i	4.40	m^2

Figure 3 shows the closed-loop result of experiment 1. Since an identification with a PRBS-signal has been carried out prior to running the experiment, and it is known that the building dynamics can be captured by the reduced-order model structure given in 2, we expect the result to be satisfactory, with few constraint violations, and little unnecessary energy use. By visual inspection of figure 3, we see that this is indeed the case (calculation of KPI's based on Blum et al. (2021) will feature in further work).

Figure 4 shows the result of applying MHE as a means of obtaining closed-loop, automatic system identification. The parameters settle around the values found in the SID-phase, which are shown in table 2. However, some parameters, like C_i and R_{ia} exhibit "nervous" behaviour, jumping up and down within the bounds set on the parameters by the MHE problem. This is not desirable, as we do not necessarily want to abandon the parameter estimates we have built up over time in favour of more recent data immediately. Due to space concerns, the closed-loop result of experiment 2 is not shown here.

Figure 6 shows the parameter evolution of experiment 3, which is MPC+MHE+RL. Here, the weighting parameters, as well as the covariance matrices of the closed-loop identification problem are found by RL updates in real-time. We see immediately that the parameter estimates are more stable, and hence more in line with the result from experiment 1. As for the closed-loop result, it can be seen by visual inspection of figure 5 that towards the end of the week, the result resembles that of experiment 1, with minor constraint violations. As the parameters of the physical model move in the direction of those obtained by SID prior to experiment 1, the trajectory of the temperature in experiment 3 will move towards the trajectory obtained in experiment 1. The constraint violation from ca. 0700-1000 on the fourth day, which is the most significant one occurring after the first three days have passed, is also present in experiment 1. Thus, it is a result of imperfect knowledge of the disturbances and/or plant-model mismatch, and not the proposed approach.

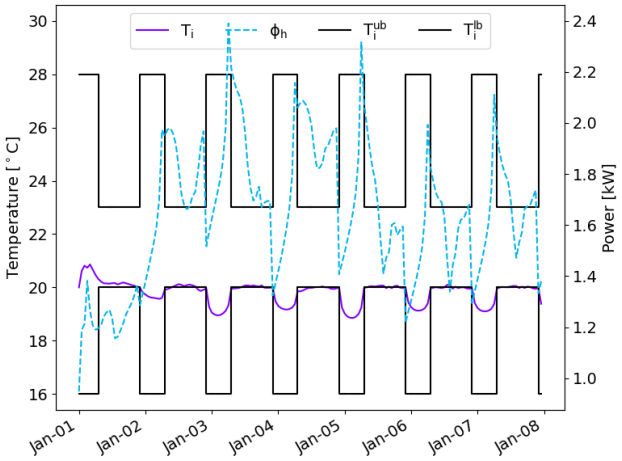

Figure 3: Closed-loop control Exp. 1

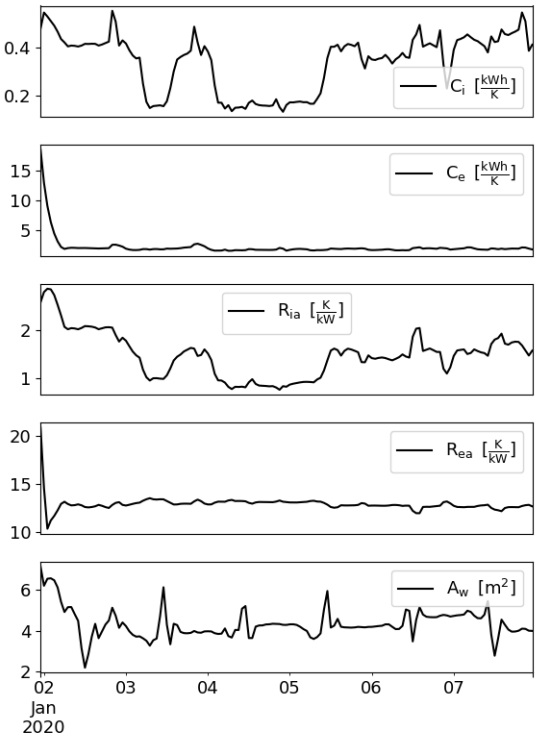

Figure 4: Parameter updating Exp. 2

https://doi.org/10.26868/25222708.2023.1578

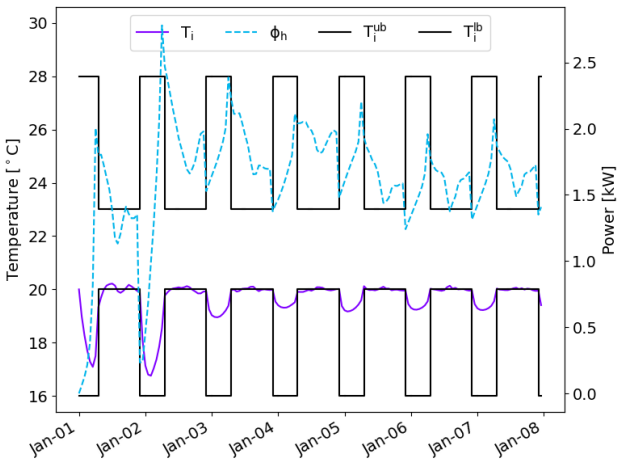

Figure 5: Closed-loop control Exp. 3

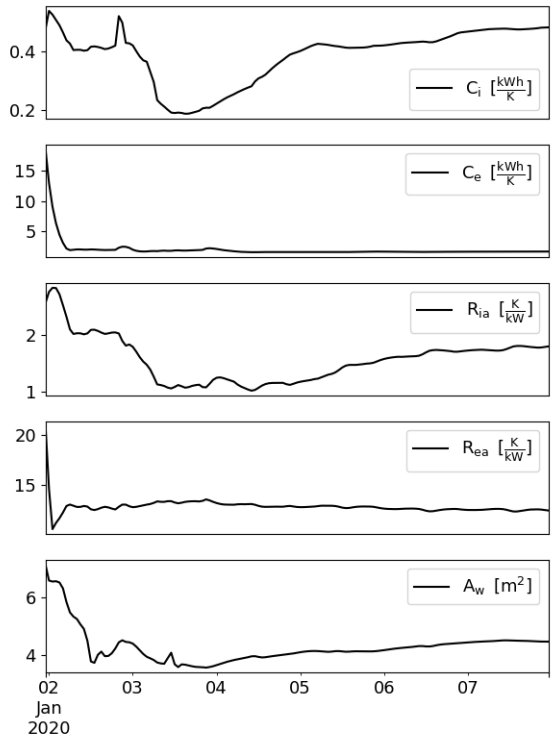

Figure 6: Parameter updating Exp. 3

Conclusion

A procedure for automatic, closed-loop system identification, based on moving horizon estimation, has been presented. The parameters of the grey-box controller model are shown to move towards the same numerical values as those yielded by a more traditional, PRBS-based approach, which puts stricter requirements on building occupancy during the duration of the identification experiment. To alleviate issues of stability in the closed-loop parameter estimates, a reinforcement learning scheme has been applied on top of the MHE-MPC scheme, using the reward provided by the environment (building) and the sensitivities at the optimal solutions of the NLPs arising from transcription of the optimal control problems to update uncertain parameters in real-time. The results are promising, showing that one obtains equivalent closed-loop results after approximately three days of operation. Thus, user discomfort and unnecessary energy usage to identify the controller model is avoided.

Further work

Further work can take many directions. First of all, a natural extension of the work presented here is to investigate the performance of the RL scheme for longer horizons, to see if the seasonal adaptations of the grey-box controller model are obtained by the MHE+RL scheme. This goes hand in hand with investigating and improving the numerical robustness of the sensitivity-based RL scheme. Furthermore, it is of interest to include richer parameterizations (i.e. extra terms) of the MPC scheme, and investigate the potential of RL to learn these terms and accommodate for more unmodelled dynamics, such as e.g. ventilation. This will necessitate using emulators with more complex HVAC- and control systems, and also providing baseline controllers with more detailed representation of the HVAC-systems. Another direction is the application of other RL algorithms, such as TD(N) and policy gradient approaches. Finally, an investigation of the approach in multi-zone settings is also of interest.

References

Arroyo, J., C. Manna, F. Spiessens, and L. Helsen (2022, March). Reinforced model predictive control (RL-MPC) for building energy management. *Applied Energy 309*, 118346.

Bacher, P. and H. Madsen (2011, July). Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings 43*, 1511–1522.

Blum, D., J. Arroyo, S. Huang, J. Drgoňa, F. Jorissen, H. T. Walnum, Y. Chen, K. Benne, D. Vrabie, M. Wetter, and L. Helsen (2021). Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings. *586-610*. Accepted: 2021-11-15T09:53:06Z Publisher: Taylor & Francis.

Büskens, C. and H. Maurer (2001). Sensitivity Analysis and Real-Time Control of Parametric Optimal Control Problems Using Nonlinear Programming Methods. In M. Grötschel, S. O. Krumke, and J. Rambau (Eds), *Online Optimization of Large Scale Systems*, pp. 57–68. Berlin, Heidelberg: Springer.

Chen, B., Z. Cai, and M. Bergés (2020). Gnu-RL: A Practical and Scalable Reinforcement Learning Solution for Building HVAC Control Using a Dif-

ferentiable MPC Policy. *Frontiers in Built Environment 0*. Publisher: Frontiers.

Drgoňa, J., J. Arroyo, I. Cupeiro Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, and L. Helsen (2020, January). All you need to know about model predictive control for buildings. *Annual Reviews in Control 50*, 190–232.

Goia, F., L. Finocchiaro, and A. Gustavsen (2015). *The ZEB Living Lab at the Norwegian University of Science and Technology: a zero emission house for engineering and social science experiments*. Danmarks Tekniske Universitet, DTU. Accepted: 2018-05-24T13:35:30Z Publication Title: 1-10.

Gros, S. and M. Zanon (2020, February). Data-Driven Economic NMPC Using Reinforcement Learning. *IEEE Transactions on Automatic Control 65*(2), 636–648. Conference Name: IEEE Transactions on Automatic Control.

Hjalmarsson, H. (2009, January). System Identification of Complex and Structured Systems. *European Journal of Control 15*(3), 275–310.

Kühl, P., M. Diehl, T. Kraus, J. P. Schlöder, and H. G. Bock (2011, January). A real-time algorithm for moving horizon state and parameter estimation. *Computers & Chemical Engineering 35*(1), 71–83.

Technical University of Denmark (2015). *Thermal Performance Characterization using Time Series Data - IEA EBC Annex 58 Guidelines*. Publication Title: Thermal Performance Characterization using Time Series Data - IEA EBC Annex 58 Guidelines.

Maree, J. P., S. Gros, and H. T. Walnum (2021, June). Adaptive control and identification for heating demand-response in buildings. In *2021 European Control Conference (ECC)*, pp. 1931–1936.

OpenAI (2023, March). GPT-4 Technical Report. arXiv:2303.08774 [cs].

OpenAI, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang (2019, December). Dota 2 with Large Scale Deep Reinforcement Learning. arXiv:1912.06680 [cs, stat].

Rawlings, J. B., D. Q. Mayne, and M. Diehl (2017). *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing. Google-Books-ID: MrJctAEACAAJ.

Rockett, P. and E. A. Hathway (2017, July). Model-predictive control for non-domestic buildings: a critical review and prospects. *Building Research & Information 45*(5), 556–571.

Serale, G., M. Fiorentini, A. Capozzoli, D. Bernardini, and A. Bemporad (2018, March). Model Predictive Control (MPC) for Enhancing Building and HVAC System Energy Efficiency: Problem Formulation, Applications and Opportunities. *Energies 11*, 631.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis (2016, January). Mastering the game of Go with deep neural networks and tree search. *Nature 529*(7587), 484–489. Number: 7587 Publisher: Nature Publishing Group.

Strachan, N., B. Fais, and H. Daly (2016, February). Reinventing the energy modelling–policy interface. *Nature Energy 1*(3), 1–3. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 3 Primary_atype: Comments & Opinion Publisher: Nature Publishing Group Subject_term: Decision making;Energy economics;Energy modelling;Policy Subject_term_id: decision-making;energy-economics;energy-modelling;policy.

Sutton, R. S. and A. Barto (2020). *Reinforcement learning: an introduction* (Second edition ed.). Adaptive computation and machine learning. Cambridge, Massachusetts London, England: The MIT Press.

Wetter, M., W. Zuo, T. Nouidui, and X. Pang (2014, July). Modelica Buildings library. *Journal of Building Performance Simulation 7*.

Yu, X., L. Georges, M. D. Knudsen, I. Sartori, and L. Imsland (2019). Investigation of the Model Structure for Low-Order Grey-Box Modeling of Residential Buildings. Rome, Italy, pp. 5076–5083.

Zanon, M. and S. Gros (2021, August). Safe Reinforcement Learning Using Robust MPC. *IEEE Transactions on Automatic Control 66*(8), 3638–3652. arXiv:1906.04005 [cs].

Zhang, Z. and K. P. Lam (2018, November). Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *Proceedings of the 5th Conference on Systems for Built Environments*, BuildSys '18, New York, NY, USA, pp. 148–157. Association for Computing Machinery.