Adaptive Control of Unknown Time Varying Dynamical Systems
with Regret Guarantees

Dee San Muthirayan  Ruijie Du  Yanning Shen  Pramod P. Khargonekar
Univ. of California, Irvine  Univ. of California, Irvine  Univ. of California, Irvine  Univ. of California, Irvine

Abstract

The study of online control of unknown time varying dynamical systems is a relatively under-explored topic. In this work, we present regret guarantee with respect to a stronger notion of system variability compared to Minasyan et al. (2021) and thus provide sub-linear regret guarantee for a much broader range of scenarios. Specifically, we give regret guarantee with respect to the number of changes compared to the average squared deviation of Minasyan et al. (2021). The online control algorithm we propose continuously updates its estimate to track the changes and employs an online optimizer to simultaneously optimize the control policy. We show that our algorithm can achieve a sub-linear regret with respect to the number of changes under two settings: (i) matched disturbance system with general convex cost functions, (ii) general system with linear cost functions. Specifically, a regret of $\Gamma^{1/5}T^{4/5}$ can be achieved, where $\Gamma_T$ is the number of changes in the underlying system and $T$ is the duration of the control episode.

1 Introduction

Control of systems with uncertainties is a central theme in control theory and has been extensively researched. There are various sub-fields in control such as stochastic control (Kumar and Varaiya, 2015; Åström, 2012), robust control (Skogestad and Postlethwaite, 2007; Zhou et al., 1996) and adaptive control (Sastry and Bodson, 2011; Ioannou and Sun, 2012) that address controller analysis and synthesis for different types of uncertainties. The robust control literature typically focuses on the problem of feedback control with unmodeled dynamics in a worst-case setting, while the adaptive control literature studies the control of systems with parametric uncertainty (Sastry and Bodson, 2011). Typically, these classical approaches are concerned with stability. Robust control focuses on worst-case performance while adaptive control focuses on asymptotic performance. There is also a substantial body of literature on combinations of robust and adaptive control formulations. These comments are somewhat generalized as there are numerous papers dealing with many variations on these themes.

Recently, there has been a new line of research in the control of uncertain systems from the perspective of finite-time performance. Such a control setting is broadly termed as online control, borrowing the notion from online learning, where a learner’s performance is assessed from its ability to learn from a finite number of samples. Online control has made significant progress in the recent years, with algorithms for unknown systems, adversarial cost functions and disturbances (Dean et al., 2018; Cohen et al., 2019; Minasyan et al., 2019; Agarwal et al., 2019a,b; Simchowitz et al., 2020), algorithms for known systems with some predictive ability of future disturbances (Li et al., 2019; Yu et al., 2020; Lin et al., 2021), and for unknown systems with predictive ability (Lale et al., 2021; Muthirayan et al., 2021). A very detailed literature review is deferred to Appendix A. Most of the previous works focus on the minimization of the regret, which is defined as the difference between the cumulative cost incurred by the online algorithm and the best policy from a certain class of policies. The regret is a measure of the rate at which the cost difference of a given policy from that of the best policy grows over time. The regret serves as a good indicator of whether the online control algorithm converges to the best policy and the rate at which it converges, and therefore has been widely used in the study and design of online algorithms.

In many practical circumstances, the underlying system may be time varying. This could be a result of the changes in the parameters of the environment, or changes in the parameters of the system. In most of online control literature, a typical assumption in online control has been that the systems are time-invariant. Very recently, Minasyan et al. (2021) explored the problem of online control for unknown time-varying linear dynamical systems. They present some...
impossibility results and sub-linear regret algorithms, with a guarantee of $O(T\sigma + T^{2/3})$, where $\sigma$ is the square root of the average squared deviation of the system parameters. Although [Minasyan et al.] (2021) present adaptive regret guarantees, the achievability of sub-linear regret in their case is limited to scenarios within a certain number of changes in the underlying system. Motivated by this observation, we study the following questions: Can we achieve sub-linear regret with respect to a stronger notion of variability like the number of changes, which includes a broader set of scenarios? Under what system, information and cost structures assumptions can we achieve such guarantees?

**Contribution:** Different from prior works in online control, which study the control of time invariant dynamical systems, the present paper studies the problem of control of a time varying dynamical system over finite time horizon. The dynamical system we consider is a linear dynamical system with arbitrary disturbances, whose system matrices can be time varying. We specifically address the question of how to learn online and optimize when the system matrices are unknown in addition to the cost functions and disturbances being arbitrary and unknown apriori. Our goal is to design algorithms with regret guarantees with respect to stronger notions of variability, such as the number of changes. Towards this end, we consider the full information structure, where in addition to the cost feedback at the end of a time step, the controller also receives feedback of the disturbance, which is equivalent to the Disturbance Action Control (DAC) structure in [Minasyan et al.] (2021). Our key contribution is an online control algorithm that can adapt to the time variations irrespective of the magnitude or the frequency of the variations with high probability regret guarantees.

The proposed algorithm adapts to the variation by utilizing the data of transitions it observes "on the fly" to determine whether a change has occurred or not, and at the same time computes an estimate of the system parameters at the current time. The key here is how the algorithm is able to reliably track the changes, which is necessary for updating the control policy in a way that the control costs are convergent or the regret is sub-linear, as the accuracy of the control policy updates are limited by the accuracy of the estimates. We present guarantees for two settings: (i) matched disturbance system with general convex cost functions, (ii) general system with linear cost functions. We show that in both these settings a regret of $O\left(\sum_{t=1}^{T} T^{2/5} T^{4/5}\right)$ is achievable with a high probability, where $\Gamma_T$ is the number of times the system changes in $T$ time steps and $T$ is the duration of the control episode. Ours is the first regret guarantee with respect to number of changes in the system.

**Notation:** We denote the spectral radius of a matrix $A$ by $\rho(A)$, the discrete time interval from $m_1$ to $m_2$ by $[m_1, m_2]$, and the sequence $(x_{m_1}, x_{m_1+1}, ..., x_{m_2})$ compactly by $x_{m_1:m_2}$. Unless otherwise specified $\|\cdot\|$ is the 2-norm of a vector and Frobenious norm of a matrix. We use $O(\cdot)$ for the the standard order notation, and $\tilde{O}(\cdot)$ denotes the order neglecting the poly-log terms in $T$.

## 2 Problem Formulation

We consider a general linear time varying dynamical system given by

$$
\begin{align*}
x_{t+1} &= Ax_t + Bu_t + B_{t,w}w_t, \\
y_t &= Cx_t + e_t,
\end{align*}
$$

where $t$ denotes the time index, $x_t \in \mathbb{R}^n$ is the state at time $t$, $u_t \in \mathbb{R}^m$ is the control input at time $t$, $y_t \in \mathbb{R}^p$ is the output and observation at time $t$, $w_t \in \mathbb{R}^q$ is the disturbance at time $t$, $e_t \in \mathbb{R}^p$ is the measurement noise at time $t$, and $\theta_t = [A_t, B_t]$ are the time varying system matrices. The system parameters for any duration $T$, $\theta_{1:T}$, are assumed to be unknown. The disturbance $w_t$ could arise from unmodeled dynamics and thus need not be stochastic. For generality, we assume that the disturbances and measurement noise are bounded and arbitrary.

As is typical in control problems, the application of the control input $u_t$ results in a cost $c_t(y_t, u_t)$. Since the decisions are made online, the decision typically incurs a cost. In the above example, the cost is a function of user’s response to the recommendations. The incurred cost serves as feedback, using which the controller can update its policy before the next time step. The cost feedback can be broadly classified as: (i) full information feedback and (ii) bandit feedback. Full information feedback is the case where the feedback can be used to infer the full cost function for the time step and bandit feedback is the case where the feedback is just the incurred cost, which cannot be directly used to infer the cost function. Without loss of generality, we assume that the controller receives full information feedback. We believe our algorithms can be extended to the bandit feedback case just as in online control of time invariant systems [Gradu et al.] (2020).

The sequence of cost functions $c_t$ for the duration $T$ are assumed to be arbitrary and unknown a priori. Meanwhile, in addition to the output observation, the controller is assumed to observe the uncertainties, i.e., the full cost function $c_t$ and the disturbance $w_t$ after its action at $t$. Thus, a control policy has only the following information at a time $t$: (i) the observations $y_{1:t}$ till $t$, (ii) the control inputs it has applied on the system thus far $u_{1:t-1}$, and (iii) the cost functions $c_{1:t-1}$ and the disturbances $w_{1:t-1}$ observed thus far. Let’s denote the set of policies that satisfy this information setting by $\Pi_f$. With the disturbance feedback, the hardness of adaptation does not still diminish because compared to the known system case, where the disturbance $w_{t-1}$ can be calculated at $t$ using the state observation and the system equations, the adaptation also has to take in to account the unknown system variations. The disturbance feedback
The primary goal is to design a control policy that minimizes the regret for the stated control problem, i.e.,

\[ R_T(\pi) = \sum_{t=1}^{T} c_t(y^T_t, u^T_t) - \min_{\kappa \in \Pi_M} \sum_{t=1}^{T} c_t(y^\kappa_t, u^\kappa_t). \]  

The primary goal is to design a control policy that minimizes the regret for the stated control problem, i.e.,

\[ \min_{\pi \in \Pi_t} R_T(\pi). \]  

In many cases, since the regret minimization problem itself can be hard, the typical goal is to design a policy that achieves sub-linear regret scaling, i.e., a regret that scales as \( T^{\alpha} \) with \( T \), with \( \alpha < 1 \) that is minimal. Such a regret scaling implies that asymptotically the realized costs converge to that of the best policy from the comparator class. Our objective is to design an adaptive policy that can track the time variations and achieve sub-linear regret up to the variations in the system. We state the assumptions below.

**Assumption 1 (System).** (i) The system is stable, i.e., \( \|C_{t+k+1}A_{t+k} \ldots A_{t+1}B_t\|_2 \leq \kappa_a \kappa_b (1 - \gamma)^k, \quad \forall \; k \geq 0, \; \forall \; t, \) where \( \kappa_a > 0, \kappa_b > 0 \) and \( \gamma \) is such that \( 0 < \gamma < 1 \), and where \( \kappa_a, \kappa_b \) and \( \gamma \) are constants. \( B_t \) is bounded, i.e., \( \|B_t\| \leq \kappa_b \). (ii) The disturbance and noise \( w_t \) and \( v_t \) is bounded. Specifically, \( \|w_t\| \leq \kappa_w \), where \( \kappa_w > 0 \) is a constant, and \( \|v_t\| \leq \kappa_v \), where \( \kappa_v > 0 \) is a constant.

**Assumption 2 (Cost Functions).** (i) The cost function \( c_t \) is convex \( \forall \; t \). (ii) For a given \( z^T = [x^T, u^T], (z')^T = [(x')^T, (u')^T] \),

\[ \|c(x, u) - c(x', u')\| \leq LR \|z - z'\|, \]

where \( R = \max\{|\|z\|, \|z'\|, 1\} \).

(iii) For any \( d > 0 \), when \( \|x\| \leq d \) and \( \|u\| \leq d \), \( \nabla_x c(x, u) \leq Gd, \nabla_u c(x, u) \leq Gd \).

**Remark 1 (System Assumptions).** Assumption 1(i) is the equivalent of stability assumption used in time invariant systems. Such an assumption is typically used in online control when the system is unknown; see for eg., Simchowitz et al. (2020); Minasyan et al. (2021). Assumption 2(iii) that noise is bounded is necessary, especially in the non-stochastic setting (Agarwal et al. 2019a; Simchowitz et al. 2020). The assumption on cost functions is also standard (Agarwal et al. 2019a).

**Definition 1.** Setting (S-1): Matched disturbance system with convex cost functions: \( B_t = B_{t,w}, C = I, e_t = 0 \).

Setting (S-2): General system with linear cost functions: \( B_{t,w} = I, \) and there exists a coefficient \( \alpha_t \in \mathbb{R}^{p+m} \) such that \( c_t(y, u) = \alpha_t^T z, \|\alpha_t\| \leq G \).

### 3 Online Learning Control Algorithm

Prior online control algorithms use an exploration-first-then-exploit strategy, wherein, the learner performs only exploration for a period of time, following which it only optimizes the control using the observations and the high confidence estimate of the system matrices it computes by the end of the exploration phase. While this strategy results in sub-linear regret when the system is time invariant, it can fail in the time varying case. For example, consider the case where the system matrices remain unchanged during the exploration phase and then changes at the instant when exploration shifts to exploitation. In this case, the estimate computed at the end of exploration is only a high confidence estimate for the exploration phase and need not be a high confidence estimate for the exploitation phase. Thus, in this case, exploration-first-then-exploit can fail to achieve sub-linear regret.

We propose an online algorithm that continuously learns to compute an estimate of the time varying system parameters and that simultaneously optimizes the control policy online. The online algorithm runs an online optimization parallel to the estimation to optimize the parameters of the control policy. The control policy we employ is the Disturbance Action Control (DAC). The DAC policy is typically used...
in online control for regulating systems with disturbances; see [Agarwal et al. (2019a)]. The important feature of the DAC policy is that the optimization problem to find the optimal fixed disturbance gain for a given sequence of cost functions is a convex problem and thus amenable to online optimization and online performance analysis. A very appealing feature of DAC is that, for time invariant systems, the optimal disturbance action control for a given sequence of cost functions is very close in terms of the performance to the optimal linear feedback controller of the state; see [Agarwal et al. (2019a)]. Thus, for time invariant systems, by optimizing the DAC online, it is possible to achieve a sub-linear regret with respect to the best linear feedback controller of the state, whose computation is a non-convex optimization problem.

**Disturbance Action Control Policy:**

The Disturbance Action Control (DAC) policy is defined as the linear feedback of the disturbances up to a certain history $h$. We denote the DAC policy by $\pi_{DAC}$. Then, the control input $u_{t}^{\pi_{DAC}}$ under the policy $\pi_{DAC}$ is given by

$$u_{t}^{\pi_{DAC}} = \sum_{k=1}^{h} M_{t}^{[k]} w_{t-k}. \quad (4)$$

Here, $M = [M^{[1]}, \ldots, M^{[h]}]$ are the feedback gains or the disturbance gains and are the parameters of the DAC policy.

**Online Optimization:**

Given the information setting $\Pi_{t}$, the optimal parameters of $M$ cannot be computed apriori, since the optimal parameters are dependent on the specific realization of the cost function and the disturbances for all times. In this case, we can employ online optimization to continuously improve the policy parameter and compute the best estimate of the optimal parameter from the information observed till any time $t$. We call such a control policy the online DAC control policy and denote it by $\pi_{DAC-O}$. We denote the parameters estimated by $\pi_{DAC-O}$ by time $t$ by $\hat{M}_{t} = [\hat{M}_{t}^{[1]}, \ldots, \hat{M}_{t}^{[h]}]$. The control input $u_{t}^{\pi_{DAC-O}}$ under the policy $\pi_{DAC-O}$ is computed by assuming the estimate $\hat{M}_{t}$ as the best parameter. Thus, the control input $u_{t}^{\pi_{DAC-O}}$ under $\pi_{DAC-O}$ is given by

$$u_{t}^{\pi_{DAC-O}} = \sum_{k=1}^{h} \hat{M}_{t}^{[k]} w_{t-k}. \quad (5)$$

Since the cost function $c_{t}$ at any time step is a function of the state, the realized cost is dependent on the past control input and therefore has memory of the past control inputs. Therefore, we can use the Online Convex Optimization with Memory (OCO-M) framework to optimize the parameters $M_{t}^{[k]}$ online.

Consider the standard online convex optimization (OCO) setting (see [Hazan et al. (2008)]). OCO is a game played between a player who is learning to pick an action that minimizes its overall cost and an adversary who has the benefit of the knowledge of the player’s decision to pick a cost function so as to maximize the cost incurred by the player. At time $t$, the player chooses a decision $M_{t}$ from some convex subset $\mathcal{M}$, where $\max_{M \in \mathcal{M}} \|M\| \leq k_{M}$, and the adversary chooses a convex cost function $f_{t}(\cdot)$. As a result, the player incurs a cost $f_{t}(M_{t})$ for its decision $M_{t}$. The goal of the player is to minimize the regret over a duration $T$, which is given by

$$R_{T} = \sum_{t=1}^{T} f_{t}(M_{t}) - \min_{M \in \mathcal{M}} \sum_{t=1}^{T} f_{t}(M).$$

In the OCO-M setting, the difference is that the cost functions are dependent on a history of the past decisions and not just the current decision $M_{t}$. More specifically, the cost functions $f_{t}$ in OCO-M are a function of the decisions up to $h$ time steps, i.e., $u_{t:t-h}$, where $h$ is a given number. Thus, the regret in the OCO-M problem, in contrast to the OCO problem is the following:

$$R_{T} = \sum_{t=1}^{T} f_{t}(M_{t:t-h}) - \min_{M \in \mathcal{M}} \sum_{t=1}^{T} f_{t}(M).$$

The OCO-M framework works only when the dependence of the costs on $M_{t}$ is restricted to a finite and a fixed history $h$. In the control setting, the cost is a function of the output which is dependent on the full history of decisions $\Pi_{t}$. Let

$$G_{t} = [G_{t}^{[1]}, G_{t}^{[2]}, \ldots, G_{t}^{[h]}], \quad \tilde{G}_{t} = [\tilde{G}_{t}^{[2]}, \tilde{G}_{t}^{[2]}, \ldots, \tilde{G}_{t}^{[t-1]}],$$

$$G_{t}^{[k]} = C_{t} A_{t-1} \ldots A_{t-k+2} A_{t-k+1}, \forall k \geq 2, \quad \tilde{G}_{t}^{[1]} = C_{t} B_{t-1},$$

and $G_{t}^{[1]} = C_{t} B_{t-1}$. Thus, the history of dependence increases with $t$ and is not fixed. In order to apply the OCO-M framework, typically, a truncated output $\tilde{y}_{t}$ is constructed, whose dependence on the history of control inputs is limited to $h$ time steps:

$$\tilde{y}_{t}^{\pi_{DAC-O}}[M_{t:t-h}|G_{t}, s_{1:t}] = s_{t}^{h} + G_{t}^{[k]} u_{t-k}^{\pi_{DAC-O}},$$

where $s_{t} = y_{t} - \sum_{k=1}^{h-1} G_{t}^{[k]} u_{t-k}^{\pi_{DAC-O}}$.

Using the truncated output, a truncated cost function $\tilde{c}_{t}$ is constructed as

$$\tilde{c}_{t}(M_{t:t-h}|G_{t}, s_{1:t}) = c_{t}(\tilde{y}_{t}^{\pi_{DAC-O}}[M_{t:t-h}|G_{t}, s_{1:t}], u_{t}^{\pi_{DAC-O}}).$$
We state the performance of the algorithm OLC-FK formally. We denote the function \( \tilde{c}_t(M_{t:t-h} | G_t, s_{1:t}) \) succinctly by \( \tilde{c}_t(M | G_t, s_{1:t}) \) when each \( M_k \) in \( M_{t:t-h} \) is equal to \( M \). This denotes the (truncated) cost that would have been incurred had the policy parameter been fixed to \( M \) at all the past \( h \) time steps. 

A standard gradient algorithm for OCO-M framework updates the decision \( M_t \) by the gradient of the function \( f_t(M_{t:t-h}) \) with all \( M_k \) in \( M_{t:t-h} \) fixed to \( M_t \). Using the same compact notation as above, this gradient is equal to \( \partial f_t(M_t) \). An interpretation of this gradient is that, it is the gradient of the cost that would have been incurred had the policy parameter been fixed at \( M_t \) the past \( h \) time steps. We employ the same idea to update the policy parameters of the DAC policy online. The online optimization algorithm we propose updates the policy parameter \( M_t \) by the gradient of the cost function \( \tilde{c}_t(M_{t} | G_t, s_{1:t}) \) where each \( M_k \) in \( M_{t:t-h} \) is fixed to \( M_t \), i.e., as

\[
M_{t+1} = \text{Proj}_M \left( M_t - \eta \frac{\partial \tilde{c}_t(M_t | G_t, s_{1:t})}{\partial M_t} \right),
\]

where \( M \) is a convex set of policy parameters. The complete online optimization algorithm is given in Algorithm 1.

**Algorithm 1 Online Learning Control with Full Knowledge (OLC-FK) Algorithm**

**Input:** Step size \( \eta \), parameters \( \theta_{1:T} \).

**Initialize** \( M_1 \in M \) arbitrarily

**for** \( t = 1, \ldots, T \) **do**

- **Apply** \( u_{t}^{\text{DAC-O}} = \sum_{k=1}^{h} M_{t}^{[k]} u_{t-k} \)
- **Observe** \( c_{t}, u_{t} \) and **incur cost** \( \tilde{c}_t(M_{t}^{[k]} | G_t, s_{1:t}) \)
- **Update** \( M_{t+1} = \text{Proj}_M \left( M_t - \eta \frac{\partial \tilde{c}_t(M_t | G_t, s_{1:t})}{\partial M_t} \right) \)

**end**

**Main Result:**

We state the performance of the algorithm OLC-FK formally below. We consider the comparator class \( \Pi_M \) as the class of DAC policies whose disturbance gain are drawn from the set \( M \).

**Definition 3.**

\[
\tilde{D} := \max \left\{ \kappa_M \kappa_h h + \frac{\kappa_a \kappa_w}{\gamma} + \kappa_e + \frac{\kappa_a \kappa_h \kappa_M \kappa_w h}{\gamma} \right\},
\]

\[
L_f := L \tilde{D} \left( \kappa_a \kappa_h \kappa_w \sqrt{h} + \kappa_w \sqrt{h} \right).
\]

\[
G_f := G \tilde{D} \text{hnm} \left( \frac{\kappa_a \kappa_h \kappa_w}{\gamma} + \kappa_w \right).
\]

\[
D := \sup_{M_1, M_2 \in M} \| M_1 - M_2 \|
\]

**Theorem 1 (Full System Knowledge).** Suppose the setting is the general setting S-2 but the cost functions are general convex functions. Then, under Algorithm 2 with \( \eta = \frac{D}{\sqrt{G_f (G_f + L_f h^2) T^{1/2}}} \), the regret with the comparator class \( \Pi_M \) as the DAC policy class (Definition 2).

\[
R_T \leq O \left( D \sqrt{G_f (G_f + L_f h^2) T^{1/2}} \right).
\]

Please see Appendix G for the full proof.

### 3.1 Disturbance Action Control without System Knowledge

In the previous case, where the system parameters are known, the control policy parameters are optimized online through the truncated cost \( \tilde{c}_t(\cdot) \), whose construction explicitly utilizes the knowledge of the underlying system parameters \( G_{t}^{[k]} \). In this case, since the underlying system parameters are not available, we construct an estimate of the truncated state and the truncated cost by estimating the underlying system parameters \( G_{t}^{[k]} \). With this approach, the control policy will have to solve an online estimation problem to compute an estimate of the system parameters. Since the parameters are time variant, the online estimation has to be run throughout, unlike the other online estimation approaches. We describe in detail how our algorithm simultaneously performs estimation and optimizes the control policy.

**Online Estimation and Optimization:**

The Online Learning Control with Zero Knowledge (OLC-ZK) of the system parameters has two components: (i) a control policy and (ii) an online estimator that runs in parallel to the control policy and throughout the control episode. The control policy and online optimization algorithm is similar to the online algorithm except that the control policy parameters are updated through an estimate of the truncated cost function. The online estimation algorithm frequently re-estimates the underlying system parameters to track the underlying changes. We discuss the details of our algorithm below.

**A. Online Control Policy:**

We use the same notation for the control policy and the control input, i.e., \( \pi_{\text{DAC-O}} \) and \( u_{t}^\text{DAC-O} \) respectively. The estimation algorithm constructs an estimate \( \tilde{G}_{t}^{[k]} \) of the parameters \( G_{t}^{[k]} \) of the system in Eq. (11) for \( k \in [1, h] \). Thus, the estimation algorithm estimates \( G_{t}^{[k]} \) only for a truncated time horizon (looking backwards), i.e., for \( k \in [1, h] \). We describe the estimation algorithm later.

The policy \( \pi_{\text{DAC-O}} \) computes the control input \( u_{t}^\text{DAC-O} \)
(zero knowledge case) by combining two terms: (i) disturbance action control just as in the full knowledge case and (ii) a perturbation for exploration. In this case, we require an additional perturbation, just as in [Dean et al. (2018)], so as to be able to run the estimation parallel to the Online DAC, the control for regulating the cost. Let $\hat{u}_t^{DAC-O}[M_t|w_{1:t}] = \sum_{k=1}^{h} M_t[k] u_{t-k}$. Therefore, the total control input by $\pi^{DAC-O}$ is given by

$$u_t^{DAC-O} = \hat{u}_t^{DAC-O}[M_t|w_{1:t}] + \delta u_t^{DAC-O}. \quad (6)$$

As in [Dean et al. (2018)], we apply a Gaussian random variable as the perturbation, i.e.,

$$\delta u_t^{DAC-O} \sim \mathcal{N}(0, \sigma^2 I), \quad (7)$$

where $\sigma$ denotes the standard deviation, and is a constant to be specified later.

In this case the policy parameters are optimized by applying OCO-M on an estimate of the truncated cost. To construct this estimate, we construct an estimate of $s_t$ and the truncated state $\hat{s}_t^{DAC-O}$. Given that $s_t$ is the state response when the control inputs are zero, we estimate $s_t$ by subtracting the contribution of the control inputs from the observed state:

$$\hat{s}_t = \sum_{k=1}^{h} \hat{G}_t[k] u_{t-k} \quad (S-1)$$

$$\hat{s}_t = y_t^{DAC-O} - \sum_{k=1}^{h} \hat{G}_t[k] u_{t-k} \quad (S-2). \quad (8)$$

Then the estimate of the truncated output follows by substituting $\hat{s}_t$ in place of $s_t$ and using the estimated $\hat{G}_t$ in place of $G_t$:

$$\hat{y}_t^{DAC-O}[M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}] = \hat{y}_t^{DAC-O}[M_{t:t-h}][\hat{G}_t, \hat{s}_{1:t}] + \sum_{k=1}^{h} \hat{G}_t[k] \hat{u}_{t-k}^{DAC-O}. \quad (9)$$

Then, the estimate of the truncated cost is calculated as

$$\hat{c}_t(M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}) = c_t(\hat{y}_t^{DAC-O}[M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}], \hat{u}_t^{DAC-O}).$$

The online update to the policy parameters is just as in Algorithm 1, i.e., by the gradient of the estimate of the truncated cost

$$M_{t+1} = \text{Proj}_M \left( M_t - \eta \frac{\partial \hat{c}_t(M_t|\hat{G}_t, \hat{s}_{1:t})}{\partial M_t} \right).$$

A. Online Estimation:

The online estimation algorithm continuously updates its estimate of the parameters at certain intervals. Specifically, the online estimator re-estimates the underlying system parameters after every $t_p = N + 2h$ time steps. The updated estimate is the standard least-squares estimation applied to the data collected from the most recent interval of duration $t_p = N + 2h$. Here, $t_p$ has to be necessarily greater than $h$, since computing the estimate of $G_t$ requires at least a length of $h$ inputs. Essentially, the online estimator algorithm ignores the past and only considers the recent history to compute an estimate of the system parameters. This allows the online estimator to reliably estimate the current value of the parameters of the system provided $N$ is of adequate size and at the same time not very large.

We denote the index of the successive periods of duration $t_p$ by $k$. We denote the start and end time of each of these periods by $t_k^-$ and $t_k^+$. Therefore, it follows that $t_k^+ = t_k^- + 1$ for all $k$. The online estimator computes the following least-squares estimate at the end of each period $k$

$$\hat{G}_k = \arg \min_{\hat{G}} \sum_{p=t_k^-}^{t_k^+} \ell_p(\hat{G}) + \lambda \|\hat{G}\|^2,$$

$$\lambda > 0, \quad \ell_p(\hat{G}) = \left\| y_p^{DAC-O} - \sum_{l=1}^{h} \hat{G}[l] \hat{u}_{p-l}^{DAC-O} \right\|^2. \quad (9)$$

Given this estimate, the estimate for a given time $t$ is set as

$$\hat{G}_t = \hat{G}_k, \quad \forall \ t \ s.t \ t_k^- + 1 \leq t \leq t_k^+ + 1. \quad (10)$$

Definition 4 (System Parameter Set). $\mathcal{G} = \{G^{[1:h]} : \|G[k]\|_2 \leq \kappa_0 (1 - \gamma)^{k-1}\}$

Algorithm 2 Online Learning Control with Zero Knowledge (OLC-ZK) Algorithm

\begin{itemize}
  \item \textbf{Input:} Step size $\eta$, $\sigma$, $\beta$, $N$, $h$
  \item Initialize $M_1 \in \mathcal{M}$ arbitrarily, $t_d = 1, k = 1, t_s = 1, t_e = N + h$
  \item for $t = 1, \ldots, T$
    \begin{itemize}
      \item Observe $y_t^{DAC-O}$
      \item Compute $\hat{G}_t$ according to Eq. (10)
      \item Apply $u_t^{DAC-O}$ from Eq. (6)
      \item Observe $c_t$, $u_t$ and incur cost $c_t(y_t^{DAC-O}, u_t^{DAC-O})$
      \item Update: $M_{t+1} = \text{Proj}_M \left( M_t - \eta \frac{\partial \hat{c}_t(M_t|\hat{G}_t, \hat{s}_{1:t})}{\partial M_t} \right)$
    \end{itemize}
  \item \textbf{end}
\end{itemize}

Main Result:

The complete algorithm is shown in Algorithm 2. We state the performance of the algorithm OLC-ZK formally below. We consider the comparator class $\Pi_4$ as the class of DAC policies similar to the full knowledge case.

Definition 5 (Parameter Setting).

$$\eta = \frac{D}{\sqrt{G_f(G_f + L_f h^2)T}}, \quad h = \frac{\log T}{\log (1/(1 - \gamma))}.$$


\[ \beta = 2\sqrt{n} \zeta \left( n \log(2) + 2 \log \left( \frac{2h}{\delta} \right) + \frac{\lambda \kappa_n \kappa_h}{\gamma \delta \sigma \sqrt{n}} \right), \]

where \( \zeta = \left( R_s + \kappa_n \kappa_n \kappa_h + \kappa_n \kappa_h \right), R_s = \kappa_n \kappa_h + \kappa + 2h \kappa_n \kappa_h \) and \( \delta > 0 \) is a constant. \( N = \Gamma_T^{-0.8} T^{4/5}, \sigma = \Gamma_T^{-0.4} T^{-1/5}. \)

**Theorem 2** (Zero System Knowledge). Consider Algorithm 2 with the parameters given by Definition 5. Suppose \( T_2 \) with the parameters given by Definition 5. Suppose regret for the unknown system case, under Algorithm 2, using a proof technique similar to the time invariant case, knowledge of the number of changes. Theorem 2. Then, for \( T_2 \) with the parameters given by Definition 5, \( \Gamma_T = O(T^d), d < 1 \) and the setting is either S-1 or S-2. Then, for \( T \) greater than certain value, \( \delta \leq \frac{1}{T}, \delta \) small and \( \delta \leq \delta, \) the regret with the comparator class \( \Pi_M \) as the DAC policy class (Definition 2),

\[ R_T \leq O \left( \Gamma_T^{1/5} T^{4/5} \right) \]

with a probability greater than \( 1 - \delta. \)

Please see Appendix C and Appendix D for the full proof.

**Remark 2.** Minasyan et al. (2021) prove a regret bound that is \( O(T \sigma + T^{2/3}) \), where \( \sigma \) is the square root of the average squared deviation of \( G_t. \) The key difference compared to Minasyan et al. (2021) is that our result is sub-linear with respect to the number of changes \( \Gamma_T \) and not \( \sigma, \) and we present a regret bound that is \( O \left( \Gamma_T^{1/5} T^{4/5} \right). \) Minasyan et al. (2021) present an adaptive regret bound that is \( O \left( I |\sigma_I + T^{2/3} \right) \) for any interval \( I \) of length \( |I|, \) where \( \sigma_I \) is the root of the average squared deviation over the interval \( I. \) Although adaptive regret is a stronger regret bound, the achievability of sub-linear regret in their result is limited to scenarios where \( \Gamma_T = O(T^{1/3}). \) In contrast, we present sub-linear regret guarantee for \( \Gamma_T = O(T^d) \) for any \( d < 1. \)

**Remark 3.** Our algorithm assumes the knowledge of total number of changes, which is the case with most of the learning algorithms for the time varying scenario. It remains an open problem to design an algorithm that does not need the knowledge of the number of changes.

3.2 Intuition

Using a proof technique similar to the time invariant case, we can derive a regret bound in terms of the tracking error and the exploration cost. Specifically, we show that the regret for the unknown system case, under Algorithm 2 takes the form

\[ R_T \leq \text{Known System Regret} + \tilde{O}(T \sigma) + \tilde{O} \left( \sum_{t=1}^{T} \| G_t - \hat{G}_t \|_2 \right), \]

where “known system regret” refers to the term in the regret bound that is equivalent to the regret incurred when there is full knowledge about the system (Theorem 1). The perturbation that is applied and the fact that the estimated parameters are only approximate, results in an additional exploration cost term and a tracking error term. Here, the exploration cost term is proportional to the standard deviation of the perturbation applied to the DAC term in the control policy. It is clear that the exploration has to be balanced against detection and tracking accuracy.

Our key idea is the following. The changes in the underlying system broadly falls into two cases: (i) a change that happens in a quick succession, and (ii) and a change that happens after a sufficiently long period. We call such long periods as stationary intervals. We can see that, and this is what makes our algorithm work, that the online estimator can accurately estimate the underlying system parameter within any stationary interval provided the stationary interval is larger than \( \epsilon r_t. \) The duration between any two stationary intervals is a non-stationary interval where the changes happen in quick succession and therefore the estimator need not be accurate. Then, summing across all stationary and non stationary intervals gives the total cumulative error. The cumulative error is of course dependent on the frequency at which the parameter estimate is updated and the level of exploration. A large \( N \) will result in less frequent updates and a small \( N \) will result in too frequent updates. Therefore, the optimal \( N \) depends on the frequency of the variation of the underlying system itself. The final regret then follows from choosing an appropriate scaling for \( \sigma \) that balances the exploration cost and tracking accuracy and an optimal scaling for \( N. \) This approach crucially depends on the knowledge of the total number of changes over the duration \( T, \) which is a standard assumption, for eg., in bandit learning settings. It remains an open problem to design a algorithm that does not need the knowledge of \( \Gamma_T. \)

4 Experiments

Here, we present the numerical performance of a variant of our OLC-ZK algorithm that is observed to perform better in experiments. In this variant, instead of updating the estimate at certain intervals, the update is continuous and is only reset upon a change point detection. Here, the online estimator of Algorithm 2 is used as the change point detector instead of as the estimator with a separate estimator that is restarted only after a detection is flagged. This allows the online learner to track the time variations more optimally unlike Algorithm 2, which always restarts at a certain specified interval. The details of the algorithm are given in Appendix E. The experiments are performed on a general linear dynamical system to showcase the general applicability and efficacy of our algorithm. We note that the basic version of OLC-ZK (Algorithm 2) is also observed to receive sub-linear regret in the numerical simulations, although the variant discussed
Figure 1: Cumulative regret of OLC-ZK with different $M$ estimation.

Figure 2: Cumulative regret of OLC-ZK with different $\hat{G}$ estimation.

above performs better. Regret analysis of the change point detection variant of the OLC-ZK is an open problem.

Parameter setting: $\eta = 0.4, N = 7, h = 2, \theta_t = [A_t, B_t], B_{t,w} = I$ and $u_t$ are randomly generated at each time step; $C_t$ is randomly initialized but keeps unchanged across time steps: $C_{t_1} = C_{t_2}, \forall t_1, t_2 \in [1, t]$. We set $e_t = 0, \forall t \in [1, t]$. The cost function is a quadratic function of $y_t$ and $u_t$: $c_t(y_t, u_t) = y_t^T Q y_t + u_t^T R u_t$. $Q$ and $R$ are two randomly generated positive semi-definite matrices. Experiments are averaged over 10 random runs. In each run, OLC-ZK and baselines use the same $Q, R, C, A_t$ and $B_t$ for fair comparison.

Baselines: OLC-ZK with fixed $M$: the online algorithm where $M$ is a fixed value and is not updated. OLC-ZK with random $M$: the online algorithm where $M$ is picked randomly. OLC-ZK with fixed $\hat{G}$: the OLC-ZK algorithm with $\hat{G}_t$ fixed to a constant value instead of an estimator. OLC-ZK with random $\hat{G}$: the OLC-ZK algorithm with $\hat{G}_t$ picked randomly instead of an estimator. OLC-TI: online learning algorithm for time invariant systems (Simchowitz et al., 2020). In contrast to ours, which continuously explores and exploits, OLC-TI explores first and then exploits.

Experimental results: Figure 1 illustrates the results averaged over 10 independent runs, and the shadowed areas depict the standard deviation. Although the standard deviation is large in the plot, cumulative regret is consistently positive for each run. Figure 1 indicates that OLC-ZK has a smaller sub-linear increase in cumulative regret and smaller variance compared to the case when a fixed $M$ or a randomly generated $M$ is used. Similarly, it can be observed from Figure 2 that the proposed OLC-ZK algorithm achieves a smaller sub-linear regret with smaller variance compared to the case when a fixed $\hat{G}$ or a randomly generated $\hat{G}$ is applied instead of Eq. (10). Most importantly, we observe that, while initially the OLC-TI algorithm is better, over time its performance worsens and converges to the OLC-ZK with an arbitrarily fixed $\hat{G}$. This is expected as the estimate from the initial exploration phase of OLC-TI can be very different from the underlying system with changes over time and thus behave like an arbitrarily fixed $\hat{G}$ after a sufficiently long time. These results illustrate the importance of tracking $G_t$ and simultaneously updating $M_t$ and thus the effectiveness of our approach in adapting to the time variations.

5 Conclusion

In this work, we study the problem of online control of unknown time varying dynamical systems with arbitrary disturbances and cost functions. Our goal is to design an online adaptation algorithm that can provably achieve sub-linear regret up to sub-linear variations in the system with respect to stronger notions of variability like the number of changes. We present system, information and cost structures along with algorithms which guarantee such results and also present some open questions.

There are several open directions to explore: (i) extension of our result to general linear dynamical systems, (ii) extension of our result to the more general Disturbance Responce Control (DRC) (Minasyan et al., 2021), and (iii) the question whether we can achieve a regret better than $\tilde{O}\left(\Gamma_1^{1/5} T^{4/5}\right)$.

References

Abbasi-Yadkori, Y. and Szepesvári, C. (2011). Regret bounds for the adaptive control of linear quadratic systems. Proceedings of the 24th Annual Conference on Learning Theory, pages 1–26.

Agarwal, N., Bullins, B., Hazan, E., Kakade, S., and Singh, K. (2019a). Online control with adversarial disturbances. International Conference on Machine Learning, pages 111–119.

Agarwal, N., Hazan, E., and Singh, K. (2019b). Logarithmic
regret for online control. *Advances in Neural Information Processing Systems*, pages 10175–10184.

Åström, K. J. (2012). *Introduction to stochastic control theory*. Courier Corporation.

Cohen, A., Hasidim, A., Koren, T., Lazic, N., Mansour, Y., and Talwar, K. (2018). Online linear quadratic control. In *International Conference on Machine Learning (ICML)*, pages 1029–1038.

Cohen, A., Koren, T., and Mansour, Y. (2019). Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret. *International Conference on Machine Learning (ICML)*, pages 1300–1309.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2018). Regret bounds for robust adaptive control of the linear quadratic regulator. In *Neural Information Processing Systems (NeurIPS)*.

Gradu, P., Hallman, J., and Hazan, E. (2020). Non-stochastic control with bandit feedback. *Advances in Neural Information Processing Systems*, 33:10764–10774.

Hazan, E., Kakade, S., and Singh, K. (2020). The non-stochastic control problem. *Algorithmic Learning Theory*, pages 408–421.

Hazan, E., Rakhlin, A., and Bartlett, P. L. (2008). Adaptive online gradient descent. *Advances in Neural Information Processing Systems*, pages 65–72.

Ioannou, P. A. and Sun, J. (2012). *Robust adaptive control*. Dover Publications.

Kumar, P. R. and Varaiya, P. (2015). *Stochastic systems: Estimation, identification, and adaptive control*. SIAM.

Lale, S., Azizzadenesheli, K., Hassibi, B., and Anandkumar, A. (2021). Model learning predictive control in nonlinear dynamical systems. *IEEE Conference on Decision and Control (CDC)*, pages 757–762.

Li, Y., Chen, X., and Li, N. (2019). Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. *Neural Information Processing Systems (NeurIPS)*, pages 14887–14899.

Lin, Y., Hu, Y., Shi, G., Sun, H., Qu, G., and Wierman, A. (2021). Perturbation-based regret analysis of predictive control in linear time varying systems. *Neural Information Processing Systems (NeurIPS)*, 34.

Mania, H., Tu, S., and Recht, B. (2019). Certainty equivalence is efficient for linear quadratic control. In *Neural Information Processing Systems (NeurIPS)*.

Minasyan, E., Gradu, P., Simchowitz, M., and Hazan, E. (2021). Online control of unknown time-varying dynamical systems. *Advances in Neural Information Processing Systems*, 34:15934–15945.

Muthirayan, D., Yuan, J., Kalathil, D., and Khargonekar, P. P. (2021). Online learning for predictive control with provable regret guarantees. *arXiv preprint arXiv:2111.15041*.

Oymak, S. and Ozay, N. (2019). Non-asymptotic identification of lti systems from a single trajectory. *2019 American control conference (ACC)*, pages 5655–5661.

Sastry, S. and Bodson, M. (2011). *Adaptive control: stability, convergence and robustness*. Dover Publications.

Simchowitz, M. and Foster, D. (2020). Naive exploration is optimal for online lqr. *International Conference on Machine Learning (ICML)*, pages 8937–8948.

Simchowitz, M., Singh, K., and Hazan, E. (2020). Improper learning for non-stochastic control. *Conference on Learning Theory (COLT)*, pages 3320–3436.

Skogestad, S. and Postlethwaite, I. (2007). *Multivariable feedback control: analysis and design*, volume 2. Citeseer.

Yu, C., Shi, G., Chung, S., Yue, Y., and Wierman, A. (2020). The power of predictions in online control.

Zhou, K., Doyle, J., and Glover, K. (1996). *Robust and optimal control*. Prentice hall.
A Literature Review

One of the first settings that was extensively explored in online control is the LQR setting with unknown system and stochastic disturbances. Abbasi-Yadkori and Szepesvári (2011) were the first to study the online LQR problem with unknown system and stochastic disturbances. For this setting, Abbasi-Yadkori and Szepesvári (2011) propose an adaptive algorithm that achieves $\sqrt{T}$ regret with respect to the best linear control policy, which is the optimal policy. Dean et al. (2018) were the first to propose an efficient algorithm for the same problem. They showed that their algorithm achieves a regret of $O(T^{2/3})$. Cohen et al. (2019) and Mania et al. (2019) improved on this result by providing an efficient algorithm with a regret guarantee of $O(T^{1/2})$ for the same problem. Mania et al. (2019) extended these results to the partial observation setting and established $O(\sqrt{T})$-regret for the partially observed Linear Quadratic Gaussian (LQG) setting. Cohen et al. (2018) provide an $O(\sqrt{T})$ algorithm for a variant of the online LQR, where the system is unknown and noise is stochastic, but the controller cost function is an adversarially chosen quadratic function. Recently, Simchowitz and Foster (2020) showed that $O(T^{1/2})$ is the optimal regret for the online LQR control problem.

While the above works focused on online LQR, there are others who studied the control of much general systems: linear dynamic systems with adversarial disturbances and adversarial cost functions. Agarwal et al. (2019a) consider the control of a known linear dynamic system with additive adversarial disturbance and an adversarial convex controller cost function. They propose an online learning algorithm that learns a Disturbance Response Controller (DRC): a linear feedback of the form

$$u_t = DAC_t(y_t, u_t) = \sum_{k=1}^{n} M_{t-k} w_{t-k+1},$$

where

$$\|DAC_t\| = \|DAC_{t+1}\| = \|DAC_{t+2}\| = \cdots = \|DAC_{T}\| = 0.$$

When the cost functions are linear, it follows that

$$\|DAC_t\| = \sum_{k=1}^{T} \|DAC_{t+k}\| = \sum_{k=1}^{T} \|DAC_{t+k}\| = \cdots = \|DAC_{T}\| = 0.$$

A Literature Review

One of the first settings that was extensively explored in online control is the LQR setting with unknown system and stochastic disturbances. Abbasi-Yadkori and Szepesvári (2011) were the first to study the online LQR problem with unknown system and stochastic disturbances. For this setting, Abbasi-Yadkori and Szepesvári (2011) propose an adaptive algorithm that achieves $\sqrt{T}$ regret with respect to the best linear control policy, which is the optimal policy. Dean et al. (2018) were the first to propose an efficient algorithm for the same problem. They showed that their algorithm achieves a regret of $O(T^{2/3})$. Cohen et al. (2019) and Mania et al. (2019) improved on this result by providing an efficient algorithm with a regret guarantee of $O(T^{1/2})$ for the same problem. Mania et al. (2019) extended these results to the partial observation setting and established $O(\sqrt{T})$-regret for the partially observed Linear Quadratic Gaussian (LQG) setting. Cohen et al. (2018) provide an $O(\sqrt{T})$ algorithm for a variant of the online LQR, where the system is unknown and noise is stochastic, but the controller cost function is an adversarially chosen quadratic function. Recently, Simchowitz and Foster (2020) showed that $O(T^{1/2})$ is the optimal regret for the online LQR control problem.

While the above works focused on online LQR, there are others who studied the control of much general systems: linear dynamic systems with adversarial disturbances and adversarial cost functions. Agarwal et al. (2019a) consider the control of a known linear dynamic system with additive adversarial disturbance and an adversarial convex controller cost function. They propose an online learning algorithm that learns a Disturbance Response Controller (DRC): a linear feedback of the form

$$u_t = DAC_t(y_t, u_t) = \sum_{k=1}^{n} M_{t-k} w_{t-k+1},$$

where

$$\|DAC_t\| = \sum_{k=1}^{T} \|DAC_{t+k}\| = \sum_{k=1}^{T} \|DAC_{t+k}\| = \cdots = \|DAC_{T}\| = 0.$$

When the cost functions are linear, it follows that

$$\|DAC_t\| = \sum_{k=1}^{T} \|DAC_{t+k}\| = \sum_{k=1}^{T} \|DAC_{t+k}\| = \cdots = \|DAC_{T}\| = 0.$$

B Proof of Theorem 1

**Remark 4.** When the cost functions are linear, it follows that

(i) For a given $z^T = [y^T, u^T], (z')^T = [(y')^T, (u')^T],$

$$\|c_t(y, u) - c_t(y', u')\| \leq L \|z - z'\|,$$

where $L = G.$

(ii) $\nabla_y c(y, u) \leq G, \nabla_u c(y, u) \leq G.$

Since $u^{\pi_{DAC-O}}_t = \sum_{k=1}^{n} M_t w_{t-k+1}$, let's define a function $f_t$ by

$$f_t(M_{t-h}|G_t, s_{1:t}) := c_t(y^{\pi_{DAC-O}}_t[M_{t-h}|G_t, s_{1:t}], u^{\pi_{DAC-O}}_t).$$

We first prove the following two intermediate results.

**Lemma 1.** Consider two policy sequences $(M_{t-h} \ldots M_{t-k} \ldots M_t)$ and $(M_{t-h} \ldots M_{t-k} \ldots M_t)$, which differ only in the policy at time $t - k$, where $k \in \{0, 1, \ldots, h\}$. Then,

$$\left|f_t(M_{t-h} \ldots M_{t-k} \ldots M_t|G_t, s_{1:t}) - f_t(M_{t-h} \ldots \hat{M}_{t-k} \ldots M_t|G_t, s_{1:t})\right|$$

$$\leq L_f \left\|M_{t-k} - \hat{M}_{t-k}\right\|,$$

where

$$L_f = L \max \left\{\kappa_M \kappa_w h + \frac{\kappa_a \kappa_w}{\gamma} + \kappa_e + \frac{\kappa_a \kappa_h \kappa_M \kappa_w h}{\gamma}, 1\right\} \left(\kappa_a \kappa_h \kappa_w \sqrt{h} + \kappa_w \sqrt{h}\right).$$

**Proof.** Let $\hat{M}_{t-h} = [M_{t-h} \ldots M_{t-h} \ldots M_t].$ Then

$$\left\|y^{\pi_{DAC-O}}_t[M_{t-h}|G_t, s_{1:t}] - y^{\pi_{DAC-O}}_t[\hat{M}_{t-h}|G_t, s_{1:t}]\right\| = \left\| \sum_{i=1}^{h} \left(M_{t-i} - \hat{M}_{t-i}\right) w_{t-i+1}\right\|$$

$$\leq L_f \sum_{i=1}^{h} \left(M_{t-i} - \hat{M}_{t-i}\right) w_{t-i+1}.$$
Assumption 1. We make the following observations. Using a similar argument it follows that the control input $u_\tilde{\pi}^{DAC-0}$ then, for all $(t, s, a)$, $\sum_{k=1}^{h} |M_t[i]w_t-i+1| \leq \|M_t[i] - \tilde{M}_t[i]\| \leq (1 - \gamma)^{k-1} \kappa_a \kappa_b \sum_{i=1}^{h} |M_t[i] - \tilde{M}_t[i]| \leq (1 - \gamma)^{k-1} \kappa_a \kappa_b \kappa_w \sqrt{h} \|M_t - \tilde{M}_t\|.

Here, (a) follows from using the definition of 2-norm of a matrix and triangle inequality, (b, c) follow from Assumption 1 and (d) follows from the inequality that for any positive numbers $a_1, a_2, \ldots, a_t, \sum_{k=1}^{h} a_k \leq \sqrt{h} \sqrt{\sum_{k=1}^{h} a_k^2}$. Let

$$\tilde{u}_t^{\pi^{DAC-0}} = \begin{cases} \sum_{i=1}^{h} M_t[i]w_t-i+1 & \text{if } k > 0 \\ \sum_{i=1}^{h} \tilde{M}_t[i]w_t-i+1 & \text{if } k = 0. \end{cases}$$

Then,

$$\|u_t^{\pi^{DAC-0}} - \tilde{u}_t^{\pi^{DAC-0}}\| \leq \sum_{i=1}^{h} |M_t[i] - \tilde{M}_t[i]| \|w_t-i+1\| \leq \kappa_w \sqrt{h} \|M_t - \tilde{M}_t\|.$$ 

Here, (e) follows from applying triangle inequality first and by using the definition of 2-norm of a matrix, (f) follows from Assumption 1. We make the following observations. Using a similar argument it follows that the control input $u_t^{\pi^{DAC-0}}$ is bounded by

$$\|u_t^{\pi^{DAC-0}}\| \leq \kappa_w \kappa_M \sqrt{h}.$$

The truncated output $\tilde{y}_t^{\pi^{DAC-0}}[M_{t:t-h}][G_t, s_{1:t}]$ is bounded by

$$\|\tilde{y}_t^{\pi^{DAC-0}}[M_{t:t-h}][G_t, s_{1:t}]\| \leq \|s_t\| + \sum_{k=1}^{h} \|G_t[k]\| \|u_t^{\pi^{DAC-0}}\| \leq \kappa_a \kappa_b \kappa_M \kappa_w \sqrt{h} \sum_{k=1}^{h} (1 - \gamma)^{k-1} \leq \frac{\kappa_a \kappa_w}{\gamma} + \kappa_e + \frac{\kappa_a \kappa_b \kappa_M \kappa_w h}{\gamma}.$$ 

Here, (g) follows from triangle inequality, (h) follows from Assumption 1 and the bound on $\|u_t^{\pi^{DAC-0}}\|$ and (i) follows from bounding $s_t$ using the definition of $s_t$ and Assumption 1. Given these observations, by Assumption 2 it follows that

$$|f_t(M_{t-h} \ldots M_{t-k} \ldots M_{t})[G_t, s_{1:t}] - f_t(M_{t-h} \ldots \tilde{M}_{t-k} \ldots M_{t})[G_t, s_{1:t}]| \leq LR \left( \|\tilde{y}_t^{\pi^{DAC-0}}[M_{t:t-h}][G_t, s_{1:t}] - \tilde{y}_t^{\pi^{DAC-0}}[\tilde{M}_{t:t-h}][G_t, s_{1:t}]\| + LR \left( \|u_t^{\pi^{DAC-0}} - \tilde{u}_t^{\pi^{DAC-0}}\| \right) \right) \leq LR \left( (1 - \gamma)^{k-1} \kappa_a \kappa_b \kappa_w \sqrt{h} + \kappa_w \sqrt{h} \right) \|M_t - \tilde{M}_t\| \leq L \max \left\{ \kappa_M \kappa_w \sqrt{h} + \frac{\kappa_a \kappa_w}{\gamma} + \kappa_e + \frac{\kappa_a \kappa_b \kappa_M \kappa_w h}{\gamma}, 1 \right\} \left( (1 - \gamma)^{k-1} \kappa_a \kappa_b \kappa_w \sqrt{h} + \kappa_w \sqrt{h} \right) \|M_t - \tilde{M}_t\|.$$ 

Here, (j) follows from using the fact that the arguments $\tilde{y}_t^{\pi^{DAC-0}}, u_t^{\pi^{DAC-0}}$ and $\tilde{u}_t^{\pi^{DAC-0}}$ are bounded as shown earlier and Assumption 2 and (h) follows from substituting for $R$ using the bounds on $\tilde{y}_t^{\pi^{DAC-0}}$ and $u_t^{\pi^{DAC-0}}$ derived earlier. The final result follows from here.

Lemma 2. For all $M$ such that $\|M[j]\| \leq \kappa_M$, $\forall j \in [1, h]$, we have that

$$\|\nabla_M f_t(M, \ldots, M[G_t, s_{1:t}]\| \leq G_f, G_f = GDhnm \left( \frac{\kappa_a \kappa_b \kappa_w}{\gamma} + \kappa_w \right),$$

$$D = \max \left\{ \kappa_M \kappa_w \sqrt{h} + \frac{\kappa_a \kappa_w}{\gamma} + \kappa_e + \frac{\kappa_a \kappa_b \kappa_M \kappa_w h}{\gamma}, 1 \right\}.$$
We now prove the main result. When each $M_k \in M_{t:t-h}$ is fixed to $M$, we denote $f_t(M_{t:t-h}|G_t, s_{1:t})$ by $F_t(M|G_t, s_{1:t})$. Thus,

$$\nabla_M f_t(M, \ldots, M|G_t, s_{1:t}) = \nabla_M F_t(M|G_t, s_{1:t}).$$

Let

$$D = \max \left\{ \kappa_M \kappa_w h + \frac{\kappa_a \kappa_w}{\gamma} + \kappa + \frac{\kappa_a \kappa_b \kappa_M \kappa_w h}{\gamma}, 1 \right\}$$

Let $\pi_{DAC}$ denote the fixed DAC policy with the policy parameter $M \in \mathcal{M}$. Then, the output and the control input under $\pi_{DAC}$ are given by $\hat{y}_t^{\pi_{DAC}}$ and $u_t^{\pi_{DAC}}$. Then, similar to the derivation in Lemma 1, we have

$$\|y_t^{\pi_{DAC}}\| \leq D, \|u_t^{\pi_{DAC}}\| \leq D.$$ Let $M_{p,q}^{[k]}$ denote the $(p,q)$th element of the matrix $M^{[k]}$. Then, it is sufficient to derive the bound for $\nabla_{M_{p,q}^{[k]}} F_t(M|G_t, s_{1:t})$, to bound the overall gradient. Then, from Assumption 2(iii) it follows that

$$\left\| \nabla_{M_{p,q}^{[k]}} F_t(M|G_t, s_{1:t}) \right\| \leq GD \left( \left\| \frac{\partial y_t^{\pi_{DAC}}}{\partial M_{p,q}^{[k]}} \right\| + \left\| \frac{\partial u_t^{\pi_{DAC}}}{\partial M_{p,q}^{[k]}} \right\| \right).$$

We have that

$$\frac{\partial y_t^{\pi_{DAC}}}{\partial M_{p,q}^{[k]}} = \sum_{i=1}^{h} G_i^{[i]} \frac{\partial u_t^{\pi_{DAC}}}{\partial M_{p,q}^{[k]}} = \sum_{i=1}^{h} G_i^{[i]} \left( \sum_{j=1}^{h} \frac{\partial M}{\partial M_{p,q}^{[j]}} \right) w_{t-i+j}$$

Then taking the norm on both sides,

$$\left\| \frac{\partial y_t^{\pi_{DAC}}}{\partial M_{p,q}^{[k]}} \right\| \leq \sum_{i=1}^{h} \sum_{j=1}^{h} \left\| G_i^{[i]} \right\|_2 \left\| \frac{\partial M}{\partial M_{p,q}^{[j]}} \right\| \|w_{t-i+j}+1\| \leq \sum_{i=1}^{h} \left\| G_i^{[i]} \right\|_2 \|w_{t-i+k+1}\|$$

Here, (a) follows from applying triangle inequality and by using the definition of 2-norm of a matrix, (b) follows from the fact that $\left\| \frac{\partial M_{p,q}^{[k]}}{\partial M_{p,q}^{[k]}} \right\| = 1$ for $j = k$ and zero otherwise, and (c) follows from Assumption 1. Similarly, we have that

$$\frac{\partial u_t^{\pi_{DAC}}}{\partial M_{p,q}^{[k]}} \left( \sum_{i=1}^{h} \frac{\partial M_{p,q}^{[i]}}{\partial M_{p,q}^{[k]}} \right) w_{t-i+1}, \text{ i.e.,} \left\| \frac{\partial u_t^{\pi_{DAC}}}{\partial M_{p,q}^{[k]}} \right\| \leq \|w_{t-k+1}\| \leq \kappa_w.$$ Here, (f) follows from the definition of $u_t^{\pi_{DAC}}$ and (g) follows from the fact that $\left\| \frac{\partial M_{p,q}^{[i]}}{\partial M_{p,q}^{[k]}} \right\| = 1$ for $i = k$ and zero otherwise. Thus, we get

$$\left\| \nabla_{M_{p,q}^{[k]}} F_t(M|G_t, s_{1:t}) \right\| \leq GD \left( \frac{\kappa_a \kappa_b \kappa_M \kappa_w}{\gamma} + \kappa_w \right).$$

The final result follows from here.

We now prove the main result. When each $M_k \in M_{t:t-h}$ is fixed to $M$ then we denote $f_t(M_{t:t-h}|G_t, s_{1:t})$ by $F_t(M|G_t, s_{1:t})$. Given this definition, the regret can be split as

$$R_T = \sum_{t=1}^{T} c_t(y_t^{\pi_{DAC} - O}, u_t^{\pi_{DAC} - O}) - \min_{\pi \in \Pi_M} \sum_{t=1}^{T} c_t(y_t^{\pi}, u_t^{\pi})$$

$$= \sum_{t=1}^{T} c_t(y_t^{\pi_{DAC} - O}, u_t^{\pi_{DAC} - O}) - \sum_{t=1}^{T} c_t(\hat{y}_t^{\pi_{DAC} - O}[M_{t:t-h}|G_t, s_{1:t}], u_t^{\pi_{DAC} - O})$$

Cost Truncation Error
Putting together all the three terms, we get the final result

\[ \sum_{t=1}^{T} f_t(M_{t:t-h}|G_t, s_{1:t}) - \min_{M \in \mathcal{M}} \sum_{t=1}^{T} F_t(M|G_t, s_{1:t}) \]

Policy Regret

\[ + \min_{M \in \mathcal{M}} \sum_{t=1}^{T} F_t(M|G_t, s_{1:t}) - \min_{\pi \in \Pi_M} \sum_{t=1}^{T} c_t(y_t^\pi, u_t^\pi). \]

Cost Approximation Error

Here, the cost truncation error is the error due to replacing the full output \( y_t^{\text{DAC-O}} \) with the truncated output \( \hat{y}_t^{\text{DAC-O}}[M_{t:t-h}|G_t, s_{1:t}] \), the cost approximation error is the error of the optimal given the truncation of the cost, and the policy approximation error is the error between the optimal DAC and the optimal linear feedback policy.

Let \( D = \sup_{M_1, M_2 \in \mathcal{M}} \| M_1 - M_2 \| \). By applying (Agarwal et al., 2019a, Theorem 4.6) to the online learning algorithm Algorithm 1 and Lemma 1 and Lemma 2, we get that

\[ \sum_{t=1}^{T} f_t(M_{t:t-h}|G_t, s_{1:t}) \leq \frac{D^2}{\eta} + TG_f^2 \eta + L_f h^2 \eta G_f T. \]

Next, we bound the cost truncation error term.

\[ \sum_{t=1}^{T} c_t(y_t^{\text{DAC-O}}, u_t^{\text{DAC-O}}) - \sum_{t=1}^{T} c_t(\hat{y}_t^{\text{DAC-O}}[M_{t:t-h}|G_t, s_{1:t}], u_t^{\text{DAC-O}}) \]

\[ \leq \sum_{t=1}^{T} LR \| y_t^{\text{DAC-O}} - \hat{y}_t^{\text{DAC-O}}[M_{t:t-h}|G_t, s_{1:t}] \| \leq \sum_{t=1}^{T} LR \sum_{k=h+1}^{T} \| G_t^{[k]} u_{t-k} \| \]

\[ \leq \sum_{t=1}^{T} LR \sum_{k=h+1}^{T} (1 - \gamma)^{k-1} \leq \frac{LRK_aK_bK_Mk_wh(1 - \gamma)^hT}{\gamma}. \]

Here, (a) follows from Assumption 2, (b) follows from definitions of \( y_t^{\text{DAC-O}} \) and \( \hat{y}_t^{\text{DAC-O}}[M_{t:t-h}|G_t, s_{1:t}] \), (c) follows from applying triangle inequality and by using the definition of 2-norm of a matrix and (d) follows from Assumption 1 and the bound on \( u_t^{\text{DAC-O}} \).

Next, we bound the cost approximation error term. This term can also be bounded in a similar way. Let us denote the \( M \) that minimizes \( \sum_{t=1}^{T} c_t(y_t^\pi, u_t^\pi) \) by \( M_*. \) Then

\[ \min_{M \in \mathcal{M}} \sum_{t=1}^{T} F_t(M|G_t, s_{1:t}) - \min_{\pi \in \Pi_M} \sum_{t=1}^{T} c_t(y_t^\pi, u_t^\pi) \leq \sum_{t=1}^{T} F_t(M_*|G_t, s_{1:t}) - \min_{\pi \in \Pi_M} \sum_{t=1}^{T} c_t(y_t^\pi, u_t^\pi) \]

\[ \leq LR \sum_{t=1}^{T} (\| y_t^{\text{DAC-O}}[M_*|G_t, s_{1:t}] - y_t^\pi \|) \leq LR \sum_{t=1}^{T} \sum_{k=h+1}^{T} \| G_t^{[k]} \| \| u_{t-k}^\pi \| \leq \frac{LRK_aK_bK_Mk_wh(1 - \gamma)^hT}{\gamma}. \]

Here, (e) follows from Assumption 2, (f) follows from the fact that \( y_t^{\text{DAC-O}}[M_*|G_t, s_{1:t}] \) is just the truncated output of \( y_t^\pi \) and applying triangle followed by using the definition of 2-norm of a matrix, and (g) follows from applying Assumption 1.

Putting together all the three terms, we get the final result.

\[ \| u_t^{\text{DAC-O}} \| \leq R_u = \kappa_M k_wh + 3\sigma \sqrt{m + \log(1/\delta)}, \quad \| s_t \| \leq R_s = \frac{K_aK_wh}{\gamma} + \kappa_n + \frac{2R_uK_aK_b}{\gamma}, \]

\[ \| y_t^{\text{DAC-O}} \| \leq \frac{K_aK_wh}{\gamma} + \kappa_e + \frac{K_aK_bR_u}{\gamma} \leq R_s, \quad \forall \ t. \]

\[ \]
Proof. The first term in the bound of \( \|u_t^{\pi_{DAC-O}}\| \) follows from bounding the DAC part of the control policy (6), following the steps in the proof of Lemma 1. The second term follows from bounding the perturbation Eq. (7) by using (Simchowitz et al., 2020, Claim D.3).

From the definition of \( \hat{s}_t \), we get under the event that \( \|u_t^{\pi_{DAC-O}}\| \leq R_u \),

\[
\hat{s}_t = s_t + \sum_{k=1}^h \left( G_t^k - \hat{G}_t^k \right) u_t^{\pi_{DAC-O}}, \quad \text{i.e.,} \quad \| \hat{s}_t \| \leq \| s_t \| + \sum_{k=1}^h \left\| G_t^k - \hat{G}_t^k \right\| \| u_t^{\pi_{DAC-O}} \|
\]

\[
\| \hat{s}_t \| \leq \frac{\kappa_a \kappa_w}{\gamma} + \kappa_c + 2R_u \sum_{k=1}^h (1 - \gamma)^{k-1} \kappa_a \kappa_b = \frac{\kappa_a \kappa_w}{\gamma} + \kappa_c + \frac{2R_u \kappa_a \kappa_b}{\gamma}.
\]

Here, (a) follows from applying triangle inequality, followed by the definition of 2-norm of a matrix and (b) follows from Assumption 1. By definition, we get, under the event that \( \|u_t^{\pi_{DAC-O}}\| \leq R_u \),

\[
y_t^{\pi_{DAC-O}} = s_t + \sum_{k=1}^{t-1} G_t^k u_{t-k}^{\pi_{DAC-O}}, \quad \text{i.e.,} \quad \| y_t^{\pi_{DAC-O}} \| \leq \| s_t \| + \sum_{k=1}^{t-1} \left\| G_t^k \right\| \| u_{t-k}^{\pi_{DAC-O}} \|
\]

\[
\leq \frac{\kappa_a \kappa_w}{\gamma} + \kappa_c + \sum_{k=1}^{t-1} (1 - \gamma)^{k-1} \kappa_a \kappa_b R_u = \frac{\kappa_a \kappa_w}{\gamma} + \kappa_c + \frac{\kappa_a \kappa_b R_u}{\gamma}.
\]

\[\square\]

Let’s define a function \( f_t \) by

\[
f_t(M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}) := c_t(\hat{y}_t^{\pi_{DAC-O}}[M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}], \hat{u}_t^{\pi_{DAC-O}}[M_t|w_{1:t}]).
\]

**Lemma 4.** Consider two policy sequences \((M_{t-h}, \ldots, M_{t-k}, \ldots, M_t)\) and \((\hat{M}_{t-h}, \ldots, \hat{M}_{t-k}, \ldots, \hat{M}_t)\), which differ only in the policy at time \( t-k \), where \( k \in \{0, 1, \ldots, h\} \). Then, with probability \( 1 - \delta / 3 \)

\[
\left| f_t(M_{t-h} \ldots M_{t-k} \ldots M_t|\hat{G}_t, \hat{s}_{1:t}) - f_t(M_{t-h} \ldots \hat{M}_{t-k} \ldots \hat{M}_t|\hat{G}_t, \hat{s}_{1:t}) \right|
\]

\[
\leq L_f \left\| M_{t-k} - \hat{M}_{t-k} \right\|, \quad L_f = L \max \left\{ \left( 1 + \frac{\kappa_a \kappa_b}{\gamma} \right) R_u + R_s, 1 \right\} \left( (1 - \gamma)^{k-1} \kappa_a \kappa_b \kappa_w \sqrt{h} + \kappa_w \sqrt{h} \right).
\]

**Proof.** The proof follows from the same steps as in the proof of Lemma 1. \[\square\]

**Lemma 5.** For all \( M \) such that \( \|M^{[j]}\| \leq \kappa_M, \forall j \in [1, h] \), we have that

\[
\left\| \nabla_M f_t(M, \ldots, M[\hat{G}_t, \hat{s}_{1:t}] \right\| \leq G_f, \quad G_f = GD \left( \frac{\kappa_a \kappa_b \kappa_w}{\gamma} + \kappa_w \right), \quad D = \max \left\{ \left( 1 + \frac{\kappa_a \kappa_b}{\gamma} \right) R_u + R_s, 1 \right\}.
\]

**Proof.** The proof follows from the same steps as in the proof of Lemma 2. \[\square\]

We now prove the main result. We can split the regret as

\[
R_T = \sum_{t=1}^T c_t(y_t^{\pi_{DAC-O}}, u_t^{\pi_{DAC-O}}) - \min_{\pi \in \Pi_N} \sum_{t=1}^T c_t(y_t^{\pi}, u_t^{\pi})
\]

\[
= \sum_{t=1}^T c_t(y_t^{\pi_{DAC-O}}, u_t^{\pi_{DAC-O}}) - \sum_{t=1}^T c_t(y_t^{\pi_{DAC-O}}[M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}], u_t^{\pi_{DAC-O}})
\]

\[\text{Cost Truncation Error}\]
\[
\begin{align*}
&+ \sum_{t=1}^{T} c_t(y^\pi_{DAC-O} (M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}, u^\pi_{DAC-O}) - \min_{M\in\mathcal{M}} \sum_{t=1}^{T} F_t(M|\hat{G}_t, \hat{s}_{1:t}) \\
&+ \min_{M\in\mathcal{M}} \sum_{t=1}^{T} F_t(M|\hat{G}_t, \hat{s}_{1:t}) - \min_{M\in\mathcal{M}} \sum_{t=1}^{T} F_t(M|G_t, s_{1:t}) \\
&+ \min_{M\in\mathcal{M}} \sum_{t=1}^{T} F_t(M|G_t, s_{1:t}) - \min_{\pi\in\Pi_F} \sum_{t=1}^{T} c_t(y^\pi_t, u^\pi_t).
\end{align*}
\]

Thus, under an event with a probability \(1 - \delta/3\)

\[
\begin{align*}
\sum_{t=1}^{T} c_t(y^\pi_{DAC-O} (M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}, u^\pi_{DAC-O}) & \leq \sum_{t=1}^{T} \left( f_t(M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}) + 3LR \left( 1 + \frac{K_a K_b}{\gamma} \right) \sqrt{m + \log(1/\delta)} \right).
\end{align*}
\]

Next, we bound the policy regret term. Let \(R = \max \left\{ (1 + \frac{K_a K_b}{\gamma}) R_a + R_s, 1 \right\} \). Then, by using the definition of \(u^\pi_{DAC-O}\), Assumption 2 and Simchowitz et al. 2020 Claim D.3, with a probability \(1 - \delta/3\)

\[
\begin{align*}
\sum_{t=1}^{T} c_t(y^\pi_{DAC-O} (M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}, u^\pi_{DAC-O}) & \leq \sum_{t=1}^{T} \left( f_t(M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}) + 3LR \left( 1 + \frac{K_a K_b}{\gamma} \right) \sqrt{m + \log(1/\delta)} \right).
\end{align*}
\]

Thus, with a probability \(1 - \delta/3\), the policy regret term is given by

\[
\begin{align*}
\sum_{t=1}^{T} c_t(y^\pi_{DAC-O} (M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}, u^\pi_{DAC-O}) & \leq \sum_{t=1}^{T} \left( f_t(M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}) + 3LR \left( 1 + \frac{K_a K_b}{\gamma} \right) \sqrt{m + \log(1/\delta)} \right).
\end{align*}
\]

Let \(D = \sup_{t,r,x,y} \| M_t - M_r \| \). By applying Agarwal et al. 2019b Theorem 4.6 to the online learning algorithm Algorithm 2 and Lemma 4 and Lemma 5 we get that

\[
\begin{align*}
\sum_{t=1}^{T} f_t(M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}) & \leq \frac{D^2}{\eta} - \frac{T^2 G_f^2 \eta}{L_f h^2 \eta G_f T}.
\end{align*}
\]

Thus, under an event with a probability \(1 - \delta/3\)

\[
\begin{align*}
\sum_{t=1}^{T} c_t(y^\pi_{DAC-O} (M_{t:t-h}|\hat{G}_t, \hat{s}_{1:t}, u^\pi_{DAC-O}) - \min_{M\in\mathcal{M}} \sum_{t=1}^{T} F_t(M|\hat{G}_t, \hat{s}_{1:t}) & \leq \frac{D^2}{\eta} + TG_f^2 \eta + L_f h^2 \eta G_f T + 3LR \left( 1 + \frac{K_a K_b}{\gamma} \right) \sqrt{m + \log(1/\delta)} T \sigma.
\end{align*}
\]

Next, we bound the model approximation term. Let \(M_*\) denote the minimizer of \(\sum_{t=1}^{T} F_t(M|\hat{G}_t, \hat{s}_{1:t})\) and let \(M_*\) denote the minimizer of \(\sum_{t=1}^{T} F_t(M|G_t, s_{1:t})\). Then,

\[
\begin{align*}
\min_{M\in\mathcal{M}} \sum_{t=1}^{T} F_t(M|\hat{G}_t, \hat{s}_{1:t}) & - \min_{M\in\mathcal{M}} \sum_{t=1}^{T} F_t(M|G_t, s_{1:t}) \overset{(a)}{=} \sum_{t=1}^{T} F_t(M_*|\hat{G}_t, \hat{s}_{1:t}) - \sum_{t=1}^{T} F_t(M_*|G_t, s_{1:t})
\end{align*}
\]
Here, (a) we use the fact that $M_\ast$ is a sub-optimal policy for $\sum_{t=1}^{T} F_t(M|\hat{G}_t, \hat{s}_{1:t})$, (b) follows from the respective definitions and (c) follows from Assumption 3. Now, under the same event, with probability $1 - \delta/3$,

$$
\|\bar{\pi}^{\text{DAC-O}}_{\ast} - \bar{\pi}^{\text{DAC-O}}_{\ast}\| \leq \|\hat{s}_t - s_t\| + R_u \sqrt{\delta} \|G_t - \hat{G}_t\|_2
$$

Here, (d) follows from triangle inequality and (e) follows from substituting for $\hat{s}_t$. Putting the terms together, we get that, under the same event, with probability $1 - \delta/3$

$$
\min_{M \in \mathcal{M}} \sum_{t=1}^{T} F_t(M|\hat{G}_t, \hat{s}_{1:t}) - \min_{M \in \mathcal{M}} \sum_{t=1}^{T} F_t(M|G_t, s_{1:t}) \leq 2LRR_u \sqrt{\delta} \|G_t - \hat{G}_t\|_2 + \frac{\kappa_a \kappa_b LRR_u (1 - \gamma)^h}{\gamma}.
$$

Next, we bound the final policy approximation error. Let $M_\ast$ be the optimizing disturbance gain for $\sum_{t=1}^{T} c_t(y_t^\pi, u_t^\pi)$ and let $\pi_\ast$ denote the policy corresponding to $M_\ast$. Then,

$$
\min_{M \in \mathcal{M}} \sum_{t=1}^{T} F_t(M|G_t, s_{1:t}) - \min_{\pi \in \Pi_{\mathcal{M}}} \sum_{t=1}^{T} c_t(y_t^\pi, u_t^\pi) \leq \sum_{t=1}^{T} c_t(y_t^{\pi_\ast}, u_t^{\pi_\ast})
$$

$$
\leq T \sum_{t=1}^{T} c_t(\bar{\pi}^{\text{DAC-O}}_{\ast} | G_t, s_{1:t}, \bar{\pi}^{\text{DAC-O}}_{\ast} | w_{1:t}) - \sum_{t=1}^{T} c_t(y_t^{\pi_\ast}, u_t^{\pi_\ast})
$$

$$
\leq LR \sum_{t=1}^{T} (\|\bar{\pi}^{\text{DAC-O}}_{\ast} | G_t, s_{1:t} - y_t^{\pi_\ast}\| + \|\bar{\pi}^{\text{DAC-O}}_{\ast} | w_{1:t} - u_t^{\pi_\ast}\|)
$$

$$
\leq LR \sum_{t=1}^{T} (\|\bar{\pi}^{\text{DAC-O}}_{\ast} | G_t, s_{1:t} - y_t^{\pi_\ast}\|) + LR \sum_{t=1}^{T} \left(\left\|\sum_{k=h+1}^{T} \tilde{G}_t^{[k]} u_t^{\pi_\ast}_{t-k}\right\|ight)
$$

Here, (f) follows from Assumption 2, (g) follows from the fact that $\bar{\pi}^{\text{DAC-O}}_{\ast} | w_{1:t} = y_t^{\pi_\ast}$ and (h) follows from the fact that $\bar{\pi}^{\text{DAC-O}}_{\ast} | G_t, s_{1:t}$ is just the truncation of $y_t^{\pi_\ast}$ and (i) follows from Assumption 1. Putting together the bound on all the terms we get that with a probability $1 - \delta/3$

$$
R_T \leq \frac{D^2}{\eta} + TG_T^2 \eta + L_f h^2 \eta G_f T + 3LR \left(1 + \frac{\kappa_a \kappa_b}{\gamma}\right) \sqrt{m + \log(1/\delta)} T \sigma + 2LRR_u \sqrt{\delta} \sum_{t=1}^{T} \|G_t - \hat{G}_t\|_2.
$$
Next, we bound the term \( \sum_{t=1}^{T} \left\| G_t - \hat{G}_t \right\|_2^2 \). We denote the system parameters in a period where it does not change by \( G \), without the time subscript. First we introduce two lemmas which we then use to derive the final bound.

**Lemma 6.** For any period \( k \) in the CPD algorithm where the system parameters are a constant, i.e., \( G_t = G \), suppose \( t_p \geq \text{chm} \log(2hN)^2 \log(2tpm)^2 \) for some constant \( c \geq 0 \), \( T \geq 3 \), then with probability \( 1 - N^{-\log(N)} - \delta \)

\[
\left\| \hat{G}_k^{\text{cd}} - G \right\|_2 \leq \frac{\beta(\delta, \lambda, \sigma, h, N)}{\sigma \sqrt{N}}, \quad \beta(\delta, \lambda, \sigma, h, N) = 2\sqrt{h} \zeta \Delta \left( \sqrt{n \log(2) + 2 \log \left( \frac{2h}{\delta} \right)} + \frac{\lambda \kappa_a \kappa_b}{\gamma \zeta \Delta \sigma \sqrt{h} N} \right) 
\]

\[
\zeta = \left( R_s + \frac{\kappa_a \kappa_b \kappa_m \kappa_h h}{\gamma} + \frac{\kappa_a \kappa_b R_s}{\gamma} \right).
\]

**Proof.** The proof follows the proof style of [Simchowitz et al. 2020] Lemma D.4 but requires many additional steps. We recall

\[
\hat{G}_k^{\text{cd}} = \arg \min_G \sum_{p=t^*_k + h}^{t^*_k - h} \ell_p \left( \hat{G} \right) + \lambda \left\| \hat{G} \right\|, \quad \ell_p \left( \hat{G} \right) = \left\| y_p^{\text{DAC-O}} - \sum_{l=1}^{h} (\hat{G}[l] - G[l]) \delta u_{p-l} \right\|^2_2.
\]

Let \( \delta_p = y_p^{\text{DAC-O}} - \sum_{l=1}^{h} G[l] \delta u_{p-l} \). Then it follows that

\[
\ell_p \left( \hat{G} \right) = \left\| \delta_p - \sum_{l=1}^{h} (G[l] - G[l]) \delta u_{p-l} \right\|^2_2.
\]

Let \( \delta u_p^T = [\delta u_{p-1}^\text{DAC-O}, \delta u_{p-2}^\text{DAC-O}, \ldots, \delta u_{p-h}]^T \). Then,

\[
\ell_p \left( \hat{G} \right) = \left\| \delta_p - (\hat{G} - G) \delta u_p \right\|^2_2.
\]

Let \( \Delta^T = [\delta_{t^*_k + h}, \ldots, \delta_{h-h}] \), \( \mathbf{U}^T = [\delta u_{t^*_k + h}^\text{DAC-O}, \ldots, \delta u_{h-h}^\text{DAC-O}] \). Then, the solution to the least squares satisfies

\[
(\hat{G}_k^{\text{cd}})^T - G^T = (\mathbf{U}^T \mathbf{U} + \lambda I)^{-1} (\mathbf{U}^T \Delta - \lambda G^T).
\]

Then, by Cauchy-Schwarz

\[
\left\| \hat{G}_k^{\text{cd}} - G \right\|_2 \leq \left\| (\mathbf{U}^T \mathbf{U} + \lambda I)^{-1} \right\|_2 \left( \left\| \mathbf{U}^T \Delta \right\|_2 + \lambda \left\| G \right\|_2 \right).
\]

Now, by expanding \( \mathbf{U}^T \Delta \) it follows that the rows of \( \mathbf{U}^T \Delta \) are given by

\[
\sum_{p=t^*_k + h}^{t^*_k - h} \delta u_{p-i}^\text{DAC-O} \delta_p^\text{T}, \quad \forall i \in [1, h].
\]

Let \( S^m := \{ v \in \mathbb{R}^m : \| v \| = 1 \} \). Then, following steps similar to [Simchowitz et al. 2020] Lemma D.4, which follows from standard matrix norm inequalities, we get

\[
\left\| \mathbf{U}^T \Delta \right\|_2 \leq \sqrt{h} \max_{i \in [1, h]} \left\| \sum_{p=t^*_k + h}^{t^*_k - h} \delta u_{p-i}^\text{DAC-O} \delta_p^\text{T} \right\|_2 \leq \sqrt{h} \max_{i \in [1, h]} \max_{v \in S^m} \left\| v^\text{T} \sum_{p=t^*_k + h}^{t^*_k - h} \delta u_{p-i}^\text{DAC-O} \delta_p^\text{T} \right\|_2.
\]
By definition
\[ \delta_p = s_p + \sum_{i=1}^{p-1} \sum_{j=1}^{h} G_p[i] M_{p-i} u_{p-i-j} + \sum_{i=h+1}^{p-1} G_p[i] \delta u_{p-i} \pi \text{DAC-O} \].

Consider the filtration \( \mathcal{F}_t \) generated by the sequence of random inputs \( \delta u_{p-i} \pi \text{DAC-O} \). The terms in \( \delta_p \) are dependent on the sequence of random inputs are the third and the second term. The third term is clearly \( \mathcal{F}_{p-i-h-1} \) measurable. The variables in second term that are dependent on the sequence of random inputs are \( M_{p-i} \). \( M_{p-i} \) is dependent only through \( \hat{G}_{p-i-1} \) given the linearity of the cost functions and the update equation for \( M_t \). By Eq. (10), it follows that \( \hat{G}_{p-i-1} \) is only a function of the random inputs up to \( p-i-h-1 \) time steps. Further, by the fact that the least-squares solution is a regularized least-squares, and the fact that \( \text{Proj}_i \) is also continuous (which follows from the fact that \( G \) is a convex set in a Hilbert space), \( \hat{G}_{p-i-1} \) is a continuous function. By the update equation for \( M_t \), it follows that \( M_{p-i} \) is also a continuous function, and therefore \( \mathcal{F}_{p-i-h-1} \) measurable. Therefore, \( \delta_p \) is overall \( \mathcal{F}_{p-i-h-1} \) measurable.

Given that \( \delta u_{p-i} \pi \text{DAC-O} \) is \( \mathcal{F}_{p-i} \) measurable and \( \delta_p \) is \( \mathcal{F}_{p-i-h-1} \) for each \( i \in [1, h] \), the self-normalized martingale inequality (Abbasi-Yadkori and Szepesvári, 2011 Theorem 16) can be applied to the sum \( \sum_{p=t_k+h}^{t_k-h} \delta u_{p-i} \pi \text{DAC-O} \delta_p \) by replacing \( k \) in (Abbasi-Yadkori and Szepesvári, 2011 Theorem 16) by \( p \) and setting \( \eta_p = \delta u_{p-i} \pi \text{DAC-O}, m_{p-1} = \delta_p \).

Then, by recognizing that the length of the sum \( \sum_{p=t_k+h}^{t_k-h} \delta u_{p-i} \pi \text{DAC-O} \delta_p \) is \( N \) and therefore setting \( V = \mathbf{c}^2 \mathbf{I}, V_N = \mathbf{\Delta}^\top \mathbf{\Delta}, \mathbf{\nabla}_N = \mathbf{\Delta}^\top \mathbf{\Delta} + V \) in (Abbasi-Yadkori and Szepesvári, 2011 Theorem 16), we have that with probability \( 1 - \delta/(2h) \)
\[ \left\| V^\top \sum_{p=t_k+h}^{t_k-h} \delta u_{p-i} \pi \text{DAC-O} \delta_p (\mathbf{\Delta}^\top \mathbf{\Delta} + \mathbf{c}^2 \mathbf{I})^{-1/2} \right\|^2_2 \leq \sigma^2 \left( \log \left( \frac{\det(\mathbf{\Delta}^\top \mathbf{\Delta} + \mathbf{c}^2 \mathbf{I})}{\zeta} \right) + 2 \log \left( \frac{2h}{\delta} \right) \right) \]

Let \( \zeta \) be a parameter corresponding to the event \( \mathcal{E}_\zeta := \| \mathbf{\Delta} \|_2 \leq \zeta \). Then, under this event we have that
\[ \det(\mathbf{\Delta}^\top \mathbf{\Delta} + \mathbf{c}^2 \mathbf{I}) \leq 2^n \zeta^2n, \text{ i.e., } \frac{\det(\mathbf{\Delta}^\top \mathbf{\Delta} + \mathbf{c}^2 \mathbf{I})}{\zeta^2n} \leq 2^n. \]

By the fact that \( (\mathbf{\Delta}^\top \mathbf{\Delta} + \mathbf{c}^2 \mathbf{I})^{-1} \leq \frac{1}{\zeta^2} \), we have that
\[ \frac{1}{\zeta^2} \left\| V^\top \sum_{p=t_k+h}^{t_k-h} \delta u_{p-i} \pi \text{DAC-O} \delta_p \right\|^2_2 \leq \left\| V^\top \sum_{p=t_k+h}^{t_k-h} \delta u_{p-i} \pi \text{DAC-O} \delta_p (\mathbf{\Delta}^\top \mathbf{\Delta} + \mathbf{c}^2 \mathbf{I})^{-1/2} \right\|^2_2. \]

Then, combining the above three equations, under the event \( \mathcal{E}_\zeta \) and with probability \( 1 - \delta/(2h) \)
\[ \frac{1}{\zeta^2} \left\| V^\top \sum_{p=t_k+h}^{t_k-h} \delta u_{p-i} \pi \text{DAC-O} \delta_p \right\|^2_2 \leq \sigma^2 \left( n \log(2) + 2 \log \left( \frac{2h}{\delta} \right) \right). \]

Next, by (Oymak and Ozyay 2019 Lemma C.2), given that the conditions of (Oymak and Ozyay 2019 Lemma C.2) are satisfied, with probability at least \( 1 - N^{-\log(N)} \)
\[ \mathbf{U}^\top \mathbf{U} \geq N \sigma^2/2, \text{ i.e., } \mathbf{U}^\top \mathbf{U} + \mathbf{\lambda} \mathbf{I} \geq N \sigma^2/2. \]

Therefore, under the event \( \mathcal{E}_\zeta \), by union bound, with probability \( 1 - \delta/2 - N^{-\log(N)} \)
\[ \left\| \hat{G}_k - G \right\|^2_2 \leq \left\| (\mathbf{U}^\top \mathbf{U} + \mathbf{\lambda} \mathbf{I})^{-1} \right\|_2 \left( \left\| \mathbf{U}^\top \mathbf{\Delta} \right\|_2 + \mathbf{\lambda} \right) \leq \frac{2\sqrt{h} \zeta}{N \sigma} \left( \sqrt{n \log(2) + 2 \log \left( \frac{2h}{\delta} \right)} + \frac{\mathbf{\lambda} \kappa_a \kappa_b}{\sqrt{h} \sigma \gamma} \right). \]
By [Simchowitz et al. 2020] Claim D.3, for $T \geq 3$, $\delta \leq 1/T$, with probability greater than $1 - \delta/2$ 
\[
\|\delta u^\text{DAC-O}_p\| \leq 3\sigma \sqrt{m + \log(1/\delta)}, \quad \forall \ p \in [t^k_s, t^k_e].
\]

Lets call the event where $\|\delta u^\text{DAC-O}_p\| \leq 3\sigma \sqrt{m + \log(1/\delta)}$, $\forall \ p \in [t^k_s, t^k_e]$ as $\mathcal{E}_{u,b}$. Therefore, under the event $\mathcal{E}_{u,b}$ 
\[
\|\Delta\|_2 \leq \sqrt{N} \max_{p \in [t^k_s, t^k_e]} \|\delta p\| = \sqrt{N} \max_{p \in [t^k_s, t^k_e]} \left( \|s_p\| + \sum_{i=1}^{p-1} \left( \sum_{j=1}^{h} \|G[i]_p M[i]_{p-i} w_{p-i-j}\| + \sum_{i=h+1}^{p-1} \|G[i]_p \delta u^\text{DAC-O}_p\| \right) \right) 
\]
\[
\leq \sqrt{N} \max_{p \in [t^k_s, t^k_e]} \left( \|s_p\| + \sum_{i=1}^{p-1} \left( \sum_{j=1}^{h} \|M[i]_{p-i}\| \|w_{p-i-j}\| + \sum_{i=h+1}^{p-1} \|G[i]_p \| \|\delta u^\text{DAC-O}_p\| \right) \right) 
\]
\[
\leq \sqrt{N} \left( R_e + \frac{\kappa_h \kappa_b \kappa_m \kappa_u h}{\gamma} + \frac{\kappa_h \kappa_b R_u}{\gamma} \right).
\]

Then for $\zeta = \sqrt{N} \zeta_{\Delta}$, the probability of event $\mathcal{E}_\zeta$ is greater than $1 - \delta/2$. Then by union bound, with probability greater than $1 - \delta - N^{-\log(N)}$
\[
\left\|\hat{G}^\text{cld}_k - G_k\right\|_2 \leq \frac{2\sqrt{h} \zeta_{\Delta}}{\sqrt{N} \sigma} \left( \sqrt{n \log (2)} + 2 \log \left( \frac{2h}{\delta} \right) + \frac{\lambda \kappa_h \kappa_b}{\gamma \zeta_{\Delta} \sigma \sqrt{hN}} \right).
\]

Next, we bound the cumulative estimation error under this event. Let $\Gamma$ changes occur after $3t_p - 1$ time steps from the previous change. We call the intervals where the change occurs only after $3t_p - 1$ time steps as stationary intervals. We index the stationary intervals by $s$. Let the underlying system parameter be given by $G_t = G_s$ in a stationary interval $s$. Let the beginning and end of a stationary interval be denoted by $t^s_b$ and $t^s_e$. Let $\Gamma^s$ denote the number of changes between the stationary interval $s$ and the previous stationary interval $s - 1$.

Now, let there be a constant $c > 0$ such that $t_p \geq chm \log (2hm)^2 \log (2t_m m)^2$. This constant will be defined later. Then, by Lemma 6 and union bound, we have that with probability $1 - T/(N^2 + \log(N)) - 2\delta/3$
\[
\left\|\hat{G}^\text{cld}_k - G_k\right\|_2 \leq \frac{\beta (2\delta/(3T), \lambda, \sigma, h, N)}{\sigma \sqrt{N}},
\]

for all periods $k$ of the online estimator within the stationary intervals, where the underlying system parameter is given by $k$. In the following, let $\mathcal{E}$ denote the event under which the previous statement holds. Then, $\mathcal{E}$ occurs with probability $1 - T/(N^2 + \log(N)) - 2\delta/3$. Let us denote $\beta (2\delta/(3T), \lambda, \sigma, h, N)$ compactly by $\beta$.

Then, under event $\mathcal{E}$, the cumulative estimation error is given by
\[
\sum_{t=1}^{T} \left\|\hat{G}_t - G_t\right\|_2 = \sum_{s=1}^{\hat{F}} \sum_{t=t^s_b}^{t^s_e} \left\|\hat{G}_t - G_t\right\|_2 
\]
\[
\leq (a) \left\{ \sum_{s=1}^{\hat{F}} \sum_{t=t^s_b}^{t^s_e} \left\|\hat{G}_t - G_t\right\|_2 + \sum_{t=t^s_b}^{t^s_e} \left\|\hat{G}_t - G_t\right\|_2 + \sum_{t=t^s_b}^{t^s_e} \left\|\hat{G}_t - G_t\right\|_2 \right\} 
\]
\[
\leq (b) \left\{ \sum_{s=1}^{\hat{F}} \sum_{t=t^s_b}^{t^s_e} \left\|\hat{G}_t - G_t\right\|_2 + \sum_{t=t^s_b}^{t^s_e} \left\|\hat{G}_t - G_t\right\|_2 + \sum_{t=t^s_b}^{t^s_e} \left\|\hat{G}_t - G_t\right\|_2 \right\} 
\]
\[
\leq (c) \left\{ \sum_{s=1}^{\hat{F}} \left[ \frac{6 \kappa_h \kappa_b \Gamma^s(N + 2h)}{\gamma} + 2 \kappa_h \kappa_b (N + 2h) + \sum_{t=t^s_b}^{t^s_e} \left\|\hat{G}_t - G_t\right\|_2 \right] \right\} 
\]
\[ \leq (d) \sum_{s=1}^{\tilde{t}} \left[ \frac{6\kappa_a \kappa_b \bar{t}_s^a (N + 2h)}{\gamma} + 2\kappa_a \kappa_b (N + 2h) + \beta(s^s_t - s_{s-1}^s) \right] \]
\[ \leq (e) \frac{8\kappa_a \kappa_b \Gamma_T (N + 2h)}{\gamma} + \frac{\beta T}{\sigma \sqrt{N}}. \]

Here, (a) follows from splitting the interval \([t_{s-1}^s, t_s^s]\), (b) follows from recognizing that, under the given scenario, any non-stationary interval can be at the most \(2\bar{t}_s^s t_p\) and therefore \(G_t = G_s\) for \(t \in [t_{s-1}^s + 2\bar{t}_s^s t_p, t_s^s]\), (c) follows from the fact that \(\|\tilde{G}_t\|_2, \|G_t\|_2 \leq \kappa_a \kappa_b / \gamma\) and the fact that the estimator need not be accurate in any non-stationary interval and the first period of length \(t_p\) in any stationary interval, (d) follows from using Eq. (1).

Therefore, by union bound, the total regret is bounded by
\[ R_T \leq \frac{D^2}{\eta} + TG^2_f\eta + L_f h^2 \eta G_f T + \frac{2LRR_a \kappa_a \kappa_b (1 - \gamma)^h T}{\gamma} \]
\[ + 3LR \left(1 + \frac{\kappa_a \kappa_b}{\gamma}\right) \sqrt{m + \log(1/\delta)} T \sigma + \frac{16LRR_a \sqrt{h} \kappa_a \kappa_b \Gamma_T (N + 2h)}{\gamma} + \frac{2LRR_a \sqrt{h} \beta T}{\sigma \sqrt{N}}, \]
with probability \(1 - T/(N^{2+\log(N)}) - \delta\). Therefore, with \(N = \Gamma_T^{-0.8} T^{4/5}, \sigma = \Gamma_T^{0.2} T^{-1/5}, \eta = \frac{D}{\sqrt{G_f (G_f + L_f h^2) T}}, h = \left(\frac{\log(T)}{\log(1/\delta)}\right), \lambda = O(1)\), we get that
\[ R_T \leq \tilde{O} \left(\Gamma_T^{0.2} T^{4/5}\right), \]
with probability \(1 - T/(N^{2+\log(N)}) - \delta\).

Now, given that \(\Gamma_T = \tilde{O} \left(T^d\right), d < 1, \text{and } N = \Gamma_T^{-0.8} T^{4/5}, \text{N = O}(T^{4/5(1-d)})\). Therefore,
\[ \lim_{T \to \infty} T/(N^{2+\log(N)}) \leq \lim_{T \to \infty} T/N^{\log(N)} = 0. \]

Therefore, for \(\Gamma_T = \tilde{O} \left(T^d\right)\) and a given \(\tilde{\delta}\) such that \(\delta \leq \tilde{\delta}\), we can find a \(T_1\) such that, for all \(T > T_1, T/N^{2+\log(N)}\) is smaller than \(\tilde{\delta}\). Also, for \(\Gamma_T = \tilde{O} \left(T^d\right), t_p = N + h > N \geq O(T^{2/5(1-d)})\), and \(hm \log(2hm)^2 \log(2t_p m)^2 = O(\log(T)^5)\). Therefore, we can find a \(T_2\) and a constant \(c > 0\) such that, for all \(T > T_2, t_p \geq chm \log(2hm)^2 \log(2t_p m)^2\). Take \(T_0 = \max\{T_1, T_2\}\). Then, for \(T > T_0\), we can find a constant \(c\) such that the condition \(t_p \geq chm \log(2hm)^2 \log(2t_p m)^2\) holds, and \(T/N^{2+\log(N)}\) is smaller than \(\delta\). Therefore, for \(T > T_0\) the regret holds with probability \(1 - 2\delta\).

D Proof of Theorem 2: Setting S-1

The proof steps are exactly same as setting S-2. The differences are minor and we only highlight the differences here without repeating all the steps. Proceeding exactly as in Setting S-2 the constant \(R_u\) would be \(\frac{\kappa_a \kappa_b \kappa_w}{\gamma} + \kappa_a + \frac{2R_a \kappa_a \kappa_b}{\gamma}\). The other difference occurs in the cost truncation error:
\[ \sum_{t=1}^{T} c_t (y_t^{\text{TDAC-O}}, u_t^{\text{TDAC-O}}) - \sum_{t=1}^{T} c_t (y_t^{\text{TDAC-O}}[M_{t:t-h} | \tilde{G}_t, \hat{s}_{1:t}], u_t^{\text{TDAC-O}}) \leq LR \left\| y_t^{\text{TDAC-O}} - \hat{y}_t^{\text{TDAC-O}}[M_{t:t-h} | \tilde{G}_t, \hat{s}_{1:t}] \right\| \]
\[ \leq LR (R_u + \kappa_w) \sqrt{h} \left\| G_t - \hat{G}_t \right\| + \frac{\kappa_a \kappa_b}{\gamma} (\kappa_w + R_u). \]

The second difference lies in the following steps.
\[ \left\| \hat{G}_t^{\text{TDAC-O}}[M_x | \tilde{G}_t, \hat{s}_{1:t}] - \hat{G}_t^{\text{TDAC-O}}[M_x | G_t, \hat{s}_{1:t}] \right\| \]
\[ = \left\| \left( \hat{s}_t - s_t \right) + \sum_{k=1}^{h} \hat{G}_t^{[k]} \hat{u}_{t-k}^{\text{TDAC-O}}[M_x | w_{1:t}] - \sum_{k=1}^{h} G_t^{[k]} u_{t-k}^{\text{TDAC-O}}[M_x | w_{1:t}] \right\| \]
\[ = \left\| \left( \hat{s}_t - s_t \right) + \sum_{k=1}^{h} \left( \hat{G}_t^{[k]} - G_t^{[k]} \right) \hat{u}_{t-k}^{\text{TDAC-O}}[M_x | w_{1:t}] \right\| \leq \left\| \hat{s}_t - s_t \right\| + R_u \sqrt{h} \left\| \hat{G}_t - G_t \right\|_2. \]
\[ \leq (R_u + \kappa_w)\sqrt{h} \| G_t - \hat{G}_t \|_2 + \frac{\kappa_a \kappa_b R_u (1 - \gamma)^h}{\gamma}. \]

The final difference is a minor difference in Lemma 6. There it is mentioned that “\( M_{p-1} \) are dependent on the random perturbation inputs only through \( \bar{G}_{p-1} \) given the linearity of the cost functions and the update equation for \( M_t \).” In Setting S-1, \( M_{p-1}s \) are again dependent on the random perturbation inputs only through \( \bar{G}_{p-1} \), but not because of linearity, but because of the definition of \( \delta_t \) used in setting S-1. This covers all the differences. The rest of the proof is the same.

E Variant of OLC-ZK with Change Point Detection (OLC-ZK-CPD)

Here, we discuss the variant of our OLC-ZK algorithm that is observed to perform better numerically. The online control policy for this variant is the same as the OLC-ZK algorithm. The difference is in the estimation part. We discuss the estimation algorithm below.

A. Online Estimation:

The online estimation algorithm for OLC-ZK-CPD is a combination of a change point detection algorithm and a regular estimation algorithm. The change point detection algorithm detects changes larger than a certain threshold and resets the estimation algorithm upon every detection. The estimation algorithm continuously updates the estimates using all of the data from the last reset point. This offers the online learner more flexibility as it only resets whenever there is a significant underlying change, while it continues to refine the estimate otherwise. Thus, the change point detection approach can track the time variations more optimally. This we observe to be the case in the numerical simulations.

A.1. Change Point Detection:

The goal of the Change Point Detection (CPD) algorithm is to detect the underlying changes in the system reliably. To do this, the CPD algorithm runs a sequence of independent estimation algorithms each of duration \( t_p = N + h \) one after the other, where the estimation algorithms are the standard least-squares estimation applied to the data collected from the respective periods of duration \( t_p = N + h \). Here, \( t_p \) has to be necessarily greater than \( h \), since computing the estimate of \( G_t \) requires at least a length of \( h \) inputs. Essentially, the CPD algorithm ignores the past and only considers the recent history to compute an estimate of the system parameters. This allows the CPD algorithm to compute a reliable estimate of the current values of the parameters of the system provided \( N \) is of adequate size and not very large. Then, provided the estimation in each period of duration \( N \) is an accurate estimate of the system parameter values in the respective periods, any change point can be detected by comparing the estimates across the different periods. More specifically, if the estimate at the end of a period is greater than a certain threshold compared to the estimate from an earlier period, we can proclaim change point detection.

We denote the index of the successive periods of duration \( t_p \) by \( k \). We denote the start and end time of each of these periods by \( t_s^k \) and \( t_e^k \). Therefore, it follows that \( t_s^k = t_e^{k-1} \) for all \( k \). The CPD algorithm computes the following least-squares estimate at the end of each period \( k \)

\[
\hat{G}_k^{cd} = \arg\min_{\bar{G}} \sum_{p = t_e^{k-1} + h}^{t_e^k} \ell_p (\bar{G}) + \lambda \| \bar{G} \|_2^2,
\]

\[
\lambda > 0, \quad \ell_p (\bar{G}) = \left \| y_p^{\text{DAC-O}} - \sum_{l=1}^{h} \bar{G}^{[l]} \delta_k^{\text{DAC-O}} u_p^{\text{DAC-O}} \right \|_2^2.
\]

We denote the first period of duration \( t_p \), after a detection, as the baseline period with index \( k = 1 \). By default, the very first period of duration \( t_p \) at the beginning of the control episode is also a period with index \( k = 1 \). The CPD algorithm proclaims change point detection, when at the end of a period \( k \)

\[
\left \| \hat{G}_k^{cd} - \hat{G}_l^{cd} \right \|_2 > \frac{2 \beta}{\sigma \sqrt{N}}, \text{ for any } \ell \text{ s.t. } 1 \leq \ell < k,
\]

where \( \beta \) is a constant to be defined later.

A.2. System Estimation:
Algorithm 3 Online Learning Control with Change Point Detection (OLC-ZK-CPD) Algorithm

**Input:** Step size $\eta, \sigma, \beta, N, h$

Initialize $M_1 \in \mathcal{M}$ arbitrarily, $t_d = 1, k = 1, t_s = 1, t_e = N + h$

for $t = 1, \ldots, T$

1. Observe $y_{\pi \text{DAC} - O}^t$

2. if $t == t_e$

   - Estimate $\hat{G}_{k}^{\text{cd}}$ according to Eq. (12)

     - if $k > 1$

       - if $\| \hat{G}_k^{\text{cd}} - \hat{G}_\ell^{\text{cd}} \|_2 > \frac{(2)\beta}{\sigma \sqrt{N}}$ for any $1 \leq \ell < k$

         - Proclaim change point detection

         - Set $t_d = t$. Set $k = 1$

       - else

         - $k = k + 1$

   - else

     - $k = k + 1$

3. $t_s = t_e$, $t_e = t_s + N + h - 1$

end

Compute $\hat{G}_t$ according to Eq. (13)

Apply $u_t^{\pi_{\text{DAC-O}}}$ from Eq. (6)

Observe $c_t, u_t$ and incur cost $c_t(y_t^{\pi_{\text{DAC-O}}}, u_t^{\pi_{\text{DAC-O}}})$

Update: $M_{t+1} = \text{Proj}_M \left( M_t - \eta \frac{\delta \text{G}(\hat{G}_t, \hat{G}_s : t_s + 1)}{\delta M_t} \right)$

end

Upon detection of a change by the CPD algorithm, the online estimation algorithm restarts the estimation of the system parameters after a delay of $h$. Let $t_d$ denote the most recent time of detection by the CPD algorithm. Then, the estimate of the system parameters for any time $t \geq t_d + 2h$ is given by

$$
\hat{G}_t = \text{Proj}_\mathcal{G} (\hat{G}_t^*), \quad \hat{G}_t^* = \arg\min_{\hat{G}} \sum_{p=t_d + h}^{t-h} \ell_p (\hat{G}) + \lambda \| \hat{G} \|_2^2, \\
\ell_p (\hat{G}) = \left\| y_p^{\pi_{\text{DAC-O}}} - \sum_{l=1}^{h} \hat{G}_t[l] \delta u_{p-l}^{\pi_{\text{DAC-O}}} \right\|_2^2. 
$$

The complete algorithm with change point detection is given in Algorithm 3.