Lessons Learned from Data-Driven Building Control Experiments: Contrasting Gaussian Process-based MPC, Bilevel DeePC, and Deep Reinforcement Learning

Loris Di Natale*, Yingzhao Lian*, Emilio T. Maddalena*, Jicheng Shi and Colin N. Jones

Abstract— This manuscript offers the perspective of experimentalists on a number of modern data-driven techniques: model predictive control relying on Gaussian processes, adaptive data-driven control based on behavioral theory, and deep reinforcement learning. These techniques are compared in terms of data requirements, ease of use, computational burden, and robustness in the context of real-world applications. Our remarks and observations stem from a number of experimental investigations carried out in the field of building control in diverse environments, from lecture halls and apartment spaces to a hospital surgery center. The final goal is to support others in identifying what technique is best suited to tackle their own problems.

I. INTRODUCTION

Many control techniques fall under the umbrella of the not strictly defined "data-driven control" and "learning-based control" categories [1], [2]. A fair number of successful implementations of these methods in different domains show that such approaches are well suited to operate physical systems while attaining high performance [3], [4]. Most of these methods aim at reducing or eliminating the engineering burden of modelling the physical system one intends to control (in the white-box sense), leveraging data to characterize system dynamics in a more general way. This is in contrast with *direct* data-driven control methods, which improve the control performance without any plant characterization [5].

Among the numerous linear modelling paradigms [6], behavioral theory offers a way to implicitly characterize the system dynamics based on past data [7]. Many nonlinear parametric models are extensions of linear ones [8], such as local linear model trees [9], Uryson operators [10], or Hammerstein models [11]. These methods typically enforce nonlinearity using polynomial convolution [12] or explicit nonlinear structures [8]. While these standard approaches suffer from the infamous curse of dimensionality, deep Neural Network (NN) models have recently become a popular choice thanks to their scalability, flexibility, and strong representative power, and were already successfully used to characterize building temperature dynamics, e.g. in [13]. Due to their ability to quickly learn objectives and dynamics even with little data, non-parametric models have also drawn the attention of system and control researchers. Gaussian Processes (GPs) are arguably the most popular choice thanks to their inherent uncertainty quantification [3], [14].

Once a model has been chosen, it is usually used either within receding horizon predictive controllers online, based on forecasts, like in typical Model Predictive Control (MPC) schemes, or to learn explicit control laws offline, e.g. through Deep Reinforcement Learning (DRL) [15]. Both techniques were already successfully deployed to control the temperature in real buildings, e.g. in [16]–[21]. Since any combination of a model and a decision maker mentioned above might be chosen, there exist many flavors of data-driven building control, as reviewed e.g. in [22]–[24].

In this work, we focus our analysis on three real-world building control experiments using distinct combinations of models and controllers: GP-based MPC [25], bilevel DeePC [26] – relying on behavioral system theory – and DRL matched with NN dynamical models [27]. This selection is representative of the various possible choices as it covers linear, nonlinear, parametric, and non-parametric models, as well as both predictive and explicit control laws. Furthermore, these controllers were experimentally validated on different case studies, namely a hospital, a lecture hall, and an apartment, in distinct locations (see Fig. 1). This paper hence offers a qualitative comparison of the pros and cons of each modelling and control technique through the lens of experimentalists. To the best of our knowledge, such a systematic comparison of different data-driven methods with experimental validations does not yet exist. This paper thus provides tools to help the reader decide which controller is well-suited for a particular application.

II. PRELIMINARIES

In this section, we briefly review the fundamentals of the three techniques explored in our experiments to create an intuition, and the interested reader can follow the provided pointers to key references for more detail. Throughout this work, we assume a data set of state, input, disturbance and output measurements has been gathered in $\{x_t, u_t, w_t, y_t\}_{t=1}^N$.

Fig. 1: The buildings where our experiments were conducted: the NEST building, Duebendorf, Switzerland (left, © Zooey Braun, Stuttgart); the São Julião hospital surgery center, Campo Grande, Brazil (right, top); the Polydome hall, Lausanne, Switzerland (right, bottom).

A. Gaussian Process-based MPC

Being grounded in Bayesian non-parametric statistics, GPs offer the user a principled way to learn an unknown function $f(x)$ from data [14]. The two principal expressions in the GP setting are the following:

$$\mu(x) = m(x) + k_X(x)^\top (K + \sigma^2 I)^{-1}(f_X - m_X) \quad \text{(1a)}$$
$$\text{var}(x) = k(x,x) - k_X(x)^\top (K + \sigma^2 I)^{-1} k_X(x) \quad \text{(1b)}$$

where $\mu(x)$ is the learned predictive function – the surrogate for the unknown $f(x)$ – and $\text{var}(x)$ reflects the degree of uncertainty associated with each prediction. The choice of the kernel function $k(\cdot,\cdot)$ is central in GP modelling and gives rise to the vector $k_X(x)$ and the matrix K when combined with a data set $\{x_m, f(x_m)\}_{m=1}^M$, with $[k_X(x)]_i := k(x_i, x)$ and $[K]_{ij} := k(x_i, x_j)$ [14]. Similarly, the choice of the mean function $m(\cdot)$ defines the vector $[m_X]_i := m(x_i)$, and $(f_X)_i := f(x_i)$. Lastly, σ^2 is a scalar that represents the variance of the noise affecting the measurements in f_X.

In order to go from learning static functions $f(x)$ to dynamical systems, one can employ GPs to build either autoregressive or state-space models (see e.g. [25] and [28]). We opted for the former and used the final expressions within our MPC formulation [25]. Whereas the mean function $\mu(x)$ plays the role of the nominal dynamics, the variance information $\text{var}(x)$ is used to tighten the optimization problem constraints, conferring on it a certain degree of robustness. Although GP-MPC can be extended to incorporate online data in real-time, this feature was not exploited in our experimental investigation since it would bring about both theoretical and computational challenges [29], [30].

B. Robust Bilevel DeePC

Behavioral system theory relies on Hankel matrices, defined as follows for a time series $s := \{s_i\}_{i=1}^T$:

$$\mathfrak{H}_L(s) := \begin{bmatrix} s_1 & s_2 & \cdots & s_{T-L+1} \\ s_2 & s_3 & \cdots & s_{T-L+2} \\ \vdots & \vdots & & \vdots \\ s_L & s_{L+1} & \cdots & s_T \end{bmatrix},$$

where L is the depth of the matrix. The key idea behind any DeePC-related schemes is to use Willems' Fundamental Lemma [31] to generate trajectory predictions from a persistently excited (PE) data set (i.e. $\mathfrak{H}_L(u_\mathbf{d})$ is full row rank, where boldface subscripts $\mathbf{d}$ denotes past data from the training set). Under the PE condition, this lemma states that any L-step Input/Output (I/O) trajectory is in the column space of the stacked Hankel matrix $\mathfrak{H}_L(u_\mathbf{d}, y_\mathbf{d}) := \begin{bmatrix} \mathfrak{H}_L(u_\mathbf{d})^\top & \mathfrak{H}_L(y_\mathbf{d})^\top \end{bmatrix}^\top$. One can hence use past data to predict future trajectories implicitly, avoiding any explicit modelling of the underlying system [32]. The single-level presented in [32] combines the trajectory prediction and the determination of control input in a multi-objective single level optimization. Instead, the trajectory prediction and the determination of control decisions were hierarchically coupled through a bilevel optimization problem in our experiments [26]:

$$\min_{\substack{\overline{y}_{pred} \\ \overline{u}_{pred}, K}} J(y_{pred}, u_{pred})$$

$$\text{s.t. } \forall\, w_{pred} \in \mathcal{W} \quad \text{(2a)}$$
$$u_{pred} = \overline{u}_{pred} + G w_{pred} \in \mathcal{U}, \quad \text{(2b)}$$
$$y_{pred} = \mathfrak{H}_{L,pred}(y_\mathbf{d})g \in \mathcal{Y}, \quad \text{(2c)}$$
$$g \in \arg\min_{g_l} \|\mathfrak{H}_{L,init}(y_\mathbf{d})g_l - y_{init}\|^2 + g_l^\top \mathcal{E}_g g_l \quad \text{(2d)}$$

$$\text{s.t. } \begin{bmatrix} \mathfrak{H}_{L,init}(u_\mathbf{d}) \\ \mathfrak{H}_{L,init}(w_\mathbf{d}) \\ \mathfrak{H}_{L,pred}(u_\mathbf{d}) \\ \mathfrak{H}_{L,pred}(w_\mathbf{d}) \end{bmatrix} g_l = \begin{bmatrix} u_{init} \\ w_{init} \\ u_{pred} \\ w_{pred} \end{bmatrix},$$

where $u_{init}, w_{init}, y_{init}$ are t_{init}-step sequences of measured inputs, disturbances and outputs preceding the current point in time. u_{pred} and y_{pred} are the predicted closed-loop inputs and outputs for a specific realization of the future disturbances w_{pred}. They are therefore uncertain and have to satisfy I/O constraints $\mathcal{U}/\mathcal{Y}$ for any realization of w_{pred} in the predictive disturbance set $\mathcal{W}$. For given realizations of w_{pred} and u_{pred}, y_{pred} is given by the lower level problem (2c) and (2d), where $\mathcal{E}_g$ is diagonal with decreasing value to model the preference of using recent data.

In our experiments, the data set was robustly updated online: the input signal was randomly perturbed to guarantee the PE condition and the perturbation was treated as an auxiliary disturbance in $\mathcal{W}$. The advantages of this bilevel approach over the single-level one [32] are detailed in [26].

C. Deep Reinforcement Learning

In Reinforcement Learning (RL), an agent interacts with the system in an iterative process: at each time step t, the agent chooses the control input u_t based on the measurement x_t. The system then evolves to the next state x_{t+1}, and returns it to the agent, along with the step reward $r(x_t, u_t)$, and the procedure repeats until some end criterion is met. The goal of any RL algorithm is to find the optimal policy $\pi(u_t|x_t)$ that maximizes the expected returns:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(x_t, u_t) \right], \tag{3}$$

where $0 < \gamma < 1$ is the discount factor, balancing short- and long-term rewards. One can readily see the link with the objective function used in MPC, with the notable difference that constraints cannot be incorporated in classical RL agents and need to be learned implicitly, typically through penalties in the reward function. In practice, one usually parametrizes the policy as π_θ and optimizes over the parameters θ. Most of the recent successes of RL were obtained through NN-based policies, which is then referred to as DRL. A plethora of algorithms to solve such (D)RL optimization problems have been developed in the past few years, and we rely on Twin Delayed Deep Deterministic policy gradients (TD3) [33] for our experiments [27].

Since classical DRL agents cannot ensure constraint satisfaction and are very data-inefficient, they cannot be deployed in a building from scratch. Indeed, they could take months or years to converge, all the while violating the comfort of the occupants [24]. Consequently, in practice, we have to rely on simulation environments to train DRL agents offline before deploying them on the real system. In this work, we leverage Physically Consistent Neural Networks (PCNNs) to learn the temperature dynamics from data [13]. These physics-inspired NNs guarantee the physical consistency of the models, e.g. the positive correlation between heating/cooling and temperature increases/decreases, which is crucial to train meaningful DRL agents that will interact with buildings.

III. EXPERIMENTAL SETUPS

The common objective of the experiments described herein was to minimize the energy consumption of each building while guaranteeing thermal comfort for the occupants, i.e. maintaining indoor temperatures within predefined bounds. Note, however, that the size, occupancy profile, and actuation systems of each case study are considerably different.

A. Hospital Surgery Center

The São Julião hospital surgery center, depicted in Fig. 1, is a busy environment where over 500 surgical procedures are carried out monthly. It is located in Campo Grande, Brazil,
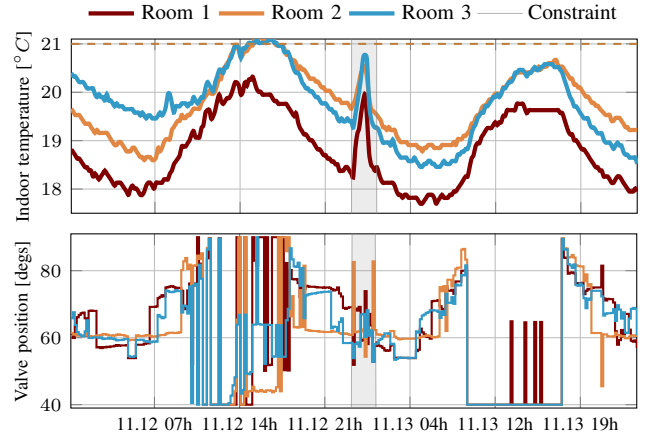

Fig. 2: Hospital November $11^{th} - 12^{th}$, 2021: three rooms controlled by the GP-MPC controller.

where daily maximum temperatures reach $30\,°C$ even in winter. The building features an industrial forced-air cooling system to guarantee indoor thermal comfort. The study presented in [25] was concerned with its ophthalmology section, composed of three rooms: two operating theaters and one intermediate zone. 31 days worth of clean data were collected, sampled every $10\,\text{min}$, and three individual GP models were then learned based roughly on 275 points each.

The regulation task was performed over four days and the temperature constraint consisted in a single upper bound at $21\,°C$. In Fig. 2, the temperatures and actuation profiles throughout two days are shown, and the period highlighted in grey corresponds to a chiller fault discussed in [25]. The adopted GP-MPC solution was shown to yield good performance even when subject to harsh outdoor conditions and strong internal disturbances (see [25, Sec. 4.1]). By means of simulations, it was also compared to a number of alternatives, such as ON/OFF and Proportional-Integral (PI) controllers, among others, and the results showcased the ability of GP-MPC to balance indoor thermal regulation and the chiller energy consumption.

B. Polydome

The *Polydome* is a freestanding $600m^2$ single-zone building on the EPFL campus, regularly used as a lecture hall, that can accommodate up to 200 people (Fig. 1). The indoor climate is maintained through a rooftop HVAC unit that we controlled for 20 days. The sampling time is $15\,\text{min}$ in this system. Its control input is the electricity power of the HVAC, outdoor temperature and solar radiation are considered as disturbance to the indoor temperature (i.e. output).

Over the different experiments, we observed two main benefits of the proposed robust bilevel DeePC method over a non-robust version. Firstly, it is more robust to uncertain external weather conditions, especially when an economic objective function is considered. When the non-robust controller was deployed, we could indeed observe comfort violations around $4\,\text{h}$ and $9\,\text{h}$, something that did not happen with the robust approach, as demonstrated in Fig. 3. Furthermore,
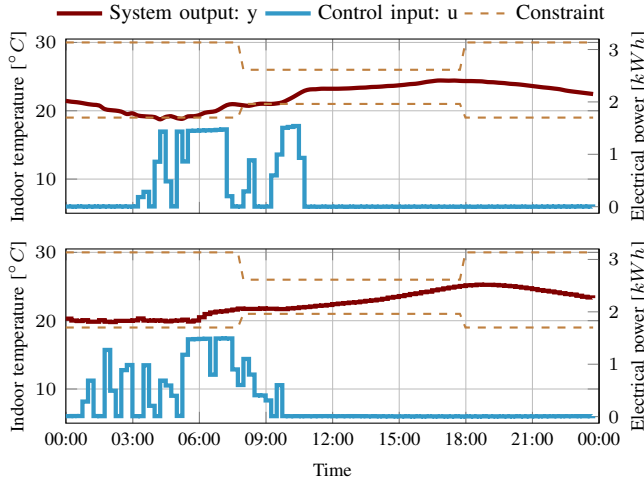
Fig. 3: *Polydome*. Top: May 14th, 2021: non-robust bilevel DeePC. Bottom: May 25th, 2021: robust bilevel DeePC

Fig. 4: UMAR, February $15^{th}-18^{th}$, 2022: one bedroom controlled by a classical rule-based controller (red), the other by the proposed DRL agent (blue). The connection was interrupted in the shaded area and default controllers took over.

these violations led to unwanted overheating behavior later in the day around 11 h.

Beyond robustness, the bilevel controller also showed strong adaptivity to the working conditions. Indeed, it performed consistently along a 20-day long experiment even though the outdoor temperature varied from 15 °C to 29 °C. The full plots of this experiment are available at https://github.com/YingZhaoleo/Building_results.

C. UMAR

The Urban Mining and Recycling (UMAR) unit is an apartment with two almost identical bedrooms and a living room between them in the NEST building, pictured in Fig. 1. All the rooms are heated or cooled by letting hot or cold water run through ceiling panels, respectively, and the water is brought to the required temperature by a large reversible heat pump in the basement. During our heating experiment, the DRL agent hence controlled when to open or close the valves of the ceiling panels, with a sampling time of 15 min.

The measurements x fed to the agents regrouped autoregressive terms of room temperatures and weather conditions, as well as time information and the comfort bounds. The reward function was designed as follows:

$$r(x,u) = -\max\{T - b_u, 0\} - \max\{b_l - T, 0\} - \lambda u,$$

where T is the indoor temperature, b_u and b_l represent the current upper and lower temperature bounds, and λ balances the comfort of the occupants and the energy consumption. More details on the setup and simulation results can be found in [27], where DRL were shown to clearly outperform industrial baselines and attain near-optimal performance.

One of the best agents was subsequently deployed in one of the bedrooms to validate its performance, and a classical hysteresis controller was deployed in the other bedroom for comparison. In Fig. 4, one can observe how the DRL agent is able to maintain the temperature close to the lower bound, thereby saving energy compared to the
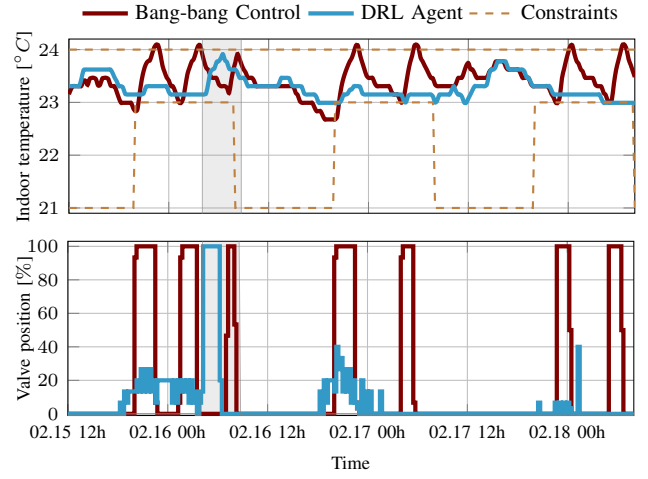
hysteresis controller, except when the default controller took over when the connection was lost for a few hours (shaded grey). Interestingly, despite not having access to forecasts, the agent learned to preheat the room to ensure the tightened temperature bounds are respected.

IV. DISCUSSION

To begin our discussion, we want to underline that the three techniques are not in complete opposition to each other. Indeed, one could utilize PCNNs within MPC formulations and, conversely, GPs or behavioral models could be used as simulators to train DRL agents offline. This also highlights the fact that, despite being often advertised as a model-free method, DRL still requires high-fidelity simulators (e.g. the models discussed herein) to be trained offline before being deployed on the real system to ensure adequate operations, a matter that is often overlooked.

The rest of this section discusses the advantages and challenges encountered using the proposed controllers on real buildings and aims at helping the reader understand which method is best suited for their application. An overview of the strengths and weaknesses of each technique is provided in Table I and the details are discussed below.

A. System characteristics

Before opting for any of the proposed methods, one should ponder over the properties of the building to be controlled and its actuation system. Whereas DeePC works well for approximately linear systems, it cannot be expected to handle strong nonlinearities as well as GP-MPC or DRL.[1] In the building domain, this becomes especially apparent if the algorithms directly interact with low-level actuators, such as

---

[1]Note that there exist nonlinear extensions of DeePC (see e.g [34], [35]) based on either a linearized model or infinite dimensional dynamics. This paper only focus on the linear version as it is the one deployed in out experiment.

		GP MPC	DeePC	PCNN DRL
A.	Integration of nonlinearities	+	−	+
	Integration of prior knowledge	+	−	+
B.	Data treatment	−	o	+
	Data efficiency	+	+	−
C.	Ease of implementation	o	−	+
	Tuning effort	−	+	o
	Training time	+	N/A	−
D.	Integration of constraints	+	+	−
	Robustness to noise	+	+	o
	Robustness to unseen data	o	+	−
E.	Offline computational complexity	−	+	−
	Online computational complexity	−	o	+
F.	Interpretability	+	+	−
	Need of forecasts	−	−	+

'+': Advantage, 'o': Neutral, '-': Disadvantage

TABLE I: Qualitative comparative assessment of the different control methods based on our experience.

ventilator damper positions in forced-air systems. In practice, it is thus preferred to use DeePC for high-level control tasks, e.g. modifying temperature setpoints or directly modulating the HVAC power consumption.

Besides their compatibility with nonlinear dynamics, the possibility of integrating prior knowledge is another advantage of both GPs and PCNNs over DeePC. A concrete example would be to enforce a positive correlation between the input thermal power applied to a zone and its temperature. This is a desired property since this flaw might otherwise be exploited by MPC or DRL controllers, leading to spurious decisions.

B. Data considerations

In any of the three techniques, data collection and pre-processing naturally represent major workloads. Moreover, even though control performance is heavily influenced by the data quality, it is often difficult to formally measure and quantify this influence, especially in the nonlinear setting (see e.g. [29]). The pragmatic approach is typically to gather 'rich enough' data-sets around different operating points and then rely on different model validation procedures.

Since GPs have the ability to interpolate [14], users have to pay particular attention to how clean the data provided to the algorithm is. On the other hand, PCNNs and NNs in general require less data cleansing and curation, but need larger batches of it to attain a good fit. Quantitatively, in our case studies, GP-MPC and DeePC were indeed much more sample efficient, using respectively 794 and 192 data points, whereas close to $100'000$ were used to train our PCNNs, although less might suffice.

C. Training and tuning

Once an appropriate batch of data has been gathered and cleaned, one needs to train and fine-tune the models and/or controllers. Bilevel DeePC is the least time-consuming method at this stage, especially thanks to the existence of theoretically optimal hyperparameters [26]. Certain regularization constants used in the single-level DeePC

are however still found through exhaustive search and cross-validation. Slightly more tuning is required for GPs since the user has to experiment with different kernel functions and additional hyperparameters, such as the delays of the autoregressive structures. Finally, the training of PCNNs and DRL agents can easily be ranked as the most demanding one since both the model and the controller rely on NNs, which are notoriously computationally intensive to fit and involve tuning several hyperparameters. Additionally, since classical DRL agents do not incorporate knowledge of constraints, they have to be exposed to all possible scenarios during the training phase to be able to react to different operating conditions online, which might extend the training time further.

Despite the above considerations, we want to highlight here that none of the techniques described in this paper require any structural or parametric information from the plant. In the building context, this translates to not needing access to any architectural information or construction materials, which is an important advantage over traditional white-box modelling approaches.

D. Behavior during the deployment

Some data-driven techniques are known to be brittle and might fail when faced with situations not presented to them during training. Classical DRL agents are typically prone to produce spurious decisions since they rely on NNs, which are particularly vulnerable to unseen data [13]. This is aggravated by the lack of explicit constraints and might lead to temperatures diverging outside the desired envelope.

Conversely, DeePC and GP-MPC both take constraints into account when they compute the control actions, typically leading to more robust operations. Nevertheless, if the constraints are violated for some reason (perhaps exogenous to the controller), aggressive responses might be triggered. This was for example the cause of the suboptimal overheat at around $10\,\mathrm{h}$ in the top plot of Fig. 3. Such events might cause performance degradation, but they can often be avoided in practice, using either the uncertainty estimates provided by GPs or the robust bilevel DeePC formulation. Note that the nature of the uncertainty considered is not the same in both cases: while GP-MPC incorporates model uncertainty, the robust bilevel DeePC characterizes external disturbances instead.

E. Computational complexity

The computational complexity has to be broken down into its offline and online components. The PCNN-DRL framework is clearly the method requiring the most computations offline, since large NNs have to be fit for both the model and the controller. Including the tuning procedure, it can easily take a few days to find good models and controllers on a GPU. By contrast, using up to 300 data points, we could fit GPs in $15\,\mathrm{s}$ to $2\,\mathrm{min}$ without GPU acceleration. Lastly, DeePC does not entail offline computations.

Concerning the online computations, DRL agents provide control inputs almost immediately since they only require

a single pass through the control NN, i.e. a few sequential matrix multiplications. On the other hand, both GP-MPC and DeePC require solving optimization problems to generate control inputs at each time step. DeePC boils down to Quadratic Programs (QPs), which were typically solved in under a second in our experiments, even with open-source solvers like OSQP. GP-MPC is surely the most computationally demanding technique at run-time as it requires solving non-convex programs with highly nonlinear equality constraints. To alleviate this load, we initialized the solver with feasible state and control trajectories (obtained with a baseline controller), and were able to compute the control actions typically within $20\,\mathrm{s}$. Recall that the sampling periods in the building domain are conventionally between 5-15 min, so that these optimizations were not an issue in practice.

F. Additional comments

Understanding the rationale behind the actions of DRL agents is not easy, especially when NN parametrizations are used. By backtracking the optimal sequences of states and controls, some explainability can be achieved in GP-MPC and DeePC, a feature that can even help users fine-tune their formulations, hence reducing the deployment time.

Remarkably, predictive control methods require forecasts of all the disturbances over the prediction horizon. Whereas weather forecasts are available on many open-source platforms, such as windy.com, internal heat gains are typically much harder to estimate. While good temperature predictions could be obtained without them in the Polydome, they were important to fit the GPs to the hospital data. On the other hand, DRL agents do not strictly require access to the aforementioned forecasts since they are able to implicitly anticipate future exogenous conditions from data. Nonetheless, one could easily add predictive information to the observations fed to the agents at each time step to create more robust control policies.

V. FINAL REMARKS

This paper provided a pragmatic, qualitative comparison among GP-MPC, DeePC, and DRL for building control. In short, GPs have the great advantages of being sample efficient and yielding uncertainty estimates, but require more manual tuning than its competitors and result in more complex real-time optimization problems when paired with MPC. DeePC, on the other hand, is only well suited for linear or linearly well-approximated systems, but remains the most straighforward method to setup and deploy. Finally, the PCNN-DRL pipeline removes the need for any expert knowledge and is flexible enough to tackle many problems, but requires large amounts of data, a time-consuming offline training phase, and can cause online constraint violations.

Each approach clearly has its own merits, and they were all able to successfully handle their task. Nevertheless, arguments were put forward to show that none of the techniques can be seen as the solution to every building control problem, and an informed choice has to be made based on the plant characteristics, the amount and quality of the data, and the available computational power.

REFERENCES

[1] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[2] M. Benosman, "Model-based vs data-driven adaptive control: an overview," *International Journal of Adaptive Control and Signal Processing*, vol. 32, no. 5, pp. 753–776, 2018.

[3] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-driven MPC for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.

[4] P. G. Carlet, A. Favato, S. Bolognani, and F. Dörfler, "Data-driven continuous-set predictive current control for synchronous motor drives," *IEEE Transactions on Power Electronics*, vol. 37, no. 6, pp. 6637–6646, 2022.

[5] S. Formentin, D. Piga, R. Tóth, and S. M. Savaresi, "Direct learning of LPV controllers from data," *Automatica*, vol. 65, pp. 98–110, 2016.

[6] L. Ljung, "System identification," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[7] J. C. Willems and J. W. Polderman, *Introduction to mathematical systems theory: a behavioral approach*. Springer Science & Business Media, 1997, vol. 26.

[8] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[9] O. Nelles, "Nonlinear system identification," 2002.

[10] P. Gallman, "An iterative method for the identification of nonlinear systems using a uryson model," *IEEE Transactions on Automatic Control*, vol. 20, no. 6, pp. 771–775, 1975.

[11] S. Billings and S. Fakhouri, "Non-linear system identification using the hammerstein model," *International Journal of Systems Science*, vol. 10, no. 5, pp. 567–578, 1979.

[12] P. G. Gallman, *Identification of nonlinear dynamical systems using the Uryson operator*. Yale University, 1971.

[13] L. Di Natale, B. Svetozarevic, P. Heer, and C. N. Jones, "Physically consistent neural networks for building thermal modeling: theory and analysis," *arXiv preprint arXiv:2112.03212*, 2021.

[14] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[16] F. Bünning, B. Huber, P. Heer, A. Aboudonia, and J. Lygeros, "Experimental demonstration of data predictive control for energy optimization and thermal comfort in buildings," *Energy and Buildings*, vol. 211, p. 109792, 2020.

[17] S. Yang, M. P. Wan, W. Chen, B. F. Ng, and S. Dubey, "Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization," *Applied Energy*, vol. 271, p. 115147, 2020.

[18] L. Di Natale, B. Svetozarevic, P. Heer, and C. Jones, "Deep Reinforcement Learning for room temperature control: a black-box pipeline from data to policies," in *Journal of Physics: Conference Series*, vol. 2042, no. 1. IOP Publishing, 2021, p. 012004.

[19] B. Svetozarevic, C. Baumann, S. Muntwiler, L. Di Natale, M. N. Zeilinger, and P. Heer, "Data-driven control of room temperature and bidirectional EV charging using deep reinforcement learning: Simulations and experiments," *Applied Energy*, vol. 307, p. 118127, 2022.

[20] S. Touzani, A. K. Prakash, Z. Wang, S. Agarwal, M. Pritoni, M. Kiran, R. Brown, and J. Granderson, "Controlling distributed energy resources via deep reinforcement learning for load flexibility and energy efficiency," *Applied Energy*, vol. 304, p. 117733, 2021.

[21] Y. Du, F. Li, K. Kurte, J. Munk, and H. Zandi, "Demonstration of Intelligent HVAC Load Management With Deep Reinforcement Learning: Real-World Experience of Machine Learning in Demand Control," *IEEE Power and Energy Magazine*, vol. 20, no. 3, pp. 42–53, 2022.

[22] J. Y. Park and Z. Nagy, "Comprehensive analysis of the relationship between thermal comfort and building control research-a data-driven literature review," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 2664–2679, 2018.

[23] E. T. Maddalena, Y. Lian, and C. N. Jones, "Data-driven methods for building controlA review and promising future directions," *Control Engineering Practice*, vol. 95, p. 104211, 2020.

[24] Z. Wang and T. Hong, "Reinforcement learning for building controls: The opportunities and challenges," *Applied Energy*, vol. 269, p. 115036, 2020.

[25] E. T. Maddalena, S. A. Muller, R. M. d. Santos, C. Salzmann, and C. N. Jones, "Experimental data-driven model predictive control of a hospital HVAC system during regular use," *Energy and Buildings*, p. 112316, 2022.

[26] Y. Lian, J. Shi, M. P. Koch, and C. N. Jones, "Adaptive robust data-driven building control via bi-level reformulation: an experimental result," *arXiv preprint arXiv:2106.05740*, 2021.

[27] L. Di Natale, B. Svetozarevic, P. Heer, and C. N. Jones, "Near-optimal deep reinforcement learning policies from data for zone temperature control," *arXiv preprint arXiv:2203.05434*, 2022.

[28] S. Eleftheriadis, T. Nicholson, M. Deisenroth, and J. Hensman, "Identification of gaussian process state space models," *Advances in neural information processing systems*, vol. 30, 2017.

[29] A. Lederer, A. Capone, J. Umlauft, and S. Hirche, "How training data impacts performance in learning-based control," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 905–910, 2020.

[30] A. Capone, G. Noske, J. Umlauft, T. Beckers, A. Lederer, and S. Hirche, "Localized active learning of gaussian process state space models," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 490–499.

[31] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[32] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 307–312.

[33] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1587–1596.

[34] J. Berberich, J. Köhler, M. A. Muller, and F. Allgower, "Linear tracking mpc for nonlinear systems part ii: The data-driven case," *IEEE Transactions on Automatic Control*, 2022.

[35] Y. Lian and C. N. Jones, "Nonlinear data-enabled prediction and control," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 523–534.