Online Convex Optimization for Data-Driven Control of Dynamical Systems

MARKO NONHOFF (Graduate Student Member, IEEE), **AND MATTHIAS A. MÜLLER** (Senior Member, IEEE)

Leibniz University Hannover, Institute of Automatic Control, Appelstraße 11, 30167 Hannover, Germany

CORRESPONDING AUTHOR: MARKO NONHOFF (e-mail: nonhoff@irt.uni-hannover.de)

ABSTRACT We propose an algorithm based on online convex optimization for controlling discrete-time linear dynamical systems. The algorithm is data-driven, i.e., does not require a model of the system, and is able to handle a priori unknown and time-varying cost functions. To this end, we make use of a single persistently exciting input-output sequence of the system and results from behavioral systems theory which enable it to handle unknown linear time-invariant systems. Moreover, we consider noisy output feedback instead of full state measurements and allow general economic cost functions. Our analysis of the closed loop reveals that the algorithm is able to achieve sublinear regret, where the measurement noise only adds an additional constant term to the regret upper bound. In order to do so, we derive a data-driven characterization of the steady-state manifold of an unknown system. Moreover, our algorithm is able to asymptotically exactly estimate the measurement noise. The effectiveness and applicational aspects of the proposed method are illustrated by means of a detailed simulation example in thermal control.

INDEX TERMS Data-driven control, linear systems, online optimization, optimal control.

I. INTRODUCTION

This paper considers the problem of controlling an unknown linear time-invariant (LTI) system subject to time-varying and a priori unknown convex cost functions. In particular, we aim to minimize the accumulated cost obtained by our proposed algorithm in closed loop with the unknown system. The main difficulty arises from the fact that the cost functions are time-varying and a priori unknown, i.e., the cost function L_t at time t is only revealed to us at time step $t + 1$. These kind of problems commonly arise in practice, e.g., in power grids due to a priori unknown renewable energy generation and unknown energy consumption [1], in data center cooling [2], or in robotics [3]. Our approach is inspired by online convex optimization (OCO) [4], [5], an online variant of classical numerical optimization. Whereas the classical OCO literature does not consider underlying dynamical systems, it has gained significant interest recently for solving optimal control tasks. Its main advantages include its ability to handle a priori unknown and time-varying cost functions, low computational complexity, and its ability to take constraints on the state and the input of the system into account. OCO-based algorithms have been proposed to control linear dynamical systems [6], [7] subject to process noise [8], [9], constraints [10], [11], or output feedback [12].

Most of the existing OCO-based algorithms in the literature discussed above depend crucially on model knowledge of the system. However, obtaining such a model can be difficult or expensive in certain applications. Hence, in recent years, direct data-based control approaches have received a considerable amount of attention, compare, e.g., [13]. In this work, we employ a result from behavioral systems theory. The so-called fundamental lemma shows that a Hankel matrix consisting of a single persistently exciting input-output trajectory spans the whole vector space of all possible input-output trajectories of an LTI system [14]. This result has recently drawn significant attention and has been applied to solve a variety of control problems, e.g., model predictive control (MPC) [15], [16], state- and output-feedback design [17], [18], [19], [20], [21], and output matching [22]. We combine the fundamental lemma with OCO in order to control dynamical systems subject to time-varying cost functions, where neither the system nor the cost functions are known to the algorithm.

Another closely related line of research is so-called optimal steady-state (OSS) control. Therein, a system is controlled to

the solution of a (possibly time-varying) optimization problem by applying gradient-based feedback and, typically, asymptotic guarantees in the form of stability of the overall system are derived [23], [24]. Again, the main focus in the literature is on model-based control with process noise and output feedback [3], [25], [26]. In [27], a data-driven method for regulating the output of a general nonlinear system, subject to a constant disturbance, to the optimal steady state of a constant cost function is proposed. In particular, the authors leverage a result from zeroth order optimization in order to avoid requiring model knowledge of the controlled system. However, performance is only analyzed in terms of the second moments of the gradients of a smooth approximation of the cost function. Most relevant to this work is [28], where output feedback and unknown systems subject to disturbances are treated by application of the fundamental lemma. To this end, a steady-state map between the input to and the output of the unknown system is estimated using only measured data. However, the cost functions are assumed to be constant and time-variability of the optimization problem is only introduced via time-varying process noise. Moreover, analysis of the closed loop's transient behavior is limited to analysis of contraction with respect to the optimal steady state, but does not consider the transient cost in terms of regret analysis.

The contribution of this work is fivefold. First, we consider an unknown system by leveraging results from data-driven control. Compared to alternative approaches in the literature, we thereby remove the need of a (set-based) model description and of an online estimation process. Second, we extend our previous results from OCO-based control [6], [10] to the case of output feedback instead of full state measurements, which requires considerable adjustments in algorithm design and analysis techniques. Third, we consider noise in the measurement process. In the relevant literature, e.g., [9], [11], [28], the main research focus is on systems subject to process noise, which is typically handled by estimating the process noise using exact measurements and model knowledge. We instead consider only noisy measurements in our theoretical work and leave the combination of both, process and measurement noise, as an interesting topic for future research. We do, however, consider both types of noise in our simulation example. Fourth, we generalize previous work [6], [10] by considering the practically relevant case of *economic* cost functions, i.e., the minimum of the cost functions at each time step need *not* be a steady state of the system. Finally, we derive a new data-driven characterization of the steady-state manifold of an LTI system by leveraging the fundamental lemma. As a main result, our analysis reveals that our proposed algorithm enjoys sublinear regret without access to a system model or exact measurements.

This paper is organized as follows. In Section II, we present the basic notions necessary in our work and discuss the problem of interest. Section III introduces and illustrates our proposed algorithm. In Section IV, we discuss our theoretical findings, in particular a regret analysis of the closed loop and asymptotic convergence of the measurement error estimates.

A numerical simulation example, namely a thermal control problem, illustrates the closed-loop performance and applicational aspects of our algorithm in Section V. Section VI concludes the paper.

We close this section by noting that a preliminary version of parts of this paper was presented at the 2021 60th IEEE Conference on Decision and Control (CDC) [29]. This work extends the previously presented results in three directions. First, we consider measurement noise in this work and study its effect on the derived regret bound, which requires adaptations in both algorithm design and theoretical analysis. We show that measurement noise only leads to an additional *constant* term in the regret bound compared to our previous work. Second, we generalize our work to consider economic cost functions, as discussed above. Third, we remove restrictive assumptions on the steady-state manifold, compare [29, Assumption 1], in order to be able to control a wider class of systems. Moreover, we include a detailed simulation example to illustrate the applicability of our proposed algorithm.

Notation: We denote the set of integer numbers in the interval $[a, b]$ and the set of integer numbers greater than or equal to zero by $\mathbb{I}_{[a,b]}$ and $\mathbb{I}_{\geq 0}$, respectively. For a vector $x \in \mathbb{R}^n$, $\|x\|$ is the euclidean norm and for a matrix $A \in \mathbb{R}^{n \times m}$ the corresponding induced matrix 2-norm is $\|A\|$, whereas its Moore-Penrose-Pseudoinverse is denoted by $A^\dagger$. The identity matrix of size $n \times n$ is given by I_n, $1_n \in \mathbb{R}^n$ denotes the vector of all ones, and $0_n \in \mathbb{R}^n$ is the vector of all zeros. A sequence $\{z_k\}_{k=0}^{N-1}$, $z_k \in \mathbb{R}^n$, induces the Hankel matrix of depth L

$$H_L(z) = \begin{bmatrix} z_0 & z_1 & \cdots & z_{N-L} \\ z_1 & z_2 & \cdots & z_{N-L+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{L-1} & z_L & \cdots & z_{N-1} \end{bmatrix} = \begin{bmatrix} H_L^1(z) \\ H_L^2(z) \\ \vdots \\ H_L^L(z) \end{bmatrix}.$$

We denote a matrix containing a subset of block rows of $H_L(z)$ by

$$H_L^{a:b}(z) = \begin{bmatrix} H_L^a(z) \\ \vdots \\ H_L^b(z) \end{bmatrix}.$$

With a slight abuse of notation, we write z for the sequence itself as well as for the stacked vector of all its components. We denote by $z_{[a:b]} = \begin{bmatrix} z_a^\top & \cdots & z_b^\top \end{bmatrix}^\top$ the stacked vector of a subset of its components. The shift operator σ is defined by $\sigma z = \begin{bmatrix} z_1^\top & \cdots & z_{N-1}^\top \end{bmatrix}^\top$. For matrices A and B, $A \otimes B$ denotes the Kronecker product.

II. SETTING

We consider linear time-invariant (LTI) systems of the form

$$x_{t+1} = Ax_t + Bu_t$$
$$y_t = Cx_t + Du_t$$
$$\tilde{y}_t = y_t + e_t \tag{1}$$

where $x_t \in \mathbb{R}^n$ is the system state, $u_t \in \mathbb{R}^m$ is the system input, $y_t \in \mathbb{R}^p$ is the true system output, $\tilde{y}_t \in \mathbb{R}^p$ is the measured system output, and $e_t \in \mathbb{R}^p$ denotes measurement noise at time instance t. We denote by $z_t = \begin{bmatrix} u_t^\top & y_t^\top \end{bmatrix}^\top$ the stacked input-output pair at time t. The system matrices (A, B, C, D) as well as the noise e_t are unknown and only measurements of u_t and $\tilde{y}_t$ are available to us. We do not impose any assumptions on the measurement noise e. We make the following assumptions on system (1).

Assumption 1: The matrix A is Schur stable, the pair (A, B) is controllable, and the pair (A, C) is observable.

Controllability and Observability are standard assumptions in the literature [28]. Compared to [29], we only consider stable systems because of the additional measurement noise. In this setting, we can estimate the measurement error asymptotically exactly, if the system is stable (compare Lemma 3 below). If the system is not stable, data-based techniques from, e.g., [19], [20] can be used to stabilize the (unknown) system. Our algorithm can then be applied to the prestabilized system. However, some of our theoretical guarantees deteriorate for this approach, compare Remark 2 for more details.

Our goal is to solve the optimal control problem

$$\min_u \quad \sum_{t=0}^T L_t(u_t, y_t) \qquad \text{s.t.} \quad (1), \qquad (2)$$

here the main difficulty arises from the fact that the time-varying cost functions $L_t : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ are a priori unknown. Specifically, we want to find a controller that computes an input u_t at every time instance t which is applied to system (1) and yields performance close to the solution of (2). Only after u_t is applied to system (1), the cost function L_t is revealed, i.e., u_t is computed by the algorithm without knowledge of the current cost function. Then, we measure the noisy output $\tilde{y}_t$ and move to the next time step. As standard in OCO, we do not attempt to solve (2) directly at each time step [4], [5]. Since the cost functions are a priori unknown, optimization would have to be carried out based on the last known cost function L_{t-1}. Then, open-loop optimization will in general not improve the closed-loop performance, due to the time-varying nature of the cost functions. Therefore, we aim to design a computationally efficient algorithm, instead of solving a (potentially large-scale) optimization problem at each step. We denote the solution to (2) in hindsight, i.e., the solution when knowing all cost functions, by $u^* = \{u_t^*\}_{t=0}^T$ and the corresponding system output by $y^* = \{y_t^*\}_{t=0}^T$. As common in OCO, we consider smooth convex cost functions as specified in Assumption 2.

Assumption 2: The cost functions $L_t(z)$ are
- α_z-strongly convex, i.e., there exists $\alpha_z > 0$ such that

$$L_t(z_1) \geq L_t(z_2) + \nabla L_t(z_2)^\top (z_1 - z_2) + \frac{\alpha_z}{2} \|z_1 - z_2\|^2 ,$$

- l_z-smooth, i.e., there exists $l_z > 0$ such that

$$L_t(z_1) \leq L_t(z_2) + \nabla L_t(z_2)^\top (z_1 - z_2) + \frac{l_z}{2} \|z_1 - z_2\|^2 ,$$

- and Lipschitz continuous with Lipschitz constant L_z, i.e., there exists $L_z > 0$ such that

$$\|L_t(z_1) - L_t(z_2)\| \leq L_z \|z_1 - z_2\| ,$$

for all $t \in \mathbb{I}_{\geq 0}$ and any two points $z_1, z_2 \in \mathbb{R}^{m+p}$.

Remark 1: We assume Lipschitz continuity for clarity of exposition of our results, even though l_z-smoothness and Lipschitz continuity cannot be satisfied *globally* simultaneously. However, if u_t and y_t remain within bounded sets for all time, Assumption 2 is satisfied on this bounded set. Moreover, techniques from [6] can be used to avoid assuming Lipschitz continuity. In this case, all triangle inequalities in the proof of Theorem 2 are replaced by Jensen's inequality which entails additional assumptions on the step size and the condition number l_z/α_z of the cost functions. Moreover, changing the regret definition below to $\mathcal{R} = \sum_{t=0}^T \|(u_t, y_t) - (\eta_t, \theta_t)\|$ also removes the necessity to assume Lipschitz continuity of the cost functions. □

Characterizing the solution to (2), i.e., u^* and y^*, for general time-varying cost functions L_t requires optimization or verifying certain dissipativity conditions [30], [31], [32] and is thus computationally expensive. For a priori unknown cost functions as considered in this work, computing u^* and y^* online is impossible altogether. Instead, we adopt a strategy of tracking the a priori unknown time-varying optimal states given by

$$(\eta_t, \theta_t) = \begin{cases} \arg\min_{u,y} & L_t(u, y) \\ \text{s.t.} & x = Ax + Bu , \\ & y = Cx + Du \end{cases}$$

where we define $\zeta_t = \begin{bmatrix} \eta_t^\top & \theta_t^\top \end{bmatrix}^\top$, $\eta_t \in \mathbb{R}^m$ is the optimal steady-state input, and $\theta_t \in \mathbb{R}^p$ is the optimal steady-state output of system (1) at time t. In case of constant convex cost functions L, steady-state operation is optimal [33]; hence, we expect that the proposed strategy yields good performance in many practical applications, in particular in case the cost functions L_t do not change too frequently. Note that the setting considered here includes as a special case our previous works [6], [10], [29], where only strongly convex, smooth cost functions were considered that are each positive definite with respect to some (time-varying) steady state (η_t, θ_t) of the system. Here, we consider more general convex cost functions that do not need to satisfy this requirement. Such cost functions often occur in practice related to some economic considerations, such as minimization of energy cost (compare the example in Section V), which is why such cost functions have been termed *economic* in the context of model predictive control (see, e.g., [34], [35], [36]).

As common in OCO, we analyze our controller's closed-loop performance in terms of regret. In light of our strategy of tracking a priori unknown and time-varying optimal steady states of system (1), we define the regret $\mathcal{R}$ as

$$\mathcal{R} := \sum_{t=0}^T L_t(u_t, y_t) - L_t(\eta_t, \theta_t), \qquad (3)$$

i.e., the accumulated difference between the closed-loop cost of our controller and the optimal steady-state cost in hindsight. The regret $\mathcal{R}$ is a measure of the performance lost due to not knowing the cost functions L_t a priori. In the literature, commonly the goal is to achieve sublinear regret[1], i.e.,

$$\limsup_{T \to \infty} \mathcal{R}/T = \limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} L_t(u_t, y_t) - L_t(\eta_t, \theta_t) \le 0.$$

Hence, if the proposed algorithm achieves sublinear regret, then the closed-loop cost is asymptotically on average no worse than the optimal steady-state cost. Such a performance result is typically also considered in the context of economic model predictive control (MPC), compare, e.g., [33], [35].

Since we do not assume knowledge of the system matrices (A, B, C, D), we assume that we have access to measurement data in the form of a prerecorded input-output sequence $\{u_k^d, y_k^d\}_{k=0}^{N-1}$ and an upper bound on the system order n. Note that we require the true system output as data instead of the (noisy) measured system output. Such data can be obtained in practice when, e.g., the prior data is recorded in a laboratory setting using more accurate measuring instruments than during online operation.

Assumption 3: The output data $y^d = \{y_k^d\}_{k=0}^{N-1}$ is noise free.

Moreover, we assume that the data sequence is persistently exciting as defined in Definition 1.

Definition 1: A signal $\{u_k\}_{k=0}^{N-1}$, $u_k \in \mathbb{R}^m$, is called persistently exciting of order L if $\mathrm{rank}(H_L(u)) = mL$.

This definition allows to characterize all possible system trajectories of (1) using only Hankel matrices of the data sequence. This result was first published in the context of behavioral system theory [14] and can be formulated in the classical state space setting as follows.

Theorem 1 [37, Theorem 3]: Suppose $\{u_k^d, y_k^d\}_{k=0}^{N-1}$ is a trajectory of system (1), where u^d is persistently exciting of order $L + n$ and let Assumption 3 be satisfied. Then, $\{\bar{u}_k, \bar{y}_k\}_{k=0}^{L-1}$ is a trajectory of (1) if and only if there exists $\alpha \in \mathbb{R}^{N-L+1}$ such that

$$\begin{bmatrix} H_L(u^d) \\ H_L(y^d) \end{bmatrix} \alpha = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix}.$$

As discussed above, we aim to track a series of a priori unknown steady states without access to a model of the system. Therefore, a data-driven definition of steady states is given in Definition 2.

Definition 2: An input-output pair (u^s, y^s) is an equilibrium of (1), if the sequence $\{u_k, y_k\}_{k=0}^n$ with $(u_k, y_k) = (u^s, y^s)$ for all $k \in \mathbb{I}_{[0,n]}$ is a trajectory of (1).

Definition 2 states that an input-output pair (u^s, y^s) is an equilibrium of system (1) if and only if a sequence consisting of (u^s, y^s) for at least $n + 1$ consecutive time steps is a trajectory of the system. We make use of Definition 2 and

the prerecorded data sequence to characterize the steady-state manifold of system (1) in Lemma 1.

Lemma 1: Let Assumption 3 be satisfied. Assume that the sequence u^d is persistently exciting of order $2n + 1$. Then, the input-output pair (u^s, y^s) is an equilibrium of (1) if and only if

$$S z^s = \begin{bmatrix} S_u & S_y \end{bmatrix} \begin{bmatrix} u^s \\ y^s \end{bmatrix} = 0,$$

where $S = (H_{n+1} H_{n+1}^\dagger - I_{(m+p)(n+1)}) \begin{bmatrix} \hat{I}_m & 0 \\ 0 & \hat{I}_p \end{bmatrix}$, $H_{n+1} = \begin{bmatrix} H_{n+1}(u^d) \\ H_{n+1}(y^d) \end{bmatrix}$, $\hat{I}_m = 1_{n+1} \otimes I_m$, and $\hat{I}_p = 1_{n+1} \otimes I_p$.

Proof: By Definition 2 and Theorem 1, (u^s, y^s) is a steady state of (1) if and only if there exists $v \in \mathbb{R}^{N-n}$ such that

$$H_{n+1} v = \begin{bmatrix} H_{n+1}(u^d) \\ H_{n+1}(y^d) \end{bmatrix} v = \begin{bmatrix} \hat{I}_m u^s \\ \hat{I}_p y^s \end{bmatrix}. \quad (4)$$

The general solution to this equation is given by

$$v = H_{n+1}^\dagger \begin{bmatrix} \hat{I}_m u^s \\ \hat{I}_p y^s \end{bmatrix} + (I_{N-n} + H_{n+1}^\dagger H_{n+1}) v,'$$

where $v' \in \mathbb{R}^{N-n}$ can be chosen arbitrary. Since the second term on the right-hand side of the above expression is in the nullspace of H_{n+1}, inserting v back into (4) yields

$$\left(H_{n+1} H_{n+1}^\dagger - I_{(m+p)(n+1)} \right) \begin{bmatrix} \hat{I}_m & 0 \\ 0 & \hat{I}_p \end{bmatrix} z^s = 0_{(m+p)(n+1)},$$

which proves the result. ∎

Lemma 1 explicitly defines the steady-state manifold of system (1) only in terms of Hankel matrices of the prerecorded data sequence.

III. ALGORITHM

In this section, we introduce our algorithm. For notational convenience, we define $U = H_{2n+\mu+1}(u^d)$ and $Y = H_{2n+\mu+1}(y^d)$, i.e., the Hankel matrices associated with the system input and output, respectively. In addition, we denote

$$H_\alpha = \begin{bmatrix} U^{1:n} \\ U^{n+1:2n+\mu+1} \\ Y^{1:n} \end{bmatrix}, \text{ and } H_\beta = \begin{bmatrix} U^{1:n} \\ U^{n+\mu+1:2n+\mu+1} \\ Y^{1:n} \\ Y^{n+\mu+1:2n+\mu} \end{bmatrix}.$$

The proposed data-driven OCO scheme is given in Algorithm 1. In the framework described above, at every time instance t, Algorithm 1

1) computes an input u_t via (5)–(11) and applies it to system (1),
2) measures the output $\tilde{y}_t$ and receives the cost function L_t,
3) moves to time step $t + 1$.

Roughly speaking, Algorithm 1 estimates the measurement noise by relying on its own predictions, applies online gradient descent (OGD) to estimate the optimal equilibrium of

Algorithm 1: Data-Driven Output Feedback.

Given step size γ, prediction horizon μ, initialization $\hat{u}_{-1} \in \mathbb{R}^{m(\mu+1)}$, $z_{-1}^s \in \mathbb{R}^{m+p}$, $u_{[-n:-1]}$, $\tilde{y}_{[-n:-1]} - \hat{e}_{[-n:-1]}$, and data (u^d, y^d). At each time t:

if $t = 0$, go to (6)

$$\hat{e}_{t-1} = \tilde{y}_{t-1} - Y^{n+1}(\alpha_{t-1} + \beta_{t-1}) \tag{5}$$

Choose α_t such that

$$H_\alpha \alpha_t = \begin{bmatrix} u_{[t-n:t-1]} \\ \hline \sigma \hat{u}_{t-1} \\ \hline 1_{n+1} \otimes u_{t-1}^s \\ \hline \tilde{y}_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]} \end{bmatrix} \tag{6}$$

$$\hat{z}_t^\mu = \begin{bmatrix} u_{t-1}^s \\ Y^{n+\mu+1}\alpha_t \end{bmatrix} \tag{7}$$

$$z_t^s = \begin{bmatrix} u_t^s \\ y_t^s \end{bmatrix} = \left(I_{m+p} - S^\dagger S\right)\left(\hat{z}_t^\mu - \gamma \nabla L_{t-1}\left(\hat{z}_t^\mu\right)\right) \tag{8}$$

$$\beta_t = \begin{cases} \arg\min_\beta & \|Q\beta\| \\ \text{s.t. } H_\beta \beta = \begin{bmatrix} 0_{mn} \\ 1_{n+1} \otimes u_t^s - U^{n+\mu+1:2n+\mu+1}\alpha_t \\ 0_{pn} \\ 1_n \otimes y_t^s - Y^{n+\mu+1:2n+\mu}\alpha_t \end{bmatrix} \end{cases} \tag{9}$$

$$\hat{u}_t = U^{n+1:n+\mu+1}(\alpha_t + \beta_t) \tag{10}$$

$$u_t = U^{n+1}(\alpha_t + \beta_t) \tag{11}$$

Measure $\tilde{y}_t$ and receive L_t

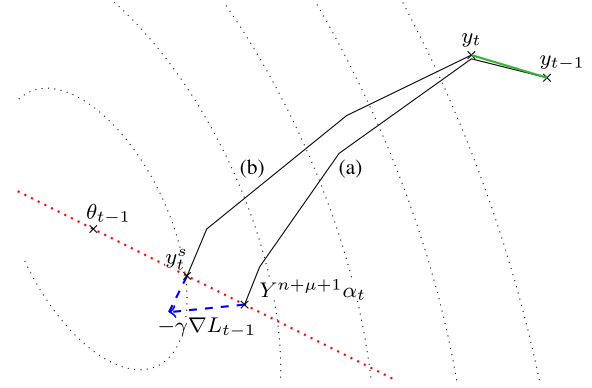Set $t = t + 1$ and go to (5)

FIGURE 1. Schematic illustration of Algorithm 1. The previous cost function is depicted by its level sets (dotted) together with the steady-state manifold (red, dotted). Algorithm 1 predicts the output μ-steps ahead (a), performs one projected online gradient descent step (blue, dashed), updates the input sequence (b), and applies the first part of the updated sequence to the system (green).

system (1), and calculates an input sequence that reaches the estimated optimal steady state. The whole procedure is illustrated in Fig. 1.

In more detail, an estimate of the measurement noise is computed in (5) by comparing the measured output $\tilde{y}_{t-1}$ to the output predicted at the previous time step $Y^{n+1}(\alpha_{t-1} + \beta_{t-1})$. The estimated measurement noise is then used in combination with the last n inputs $u_{[t-n:t-1]}$ and outputs $\tilde{y}_{[t-n:t-1]}$ in (6) to initialize a prediction step. As an input for prediction, we take the shifted previously predicted input sequence $\sigma \hat{u}_{t-1}$ and append it with the previously estimated optimal steady-state input u_{t-1}^s. Thus, α_t in (6) encodes the prediction at time t. In (7), the previously estimated steady-state input u_{t-1}^s and the μ-step ahead prediction $Y^{n+\mu+1}\alpha_t$ are collected in preparation for the projected OGD step in (8), where the parameter μ can be interpreted as the prediction horizon of Algorithm 1. As common in OGD, we perform one gradient descent step in (8) based on the previous cost function L_{t-1}, since we do not have access to the current cost function L_t yet. Note that multiplication by $I_{m+p} - S^\dagger S$ is equivalent to orthogonal projection onto the null space of S, which corresponds to the steady-state

manifold by Lemma 1. Thus, in (8), we perform one online gradient descent step and project it onto the steady-state manifold of system (1). The resulting input-output pair z_t^s can be regarded as an estimate for the optimal steady state. In (9), we compute an input sequence which, if applied in addition to the input sequence used for prediction (i.e., $U\alpha_t$), reaches the estimated optimal steady state z_t^s in μ steps and remains at z_t^s for another $n + 1$ steps in order to ensure that the unknown internal states x_t of system (1) reach the desired steady state. In order to be able to reach the estimated optimal steady state z_t^s in μ time steps, we require the prediction horizon μ to be sufficiently long.

Assumption 4: The prediction horizon satisfies $\mu \geq \mu^*$, where μ^* is the controllability index of system (1), i.e.,

$$\text{rank}([B \quad AB \quad \dots \quad A^{\mu^*-1}B]) = n.$$

Note that $n \geq \mu^*$ always holds. Since we require an upper bound of the system order to be available, it is therefore possible to satisfy Assumption 4 without knowing μ^*. However, simulations suggest that a shorter prediction horizon can sometimes be beneficial for the algorithm's performance, since decreasing the prediction horizon forces the algorithm to reach the desired steady state z^s in less steps in (9), resulting in a more aggressive controller. Finally, we update the predicted input sequence $\hat{u}_t$ in (10). Note that $U^{n+1:n+\mu+1}\alpha_t = \begin{bmatrix} \sigma \hat{u}_{t-1} \\ u_{t-1}^s \end{bmatrix}$. Thus, the predicted input sequence is updated by shifting it, appending u_{t-1}^s, and adding the input sequence encoded in β_t, which steers the system to the new estimate of the optimal steady state z_t^s. Last, the first part of $\hat{u}_t$ is applied in (11) to system (1). Then, we measure the new (noisy) system output $\tilde{y}_t$, receive the cost function L_t, and move to the next time step $t + 1$.

The matrix Q in the cost function of (9) can be tuned to achieve satisfactory performance, e.g., $Q = I$ minimizes the

norm of β and can be beneficial if there is process noise affecting the system (1), $Q = U$ minimizes the input difference needed to steer the system to z_t^s (instead of z_{t-1}^s), and $Q = Y$ similarly minimizes the deviation of the system's output from the predicted output. Moreover, weighted combinations are possible by stacking the matrices I, U, and Y in Q (compare Section V).

Since we compute input-output sequences of length $2n + \mu + 1$ in Algorithm 1 (n steps for initialization, μ steps for prediction, and $n + 1$ steps as a terminal constraint ensuring steady-state operation), by Theorem 1 we need persistency of excitation of order $3n + \mu + 1$.

Assumption 5: The input u^d of the data sequence is persistently exciting of order $3n + \mu + 1$.

Note that persistency of excitation of order $3n + \mu + 1$ requires a data sequence of length

$$N \geq (m+1)(3n + \mu + 1) - 1.$$

Finally, we derive explicit formulas to solve (6) and (9) in Algorithm 1. In order to do so, we need to ensure that (6) and (9) always have a feasible solution, which is guaranteed if their respective right-hand sides describe (parts of) valid input-output sequences of system (1) by Theorem 1.

Lemma 2: Let Assumptions 1, 3–5 be satisfied and assume that the initialization $u_{[-n:-1]}$, $\tilde{y}_{[-n,-1]} - \hat{e}_{[-n,-1]}$ is an input/output sequence of system (1). Then, (6) and (9) have a feasible solution for all $t \in \mathbb{I}_{\geq 0}$.

Proof: Assume that at time step t, $u_{[t-n:t-1]}$ and $\tilde{y}_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]}$ are a valid n-step trajectory of system (1). Then, α_t can be chosen according to (6), compare [22]. Moreover, there exists β_t' such that

$$H_\beta \beta_t' = \begin{bmatrix} 0_{mn} \\ 1_{n+1} \otimes u_t^s \\ 0_{pn} \\ 1_n \otimes y_t^s \end{bmatrix}$$

by Assumption 4, since system (1) can be steered from 0 to any steady state in μ steps by controllability, and β_t'' such that

$$H_\beta \beta_t'' = \begin{bmatrix} 0_{mn} \\ U^{n+\mu+1:2n+\mu+1}\alpha_t \\ 0_{pn} \\ Y^{n+\mu+1:2n+\mu}\alpha_t \end{bmatrix}$$

by Assumption 4 and because α_t encodes a valid input-output sequence. Since sums of input-output sequences of a linear time-invariant system are input-output sequences of the same system, there exists a solution to (9) at time step t given by $\beta_t = \beta_t' - \beta_t''$. Then, at time step $t + 1$, the right-hand side of (6) is a valid input-output sequence because of Theorem 1 and

$$U^{2:n+1}(\alpha_t + \beta_t) \overset{(6),(11)}{=} u_{[t-n+1:t]}$$

$$Y^{2:n+1}(\alpha_t + \beta_t) \overset{(5)}{=} \tilde{y}_{[t-n+1:t]} - \hat{e}_{[t-n+1:t]}.$$

Thus, (6) and (9) always have a solution by induction if the algorithm is initialized with a valid input-output sequence $u_{[-n:-1]}$ and $\tilde{y}_{[-n,-1]} - \hat{e}_{[-n,-1]}$. ∎

Lemma 2 states that we need a feasible initialization $u_{[-n:-1]}$, $\tilde{y}_{[-n:-1]} - \hat{e}_{[-n:-1]}$. Thus, in the following, we assume that Algorithm 1 is initialized correctly. This can be ensured, e.g., by choosing $u_{[-n:-1]} = 0$ and $\hat{e}_{[-n:-1]} = \tilde{y}_{[-n:-1]}$ or by solving

$$(\alpha_0, \hat{e}_{[-n:-1]}) =$$

$$\arg\min_{\alpha, \hat{e}} \quad \left\| Y^{1:n}\alpha - (\tilde{y}_{[-n:-1]} - \hat{e}_{[-n:-1]}) \right\| + \lambda \left\| \begin{bmatrix} \alpha \\ \hat{e} \end{bmatrix} \right\|$$

$$\text{s.t.} \quad U\alpha = \begin{bmatrix} u_{[-n:-1]} \\ \sigma \hat{u}_{-1} \\ 1_{n+1} \otimes u_{t-1}^s \end{bmatrix},$$

where $\lambda \in \mathbb{R}_{\geq 0}$ is a weighting factor, for some initialization $\hat{u}_{-1}$, u_{-1}^s, instead of solving (5) - (6) at time step $t = 0$.

Note that one solution to (6) is given by the pseudo-inverse

$$\alpha_t = H_\alpha^\dagger \begin{bmatrix} u_{[t-n:t-1]} \\ \hline \sigma \hat{u}_{t-1} \\ 1_{n+1} \otimes u_{t-1}^s \\ \hline \tilde{y}_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]} \end{bmatrix}. \tag{12}$$

Moreover, if $Q^\top Q \succ 0$, i.e., $Q^\top Q$ is positive definite, then the unique solution to (9) is given by the weighted pseudo-inverse [38]

$$\beta_t = \left(I_{N-2n-\mu} - \left(Q \left(I_{N-2n-\mu} - H_\beta^\dagger H_\beta \right) \right)^\dagger Q \right) H_\beta^\dagger g_t, \tag{13}$$

where g_t is the right-hand side of (9)

$$g_t = \begin{bmatrix} 0_{mn} \\ 1_{n+1} \otimes u_t^s - U^{n+\mu+1:2n+\mu+1}\alpha_t \\ 0_{pn} \\ 1_n \otimes y_t^s - Y^{n+\mu+1:2n+\mu}\alpha_t \end{bmatrix}. \tag{14}$$

If $Q^\top Q \succeq 0$ is only positive semidefinite, then the solution to (9) is not unique and (13) is only one possible solution. In the following, we assume that (9) is solved using (13) in both cases. Thus, the necessary online computations in Algorithm 1 reduce to one gradient evaluation and multiple matrix-vector multiplications.

IV. THEORETICAL RESULTS

In this section, we discuss theoretical guarantees for Algorithm 1, in particular a bound on the regret $\mathcal{R}$. In order to derive such a bound, we first analyze the error estimates $\hat{e}$. Lemma 3 states that the measurement error estimates $\hat{e}$ converge to the true measurement error e. Thus, Algorithm 1 is able to (asymptotically) exactly recover the measurement error e and control the true system output y_t, even though only noisy measurements $\tilde{y}_t$ are available at each time step.

Lemma 3: Let Assumptions 1, 3-5 be satisfied and assume that the initialization $u_{[-n:-1]}$, $\tilde{y}_{[-n,-1]} - \hat{e}_{[-n,-1]}$ is an input/output sequence of system (1). Then, the error of the measurement noise estimates $\hat{e} - e$ follows the unforced system dynamics, i.e., $\hat{e} - e$ is an output of (1) with $u \equiv 0$ and

$$\lim_{t \to \infty} \hat{e}_t - e_t = 0.$$

Proof: For every $t \geq 0$, let

$$\alpha_t^* = H_\alpha^\dagger \begin{bmatrix} u_{[t-n:t-1]} \\ \sigma \hat{u}_{t-1} \\ 1_{n+1} \otimes u_{t-1}^s \\ y_{[t-n:t-1]} \end{bmatrix},$$

i.e., the prediction with the real outputs $y_{[t-n:t-1]}$, compare (6), and

$$\epsilon_t = \alpha_t - \alpha_t^*. \tag{15}$$

Note that α_t is well-defined at all times due to Lemma 2. Then, we have $U\epsilon_t = 0$ by definition of ϵ_t and

$$Y^{1:n} \epsilon_t \overset{(6)}{=} \tilde{y}_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]} - y_{[t-n:t-1]}$$

$$\overset{(1)}{=} e_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]}. \tag{16}$$

Moreover, $y_{t-1} = Y^{n+1}(\alpha_{t-1}^* + \beta_{t-1})$ by Theorem 1, i.e., the coefficients $\alpha_{t-1}^* + \beta_{t-1}$ encode the true initialization $(u_{[t-n-1:t-2]}, y_{[t-n-1:t-2]})$ and input $U^{n+1}(\alpha_{t-1}^* + \beta_{t-1}) = U^{n+1}(\alpha_{t-1} + \beta_{t-1}) \overset{(11)}{=} u_{t-1}$ and, therefore, predict the output y_{t-1} correctly. Combining this fact with (16) yields

$$Y^n \epsilon_t = \tilde{y}_{t-1} - \hat{e}_{t-1} - y_{t-1}$$

$$\overset{(5)}{=} Y^{n+1}(\alpha_{t-1} + \beta_{t-1}) - y_{t-1}$$

$$= Y^{n+1}(\alpha_{t-1} - \alpha_{t-1}^*) = Y^{n+1}\epsilon_{t-1},$$

which implies

$$Y^{1:n} \epsilon_t \overset{(16)}{=} Y^{2:n+1} \epsilon_{t-1}. \tag{17}$$

Combining the above results $U\epsilon_t = 0$ and (17), we conclude that the error sequence $e_t - \hat{e}_t$ follows the unforced system dynamics. In more detail, at each time step t the sequence generated by $Y\epsilon_t$ by (17) is initialized by the endpiece of the initialization of $Y\epsilon_{t-1}$ (i.e., $Y^{2:n}\epsilon_{t-1}$) appended with the one step ahead prediction at time $t - 1$ (i.e., $Y^{n+1}\epsilon_{t-1}$). Therefore, we have by Theorem 1 and $U\epsilon_t = 0$ for all t that $e_t - \hat{e}_t$ is the output of a trajectory of the unforced system for all t. Since the unforced system dynamics are stable by Assumption 1, we obtain the result $\lim_{t \to \infty} e_t - \hat{e}_t = 0$. ∎

Next, we are able to derive an upper bound on the regret $\mathcal{R}$ as stated in Theorem 2.

Theorem 2: Let Assumptions 1–5 be satisfied and choose $0 < \gamma \leq \frac{2}{\alpha_z + l_z}$. Moreover, assume that the initialization $u_{[-n:-1]}$, $\tilde{y}_{[-n,-1]} - \hat{e}_{[-n,-1]}$ is an input/output sequence of system (1). Then, the regret (3) can be upper bounded by

$$\mathcal{R} \leq C_\mu + C_\zeta \sum_{t=0}^{T} \|\zeta_t - \zeta_{t-1}\| + C_e E_0,$$

where $E_0 = \|e_{[-n:-1]} - \hat{e}_{[-n:-1]}\|$ and $C_\mu, C_\zeta, C_e < \infty$ are constants independent of T and $\zeta_{-1} = z_{-1}^s$.

The proof is given in the appendix. The upper bound on the regret depends on constants, which in turn depend on system and problem parameters, $\sum_{t=0}^{T} \|\zeta_t - \zeta_{t-1}\|$, and $E_0 = \|e_{[-n:-1]} - \hat{e}_{[-n:-1]}\|$, i.e., the initialization error of the measurement error estimates. The quantity $\sum_{t=0}^{T} \|\zeta_t - \zeta_{t-1}\|$, commonly termed path length in the literature [39], can be regarded as a measure of the variation of the cost functions. A bound which depends on the variation of the cost functions is to be expected, since in our framework, the cost function L_t is only available to the algorithm at time step $t + 1$, i.e., there is a one-step delay between the cost function becoming active and being used to control the system. Thus, it is impossible to achieve low regret if the cost functions vary too frequently. A bound which depends linearly on $\sum_{t=0}^{T} \|\zeta_t - \zeta_{t-1}\|$ is well aligned with other results on dynamic regret in the literature, compare, e.g., [7], [10]. A sublinear regret can therefore be achieved if the path length is sublinear in T. Moreover, introduction of measurement noise to the control problem only introduces an additional *constant* term $C_e E_0$ in the regret upper bound compared to [29]. This is due to the convergence of $\hat{e}$ to e as shown in Lemma 3.

Finally, consider the case where the optimal steady state is constant, i.e., $\zeta_t = \zeta_{t'}$ for some $t' \in \mathbb{I}_{\geq 0}$ and all $t \geq t'$. Following the proof of Theorem 2, it can be shown that

$$\sum_{t=0}^{T} \|z_t - \zeta_t\| \leq C_\mu' + C_\zeta' \sum_{t=0}^{T} \|\zeta_t - \zeta_{t-1}\| + C_e E_0,$$

where $C_\mu,' C_\zeta' < \infty$ are again constants independent of T. Thus,

$$\sum_{t=t'+1}^{T} \|z_t - \zeta_t\| \leq C_\mu' + C_e E_0,$$

which implies that, in the case that the optimal steady state is constant, the closed loop with Algorithm 1 asymptotically converges to the optimal steady state.

Remark 2: (Unstable systems) Suppose that Assumption 1 is not satisfied because the system is not Schur stable. In this case, it is possible to design a linear stabilizing feedback [19], [20] and apply Algorithm 1 to the stabilized system, as discussed above. In particular, at each time step t, $u_t + v_t$ is applied to the system, where u_t is the input computed in (11) and

$$v_t = v_t^y + v_t^e = K \begin{bmatrix} v_{[t-n:t-1]} \\ \tilde{y}_{[t-n:t-1]} \end{bmatrix}$$

$$= K \begin{bmatrix} v_{[t-n:t-1]}^y \\ y_{[t-n:t-1]} \end{bmatrix} + K \begin{bmatrix} v_{[t-n:t-1]}^e \\ e_{[t-n:t-1]} \end{bmatrix},$$

where K is the stabilizing controller. In order to do so, every Hankel matrix of the open-loop system in Algorithm 1 has to be replaced by Hankel matrices of the stabilized system. Moreover, a mapping $V(y^s)$ can be computed that maps a steady-state output y^s of the system to a steady-state input of the stabilizing controller. Finally, the cost functions have to be reformulated to $L_t(u + V(y), y)$ to account for the stabilizing input when determining the optimal steady state. Then, it is still possible to derive a regret upper bound for the prestabilized system, but the theoretical guarantees deteriorate in two ways:

1) The estimates of the measurement error do not asymptotically exactly converge to the true measurement error as stated in Lemma 3, because the stabilized system is only practically (and not asymptotically) stable due to the measurement noise. Instead, the estimates $\hat{e}$ inherit their stability properties from the stabilized system. For example, assume that the stabilizing feedback stabilizes some robust positive invariant (RPI) set. In this case, the estimates $\hat{e}$ converge to the same RPI set around the true measurement error e.

2) The regret bound is increased by additional terms $\|Y^{n+\mu+1}H_\alpha^\dagger\|(C_{sl}Tv_{max} + C_{sc}v_{max})$, where the constants C_{sl} and C_{sc} only depend on system parameters A, B, C, D, n, and the prediction horizon μ, and v_{max} is an upper bound on the error feedback $\|v_t^e\| \leq v_{max}$ for all t. Thus, the regret upper bound for the prestabilized system becomes *linear* in T, which is to be expected since the stabilizing controller feeds back the measurement error e_t at every time step, thereby preventing us from staying at the optimal steady state. In this regard, the feedback of the measurement error v_t^e can also be interpreted as process noise acting on a stable system. □

V. APPLICATION EXAMPLE - THERMAL CONTROL
A. SETTING
In this section, we test our OCO-based control scheme on a thermal control problem. Specifically, we consider a Heating Ventilation and Air Conditioning (HVAC) system which controls the temperature of five nonuniform zones. The HVAC system is equipped with a sensor in zone 1, 4, and 5, and actuators adjusting the supply air rate in every zone. The zones are depicted in Fig. 2. We consider the linear thermal dynamics model proposed in [11], [40] given by

$$C_i\dot{T}_i = \frac{T^o - T_i}{R_i} + \sum_{j \in \mathcal{N}(i)} \frac{T_j - T_i}{R_{ij}} + u_{i,t} + q_{i,t},$$

where C_i is the thermal capacitance of zone i, T_i is the zone temperature of zone i, T^o is the outdoor temperature, R_i is the thermal resistance between the i-th zone and outside, R_{ij} is the thermal resistance between zones i and j, $\mathcal{N}(i)$ denotes the set of zones neighboring zone i, $u_{i,t}$ is the control input at time t associated with zone i, and $q_{i,t}$ denotes (unknown) process noise, caused, e.g., by additional heat sources in zone
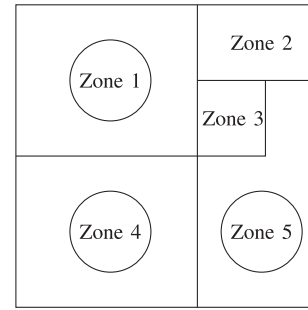

FIGURE 2. Schematic illustration of the 5 zones controlled by an HVAC system. Measured zones are indicated by circles.

i. For zone 3, we set $R_3 = \infty$ in our simulation since it is surrounded by other zones and, therefore, not directly influenced by the outdoor temperature. Note that we do not consider process noise in our theoretical work, but do consider it in the simulation as an additional difficulty. We define the system states as $x_t = \begin{bmatrix} x_{1,t} & x_{2,t} & x_{3,t} & x_{4,t} & x_{5,t} \end{bmatrix}^\top \in \mathbb{R}^5$, where $x_{i,t} = \Delta T_i = T_i - T^o$ denotes the difference between the temperature of the i-th zone and the outside temperature at time t. Since there are sensors only in zones 1, 4, and 5, but an actuator in every zone, we set

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = I_{n=5}, \quad D = 0.$$

Then, we discretize the thermal dynamics with sample time $t_s = 60$ s. The cost function consists of a term penalizing the deviation from a desired temperature T_t^{set} and a term minimizing control cost

$$L_t(u, y) = \frac{1}{2}(y - \Delta T_t^{set})^\top \Lambda_t (y - \Delta T_t^{set}) + \frac{\lambda_t p_t}{2} \|u\|^2,$$

where $\Delta T_t^{set} = T_t^{set} - T^o$, $\Lambda_t \in \mathbb{R}^{3 \times 3}$ $\lambda_t, \in \mathbb{R}$ are a priori unknown time-varying parameters, and p_t denotes the a priori unknown energy cost. In particular, λ_t and Λ_t are weighting factors, trading off user comfort and control cost. We set $T^o = 15\ °C$, $\lambda_t = 10$, $\Lambda_t = I_p$. However, we change Λ_t to $\Lambda_t = 0.1I_p$ between 0 am and 6 am, in order to save energy during the night. The normalized energy cost p_t is shown in Fig. 3. We choose $T_t^{set} = 18° C \cdot 1_p$ but switch it, a priori unbeknown to the algorithm, at 9 am to $T_t^{set} = 21°C \cdot 1_p$. In Algorithm 1, we choose $Q = \begin{bmatrix} I_{N-2n-\mu} & U^\top \end{bmatrix}^\top$, $\mu \in \{10, 30\}$, and $\gamma = 0.15$, which satisfies $\gamma \leq \frac{2}{l_z + \alpha_z}$. At $t = n = 5$, we initialize the algorithm with $\hat{u}_4 = 0_{\mu+1}$, $z_4^s = 0_{m+p}$, $u_{[0:4]} = 0_{mn}$, and $\hat{e}_{[0:4]} = \tilde{y}_{[0:4]}$, while the real (unknown) initial condition is $x_0 + T^o = 17\ °C$ for each zone. Note that Algorithm 1 does not control the system for the first $n = 5$ time steps. Finally, we sample $q_{t,i}$ uniformly from the interval $[-0.1, 0.1]$.
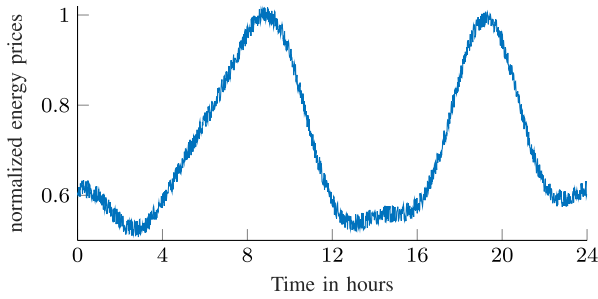
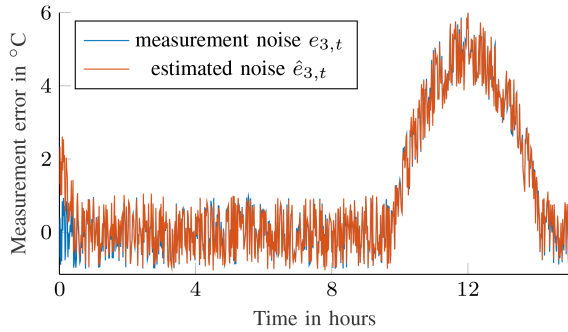**FIGURE 3.** Energy prices p_t over one day.

FIGURE 4. Real and estimated measurement error in zone 5.

B. PREDICTION HORIZON AND ROBUSTNESS TO MEASUREMENT NOISE

First, we simulate the proposed Algorithm 1 with different prediction horizons and assess its robustness with respect to measurement noise. To this end, we sample the measurement error e_t uniformly from the interval $[-1, 1]$. Moreover, we increase the measurement error of the sensor in the fifth zone $e_{3,t}$ between 10 am and 2 pm as shown in Fig. 4 to simulate a failing sensor.

The results are illustrated in Figs. 4 and 5. Fig. 4 shows the measurement error in the fifth zone $e_{3,t}$ and the corresponding estimate of Algorithm 1 $\hat{e}_{3,t}$ for the first 15 hours. Initially, the estimate is off by 2 °C because of the wrong initialization, but then converges to the true measurement error in accordance with Lemma 3. Note that a slight mismatch persists due to process noise.

Fig. 5 shows the true closed-loop temperatures and inputs of zones 2 and 5, only one of which can be measured, together with the optimal steady state (η, θ) for both zones. Even though the temperature in zone 2 cannot be measured and the algorithm has to cope with process and measurement noise, the closed loop closely tracks the optimal steady state. This is true for sudden changes, i.e., the change of Λ_t at 6 am and the change in T^{set} at 9 am, as well as for gradual changes due to the fluctuation of the energy prices p_t. Note that the increase in measurement noise around 12 am has no influence on the control performance. The noise in the true temperatures is due to process noise.
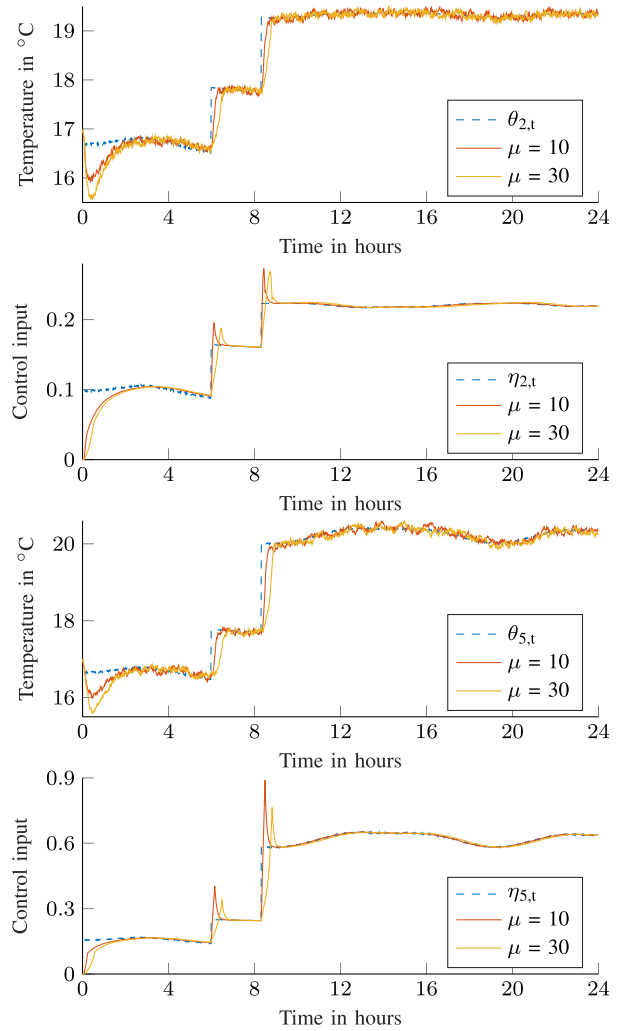

FIGURE 5. Optimal steady state (blue, dashed), closed-loop real temperatures and inputs for $\mu = 10$ (red) and $\mu = 30$ (yellow). From top to bottom: Temperatures in zone 2; Control inputs in zone 2; Temperatures in zone 5; Control inputs in zone 5.

Comparing the different values for the prediction horizon $\mu \in \{10, 30\}$, Fig. 5 indicates that a shorter prediction horizon yields a more aggressive controller. The accumulated cost over the whole day are approximately 7165 for $\mu = 10$ and 7247 for $\mu = 30$ for the same noise realization. Thus, a shorter prediction horizon yields (slightly) superior performance in this example.

C. COMPARISON TO RELATED WORK

In a second experiment, we compare Algorithm 1 to the method proposed in [28] for a similar setting (compare the discussion in the Introduction). In order to achieve satisfactory performance for both algorithms, we have to reduce the measurement error and sample it uniformly from the interval $[-0.1, 0.1]$. For Algorithm 1, we choose the same parameters and initialization as before and choose $\mu = 10$. For the algorithm proposed in [28], we set the step size η to 0.005.
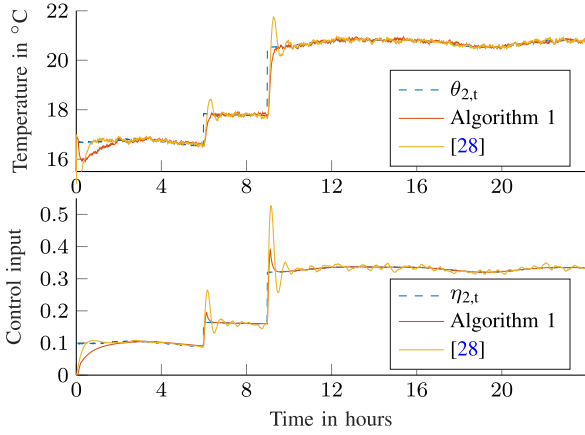
FIGURE 6. Optimal steady state (blue, dashed), closed-loop real temperatures and inputs for Algorithm 1 (red) and the algorithm proposed in [28] (yellow). From top to bottom: Temperatures in zone 2; Control inputs in zone 2.

The results are illustrated in Fig. 6. It can be seen that both algorithms are able to track the time-varying optimal steady state closely. Moreover, for these parameters, both algorithms achieve almost the same closed-loop cost. However, the algorithm proposed in [28] does so with a higher overshoot, more oscillations, and larger control inputs.

VI. CONCLUSION

In this paper, we proposed a data-driven OCO-based scheme for controlling linear dynamical systems subject to measurement noise. We only use a single persistently exciting data trajectory instead of a model of the system and output feedback to derive the control algorithm. The control scheme achieves a similar sublinear regret bound as comparable algorithms from the literature, despite only having access to noisy measurements. In particular, we show that adding measurement noise to the control problem only leads to an additional *constant* term in the regret bound. Compared to previous work, the proposed algorithm is able to handle the more general and practically important case of *economic* cost functions and additionally allows to relax previous assumptions on the steady-state manifold of the system.

Future work includes obtaining theoretical guarantees for the case of both process and measurement noise, as well as considering noisy a priori data. Furthermore, enabling the proposed algorithm to handle state and input constraints, which has already been achieved in a model-based setting, is an interesting direction of future research.

APPENDIX

A. PROOF OF THEOREM 2

Before we prove the regret bound, we first derive some auxiliary results. Note that α_t and β_t in (6) and (9) always have a

solution by Lemma 2. Then, by (6), we have

$$
\begin{bmatrix} U^{1:n} \\ Y^{1:n} \end{bmatrix} \alpha_t = \begin{bmatrix} U^{2:n}\alpha_{t-1} \\ u_{t-1} \\ Y^{2:n}\alpha_{t-1} \\ \tilde{y}_{t-1} - \hat{e}_{t-1} \end{bmatrix} \overset{(9)}{=} \begin{bmatrix} U^{2:n}(\alpha_{t-1} + \beta_{t-1}) \\ u_{t-1} \\ Y^{2:n}(\alpha_{t-1} + \beta_{t-1}) \\ \tilde{y}_{t-1} - \hat{e}_{t-1} \end{bmatrix}
$$

$$
\overset{(5),(11)}{=} \begin{bmatrix} U^{2:n+1} \\ Y^{2:n+1} \end{bmatrix} (\alpha_{t-1} + \beta_{t-1}). \tag{18}
$$

Moreover, the input sequence generated by α_t is given by

$$
U^{n+1:2n+\mu}\alpha_t \overset{(6)}{=} \begin{bmatrix} \sigma\hat{u}_{t-1} \\ 1_n \otimes u^s_{t-1} \end{bmatrix}
$$

$$
\overset{(10)}{=} \begin{bmatrix} \sigma(U^{n+1:n+\mu+1}(\alpha_{t-1} + \beta_{t-1})) \\ 1_n \otimes u^s_{t-1} \end{bmatrix}
$$

$$
\overset{(9)}{=} U^{n+2:2n+\mu+1}(\alpha_{t-1} + \beta_{t-1}). \tag{19}
$$

Hence, α_t and $\alpha_{t-1} + \beta_{t-1}$ give rise to the same initialization (18) and the same input sequence (19) and, therefore, must produce the same output trajectory [22]

$$
Y^{n+1:2n+\mu}\alpha_t = Y^{n+2:2n+\mu+1}(\alpha_{t-1} + \beta_{t-1}). \tag{20}
$$

Note that

$$
Y^{n+\mu+1:2n+\mu}(\alpha_t + \beta_t) \overset{(9)}{=} 1_n \otimes y^s_t,
$$

$$
U^{n+\mu+1:2n+\mu+1}(\alpha_t + \beta_t) \overset{(9)}{=} 1_{n+1} \otimes u^s_t,
$$

which implies that the predicted system is at the equilibrium (u^s_t, y^s_t) for n time steps and at the $(n+1)$-th time step, u^s_t is applied again. Hence, the system remains at the same equilibrium and we have

$$
Y^{n+\mu+1:2n+\mu+1}(\alpha_t + \beta_t) = 1_{n+1} \otimes y^s_t. \tag{21}
$$

Moreover, we need the following key result on the convergence rate of projected gradient descent from [41, Theorem 2.2.14]. Let $L(z)$ be an α_z-strongly convex and l_z-smooth function to be minimized. Then, one projected gradient descent step $z_1 = \Pi_Z(z_0 - \gamma\nabla L(z_0))$, where $\Pi_Z(\cdot)$ denotes projection onto the set Z and $\gamma \leq \frac{2}{\alpha_z + l_z}$ is the step size, satisfies

$$
\|z_1 - z_0\| \leq \kappa\|z_0 - z^*\|, \tag{22}
$$

where $z^* = \arg\min_{z \in Z} L(z)$ and $\kappa = 1 - \alpha_z\gamma$.

Now, we are ready to bound the regret $\mathcal{R}$ of Algorithm 1. By definition of the regret and Lipschitz continuity of the cost functions, we have

$$
\mathcal{R} \overset{(3)}{=} \sum_{t=0}^{T} L_t(z_t) - L_t(\zeta_t) = C_\mu + \sum_{t=\mu}^{T} L_t(z_t) - L_t(\zeta_t)
$$

$$
\leq C_\mu + L_z \sum_{t=0}^{T-\mu} \|z_{t+\mu} - \zeta_{t+\mu}\|
$$

where $C_\mu = \sum_{t=0}^{\mu-1} L_t(z_t) - L_t(\zeta_t)$ is a constant which is independent of T. Applying the triangle inequality yields

$$\mathcal{R} \leq C_\mu + L_z \sum_{t=0}^{T-\mu} \left\| z_{t+\mu} - \hat{z}_t^\mu \right\| + L_z \sum_{t=0}^{T-\mu} \left\| \hat{z}_t^\mu - \zeta_{t-1} \right\|$$

$$+ L_z \sum_{t=0}^{T-\mu} \left\| \zeta_{t+\mu} - \zeta_{t-1} \right\|.$$

Again applying the triangle inequality, we get

$$\sum_{t=0}^{T-\mu} \left\| \zeta_{t+\mu} - \zeta_{t-1} \right\| = \sum_{t=0}^{T-\mu} \left\| \sum_{i=0}^{\mu} \zeta_{t+i} - \zeta_{t+i-1} \right\|$$

$$\leq \sum_{t=0}^{T-\mu} \sum_{i=0}^{\mu} \left\| \zeta_{t+i} - \zeta_{t+i-1} \right\| \leq (\mu + 1) \sum_{t=0}^{T} \left\| \zeta_t - \zeta_{t-1} \right\|,$$

which implies

$$\mathcal{R} \leq C_\mu + L_z \sum_{t=0}^{T-\mu} \left\| z_{t+\mu} - \hat{z}_t^\mu \right\| + L_z \sum_{t=0}^{T-\mu} \left\| \hat{z}_t^\mu - \zeta_{t-1} \right\|$$

$$+ L_z (\mu + 1) \sum_{t=0}^{T} \left\| \zeta_t - \zeta_{t-1} \right\|. \tag{23}$$

Next, we will establish bounds for the first two sums in (23) separately. First, we bound the predicted regret $\sum_{t=0}^{T-\mu} \| \hat{z}_t^\mu - \zeta_{t-1} \|$. We note that ζ_t is only defined for $t \in \mathbb{I}_{[0,T]}$. Therefore, we are free to choose $\zeta_{-1} = z_{-1}^s$. Thus, we have for each $\tau \in \mathbb{I}_{[0,T]}$

$$\sum_{t=0}^{\tau} \left\| \hat{z}_t^\mu - \zeta_{t-1} \right\| \stackrel{(7)}{=} \sum_{t=0}^{\tau} \left\| \begin{bmatrix} u_{t-1}^s \\ Y^{n+\mu+1} \alpha_t \end{bmatrix} - \zeta_{t-1} \right\|$$

$$\stackrel{(20)}{=} \sum_{t=0}^{\tau} \left\| \begin{bmatrix} u_{t-1}^s \\ Y^{n+\mu+2}(\alpha_{t-1} + \beta_{t-1}) \end{bmatrix} - \zeta_{t-1} \right\|$$

$$\stackrel{(9)}{=} \sum_{t=0}^{\tau} \left\| \begin{bmatrix} u_{t-1}^s \\ y_{t-1}^s \end{bmatrix} - \zeta_{t-1} \right\|$$

$$\leq \sum_{t=0}^{\tau-1} \left\| z_t^s - \zeta_{t-1} \right\| + \sum_{t=0}^{\tau-1} \left\| \zeta_t - \zeta_{t-1} \right\|$$

$$\stackrel{(8),(22)}{\leq} \kappa \sum_{t=0}^{\tau} \left\| \hat{z}_t^\mu - \zeta_{t-1} \right\| + \sum_{t=0}^{\tau-1} \left\| \zeta_t - \zeta_{t-1} \right\|.$$

Since $\kappa < 1$, rearranging yields

$$\sum_{t=0}^{\tau} \left\| \hat{z}_t^\mu - \zeta_{t-1} \right\| \leq \frac{1}{1-\kappa} \sum_{t=0}^{\tau-1} \left\| \zeta_t - \zeta_{t-1} \right\|. \tag{24}$$

Having established a bound on the second sum in (23), we proceed to bound the accumulated prediction error $\sum_{t=0}^{T-\mu} \| z_{t+\mu} - \hat{z}_t^\mu \|$. In order to do so, we first need a bound

on $\sum_{t=0}^{T} \| g_t \|$, where g_t is defined in (14). First, note that

$$\sum_{t=0}^{T} \left\| z_t^s - \zeta_{t-1} \right\| \stackrel{(8),(22)}{\leq} \kappa \sum_{t=0}^{T} \left\| \hat{z}_t^\mu - \zeta_{t-1} \right\|$$

$$\stackrel{(24)}{\leq} \frac{\kappa}{1-\kappa} \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\|, \tag{25}$$

where we used (24) with $\tau = T$ in the last line. Therefore,

$$\sum_{t=0}^{T} \left\| z_t^s - z_{t-1}^s \right\| \leq \sum_{t=0}^{T} \left\| z_t^s - \zeta_{t-1} \right\| + \sum_{t=0}^{T} \left\| z_{t-1}^s - \zeta_{t-1} \right\|$$

$$\leq 2 \sum_{t=0}^{T} \left\| z_t^s - \zeta_{t-1} \right\| + \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\|$$

$$\stackrel{(25)}{\leq} \frac{1+\kappa}{1-\kappa} \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\|, \tag{26}$$

where we used $\zeta_{-1} = z_{-1}^s$, positivity of the norm and the triangle inequality in the second inequality. Then, we have

$$\sum_{t=0}^{T} \| g_t \| \stackrel{(14)}{=} \sum_{t=0}^{T} \left\| \begin{bmatrix} 1_{n+1} \otimes u_t^s - U^{n+\mu+1:2n+\mu+1} \alpha_t \\ 1_n \otimes y_t^s - Y^{n+\mu+1:2n+\mu} \alpha_t \end{bmatrix} \right\|$$

$$\stackrel{(6),(20),(21)}{=} \sum_{t=0}^{T} \left\| \begin{bmatrix} 1_{n+1} \otimes (u_t^s - u_{t-1}^s) \\ 1_n \otimes (y_t^s - y_{t-1}^s) \end{bmatrix} \right\|$$

Rearranging the vector on the right-hand side yields

$$\sum_{t=0}^{T} \| g_t \| \leq \sum_{t=0}^{T} \left\| 1_{n+1} \otimes \begin{bmatrix} u_t^s - u_{t-1}^s \\ y_t^s - y_{t-1}^s \end{bmatrix} \right\|$$

$$\leq \sqrt{n+1} \sum_{t=0}^{T} \left\| z_t^s - z_{t-1}^s \right\|$$

$$\stackrel{(26)}{\leq} \sqrt{n+1} \frac{1+\kappa}{1-\kappa} \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\|. \tag{27}$$

Let $\tilde{Q} = (I_{N-2n-\mu} - (Q(I_{N-2n-\mu} - H_\beta^\dagger H_\beta))^\dagger Q) H_\beta^\dagger$, then we have $\beta_t = \tilde{Q} g_t$ by (13). Hence,

$$\sum_{t=0}^{T} \| \beta_t \| \stackrel{(27)}{\leq} C_\beta \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\|, \tag{28}$$

where $C_\beta = \|\tilde{Q}\| \sqrt{n+1} \frac{1+\kappa}{1-\kappa}$.

Having established a bound on $\sum_{t=0}^{T} \| \beta_t \|$, we proceed to first bound the output prediction error $\sum_{t=0}^{T-\mu} \| Y^{n+\mu+1} \alpha_t - y_{t+\mu} \|$, and then the full prediction error $\sum_{t=0}^{T-\mu} \| z_{t+\mu} - \hat{z}_t^\mu \|$. To this extent, we first bound the error of the measurement noise estimates in (5). Let

$$S_O = \begin{bmatrix} C^\top & (CA)^\top & \dots & (CA^{n-1})^\top \end{bmatrix}^\top$$

be the system's observability matrix and recall the definition of $\epsilon_t \stackrel{(15)}{=} \alpha_t - \alpha_t^*$ in the proof of Lemma 3. Since $U \epsilon_t = 0$,

$Y^{1:n}\epsilon_t$ describes a trajectory of the unforced system by Theorem 1. Thus, by observability there exists a unique internal state x_t^ϵ such that

$$Y^{1:n}\epsilon_t = S_O x_{t-n}^\epsilon$$

holds for all t. Hence,

$$x_{t-n}^\epsilon = S_O^\dagger Y^{1:n}\epsilon_t.$$

Recalling that the error trajectory follows the unforced system dynamics by Lemma 3, we can conclude

$$Y^{1:n}\epsilon_t = S_O x_{t-n}^\epsilon = S_O A x_{t-n-1}^\epsilon$$
$$= S_O A S_O^\dagger Y^{1:n}\epsilon_{t-1}.$$

Using these arguments repeatedly, we get

$$\sum_{t=0}^{T-\mu} \left\| e_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]} \right\| \overset{(16)}{=} \sum_{t=0}^{T-\mu} \left\| Y^{1:n}\epsilon_t \right\|$$

$$= \sum_{t=0}^{T-\mu} \left\| S_O A S_O^\dagger Y^{1:n}\epsilon_{t-1} \right\| = \sum_{t=0}^{T-\mu} \left\| S_O A^t S_O^\dagger Y^{1:n}\epsilon_0 \right\|$$

$$\leq \| S_O \| \left\| S_O^\dagger \right\| \sum_{t=0}^{T-\mu} \| A^t \| \left\| e_{[-n:-1]} - \hat{e}_{[-n:-1]} \right\|.$$

Note that $\| S_O \| \| S_O^\dagger \| = \frac{\sigma_{max}(S_O)}{\sigma_{min}(S_O)}$ where $\sigma_{max}(S_O)$, $\sigma_{min}(S_O)$ denote the largest and smallest singular value of S_O, respectively. Moreover, since A is Schur stable by Assumption 1, there exist constants $c > 0$ and $\lambda \in (0, 1)$ such that $\| A^t \| \leq c\lambda^t$. Thus,

$$\sum_{t=0}^{T-\mu} \left\| e_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]} \right\| \leq \frac{\sigma_{max}(S_O)}{\sigma_{min}(S_O)} E_0 c \sum_{t=0}^{T-\mu} \lambda^t$$

$$\leq \frac{\sigma_{max}(S_O)}{\sigma_{min}(S_O)} \frac{c}{1-\lambda} E_0. \quad (29)$$

Next, let

$$\bar{\alpha}_t^* = H_\alpha^\dagger \begin{bmatrix} u_{[t-n:t+\mu]} \\ 1_n \otimes u_{t-1}^s \\ y_{[t-n:t-1]} \end{bmatrix}. \quad (30)$$

Then, we have $y_{t+\mu} = Y^{n+\mu+1}\bar{\alpha}_t^*$. Moreover, we have that

$$u_{t+j} \overset{(11)}{=} U^{n+1}(\alpha_{t+j} + \beta_{t+j})$$

$$\overset{(6),(10)}{=} U^{n+2}(\alpha_{t+j-1} + \beta_{t+j-1}) + U^{n+1}\beta_{t+j}.$$

Applying (6) and (10) repeatedly, we get

$$u_{t+j} = U^{n+j+1}\alpha_t + \sum_{i=0}^{j} U^{n+i+1}\beta_{t+j-i} \quad (31)$$

for $0 \leq j \leq \mu - 1$ and

$$u_{t+\mu} = u_{t-1}^s + \sum_{i=0}^{\mu} U^{n+i+1}\beta_{t+\mu-i}. \quad (32)$$

Define $C_1 = \| Y^{n+\mu+1} H_\alpha^\dagger \|$. Combining the above results, we are now ready to bound the output prediction error $\sum_{t=0}^{T-\mu} \| Y^{n+\mu+1}\alpha_t - y_{t+\mu} \|$. By Theorem 1, any α_t satisfying (6) results in the same output $Y^{n+\mu+1}\alpha_t$ since the vector on the right-hand side of (6) uniquely specifies the input sequence and initial condition (compare [22]). Hence, in the following we assume without loss of generality that α_t is chosen according to (12).

$$\sum_{t=0}^{T-\mu} \left\| Y^{n+\mu+1}\alpha_t - y_{t+\mu} \right\| = \sum_{t=0}^{T-\mu} \left\| Y^{n+\mu+1}(\alpha_t - \bar{\alpha}_t^*) \right\|$$

$$\overset{(12),(30)}{\leq} C_1 \sum_{t=0}^{T-\mu} \left\| \begin{bmatrix} u_{[t-n:t-1]} - u_{[t-n:t-1]} \\ U^{n+1}\alpha_t - u_t \\ \vdots \\ U^{n+\mu+1}\alpha_t - u_{t+\mu} \\ 1_n \otimes (u_{t-1}^s - u_{t-1}^s) \\ \tilde{y}_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]} + y_{[t-n:t-1]} \end{bmatrix} \right\|$$

$$\overset{(31),(32)}{\leq} C_1 \sum_{t=0}^{T-\mu} \left\| \begin{bmatrix} U^{n+1}\beta_t \\ \sum_{i=0}^{1} \tilde{U}^{n+i+1}\beta_{t+1-i} \\ \vdots \\ \sum_{i=0}^{\mu} U^{n+i+1}\beta_{t+\mu-i} \\ e_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]} \end{bmatrix} \right\|$$

Define $C_e = C_1 \frac{\sigma_{max}(S_O)}{\sigma_{min}(S_O)} \frac{c}{1-\lambda}$. Then, we get

$$\sum_{t=0}^{T-\mu} \left\| Y^{n+\mu+1}\alpha_t - y_{t+\mu} \right\|$$

$$\leq C_1 \sum_{t=0}^{T-\mu} \sum_{i=0}^{\mu} \left\| U^{n+1:n+1+i}\beta_{t+\mu-i} \right\|$$

$$+ C_1 \sum_{t=0}^{T-\mu} \left\| e_{[t-n:t-1]} - \hat{e}_{[t-n:t-1]} \right\|$$

$$\overset{(29)}{\leq} C_1 \sum_{t=0}^{T-\mu} \sum_{i=0}^{\mu} \left\| U^{n+1:n+\mu+1} \right\| \left\| \beta_{t+\mu-i} \right\| + C_e E_0$$

$$\leq C_1 \left\| U^{n+1:n+\mu+1} \right\| (\mu + 1) \sum_{t=0}^{T} \left\| \beta_t \right\| + C_e E_0$$

$$\overset{(28)}{\leq} C_2 \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\| + C_e E_0, \quad (33)$$

where $C_2 = C_1 \| U^{n+1:n+\mu+1} \| (\mu + 1) C_\beta$.

Next, we are finally ready to bound the prediction error $\sum_{t=0}^{T-\mu} \| \hat{z}_t^\mu - z_{t+\mu} \|$:

$$\sum_{t=0}^{T-\mu} \left\| \hat{z}_t^\mu - z_{t+\mu} \right\| \overset{(7)}{=} \sum_{t=0}^{T-\mu} \left\| \begin{bmatrix} u_{t-1}^s - u_{t+\mu} \\ Y^{n+\mu+1}\alpha_t - y_{t+\mu} \end{bmatrix} \right\|$$

$$\overset{(32)}{\leq} \sum_{t=0}^{T-\mu} \left\| \sum_{i=0}^{\mu} U^{n+i+1} \beta_{t+\mu-i} \right\| + \sum_{t=0}^{T-\mu} \left\| Y^{n+\mu+1} \alpha_t - y_{t+\mu} \right\|$$

Positivity of the norm and inserting (33) yields

$$\sum_{t=0}^{T-\mu} \left\| \hat{z}_t^{\mu} - z_{t+\mu} \right\| \leq \left\| U^{n+1:n+\mu+1} \right\| \sum_{t=0}^{T-\mu} \sum_{i=0}^{\mu} \left\| \beta_{t+\mu-i} \right\|$$

$$+ C_2 \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\| + C_e E_0$$

$$\leq \left\| U^{n+1:n+\mu+1} \right\| (\mu+1) \sum_{t=0}^{T} \left\| \beta_t \right\|$$

$$+ C_2 \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\| + C_e E_0$$

$$\overset{(28)}{\leq} \left\| U^{n+1:n+\mu+1} \right\| (\mu+1) C_\beta \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\|$$

$$+ C_2 \sum_{t=0}^{T-1} \left\| \zeta_t - \zeta_{t-1} \right\| + C_e E_0. \tag{34}$$

The result then follows from inserting (24) with $\tau = T - \mu$ and (34) into (23). ∎

REFERENCES

[1] M. Picallo, S. Bolognani, and F. Dörfler, "Closing the loop: Dynamic state estimation and feedback optimization of power grids," *Electric Power Syst. Res.*, vol. 189, 2020, Art. no. 106753.

[2] N. Lazic et al., "Data center cooling using model-predictive control," in *Proc. 32nd Conf. Neural Inf. Process. Syst.*, 2018, pp. 3818–3827.

[3] T. Zheng, J. Simpson-Porco, and E. Mallada, "Implicit trajectory planning for feedback linearizable systems: A time-varying optimization approach," in *Proc. Amer. Control Conf.*, 2020, pp. 4677–4682.

[4] E. Hazan, "Introduction to online convex optimization," *Foundations Trends Optim.*, vol. 2, no. 3/4, pp. 157–325, 2016.

[5] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proc. IEEE*, vol. 108, no. 11, pp. 2032–2048, Nov. 2020.

[6] M. Nonhoff and M. A. Müller, "Online gradient descent for linear dynamical systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 945–952, 2020.

[7] Y. Li, X. Chen, and N. Li, "Online optimal control with linear dynamics and predictions: Algorithms and regret analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14887–14899.

[8] N. Agarwal, B. Bullins, E. Hazan, S. Kakade, and K. Singh, "Online control with adversarial disturbances," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, vol. 97, pp. 111–119.

[9] E. Hazan, S. Kakade, and K. Singh, "The nonstochastic control problem," in *Proc. 31st Int. Conf. Algorithmic Learn. Theory*, 2020, vol. 117, pp. 408–421.

[10] M. Nonhoff and M. A. Müller, "An online convex optimization algorithm for controlling linear systems with state and input constraints," in *Proc. Amer. Control Conf.*, 2021, pp. 2523–2528.

[11] Y. Li, S. Das, and N. Li, "Online optimal control with affine constraints," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, pp. 8527–8537.

[12] M. Simchowitz, K. Singh, and E. Hazan, "Improper learning for non-stochastic control," in *Proc. 33rd Conf. Learn. Theory*, 2020, vol. 125, pp. 3320–3436.

[13] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annu. Rev. Control*, vol. 52, pp. 42–64, 2021.

[14] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor, "A note on persistency of excitation," in *Proc. IEEE Conf. Decis. Control*, 2005, vol. 3, pp. 2630–2631.

[15] J. Coulson, J. Lygeros, and F. Dörfler, "Data enabled predictive control: In the shallows of the DeePc," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 307–312.

[16] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021.

[17] J. Berberich, A. Koch, C. W. Scherer, and F. Allgöwer, "Robust data-driven state-feedback design," in *Proc. Amer. Control Conf.*, 2020, pp. 1532–1538.

[18] A. Xue and N. Matni, "Data-driven system level synthesis," in *Proc. 3rd Conf. Learn. Dyn. Control*, 2021, vol. 144, pp. 189–200.

[19] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Automat. Contr.*, vol. 65, no. 3, pp. 909–924, Mar. 2020.

[20] J. Berberich, C. W. Scherer, and F. Allgöwer, "Combining prior knowledge and data for robust controller design," 2020, *arXiv:2009.05253v3*.

[21] H. J. van Waarde, M. K. Camlibel, and M. Mesbahi, "From noisy data to feedback controllers: Nonconservative design via a matrix s-lemma," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 162–175, Jan. 2022.

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

[23] S. Menta, A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Stability of dynamic feedback optimization with applications to power systems," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput.*, 2018, pp. 136–143.

[24] M. Colombino, E. Dall'Anese, and A. Bernstein, "Online optimization as a feedback controller: Stability and tracking," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 422–432, Mar. 2020.

[25] L. S. P. Lawrence, Z. E. Nelson, E. Mallada, and J. W. Simpson-Porco, "Optimal steady-state control for linear time-invariant systems," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 3251–3257.

[26] L. Cothren, G. Bianchin, and E. Dall'Anese, "Data-enabled gradient flow as feedback controller: Regulation of linear dynamical systems to minimizers of unknown functions," in *Proc. 4th Annu. Learn. Dyn. Control Conf.*, 2022, vol. 168, pp. 234–247.

[27] Z. He, S. Bolognani, J. He, F. Dörfler, and X. Guan, "Model-Free Nonlinear Feedback Optimization," 2022, *arXiv2201.02395*.

[28] G. Bianchin, M. Vaquero, J. Cortes, and E. Dall'Anese, "Online stochastic optimization for unknown linear systems: Data-driven synthesis and controller analysis," 2021, *arXiv:2108.13040v1*.

[29] M. Nonhoff and M. A. Müller, "Data-driven online convex optimization for control of dynamical systems," in *Proc. 60th IEEE Conf. Decis. Control*, 2021, pp. 3640–3645.

[30] M. A. Müller, D. Angeli, and F. Allgöwer, "On necessity and robustness of dissipativity in economic model predictive control," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1671–1676, Jun. 2015.

[31] L. Grüne and M. A. Müller, "On the relation between strict dissipativity and turnpike properties," *Syst. Control Lett.*, vol. 90, pp. 45–53, 2016.

[32] L. Grüne, S. Pirkelmann, and M. Stieler, *Strict Dissipativity Implies Turnpike Behavior for Time-Varying Discrete Time Optimal Control Problems*. Cham, Switzerland: Springer International Publishing, 2018, pp. 195–218.

[33] D. Angeli, R. Amrit, and J. B. Rawlings, "Receding horizon cost optimization for overly constrained nonlinear plants," in *Proc. 48h IEEE Conf. Decis. Control*, 2009, pp. 7972–7977.

[34] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Madison, Wisconsin, USA: Nob Hill Publishing, 2017.

[35] T. Faulwasser, L. Grüne, and M. A. Müller, "Economic nonlinear model predictive control," *Foundations Trends Syst. Control*, vol. 5, no. 1, pp. 1–98, 2018.

[36] J. B. Rawlings, D. Angeli, and C. N. Bates, "Fundamentals of economic model predictive control," in *Proc. IEEE 51st IEEE Conf. Decis. Control*, 2012, pp. 3851–3861.

[37] J. Berberich and F. Allgöwer, "A trajectory-based framework for data-driven system analysis and control," in *Proc. IEEE Eur. Control Conf.*, 2020, pp. 1365–1370.

[38] L. Eldén, "A weighted pseudoinverse, generalized singular values, and constrained least squares problems," *BIT*, vol. 22, pp. 487–502, 1982.

[39] Y. Li, G. Qu, and N. Li, "Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4761–4768, Oct. 2021.

[40] X. Zhang, W. Shi, X. Li, B. Yan, A. Malkawi, and N. Li, "Decentralized temperature control via HVAC systems in energy efficient buildings: An approximate solution procedure," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2016, pp. 936–940.

[41] Y. Nesterov, *Lectures on Convex Optimization* (Springer Optimization and Its Applications), 2nd ed., ser, vol. 137, 2018.

MARKO NONHOFF (Graduate Student Member, IEEE) received the master's degree in engineering cybernetics from the University of Stuttgart, Stuttgart, Germany, in 2018. Since then, he has been a Research Assistant with the Leibniz University Hannover, Hanover, Germany, where he is currently working toward the Ph.D. degree under the supervision of Prof. Matthias A. Müller. His research interests include learning-based control and online optimization.

MATTHIAS A. MÜLLER (Senior Member, IEEE) received the Diploma degree in engineering cybernetics in 2009 from the University of Stuttgart, Stuttgart, Germany, and the M.S. degree in electrical and computer engineering in 2009 from the University of Illinois at Urbana-Champaign, Champaign, IL, USA. In 2014, he received the Ph.D. degree in mechanical engineering from the University of Stuttgart, Germany, for which he received the 2015 European Ph.D. Award on control for complex and heterogeneous systems. Since 2019, he has been the Director of the Institute of Automatic Control and Full Professor with the Leibniz University Hannover, Hanover, Germany. He received an ERC Starting Grant in 2020 and was the recipient of the inaugural Brockett-Willems Outstanding Paper Award for the best paper published in Systems and Control Letters in the period 2014–2018. His research interests include nonlinear control and estimation, model predictive control, and data-/learning-based control, with application in different fields including biomedical engineering.