

A Hybrid Systems Model of Feedback Optimization for Linear Systems: Convergence and Robustness

Oscar Jed R. Chuy¹, Matthew T. Hale¹, and Ricardo G. Sanfelice²

Abstract—Feedback optimization algorithms compute inputs to a system using real-time output measurements, which helps mitigate the effects of disturbances. However, existing work often models both system dynamics and computations in either discrete or continuous time, which may not accurately model some applications. In this work, we model linear system dynamics in continuous time and the computation of inputs in discrete time, leading to a new hybrid system model of feedback optimization. First, we establish well-posedness of this hybrid model and establish completeness of solutions while ruling out Zeno behavior. Then, we show that the state of the system converges exponentially fast to a ball of known radius about a desired goal state. Next, we analytically show that this system is robust to perturbations over bounded (hybrid) time horizons in (i) the values of measured outputs, (ii) the matrices that model the linear time-invariant system, and (iii) the times at which inputs are applied to the system. Simulation results confirm that this approach successfully mitigates the effects of disturbances.

I. INTRODUCTION

Many automation tasks require optimizing the behavior of a dynamical system, which often involves solving a planning problem offline. With accurate system models, an optimization problem may be solved in a feedforward configuration to generate a reference that is used to drive the system in question [1]–[7]. However, errors in a system model can lead to sub-optimal solutions [1] because the inputs applied to a system may not actually produce the intended outputs.

If models are inaccurate, one alternative approach called “feedback optimization” instead *measures* system outputs [1], [3], [7]–[9] and then uses those measurements to optimize inputs with an in-the-loop optimization algorithm. Feedback optimization has been shown to have several benefits in certain settings: it is robust to inaccurate system models and time-varying parameters, achieves constraint satisfaction with minimal model dependence, and eliminates the need for pre-computed set points or reference signals [1], [3]. This approach has been used, for example, in decentralized settings [10], [11], gradient-based feedback control [9],

zeroth-order optimization [1], chemical processes [3], and network congestion control [12].

In existing work, feedback optimization has usually been applied to systems with either (a) continuous-time dynamics and a continuous-time optimization algorithm in the loop or (b) discrete-time dynamics and a discrete-time optimization algorithm in the loop [1], [13], [14]. However, physical systems are often modeled in continuous time and digital computers are naturally modeled in discrete time, which means that practical implementations can produce dynamics that are not captured by (a) or (b).

We seek to show that feedback optimization retains its robustness guarantees with continuous-time dynamics and discrete-time optimization, and we therefore develop a hybrid systems model of it.

The contributions of this paper are the following:

- We model feedback optimization as a hybrid system and show that (i) it is free from Zeno behavior and (ii) all maximal solutions are complete (Proposition 1).
- We bound the distance between the state of the hybrid feedback optimization model and a desired goal state (Theorem 1 and Theorem 2).
- We prove robustness of the hybrid model to perturbations by showing that over bounded (hybrid) time horizons there is bounded difference between the solutions to a perturbed system and the solutions to a nominal system (Theorem 3).
- We validate with simulations that hybrid feedback optimization successfully rejects disturbances (Section VI).

Prior work in [15]–[17] combines continuous-time dynamics and discrete-time optimization in a sampled-data feedback optimization setting, though we develop new analytical robustness guarantees by developing a hybrid model in the framework of [18]. The main results in [15], [16] show that the closed-loop system is stable for large enough sample times and that it is practically stable under time-varying disturbances with bounded variation. Results in [17] show global exponential stability of the closed-loop system when inputs to the system change at a fixed rate (with periodic sampling times).

Our results consider a hybrid system model in the framework of [18], which allows us to characterize system behavior at all times (permits aperiodic sampling), and unlocks robustness analyses that we apply in this work. We use this model to show that hybrid feedback optimization is simultaneously robust to several types of disturbances, including disturbances in the values of measured outputs, errors in the times at which new inputs are applied to the

¹School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA USA. Emails: {ochuy3, matthale}@gatech.edu.

²School of Electrical and Computer Engineering, University of California, Santa Cruz, CA USA. Email: ricardo@ucsc.edu.

All authors were supported by AFOSR under grant FA9550-19-1-0169. Chuy and Hale were supported by ONR under grants N00014-21-1-2495, N00014-22-1-2435, and N00014-26-1-2068, and AFRL under grants FA8651-22-F-1052 and FA8651-23-F-A006. Sanfelice was supported by NSF Grants no. CNS-2039054 and CNS-2111688, by AFOSR Grants nos. FA9550-23-1-0145, FA9550-23-1-0313, and FA9550-23-1-0678, by AFRL Grant nos. FA8651-22-1-0017 and FA8651-23-1-0004, by ARO Grant no. W911NF-20-1-0253, and by DoD Grant no. W911NF-23-1-0158.

system, and errors in all of the matrices that model the linear time-invariant system that is being controlled. To the best of our knowledge, this paper is the first to analytically prove that a hybrid/sampled-data feedback optimization model is simultaneously robust to all of these disturbances.

The rest of the paper is organized as follows. Section II provides background. Section III gives problem statements and the hybrid feedback optimization model. Section IV derives properties of its solutions. Then, Section V establishes the solutions' convergence and the system's robustness. Section VI presents simulations. Due to space constraints, proofs will be published elsewhere.

II. PRELIMINARIES

This section gives background on feedback optimization.

A. Notation

Let \mathbb{R} denote the reals and let \mathbb{N} denote the non-negative integers. For a differentiable function $\Phi : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$, let $\nabla_u \Phi$ denote the partial derivative with respect to its first argument. The symbol I_b denotes the identity matrix of dimension b . The 2-norm of a vector x is denoted $\|x\|$. For a non-empty, compact, convex set \mathcal{Z} , the symbol $\Pi_{\mathcal{Z}}[v]$ denotes the Euclidean projection of a point v onto \mathcal{Z} , i.e., $\Pi_{\mathcal{Z}}[v] = \arg \min_{z \in \mathcal{Z}} \|v - z\|$. We denote the diameter of a non-empty, compact, convex set \mathcal{Z} by $d_{\mathcal{Z}}$. We use $\lambda_i(N)$ to denote the i^{th} eigenvalue of a matrix N , and we use $\Re\{a\}$ to denote the real part of a complex number a . We also write $\lambda_{\min}(N)$ and $\lambda_{\max}(N)$ for the smallest and largest (real) eigenvalues of a symmetric matrix N , respectively. Given $r \geq 0$, we use $B_r(\tilde{x}) := \{x \in \mathbb{R}^n : \|x - \tilde{x}\| \leq r\}$ to denote the closed Euclidean ball of radius r about the point $\tilde{x} \in \mathbb{R}^n$. For a set S and a point x , we use $\|x\|_S := \inf_{s \in S} \|x - s\|$. For vector elements x and y we denote $(x^\top \ y^\top)^\top = (x, y)$.

B. Feedback Optimization Background and Setup

Now we review "feedback optimization" as defined in the literature, e.g., [1], [19]. At a high level, this class of problems uses real-time measurements from a dynamical system that are fed into an optimization algorithm in a closed-loop structure. The goal in doing so is to compute inputs that optimize the steady-state behavior of the dynamical system.

To the best of our knowledge there has not been a systematic investigation of the robustness of hybrid feedback optimization for a continuous-time system driven by discrete-time computations. Other existing work has developed hybrid models of optimization algorithms [20]–[22], though we differ by developing a hybrid model of feedback optimization. There has also been work on hybrid model predictive control [23], but our aim is different because we implement a hybrid framework for feedback optimization. Work in [15]–[17] studies feedback optimization in a sampled-data setting, but our use of a hybrid model lets us derive new analytical robustness guarantees that include robustness to errors in the times at which new inputs are applied, perturbations to the values of sampled outputs, and errors in the plant model.

Suppose we have the linear time-invariant (LTI) system

$$\dot{x} = Ax + Bu \quad y = \Psi x + d, \quad (1)$$

where $x \in \mathbb{R}^n$ is the system's state, $u \in \mathcal{U} \subset \mathbb{R}^m$ is its input, \mathcal{U} is a non-empty, compact, convex set, and $y \in \mathbb{R}^p$ is its output. The vector $d \in \mathbb{R}^p$ is a constant, unknown disturbance (e.g., bias), which is a typical component of feedback optimization problem formulations [1], [9], [10], [13], [24]. Such disturbances arise for example, in various power systems applications [1], [9], [10], [13], [24].

Assumption 1. The matrix A is Hurwitz.

Without loss of generality, the eigenvalues of A are ordered such that $\lambda_n \leq \lambda_{n-1} \leq \dots \leq \lambda_1 < 0$.

Remark 1. We adopt the "stabilize then optimize" approach [10], [15] in which we suppose that a stabilizing controller has already been applied to the system. If Assumption 1 is not satisfied, then it can be enforced for any stabilizable system by doing pre-feedback with a stabilizing controller.

Assumption 1 ensures that (1) will eventually reach steady state when its input is constant, which allows its steady-state behavior to be optimized. From (1), the steady-state input-to-output map is $u \mapsto Hu + d$, where $H := -\Psi A^{-1}B$.

Remark 2. While one could envision using an estimator to determine d , we are interested in cases in which the matrices A , B , and Ψ are not known exactly, and an observer may have poor accuracy under these conditions. For implementation, exact knowledge of H is unnecessary as errors are allowed as covered in Section V-B.

To optimize the system's steady-state behavior, one can drive its input and output to a solution of

$$\underset{u, y}{\text{minimize}} \quad \Phi(u, y) \quad (2a)$$

$$\text{subject to} \quad y = Hu + d, \quad u \in \mathcal{U}, \quad y \in \mathbb{R}^p \quad (2b)$$

where $\Phi : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ is strongly convex in (u, y) . These properties ensure (2a)-(2b) has a unique solution.

By incorporating the constraint $y = Hu + d$ into (2a), one could in principle reduce this problem to

$$\underset{u}{\text{minimize}} \quad \tilde{\Phi}(u, Hu + d) \quad \text{subject to} \quad u \in \mathcal{U} \quad (3)$$

However, the problem in (3) cannot be solved in practice because the substitution $y = Hu + d$ would require exact knowledge of the disturbance d , which may not be available. Instead, feedback optimization is used to repeatedly measure y and then optimize over u . In this work, y is sampled at discrete instants of time, and we use y_s to denote its sampled value.

Remark 3. The underlying LTI system need not always be at steady state, but we will use a standard technique in the feedback optimization literature to approximate its outputs as coming from a system at steady state [8]–[10]. Mathematically, we approximate a sampled output y_s as

coming from the steady-state map $y_s = Hu + d$ when the input to the system is u . This approximation is justified, for example, when the dynamics of the system converge sufficiently quickly. Our results in Sections IV and V still analyze the dynamics for the system, and they use this approximation only to relate outputs to inputs. Since we optimize over u , we must account for u when deriving the gradient; however, in its computation we use the sampled value instead.

We study a closed-loop system that connects the LTI system with a gradient descent algorithm that computes the input u . With a sampled output y_s , the optimization update law is $u_{k+1} = \Pi_{\mathcal{U}}[u_k - \gamma \nabla_u \Phi(u_k, y_s)]$, where u_k is the k^{th} iterate of the gradient descent algorithm. Then, the closed-loop interconnected LTI system is

$$\begin{aligned} \text{Plant: } & \begin{cases} \dot{x} = Ax + Bu & y = \Psi x + d, \end{cases} & (4) \\ \text{Controller: } & \begin{cases} u_{k+1} = \Pi_{\mathcal{U}}[u_k - \gamma \nabla_u \Phi(u_k, y_s)]. \end{cases} \end{aligned}$$

We will formulate and analyze a hybrid model for this interconnection. Neither the changes in the input nor the measurements of the output will be assumed to occur periodically, but instead both can occur sporadically.

C. Background on Hybrid Systems

In this paper, a hybrid system \mathcal{H} takes the form

$$\mathcal{H} = \begin{cases} \dot{\zeta} \in F(\zeta) & \zeta \in C \\ \zeta^+ \in G(\zeta) & \zeta \in D \end{cases},$$

where $\zeta \in \mathbb{R}^n$ is the system's state vector and the maps F and G are set valued in general. The set-valued map F defines the flow map and governs the continuous dynamics within the flow set C , while G defines the jump map, which models the system's discrete behavior within the jump set D .

Definition 1 (Hybrid Basic Conditions [18]). A hybrid system \mathcal{H} with data (C, F, D, G) satisfies the hybrid basic conditions if

- 1) C and D are closed subsets of \mathbb{R}^n ;
- 2) $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is outer semicontinuous¹, and locally bounded² relative to C , $C \subset \text{dom } F$, and $F(\zeta)$ is convex for every $\zeta \in C$
- 3) $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is outer semicontinuous and locally bounded relative to D , and $D \subset \text{dom } G$.

If a hybrid system satisfies the hybrid basic conditions, then it is well-posed by [18, Theorem 6.30]. We use this property in Section V to show that errors in the models of F and G up to a certain threshold produce bounded changes in the resulting closed-loop system trajectories over compact

¹A set-valued mapping $M : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ is outer semicontinuous (osc) at $x \in \mathbb{R}^m$ if for every sequence of points $\{x_i\}_{i \in \mathbb{N}}$ convergent to x and any convergent sequence of points $\{y_i\}_{i \in \mathbb{N}}$ with $y_i \in M(x_i)$, one has $y \in M(x)$, where $\lim_{i \rightarrow \infty} y_i = y$ [18].

²A set-valued mapping $M : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ is locally bounded at $x \in \mathbb{R}^m$ if there is a neighborhood U_x of x such that $M(U_x) \subset \mathbb{R}^n$ is bounded [18].

time horizons. It is not automatic to formulate a well-posed hybrid system, and this paper will do so for a hybrid model of feedback optimization.

For a hybrid system \mathcal{H} , its solutions, denoted by ϕ , are hybrid arcs that can in general be maximal³, complete⁴, and Zeno⁵. Complete solutions are defined over arbitrarily long time horizons, and Zeno behavior implies that solutions undergo an infinite number of jumps in finite time. Zeno behavior implies that a system's states stop flowing in finite time, which we will rule out for feedback optimization.

III. A HYBRID MODEL FOR FEEDBACK OPTIMIZATION

The problems we solve in this paper are as follows.

Problem 1. Formulate a well-posed hybrid feedback optimization model of the plant and controller in (4) that captures (i) how outputs are intermittently measured and used in the optimization-based controller, and (ii) how inputs are computed and applied to the system.

Problem 2. Bound the steady-state error of the system relative to a desired goal state in terms of system parameters.

Problem 3. Show that the hybrid feedback optimization model is robust to perturbations in the LTI system model, measurements of outputs, and the times at which inputs are applied to the LTI system, in the sense that these perturbations induce bounded changes in solutions.

A. Overview of Hybrid Feedback Optimization

The continuous-time system in (4) receives inputs from the discrete-time optimization algorithm in (4), and those inputs only change at certain instants of time. Between these changes, inputs applied to the system are held constant. Similarly, the optimization algorithm measures an output of the system and uses it to perform some number of computations to optimize inputs. This sampled value of the output is held constant while optimizing an input, and it does not change until a new output is sampled.

The flow map F from Definition 1 will model the LTI system dynamics in (1) with piecewise constant inputs. The jump map G from Definition 1 will model both the sampling of outputs and the application of new inputs to the system. It will also model computations done by the optimization algorithm, which are generated at discrete points in time between the times at which the output value is sampled.

B. Optimization Problem Setup

We consider objectives Φ of the form

$$\Phi(u, y_s) = \frac{1}{2} u^\top Q_u u + \frac{1}{2} (y_s - \hat{y})^\top Q_y (y_s - \hat{y}), \quad (5)$$

³A solution ϕ to \mathcal{H} is maximal if there does not exist another solution ψ to \mathcal{H} such that $\text{dom } \phi$ is a proper subset of $\text{dom } \psi$ and $\phi(t, j) = \psi(t, j)$ for all $(t, j) \in \text{dom } \phi$ [18].

⁴The solution ϕ is complete if $\text{dom } \phi$ is unbounded, i.e., if $\text{length}(\text{dom } \phi) = \sup_t \text{dom } \phi + \sup_j \text{dom } \phi = \infty$ [18].

⁵The solution ϕ is Zeno if it is complete and $\sup_t \text{dom } \phi < \infty$ [18].

where $Q_u \in \mathbb{R}^{m \times m}$ and $Q_y \in \mathbb{R}^{p \times p}$ are symmetric and positive definite, and $\hat{y} \in \mathbb{R}^p$ is a constant, desired output value that is user-specified.

Quadratic objectives have been widely used in the feedback optimization literature [1], [10], [11], [13], [19], [25], and we emphasize that our hybrid model is not restricted to using quadratic objectives. Instead, in this initial work we focus on quadratic objectives to develop a hybrid systems model for feedback optimization, and we defer the study of other objectives (including nonconvex objectives) to future work that will build on the present paper.

C. Hybrid Modeling and Flow and Jump Sets

The state of the hybrid system includes $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ which are, respectively, the state and input of the LTI system in (1). It also includes the vector $y_s \in \mathbb{R}^p$, which is the value of the sampled output of the LTI system that is used in the underlying optimization algorithm, $z \in \mathbb{R}^m$, which is the current iterate of that optimization algorithm, $\tau_c \in \mathbb{R}$, which is a timer that tracks the amount of continuous time left until the input to the system changes, and $\tau_g \in \mathbb{R}$, which is a timer that accounts for the amount of continuous time needed to complete an iteration of the optimization algorithm. The full state is $\zeta := (x, u, y_s, z, \tau_c, \tau_g) \in \mathcal{X} := \mathbb{R}^{n+2m+p+2}$.

The timers τ_c and τ_g count down from some positive numbers to zero, and jumps occur only when they reach zero. The state is allowed to flow while both $\tau_c > 0$ and $\tau_g > 0$, and the state stops flowing and undergoes a jump when $\tau_c = 0$ and/or $\tau_g = 0$. These conditions are captured by the flow set C and jump set D , defined as

$$C := \{\zeta \in \mathcal{X} \mid \tau_c \in [0, \tau_{c,\max}], \tau_g \in [0, \tau_{g,\text{comp}}]\} \quad (6)$$

$$D := \{\zeta \in \mathcal{X} \mid \tau_c = 0 \text{ or } \tau_g = 0\}, \quad (7)$$

where $\tau_{c,\max} > 0$ is the maximum amount of time between changes in the input and $\tau_{g,\text{comp}} > 0$ is the amount of time needed to perform a gradient descent iteration.

D. Flow Map Definition

The flow map is derived from (1), which defines \dot{x} . We note that y_s , the sampled output used by the optimization algorithm, does not vary continuously because it is measured at certain time instants and is held constant between measurements. The timers τ_c and τ_g count down to zero continuously and with unit rate, while all other states only change during jumps. Therefore, the flow map is

$$F(\zeta) := (Ax + Bu, 0, 0, 0, -1, -1) \forall \zeta \in C. \quad (8)$$

E. Jump Map Definition

The jump map has three cases: (i) $\tau_g = 0$ with $\tau_c > 0$, (ii) $\tau_c = 0$ with $\tau_g > 0$, and (iii) $\tau_c = \tau_g = 0$.

In Case (i), a single gradient descent step has been completed, but since $\tau_c > 0$, the iterate z is not applied as the system input. The jump map for this case updates the state z using a gradient descent step of the form $z^+ = \Pi_{\mathcal{U}}[z - \gamma \nabla_u \Phi(z, y_s)]$, where $\gamma > 0$ is a stepsize. Here

$y_s \in \mathbb{R}^p$ is the most recently sampled value of the system output. The jump map resets τ_g to $\tau_{g,\text{comp}}$ so that the computation of a new gradient descent iteration can begin. Other states jump to their current values, which leaves them unchanged. The jump map for this case is

$$G_1(\zeta) = \begin{pmatrix} x \\ u \\ y_s \\ \Pi_{\mathcal{U}}[z - \gamma \nabla_u \Phi(z, y_s)] \\ \tau_c \\ \tau_{g,\text{comp}} \end{pmatrix} \quad \text{for all } \zeta \in D_1.$$

where $D_1 = \{\zeta \in \mathcal{X} : \tau_c > 0, \tau_g = 0\}$.

In Case (ii), a new input is applied to the system, a new output is sampled, and the timer τ_c resets to some point in the interval $[\tau_{c,\min}, \tau_{c,\max}]$, where $0 < \tau_{c,\min} \leq \tau_{c,\max}$. This range of times represents indeterminacy in the amount of time that elapses between the application of successive inputs to the system. When the input changes, it is set equal to z . When τ_c reaches 0, the LTI system output is sampled and stored in y_s based on Remark 3, which is held constant until the next sample. The states x and z do not change, and the jump map is

$$G_2(\zeta) = \begin{pmatrix} x \\ z \\ Hu + d \\ z \\ [\tau_{c,\min}, \tau_{c,\max}] \\ \tau_g \end{pmatrix} \quad \text{for all } \zeta \in D_2, \quad (9)$$

where, $D_2 = \{\zeta \in \mathcal{X} : \tau_c = 0, \tau_g > 0\}$ and as described in Remark 3, we approximate the sampled output y_s as $Hu + d$.

In Case (iii), we combine Cases (i) and (ii), and the system executes G_1 and then G_2 or G_2 and then G_1 . The full jump map G is defined as

$$G(\zeta) := \begin{cases} G_1(\zeta) & \text{if } \tau_c > 0 \text{ and } \tau_g = 0 \text{ Case (i)} \\ G_2(\zeta) & \text{if } \tau_c = 0 \text{ and } \tau_g > 0 \text{ Case (ii)} \\ G_3(\zeta) & \text{if } \tau_c = 0 \text{ and } \tau_g = 0 \text{ Case (iii)}, \end{cases} \quad (10)$$

where $G_3(\zeta) = G_1(\zeta) \cup G_2(\zeta)$. Then, the full hybrid model of feedback optimization is

$$\mathcal{H}_{FO} := (C, F, D, G), \quad (11)$$

where C is from (6), F is from (8), D is from (7), and G is from (10).

IV. PROPERTIES OF HYBRID FEEDBACK OPTIMIZATION

In this section, we show \mathcal{H}_{FO} satisfies certain technical conditions that ensure its solutions exist for all time, which completes our solution to Problem 1.

A. Well-Posedness and Existence of Solutions

Toward establishing that solutions to \mathcal{H}_{FO} are defined for all time, we have the following.

Lemma 1. The hybrid feedback optimization model \mathcal{H}_{FO} in (11) is well-posed in the sense that it satisfies Definition 1.

Proof. See authors' technical report [26] \square

The following result shows that all maximal solutions to the system \mathcal{H}_{FO} are complete and non-Zeno.

Proposition 1 (Completeness of Maximal Solutions). Consider the hybrid feedback optimization model \mathcal{H}_{FO} from (11). From every point in $C \cup D$ there exists a nontrivial solution. All maximal solutions are complete and non-Zeno.

Proof. See authors' technical report [26] \square

Remark 4. Even though Case (iii) in (10) introduces non-determinism (whether G_1 or G_2 is first), this is an intended property that only applies two consecutive jumps (hence, Zeno is not possible).

B. Algorithm Framework

We impose the following assumption about the computation of inputs and the value of the inputs applied.

Assumption 2. There exists $\ell \in \mathbb{N}$ with $\ell \geq 1$ such that $\ell\tau_{g,comp} \leq \tau_{c,min}$.

Assumption 2 ensures that, when the system is properly initialized, at least ℓ gradient descent iterations are performed between any consecutive changes in the input. This condition is a mild form of timescale separation and it will be used later in Theorem 1.

Consider an initial condition $\phi(0,0) = \nu \in \mathcal{X}$ that satisfies

$$\tau_c(0,0) \in [\tau_{c,min}, \tau_{c,max}], \tau_g(0,0) = \tau_{g,comp}, \text{ and} \\ z(0,0) = u(0,0), \quad (12)$$

and consider a solution ϕ to \mathcal{H}_{FO} with those initial conditions. With regards to the state component z , we add a subscript to help denote how many gradient descent iterations there have been in computing the next value of the input u . More generally, we use $\alpha(i)$ to denote the number of iterates that are performed when computing the $(i+1)^{th}$ value of the input u . Assumption 2 implies that $\alpha(i) \geq \ell \geq 1$ for all $i \in \mathbb{N}$.

At the initial hybrid time $(0,0)$, the initial input to the system $u(0,0)$ is applied and held constant. The system then performs $\alpha(0)$ gradient descent iterations (which are $\alpha(0)$ Case (i) jumps) before the input to the LTI system is changed. When a Case (ii) jump occurs and is the $(\alpha(0)+1)^{th}$ jump, a new input is applied $u(t_{\alpha(0)+1}, \alpha(0)+1)$ where $\alpha(0)$ Case (i) jumps and one Case (ii) jump have occurred. When computing $u(t_{\alpha(0)+1}, \alpha(0)+1)$, the k^{th} such iterate is denoted $z_k(t_k, k)$ for any $k \in \{0, 1, \dots, \alpha(0)\}$. For computing $u(t_{\alpha(0)+1}, \alpha(0)+1)$, we denote the last iterate $\alpha(0)$ by $z_{\alpha(0)}(t_{\alpha(0)}, \alpha(0))$.

When τ_c reaches zero, a Case (ii) jump occurs and is modeled in G_2 in (9). The input u is then set equal to the most recent optimization iterate, i.e. $u(t_{\alpha(0)+1}, \alpha(0)+1) = z_{\alpha(0)}(t_{\alpha(0)}, \alpha(0))$, and that iterate is used as the next initial iterate for computing the next input so $z_0(t_{\alpha(0)+1}, \alpha(0)+1) = z_{\alpha(0)}(t_{\alpha(0)}, \alpha(0))$, i.e., the initial iterate when computing the next input is set equal to the

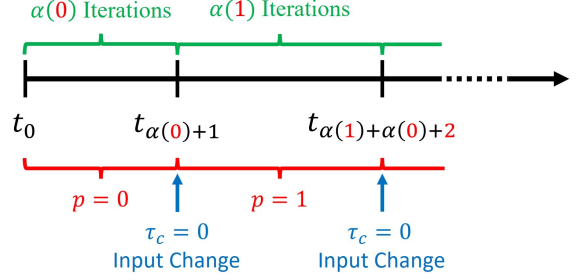


Fig. 1. A visual representation of the input evolution of \mathcal{H}_{FO} . There are $\alpha(p)$ iterations of gradient descent when computing the $(p+1)^{th}$ input. The input is changed at hybrid times of the form $(t_{\bar{\alpha}(p)+p}, \bar{\alpha}(p)+p)$.

final iterate that was generated when computing the previous input. For a generalized case with $p \in \mathbb{N}$, we use $\alpha(p)$ to denote the number of gradient descent iterations that are generated when computing the $(p+1)^{th}$ input to the system. We denote by $\bar{\alpha}(p)$ the total number of gradient descent iterations that have been computed for any input up until the p^{th} jump in u . That is, $\bar{\alpha}(0) = 0$ and $\bar{\alpha}(p) = \sum_{i=0}^{p-1} \alpha(i)$. Figure 1 illustrates changes in inputs for the first two jumps.

We can identify the general pattern that occurs at an arbitrary Case (i) jump. Suppose that p total Case (ii) jumps have occurred so far, which means that the input has changed p times. Then, when a Case (i) jump occurs, it computes an optimization iterate of the form

$$z_{k+1}(t_{\bar{\alpha}(p)+p+k+1}, \bar{\alpha}(p)+p+k+1) = \\ \Pi_U [z_k(t_{\bar{\alpha}(p)+p+k}, \bar{\alpha}(p)+p+k) \\ - \gamma \nabla_u \Phi(z_k(t_{\bar{\alpha}(p)+p+k}, \bar{\alpha}(p)+p+k))],$$

where for notational simplicity we have used

$$\nabla_u \Phi(z_k(t_{\bar{\alpha}(p)+p+k}, \bar{\alpha}(p)+p+k)) = \\ Q_u z_k(t_{\bar{\alpha}(p)+p+k}, \bar{\alpha}(p)+p+k) \\ + H^\top Q_y (y_s(t_{\bar{\alpha}(p)+p+k}, \bar{\alpha}(p)+p+k) - \hat{y}),$$

which by Remark 3 enforces how, for the optimization problem, we optimized over u but sampled y_s rather than computing it. The hybrid time $(t_{\bar{\alpha}(p)+p+k}, \bar{\alpha}(p)+p+k)$ accounts for $\bar{\alpha}(p)$ total optimization iterations that were computed up to the p^{th} change in the input, p changes in the input, and k optimization iterations that have been computed for the $(p+1)^{th}$ input.

For a Case (ii) jump, the $(p+1)^{th}$ input to the system is

$$u(t_{\bar{\alpha}(p+1)+p+1}, \bar{\alpha}(p+1)+p+1) = \\ z_{\alpha(p)}(t_{\bar{\alpha}(p)+\alpha(p)+p}, \bar{\alpha}(p)+\alpha(p)+p),$$

which indicates that the new value of the input is equal to the most recently computed optimization iterate. The initial iterate for computing the $(p+2)^{th}$ input is $z_0(t_{\bar{\alpha}(p+1)+p+1}, \bar{\alpha}(p+1)+p+1)$, and it is set equal to the same value as $u(t_{\bar{\alpha}(p+1)+p+1}, \bar{\alpha}(p+1)+p+1)$.

V. CONVERGENCE ANALYSIS

This section solves Problem 2 and bounds the distance from the state of the LTI system to its optimal steady-state value. The optimization problem in (2) with an objective of the form of (5) has a solution that we denote (\tilde{u}, \tilde{y}) , and this solution depends on the unknown disturbance d and the constant reference \hat{y} . The optimal steady-state value of x is $\tilde{x} = -A^{-1}B\tilde{u}$ (where A^{-1} exists under Assumption 1), and thus \tilde{x} incorporates the effects of d and \hat{y} .

A. Complete Hybrid Convergence

The next result is our first main result. It bounds the distance from the state of the underlying LTI system, namely x , to its optimal steady-state value, \tilde{x} . To state this result, we define $\mathcal{Y} = \{y \in \mathbb{R}^p : y = Hu + d, u \in \mathcal{U}\}$, which is compact due to \mathcal{U} being compact. Mathematically, we bound the distance between a solution ϕ of \mathcal{H}_{FO} and the set

$$\mathcal{A} := B_r(\tilde{x}) \times \mathcal{U} \times \mathcal{Y} \times \mathcal{U} \times [0, \tau_{c,\max}] \times [0, \tau_{g,\text{comp}}], \quad (13)$$

where $r = M \|B\| d_{\mathcal{U}} \rho^{-1} (2 - \exp(-\rho\tau_{c,\min}) + q^{\frac{\ell}{2}})$, $M \geq 1$ is a constant, $q := 1 - 2\gamma\lambda_{\min}(Q_u) + \gamma^2 L^2 \in (0, 1)$,

$$\rho = \min_{i \in \{1, \dots, n\}} |\Re\{\lambda_i(A)\}|, \quad (14)$$

and $\tilde{x} = -A^{-1}B\tilde{u}$, where (\tilde{u}, \tilde{y}) is the solution to (2a)-(2b). By definition, we have $\|\phi(t, j)\|_{\mathcal{A}} = \|x(t, j)\|_{B_r(\tilde{x})}$. Hence, bounding $(t, j) \mapsto \|\phi(t, j)\|_{\mathcal{A}}$ for each solution to \mathcal{H}_{FO} allows us to characterize the error between x and \tilde{x} while accounting for the full dynamics of \mathcal{H}_{FO} .

Theorem 1 (Complete Hybrid Convergence). Consider the hybrid system \mathcal{H}_{FO} from (11) and suppose that Assumptions 1 and 2 hold. Consider objectives of the form of (5), and suppose that the gradient descent algorithm uses a stepsize $\gamma \in (0, \frac{2}{\lambda_{\min}(Q_u) + L})$, where Q_u is from (5) and $L := \lambda_{\max}(Q_u + H^T Q_y H)$. For each maximal solution ϕ to \mathcal{H}_{FO} with initial condition $\phi(0, 0) = \nu$ that satisfies (12), for each $(t, j) \in \text{dom } \phi$,

$$\begin{aligned} \|\phi(t, j)\|_{\mathcal{A}} &\leq M \exp(-\rho t) \|\phi(0, 0)\|_{\mathcal{A}} \\ &+ \frac{M^2 \|B\| d_{\mathcal{U}}}{\rho} \left(2 - \exp(-\rho\tau_{c,\max}) + q^{\frac{\ell}{2}}\right) \exp(-\rho t) \\ &- \frac{M \|B\| d_{\mathcal{U}}}{\rho} \left(1 + q^{\frac{\ell}{2}} \exp(\rho\tau_{c,\min})\right) \exp(-\rho t) \end{aligned}$$

where $d_{\mathcal{U}} = \max_{u_1, u_2 \in \mathcal{U}} \|u_1 - u_2\|$ is the diameter of the set \mathcal{U} , ρ is from (14), $\ell \geq 1$ is from Assumption 2, \mathcal{A} is from (13), $q := 1 - 2\gamma\lambda_{\min}(Q_u) + \gamma^2 L^2 \in (0, 1)$, and $M \geq 1$ is a constant. In particular, each such solution satisfies $\lim_{t+j \rightarrow \infty} \|\phi(t, j)\|_{\mathcal{A}} = 0$.

Proof. See authors' technical report [26] \square

Remark 5. In the limit, the state x is asymptotically no farther than a distance r from \tilde{x} . The quantity r can be reduced with a pre-feedback controller that is applied before feedback optimization is used. More precisely, if the pair (A, B) is controllable, then using pole placement we can use any $\eta > 0$ to select ρ such that $\rho \geq M \|B\| d_{\mathcal{U}} \eta^{-1} (2 -$

$\exp(-\rho\tau_{c,\min}) + q^{\frac{\ell}{2}})$. This ρ ensures $r \leq \eta$ and hence that $\|x(t, j)\|_{B_\eta(\tilde{x})} \rightarrow 0$ for any desired $\eta > 0$.

B. Global Convergence and Robustness

Theorem 1 relies on (12), which restricts it to only apply to initial conditions with $\tau_c(0, 0) \in [\tau_{c,\min}, \tau_{c,\max}]$ and $\tau_g(0, 0) = \tau_{g,\text{comp}}$. We next derive a global convergence result that applies to solutions that begin from arbitrary initial conditions, including those that violate the conditions in (12).

Theorem 2 (Global Complete Hybrid Convergence). Consider the hybrid system \mathcal{H}_{FO} from (11) and suppose that Assumptions 1 and 2 hold. Consider objectives of the form of (5), and suppose that the gradient descent algorithm uses a stepsize $\gamma \in (0, \frac{2}{\lambda_{\min}(Q_u) + L})$, where Q_u is from (5) and $L := \lambda_{\max}(Q_u + H^T Q_y H)$. For each maximal solution ϕ to \mathcal{H}_{FO} with initial condition $\phi(0, 0)$, for each $(t, j) \in \text{dom } \phi$,

$$\begin{aligned} \|\phi(t, j)\|_{\mathcal{A}} &\leq M \exp(-\rho t) \|\phi(0, 0)\|_{\mathcal{A}} \\ &+ \frac{M^2 \|B\| d_{\mathcal{U}}}{\rho} \left(2 - \exp(-2\rho\tau_{c,\max}) + q^{\frac{\ell}{2}}\right) \exp(-\rho t) \\ &- \frac{M \|B\| d_{\mathcal{U}}}{\rho} \left(1 + q^{\frac{\ell}{2}} \exp(\rho\tau_{c,\min})\right) \exp(-\rho t), \end{aligned}$$

where $d_{\mathcal{U}} = \max_{u_1, u_2 \in \mathcal{U}} \|u_1 - u_2\|$ is the diameter of the set \mathcal{U} , ρ is from (14), $\ell \geq 1$ is from Assumption 2, \mathcal{A} is from (13), $q := 1 - 2\gamma\lambda_{\min}(Q_u) + \gamma^2 L^2 \in (0, 1)$, and $M \geq 1$ is a constant.

Proof. See authors' technical report [26] \square

In Theorem 2 one can use the value of ρ in Remark 5 to force x to converge to a ball of any radius $\eta > 0$ about \tilde{x} .

We can represent modeling errors as perturbations applied to the nominal system \mathcal{H}_{FO} . We consider both errors in the LTI dynamics and errors in the timer dynamics. We first consider the perturbed domain of the flow map, which is

$$\begin{aligned} \mathcal{C}_\ell &:= \{\zeta \in \mathcal{X} : \tau_c \in [0, \tau_{c,\max} + \theta_{c,\max}], \\ &\tau_g \in [0, \tau_{g,\text{comp}} + \theta_{g,\text{comp}}]\}, \quad (15) \end{aligned}$$

where $\theta_{c,\max} \in (-\tau_{c,\max}, \infty)$ and $\theta_{g,\text{comp}} \in (-\tau_{g,\text{comp}}, \infty)$. These perturbations allow τ_c to take values larger than $\tau_{c,\max}$ and similar for τ_g . Although timers may be reset to inaccurate values, jumps are still triggered when at least one timer reaches zero and hence we use $D_\ell := D$ in the perturbed system model.

In the flow map, there may be model errors in the A and B matrices, and the two timers may count down at a rate that is not exactly 1. We define the perturbed flow map as

$$F_\ell(\zeta) := \begin{pmatrix} (A + \hat{A})x + (B + \hat{B})u \\ 0 \\ 0 \\ 0 \\ -1 + \kappa_c \\ -1 + \kappa_g \end{pmatrix},$$

where $\hat{A} \in \mathbb{R}^{n \times n}$ models errors in the A matrix, $\hat{B} \in \mathbb{R}^{n \times m}$ models errors in the B matrix, $\kappa_c \in (-\infty, 1)$ models errors in the rate at which τ_c counts down, and $\kappa_g \in (-\infty, 1)$ models errors in the rate at which τ_g counts down.

Finally we define the perturbed jump map as

$$G_\iota(\zeta) := \begin{cases} G_{1,\iota}(\zeta) & \text{if } \tau_c > 0 \text{ and } \tau_g = 0 & \text{Case (i)} \\ G_{2,\iota}(\zeta) & \text{if } \tau_c = 0 \text{ and } \tau_g > 0 & \text{Case (ii)} \\ G_{3,\iota}(\zeta) & \text{if } \tau_c = 0 \text{ and } \tau_g = 0 & \text{Case (iii)} \end{cases}.$$

For $G_{1,\iota}$ we have

$$G_{1,\iota}(\zeta) := \begin{pmatrix} x \\ u \\ y_s \\ \Pi_{\mathcal{U}}[z - \gamma \nabla_u \Phi(z, y_s)] \\ \tau_c \\ \tau_{g,comp} + \theta_{g,comp} \end{pmatrix},$$

where $\theta_{g,comp}$ is from (15). This perturbed jump map allows for τ_g to be reset to values other than $\tau_{g,comp}$ after a gradient descent iteration is performed. For $G_{2,\iota}$ we have

$$G_{2,\iota}(\zeta) := \begin{pmatrix} x \\ z \\ (H + \hat{H})u + d \\ z \\ [\tau_{c,min} + \theta_{c,min}, \tau_{c,max} + \theta_{c,max}] \\ \tau_g \end{pmatrix}.$$

Here, $\hat{H} \in \mathbb{R}^{p \times m}$ models perturbations to the matrix H , including those that come from the errors \hat{A} and \hat{B} as described above, as well as errors in the output map Ψ . The interval to which τ_c is reset is perturbed with constants $\theta_{c,min} \in (-\tau_{c,min}, \infty)$ and $\theta_{c,max} \in (-\tau_{c,max}, \infty)$ that satisfy $0 < \tau_{c,min} + \theta_{c,min} \leq \tau_{c,max} + \theta_{c,max}$, which ensures that τ_c is reset to a non-empty set, though both endpoints of the interval can be perturbed. Finally we have $G_{3,\iota}(\zeta) := G_{1,\iota}(\zeta) \cup G_{2,\iota}(\zeta)$. We define

$$\iota = \max\{\theta_{g,comp}, \|\hat{A}x\|, \|\hat{B}u\|, \|\hat{H}u\|, \kappa_c, \kappa_g, \theta_{c,min}, \theta_{c,max}\} \quad (16)$$

to be the maximum size of any perturbation at the state $\zeta \in \mathcal{X} := \mathbb{R}^{n+2m+p+2}$. The perturbed hybrid system model is

$$\mathcal{H}_{FO}^\iota = \begin{cases} \dot{\zeta} \in F_\iota(\zeta) & \zeta \in C_\iota \\ \zeta^+ \in G_\iota(\zeta) & \zeta \in D_\iota \end{cases}.$$

Our next theorem provides robustness guarantees for \mathcal{H}_{FO} . First, we require the following definition.

Definition 2 ((τ, ϵ)-closeness [18, Definition 5.23]). Given $\tau, \epsilon > 0$, two hybrid arcs ϕ_1 and ϕ_2 are (τ, ϵ)-close if

- 1) for all $(t, j) \in \text{dom } \phi_1$ with $t+j \leq \tau$ there exists s such that $(s, j) \in \text{dom } \phi_2$, $|t-s| < \epsilon$, and $|\phi_1(t, j) - \phi_2(s, j)| < \epsilon$;
- 2) for all $(t, j) \in \text{dom } \phi_2$ with $t+j \leq \tau$ there exists s such that $(s, j) \in \text{dom } \phi_1$, $|t-s| < \epsilon$, and $|\phi_2(t, j) - \phi_1(s, j)| < \epsilon$.

To the best of our knowledge, the next result is the first analytical characterization of the robustness of feedback optimization in a hybrid or sampled-data setting.

Theorem 3 (Robustness of \mathcal{H}_{FO}). Consider the hybrid system \mathcal{H}_{FO}^ι with ι as defined in (16), and suppose that Assumptions 1 and 2 hold. Consider objectives of the form of (5), and suppose that the gradient descent algorithm uses a stepsize $\gamma \in \left(0, \frac{2}{\lambda_{min}(Q_u) + L}\right)$. Then, for every $\epsilon > 0$ and $\tau > 0$, there exists $\delta > 0$ with the following property: for every solution ϕ_δ to $\mathcal{H}_{FO}^{\delta\iota}$, there exists a solution ϕ to \mathcal{H}_{FO} such that ϕ_δ and ϕ are (τ, ϵ) -close.

Proof. See authors' technical report [26] \square

For perturbations in the model and timers, Theorem 3 only holds over bounded hybrid time horizons, so the result is applied up until a chosen time τ . Then for some error ϵ , there is a nonzero perturbation $\delta\iota$ of the perturbed system $\mathcal{H}_{FO}^{\delta\iota}$ such that its solution ϕ_δ is (τ, ϵ) -close to the solutions of the unperturbed system \mathcal{H}_{FO} .

VI. SIMULATION RESULTS

In this section we present two sets of simulation results: one with the nominal system \mathcal{H}_{FO} and one with the perturbed system \mathcal{H}_{FO}^ι . We consider the LTI dynamics

$$\dot{x} = \begin{bmatrix} -3 & 0 & 0 & 0 \\ 3 & -3 & 0 & 0 \\ 0 & 3 & -3 & 0 \\ 0 & 0 & 3 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \quad y = x,$$

and the feedback optimization problem we solve is

$$\begin{aligned} & \text{minimize}_{u, y} \quad \Phi(u, y) := \frac{1}{2} Q_u u^2 + \frac{1}{2} (y - \hat{y})^\top Q_y (y - \hat{y}) \\ & \text{subject to} \quad y = Hu + d, \quad u \in \mathcal{U}, \quad y \in \mathbb{R}^4 \end{aligned}$$

where we use $Q_u = 0.08$, $Q_y = 0.3I_4$, $\mathcal{U} = [-5, 5]$, $d = (0.2, 0.2, 0.2, 0.2)^\top$, and $\hat{y} = (4, 4, 4, 4)^\top$.

For simulations, the Hybrid Equations Toolbox (Version 3.0.0.76 [27]) was used, along with the initial conditions

$$\begin{aligned} x(0, 0) &= (0, 5, 10, 15)^\top, \quad u(0, 0) = 0, \quad \tau_c(0, 0) = 0.175, \\ y_s(0, 0) &= x(0, 0) + d, \quad z(0, 0) = 0, \quad \tau_g(0, 0) = 0.05, \end{aligned} \quad (18)$$

where $\tau_{g,comp} = 0.05$, $\tau_{c,min} = 0.15$, and $\tau_{c,max} = 0.20$ which ensures at least three iterations with stepsize $\gamma = 0.35 \leq \frac{2}{\lambda_{min}(Q_u) + L}$. We see in the left-hand plot of Figure 2 that $\|x - \tilde{x}\|$ converges to $3.26 \cdot 10^{-15}$, and thus x is asymptotically close to its desired value.

We next consider the perturbed case. Let $J_{a,b}$ denote the matrix of all ones in $\mathbb{R}^{a \times b}$. For the timers, we use perturbations of the form $\theta_{c,min} = \theta_{c,max} = \theta_{g,comp} = 0.5$ and $\kappa_c = \kappa_g = \kappa$ for several values of κ , and for the LTI dynamics we use perturbations of the form $\hat{A} = \mu J_{n \times n}$, $\hat{B} = \mu J_{n \times m}$, and $\hat{H} = \mu J_{p \times m}$ for several values of μ . The values of κ and μ used in simulations are shown in Table I, along with the values they induce in the steady-state error $\|x - \tilde{x}\|$ (from 0.86 to 12.22). Using these values, the perturbation ι from (16) can be quite large because it is the maximum of several terms, including $\|\hat{A}x\|$, $\|\hat{B}u\|$, and $\|\hat{H}u\|$, which can be large because they depend on x or u . A second-order regression takes the form $\|x - \tilde{x}\| \approx$

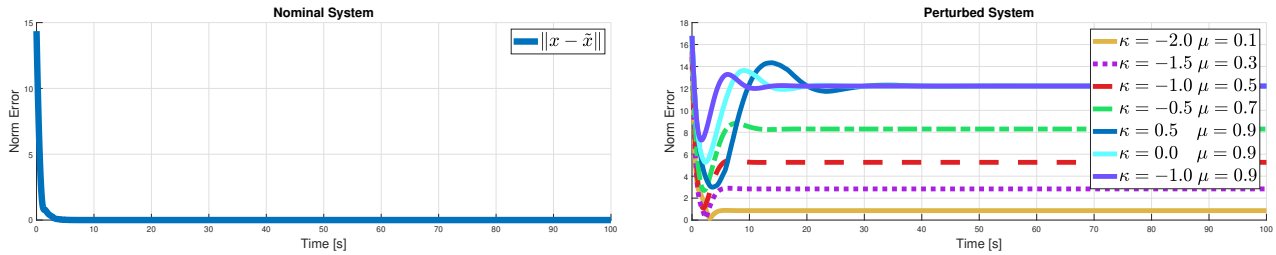


Fig. 2. The left-hand plot shows the convergence of the state x under the dynamics \mathcal{H}_{FO} from the initial condition in (18). We see that x attains an asymptotic error of $3.26 \cdot 10^{-15}$, which illustrates that it converges quite close to its desired value. The right-hand plot shows the convergence of the state x under the perturbed dynamics \mathcal{H}_{FO}^l from the initial condition in (18). In this case, x attains an asymptotic error of at most 12.22 relative to the errors from \mathcal{H}_{FO} under large model and timer perturbations. Under these perturbations, Assumption 1 still guarantees exponential convergence.

TABLE I

VALUES OF PERTURBATIONS AND THE ASYMPTOTIC ERROR FOR THE PERTURBED SYSTEM \mathcal{H}_{FO}^l RELATIVE TO \mathcal{H}_{FO} .

κ	Value of $\ x - \tilde{x}\ $				
	-2.0	-1.5	-1.0	-0.5	0.5
$\mu = 0.1$	0.86	0.86	0.86	0.86	0.86
$\mu = 0.3$	2.84	2.84	2.84	2.84	2.84
$\mu = 0.5$	5.26	5.26	5.26	5.26	5.26
$\mu = 0.7$	8.30	8.30	8.30	8.30	8.30
$\mu = 0.9$	12.22	12.22	12.22	12.22	12.22

$0.217 + 6.054\mu + 8.036\mu^2$, showing that the steady-state error is dependent on the model perturbations rather than the time errors. This agrees with the right-hand plot of Figure 2 where κ affected the convergence rate.

VII. CONCLUSION

This paper presented a hybrid system model for feedback optimization that considers continuous-time dynamics with discrete-time optimization. We showed that its maximal solutions are complete and non-Zeno, and then we bounded their distance to a desired goal state. We also presented what are, to the best of our knowledge, the first analytical robustness results for feedback optimization in a hybrid/sampled-data setting. Future work includes using nonconvex objective functions and using hybrid feedback optimization for systems with nonlinear dynamics.

REFERENCES

- [1] A. Hauswirth, Z. He, S. Bolognani, G. Hug, and F. Dörfler, "Optimization algorithms as robust feedback controllers," *Annual Reviews in Control*, 2024.
- [2] H. Xu, A. D. Domínguez-García, and P. W. Sauer, "Optimal tap setting of voltage regulation transformers using batch reinforcement learning," *IEEE Trans. Power Syst.*, 2020.
- [3] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III, *Process Dynamics and Control*, 4th ed. John Wiley & Sons, 2010.
- [4] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," in *Proceedings of the 2000 IEEE Conference on Computer Communications*, 2000.
- [5] A. Jokic, M. Lazar, and P. P. J. van den Bosch, "On constrained steady-state regulation: Dynamic kkt controllers," *IEEE Trans. Autom. Control*, 2009.
- [6] F. Bignucolo, R. Caldon, and V. Prandoni, "Radial mv networks voltage regulation with distribution management system coordinated controller," *Electric Power Systems Research*, 2008.
- [7] L. Ortman, A. Hauswirth, I. Caduff, F. Dörfler, and S. Bolognani, "Experimental validation of feedback optimization in power distribution grids," *Electric Power Systems Research*, 2020.
- [8] S. Menta, A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Stability of dynamic feedback optimization with applications to power systems," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2018.
- [9] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Timescale separation in autonomous optimization," *IEEE Trans. Autom. Control*, 2021.
- [10] W. Wang, Z. He, G. Belgioioso, S. Bolognani, and F. Dörfler, "Decentralized feedback optimization via sensitivity decoupling: Stability and sub-optimality," in *2024 European Control Conference (ECC)*, 2024.
- [11] G. Behrendt, M. Longmire, Z. I. Bell, and M. Hale, "Distributed asynchronous discrete-time feedback optimization," *IEEE Trans. Autom. Control*, 2024.
- [12] S. Low and D. Lapsley, "Optimization flow control. i. basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, 1999.
- [13] W. Wang, Z. He, G. Belgioioso, S. Bolognani, and F. Dörfler, "Online feedback optimization over networks: A distributed model-free approach," in *2024 IEEE 63rd Conference on Decision and Control*, 2024.
- [14] M. Ellis, H. Durand, and P. D. Christofides, "A tutorial review of economic model predictive control methods," *Journal of Process Control*, 2014.
- [15] G. Belgioioso, D. Liao-McPherson, M. Hudoba de Bady, S. Bolognani, R. S. Smith, J. Lygeros, and F. Dörfler, "Online feedback equilibrium seeking," *IEEE Trans. Autom. Control*, 2025.
- [16] L. Cothren, G. Bianchin, S. Dean, and E. Dall'Anese, "Perception-based sampled-data optimization of dynamical systems," 2023.
- [17] Y. Chen, F. Bullo, and E. Dall'Anese, "Sampled-data systems: Stability, contractivity and single-iteration suboptimal mpc," 2025.
- [18] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012.
- [19] D. Krishnamoorthy and S. Skogestad, "Real-time optimization as a feedback control problem – a review," *Comput. Chem. Eng.*, 2022.
- [20] K. R. Hendrickson, D. M. Hustig-Schultz, M. T. Hale, and R. G. Sanfelice, "Exponentially converging distributed gradient descent with intermittent communication via hybrid methods," in *2021 60th IEEE Conference on Decision and Control*, 2021.
- [21] D. M. Hustig-Schultz, K. Hendrickson, M. Hale, and R. G. Sanfelice, "A totally asynchronous block-based heavy ball algorithm for convex optimization," in *2023 American Control Conference (ACC)*, 2023.
- [22] K. R. Hendrickson, D. M. Hustig-Schultz, M. T. Hale, and R. G. Sanfelice, "Distributed nonconvex optimization with exponential convergence rate via hybrid systems methods," 2025.
- [23] B. Altun, P. Ojaghi, and R. G. Sanfelice, "A model predictive control framework for hybrid dynamical systems," *IFAC-PapersOnLine*, 2018.
- [24] M. Colombino, E. Dall'Anese, and A. Bernstein, "Online optimization as a feedback controller: Stability and tracking," *IEEE Transactions on Control of Network Systems*, 2020.
- [25] L. S. P. Lawrence, J. W. Simpson-Porco, and E. Mallada, "Linear-convex optimal steady-state control," *IEEE Trans. Autom. Control*, 2021.
- [26] O. J. Chuy, M. Hale, and R. Sanfelice, "A hybrid systems model of feedback optimization for linear systems: Convergence and robustness," 2025.
- [27] R. Sanfelice, D. Copp, and P. Nanez, "A toolbox for simulation of hybrid systems in matlab/simulink," in *Proceedings of the 16th international conference on Hybrid systems: computation and control - HSCC '13*. ACM Press, 2013, pp. 101–106.