Data-Driven Predictive Control With Improved Performance Using Segmented Trajectories

Edward O’Dwyer*, Eric C. Kerrigan, Senior Member, IEEE, Paola Falugi, Member, IEEE, Marta Zagorowska, and Nilay Shah

Abstract—A class of data-driven control methods has recently emerged based on Willems’ fundamental lemma. Such methods can ease the modeling burden in control design but can be sensitive to disturbances acting on the system under control. In this article, we propose a restructuring of the problem to incorporate segmented prediction trajectories. The proposed segmentation leads to reduced tracking error for longer prediction horizons in the presence of unmeasured disturbance and noise when compared with an unsegmented formulation. The performance characteristics are illustrated in a set-point tracking case study in which the segmented formulation enables more consistent performance over a wide range of prediction horizons. The method is then applied to a building energy management problem using a detailed simulation environment. The case studies show that good tracking performance is achieved for a range of horizon choices, whereas performance degrades with longer horizons without segmentation.

Index Terms—Building energy management, data-driven predictive control, optimal control, Willems’ fundamental lemma.

I. INTRODUCTION

The increased focus on digital technology in recent times has drawn attention to data-driven control methods, with applications ranging from building control [1], to autonomous vehicles [2]. By reducing the modeling burden in the control design phase, the deployment of advanced control can be streamlined, with control decisions directly obtained from data measurements [3]. Control methods that rely directly on measurements, without using explicit models, are called direct data-driven methods. In data-rich environments, such methods may then be advantageous. Nonetheless, direct data-driven methods often lack the theoretical foundations of indirect data-driven methods, whereby data are needed to explicitly derive a model which is then used for control [4].

To overcome this, Willems’ fundamental lemma [5] has recently been used as a foundation for a large body of data-driven control research, because it provides theoretical foundations for control decisions obtained directly from data. Methods stemming from the lemma rely on the insight that, under certain straightforward excitation conditions, historical data structures can be used to project all permissible trajectories of a linear time-invariant (LTI) system. By building on the fundamental lemma, theoretical foundations can be achieved for certain direct data-driven methods. Predictive control representations have been developed that can offer stability and certain robustness guarantees [4], without requiring the derivation of a parametric model. For example, a data-driven counterpart to model predictive control (MPC) named data-enabled predictive control (DeePC) was proposed in [6] and shown to be competitive with MPC in the deterministic case. In [7], an equivalence was then shown between this direct data-driven approach and an alternative subspace-based approach [subspace predictive control (SPC)], in which the parameters of a multistep prediction model are derived using the same training data criteria. This was expanded upon in [8], where further analysis of the performance of these direct and indirect formulations was carried out for different systems using various relaxation and regularization techniques. The results suggested that noisy data have a greater impact on direct formulations, while system nonlinearities have a greater impact on indirect formulations. Additionally, rather than relying on a single-training period for data acquisition, a strategy was developed in [9] by which multiple, potentially short, datasets can be used instead, without compromising the theoretical foundation of the fundamental lemma.

Uncertainty in data measurements will impact the performance of a data-driven approach, and thus, several methods have been developed to ensure viability in stochastic settings. Alpago et al. [10] supplement a data-driven controller with a data-driven extended Kalman filter to reduce sensitivity to noise. Robust formulations have also been developed to enable performance guarantees under certain conditions of system stochasticity, such as the robust modification proposed in [11], ensuring exponential stability in the presence of measurement noise. In [12], a chance-constrained distributionally robust
formulation was developed for stochastic LTI systems, providing probabilistic guarantees on performance. Tractable, robust formulations are proposed to ensure performance guarantees under uncertainty in [13], while in [14], a correspondence was found between the fundamental lemma perspective and that of system-level synthesis (SLS), which was then exploited to formulate a robust closed-loop data-predictive controller. A robust building-level controller was implemented in a real building in [15], whereby an active excitation method was used to allow for the data structures to be updated continually while maintaining the persistence of excitation.

In the presence of noise or unmeasured disturbances, data-driven approaches use regularization and relaxation to maintain robustness guarantees and to align the DeePC problem to more established SPC methods [8]. However, while methods exist to preclude acausal relationships between inputs and outputs in SPC, the lack of an explicit characterization of the input–output relationship in direct DeePC mean that enforcing such conditions is not straightforward. Performance under noise and disturbance can then be affected, particularly for longer prediction horizons. In this article, we propose a restructuring of the data-enabled predictive controller formulation whereby the prediction trajectory is divided into multiple shorter trajectories, denoted segments. These segments can be identified in the same manner as the unsegmented formulation, with less training data. By exploiting the problem structure, computational effort can be reduced for longer horizon choices. This restructuring precludes most potential acausal dependencies between inputs and outputs, which could otherwise lead to degraded performance. The method is analyzed and compared with the unsegmented version in a set-point tracking case study with different horizon lengths and disturbance and noise realizations. With segmentation, a performance improvement is shown in terms of set-point tracking error, while a linear computational time increase is observed for increasing horizon lengths, which favors the nonlinear increase observed for an unsegmented formulation. Following this, a building energy management case study is implemented, based on a detailed building simulation environment with realistic disturbances. Comfort and energy cost objectives are solved in a prioritized manner. As the segmented-trajectory approach behaves more consistently than the unsegmented approach for longer horizon lengths, a reduction in both cost and energy consumption is achieved for a one-day-ahead prediction horizon.

In Section II, a background to the unsegmented data-driven predictive approach is provided based on the fundamental lemma, followed by the proposed modifications that result in a segmented formulation. In Section III, a set-point tracking case study is presented, with an analysis provided of the control performance and computational time associated with the segmented and unsegmented formulations. This is followed in Section IV by the building energy management case study, which is used to illustrate the benefits of the segmented formulation in a relevant application. This article ends with conclusions in Section V.

II. MODIFIED DATA-DRIVEN PREDICTIVE CONTROL FORMULATION

A. Data-Driven Predictive Control Preliminaries

As noted in the introduction, different variations of data-predictive control have been proposed. We provide a brief overview of the direct data-enabled predictive controller of [6] and the indirect multistep prediction approach of [7] and [8] in this section.

A discrete-time $n$th-order LTI state-space system can be represented at sample instant $k$ by

$$
\begin{align*}
x[k+1] &= Ax[k] + Bu[k] \\
y[k] &= Cx[k] + Du[k]
\end{align*}
$$

where $x[k] \in \mathbb{R}^n$ is the system state-vector, $u[k] \in \mathbb{R}^m$ and $y[k] \in \mathbb{R}^p$ are the input and output vectors, respectively, and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$ are parameter matrices, the parameters of which are assumed to be unknown. Following the terminology of [16], we define $\mathcal{B}$ as the behavior of (1), where the behavior is defined as the set of possible outcomes of the system. The lag of the system is denoted $\ell$, defined as the smallest integer for which the observability matrix $\mathcal{O}(A, C) := [C, CA, \ldots, CA^{\ell-1}]$ has full rank. With Willems’ fundamental lemma [5], arbitrary input and output sequences can be derived from a sufficiently long set of input–output data without explicitly estimating the parameters of (1). The input–output sequences will be called trajectories. A representation of the system can then be found and used for predictive control only in terms of measured data.

An offline data collection procedure is carried out to achieve this in which $T_0 \in \mathbb{Z}_{>0}$ sequences of persistently exciting input and output data measurements are given as $u_T = [u_0^T, \ldots, u_{T_0-1}^T] \in \mathbb{R}^{mT_0}$ and $y_T = [y_0^T, \ldots, y_{T_0-1}^T] \in \mathbb{R}^{pT_0}$, respectively, with $\mathbb{Z}_{>0}$ denote the set of positive integers. A trajectory $w$ is defined as persistently exciting of order $L_0$, $L_0 \in \mathbb{Z}_{>0}$, if the Hankel matrix $\mathcal{H}_{L_0}(w)$ is of full row-rank with

$$
\mathcal{H}_{L_0}(w) := \begin{bmatrix}
\begin{array}{cccc}
0 & \cdots & 0 & w_{T_0-L_0+1} \\
\vdots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & w_{T_0}
\end{array}
\end{bmatrix}
$$

Note that nonsquare Hankel matrices are permitted in this definition. From [5], for a controllable, observable $\mathcal{B}$, if $w \in \mathcal{B}$ is a persistently exciting, $T_0$-samples-long trajectory of order $\ell + n$, then any $t$-samples-long trajectory in $\mathcal{B}$ can be described as a linear combination of the columns of $\mathcal{H}(w)$, and any $\mathcal{H}(w)g$ is a trajectory of $\mathcal{B}$, where $g \in \mathbb{R}^{T_{\ell+t+1}}$. For persistent excitation, $T_0 \geq (m+1)(\ell+n) - 1$. Here, we seek to construct trajectories of length $N + T_{\text{ini}}$, where $N \in \mathbb{Z}_{>0}$ is the prediction horizon and $T_{\text{ini}} \in \mathbb{Z}$ is some initialization length. Following [17, Lemma 1], by fixing the first $T_{\text{ini}}$ samples of a trajectory, the subsequent $N$ samples are uniquely specified if $T_{\text{ini}} \geq \ell$.

The training data sequences $u_T$ and $y_T$ are arranged in the Hankel form of (2) with $T_0 \geq (m+1)(T_{\text{ini}} + N + n) - 1$ and $L_0 = T_{\text{ini}} + N$. The training data structures can then be be defined at this point as $U_T := \mathcal{H}_{T_{\text{ini}}} + N(u_T)$ and $Y_T := \mathcal{H}_{T_{\text{ini}}} + N(y_T)$.
These matrices are then partitioned such that the first $T_{\text{ini}}$ block rows of $U_u$ and $Y_u$ are denoted by the subscript $\alpha$ and are referred to as initialization data, with the remaining rows denoted by $\beta$ and referred to as prediction data. The partitioned data matrices are, thus, defined as

$$\begin{bmatrix} U_\alpha \\ U_\beta \end{bmatrix} := \mathcal{H}_{T_{\text{ini}}+N} (u_f)$$
$$\begin{bmatrix} Y_\alpha \\ Y_\beta \end{bmatrix} := \mathcal{H}_{T_{\text{ini}}+N} (y_f).$$

Defining initialization sequences $u_{\text{ini}} \in \mathbb{R}^{m_{\text{ini}}}$ and $y_{\text{ini}} \in \mathbb{R}^{p_{\text{ini}}}$ as the $T_{\text{ini}}$ most recent measurements, any future trajectories $u_f \in \mathbb{R}^{m_N}$ and $y_f \in \mathbb{R}^{p_N}$ can be found as the solution to

$$\begin{bmatrix} U_\alpha \\ Y_\alpha \\ U_\beta \\ Y_\beta \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ y_f \end{bmatrix}$$

where $g \in \mathbb{R}^{T_u - T_{\text{ini}} - N + 1}$. In this form, the data structures on the left-hand side of (4) contain the persistently excited training data, whereas the right-hand side contains the predicted trajectories, divided into initial, $u_{\text{ini}}$, and $y_{\text{ini}}$, and future, $u_f$, and $y_f$, portions.

This leads to the insight that $u_f$ and $y_f$, the future trajectories of $\mathcal{B}$, can be found for a given training dataset and given initialization trajectories $u_{\text{ini}}$ and $y_{\text{ini}}$. From this, a DeePC formulation was proposed in [6], whereby the following optimization is carried out:

$$\min_{g,u_f,y_f} V (g,u_f,y_f)$$

s.t.

$$\begin{bmatrix} U_\alpha \\ Y_\alpha \\ U_\beta \\ Y_\beta \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \\ y_f \end{bmatrix}$$

$$u_f \in \mathcal{U}$$

$$y_f \in \mathcal{Y}$$

with $V(\cdot)$ representing an objective to be minimized and $\mathcal{U}$ and $\mathcal{Y}$ representing the input and output constraint sets, respectively.

Whereas model parameters are not explicitly derived in this formulation, an equivalence was identified in [7] and [8] between this form and a predictive control formulation based on a multistep prediction model derived from data. This multistep model version is referred to in [8] as an indirect data-driven formulation, in contrast to direct data-driven formulations in which no model is identified, as such as in (5)–(8).

Using the indirect formulation, the same training data structures can be used, but here they are used to derive a multistep predictor $P^*$ by the least-squares method as

$$P^* = \arg \min_P \| P \begin{bmatrix} U_\alpha \\ Y_\alpha \\ U_\beta \end{bmatrix} - Y_\beta \|_F^2$$

where $\| \cdot \|_F$ denotes the Frobenius norm. Using the Moore–Penrose inverse (denoted $\dagger$), this can be expressed explicitly as

$$P^* := Y_\beta \begin{bmatrix} U_\alpha \\ Y_\alpha \\ U_\beta \end{bmatrix} \dagger.$$  \hspace{1cm} (10)

This predictor can then be used to derive future trajectories as

$$y_f = P^* \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_f \end{bmatrix}.$$  \hspace{1cm} (11)

With such a representation, a controller can be formulated using (11) to describe the system dynamics. Such a controller can be viewed as a form of SPC. To examine the model defined by (11) in Section II-B, it is useful here to define a partitioned version of $P^*$, given as $[P^*_1 \quad P^*_2 \quad P^*_3]$, where $P^*_1 \in \mathbb{R}^{p_N \times m_{\text{ini}}}$ is associated with the initialization input trajectory, $P^*_2 \in \mathbb{R}^{p_N \times m_{\text{ini}}}$ is associated with the initialization output trajectory and $P^*_3 \in \mathbb{R}^{p_N \times p_N}$ is associated with the future input trajectory.

An indirect data-driven predictive control formulation can then be represented by replacing (4) with (11). It should be noted that in (11), acausal dependencies between $y_f$ and $u_f$ are permitted. To ensure causality, a lower-block triangular structure would need to be enforced on $P^*_3$ [18].

To aid in the design of a direct data-driven controller and to understand its performance, Dörfler et al. [8] align the direct data-driven approach to the SPC approach via suitable regularization and relaxation choices. In particular, the regularization term $\|(I - \Pi)g\|_2^2$ is included in the proposed controller’s objective function, where

$$\Pi = \begin{bmatrix} U_\alpha \\ Y_\alpha \\ U_\beta \end{bmatrix} \dagger.$$  \hspace{1cm} (12)

For a noise-free, causal LTI system, trajectories that would necessitate an acausal relationship between inputs and outputs would not satisfy (4), as per the fundamental lemma. However, in the presence of noise and nonlinearity, and a relaxed form of (4), no such guarantee applies. Unlike the SPC case, restricting the set of allowable trajectories to those that satisfy causality conditions is not straightforward, since the link between inputs and outputs is not explicitly described. The objective of segmentation is to restructure (6) in a manner that discourages acausal relationships without significantly increasing complexity.

In Section II-B, both the direct and indirect methods summarized here will be used to illustrate the rationale of a modified version of the data-predictive control approach, which is the main contribution of this work.

### B. Segmentation of Prediction Trajectory

In the context of the data-driven controller described in Section II-A, relaxations of the initialization constraints and regularization of the optimization variables can be introduced to improve performance in this regard in the direct formulation. Similarly, slack variables can be introduced to the indirect form.
Nonetheless, the number of parameters of \( P_2^s \in \mathbb{R}^{m \times N} \) in the indirect form increases with the horizon length. For example, the final entry of the predicted output sequence, \( y_f[N] \), is a function of \( N \) preceding inputs \( (u_f[1], \ldots, u_f[N]) \). Thus, for longer prediction horizons, the impact of over-parameterization may become more pronounced, as will be shown in the illustrative example in Section III. In the direct formulation, model parameters are not explicitly derived; however, in [7] it is reasoned that the same model is implicitly identified in the direct form as the indirect form leading to the same performance drop for longer horizons. It should be noted that with perfect data, the fixing of \( u_{ini} \) and \( y_{ini} \) ensures a unique representation of \( u_f \) and \( y_f \), and thus, the issue does not arise.

A modification is proposed here whereby the prediction trajectory is divided into segments of length \( T_{ini} \) (with \( T_{ini} \leq N \)) to decouple the relationship between horizon length and the number of parameters (implicit parameters in the direct formulation, explicit parameters in the indirect formulation), thereby ensuring better scalability to problems with longer prediction horizons. The key insight here is that by assuming the system does not change over the prediction horizon, we can construct the full horizon using shorter trajectories. Each prediction trajectory acts as the initialization trajectory for its subsequent segment.

The shorter trajectories used in this formulation necessitate a change in the training data matrix definitions. In Section II, \( T_0 \) dictates the training length, where the conditions \( T_0 \geq (m + 1)(T_{ini} + N + n) - 1 \) and \( T_{ini} \geq \ell \) were imposed. For the segmented form, we replace \( T_0 \) with \( T_n \). Since each segment is at most of length \( T_{ini} \), rather than \( N \) (the final segment may be shorter), we now impose \( T_n \geq (m + 1)(2T_{ini} + n) - 1 \). Notably, for \( T_{ini} < N \) a shorter training period is then sufficient.

The updated training sequences are then defined as \( u_{ \alpha_i} = [u_1^T, \ldots, u_{T_n}^T] \in \mathbb{R}^{mT_n} \) and \( y_{ \alpha_i} = [y_1^T, \ldots, y_{T_n}^T] \in \mathbb{R}^{pT_n} \) and the associated Hankel matrices are defined as

\[
\begin{bmatrix}
U_{ \alpha_i}
\end{bmatrix} := \mathcal{H}_{2T_{ini}} (u_{ \alpha_i}) \\
\begin{bmatrix}
Y_{ \alpha_i}
\end{bmatrix} := \mathcal{H}_{2T_{ini}} (y_{ \alpha_i})
\]

with \( U_{ \alpha_i} \in \mathbb{R}^{mT_{ini} \times (T_n - 2T_{ini} + 1)} \) and \( Y_{ \alpha_i} \in \mathbb{R}^{pT_{ini} \times (T_n - 2T_{ini} + 1)} \).

The trajectories \( u_f \) and \( y_f \) are partitioned into \( F \) segments given as \( [u_f^T, \ldots, u_f^F]^T = u_f^F \) and \( [y_f^T, \ldots, y_f^F]^T = y_f \) respectively, where \( u_f^i \in \mathbb{R}^{mT_{ini}} \) and \( y_f^i \in \mathbb{R}^{pT_{ini}} \), \( \forall i \in \{1, \ldots, F - 1\} \), and the final segments \( u_f^F \in \mathbb{R}^{m(N - (F - 1)T_{ini})} \) and \( y_f^F \in \mathbb{R}^{p(N - (F - 1)T_{ini})} \). For notational brevity, \( u_{ini} \) and \( y_{ini} \) are replaced by \( u_{f0} \) and \( y_{f0} \), respectively.

A diagram illustrating the segmentation concept is shown in Fig. 1 for a prediction trajectory divided into three segments. The indirect approach can now be reformulated. The new multistep predictor matrix, denoted \( P_s^* \), can be found as

\[
P_s^* = Y_{ \beta_f} \begin{bmatrix}
U_{ \alpha_i}
\end{bmatrix}^\dagger.
\]

As in the unsegmented case, \( P_s^* \) can be partitioned and each of the \( F \) segments of the prediction trajectory can then be represented as

\[
y_f = \begin{bmatrix}
P_{s1}^* \\
P_{s2}^* \\
\vdots \\
P_{sF}^*
\end{bmatrix} \begin{bmatrix}
u_{f0} \\
u_{f1} \\
\vdots \\
u_{fF-1}
\end{bmatrix} \forall i \in \{1, \ldots, F\}.
\]

If the length \( N \) of the desired trajectory is not a multiple of \( T_{ini} \), i.e., if \( T_{ini}F \) is longer than \( N \), the final \( T_{ini}F - N \) terms of the final segment \( F \) can be ignored.

To observe the structure of the predictor over the full horizon in the same form as (11), the segments can be stacked and rearranged, resulting in

\[
\begin{bmatrix}
y_{f1} \\
y_{f2} \\
\vdots \\
y_{fF-1}
\end{bmatrix} = \begin{bmatrix}
u_{f1} \\
u_{f2} \\
\vdots \\
u_{fF-1}
\end{bmatrix} + \begin{bmatrix}
I_F \otimes P_{s1}^* \\
I_F \otimes P_{s2}^* \\
\vdots \\
I_F \otimes P_{sF}^*
\end{bmatrix} \begin{bmatrix}
u_{f0} \\
u_{f1} \\
\vdots \\
u_{fF-1}
\end{bmatrix}. 
\]

This can be represented more concisely as

\[
y_f = \begin{bmatrix}
\Phi_1 \\
\Phi_2 \\
\Phi_3
\end{bmatrix} \begin{bmatrix}
u_{ini} \\
y_{ini} \\
u_f
\end{bmatrix}
\]

where \( \Phi_1, \Phi_2, \Phi_3 \) are the segmented counterparts of \( P_{s1}^* \), \( P_{s2}^* \), and \( P_{s3}^* \), with the same dimensions. In this form, \( \Phi_3 \) is
lower block triangular, given as

\[
\begin{bmatrix}
P_{s_3}^* & 0 & \cdots & 0 \\
0 & P_{s_2}^* + P_{s_1}^* & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & P_{s_3}^* + P_{s_2}^* + P_{s_1}^*
\end{bmatrix}.
\] (17)

Notably, \( \Phi_3 \) is lower-block triangular, implying that any output segment cannot be influenced by inputs from future segments. Although this does not ensure causality within segments, since outputs can be influenced by future inputs from the same segment, these can be at most \( T_{\text{ini}} \) samples ahead. In contrast, without segmentation outputs can be influenced by inputs from the full horizon. The impact of the restructuring on \( \Phi_3 \) is shown in the numerical example of Section III.

Next, the segmented structure is translated to the direct formulation in the same manner as Section II-A. Equation (4) can be replaced by the following:

\[
\begin{bmatrix}
U_{\alpha_i} \\
U_{\beta_i} \\
Y_{\beta_i}
\end{bmatrix} \cdot g_i = \begin{bmatrix}
u_{f_{i-1}} \\
u_f \\
y_{f_{i-1}} \\
y_f
\end{bmatrix} \quad \forall i \in \{1, \ldots, F\}
\] (18)

where \( g_i \in \mathbb{R}^{T_{\text{ini}}+1}, \forall i \in \{1, \ldots, F\} \).

Using (18) to predict future input and output trajectories of the system, we can formulate a predictive controller. Here, we seek to minimize a cost function given as \( V_{\alpha}(\cdot) \) along with the regularization term given as \( \| (I - I_F \otimes \Pi_i) g \|_2^2 \), where \( \otimes \) denotes the Kronecker product and \( I_j \) denotes the identity matrix of size \( j \times j \). The objective is then

\[
\min_{g_{i_1}, \ldots, g_{i_F}} \sum_{i=1}^{F} V_{\alpha}(g_i) + \| (I - I_F \otimes \Pi_i) g \|_2^2.
\] (19)

The constraints are given as

\[
\begin{bmatrix}
U_{\alpha_i} \\
Y_{\alpha_i}
\end{bmatrix} \cdot g_i = \begin{bmatrix}
u_{f_0} \\
y_{f_0}
\end{bmatrix} \quad (20)
\]

\[
\begin{bmatrix}
0 \\
0 \\
0 \end{bmatrix} \cdot g_i \in U, \forall i \in \{1, \ldots, F\}
\] (21)

\[
\begin{bmatrix}
U_{\beta_i} \\
Y_{\beta_i}
\end{bmatrix} \cdot g_{i-1} = \begin{bmatrix}
u_{f_0} \\
y_{f_0}
\end{bmatrix} \quad \forall i \in \{2, \ldots, F\}
\] (22)

\[
\begin{bmatrix}
U_{\beta_i}, g_i \\
Y_{\beta_i}, g_i
\end{bmatrix} \in \mathcal{U}, \forall i \in \{1, \ldots, F\}
\] (23)

where \( \mathcal{U} \) denotes a column of zeros of length \( a \). No penalty on the input is included. Since the length of \( g \) in the unsegmented form is at least \( m(T_{\text{ini}} + N + n) + n \) and the length of \( [g_1, \ldots, g_F] \) for the segmented formulation is at least \( F(m(T_{\text{ini}} + n) + n) \), the number of decision variables increases with segmentation. Nonetheless, the block-diagonal structure of the Hessian matrix in the segmented problem implies that benefits can be achieved through the use of sparse quadratic programming solvers, leading to a linear increase in computational time with increasing horizon length. In Section III, this is shown empirically for the segmented and unsegmented forms.

III. ILLUSTRATIVE EXAMPLE: TWO-MASS SYSTEM

A. System Description

A two-mass-spring-damper example is used to illustrate the performance of the segmented predictive controller compared with the unsegmented version. The code needed to reproduce these examples is available on Code Ocean. The system comprises two masses, two springs and two dampers, and is described in the following equations:

\[
\dot{x}(t) = Ax(t) + B (u(t) + d(t))
\] (24)

\[
y(t) = Cx(t) + v(t)
\] (25)

where \( x = (x_1, x_2, \dot{x}_1, \dot{x}_2) \) with \( x_1 \) and \( x_2 \) representing the displacement of masses \( m_1 \) and \( m_2 \), respectively, and their corresponding velocities (shown in Fig. 2). Then, \( u(t) \) is the input force applied to the mass \( m_1 \) and \( y(t) \) is the measured displacement of \( m_2 \). An additional disturbance \( d(t) \) can be applied to \( m_1 \) as well as measurement noise \( v(t) \). The observation matrix \( C := [0 \ 1 \ 0 \ 0] \), while the parameter matrices \( A \) and \( B \) are given as

\[
A := \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
m_1 & m_1 & m_1 & m_1 \\
-k_2 & -k_2 & -k_2 & -k_2
\end{bmatrix}
\] (26)

\[
B := \begin{bmatrix}
0 \\
0 \\
1 \\
0
\end{bmatrix}
\]

where the masses are defined \( m_1 = 0.5 \) and \( m_2 = 1.5 \), the spring constants are defined as \( k_1 = 2 \) and \( k_2 = 2 \) and the damping constants are defined as \( c_1 = 1 \) and \( c_2 = 1 \). The system is shown in Fig. 2.

B. Problem Formulation

We investigate a case study using this system whereby we seek to control the displacement \( y \) to track a set-point \( y_{sp} \), by calculating an input trajectory \( u \) using a data-driven predictive controller. We adopt a direct data-driven formulation and compare segmented and unsegmented versions of the strategy in scenarios with different realizations of a time-varying unmeasured disturbance \( d \) applied to \( m_1 \) and different realizations of measurement noise \( v \). A one-second sample time
is used for the predictive controller, with input and disturbance signals held constant for the duration of the sample. The input force is constrained to the interval \([-1, 1]\) and results are compiled from a 100-s run.

C. Prioritized Objective Formulation

To handle the regularization, relaxation and set-point deviation penalties, a prioritized framework is used as described in [19]. The tracking problem from Section III-B is solved in two stages whereby a feasibility stage is followed by a set-point deviation minimization stage. The first optimization minimizes the initialization slacks given as \(\varepsilon_{f_i} \in \mathbb{R}^{T_{\text{ini}}}, \forall i \in \{1, \ldots, F\}\). The objective and constraints of this linear problem are defined as follows:

\[
J_1^* := \min_{g_1, \ldots, g_F, \varepsilon_{f_1}, \ldots, \varepsilon_{f_F}} \sum_{i=1}^{F} \sum_{j=1}^{T_{\text{ini}}} \varepsilon_{f_i,j} \quad \text{s.t.} \quad u_{\text{ini}}, g_1 = u_{\text{fo}} \quad (27)
\]

\[
\begin{bmatrix}
Y_{\alpha_i} \\
-Y_{\alpha_i}
\end{bmatrix} g_i - \begin{bmatrix}
\varepsilon_{f_i} \\
\varepsilon_{f_i}\end{bmatrix} \leq \begin{bmatrix}
y_{\text{fo}} \\
-y_{\text{fo}}
\end{bmatrix} \quad (28)
\]

\[
\begin{bmatrix}
-U_{\beta_i} \\
-Y_{\alpha_i}
\end{bmatrix} \begin{bmatrix}
g_{i-1} \\
g_i
\end{bmatrix} = \mathbf{0}_{T_{\text{ini}}} \quad \forall i \in \{2, \ldots, F\} \quad (29)
\]

\[
\begin{bmatrix}
Y_{\beta_i} \\
-Y_{\alpha_i}
\end{bmatrix} \begin{bmatrix}
g_{i-1} \\
g_i
\end{bmatrix} - \varepsilon_{f_i} \leq \mathbf{0}_{T_{\text{ini}}} \quad \forall i \in \{2, \ldots, F\} \quad (30)
\]

\[
\begin{bmatrix}
Y_{\beta_i} \\
-Y_{\alpha_i}
\end{bmatrix} \begin{bmatrix}
g_{i-1} \\
g_i
\end{bmatrix} - \varepsilon_{f_i} \leq \mathbf{0}_{T_{\text{ini}}} \quad \forall i \in \{2, \ldots, F\} \quad (31)
\]

\[
\begin{bmatrix}
Y_{\beta_i} \\
-Y_{\alpha_i}
\end{bmatrix} \begin{bmatrix}
g_{i-1} \\
g_i
\end{bmatrix} - \varepsilon_{f_i} \leq \mathbf{0}_{T_{\text{ini}}} \quad \forall i \in \{2, \ldots, F\} \quad (32)
\]

\[
\begin{bmatrix}
Y_{\beta_i} \\
-Y_{\alpha_i}
\end{bmatrix} \begin{bmatrix}
g_{i-1} \\
g_i
\end{bmatrix} - \varepsilon_{f_i} \leq \mathbf{0}_{T_{\text{ini}}} \quad \forall i \in \{2, \ldots, F\} \quad (33)
\]

\[
\begin{bmatrix}
Y_{\beta_i} \\
-Y_{\alpha_i}
\end{bmatrix} \begin{bmatrix}
g_{i-1} \\
g_i
\end{bmatrix} - \varepsilon_{f_i} \leq \mathbf{0}_{T_{\text{ini}}} \quad \forall i \in \{2, \ldots, F\} \quad (34)
\]

\[
\begin{bmatrix}
I_F \otimes Y_{\beta_i} \\
-I_F \otimes Y_{\beta_i}
\end{bmatrix} \begin{bmatrix}
g_i \\
g_f
\end{bmatrix} + \begin{bmatrix}
\varepsilon_{y} \\
\varepsilon_{y}
\end{bmatrix} \leq \begin{bmatrix}
y_{\text{sp}} \\
-y_{\text{sp}}
\end{bmatrix} \quad (35)
\]

\[
\begin{bmatrix}
I_F \otimes Y_{\beta_i} \\
-I_F \otimes Y_{\beta_i}
\end{bmatrix} \begin{bmatrix}
g_{i-1} \\
g_i
\end{bmatrix} - \varepsilon_{f_i} \leq \mathbf{0}_{T_{\text{ini}}} \quad \forall i \in \{2, \ldots, F\} \quad (36)
\]

The second optimization objective is composed of a penalty on the sum of the absolute deviation of the output from the set point, given as \(\varepsilon_{y} \in \mathbb{R}^{N}\), and a regularization penalty on \(g\) with the relative weight between the two penalties set by choice of \(\lambda_g > 0\). The quadratic objective and linear constraints of this problem are given as

\[
J_2^* := \min_{\varepsilon_{f_1}, \ldots, \varepsilon_{f_F}} \sum_{i=1}^{N} \varepsilon_{f_i} + \lambda_g \|g - (I - I_F \otimes \Pi_y) g\|_2^2 \quad (37)
\]

\[
\sum_{i=1}^{F} \sum_{j=1}^{T_{\text{ini}}} \varepsilon_{f_i,j} \leq J_1^* \quad (38)
\]

\[
\begin{bmatrix}
I_F \otimes Y_{\beta_i} \\
-I_F \otimes Y_{\beta_i}
\end{bmatrix} \begin{bmatrix}
g_i \\
g_f
\end{bmatrix} + \begin{bmatrix}
\varepsilon_{y} \\
\varepsilon_{y}
\end{bmatrix} \leq \begin{bmatrix}
y_{\text{sp}} \\
-y_{\text{sp}}
\end{bmatrix} \quad (39)
\]

For the unsegmented formulation, the same optimization is used with \(F = 1\), constraints (30)-(32) omitted and \(\alpha_s\) and \(\beta_s\) subscripts replaced by \(\alpha\) and \(\beta\), respectively.

To generate training data, the system was simulated in open-loop with the input force \(u\) varied at 10-s intervals by drawing a sample from a uniform distribution in the interval \([-1, 1, 1\) to generate a persistently exciting training set of input and output data. A disturbance signal, which acts as the unmeasured disturbance, was also generated and applied to the system. This disturbance force was composed of a sinusoidal component of amplitude 0.2 N, a bias of 0.2 N, and a frequency of 0.01 Hz added to a uniformly distributed random noise component drawn from the interval \([-0.15, 0.15\] N]. The measurement noise was drawn from a normal distribution with zero mean and a standard deviation of 0.1 m.

D. Performance Analysis: Disturbance and Measurement Noise

To illustrate the behavioral differences between the segmented and unsegmented formulations, the system is simulated for 100 different realizations of the stochastic disturbance term followed by 100 different realizations of the measurement noise term for several prediction horizon lengths. The two formulations are compared by observing the set-point deviations achieved for each formulation across the different horizon choices for each noise and disturbance realization. The scenarios with unmeasured disturbance present are first shown, followed by the scenarios with measurement noise present. The benefits of the segmented approach are then discussed.

1) Tuning: In all cases, the regularization weight \(\lambda_g = 0.5\) and initialization length \(T_{\text{ini}} = 5\). It should be noted that, though the outcomes were not sensitive to these choices in this case study, parameter selection is not necessarily a trivial task, as indicated in [20]. In this work, the choices were made by evaluating the set-point tracking performance of both segmented and unsegmented formulations for different parameter values by trial and error. A more rigorous tuning approach that does not require a model would be needed for real-world implementation purposes. This is considered outside the scope of this work, but automated tuning methods have been developed with this in mind [21].

2) Disturbance Realizations: In Fig. 3, the results for each realization of the disturbance are shown in the form of a box plot in which results are grouped in terms of horizon length. In the plot, a notch is centered on the median and the lower and upper sides of the boxes themselves show the lower and upper quartiles, respectively.

It can be seen that the segmented formulation tends to outperform the unsegmented version, with the performance more pronounced in longer horizons. The segmented formulation reduced the impact of the disturbance on the optimization problem, as will be further presented in Section III-D4, enabling better tracking accuracy. Notably, the performance of the segmented formulation is unaffected by the choice of the prediction horizon. The difference in performance is illustrated in Fig. 4 in which the output of the system is plotted for each realization with \(N = 100\) for both segmented and unsegmented. The segmented outputs are generally grouped more closely to the set point than those of the unsegmented formulation.
Fig. 3. Performance of segmented and unsegmented formulations for different realizations of the unmeasured disturbance across different prediction horizons.

Fig. 5. Performance of segmented and unsegmented formulations for different realizations of the measurement noise across different prediction horizons.

TABLE I

| Outperforming scenarios (% of scenarios) | Average performance improvement (% error reduction) |
|----------------------------------------|---------------------------------------------------|
| N = 10                                  | 69%                                               | 13%                                               |
| N = 20                                  | 74%                                               | 17%                                               |
| N = 40                                  | 83%                                               | 28%                                               |
| N = 60                                  | 88%                                               | 36%                                               |
| N = 80                                  | 85%                                               | 34%                                               |
| N = 100                                 | 86%                                               | 32%                                               |

By comparing the outcomes of each disturbance realization individually, the results are summarized in Table I. Table I shows two metrics. The first, labeled Outperforming scenarios, is the percentage of realizations for which segmentation led to a lower set-point error, with set-point error defined as the sum of the distances of the output to the set-point across the full simulation. The second, average performance improvement, shows the average percentage set-point error reduction achieved through segmentation across all disturbance realizations. For longer horizons, segmentation improved performance in approximately 85% of cases, with an average improvement of over 30%.

3) Measurement Noise Realizations: Next, the results for different realizations of the measurement noise are shown in Fig. 5. Once again, segmentation leads to performance benefits in terms of set-point error reduction and the segmented performance is consistent across prediction horizon lengths.

For longer horizons, segmentation led to a performance improvement in over 85% of the simulated cases with an average reduction in set-point error of over 30%. These results are summarized in Table II.

4) Benefits of Segmentation: To understand the performance benefits of segmentation, it is instructive to consider the discussion of Section II-B in the context of the case study results. It was mentioned in Section II-B that causality entails a block-diagonal structure on $P_3^*$ and $\Phi_1$ for unsegmented and segmented formulations, respectively. Though these matrices are related to an SPC formulation, rather than the direct DeePC formulation, the latter is regularized to align with the former, making their structures relevant to DeePC. By visualizing the values of these square matrices taken from the simulated examples in the form of heat maps, the impact of the disturbance can be observed on each. Such heat maps are shown in Fig. 6 for unsegmented and segmented formulations, with and without disturbance for a case with $N = 30$. Without disturbance, $P_3^*$ and $\Phi_1$ are very similar, as shown in Fig. 6(a)–(d). In the disturbing cases, $\Phi_1$ taken from the segmented case [Fig. 6(d)] is similar to the undisturbed cases, whereas the unsegmented cases shown in Fig. 6(c) are quite different. Of particular note is the presence of nonzero terms in the upper triangular portion of the matrix.

E. Performance Analysis: Computation Time

As the problem structure of the segmented formulation is different from the unsegmented formulation, it is worth considering the computational time required for each. Without segmentation, the number of decision variables for the case study is $2N + 30$. With segmentation, this becomes $8N$, which is larger than the unsegmented form for $N > 5$. Despite the increase in decision variables, the block-diagonal structure of the Hessian in the segmented problem allows for the problem
Fig. 6. Heat maps showing the composition of (a) $P_3^*$ unsegmented, no disturbance; (b) $\Phi_3$ segmented, no disturbance; and (c) $P_3^*$ unsegmented, with disturbance; (d) $\Phi_3$ segmented, with disturbance. Notice that the upper triangular portion of (c) is nonzero.

Fig. 7. Computational time for segmented and unsegmented formulations.

sparsity to be exploited. This was examined by observing the computation time needed to solve the second-level quadratic optimization with various prediction horizons. Parameter choices of $\lambda_g = 0.5$ and $T_{ini} = 5$ were used for all cases. All scenarios were computed using the quadprog function, with the interior-point-convex algorithm, using the sparse setting for the internal linear solver in MATLAB 2020b on a 2.9-GHz processor, with the results shown in the log–log plot in Fig. 7. For shorter horizons, the solution time of the segmented formulation is slightly higher, while the converse is true for longer horizons ($N \geq 60$). Indeed the computation time of the segmented formulation increases approximately linearly with increasing horizon length, while the computation time of the unsegmented formulation is nonlinear.

IV. APPLICATION TO BUILDING ENERGY MANAGEMENT

A. Building Energy Management Challenge

An active area of research in recent times concerns the use of predictive control for building energy management. Modern energy systems require more flexibility to handle the combined influences of increased renewable generation and increased electrification of heating and transport. Making use of buildings as active, flexible components in such an energy landscape is a key requirement in global decarbonization efforts [22]. Despite the pressing need for advanced control technologies, the underlying model complexity of a building and the wide variation in building designs has led to the model development process acting as a significant barrier to technology uptake [23]. Consequently, data-driven predictive control techniques have recently received attention for the application of building energy management [24].

The segmented formulation proposed in this article is suited to this domain. Diurnal building usage and energy tariff patterns, along with the slow thermal dynamics of well-insulated buildings, make longer prediction horizons advantageous. Furthermore, many disturbances tend to impact the energy demand of a building. Measurements of these may not be available. External temperature, solar radiation, and internal gains will influence the building’s behavior, potentially corrupting the ability of a data-driven algorithm to identify input–output behavior from a given dataset. A simulated case study was carried out to investigate the performance of the segmented formulation in this setting, using state-of-the-art EnergyPlus [25] building simulation software and comparing the performance of the unsegmented and segmented formulations.

B. Building Simulation Environment

A popular technique for building thermal simulation is to represent the structure as a resistance–capacitance ($RC$) network [26], particularly when knowledge of the physical composition of the building is available. The materials making up the walls, floors, ceilings, and windows are represented as configurations of resistances and capacitances whereby current flows through the circuit are analogous to heat flows through building components. Here, an EnergyPlus model of a six-room apartment was created based on standard building materials and thermal behavior characteristics taken from the Tabula Webtool [27], and the underlying thermal model was extracted using the building $RC$ modeling (BRCM) toolbox [28]. This resulting thermal model can be represented as a 102-state, linear, state–space system with six inputs (radiators in each room) and six outputs (the room temperatures). A schematic of the apartment layout can be seen in Fig. 8.

The building model is influenced by the ambient temperature and solar irradiance from different orientations. For this, weather data from a London-based weather station was obtained from the Centre for Environmental Data Analysis (CEDA) archive [29]. The occupancy profile used in the
simulation was taken from the occupancy-integrated archetype approach of [30]. During occupied periods, a comfort set-point band between 20 °C and 22 °C was desired, while in unoccupied times, the temperatures were allowed to vary between 16 °C and 26 °C. The input in each room was constrained between 0 and the upper heat supply limit of the radiator in the room. The radiators were sized to emit a maximum of 100 W per m² of floor area. The simulation ran with a 10-s sample time.

A separate data-driven predictive controller in each room with a sample time of 15 min was used to dictate the heat flow from the radiator to the room. The future set-point requirements were known to the controllers, as well as the current and previous room temperature and heat flow measurements. No measurements or forecasts of the weather were available to the controllers. A training period was carried out in which the radiators attempted to track a set point varying between the upper and lower set-point bounds, using a PI controller. Note that this approach implies that the comfort set-point bounds should not be violated during the training period. The length of the training period depended on the formulation used (segmented or unsegmented) and the prediction horizon chosen for a particular scenario.

A set of scenarios were designed to compare the performance of the segmented and unsegmented formulations using different prediction horizons in this simulation environment. For these scenarios, we seek to minimize the deviation of the room temperatures outside the comfort bounds at a minimal cost. It was assumed that a heat pump supplies heat to the radiators with a coefficient of performance (COP) of 2.5, with electricity purchased via a time-varying tariff. For this, wholesale electricity price data was used with the octopus agile pricing tariff mechanism applied [31].

The formulation of Section III was modified slightly to incorporate an energy cost in the objectives. Once again, a prioritized framework was used, with the slack variables minimized first, followed by discomfort minimization in a second optimization, before finally minimizing the energy cost. The first two optimization levels are formulated as in (27)–(39). The financial cost is considered in the third optimization problem. The predicted electricity price for the period from \( k+1 \) to \( k+N \) is given as \( C_{\text{elec}} = (c[k+1], \ldots, c[k+N]) \in \mathbb{R}^N \). The predicted electricity cost for heat pump consumption associated with the room over the prediction horizon was then included in the third-level objective as follows:

\[
\min_{g_{\epsilon_1}, \ldots, g_{\epsilon_N}, c_{\text{elec}} \in R^F} \quad \eta C_{\text{elec}} I F \otimes U_{\beta_i} + \lambda_g \sum_{i=1}^{F} |g_{\epsilon_i}| \quad (40)
\]

where \( \eta \) denotes the heat pump COP.

The constraints for this third-level problem are the same as for the second-level problem, with an additional constraint needed to enforce the optimal comfort performance, given as

\[
\|c_{\epsilon_y}\|_1 \leq \|c_{\epsilon_y}^*\|_1 \quad (41)
\]

where \( c_{\epsilon_y}^* \) is the optimal \( c_{\epsilon_y} \) computed in the second-level optimization problem. A decentralized architecture was used, in which each room has a separate controller and no communication between controllers occurs.

C. Performance Analysis of Data-Driven Controllers

Simulations were carried out to analyze the performance of the controllers for a three-week period using different prediction horizon lengths with the segmented and unsegmented formulations. Prediction horizons from ten samples (2.5 h) to 95 samples (just under one day) are investigated. In all cases, \( T_{\text{ini}} = 5 \) and \( \lambda_g = 1 \) as these values were found to perform best for both segmented and unsegmented formulations. The results are summarized in Fig. 9, where the total heating cost for the apartment is plotted on the y-axis and a discomfort metric is plotted on the x-axis. This discomfort metric is defined as the summation of absolute deviations from the comfort temperature set-point band in Kelvin (K), summed across each zone, scaled appropriately to achieve units of K · hr.

The first noticeable feature of the results is that the segmented results for all horizon choices are closely grouped together, whereas the unsegmented results vary quite a lot in terms of both cost and comfort. Furthermore, in terms of comfort, all unsegmented cases with prediction horizons longer than 2.5 h fare significantly worse than the segmented cases. Although the unsegmented formulation achieves a lower cost than the segmented formulation for horizons longer than 2.5 h, this comes at the expense of more discomfort. Since the objectives are prioritized and the comfort objective has a higher priority than the financial one, such behavior is not indicative of a tradeoff between objectives, it is indicative of poor control performance of the unsegmented formulation.

Table III provides the underlying values of the discomfort results. An interesting trend that can be seen in the segmented formulation is that longer prediction horizons lead to improved comfort. This is expected since preheating can be better exploited with longer predictions. Without segmentation, this benefit is not realized as control performance is compromised with longer predictions.

The heating cost associated with each scenario is provided in Table IV. The segmented costs are similar for all horizon choices. For the unsegmented cases, low costs are achieved in some cases, for example, \( N = 20 \) and \( N = 40 \), however, these costs come at the expense of comfort. Only the \( N = 10 \) unsegmented formulation is competitive with the segmented formulation.

A one-week window of the average apartment temperatures using \( N = 95 \) with the segmented and unsegmented formulations is plotted in Fig. 10 to illustrate the differing

\[ \text{Fig. 9. Cost versus comfort objectives for different prediction horizons using segmented and unsegmented formulations.} \]
TABLE III

|                  | $N = 10$ | $N = 20$ | $N = 40$ | $N = 60$ | $N = 95$ |
|------------------|---------|---------|---------|---------|---------|
| Unsegmented (K·hr) | 89.5    | 576.4   | 484.7   | 561.7   | 259.7   |
| Segmented (K·hr)  | 85.3    | 82.2    | 80.6    | 75.7    | 71.2    |

TABLE IV

|                  | $N = 10$ | $N = 20$ | $N = 40$ | $N = 60$ | $N = 95$ |
|------------------|---------|---------|---------|---------|---------|
| Unsegmented (£)  | 63.1    | 61.2    | 61.4    | 62.6    | 66.7    |
| Segmented (£)    | 63.5    | 63.1    | 63.3    | 63.4    | 63.4    |

control performance. The electricity price for the same period is also shown. The unsegmented formulation overheat the apartment during the unoccupied periods compared with the segmented formulation. In both formulations, the controllers tend to preheat the apartment in advance of an electricity price spike, however, without segmentation, the rooms are held at a higher temperature than is necessary.

As in the examples of Section III, the performance of the unsegmented strategy breaks down as the prediction horizon length increases, while the segmented formulation is more consistent in a wider range of operational strategies.

V. CONCLUSION

This article proposes a restructuring of a data-driven predictive control formulation for linear systems with unmeasured disturbances and noise. The proposed formulation modifies an existing DeePC approach by segmenting the prediction horizon. By doing so, the formulation performs better than the unsegmented formulation in the presence of unmeasured disturbance, particularly for longer prediction horizons.

The method was analyzed here first using a set of case studies based on a two-mass-spring-damper system. Under various disturbance and noise realizations, the segmented formulation outperformed the unsegmented formulation in terms of set-point tracking when disturbances were present, particularly with longer prediction horizons. The computation time associated with the proposed segmented formulation scales linearly with horizon length, improving on the time increase observed for the unsegmented formulation.

The segmented formulation was applied to a building energy management case study to demonstrate the importance of these performance characteristics in a more realistic setting, using a state-of-the-art building simulation environment with realistic weather profiles acting as unmeasured disturbances. The segmented formulation performed consistently with horizon length variation in terms of occupant comfort levels and energy consumption. Without segmentation, the comfort minimization performance of the controller was significantly worse for prediction horizons longer than ten samples. For the scenario with a one-day-ahead prediction horizon, the segmented approach reduced discomfort by 72% and cost by 5% relative to the unsegmented approach.

Further work is needed to assess the impact of segmentation on the various extensions of the data-predictive controller that have been developed, such as formulations with robustness guarantees and formulations, for time-varying parameters and nonlinear systems. Additionally, methods for offset-free control in the presence of disturbance would also be beneficial to the data-driven context. Computational efficiency and hyper-parameter selection are also key aspects that require further focus to ensure algorithms are tailored appropriately to a given context.

REFERENCES

[1] E. T. Maddalena, Y. Lian, and C. N. Jones, “Data-driven methods for building control—A review and promising future directions,” Control Eng. Pract., vol. 95, Feb. 2020, Art. no. 104211.

[2] V. Bhattacharyya, A. F. Canosa, and B. HomChaudhuri, “Fast data-driven model predictive control strategy for connected and automated vehicles,” ASME Lett. Dyn. Syst. Control, vol. 1, no. 4, pp. 1–5, Oct. 2021.

[3] J. Wang, S. Li, H. Chen, Y. Yuan, and Y. Huang, “Data-driven model predictive control for building climate control: Three case studies on different buildings,” Building Environ., vol. 160, Aug. 2019, Art. no. 106204.

[4] C. De Persis and P. Tesi, “Formulas for data-driven control: Stabilization, optimality, and robustness,” IEEE Trans. Autom. Control, vol. 65, no. 3, pp. 909–924, Mar. 2020.

[5] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor, “A note on persistency of excitation,” Syst. Control Lett., vol. 54, no. 4, pp. 325–329, Apr. 2005.

[6] J. Coulson, J. Lygeros, and F. Dorfler, “Data-enabled predictive control: In the shallows of the DeePC,” in Proc. 18th Eur. Control Conf. (ECC), Jun. 2019, pp. 307–312.

[7] F. Fiedler and S. Lucia, “On the relationship between data-enabled predictive control and subspace predictive control,” 2020, arXiv:2011.13868.

[8] F. Dorfler, J. Coulson, and I. Markovsky, “Bridging direct & indirect data-driven control formulations via regularizations and relaxations,” 2021, arXiv:2101.01277.

[9] H. J. van Waarde, C. De Persis, M. K. Camlibel, and P. Tesi, “Willems’ fundamental lemma for state-space systems and its extension to multiple datasets,” IEEE Control Syst. Lett., vol. 4, no. 3, pp. 602–607, Jul. 2020.

[10] D. A. S. Alpago, F. Dorfler, and J. Lygeros, “An extended Kalman filter for data-enabled predictive control,” IEEE Control Syst. Lett., vol. 4, no. 4, pp. 994–999, Oct. 2020.

[11] J. Berberich, J. Kohler, M. A. Muller, and F. Allgower, “Data-driven model predictive control with stability and robustness guarantees,” IEEE Trans. Autom. Control, vol. 66, no. 4, pp. 1702–1717, Apr. 2021.
[12] J. Coulson, J. Lygeros, and F. Dörfler, “Distributionally robust chance constrained data-enabled predictive control,” 2020, arXiv:2006.01702.

[13] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, “Robust data-enabled predictive control: Tractable formulations and performance guarantees,” 2021, arXiv:2105.07199.

[14] Y. Lian and C. N. Jones, “From system level synthesis to robust closed-loop data-enabled predictive control,” 2021, arXiv:2102.06553.

[15] L. Huang, J. Zhen, J. Lygeros, and F. Dörfler, “Robust data-enabled predictive control: Tractable formulations and performance guarantees,” 2021, arXiv:2106.05740.

[16] J. C. Willems, “The behavioral approach to open and interconnected systems,” IEEE Control Syst., vol. 27, no. 6, pp. 46–99, Dec. 2007.

[17] I. Markovsky and P. Rapisarda, “Data-driven simulation and control,” Int. J. Control, vol. 81, no. 12, pp. 1946–1959, 2008.

[18] S. J. Qin, W. Lin, and L. Ljung, “A novel subspace identification approach with enforced causal models,” Automatica, vol. 41, no. 12, pp. 2043–2053, Dec. 2005.

[19] E. C. Kerrigan and J. M. Maciejowski, “Designing model predictive controllers with prioritised constraints and objectives,” in Proc. IEEE Int. Symp. Comput. Aided Control Syst. Design, Dec. 2002, pp. 33–38.

[20] V. Chinde, Y. Lin, and M. J. Ellis, “Data-enabled predictive control for building HVAC systems,” J. Dyn. Syst., Meas., Control, vol. 144, no. 8, Aug. 2022, Art. no. 081001.

[21] E. O’Dwyer, P. Falugi, N. Shah, and E. Kerrigan, “Robust data-driven predictive control: Tractable formulations and performance guarantees,” 2021, arXiv:2106.05740.

[22] R. Lowes, J. Rosenow, M. Qadrdan, and J. Wu, “Hot stuff: Research and policy principles for heat decarbonisation through smart electrification,” Energy Res. Social Sci., vol. 70, Dec. 2020, Art. no. 101735.

[23] E. Atam and L. Helsen, “Control-oriented thermal modeling of multizone buildings: Methods and issues: Intelligent control of a building system,” IEEE Control Syst. Mag., vol. 36, no. 3, pp. 86–111, Jun. 2016.

[24] A. Korthigamanathan, M. De Rosa, E. Mangina, and D. P. Finn, “Data-driven predictive control for unlocking building energy flexibility: A review,” Renew. Sustain. Energy Rev., vol. 135, Jan. 2021, Art. no. 110120.

[25] DOE. (2017). EnergyPlus | EnergyPlus. [Online]. Available: https://www.energyplus.net/

[26] E. O’Dwyer, L. De Tommasi, K. Kouramas, M. Cychowski, and G. Lightbody, “Modelling and disturbance estimation for model predictive control in building heating systems,” Energy Buildings, vol. 130, pp. 532–545, Oct. 2016.

[27] TABULA. TABULA WebTool. Accessed: Oct. 12, 2020. [Online]. Available: http://webtool.building-typology.eu/#bm

[28] D. Szturzegger, G. Gyalistras, V. Semeraro, M. Morari, and R. S. Smith, “BRCM MATLAB toolbox: Model generation for model predictive building control,” in Proc. Amer. Control Conf., Jun. 2014, pp. 1063–1069.

[29] Met Office. (2006). MIDAS: U.K. Hourly Weather Observation Data. NCAS British Atmospheric Data Centre. [Online]. Available: https://catalogue.ceda.ac.uk/uuid/916ac4bbc46f7685ae9a5e10451bae7c

[30] G. Buttitta, W. J. N. Turner, O. Neu, and D. P. Finn, “Development of occupancy-integrated archetypes: Use of data mining clustering techniques to embed occupant behaviour profiles in archetypes,” Energy Buildings, vol. 198, pp. 84–99, Sep. 2019.

[31] Octopus Energy. (2020). Frequently Asked Questions About Octopus Tracker | Octopus Energy. [Online]. Available: https://octopus. energy/tracker-faqs/

Edward O’Dwyer received the B.Eng. degree in energy engineering from the University College Cork, Cork, Ireland, in 2012, and the Ph.D. degree in control engineering from the University College Cork, in 2016. He is currently a Research Associate with the Sargent Centre for Process Systems Engineering, Department of Chemical Engineering, Imperial College London, London, U.K. His research interests include modeling, control, and optimization of buildings, energy systems, and smart cities, with an emphasis on decarbonization and sustainability.

Eric C. Kerrigan (Senior Member, IEEE) is currently a Professor of control and optimization with the Imperial College London, London, U.K. His research interests include the design of efficient numerical methods and computing architectures for solving optimal control and estimation problems in real time, with applications in the design of aerospace, renewable energy, and information systems. Dr. Kerrigan is on the IEEE Control Systems Society Board of Governors. He is the Chair of the IPAC Technical Committee on Optimal Control. He is a Senior Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEM TECHNOLOGY and an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and the European Journal of Control.

Paola Falugi (Member, IEEE) received the Ph.D. degree in systems engineering from the University of Bologna, Bologna, Italy, in 2002. She is currently a Research Associate with the Department of Electrical and Electronic Engineering, Imperial College London, London, U.K. She was a Post-Doctoral Researcher with the University of Florence, Florence, Italy, for four years, and Supélec, Gif-sur-Yvette, France, for one year. Her research interests include predictive control and control of constrained nonlinear systems, robust control, system identification, and energy network planning.

Marta Zagorowska received the B.Sc. and M.Sc. degrees in automation and robotics from the AGH University of Science and Technology, Kraków, Poland, in 2012 and 2013, respectively, and the Ph.D. degree from the Imperial College London, London, U.K. in 2020, under the supervision of Prof. Nina Thornhill. Dr. Charlotte Skouren from ABB, Fornebu, Norway, joined the group of Prof. Eric Kerrigan to work on robust control and optimization at the Imperial College London. In 2022, she joined the Automatic Control Laboratory, ETH Zürich, Zürich, Switzerland, to work on the applications of data-driven control methods in industrial processes. Her research interests include the development of efficient optimization and control algorithms and their implementation in industrial applications to bridge the gap between applications and theory.

Nilay Shah is currently a Professor of process systems engineering with the Imperial College London, London, U.K. His research interests include the use of models and process systems engineering techniques to understand and design low-carbon energy and industrial systems.