Optimistic Online Non-stochastic Control via FTRL

Naram Mhaisen and George Iosifidis

Abstract— **This paper brings the concept of "optimism" to the new and promising framework of online Non-stochastic Control (NSC). Namely, we study how can NSC benefit from a prediction oracle of unknown quality responsible for forecasting future costs. The posed problem is first reduced to an optimistic learning with delayed feedback problem, which is handled through the Optimistic Follow the Regularized Leader (OFTRL) algorithmic family. This reduction enables the design of `OptFTRL-C`, the first Disturbance Action Controller (DAC) with optimistic policy regret bounds. These new bounds are commensurate with the oracle's accuracy, ranging from $\mathcal{O}(1)$ for perfect predictions to the order-optimal $\mathcal{O}(\sqrt{T})$ even when all predictions fail. By addressing the challenge of incorporating untrusted predictions into control systems, our work contributes to the advancement of the NSC framework and paves the way towards effective and robust learning-based controllers.**

I. Introduction

We study the NSC framework originally introduced in [1]: Consider a time-slotted dynamical system where at each time slot, the controller observes the system state $\boldsymbol{x}_t \in \mathbb{R}^{d_x}$ and decides an action $\boldsymbol{u}_t \in \mathbb{R}^{d_u}$ which then induces a cost $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$, and causes a transition to a new state $\boldsymbol{x}_{t+1}$. Note that the cost and the new state are revealed to the controller *after* it commits its action. Similar to [1], we study Linear Time Invariant (LTI) systems where the state transition is parameterized by matrices $A \in \mathbb{R}^{d_x \times d_x}$, $B \in \mathbb{R}^{d_x \times d_u}$ and a *disturbance* vector $\boldsymbol{w}_t \in \mathbb{R}^{d_x}$:

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{w}_t. \tag{1}$$

$\boldsymbol{w}_t$ can be arbitrarily set by an *adversary*, and we only restrict it to be upper-bounded, i.e., $\|\boldsymbol{w}_t\| \leq w, \forall t$. Similarly, the adversary is allowed to select any l-Lipschitz convex cost function $c_t : \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \mapsto \mathbb{R}$ at each slot. Since neither the cost nor disturbances are confined to follow a fixed and/or known distribution, the control problem is "Non-stochastic".

The controller (or learner) aims to find a (possibly time-varying) policy that maps states to actions, $\pi : \boldsymbol{x} \mapsto \boldsymbol{u}$, from a given policy class Π, such that the cost trajectory $\{c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)\}_{t=1}^T$ is as small as possible. To quantify the learner's performance, we employ the *policy regret* metric, as presented in [2]. Intuitively, the policy regret measures the accumulated differences between the cost incurred by the learner's policy, and that of a stationary unknown cost-minimizing policy designed with access to all future cost functions and disturbances:

$$\mathcal{R}_T \doteq \sum_{t=1}^{T} c_t\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) - \min_{\pi \in \Pi} \sum_{t=1}^{T} c_t\big(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi)\big), \tag{2}$$

where $(\boldsymbol{x}_t(\pi), \boldsymbol{u}_t(\pi))$ is the *counterfactual* state-action sequence that would have emerged under the benchmark policy, whereas $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ is the *actual* state-action sequence that emerged from following possibly different policies by the learner. A sub-linear regret means that the cost endured by the learner will converge to that of the optimal unknown policy at the same sub-linear rate, i.e., ${}^{\mathcal{R}_T}\!/_{T} \rightarrow 0$ as $T \rightarrow \infty$. Note that the benchmark depends on the actual *witnessed* costs and disturbances, encoding a stronger concept of "adaptability" unlike the pessimistic $\mathcal{H}_\infty$ controllers, which assume worst case costs [3]. Remarkably, the Gradient Perturbation Controller (GPC) achieves $\mathcal{O}(T^{1/2})$ regret, which is order optimal, and was shown to deliver superior performance to the more conventional $\mathcal{H}_2$ controllers [1].

In contrast to the typical NSC framework, here we consider the existence of a *prediction oracle*. This oracle provides the learner, before committing to the action $\boldsymbol{u}_t$, with a forecast for the as-yet-unobserved cost functions $\{c_\tau(\cdot,\cdot)\}_{\tau=t}^{t+d}$ for a specific horizon of d slots. We denote the oracle's outcome as $\tilde{c}_\tau(\cdot,\cdot)$. Notably, the predictions themselves can be influenced by the adversary, meaning that no assumptions on the oracle's *accuracy* are made; the forecasted parameters may either deviate arbitrarily from the truth, or be actually accurate in the best case.

The motivation for this addition on the NSC framework stems from the abundance of machine learning forecasting models, which provide high potential improvement if they are adequately accurate. The effect of such predictions on the classical regret metric (not the policy regret) has been studied in the literature of optimistic online learning, where the ultimate objective is to provide regret guarantees that scale with the accuracy of predictions while always staying sub-linear. Namely: $\mathcal{R}_T = \mathcal{O}(1)$ when all predictions are accurate and $\mathcal{O}(\sqrt{T})$ in all cases. That is, we are assured of reaching the performance of the benchmark policy, yet at a significantly improved rate when predictions happen to be precise. Hence, optimistic online learning represents a highly desirable combination of the best of both worlds: optimal worst-case guarantees with achievable best case guarantees. In fact, optimistic learning algorithms have been attracting considerable attention as the driving force behind recent state-of-the-art results in online constrained optimization [4], online discrete optimization [5], online sub-modular optimization [6], and online fairness [7] to name a few.

Unfortunately, such optimistic algorithms are yet to find their way to online NSC. This might be surprising given that previous NSC results were obtained through a streamlined reduction to the standard Online Convex Optimization (OCO) framework [8]. Hence, one might anticipate that optimistic

N. Mhaisen and G. Iosifidis are with *the Faculty of Electrical Engineering, Mathematics and Computer Science. Delft University of Technology. The Netherlands.* {n.mhaisen, g.iosifidis}@tudelft.nl.

NSC algorithms can be derived using a similar approach. Interestingly, this is not the case. Combining optimistic learning and NSC poses unique challenges for both frameworks. **First**, existing optimistic learning algorithms do not consider cost functions *with memory* and hence cannot handle states. Particularly, these algorithms update their regularizers at each time slot based on the prediction accuracy of the preceding slot's cost [9], [10]. However, in stateful systems, an action made at t will have an effect that spans across all slots until $t+d$, and thus uses predictions for all these slots. Since the accuracy of such multi-step predictions is not available until $t+d+1$, we cannot update the regularizer in the same standard way. **Second**, the guarantee of NSC is established via a reduction to the OCO with Memory (OCO-M) framework [11], which in turn is reduced to the standard OCO [12] via the concept of slowly moving decision variables [1, Thm. 4.6] [11, Thm. 3.1]. This later reduction cannot be utilized in optimistic learning where accurate predictions lead to little or *no* regularization, driving consecutive decisions of the optimistic algorithm to vary arbitrarily (up to the decision set diameter) [10, Sec 2.2], [13, Sec 7.4].

This paper tackles exactly these challenges and aims to answer the question: *is it possible to design an online learning algorithm whose policy regret is commensurate with the accuracy of an exogenous prediction oracle, while always staying sub-linear?* Our paper answers this positively and builds upon recent advances in online learning, introducing, to our knowledge, the first optimistic controller for NSC.

We achieve such optimistic guarantees by departing from the standard analysis approach of reducing the learner's non-stationary policy to a stationary one [1], [14], [15], and instead directly analyzing the non-stationary policy. Specifically, to address the first challenge, we demonstrate that the additive separability of the linearized costs allows expressing the costs as a sum of *memoryless* but *delayed* functions of each of the decision variables. Next, to tackle the second challenge, we analyze the performance of each decision variable *separately* via an alternative reduction to the framework of "optimism with delay" [16]. Nonetheless, we customize this later framework with a specific "hint" design that exploits the structure of NSC where the cost is indeed delayed but still *gradually* being revealed at each step, leading to tighter bounds. We make these intuition-focused points concrete in our upcoming analysis.

The main contribution is designing the first optimistic controller with policy regret that scales from $\mathcal{O}(1)$ to $\mathcal{O}(\sqrt{T})$, depending on the predictions' accuracy. The methodology is based on a new perspective on stateful systems as systems with delayed feedback, for which we build upon recent results on delay and optimism. The next section reviews the related works. Sec. III provides the necessary background for our new algorithm, `OptFTRL-C`, introduced in Section IV. We then present numerical examples in Sec. V and conclude.

II. Related Work

The NSC problem was initiated in the seminal work of [1], which introduced the first controller with sub-linear policy regret for dynamical systems, generalizing the classical control problem to adversarial convex costs functions and adversarial disturbances. These results were further refined for *strongly* convex functions [17], [18], [19]; and systems where the actions are subject to fixed or adversarially-changing constraints [20], [21]. Follow-up works also looked at the NSC problem under more general assumptions on the system matrices, including unknown (A, B) [2], systems with bandit feedback [22], and time-varying systems [23]. Expectedly, the regret bounds deteriorate in these cases, e.g., becoming $\mathcal{O}(T^{2/3})$ for unknown systems and $\mathcal{O}(T^{3/4})$ for systems with bandit feedback. Efficiency is also investigated in [15] with projection-free methods. Going beyond the typical bounds in terms of T, [24] introduced an adaptive FTRL based controller whose bound is proportional to the witnessed costs and perturbations: $\mathcal{O}((\sum_{t=1}^{T} g_t^2)^{1/2})$ instead of $\mathcal{O}(T^{1/2})$, where g_t depends on the witnessed costs and perturbations. These works do not consider the existence of an untrusted prediction oracle, as the model adopted here.

The NSC framework was also investigated using other metrics such as dynamic regret [14], adaptive regret [25], and competitive ratio [26], [27]. For dynamic and adaptive regret, methods with static regret guarantees, as discussed here, are used as building blocks for "meta" algorithms with static/adaptive guarantees [23], [28]. For competitive ratio, it was demonstrated in [29] that a regret guarantee against the optimal DAC policy automatically implies a competitive ratio with an additive sub-linear term. Hence, the presented algorithm is still highly relevant even for other metrics.

Our work is also related to the robust MPC literature [30], [31], where (possibly inaccurate) predictions are used. However, the main difference is that while our algorithm is robust to bad predictions, it is not designed based on them. Specifically, our benchmark policy changes when worst case costs and predictions are not witnessed (as dictated by the regret metric). A newer line of research studies the MPC-style algorithms under the regret metric [32], [33]. Perfect predictions are assumed in [32], whose bound was later generalized in [33] with parameterized predictions error. While these works do not confine their policy class (i.e., directly optimizing $\{\boldsymbol{u}_t\}_t$), their regret is linearly dependent on the prediction error, which renders them suffer linear regret in the worst case scenario.

Optimism is an extension to the OCO framework that allows incorporating predictions of future costs without having the regret bound linearly depend on their quality. The optimistic learning framework, in its current prevalent form, originated in [9] where an Online Mirror Descent (OMD) algorithm was presented. Thereafter, optimism was studied under another equivalent[1] framework (FTRL). Optimistic FTRL (OFTRL) has since undergone multiple improvements [13], [35] and we refer the reader to [36, Sec. 7.12] for comprehensive overview. Only recently, a close connection between delay and optimism emerged [16], a development we leverage in our current analysis of the NSC framework.

[1] see [34, Sec. 6] for a discussion on the link between OMD and FTRL.

While optimistic learning has been studied for stochastic predictions [37], [38], and adversarial predictions [10], [5], these findings have not been applied to dynamical systems. In dynamical systems, the study of predictions is limited to either *perfect* predictions [39], [40], or *fixed* quadratic (thus, strongly convex) cost functions [41], [42], [43]. Our paper contributes towards filling this gap. Predictions may also be viewed via the lens of "context" [44] in the different problem of *stochastic* MDPs with *finite* state and action space. Lastly, the previous works of [43], [42] consider that the full predictions are provided *a priori*, meaning that the learner cannot benefit from updated (and possibly more accurate) predictions. Like MPC, we drop this assumption and allow the learner to use the most recent available predictions.

III. Preliminaries

Notation. We denote scalars by small letters, vectors by bold small letters, and matrices by capital letters. Time indexing is done via a subscript. We denote by $\{\boldsymbol{a}_t\}_{t=1}^T$ the set $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_T\}$. $M = [M^{[i]}|M^{[j]}]$ denotes the augmentation of $M^{[i]}$ and $M^{[j]}$. $\|\cdot\|$ denotes the ℓ_2 norm for vectors and the Frobenius norm for matrices. $\|\cdot\|_*$ is the dual norm.$\langle\cdot,\cdot\rangle$ is the dot product for vectors and the Frobenius product for matrices. $\|\cdot\|_{\text{op}}$ is the matrix spectral norm (the induced ℓ_2 norm). We use $h_{a:b}$ to indicate $\sum_{s=a}^b h_s$ when s is irrelevant, and $f(\cdot)$ when the function's argument are irrelevant.

DAC policy class. The class Π under consideration in this paper, and in the broader NSC literature, is the Disturbance Action Controllers (DAC) (see [45, Ch. 6] for a general reference). The motivation behind DAC lies in the combination of expressive power and efficient parametrization. Specifically, DAC can approximate the large class of linear controllers, which, for instance, is guaranteed to include the universally optimal controller in the case of stochastic disturbances and quadratic costs (LQR settings). Simultaneously, the use of DAC actions has been demonstrated to induce *convex* cost functions in its parametrization.

A policy $\pi \in \Pi$, with a memory length of p, is parameterized by p matrices $M \doteq \left[M^{[1]}|M^{[2]}|\ldots|M^{[j]}|\ldots|M^{[p]}\right]$, $M^{[j]} \in \mathbb{R}^{d_u \times d_x}$, and a fixed stabilizing controller K. We define the set $\mathcal{M} \doteq \{M : \|M\| \leq \kappa_M\}$, with the standard bounded variable assumption. The action at a step t according to a policy $\pi_t \in \Pi$ is then calculated as follows:

$$\boldsymbol{u}_t = K\boldsymbol{x}_t + \sum_{j=1}^{p} M_t^{[j]} \boldsymbol{w}_{t-j}. \tag{3}$$

Note that K is a stabilizing controller calculated a priori and provided as an input. This is formalized through the *strong stability* assumption [46, Def. 3.1], which is standard in NSC. It ensures that the DAC controller is supplied with a controller K such that $\|(A+BK)^t\|_{\text{op}} \leq \kappa(1-\delta)^t$ for $\delta \in (0,1]$, where $\kappa > 0$. Here, we make a slightly stricter assumption, enabling us to simplify the analysis:

Assumption 1: The system (A, B) is intrinsically stable: $\|A\|_{\text{op}} \leq 1-\delta$ for some $\delta \in (0,1]$.

The above assumption allows us to satisfy the strong stability assumption with K being the zero matrix. Otherwise, we can revert to the strong stability assumption itself. This simplification facilitates the analysis without much loss in generality, as discussed in, for example, [25, Remark 4.1]. Additionally, we assume $\|B\| \leq \kappa_B$. The boundedness of $\|A\|$ follows from its spectral norm bound.

When using DAC policies, it is known that the state at $t+1$ can be described as a linear transformation of the parameters chosen by the learner in the previous t slots $M_1, M_2, \ldots, M_t$:

$$\begin{aligned}\boldsymbol{x}_{t+1} &= \sum_{i=0}^{t} A^i \left(B \sum_{j=1}^{p} \left(M_{t-i}^{[j]}\ \boldsymbol{w}_{t-i-j} \right) + \boldsymbol{w}_{t-i} \right) \\ &= \sum_{i=0}^{t} A^i \big(B M_{t-i}\ \overline{\boldsymbol{w}}_{t-i-1} + \boldsymbol{w}_{t-i} \big),\end{aligned} \tag{4}$$

where we defined $\overline{\boldsymbol{w}}_{t-i-1} \doteq (\boldsymbol{w}_{t-i-1}, \ldots, \boldsymbol{w}_{t-i-p})$ so as to express the vector $\sum_{j=1}^p M_t^{[j]} \boldsymbol{w}_{t-j}$ compactly as $M_t \overline{\boldsymbol{w}}_{t-1}$. The above expression for the state can be obtained by simply unrolling the dynamic[2] in (1). This is proven, e.g., in [45, Lem. 7.3] and [25, Lem. 4.3].

Cost functions. While our guarantees continue to hold for the general convex cost function, we assume here that the cost functions are *linearized*:

Assumption 2: The cost is linear in the state and action $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) = \langle \boldsymbol{\alpha}_t, \boldsymbol{x}_t \rangle + \langle \boldsymbol{\beta}_t, \boldsymbol{u}_t \rangle$.

The linearity assumption provides a useful structure in our analysis (separability) and enables us to quantify the prediction error in terms of the parameters $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$. It does not, however, compromise the presented regret guarantees. Specifically, our bounds continue to hold for the general convex case. In fact, the linear costs are the *most challenging*[3] in the online learning settings, and hence the regret caused by a sequence of general convex function can be indeed *upper bounded* by the regret caused by *linearization* of those functions. Thus, online learning works often focused on the linear case (see discussion on linearization in [34, Sec. 2.1] and [12, Sec. 2.4]). Future costs $\{c_\tau(\cdot,\cdot)\}_{\tau=t}^{t+d}$ can thus be predicted through the oracle output $\{\boldsymbol{\alpha}_\tau, \boldsymbol{\beta}_\tau\}_{\tau=t}^{t+d}$.

Predictions. The prediction model we use has several advantages over those that appear in the preceding section. (i) We allow the prediction oracle to update its forecast *at every decision slot* t (similar to MPC). This flexibility is important, as in practice predictions can improve with time. (ii) our analysis reveals that the parameter d only needs to scale logarithmically with T. This implies that predictions are not required for the entire future. In fact, the proposed algorithm accommodates any number of predictions by simply setting $\boldsymbol{\alpha}_\tau, \boldsymbol{\beta}_\tau$ to zero for steps where no prediction is available. In other words, our algorithm can operate with no predictions (an oracle that produces $\mathbf{0}$ for all t) or limited horizon predictions. (iii) the presented algorithm and its guarantees place no assumptions on the predictions' quality. To the best of our knowledge, this represents the most general prediction-based setting for online control.

[2] Assuming that the initial state $\boldsymbol{x}_1$ (before executing $M_1, \ldots, M_t$) is $\mathbf{0}$.

[3] As indicated in the preceding section, in the case of strongly convex costs, a refined and much tighter regret bound of $\mathcal{O}(\log(T))$ (compared to $\mathcal{O}(\sqrt{T})$) is possible [19].

IV. Online control with Optimistic FTRL

We introduce first few definitions to facilitate the presentation of the algorithm. Define the *forward cost* function:

$$F_t(M) \doteq \sum_{i=0}^{d} f^{(i)}_{t+i}(M)$$

where each $f^{(i)}_{t+i}(M)$ function shall describe the contribution of M to the cost experienced at slot $t+i$, and is defined as

$$f^{(i)}_t(M) \doteq \begin{cases} \langle \boldsymbol{\alpha}_t, \boldsymbol{\psi}^{i-1}_t(M)\rangle & \text{if } i \geq 1 \\ \langle \boldsymbol{\beta}_t, \boldsymbol{\psi}_t(M)\rangle & \text{if } i = 0 \end{cases} \tag{5}$$

with $\boldsymbol{\psi}^i_t : \mathbb{R}^{d_u \times (d_x p)} \mapsto \mathbb{R}^{d_x}$, and $\boldsymbol{\psi}_t : \mathbb{R}^{d_u \times (d_x p)} \mapsto \mathbb{R}^{d_u}$ being the following linear transformations, which are used to simplify the presentation of the action and state expression in (3) and (4), respectively:

$$\boldsymbol{\psi}^i_{t+1}(M) \doteq A^i(BM\overline{\boldsymbol{w}}_{t-i-1} + \boldsymbol{w}_{t-i}), \tag{6}$$

$$\boldsymbol{\psi}_t(M) \doteq M\overline{\boldsymbol{w}}_{t-1}, \tag{7}$$

The role of the functions in (5) will become clear later in the analysis. Roughly, the cost c_t will be expressed as a sum of them. Denoting by $G^{(i)}_t = \nabla_M f^{(i)}_t(M)$, we have:

$$G^{(i)}_t = \begin{cases} B^\top (A^{i-1})^\top \boldsymbol{\alpha}_t \overline{\boldsymbol{w}}^\top_{t-i-2} & \text{if } i \geq 1 \\ \boldsymbol{\beta}_t \overline{\boldsymbol{w}}^\top_{t-1} & \text{if } i = 0 \end{cases} \tag{8}$$

Note that $G^{(i)}_t$ is a $d_u \times d_x p$ matrix with the (m,n)-th element being the partial derivative of $f^{(i)}_t$ w.r.t. the (m,n)-th element of M. From the above, we can get the bounds

$$\|G^{(i)}_t\| \leq \begin{cases} \alpha\kappa_B p w(1-\delta)^{i-1} & \doteq g^{(i)} \quad \text{if } i \geq 1 \\ \beta p w & \doteq g^{(0)} \quad \text{if } i = 0 \end{cases} \tag{9}$$

using that for any matricies A,B, and vector $\boldsymbol{w}$ $\|AB\| = \|A\|\|B\|$ and $\|A\boldsymbol{w}\| \leq \|A\|_{\text{op}}\|\boldsymbol{w}\|$. We also define the prediction $\tilde{G}^{(i)}_t$, which we can construct by plugging the oracle's output in (8), and hence we have the *partial* prediction error:

$$\Delta^{(i)}_t \doteq \|G^{(i)}_t - \tilde{G}^{(i)}_t\| \tag{10}$$

We highlight that from (9), the magnitude of prediction error decrease exponentially with i: $\Delta^{(i)}_t \propto (1-\delta)^i \leq e^{-i}$. We refer to i therefore as the attenuation level.

Similarly, we define G_t as the matrix of partials of $F_t(\cdot)$, with its prediction as $\tilde{G}_t$, Finally, we define the hybrid hint matrix H_t, which aims to approximate the sum $G_{t-d:t}$

$$H_t \doteq \underbrace{\Bigg\{ \sum_{i=0}^{d-1} \sum_{j=0}^{d-i-1} G^{(j)}_{t-d+i+j}}_{\text{available at } t} \underbrace{+ \sum_{j=d-i}^{d} \tilde{G}^{(j)}_{t-d+i+j} \Bigg\} \mathbb{1}_{d\geq 1} + \tilde{G}_t}_{\text{future predictions}} \tag{11}$$

We denote by Δ_t the prediction error $\Delta_t \doteq \|G_{t-d:t} - H_t\|$. Due to the definition of $F_t(\cdot)$, certain elements of the summands (in $G_{t-d:t}$) are partially observed by t and are hence directly used in constructing H_t. The remaining elements in the sum are obtained from the prediction oracle.

With these definitions at hand, we can now introduce the main algorithmic step. We propose an algorithm (OptFTRL-C) for optimizing the policy parameters M_t, $t \in [T]$. The algorithm uses the update formula:

Algorithm 1 OptFTRL-C

```
Input: System $(A, B)$, parameter $d$, DAC parameters $\kappa_M, p$.
Output: Policy parameters $M_t$ at each slot $t = 1, \dots, T$.
1: for each time slot $t = 1, \dots, T$ do
2:     Use action $\boldsymbol{u}_t = \sum_{j=1}^{p} M^{[j]}_t \, \boldsymbol{w}_{t-j}$
3:     Observe cost $c_t(\boldsymbol{x_t}, \boldsymbol{u_t})$ and record the gradient $G_{t-d}$
4:     Observe new state $\boldsymbol{x_{t+1}}$ and record $\boldsymbol{w}_t$
5:     Calculate $\Delta_{t-d}$ and update parameter $\lambda_{t+1}$ via (13)
6:     Receive future predictions $\{c_\tau(\cdot,\cdot)\}_{\tau=t}^{t+d}$
7:     Construct $H_{t+1}$ as in (11)
8:     Calculate $M_{t+1}$ via (12)
9: end for
```

$$M_{t+1} = \underset{M \in \mathcal{M}}{\operatorname{argmin}} \left\{ \langle G_{1:t-d} + H_{t+1}, M\rangle + r_{t+1}(M) \right\}, \tag{12}$$

where $r_t(\cdot)$ are strongly convex regularizers defined as:

$$r_{t+1}(M) = \frac{\lambda_{t+1}}{2}\|M\|^2,$$

$$\lambda_{t+1} = \frac{4}{\kappa_M} \max_{j \leq t-d-1} \Delta_{j-d+1:j} + \frac{\sqrt{5}}{\kappa_M}\sqrt{\sum_{i=1}^{t-d} \Delta_i^2}. \tag{13}$$

In essence, the regularizer is a λ_{t+1}-strongly convex function, where λ_{t+1} is proportional to the observed prediction error up to t. At each t, we set the strong convexity to be the maximum sum of d consecutive witnessed errors, added to the root of the accumulated squared witnessed errors. This later term is aligned with memoryless OFTRL, whereas the former is necessary to adjust for the memory/delay effect.

The steps of the OptFTRL-C routine are outlined in Algorithm 1. OptFTRL-C first executes an action $\boldsymbol{u}_t$ (line 2). Then, the cost function is revealed, completing the necessary information to compute G_{t-d} (line 3, recall that all the costs from $t-d, \dots, t$ are required to know G_{t-d}). The system then transitions to state $\boldsymbol{x}_{t+1}$, effectively revealing the disturbance vector $\boldsymbol{w}_t$ (line 4). At this point, we can calculate Δ_{t-d} (since we know the ground truth G_{t-d}) and update the strong convexity parameter accordingly (line 5). Then, the oracle forecasts the next d costs functions (line 6), enabling us to construct the hybrid hint matrix (line 7). Finally, the next action $\boldsymbol{u}_{t+1}$ is committed through updating the policy parameters (line 8). The regret of this OptFTRL-C is characterized in the following theorem:

***Theorem** 1: Let (A, B) be an LTI system with the memory parameter d defined as in Lemma 1. Let $\{c_t(\cdot,\cdot)\}_{t=1}^T$, $\{\boldsymbol{w}\}_{t=1}^T$ be any sequence of costs and disturbances, respectively. Let $\Delta^{(i)}_t$ be the prediction error for these sequences at time t with attenuation level i, as defined in* (10)*. Then, under Assumptions* 1 *and* 2*, algorithm OptFTRL-C produces actions $\{\boldsymbol{u}_t\}_{t=1}^T$ such that for all T, the following holds:*

$$\mathcal{R}_T = \mathcal{O}\left(\sqrt{\sum_{t=1}^{T} \Bigg(\sum_{i=0}^{d} \sum_{j=i}^{d} \Delta^{(d-j+i)}_{t+i} \Bigg)^2} \right).$$

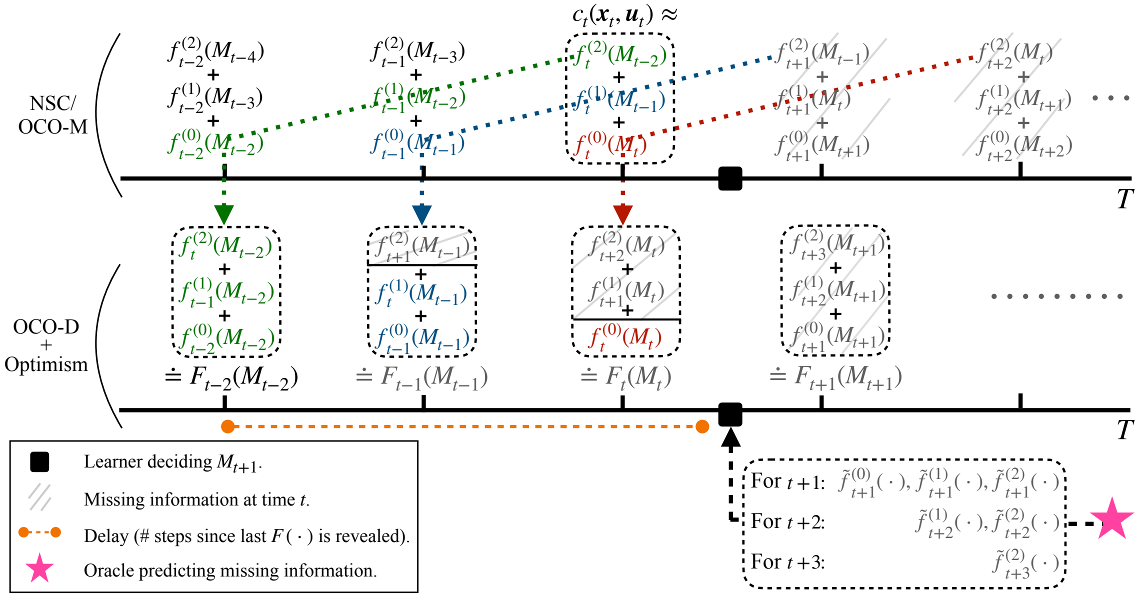

$c_t(\boldsymbol{x}_t,\boldsymbol{u}_t) \approx$
NSC/
OCO-M
$f_{t-2}^{(2)}(M_{t-4}) + f_{t-2}^{(1)}(M_{t-3}) + f_{t-2}^{(0)}(M_{t-2})$
$f_{t-1}^{(2)}(M_{t-3}) + f_{t-1}^{(1)}(M_{t-2}) + f_{t-1}^{(0)}(M_{t-1})$
$f_{t}^{(2)}(M_{t-2}) + f_{t}^{(1)}(M_{t-1}) + f_{t}^{(0)}(M_{t})$
$f_{t+1}^{(2)}(M_{t-1}) + f_{t+1}^{(1)}(M_{t}) + f_{t+1}^{(0)}(M_{t+1})$
$f_{t+2}^{(2)}(M_{t}) + f_{t+2}^{(1)}(M_{t+1}) + f_{t+2}^{(0)}(M_{t+2})$
T
OCO-D
+
Optimism
$f_{t}^{(2)}(M_{t-2}) + f_{t-1}^{(1)}(M_{t-2}) + f_{t-2}^{(0)}(M_{t-2}) \doteq F_{t-2}(M_{t-2})$
$f_{t+1}^{(2)}(M_{t-1}) + f_{t}^{(1)}(M_{t-1}) + f_{t-1}^{(0)}(M_{t-1}) \doteq F_{t-1}(M_{t-1})$
$f_{t+2}^{(2)}(M_{t}) + f_{t+1}^{(1)}(M_{t}) + f_{t}^{(0)}(M_{t}) \doteq F_{t}(M_{t})$
$f_{t+3}^{(2)}(M_{t+1}) + f_{t+2}^{(1)}(M_{t+1}) + f_{t+1}^{(0)}(M_{t+1}) \doteq F_{t+1}(M_{t+1})$
T
Learner deciding M_{t+1}.
Missing information at time t.
Delay (# steps since last $F(\cdot)$ is revealed).
Oracle predicting missing information.
For $t+1$: $\tilde{f}_{t+1}^{(0)}(\cdot), \tilde{f}_{t+1}^{(1)}(\cdot), \tilde{f}_{t+1}^{(2)}(\cdot)$
For $t+2$: $\tilde{f}_{t+2}^{(1)}(\cdot), \tilde{f}_{t+2}^{(2)}(\cdot)$
For $t+3$: $\tilde{f}_{t+3}^{(2)}(\cdot)$

Fig. 1: Our methodology in designing `OptFTRL-C`. Up: The NSC to OCO-M reduction, with parameter $d = 2$ (Sec. IV-A). Down: an equivalent *delayed OCO* formulation, obtained via rearrangement, which we append with an oracle (Sec. IV-B).

Discussion. `OptFTRL-C` achieves the sought-after accuracy-modulated regret bound that holds for systems with memory. It strictly generalizes previous optimistic online learning bounds by incorporating memory (hence states), and strictly generalizes previous online non-stochastic control bounds by handling predictions of unknown quality. Namely, `OptFTRL-C`'s bound has the following characteristics.

Prediction-commensurate: in the *best* case predictions ($\Delta_t^{(i)} = 0, \forall t$), the bound collapses to $\mathcal{O}(1)$, which is *constant*. On ther other hand, in the *worst* case, we get

$$\sum_{i=0}^{d}\sum_{j=i}^{d}\Delta_{t+i}^{(d-j+i)} \leq 2\sum_{i=0}^{d}\sum_{j=i}^{d} g^{(d-j+i)} = 2\sum_{i=0}^{d}\sum_{k=i}^{d} g^{(k)} = 2\sum_{k=0}^{d}\sum_{i=0}^{k} g^{(k)} = 2\sum_{k=0}^{d}(k+1)g^{(k)} = 2g^{(0)}+2\sum_{k=1}^{d}(k+1)g^{(k)} \leq 2\beta p w + \frac{2(1+\delta)\alpha\kappa_B p w}{\delta^2} \doteq m$$

where the first equality follows from the triangular inequality and (9), and in the last inequality we used $\sum_{i=0}^{\infty} i(1-\delta)^i \leq {}^{1-\delta}/_{\delta^2}$ and $\sum_{i=0}^{\infty}(1-\delta)^i \leq {}^{1}/_{\delta}$. Hence, the regret becomes $\mathcal{O}(m\sqrt{T})$, which is order-optimal in T [36, Sec 5.1], achieving the optimistic premise.

Memory-commensurate: Apart from prediction adaptivity, `OptFTRL-C`'s performance is interpretable with respect to the spectrum of stateless to stateful systems. Consider the stateless case ($\boldsymbol{x}_t = \boldsymbol{0}, \forall t$). Hence, $c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) = \langle \boldsymbol{\beta}_t, \boldsymbol{u}_t \rangle$. In this case, `OptFTRL-C` requires predictions only for the next step (as per (11)), The resulting bound becomes $\mathcal{O}((\sum_{t=1}^{T}\Delta_t^{0})^{1/2}) = \mathcal{O}((\sum_t \|\boldsymbol{\beta}_t - \tilde{\boldsymbol{\beta}}\|_t)^{1/2})$, recovering the optimistic bound for *stateless* online learning [10].

On the other hand, consider a system with a general memory d. Then, `OptFTRL-C` uses predictions not only for the next step, but for the next $d+1$ steps $\{\boldsymbol{\beta}_\tau, \boldsymbol{\alpha}_t\}_{\tau=t}^{t+d}$, as dictated by (11). However, the dependence on future predictions' accuracy decays exponentially (recall that $\Delta_t^{(i)} \propto (1-\delta)^i$). For example, when $d = 1$ the resulting bound is $\mathcal{O}\big((\sum_{t=1}^{T}\Delta_t^{(0)} + \Delta_t^{(1)} + \Delta_{t+1}^{(1)})^{1/2}\big)$. I.e., we pay for the error in $d+1$ predictions[4], but with an exponentially decaying rate.

Now that we have described `OptFTRL-C` and characterized its policy regret, we present the main tools necessary to prove Theorem 1. Our proof is structured into two primary parts. First, we demonstrate that the regret in linearized NSC is a specific instance within the OCO-M framework, achieved through a particular selection of separable functions (sub-section IV-A, visualized in the upper timeline of Fig. 1). Second, we establish that the regret of OCO-M with separable functions is in turn a particular case within the Delayed OCO (OCO-D) framework with a specific structure of delay (sub-section IV-B, visualized in the lower timeline of Fig. 1). While the first part is fairly standard in NSC, we do not reduce its resulting OCO-M instance to standard OCO, but to OCO-D instead.

A. NSC with linearized costs is separable OCO-M.

In this subsection, we show how the cost at each time slot t can be approximated by the sum of a finite number of functions of only the past d decisions. Formally:

Lemma 1: Given $d \geq \frac{1}{\delta}\log(\frac{z}{\delta\epsilon}T)$, $z \doteq w(\kappa_B\kappa_M p + 1)$. Then for any $\epsilon > 0$, the following holds

$$\sum_{t=1}^{T}\left| c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - \sum_{i=0}^{d} f_t^{(i)}(M_{t-i}) \right| \leq l\epsilon.$$

Proof: Define the counterfactual state $\hat{\boldsymbol{x}}_t$ as the state reached starting from $\boldsymbol{0}$ and then executing d DAC actions based on policies $M_{t-d}, M_{t-d+1}, \ldots, M_{t-1}$. In other words,

[4] The fact that earlier errors get repeated is due to the compounding effect.

this is the state reached at t by following the learner's policies but assuming that $\boldsymbol{x}_{t-d-1}$ was $\mathbf{0}$. From (4):

$$\hat{\boldsymbol{x}}_t = \sum_{i=0}^{d-1} A^i\big(BM_{t-i-1}\,\overline{\boldsymbol{w}}_{t-i-2} + \boldsymbol{w}_{t-i-1}\big). \qquad (14)$$

For now, assume that $\|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\| \le \frac{\epsilon}{T}$. Then, by Lipschitzness:

$$\sum_{t=1}^{T} |c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - c_t(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t)| < l\epsilon.$$

However, $c_t(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t)$ can be written in terms of $f_t^{(i)}(\cdot)$ using Assumption 1 and the definitions in (6) and (7) :

$$\begin{aligned}
c_t(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t) &= \langle \boldsymbol{\alpha}_t, \sum_{i=0}^{d-1} \boldsymbol{\psi}_t^i(M_{t-i-1})\rangle + \langle \boldsymbol{\beta}_t, \boldsymbol{\psi}_t(M_t)\rangle \\
&= \sum_{i=0}^{d-1} \langle \boldsymbol{\alpha}_t, \boldsymbol{\psi}_t^i(M_{t-i-1})\rangle + \langle \boldsymbol{\beta}_t, \boldsymbol{\psi}_t(M_t)\rangle \\
&= \sum_{i=1}^{d} \langle \boldsymbol{\alpha}_t, \boldsymbol{\psi}_t^{i-1}(M_{t-i})\rangle + \langle \boldsymbol{\beta}_t, \boldsymbol{\psi}_t(M_t)\rangle \\
&= \sum_{i=1}^{d} f_t^{(i)}(M_{t-i}) + \langle \boldsymbol{\beta}_t, \boldsymbol{\psi}_t(M_t)\rangle = \sum_{i=0}^{d} f_t^{(i)}(M_{t-i}).
\end{aligned}$$

It remains to show that $\|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\| \le \frac{\epsilon}{T}$. From (4) and (14), we have $\|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\| \le$

$$\begin{aligned}
&\|\sum_{i=d}^{t-1} A^i\big(BM_{t-i-1}\overline{\boldsymbol{w}}_{t-i-2} + \boldsymbol{w}_{t-i-1}\big)\| \\
&\le \sum_{i=d}^{t-1} (\|A^i BM\|_{op}\|\|\overline{\boldsymbol{w}}_{t-i-2}\| + \|A^i\|_{op}\|\boldsymbol{w}_{t-i-1}\|) \\
&\overset{(a)}{\le} \sum_{i=d}^{t-1} \|A^i\|_{op}\|BM\|\|\overline{\boldsymbol{w}}_{t-i-2}\| + \|A^i\|_{op}\|\boldsymbol{w}_{t-i-1}\| \\
&\overset{(b)}{\le} \sum_{i=d}^{t-1} (1-\delta)^i w(\kappa_B\kappa_M p + 1) \le z\int_{i=d}^{\infty} e^{-\delta i} di = \frac{z}{\delta}e^{-\delta d},
\end{aligned}$$

where (a) follow from the sub-multipilicitive property of $\|\cdot\|_{\text{op}}$, and $\|B\|_{\text{op}} \le \|B\|$, and (b) from $1 - x \le e^{-x}$. Substituting d, makes the last term $\frac{\epsilon}{T}$. ■

The closeness of $\boldsymbol{x}_t$ and $\hat{\boldsymbol{x}}_t$ is a known fact in NSC, and it is due to the (strong) stability assumption. Lastly, we note that it is possible to set d adaptively without knowledge of T by using $d = \frac{1}{\delta}\log(\frac{zt}{\delta\epsilon})$.[5]

B. separable OCO-M is OCO-D

Now that we have approximated the cost at t by a separable function with memory, we show in this subsection that the separated functions can be rearranged to represent an equivalent OCO with delayed feedback formulation.

***Lemma** 2:* Let $f_t(M_0, \dots, M_d)$ be a separable function: $f_t(M_0, \dots, M_d) = \sum_{i=0}^{d} f_{t,i}(M_i)$, $f_{t,i}(M) : \mathbb{R}^{(d_u \times d_x p)} \mapsto \mathbb{R}$. Let $\mathcal{A}$ be an online learning algorithm whose decisions M_{t+1} depend on the history set $\mathbb{H}_t \doteq \{\cup_{i=0}^{d} f_{\tau,i}(\cdot)\}_{\tau=1}^{t}$. Define $J_t(M) \doteq \sum_{i=0}^{d} f_{t+i,i}(M)$, and the history set w.r.t. $J_t(\cdot)$ as $\mathbb{H}_t^J$ Then,

$$\begin{aligned}
\mathbb{H}_t^J &= \{J_\tau(\cdot)\}_{\tau=1}^{t-d}, \text{ and} &(15)\\
\sum_{t=1}^{T} f_t(M_{t-d}, \dots, M_t) &= \sum_{t=1}^{T} J_t(M_t). &(16)
\end{aligned}$$

(16) means that the accumulated cost, with memory, is equivalent to that of the memoryless function $J_t(\cdot)$. However, from (15) $\mathcal{A}$ has delayed feedback w.r.t. $J_t(\cdot)$; when deciding M_{t+1}, feedback up to only $t - d$ is available.

Proof: the first part is immediate from the definition of $J_\tau(\cdot)$; any $J_\tau(\cdot)$ with $\tau > t - d$ would require a function that is not in the original history set $\mathbb{H}_t$.

The second part is mainly index manipulation:

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(M_{t-d}, \dots, M_t) = \sum_{t=1}^{T}\sum_{i=0}^{d} f_{t,i}(M_{t-i}) \\
&= \sum_{i=0}^{d}\sum_{t=1}^{T-i} f_{t+i,i}(M_t) = \sum_{t=1}^{T}\sum_{i=0}^{d} f_{t+i,i}(M_t) = \sum_{t=1}^{T} J_t(M_t)
\end{aligned}$$

Where the first equality holds by separability, the second by shifting the sum index and using the convention $f_\tau(M_{t<1}) \doteq 0, \forall\tau$, the third by using $f_{t>T}(\cdot) \doteq 0$. ■

Now we are ready to prove Theorem 1: *Proof:* Denote with $\pi^\star$ the cost-minimizing policy, and let $M^\star$ be its parametrization. Then, from (2):

$$\begin{aligned}
\mathcal{R}_T &= \sum_{t=1}^{T} c_t(\boldsymbol{x}_t, \boldsymbol{u}_t) - c_t(\boldsymbol{x}_t(\pi^\star), \boldsymbol{u}_t(\pi^\star)) \\
&\overset{(a)}{\le} \sum_{t=1}^{T} \big(\sum_{i=0}^{d} f_t^{(i)}(M_{t-i}) - \sum_{i=0}^{d} f_t^{(i)}(M^\star)\big) + 2l\epsilon \\
&\overset{(b)}{=} \sum_{t=1}^{T} f_t(M_{t-d}, \dots, M_t) - f_t(M^\star, \dots, M^\star) \\
&\overset{(c)}{=} \sum_{t=1}^{T} F_t(M_t) - F_t(M^\star) \doteq \mathcal{R}_T^F, \qquad (17)
\end{aligned}$$

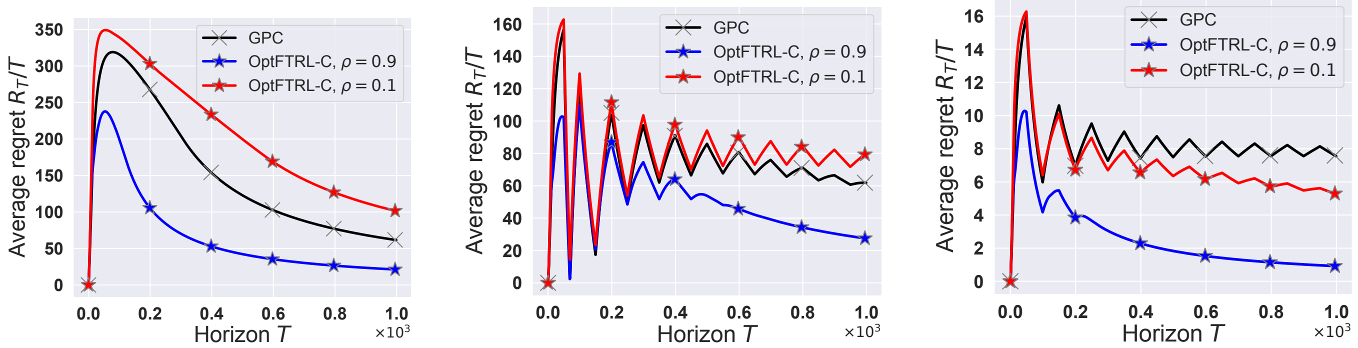
where (a) follows by Lemma 1, which gives both an upper bound on the learner cost and lower bound on the benchmark's cost, (b) by writing the sum of $f_t^{(i)}(\cdot)$ as a single function with memory, and (c) by Lemma 2, with $f_{t,i}(M_{t-i}) = f_t^{(i)}(M_{t-i})$, and hence, $J_t(M) = F_t(M)$. To bound the delayed feedback regret $\mathcal{R}_T^F$, we use [16, Thm. 11], which we restate below using the notation of this paper:[6]

***Theorem** 2 ([16, Thm. 11]):* Let λ_t be non-decreasing on t defined as in (13). Let $r(\cdot)$ be λ_t strongly convex regularizer. Then, the sum in (17), can be bounded as:

$$\mathcal{R}_T^F \le 8\kappa_M \max_{t\in[T]} \Delta_{t-d:t-1} + 2\sqrt{5}\kappa_M\sqrt{\sum_{t=1}^{T}\Delta_t^2}.$$

[5] This would result in a sum of the form $\sum_t 1/t \le \log(T)$. We leave this part for the extended version and, inline with previous works such as [15], consider here the d is sufficiently large to have the approximation error ϵ as a negligible constant w.r.t. T.

[6] Mapping their notation to ours, we have $\boldsymbol{g}_{t-D:t} = G_{t-d:t}$, $\boldsymbol{h}_t = H_t$, $\boldsymbol{g}_{t-D:t} - h_t = \Delta_t$, $\boldsymbol{a}_{t,F} = 2\kappa_M\Delta_t$, $\boldsymbol{b}_{t,F} = 1/2\Delta_t^2$, and $\alpha = \kappa_M^2$.



(a) Stationary costs.

(b) Alternating costs.

(c) Alternating costs, low magnitude.

Fig. 2: The average regret against the optimal policy under various scenarios (cost and disturbances trajectories).

Now, note that the sum $G_{t-d:t}$ can be written as:

$$G_{t-d:t} = \sum_{i=0}^{d} \sum_{j=0}^{d} G^{(j)}_{t-d+i+j}$$

hence we get that $\Delta_t = \|G_{t-d:t} - H_t\| \leq$

$$\sum_{i=0}^{d-1} \sum_{j=d-i}^{d} \|G^{(j)}_{t-d+i+j} - \tilde{G}^{(j)}_{t-d+i+j}\| + \sum_{j=0}^{d} \Delta^{(j)}_{t+j}$$

$$= \sum_{i=0}^{d} \sum_{j=d-i}^{d} \Delta^{(j)}_{t-d+i+j} = \sum_{i=0}^{d} \sum_{j=0}^{i} \Delta^{(j+d-i)}_{t+j}$$

$$= \sum_{i=0}^{d} \sum_{j=i}^{d} \Delta^{(d-j+i)}_{t+i}$$

substituting gives the result. ■

V. Numerical Example

Recall that `OptFTRL-C` was designed to take advantage of a prediction oracle that forecasts future cost functions with unknown accuracy. Theorem 1 then demonstrated that the average policy regret $\mathcal{R}_T/T$ of `OptFTRL-C` always converges to 0, but does so faster for accurate predictions. We therefore plot the policy regret of `OptFTRL-C` when provided with either accurate or inaccurate predictions. The implementation code of the policies `OptFTRL-C`, GPC, and the benchmark $\pi^\star$, as well as the code necessary to reproduce all experiments, is available at the GitHub repository [47].

We consider a system with $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{R}^2$, $p = d = 10$,[7] and hence $M \in \mathbb{R}^{2\times 20}$. The dynamics are $A = 0.9 \times \boldsymbol{I}_2, B = \boldsymbol{I}_2$, with perturbation $\boldsymbol{w}_t \in [-1,1]^2$ of maximum magnitude of $w = \sqrt{2}$. We consider a linear cost $c_t = \langle \boldsymbol{\alpha}_t, \boldsymbol{x}_t \rangle$, with $\boldsymbol{\alpha}_t \in [-1,1]^2$ and hence $\alpha = \sqrt{2}$. With these choices, we have the upper bound on the gradient $\|G_t\| \leq {}^{\alpha\kappa_B p w}/_{0.1} \leq 300$.

In the accurate prediction case, we set $\tilde{c}_t(\cdot,\cdot) = c_t(\cdot,\cdot)$ with probability $\rho = 0.9$. Otherwise, we set $\tilde{c}_t(\cdot,\cdot)$ to be uniformally random ($\boldsymbol{\alpha}_t \in [-1,1]$). Hence, ρ represents the probability of correctly predicting $c_t(\cdot,\cdot)$, and we sample it at every slot. For inaccurate prediction, we set $\rho = 0.1$.

	GPC	`OptFTRL-C` $\rho = 0.9$	`OptFTRL-C` $\rho = 0.1$	Optimal
Scenario (a)	$314,694$	$355,061$	$274,965$	$376,198$
Scenario (b)	$34,994$	$69,468$	$16,934$	$96,869$
Scenario (c)	642	$7,276$	$2,928$	$8,191$

TABLE I: Accumulated reward (negative cost) $-\sum_{t=1}^{T} c_t(\boldsymbol{x}_t, \boldsymbol{u}_t)$ of the different policies.

We compare with GPC, which was shown to outperform the classical $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controllers in a wide range of situations[8]. *In scenario (a)*, The cost trajectory is set as $\boldsymbol{\alpha}_t = (1,1), \boldsymbol{w}_t = (1,1), \forall t$. This represents a simple case where the cost function does not fluctuate. It can be seen from Fig. 2a that `OptFTRL-C` provides the expected acceleration when the prediction is accurate, achieving an average of 60.2% smaller $\mathcal{R}_T/T$ value compared to GPC. At the same time, the average regret still *attenuates* at the same $\mathcal{O}(1/\sqrt{T})$ rate even in the case of inaccurate predictions, but with an average performance degradation of 47.3%.

In scenario (b), we deploy *an alternating* cost function. Namely, $\boldsymbol{\alpha}_t$ alternate between $(1,1)$ and $(-0.5,-0.5)$ every 50 steps. The disturbances are still $\boldsymbol{w}_t = (1,1), \forall t$. This alternating cost represents an adversarial fluctuation in the cost trajectory, and it is where the non-stochastic framework demonstrates its efficacy. Namely, this fluctuation is enough to violate the guarantees of $\mathcal{H}_2$ controllers, but at the same time, it is small in magnitude, rendering $\mathcal{H}_\infty$ controller overly pessimistic. It is worth noting that the fluctuation observed in Fig. b is attributed to the update rule of GPC. This update directly modifies the decision variables M_{t+1} based on the observed cost. In contrast, FTRL aggregates both past and future costs to determine M_{t+1}. With accurate predictions, this method does not induce much fluctuation as it foresees the upcoming small disturbance. Overall, `OptFTRL-C` with good prediction achieves an improvement of 32% in $\mathcal{R}_T/T$ value over GPC, while having a 13.4% degradation when fed with the inaccurate oracle.

In scenario (c) we also deploy an alternating cost function

[7] The DAC parameter p and the cost's memory d are denoted h and H in [1], where they are also set equal. While both are commonly referred to as "memory", we refer to d also as the delay due to the duality we presented.

[8] For additional details on the performance of GPC vs traditional ones, see the experiments in the tutorial [48], and its associated codebase [49].

but with different lower magnitudes. Namely, $\boldsymbol{\alpha}_t$ alternate between $(0.1, 0.1)$ and $(-0.5, -0.5)$ every 50 steps, and $w_t = (0.1, 0.1)$. The goal of this scenario is to show that `OptFTRL-C` can have an advantage over GPC even regardless of the prediction quality through *adapatablity* to "easy" environments (i.e., environments with small gradients). In general, GPC performance takes a hit since its learning rate is tuned with the upper bound for the gradient (i.e., $\alpha = w = \sqrt{2}, T = 10^3$). The alternating frequency, on the other hand, contributes to distinguishing more the effect of good predictions. In this scenario `OptFTRL-C` achieves an improvement of 16.8% and 69.8% over GPC for $\rho = 0.1$ and $\rho = 0.9$, respectively. In summary, `OptFTRL-C` leverages predictions without sacrificing its resilience to their inaccuracy, or to the costs' adversity.

VI. Conclusion

This paper looked at the NSC problem with the aim of designing DAC controllers that can leverage predictions of unknown quality. The overreaching goal is to have a controller with meaningful policy regret guarantees that are accelerated by good predictions but not void when these predictions fail. By looking at OCO-M from the lens of Delayed OCO, we were able to present the first optimistic DAC controller. Our work paves the way for further research in the ongoing pursuit of data and learning-driven control.

References

[1] N. Agarwal, B. Bullins, E. Hazan, S. Kakade, and K. Singh, "Online control with adversarial disturbances," in *Proc. of ICML*, 2019.

[2] E. Hazan, S. Kakade, and K. Singh, "The nonstochastic control problem," in *Proc. of ALT*, 2020.

[3] A. Karapetyan, A. Iannelli, and J. Lygeros, "On the regret of h∞ control," in *Proc. of IEEE CDC*, 2022.

[4] D. Anderson, G. Iosifidis, and D. J. Leith, "Lazy lagrangians for optimistic learning with budget constraints," *IEEE/ACM Trans. on Networking*, vol. 31, no. 5, pp. 1935–1949, 2023.

[5] N. Mhaisen, A. Sinha, G. Paschos, and G. Iosifidis, "Optimistic no-regret algorithms for discrete caching," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 3, pp. 1–28, 2022.

[6] T. Si-Salem, G. Özcan, I. Nikolaou, E. Terzi, and S. Ioannidis, "Online submodular maximization via online convex optimization," in *Proc. of AAAI*, 2024.

[7] F. Aslan, G. Iosifidis, J. A. Ayala-Romero, A. Garcia-Saavedra, and X. Costa-Perez, "Fair resource allocation in virtualized o-ran platforms," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 8, no. 1, 2024.

[8] E. Hazan, "Introduction to Online Convex Optimization," *arXiv:1909.05207*, 2019.

[9] A. Rakhlin and K. Sridharan, "Online learning with predictable sequences," in *Proc. of COLT*, 2013.

[10] M. Mohri and S. Yang, "Accelerating Online Convex Optimization via Adaptive Prediction," in *Proc. of AISTATS*, 2016.

[11] O. Anava, E. Hazan, and S. Mannor, "Online learning for adversaries with memory: Price of past mistakes," in *Proc. of NeurIPS*, 2015.

[12] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, 2012.

[13] P. Joulani, A. György, and C. Szepesvári, "A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, and variational bounds," in *Proc. of COLT*, 2017.

[14] P. Zhao, Y.-X. Wang, and Z.-H. Zhou, "Non-stationary online learning with memory and non-stochastic control," in *Proc. of AISTATS*, 2022.

[15] H. Zhou, Z. Xu, and V. Tzoumas, "Efficient online learning with memory via frank-wolfe optimization: Algorithms with bounded dynamic regret and applications to control," in *Proc. of CDC*, 2023.

[16] G. E. Flaspohler, F. Orabona, J. Cohen, S. Mouatadid, M. Oprescu, P. Orenstein, and L. Mackey, "Online learning with optimism and delay," in *Proc. of ICML*, 2021.

[17] M. Simchowitz, "Making non-stochastic control (almost) as easy as stochastic," in *Proc. of NeurIPS*, 2020.

[18] N. Agarwal, E. Hazan, and K. Singh, "Logarithmic regret for online control," in *Proc. of NeurIPS*, 2019.

[19] D. Foster and M. Simchowitz, "Logarithmic regret for adversarial online control," in *Proc. of ICML*, 2020.

[20] Y. Li, S. Das, and N. Li, "Online optimal control with affine constraints," in *Proc. of the AAAI*, 2021.

[21] X. Liu, Z. Yang, and L. Ying, "Online nonstochastic control with adversarial and static constraints," *arXiv:2302.02426*, 2023.

[22] P. Gradu, J. Hallman, and E. Hazan, "Non-stochastic control with bandit feedback," in *Proc. of NeurIPS*, 2020.

[23] P. Gradu, E. Hazan, and E. Minasyan, "Adaptive regret for control of time-varying dynamics," in *Proc. of L4DC*, 2023.

[24] N. Mhaisen and G. Iosifidis, "Adaptive online non-stochastic control," *arXiv:2310.02261*, 2023.

[25] Z. Zhang, A. Cutkosky, and I. Paschalidis, "Adversarial tracking control via strongly adaptive online learning with memory," in *Proc. of AISTATS*, 2022.

[26] G. Shi, Y. Lin, S.-J. Chung, Y. Yue, and A. Wierman, "Online optimization with memory and competitive control," in *Proc. of NeurIPS*, 2020.

[27] G. Goel and B. Hassibi, "Competitive control," *IEEE Trans. Autom. Control*, vol. 68, no. 9, pp. 5162–5173, 2023.

[28] M. Simchowitz, K. Singh, and E. Hazan, "Improper learning for non-stochastic control," in *Proc. of COLT*, 2020.

[29] G. Goel, N. Agarwal, K. Singh, and E. Hazan, "Best of both worlds in online control: Competitive ratio and policy regret," in *Proc. of L4DC*, 2023.

[30] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 2007.

[31] G. E. Dullerud and F. Paganini, *A course in robust control theory: a convex approach*. Springer Science & Business Media, 2013, vol. 36.

[32] Y. Lin, Y. Hu, G. Shi, H. Sun, G. Qu, and A. Wierman, "Perturbation-based regret analysis of predictive control in linear time varying systems," in *Proc. of NeurIPS*, 2021.

[33] Y. Lin, Y. Hu, G. Qu, T. Li, and A. Wierman, "Bounded-regret mpc via perturbation analysis: Prediction error, constraints, and nonlinearity," in *Proc. of NeurIPS*, 2022.

[34] H. B. McMahan, "A Survey of Algorithms and Analysis for Adaptive Online Learning," *J. Mach. Learn. Res.*, vol. 18, no. 1, 2017.

[35] N. Mhaisen, G. Iosifidis, and D. Leith, "Online caching with no regret: Optimistic learning via recomm." *IEEE Trans. Mob. Comput.*, 2023.

[36] F. Orabona, "A Modern Introduction to Online Learning," *arXiv:1912.13213*, 2023.

[37] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. Andrew, "Online convex optimization using predictions," in *Proc. of SIGMETRICS*, 2015.

[38] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman, "Using predictions in online optimization: Looking forward with an eye on the past," *SIGMETRICS Perform. Eval. Rev.*, vol. 44, no. 1, 2016.

[39] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman, "The power of predictions in online control," in *Proc. of NeurIPS*, 2020.

[40] Y. Li, X. Chen, and N. Li, "Online optimal control with linear dynamics and predictions: Algorithms and regret analysis," in *Proc. of NeurIPS*, 2019.

[41] R. Zhang, Y. Li, and N. Li, "On the regret analysis of online lqr control with predictions," in *Proc. of ACC*, 2021.

[42] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman, "Competitive control with delayed imperfect information," in *Proc. of ACC*, 2022.

[43] T. Li, R. Yang, G. Qu, G. Shi, C. Yu, A. Wierman, and S. Low, "Robustness and consistency in linear quadratic control with untrusted predictions," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 1, 2022.

[44] O. Levy and Y. Mansour, "Optimism in face of a context: Regret guarantees for stochastic contextual mdp," in *Proc. of AAAI*, 2023.

[45] E. Hazan and K. Singh, "Introduction to online nonstochastic control," *arXiv:2211.09619*, 2022.

[46] A. Cohen, A. Hasidim, T. Koren, N. Lazic, Y. Mansour, and K. Talwar, "Online linear quadratic control," in *Proc. of ICML*, 2018.

[47] Optimistic-NSC. [Online]. Available: https://github.com/Naram-m/Optimistic-NSC

[48] ICML 2021 NSC tutorial. [Online]. Available: https://icml.cc/virtual/2021/tutorial/10838

[49] NSC-tutorial - code/experiments. [Online]. Available: https://sites.google.com/view/nsc-tutorial/codeexperiments