The Internal Model Principle of Time-Varying Optimization

Gianluca Bianchin, *Member, IEEE*, and Bryan Van Scoy, *Member, IEEE*

***Abstract*—Time-varying optimization problems are central to many engineering applications, where performance metrics and system constraints evolve dynamically with time. Several algorithms have been proposed to address these problems; a common characteristic among them is their implicit reliance on knowledge of the optimizers' temporal variability. In this paper, we provide a fundamental characterization of this property: we show that an algorithm can track time-varying optimizers if and only if it incorporates a model of the temporal variability of the optimization problem. We refer to this concept as the *internal model principle of time-varying optimization.* Our analysis relies on showing that time-varying optimization problems can be recast as output regulation problems and, by using tools from center manifold theory, we establish necessary and sufficient conditions for exact asymptotic tracking. As a result, these findings enable the design of new algorithms for time-varying optimization. We demonstrate the effectiveness of the approach through numerical experiments on both synthetic problems and the dynamic traffic assignment problem from traffic control.**

***Index Terms*—Time-varying optimization, Internal model principle, Gradient regulation, traffic control.**

I. Introduction

TIME-VARYING optimization problems play a central role in several scientific domains, as they underpin many important contemporary engineering problems. Examples include training in Machine Learning [1], [2], dynamic signal estimation in Signal Processing [3], trajectory tracking in Robotics [4], system optimization in Industrial Control [5], and much more. Historically, discrete-time algorithms for time-varying optimization have been proposed and studied first since they emerged as a natural extension of their time-invariant counterparts, allowing for cost functions and constraints that may change over time [6]. These approaches build on the classical perspective on mathematical optimization, which seeks to construct methods to determine optimizers and consist of iterative procedures implemented on digital devices. In recent years, however, numerous optimization algorithms have been formulated and studied as continuous-time processes [7]–[9], mainly due to the wide availability of tools for Ordinary Differential Equations (ODEs) that can facilitate their analysis [9].

G. Bianchin is with the ICTEAM institute and the Department of Mathematical Engineering (INMA) at the University of Louvain, Belgium. Email: gianluca.bianchin@uclouvain.be

B. Van Scoy is with the Department of Electrical and Computer Engineering, Miami University, Oxford, OH 45056, USA. Email: bvanscoy@miamioh.edu

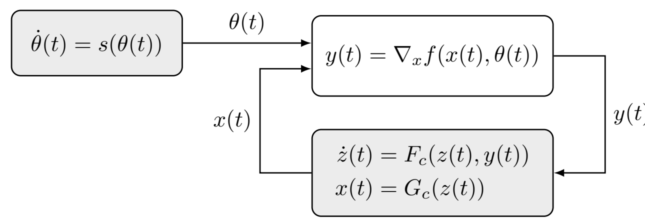


Fig. 1: Architecture of the gradient-feedback design scheme studied in this work. An optimization algorithm is to be designed (bottom block), having access only to gradient evaluations of the loss function to be minimized (top right block), and generating a sequence of exploration points $x(t)$ at which the gradient shall be evaluated. The loss function to be optimized varies with time, where the temporal variability $\theta(t)$ is assumed to be unmeasurable and generated by an exosystem (top left block). Shaded blocks emphasize the presence of dynamics.

Motivated by these recent developments, in this paper, we study time-varying convex optimization problems and focus on the use of continuous-time dynamics to track exactly optimal solutions. Several approaches have already been developed for this purpose [10]–[13], yet all these techniques implicitly require full knowledge of the temporal variability of the problem [13]. Unfortunately, in most practical applications, having such knowledge is impractical, either because the temporal variability enters the optimization in the form of exogenous disturbances that are unknown and cannot be measured (see, e.g., [14], [15]), or simply because it is unrealistic to ask for a noiseless model of how the problem changes with time. Departing from this, in this paper we pose the following question: is it possible to track (exactly and asymptotically) a minimizer of a time-varying optimization problem without any knowledge of the temporal variability of the optimization? Interestingly, we prove a fundamental result showing that tracking can be achieved if and only if the temporal variability of the problem can be 'observed' by the algorithm, and the latter incorporates a suitably reduplicated model of such a variability. We refer to this conclusion as *the internal model principle of time-varying optimization,* akin to its control-theoretic counterpart [16], [17]. Our approach relies on reinterpreting the optimization algorithm design as a nonlinear, multivariable regulation problem [18], and our analysis uses tools from center manifold theory [17], [19]. Fig. 1 illustrates the architecture of the gradient-feedback design scheme studied in this paper.

Related works. The literature on methods for time-varying optimization is mainly divided into two classes of solutions. The first class consists of methods that do not utilize any model of the temporal variability of the problem [20]–[23]; instead, they seek to solve a sequence of static problems. Several established approaches belong to this class, including the online gradient descent method [24] and the online Newton step algorithm [25] – see [26] and references therein. Because the temporal variability of the problem is unknown (or ignored), algorithms in this class can make a decision only *after* each variation has been observed, thus incurring a certain 'regret.' Mathematically, these techniques are capable of reaching only a neighborhood of an optimizer [13], and exact tracking is out of reach in general for these approaches. Moreover, suitable assumptions need to be made to provide convergence guarantees for these methods; we refer to [13], [26] and references therein for a detailed discussion. In contrast, the second class of methods uses a model of the temporal evolution of the problem to exactly track the optimal trajectories [2], [27]–[29]. Particularly celebrated is the prediction-correction algorithm (see [10], the recent work [11], and the discrete implementations [30]), whereby at each time a prediction step is used to anticipate how the optimizer will evolve over time, and a correction step is used to seek a solution to each instantaneous optimization problem. We refer to [13] for a recent overview of the topic. Recent years have witnessed a growing interest in this problem: [4] uses contraction to study these methods; [31] uses sampling to estimate the temporal variability of the problem; a recent survey has appeared in [32]; constrained optimization problems are studied in [33]; recent works [34]–[37] have modified these methods to achieve fixed-time convergence. With respect to this body of literature, which assumes precise knowledge of the temporal variability of the problem (see detailed discussion in Section II-A), our focus here is to answer the following fundamental question: what is the least-restrictive information that is needed to track, asymptotically with zero error, the optimizer of a time-varying problem? To the best of the authors' knowledge, a rigorous answer to this question is missing in the literature.

In this work, we recast the problem of designing optimization algorithms for time-varying optimization as a nonlinear output regulation problem. This connects our work with the literature on output regulation, which is a well-established area of research. Initial works can be traced back to the 1970s [16], [18], [38] focusing on linear systems, and later extended to nonlinear systems using both local [39], [40] and global [41] approaches for the analysis. In recent years, the field has received new attention using modern methods, see, e.g., [42]–[44] and the recent tutorial [45]. To the best of our knowledge, this is the first work in the literature that establishes a connection between output regulation and optimization algorithms design.

Contributions. This paper features four main contributions. First, we recast the problem of designing a time-varying optimization algorithm as a nonlinear multivariable regulation problem, whereby the signal to be regulated is the gradient of the loss function. We leverage this formulation to derive necessary and sufficient conditions to achieve exact asymptotic tracking for a large class of optimization methods (described by sufficiently-smooth functions). In a net departure from existing approaches (e.g., [11]–[13], [24]), our characterization is general and allows us to study not only a single optimization method, but an entire class, which enables us to derive fundamental results for all algorithms in this class. Second, by harnessing tools from center manifold theory [17], [19], we provide necessary and sufficient conditions for an optimization algorithm to ensure tracking. Interestingly, these conditions depend on the properties of the loss function (through a gradient invertibility-type condition) and on the inner model describing the temporal variability of the problem. This property allows us to prove the *internal model principle of time-varying optimization,* which states that for an optimization algorithm to achieve asymptotic tracking, it must incorporate a reduplicated model of the temporal variability of the problem. This feature is implicit in all existing approaches for time-varying optimization [10], [11], [34], [36], [37] but, to the best of the authors' knowledge, lacked a rigorous understanding until now. Third, we derive necessary and sufficient conditions for the existence of a tracking algorithm. Fourth, we use our characterizations to design algorithms for time-varying optimization. With respect to the existing literature, our algorithm does not require one to know or measure exactly the temporal variability of the problem and thus uses less stringent assumptions. Finally, we illustrate the applicability of the approach numerically on both synthetic problems and the traffic assignment problem in transportation.

Organization. Section II presents the problem of interest, Section III the parameter-feedback problem (where the temporal variability of the problem can be explicitly measured), and Section IV the gradient-feedback problem (where the algorithm has access only to first-order functional evaluations). Section V discusses the tracking accuracy in relationship to the fidelity of the internal model, Section VI presents extensions to constrained optimization and discrete-time algorithms, Section VII validates numerically the results, and Section VIII illustrates our conclusions. Finally, in the Appendix, we summarize basic facts on center manifold theory used in the paper.

Notation. We denote by $\mathbb{S}^n$ the space of $n \times n$ symmetric real matrices. Given an open set U, we say that $f : U \to \mathbb{R}$ is of differentiability class C^k if it has a k^{th} derivative that is continuous in U. The gradient of $f(x,\theta) : \mathbb{R}^n \times \Theta \to \mathbb{R}, \Theta \subseteq \mathbb{R}^p$, with respect to $x \in \mathbb{R}^n$ is denoted by $\nabla_x f(x,\theta) : \mathbb{R}^n \times \Theta \to \mathbb{R}^n$. The partial derivatives of $\nabla_x f(x,\theta)$ with respect to x and θ are denoted by $\nabla_{xx} f(x,\theta) : \mathbb{R}^n \times \Theta \to \mathbb{S}^n$ and $\nabla_{x\theta} f(x,\theta) : \mathbb{R}^n \times \Theta \to \mathbb{R}^{n\times p}$, respectively.

II. Problem Setting

A. Optimization objectives

We consider the time-varying optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x,\theta(t)), \tag{1}$$

where $t \in \mathbb{R}_{\geq 0}$ denotes time and $f : \mathbb{R}^n \times \Theta \to \mathbb{R}$, $\Theta \subseteq \mathbb{R}^p$, is a loss function that is parametrized by the time-varying parameter vector $\theta : \mathbb{R}_{\geq 0} \to \Theta$. We make the following standard assumptions on the loss.

Assumption 1 (***Properties of the objective function***)**.** The map $x \mapsto f(x,\theta)$ is convex and $x \mapsto \nabla_x f(x,\theta)$ is Lipschitz continuous in $\mathbb{R}^n$, for each $\theta \in \Theta$. □

Convexity and smoothness are standard assumptions in optimization [26], which have been widely used in works on related problems [8], [10]–[15], [24], [28], [29], [31]–[33].

In (1), the parameter $\theta(t)$ is used to model the temporal variability of the problem. We will require that $\theta(t)$ belongs to a certain class of temporal variabilities, as specified next.

Assumption 2 (***Class of temporal variabilities***)**.** There exists a smooth (i.e., C^∞) vector field $s : \Theta \to \mathbb{R}^p$ and $\theta(0) \in \Theta$ such that the parameter vector $\theta(t)$ satisfies

$$\dot{\theta}(t) = s(\theta(t)), \tag{2}$$

for $t \in \mathbb{R}_{\geq 0}$. Moreover, $\theta = 0$ is an equilibrium of (2) and the trajectories of (2) are bounded. □

We stress that, a priori, we do not assume that $s(\theta)$ nor $\theta(0)$ are known (see Problem 0 shortly below for additional details). Assumption 2 characterizes the class of temporal variabilities taken into consideration. This assumption is mild, as it only requires that $\theta(t)$ is deterministic, sufficiently smooth (so that its derivative is some C^∞ function of $\theta(t)$ as in (2)), and its trajectories remain bounded. In line with [38], [39], we call the autonomous system (2) the *exosystem*.

For simplicity of the presentation, we assume that Θ is some neighborhood of the origin of $\mathbb{R}^p$. We put no restrictions on the size of this neighborhood (which is, e.g., allowed to be the entire space $\Theta = \mathbb{R}^p$), and thus on the size of $\theta(t)$ nor on its temporal variation. Moreover, there is no restriction with asking that Θ contains the origin because, if $\theta(t)$ takes values in the neighborhood of any other point, such a point can be shifted to the origin through a time-invariant change of variables without altering the solutions of (1).

Remark 1 (***Parametrization in time-varying optimization***)**.** In (1), the time variation is captured implicitly by the parameter $\theta(t)$. A related, yet slightly more general, problem is:

$$\min_{x\in\mathbb{R}^n} f_0(x,t), \tag{3}$$

where the dependency on time is explicit. Problem (1) can be recast uniquely as in (3) by letting $f_0(x,t) = f(x,\theta(t))$ for all x and t. On the other hand, in general, there exists an infinite number of ways to parametrize (3) as in (1), thus leading to possible ambiguities. For instance, any $f_0(x,t)$ may be parametrized by $\theta(t) = t$ (so that $f_0 \equiv f$), although this is not compatible with our boundedness trajectory assumption (Assumption 2). □

Remark 2 (***Discrete-time implementations***)**.** In this work, we pursue the continuous-time time-varying optimization problem (1), together with a continuous-time optimization algorithm. This is inspired by control applications (see [14] and references therein), where (1) models performance objectives associated with a physical plant to be controlled, and $\theta(t)$ models exogenous disturbances affecting the plant. If instead the parameter vector were to evolve in discrete time, we would have the alternative optimization problem

$$\min_{x\in\mathbb{R}^n} f\left(x,\theta_k\right), \tag{4}$$

where $k \in \mathbb{N}_{\geq 0}$ denotes time or iteration, and $f : \mathbb{R}^n \times \Theta \to \mathbb{R}$. While we could discretize $\theta(t)$ and then study discrete-time algorithms to solve this problem, physical control applications motivate studying the continuous-time problem directly. We provide additional details on discrete-time implementations of our method in Section VI-B. □

In what follows, we say that $x^\star(t)$, with $x^\star : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, is a *critical trajectory* of (1) if it satisfies:

$$0 = \nabla_x f(x^\star(t),\theta(t)), \quad \forall t \in \mathbb{R}_{\geq 0}. \tag{5}$$

We assume the existence of a critical trajectory and that any critical trajectory is continuous. The existence of a critical trajectory can be guaranteed under standard assumptions on the optimization problem; for example, coercivity of the cost [46] (i.e., $f(x,\theta) \to \infty$ when $\|x\| \to \infty$), or (by Weierstrass' theorem [46]) when the search domain can be restricted to a compact set without altering the optimizers. Continuity of the critical trajectories can also be ensured under standard assumptions: for example, by Berge's theorem [46], by requiring that $f(x,\theta)$ is continuous in θ.

In our analysis, we will use a *critical point with θ at rest*, which is a constant vector $x^\star_\circ \in \mathbb{R}^n$, defined implicitly as:

$$0 = \nabla_x f(x^\star_\circ, 0). \tag{6}$$

We assume that $x^\star_\circ$ exists and is locally unique[1].

Remark 3 (***Critical trajectory vs. critical point at rest***)**.** We stress that $x^\star(t)$ and $x^\star_\circ$ are distinct quantities. Unlike the critical trajectory (defined in (5)), which denotes the optimal input to be tracked by the optimization algorithm, the critical point with θ at rest (defined in (6)) is a constant vector that need not be tracked and may never coincide with $x^\star(t)$. □

We conclude this discussion by reviewing two important classes of optimization methods commonly used to solve (1).

Remark 4 (***Basic gradient flow algorithms***)**.** The basic gradient flow algorithm [26], [48] for (1) reads as:

$$\dot{x}(t) = -\eta\nabla_x f(x(t),\theta(t)), \tag{7}$$

where $\eta > 0$ models the algorithm's step size. It is known [48] that this class of algorithms is capable of computing (asymptotically) critical points of (1) only when $\theta(t)$ is a constant signal; in all other cases, the algorithm converges to a neighborhood of a critical point and convergence is inexact. We refer the reader to Remark 11 for an insightful interpretation of this fundamental limitation. □

Remark 5 (***Prediction-correction algorithms***)**.** An established family of methods to solve (1) is that of prediction-correction

[1]Existence and local uniqueness of $x^\star_\circ$ can be guaranteed, for instance, under the assumptions of the Implicit Function Theorem [47]. Namely, let $X_\circ \times \Theta_\circ$ be some neighborhood of $(x^\star_\circ, 0)$, $x^\star_\circ$ exists and is locally unique when: (i) f is C^1 on $X_\circ \times \Theta_\circ$, (ii) $x \mapsto f(x,\theta)$ is C^2 on $X_\circ$ for each $\theta \in \Theta_\circ$, and (iii) the $\nabla^2_{xx} f(x,\theta)\big|_{x=x^\star_\circ,\theta=0}$ is locally positive definite.

algorithms [10], [11], [34], [36], [37]. Tailored to our setting, the basic method of this class reads as:

$$\dot{x}(t) = -\nabla_{xx}^{-1} f(x(t),\theta(t))[\eta \nabla_x f(x(t),\theta(t)) + \nabla_{x\theta} f(x(t),\theta(t)) \cdot s(\theta(t))], \quad (8)$$

where $\nabla_{xx} f(x,\theta)$ is the Hessian of $f(x,\theta)$, and $\nabla_{x\theta} f(x,\theta)$ is the partial derivative of $\nabla_x f(x,\theta)$ with respect to θ. Under strong convexity assumptions, this algorithm is known to converge exponentially, with zero asymptotic error, to a critical trajectory [10], [11]. Variations of this algorithm have also been proposed to achieve fixed-time or finite-time convergence [34]. We conclude by noting that implementing (8) requires (i) the cost to be twice continuously differentiable and strongly convex, (ii) knowledge of the maps $s(\theta)$, $\nabla_x f(x,\theta)$, $\nabla_{xx} f(x,\theta)$, and $\nabla_{x\theta} f(x,\theta)$, and (iii) knowledge of $\theta(t)$. □

B. Problem statement

In line with the literature [10]–[12], [26], [34], we focus on gradient-type algorithms for solving (1), which are algorithms that have access to first-order oracles of the cost; that is, functional evaluations of the map:

$$(t,x) \mapsto \nabla_x f(x,\theta(t)). \quad (9)$$

Remark 6 (***Evaluating the gradient***)**.** In the optimization literature [26], functional evaluations of (9) are generally obtained through two main approaches: (i) the algorithm has access to both the analytical form of $\nabla_x f(x,\theta)$ and $\theta(t)$ at each time t (either because this signal is known or measurable), or (ii) an oracle returns evaluations of (9) (computed, e.g., through numerical differentiation). These alternatives will be discussed more in detail in (O1)–(O4), shortly below. □

We let the optimization algorithm be described by a dynamic internal state $z(t)$, taking values on an open subset $\mathcal{Z} \subseteq \mathbb{R}^{n_c}, n_c \in \mathbb{N}_{>0}$. The optimization algorithm generates a sequence of points $x(t) \in \mathbb{R}^n$ (called *exploration signal*), and has access to functional evaluations of (9) at these points (called *gradient feedback signal*):

$$y(t) = \nabla_x f(x(t),\theta(t)). \quad (10a)$$

Together with this gradient feedback signal, the optimization algorithm is described by[2]

$$\dot{z}(t) = F_c(z(t),y(t)), \qquad x(t) = G_c(z(t)), \quad (10b)$$

where $F_c : \mathcal{Z} \times \mathbb{R}^n \to \mathbb{R}^{n_c}$ and $G_c : \mathcal{Z} \to \mathbb{R}^n$ are functions to be designed. In the remainder, we refer to (10) as a *dynamic gradient-feedback optimization algorithm*. The architecture of the presented gradient-feedback algorithm is illustrated in Fig. 1. For simplicity of the presentation, we will require that $F_c(z,y)$ and $G_c(z)$ are such that[3]:

$$0 = F_c(z_\circ^\star, 0) \qquad \text{and} \qquad x_\circ^\star = G_c(z_\circ^\star), \quad (11)$$

for some locally unique $z_\circ^\star \in \mathbb{R}^{n_c}$. Together with (6), this ensures that $z_\circ^\star$ is an equilibrium of (10) when the gradient feedback signal $y(t)$ is equal to zero.

The dynamics of algorithm (10), coupled with the exosystem (2), have the form of a nonlinear autonomous system:

$$\dot{z}(t) = F_c(z(t),y(t)), \quad (12a)$$
$$y(t) = \nabla_x f(G_c(z(t)),\theta(t)), \quad (12b)$$
$$\dot{\theta}(t) = s(\theta(t)). \quad (12c)$$

Definition 1 (***Exact asymptotic tracking***)**.** We say that (10) *exactly asymptotically tracks a critical trajectory* of (1) if there exists a neighborhood of $(z_\circ^*, 0)$ in $\mathcal{Z} \times \Theta$ such that, for each initial condition $(z(0),\theta(0))$ in the neighborhood, the solution of (12) satisfies $y(t) \to 0$ as $t \to \infty$. □

By considering the entire class of optimization algorithms as in (10), and without requiring any assumption in addition to (9), we begin by posing the following fundamental question.

Problem 0 (***Minimal knowledge for exact asymptotic tracking***)**.** Consider the class of optimization algorithms (10). Determine the minimal necessary knowledge (beyond (9)), concerning the exosystem (2) and the optimization (1), needed to design an algorithm from this class that tracks, with zero asymptotic error, the critical trajectories of (1). □

Before addressing this question, we first outline various pieces of information on the exosystem and objective that the algorithm may (or may not) be able to access. Two (non mutually-exclusive[4]) assumptions on the exosystem from the literature (see, e.g., [10]–[12], [26], [34], [48]) are:

(E1) The parameter signal $\theta(t)$ is known or measurable at each $t \in \mathbb{R}_{\geq 0}$, but no knowledge of $s(\theta)$ is available.
(E2) The vector field $s(\theta)$ is known, but $\theta(0)$ (and hence also $\theta(t)$) is unknown.

A second distinction can be made in terms of what knowledge concerning the objective function is available. In this case, several (non mutually-exclusive) options can be found in the literature (see, e.g., [10]–[12], [26], [34], [48]):

(O1) The algorithm has access to oracle evaluations of (9).
(O2) The analytical form of the gradient $\nabla_x f(x,\theta)$ is known.
(O3) The sensitivity $\nabla_{x\theta} f(x,\theta)$ of the cost with respect to the parameter is known.
(O4) The Hessian $\nabla_{xx}^2 f(x,\theta)$ of the cost is known.
(O5) The objective $f(x,\theta)$ is quadratic in both x and θ.

These are not assumptions we impose a priori in this work, but rather widely used requirements in the literature [10]–[12], [26], [34], [48]. Various methods have been developed in the literature based on different combinations of these assumptions: for example, under (E1) and (O1), the basic gradient flow algorithm (7) converges (asymptotically) to a neighborhood of the critical points (see Remark 4). Under assumptions (E1), (E2), (O2), (O3), and (O4), the prediction-correction method (8) converges (asymptotically or in fixed-time) exactly to the optimizers (see Remark 5). Table I pro-

[2]While one could allow G_c to also depend on y, so that $x(t) = G_c(z(t),y(t))$, we will show in Section IV that this is unnecessary.

[3]Notice that this is without loss of generality since $F_c(z,y)$ and $G_c(z)$ are to be designed and $x_\circ^\star$ is known through (6).

[4]Notice that requiring both (E1) and (E2) is equivalent to asking that both $s(\theta)$ as well as $\theta(0)$ are known.

TABLE I: Summary of required knowledge and known guarantees by various established approaches. The symbol (✓) is used when (O1) follows from (E1) and (O2). See Remarks 4, 5, and 7 for discussions.

	Approximate solutions	(E1) $\theta(t)$	(E2) $s(\theta)$	(O1) $(t,x)\mapsto\nabla_x f(x,\theta(t))$	(O2) $\nabla_x f(x,\theta)$	(O3) $\nabla_{x\theta} f(x,\theta)$	(O4) $\nabla^2_{xx} f(x,\theta)$	(O5) f quadratic
Basic gradient flow [26], [48]	✓			✓				
Control-based methods [12]			✓	✓	✓	✓		✓
Prediction-correction [10], [11], [34]		✓	✓	(✓)	✓	✓	✓	
This work (Section III)		✓		(✓)	✓			
This work (Section IV)			✓	✓	✓			

vides a comparison of the assumptions required by established methods and their positioning with respect to this work.

Remark 7 (***Generality of the algorithm class***)**.** The gradient flow algorithm (7) can be viewed as a special case of (10) with $F_c(z,y) = -\eta y$ and $G_c(z,y) = z$. Similarly, the prediction-correction method (8) can be viewed as a special case of (10) with $z = (z_1, z_2)$ and

$$\begin{aligned} F_c(z,y) &= \begin{bmatrix} -\nabla_{xx}^{-1} f(z_1,z_2)[y + \nabla_{x\theta} f(z_1,z_2)\cdot s(z_2)] \\ s(z_2) \end{bmatrix}, \\ G_c(z) &= z_1, \end{aligned} \tag{13}$$

with $z_2(0) = \theta(0)$. It follows that (10) is general enough to encompass these two established methods as special cases, and thus all conclusions concerning the class of algorithms (10) made here will also apply to these methods. □

Our second objective is to design optimization algorithms from the class (10) that track, with zero asymptotic error, the critical trajectories of (1). We formalize this notion next.

Problem 1 (***Design of dynamic gradient-feedback optimization algorithms***)**.** Design $F_c(z,y)$, $G_c(z)$, and n_c, requiring the minimal necessary knowledge for exact asymptotic tracking (as in Problem 0), so that (10) exactly asymptotically tracks a critical trajectory of (1). □

III. The parameter-feedback problem

In many cases of interest [26], having access to $y(t)$ is a byproduct of having access to both the function $\nabla_x f(x,\theta)$ and knowledge or measurements of $\theta(t)$; see Remark 6. In these cases, assumptions (E1) and (O2) are automatically satisfied. In this section, we analyze this scenario. We anticipate that the results derived in this section will also serve as an intermediate step to tackle the more challenging problem where assumption (E1) is relaxed, which is the focus of Section IV.

When the algorithm has access to both $\theta(t)$ at each time $t \in \mathbb{R}_{\geq 0}$, and the function $\nabla_x f(x,\theta)$, the measurements $y(t)$ do not provide additional information. Hence, under the assumptions of this section, the optimization algorithm (10) can be replaced by the algebraic relationship[5]

$$x(t) = H_c(\theta(t)), \tag{14}$$

where $H_c : \Theta \to \mathbb{R}^n$ is a mapping to be designed. Because of the explicit dependence on $\theta(t)$, we will refer to (14) to as a *parameter-feedback* optimization algorithm. In analogy with (11), for simplicity of the presentation, in this section, we will impose that $H_c(\theta)$ satisfies:

$$x_\circ^\star = H_c(0),$$

and require that H_c is of class C^0.

The composition of (2), (10a), and (14) is given by:

$$y(t) = \nabla_x f(H_c(\theta(t)), \theta(t)), \tag{15a}$$
$$\dot\theta(t) = s(\theta(t)). \tag{15b}$$

In analogy with Definition 1, we will say that the parameter-feedback algorithm (14) *exactly asymptotically tracks a critical trajectory* of (1) if there exists a neighborhood $\Theta_s \subset \Theta$ of the origin such that, for each initial condition $\theta(0) \in \Theta_s$, the solution of (15) satisfies $y(t) \to 0$ as $t \to \infty$.

For the algorithm (14) considered in this section, Problems 0 and 1 are reformulated as follows.

Problem 2 (***Minimal knowledge for parameter feedback***)**.** Consider the class of optimization algorithms (14). Determine the minimal necessary knowledge concerning (1) and (2), needed to design an algorithm from this class that exactly asymptotically tracks a critical trajectory of (1). □

Problem 3 (***Design of parameter-feedback optimization algorithms***)**.** Design $H_c(\theta)$, requiring minimal knowledge for parameter-feedback (as in Problem 2), so that (14) exactly asymptotically tracks a critical trajectory of (1). □

A. Fundamental results

Solvability of the parameter-feedback problem will depend on the existence of a function that zeros the gradient on the set of limit points of the exosystem; we now define these notions.

Definition 2 (***Mapping zeroing the gradient***)**.** We say that a mapping $H_c : \Theta \to \mathbb{R}^n$ *zeros the gradient* at the point $\theta \in \Theta$ if

$$0 = \nabla_x f(H_c(\theta), \theta). \tag{16}$$

Moreover, we say that H_c zeros the gradient *on a set* $\Theta_\circ \subseteq \Theta$ if (16) holds for all $\theta \in \Theta_\circ$. □

Definition 3 (***Limit point and limit set***)**.** A point $\theta_\omega \in \Theta$ is a *limit point with respect to the initialization* $\theta_\circ \in \Theta$ if there exists a sequence $\{t_i\}_{i\in\mathbb{N}_{\geq 0}}$ with $t_i \to \infty$ as $i \to \infty$ such that the exosystem (2) with $\theta(0) = \theta_\circ$ satisfies $\theta(t_i) \to \theta_\omega$ as $i \to \infty$. Let $\Omega(\theta_\circ)$ denote the set of all limit points (i.e., for all sequences t_i) of (2) with respect to the initialization $\theta_\circ \in \Theta$. Given $\Theta_\circ \subseteq \Theta$, the set $\Omega(\Theta_\circ) := \cup_{\theta_\circ\in\Theta_\circ}\Omega(\theta_\circ)$ is called the *limit set with respect to initializations in* $\Theta_\circ$ [49]. □

Intuitively, $\Omega(\Theta_\circ)$ denotes the set of all limit points (equilibria, limit cycles, etc.) that can be reached by the exosystem

[5] While one could consider a dynamic optimization algorithm of the form $\dot z(t) = F_c(z(t), \theta(t))$ and $x(t) = G_c(z(t))$, we will prove in Theorem 1 that a dynamic structure is unnecessary.

when initialized at points in $\Theta_\circ$. By the boundedness trajectories assumption (Assumption 2), $\Omega(\Theta_\circ)$ is a bounded set.

The following result characterizes all parameter-feedback optimization algorithms that achieve asymptotic tracking.

Theorem 1 (***Existence and characterization of parameter-feedback algorithms***)**.** Let Assumptions 1–2 hold. The parameter-feedback algorithm (14) asymptotically tracks a critical trajectory of (1) if and only if there exists a neighborhood $\Theta_\circ \subset \Theta$ of the origin such that the mapping H_c zeros the gradient on the limit set $\Omega(\Theta_\circ)$.

Proof. (Only if) Suppose $y(t) \to 0$ as $t \to \infty$ for initializations $\theta(0) \in \Theta_\circ$; we will show that H_c zeros the gradient on $\Omega(\Theta_\circ)$. By Assumption 2, the trajectories of the exosystem are bounded, and thus, by the Bolzano–Weierstrass theorem, there exists an increasing subsequence $\{t_i\}_{i\in\mathbb{N}_{\geq 0}}$ such that $\theta(t_i)$ converges to some limit point $\theta_\omega \in \Omega(\theta(0))$. We then have:

$$\lim_{i\to\infty} y(t_i) = \lim_{i\to\infty} \nabla_x f(H_c(\theta(t_i)), \theta(t_i)) = \nabla_x f(H_c(\theta_\omega), \theta_\omega), \tag{17}$$

where the second identity follows by the continuity of the gradient (see Assumption 1) and that of H_c. Because, $y(t) \to 0$, the left-hand side of (17) is equal to zero, and thus (17) implies that $\nabla_x f(H_c(\theta_\omega), \theta_\omega) = 0$. Since this holds for any limit point $\theta_\omega \in \Omega(\Theta_\circ)$, the statement follows.

(If) Suppose $\theta(0) \in \Theta_\circ$ and that H_c zeros the gradient on $\Omega(\Theta_\circ)$. Then, the right-hand side of (17) is equal to zero, which implies the existence of a sequence t_i such that $y(t_i) \to 0$ as $i \to \infty$. Since this holds for any limit point $\theta_\omega \in \Omega(\theta(0))$, any convergent subsequence of $y(t)$ converges to zero. Moreover, $y(t)$ is bounded due to the Lipschitz continuity of the gradient, so $y(t)$ also converges to zero as $t \to \infty$. By iterating the reasoning for all $\theta(0) \in \Theta_\circ$, it follows that $y(t) \to 0$ for all $\theta(0) \in \Theta_\circ$, and the claim follows. □

The theorem shows that the existence of a parameter-feedback algorithm is dependent upon the existence of a mapping that zeros the gradient. An application of this theorem is illustrated in Example 1, which shows that a parameter-feedback algorithm may fail to exist in some circumstances. The theorem also provides a full characterization of all parameter-feedback algorithms that achieve asymptotic tracking. Loosely speaking, $x = H_c(\theta)$ is a parameter-feedback algorithm if and only if it zeros the gradient on the limit set of the exosystem. As a byproduct, the result also provides a necessary and sufficient condition for the solvability of the parameter-feedback problem: the problem is solvable if and only if the set of solutions to the system of equations $0 = \nabla_x f(x, \theta)$ can be expressed, at all limit points of the exosystem, as the graph of a function $x = H_c(\theta)$. We present a set of sufficient conditions for this to hold in Section III-B.

By Theorem 1, the problem of designing a parameter-feedback algorithm for exact asymptotic tracking can be reduced to that of finding a mapping $x = H_c(\theta)$ that zeros the gradient. Therefore, we have the following.

Answer to Problem 2. When a parameter-feedback algorithm that achieves exact tracking exists, the minimal knowledge needed to design such an algorithm is (O2). Notice that assumption (E1) is implicitly required to execute the algorithm (14).

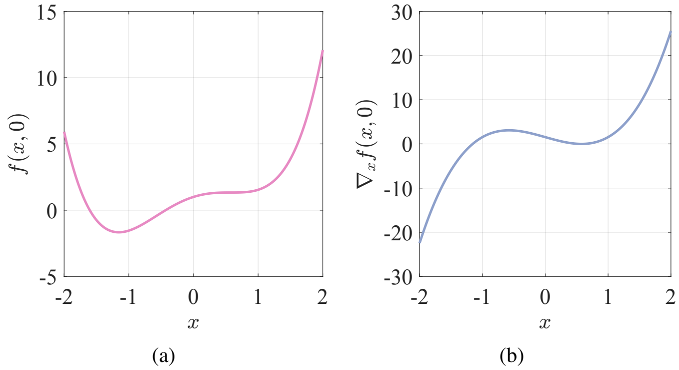


Fig. 2: Investigation of the condition (16). (Left) Loss function $f(x,\theta)$ studied in Example 1, plotted for $\theta = 0$. (Right) Gradient $\nabla f(x,\theta)$. The function $f(x,0)$ admits two critical points: $x^\star_{\circ,1} = \frac{1}{\sqrt{3}}$ and $x^\star_{\circ,2} = -\frac{2}{\sqrt{3}}$. At $x^\star_{\circ,1}$, condition (16) is not satisfied, since with an upward shift of the graph of $\nabla_x f(x,0)$, $x^\star_{\circ,1}$ is no longer a critical point of $f(x,0)$. On the other hand, (16) holds for $x^\star_{\circ,2}$, since $x^\star_{\circ,2}$ varies continuously as θ is perturbed. See Example 1 for a discussion.

We illustrate the applicability of the result and the necessity of the provided condition in the following example.

Example 1. Consider an instance of (1) with $n = p = 1$,

$$f(x, \theta) = (x-1)^2(x+1)^2 + \frac{8}{3\sqrt{3}}x + \theta x. \tag{18}$$

See Fig. 2(a) for an illustration of this function. For $\theta = 0$, the optimization problem associated with (18) admits two critical points: $x^\star_{\circ,1} = \frac{1}{\sqrt{3}}$ and $x^\star_{\circ,2} = -\frac{2}{\sqrt{3}}$; indeed, it follows from direct inspection that $\nabla_x f(x^\star_{\circ,1}, 0) = 0$ and $\nabla_x f(x^\star_{\circ,2}, 0) = 0$. At the critical point $x^\star_{\circ,1}$, a function $H_c(\theta)$ as in (16) does not exist. This can be visualized with the aid of Fig. 2(b): if $\theta = 0$ is perturbed to $\theta + \epsilon$, $\epsilon > 0$, the graph of $\nabla_x f(x, 0)$ (illustrated in Fig. 2(b)) shifts upward and the equation $\nabla_x f(x, \epsilon) = 0$ no longer admits a solution in a neighborhood of $x^\star_{\circ,1}$.

On the other hand, for the critical point $x^\star_{\circ,2}$, any arbitrarily small upward or downward shift of the graph of $\nabla_x f(x, 0)$ results in a continuous perturbation of $x^\star_{\circ,2}$ (see Fig. 2(b)), thus suggesting existence of $x = H_c(\theta)$ as in (16). This graphical observation can be formalized with the aid of the implicit function theorem [47], as described next. Define $F(x, \theta) := \nabla_x f(x, \theta)$ and notice that F is continuously differentiable with $F(x^\star_{\circ,2}, 0) = 0$. By the implicit function theorem, there exists a neighborhood $\Theta_\circ$ of $x^\star_{\circ,2}$ and a function $H_c : \Theta_\circ \to \mathbb{R}^n$ such that $F(H_c(\theta), \theta) = 0$ in $\Theta_\circ$, provided that $\frac{\partial F}{\partial x}\big|_{(x,\theta)=(x^\star_{\circ,2},0)} \neq 0$. By inspection, it is immediate to see that the latter condition is satisfied. □

We conclude by observing that the proof of Theorem 1 is constructive in that it provides an explicit procedure to construct parameter-feedback algorithms.

Answer to Problem 3. When a parameter-feedback algorithm exists, a procedure to design it requiring minimal knowledge (O2) is to first determine the limit set $\Omega(\Theta_\circ)$ for the given set of initial conditions $\Theta_\circ$ and then find a mapping $H_c(\theta)$ that zeros the gradient on the limit set, in which case the parameter-feedback algorithm is given by (14).

We conclude by illustrating the parameter-feedback design procedure on a quadratic problem.

Example 2. Consider an instance of (1) with quadratic cost and time-variability that depends linearly on $\theta(t)$ (which, e.g., has been investigated in [12]):

$$f(x(t),\theta(t)) = \tfrac{1}{2}x(t)^{\mathsf{T}}Rx(t) + x(t)^{\mathsf{T}}Q\theta(t), \tag{19}$$

with matrices $R \in \mathbb{S}^n$ and $Q \in \mathbb{R}^{n\times p}$. This loss is convex and Lipschitz smooth and thus satisfies Assumption 1. In this case, the signal we wish to regulate to zero is:

$$y(t) = \nabla_x f(x(t),\theta(t)) = Rx(t) + Q\theta(t).$$

For arbitrary θ, this problem admits a critical point if and only if $\operatorname{Im} Q \subseteq \operatorname{Im} R$, in which case $x_\circ^\star$ is unique. Applying Theorem 1 amounts to finding a linear transformation $H_c \in \mathbb{R}^{n\times p}$ such that $0 = (RH_c + Q)\theta$ for all θ in a neighborhood of the origin. Assuming $\operatorname{Im} Q \subseteq \operatorname{Im} R$, we can choose $H_c = -R^\dagger Q$, where $R^\dagger$ is the pseudo-inverse of R. Note that, by substituting into (15), we have $y(t) = RH_c\theta(t) + Q\theta(t) = 0$ for all times $t \in \mathbb{R}_{\geq 0}$. Namely, the gradient is identically zero. This property holds because the mapping $H_c(\theta)$ derived here zeros the gradient globally (and not only in some neighborhood of the origin). In fact, this feature holds for the any optimization problem for which the gradient of the cost admits a mapping that zeros the gradient globally. A set of sufficient conditions to check for this property are given in Theorem 3. □

Remark 8 (***Prediction-correction algorithms asymptotically compute maps that zero the gradient***)**.** As an illustration, consider the quadratic problem studied in Example 2. Let $x_{\mathrm{PC}}(t)$ denote the iterates of the prediction-correction algorithm (8) applied to the quadratic objective (19), and let $x_{\mathrm{PF}}(t) = H_c\theta(t)$ with $H_c = -R^\dagger Q$ be the parameter-feedback algorithm. Then, the error $e(t) = x_{\mathrm{PC}}(t) - x_{\mathrm{PF}}(t)$ satisfies the dynamics $\dot{e}(t) = -e(t)$, indicating that the iterates of prediction-correction converge exponentially to those of the parameter-feedback algorithm, which zeros the gradient. □

B. Sufficient conditions for the existence of a parameter-feedback algorithm

Existence of a map $x = H_c(\theta)$ as in (16) can be ensured for general problems with the aid of the implicit function theorem [47], as illustrated by the following result.

Proposition 2 (***Existence of local parameter-feedback algorithms***)**.** Let Assumptions 1–2 hold, and let $X_\circ \times \Theta_\circ$ be some neighborhood of $(x_\circ^\star, 0)$. Further, assume that:

(i) the loss function f is C^1 on $X_\circ \times \Theta_\circ$,
(ii) $x \mapsto f(x,\theta)$ is C^2 on $X_\circ$ for each $\theta \in \Theta_\circ$, and
(iii) the Hessian $\nabla^2_{xx} f(x,\theta)\big|_{x=x_\circ^\star,\theta=0}$ is positive definite.

Then, there exists a C^0 mapping $H_c : \Theta \to \mathbb{R}^n$ that zeros the gradient on some neighborhood of the origin of Θ. □

Proof. Define $F(x,\theta) = \nabla_x f(x,\theta)$ and note that, under the stated assumptions, F is C^0 on $X_\circ \times \Theta_\circ$, the mapping $x \mapsto F(x,\theta)$ is C^1 on $X_\circ$ for all $\theta \in \Theta_\circ$, and

$$\det\left[\frac{\partial F(x,\theta)}{\partial x}\right]_{(x,\theta)=(x_\circ^\star,0)} \neq 0.$$

Then, the result follows from the implicit function theorem (see, e.g., [50, Thm. 1]) applied to the first-order optimality conditions $0 = F(x,\theta)$ at the point $(x,\theta) = (x_\circ^\star, 0)$. □

In other words, Proposition 2 states that when the loss function is sufficiently smooth and the Hessian is locally positive definite, the existence of $H_c(\theta(t))$ is guaranteed by the implicit function theorem. Notice that these conditions are sufficient but not necessary.

Remark 9 (***Computing mappings that zero the gradient***)**.** Under the assumptions of Proposition 2, the implicit function theorem gives that the linear function

$$\hat{H}_c(\theta) = x_\circ^\star - \left(\nabla^2_{xx} f(x,\theta)\big|_{x=x_\circ^\star \theta=0}\right)^{-1} \nabla_{x\theta} f(x,\theta)\big|_{x=x_\circ^\star,\theta=0}\theta, \tag{20}$$

is a first-order approximation of a mapping that zeros the gradient on a neighborhood of the origin. □

Although the conditions in Proposition 2 are immediate to verify, the convergence claims of Theorem 1 and Proposition 2 are of local nature, namely, $y(t) \to 0$ is ensured provided that $\theta(0)$ is sufficiently close to the origin. The following result provides a sufficient condition for global convergence.

Theorem 3 (***Existence of global parameter-feedback algorithms***)**.** Let Assumptions 1–2 hold. Further, assume:

(i) the loss function f is C^1 on $\mathbb{R}^n \times \Theta$,
(ii) $x \mapsto f(x,\theta)$ is C^3 on $\mathbb{R}^n$ for each $\theta \in \Theta$,
(iii) the Hessian $\nabla^2_{xx} f(x,\theta)$ is positive definite on $\mathbb{R}^n \times \Theta$,
(iv) the mappings $\nabla^2_{xx} f$ and $\frac{\partial \nabla^2_{xx} f}{\partial x}$ are C^0 on $\mathbb{R}^n \times \Theta$.

Then, there exits a C^0 mapping $H_c : \Theta \to \mathbb{R}^n$ that zeros the gradient everywhere in Θ. □

Proof. Define $F(x,\theta) = \nabla_x f(x,\theta)$ and note that, under the stated assumptions, F is C^0 on $\mathbb{R}^n \times \Theta$, the mapping $x \mapsto F(x,\theta)$ is C^2 on $\mathbb{R}^n$ for all $\theta \in \Theta$,

$$\det\left[\frac{\partial F(x,\theta)}{\partial x}\right] \neq 0, \quad \forall (x,\theta) \in \mathbb{R}^n \times \Theta,$$

and the mappings $\frac{\partial F}{\partial x}$ and $\frac{\partial^2 F}{\partial x^2}$ are C^0 in $\mathbb{R}^n \times \Theta$. Hence, assumptions (B1)–(B5) of a global version of the implicit function theorem [50, Thm. 2] (see also [51, Thm. 6]) are satisfied for the first-order optimality conditions $0 = F(x,\theta)$, and the claim follows. □

Theorem 3 shows that, under additional continuity assumptions on the loss function and when the Hessian of the loss is positive definite everywhere, there exists a mapping $H_c : \Theta \to \mathbb{R}^n$ that zeros the gradient everywhere in Θ.

By combination of Theorem 3 and Theorem 1, it follows that under the assumptions of Theorem 3, the parameter-feedback algorithm $x(t) = H_c(\theta(t))$ ensures that $y(t) \to 0$ as $t \to \infty$ for all initial conditions $\theta(0) \in \Theta$. This result ensures the existence of parameter-feedback algorithms for the particular class of strictly convex loss functions [11], [12].

IV. The dynamic gradient-feedback problem

In the previous section, we analyzed algorithms assuming knowledge of the function $\nabla_x f(x,\theta)$ and of the signal $\theta(t), t \in \mathbb{R}_{\geq 0}$ (cf. assumptions (E1) and (O2)). In this section, we relax these assumptions and require only that $y(t)$ is available through oracle evaluations of (9). For this reason, we will shift our attention back to the general class of dynamic gradient-feedback algorithms (10).

Because $\theta(t)$ is unmeasurable by the algorithm, the temporal variability of the cost can only be evaluated through measurements of $y(t)$; for this reason, we impose the following.

Assumption 3 (***Detectability of the exosystem***)**.** Let

$$Q := \left[\frac{\partial \nabla_x f}{\partial \theta}\right]_{(x,\theta)=(x_\circ^\star,0)}, \qquad S := \left[\frac{\partial s}{\partial \theta}\right]_{\theta=0}. \tag{21}$$

The pair (Q,S) is detectable. □

The detectability property of Assumption 3 is equivalent to the existence of a local exponential observer for all (non-asymptotically stable) modes of $\theta(t)$ based on measurements $y(t)$ [52, Cor. 3.4]. Undetectability corresponds to a redundant description of the exogenous signal: if some modes of $\theta(t)$ are undetectable, these modes do not influence the gradient (and, as a consequence, the critical points of (1)), and thus they can be removed from the optimization problem (1) without altering its critical points. In other words, Assumption 3 ensures that (2) does not contain redundancies.

A. Existence and characterization of dynamic gradient feedback algorithms

We begin by considering the class of algorithms (10), restricted to ensure local exponential stability, and we characterize a set of necessary and sufficient conditions for this class to achieve exact asymptotic tracking.

Theorem 4 (***Gradient-feedback algorithm characterization***)**.** Suppose Assumptions 1–3 hold, and assume that $F_c(z,y)$ and $G_c(z)$ are such that the equilibrium $z = z_\circ^\star$ of

$$\dot z(t) = F_c(z(t), \nabla_x f(G_c(z(t)), 0)), \tag{22}$$

is locally exponentially stable. The optimization algorithm (12) exactly asymptotically tracks a critical trajectory of (1) if and only if there exists a neighborhood $\Theta_\circ \subset \Theta$ of the origin and a C^2 mapping $z = \sigma(\theta)$ with $\sigma(0) = z_\circ^\star$, such that:

$$\left.\frac{\partial \sigma(\theta)}{\partial \theta}\right|_{\theta=\theta_\omega} s(\theta_\omega) = F_c(\sigma(\theta_\omega), 0), \tag{23a}$$

$$0 = \nabla_x f(G_c(\sigma(\theta_\omega)), \theta_\omega), \tag{23b}$$

at all limit points $\theta_\omega \in \Omega(\Theta_\circ)$.

Proof. (Only if). We first prove that $\lim_{t\to\infty} y(t) = 0$ implies (23). The coupled dynamics (12) have the form:

$$\begin{aligned} \dot z &= (A_c + B_c R M) z + B_c Q \theta + \chi(z,\theta),\\ \dot \theta &= S\theta + \psi(\theta), \end{aligned} \tag{24}$$

for some mappings $\chi(x,\theta)$ and $\psi(\theta)$ that vanish at the origin along with their first-order derivatives, with Q and S are defined in (21), and

$$A_c = \left[\frac{\partial F_c}{\partial z}\right]_{(z,y)=(z_\circ^\star,0)}, \qquad B_c = \left[\frac{\partial F_c}{\partial y}\right]_{(z,y)=(z_\circ^\star,0)},$$
$$R = \left[\frac{\partial \nabla_x f}{\partial x}\right]_{(x,\theta)=(x_\circ^\star,0)}, \qquad M = \left[\frac{\partial G_c}{\partial z}\right]_{z=z_\circ^\star}.$$

By assumption, the eigenvalues of the matrix $A_c + B_c R M$ are in $\mathbb{C}^-$. By Theorem 9, the system (24) has a center manifold at $(z_\circ^\star, 0)$: the graph of a mapping $z = \sigma(\theta)$, with $\sigma(\theta)$ satisfying (see (41))

$$\sigma(s(\theta)) = F_c(\sigma(\theta), \nabla_x f(G_c(\sigma(\theta)), \theta)).$$

Next, similarly to the parameter-feedback case (Theorem 1), by Assumption 2, $\theta(t_i)$ converges to some limit point $\theta_\omega \in \Omega(\theta(0))$ for some subsequence $\{t_i\}_{i\in\mathbb{N}_{\geq 0}}$. By continuity of the gradient (Assumption 1) and that of G_c,

$$\lim_{i\to\infty} y(k_i) = \lim_{i\to\infty} \nabla_x f(G_c(\theta_{t_i}), \theta_{t_i}) = \nabla_x f(G_c(\theta_\omega), \theta_\omega). \tag{25}$$

Because, $y(t) \to 0$, the left-hand side of (25) is equal to zero, and thus $\nabla_x f(G_c(\theta_\omega), \theta_\omega) = 0$. Since this holds for any limit point $\theta_\omega \in \Omega(\theta_0)$, the claim follows.

(If). We now prove that (23) implies $\lim_{t\to\infty} y(t) = 0$. It follows from (23) and Theorem 10 that the graph of the mapping $z = \sigma(\theta)$ (i.e., $(\sigma(\theta), \theta)$) is a center manifold for the coupled dynamics (12). Moreover, such manifold is locally attractive; namely, $z(t) \to \sigma(\theta(t))$ as $t \to \infty$. Then, the fulfillment of (23b) guarantees that the right-hand side of (25) $y(t) \to 0$. The conclusion then follows by iterating the *(If)* part of the proof of Theorem 1. □

The two conditions in (23) fully characterize the class of optimization algorithms that achieve exact asymptotic tracking: (10) tracks a critical trajectory if and only if, for some mapping σ, the composite function $G_c \circ \sigma$ zeros the gradient locally (see (23b)), and the controller $F_c(z,y)$ is algebraically related to the exosystem $s(\theta)$ as given by (23a). Notice that, by Theorem 1, the former condition implies that

$$x(t) = G_c(\sigma(\theta)), \tag{26}$$

is a parameter-feedback optimization algorithm for (1). Finally, we note that the existence of an exponentially stable system as in (22) is always guaranteed under Assumption 3 [52, Sec. 3].

The following interpretation follows from our findings.

Remark 10 (***The internal model principle***)**.** We refer to condition (23a) as the *internal model principle of time-varying optimization*, as it encapsulates the requirement that an optimization algorithm must incorporate an internal model of the exosystem to achieve exact asymptotic tracking. Note that the use of a copy of the temporal variability of the optimization problem is explicit in the prediction-correction algorithm (see (13)). □

Theorem 4 allows us to re-interpret the basic gradient-flow algorithm as follows.

Remark 11 (***Internal model-based interpretation of basic gradient flow methods***)**.** Recall that the gradient-flow algorithm (4) can be viewed as an instance of (10a) with $F_c(z, y)$ and $G_c(z)$ as in Remark 7. By direct substitution into (23), it is immediate to see that this algorithm satisfies (23a) with $s(\theta) = 0$ and $\sigma(\theta)$ arbitrary. Since $s(\theta) = 0$ describes the internal model of a constant signal, it follows from Theorem 4 that these algorithms achieve exact asymptotic tracking if and only if $\theta(t)$ is a constant signal. This observation is in line with the inexact convergence properties of these algorithms typically given for these algorithms [14], [26], [48]. □

Since by, Theorem 4, the problem of exact asymptotic tracking can be reduced to that of finding an optimization algorithm that is algebraically equivalent to the exosystem and a mapping that zeros the gradient, we have the following.

Answer to Problem 0. When an algorithm as in (10) that achieves exact asymptotic tracking exists, the minimal knowledge needed to design such an algorithm is given by assumptions (E2) and (O2). Notice that assumption (O1) is implicitly required to execute the algorithm (10).

By comparison with *Answer to Problem 2*, it follows that relaxing the knowledge of $\theta(t)$ (i.e., relaxing (E1)) comes at the cost of requiring knowledge of the exosystem (i.e., requiring (E2)).

By Theorem 4, the exosystem state θ and that of the optimization z must be related, everywhere in $\Theta_\circ$, by

$$z(t) = \sigma(\theta(t)). \tag{27}$$

Intuitively, (27) is interpreted as the existence of a change of coordinates between the state of the exosystem and that of the optimization; see Section IV-C for a discussion on the invertibility properties of the function $\sigma(\theta)$.

Remark 12 (***Special case with σ being the identity operator***)**.** An important special case is obtained when σ is the identity operator on Θ; in this case, (23) simplifies to $s(\theta) = F_c(\theta, 0)$, which states that the controller vector field $F_c(z, y)$ must coincide with that of the exosystem $s(\theta)$ in $\Omega(\Theta_\circ)$. In this case, (27) gives $z(t) = \theta(t)$; namely, the controller state $z(t)$ and that of the exosystem $\theta(t)$ coincide in $\Omega(\Theta_\circ)$. □

While Theorem 4 provides a full characterization of all gradient-feedback algorithms that achieve tracking, it remains to address under what conditions such an algorithm is guaranteed to exist. This question is addressed next.

Theorem 5 (***Existence of gradient-feedback algorithms***)**.** Suppose Assumptions 1–3 hold. Then, there exists a gradient-feedback algorithm, such that $z = z_\circ^\star$ is exponentially stable for (22), that solves Problem 1 if and only if there exists a mapping $H_c : \Theta \to \mathbb{R}^n$ that zeros the gradient on the limit set of (2) with respect to its initializations. □

Proof. (Only if) By Theorem 4, there exists a mapping $z = \sigma(\theta)$ such that (23b) holds. Then, (16) holds immediately by letting $H_c(\theta) = G_c(\sigma(\theta))$.

(If) We will prove this claim by constructing a gradient-feedback algorithm that achieves $y(t) \to 0$ as $t \to \infty$. First, notice that by Assumption 3, there exists a matrix L such that $S - LQ$ has eigenvalues in $\mathbb{C}^-$. Consider the algorithm (10) with $n_c = p$ and[6]

$$F_c(z, y) = s(z) + L(y - \nabla_x f(H_c(z), z)), \tag{28a}$$
$$G_c(z) = H_c(z), \tag{28b}$$

where $H_c(z)$ is as in (16). The Jacobian of $\dot{z}(t) = F_c(z(t), y(t))$ with respect to z is given by $S - LQ$ (two terms of the form LRM with opposite sign cancel out in forming the Jacobian: one from the gradient term $\nabla_x f(H_c(z), z)$ and the other one from y); since $S - LQ$ has eigenvalues in $\mathbb{C}^-$, $z = z_\circ^\star$ is exponentially stable for (22). The claim thus follows by Theorem 4 with σ the identity operator on Θ. □

Interestingly, the conditions for the existence of a gradient-feedback algorithm and those for the existence of a parameter-feedback algorithm (cf. Section III) are identical. This is not surprising, since a parameter-feedback algorithm is assumed to have access to $\theta(t)$ while a gradient-feedback algorithm needs to measure $\theta(t)$ indirectly through $y(t)$. More precisely, as stated by (27), the dynamic state of the controller $z(t)$ acts as an alternative representation (i.e., in different coordinates) of the exosystem state $\theta(t)$, while the exploration signal $x(t)$ acts as a parameter-feedback algorithm (see (26)).

B. Design of dynamic gradient feedback algorithms

We now focus on addressing Problem 1. A design procedure to construct $F_c(z, y)$ and $G_c(z)$ is presented in Algorithm 1, which is derived from the proof of Theorem 5.

The algorithm uses a Luenberger observer to estimate the exosystem state $\theta(t)$ (cf. line 4), and a parameter feedback algorithm is then applied to the estimated exosystem state to regulate the gradient to zero (precisely, $G_c(z)$ is designed following the approach of Theorem 1 – cf. line 3 of Algorithm 1). Notice that the algorithm uses the particular choice for $\sigma(\theta)$ being the identity operator (see Remark 12). Note that line 2 of the algorithm guarantees that the equilibrium of (22) is locally exponentially stable, as required by Theorem 4, and that the existence of such L is guaranteed by Assumption 3.

Algorithm 1: Gradient-feedback algorithm design

Data: $s(\theta)$, $\nabla_x f(x, \theta)$, $G_c(\theta)$ satisfying (23b),
Jacobian matrices Q and S in (21)

1 $n_c \leftarrow p$;
2 $L \leftarrow$ any matrix such that $S - LQ$ is Hurwitz;
3 $G_c(z) \leftarrow H_c(z)$;
4 $F_c(z, y) \leftarrow s(z) + L(y - \nabla_x f(H_c(z), z))$;

Result: $F_c(z, y)$, $G_c(z)$, and n_c that solve Problem 1

Theorem 6 (***Correctness of Algorithm 1***)**.** Suppose Assumptions 1–3 hold. Then, the control algorithm (10) designed using Algorithm 1 asymptotically tracks a critical trajectory of (1).

Remark 13 (***Alternative observer choices***)**.** Instead of a Luenberger observer, alternative dynamic observers could be

[6]Note that the second argument of the gradient is evaluated at the algorithm state z instead of the parameter vector θ as the latter is unknown.

considered in Line 4 of the algorithm to achieve different asymptotic or transient properties of the resulting gradient-feedback optimization algorithm; we leave the investigation of alternative state observer algorithms to future works. □

We illustrate the algorithm design procedure on a quadratic problem in the following example.

Example 3. Consider the quadratic problem studied in Example 2, and assume that the exosystem follows the linear model $\dot{\theta} = S\theta$ for some matrix $S \in \mathbb{R}^{p\times p}$ as in Assumption 2. Further, suppose that Q is such that the unobservable subspace $\mathcal{N} = \cap_{i\geq 0} \ker(QS^i)$ is the empty set $\mathcal{N} = \emptyset$, so that Assumption 3 is satisfied; notice that this assumption holds, for example, when $n = p$ and Q is invertible. According to Theorem 5, an optimization algorithm given by

$$\dot{z} = A_c z + B_c y, \qquad x = G_c z, \qquad y = Rx + Q\theta, \tag{29}$$

where $A \in \mathbb{R}^{n_c\times n_c}$, $B_c \in \mathbb{R}^{n_c\times n}$, $G_c \in \mathbb{R}^{n\times n_c}$, achieves asymptotic tracking if and only if there exists a linear transformation $\Sigma \in \mathbb{R}^{n_c\times p}$ such that:

$$\Sigma S = A_c\Sigma, \tag{30a}$$
$$0 = RG_c\Sigma + Q. \tag{30b}$$

Next, we apply Algorithm 1 to design an algorithm that solves this problem. The mapping $H_c(\theta) = -R^\dagger Q\theta$ zeros the gradient (see Example 2). Following Algorithm 1, we set $n_c = p$, choose matrix L such that $S - LQ$ is Hurwitz, define $G_c(z) = H_c(z) = -R^\dagger Qz$, and then select A_c and B_c such that $F_c(z, y) = A_c z + B_c y$ satisfies:

$$\begin{aligned} F_c(z, y) &= s(z) + L(y - \nabla_x f(H_c(z), z)), \\ &= Sz + L(y - (RH_c(z) + Qz)), \\ &= Sz + Ly, \end{aligned}$$

which gives $A_c = S$ and $B_c = L$. In summary, a dynamic gradient-feedback algorithm is (29) with $A_c = S$, $B_c = L$, and $G_c = -R^\dagger Q$, where L is any matrix such that $S - LQ$ is Hurwitz. With this choice, conditions (30) are satisfied with $\Sigma = I$ (see Remark 12). Finally, the autonomous model (22) becomes $\dot{z}(t) = (S - LQ)z(t)$, which is asymptotically stable, as required by Theorem 4. □

C. The optimization algorithm as a copy of the exosystem

As shown by (27), the state of optimization algorithms that achieve asymptotic tracking and that of the exosystem are related by a change of coordinates. In this section, we show that $\sigma(\theta)$ is an injective map and thus is a well-posed change of coordinates. For Algorithm 1, we have the following.

Proposition 7 (***Injectivity of*** $\sigma(\theta)$ ***from Algorithm 1***)**.** Let the assumptions of Theorem 5 hold and let $F_c(z, y)$ and $G_c(z)$ be obtained by Algorithm 1. Then, there exists an injective map $\sigma(\theta)$ such that (27) holds. □

Proof. The claim follows by noting that $F_c(z, y)$ and $G_c(z)$ obtained by Algorithm 1 satisfy (23) with σ the identity. □

It follows from Proposition 7 that, for any optimization method returned by Algorithm 1, there exists an injective mapping that, locally, maps the state of the exosystem into the state of the optimization algorithm.

This property is not exclusive to Algorithm 1; in fact, any algorithm that achieves exact asymptotic tracking possesses the same characteristic, as demonstrated next.

Proposition 8 (***Injectivity of any*** $\sigma(\theta)$)**.** Let the assumptions of Theorem 5 hold and assume that $s(\theta)$ is C^∞. For any $F_c(z, y)$ and $G_c(z)$ that achieve asymptotic tracking, there exists a neighborhood $\Theta_\circ \subset \Theta$ of the origin such that $\sigma(\theta)$ is injective in $\Theta_\circ$. □

Proof. By contradiction, assume that $F_c(z, y)$ and $G_c(z)$ satisfy (23), but $\sigma(\theta)$ is not injective at the origin; namely, there exists nonzero $\theta' \in \Theta_\circ$ such that $\sigma(\theta') = z_0^\star$. Then from (23b), $0 = \nabla_x f(G_c(\sigma(\theta')), \theta')$. Moreover, from Theorem 4,

$$\frac{\partial\sigma(\theta')}{\partial\theta} s(\theta') = F_c(\sigma(\theta'), 0) = 0. \tag{31}$$

For C^∞ vector fields $h_1(x)$ and $h_2(x)$, we let $L_{h_1}(h_2)(x) = \frac{\partial h_2(x)}{\partial x} h_1(x)$. By application of (31), we have

$$0 = L_s(\nabla_x f)(x, \theta') = L_s(L_s(\nabla_x f))(x, \theta') = \dots .$$

Hence, the matrix of Lie derivatives of the measurable signal, which, intuitively, characterizes the information content of the gradient signal:

$$\begin{bmatrix} L_s(\nabla_x f)(x, \theta) \\ L_s(L_s(\nabla_x f))(x, \theta) \\ \vdots \end{bmatrix},$$

is not invertible at $\theta = \theta'$. By [53, Thm. 3.13], the system (2) is not weakly observable, thus violating Assumption 3. □

V. Fidelity of the Internal Model vs Accuracy

In Section IV, we have shown that a necessary condition for exact asymptotic tracking is that the algorithm incorporates an internal model of the temporal variability of the problem. Departing from this finding, in this section we tackle the following question: *how will the algorithm perform when the temporal variability of the problem is known only approximately?* For illustration purposes, we focus on the quadratic problem (19) (see Examples 2 and 3). Suppose that an imprecise internal model is available; that is, there exists $\Delta \in \mathbb{R}^{n_c\times p}$ such that (30) is modified to[7]:

$$\Sigma S + \Delta = A_c\Sigma \qquad \text{and} \qquad 0 = RG_c\Sigma + Q. \tag{32}$$

To analyze the asymptotic behavior of $y(t)$, it is useful to define the auxiliary variable $\tilde{z}(t) = z(t) - \Sigma\theta(t)$. Using (32) and (29), the dynamics of $\tilde{z}$ follow the model:

$$\dot{\tilde{z}} = (A_c + B_c RG_c)\tilde{z} + \Delta\theta. \tag{33}$$

[7]To explicitly model uncertainty in the matrix S, equation (32) could be rewritten as $\Sigma(S + \Delta_s) = A_c\Sigma$. By comparison, the two approaches are equivalent with the substitution $\Delta = \Sigma\Delta_s$.

The corresponding gradient signal in terms of $\tilde{z}$ is $y = RG_c\tilde{z}$. Assuming that $(A_c + B_cRG_c)$ is Hurwitz stable (see Theorem 4), the Final Value Theorem gives:

$$\begin{aligned} y_\infty &:= \lim_{t\to\infty} y(t) \\ &= \lim_{s\to 0} sRG_c(sI - A_c - B_cRG_c)^{-1}\Delta(sI - S)^{-1}\theta(0). \end{aligned} \tag{34}$$

As a first illustrative scenario, suppose $S = 0$; namely, that the exosystem states are constants at all times. Notice that, since $S = 0$, from (30a), A_c incorporates an internal model of S if and only if $A_c\Sigma = 0$; precisely, if and only if the dimension of the kernel of A_c is at least p. In this case, assuming that the perturbed A_c is such that $(A_c+B_cRG_c)$ remains Hurwitz stable, from (34), we have $y_\infty = -RG_c(A_c+B_cRG_c)^{-1}\Delta\theta(0)$. As expected, when $\Delta = 0$, A_c incorporates an exact internal model, and $y_\infty = 0$ for any $\theta(0)$. In contrast, when $\Delta \neq 0$,

$$\|y_\infty\| \leq \|RG_c(A_c + B_cRG_c)^{-1}\|\|\theta(0)\|\|\Delta\|,$$

namely, bounded errors in Δ result in a bounded y_∞.

Interestingly, analogous continuity properties may not hold when the trajectories of (2) are not bounded (thus violating Assumption 2). Consider an instance of (19) with $n = p = 2$, $R = Q = I$, and $S = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$. The solutions of (2) with this exosystem are a linear combination of modes e^{0t} and te^{0t}, and thus grow unbounded at a linear rate. Suppose algorithm (29) is used with $G_c = I$, $B_c = I$, and $A_c = \left[\begin{smallmatrix} -\epsilon_1 & 1 \\ 0 & -\epsilon_2 \end{smallmatrix}\right]$, where $\epsilon_1, \epsilon_2 \in [0, 1)$. When $\epsilon_1 = \epsilon_2 = 0$, (32) holds with $\Sigma = -I$ and $\Delta = 0$, in which case $y_\infty = 0$ and exact asymptotic tracking is achieved. On the other hand, when $\epsilon_1, \epsilon_2 \neq 0$, (32) holds with $\Delta = \left[\begin{smallmatrix} \epsilon_1 & 0 \\ 0 & \epsilon_2 \end{smallmatrix}\right]$ and, from (34), we obtain:

$$y_\infty = \lim_{s\to 0} \begin{bmatrix} \frac{\epsilon_1}{s-\alpha_1} & \frac{\epsilon_1}{s(s-\alpha_1)} + \frac{\epsilon_2}{(s-\alpha_1)(s-\alpha_2)} \\ 0 & \frac{\epsilon_2}{s-\alpha_2} \end{bmatrix} \begin{bmatrix} \theta_1(0) \\ \theta_2(0) \end{bmatrix},$$

where $\alpha_i = 1 - \epsilon_i$, $i \in \{1, 2\}$. Since $\epsilon_1 \neq 0$, and provided that $\theta_2(0) \neq 0$, the above identity implies that $\|y(t)\| \to \infty$ as $t \to \infty$, i.e., the asymptotic tracking error grows unbounded.

This example illustrates that the tracking accuracy depends on the fidelity of the internal model but also the asymptotic behavior of the exosystem.

VI. Extensions

In this section, we discuss possible extensions to constrained optimization problems and to discrete-time algorithms.

A. Constrained problems

Our results can be extended to constrained time-varying optimization. Consider the equality-constrained problem

$$\begin{aligned} &\text{minimize} && f(x, \theta(t)) \\ &\text{subject to} && h_i(x, \theta(t)) = 0, \quad i = 1, \ldots, m, \end{aligned}$$

where the constraint functions $h_i(x, \theta(t))$ depend on the parameter vector. The associated Lagrangian function is

$$L(x, \lambda, \theta(t)) = f(x, \theta(t)) + \sum_{i=1}^{m} \lambda_i h_i(x, \theta(t)),$$

where λ_i is the Lagrange multiplier associated with the i^{th} equality constraint. A pair $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m$ is said to be a saddle-point of the Lagrangian if, for all $(\bar{x}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^m$,

$$L(x, \bar{\lambda}, \theta(t)) \leq L(x, \lambda, \theta(t)) \leq L(\bar{x}, \lambda, \theta(t)).$$

For any such saddle-point, if strong duality holds, x is primal optimal, λ is dual optimal, and the optimal duality gap is zero. Moreover, the gradient of the Lagrangian (assuming it exists) is zero at any saddle-point. It follows from the derivations in the previous sections that the gradient-feedback and parameter-feedback algorithms can be directly applied to seek a stationary point of the Lagrangian function by replacing (in (1)) the variable x with the extended decision variable $\tilde{x} = (x, \lambda)$ and by letting $f(\tilde{x}, \theta) = L(x, \lambda, \theta)$. Notice that, if the critical point computed by (10b) is also a saddle-point and strong duality holds, then it is also a solution to the equality-constrained problem. When strong duality does not hold, however, such a saddle-point may not correspond to an optimizer [54, Ch. 5].

B. Discrete-time algorithms

In practice, optimization algorithms are often implemented on digital machines and thus, in these cases, implementing (10) requires discretizations. In these cases, a possible solution is to pursue a *design by emulation* [55], which consists in converting the continuous-time algorithm (10) into a discrete-time one by approximation (that is, by approximating the time-derivative by a discrete difference using, e.g., Backward/Forward Difference, Tustin, ZOH methods, etc.). Then, the continuous-time guarantees given here can be translated into discrete-time ones (using standard tools from emulation [55] or sampled-data systems [56]), provided that the sampling time is sufficiently small. In these cases, exact tracking established here will continue to hold at the discrete time instants, while tracking will be inexact in-between sampling times and the error can be bounded in terms of the sampling frequency using the explicit bounds in [55].

An alternative method that could be pursued is that of discretizing the temporal variability $\theta(t)$, thus leading to a discrete-time optimization problem as in (4). For this alternative formulation, the critical trajectories are discrete signals and, in this case, it would be natural to seek a fully-discrete alternative to (10) and pursue an analysis in discrete time. Such a design is beyond the scope of this work and will be left as the focus of future works.

VII. Simulation results

In this section, we illustrate our results and optimization design method through a set of numerical simulations. For implementation on digital machines, all algorithms have been discretized by using the explicit Runge-Kutta-Fehlberg method of order $4(5)$ with adaptive stepsize [57].

A. Optimization design for quadratic costs

We begin by numerically investigating the quadratic instance of (1) with linear temporal variability, discussed previously in Examples 2 and 3. With dimensions $n = p = 4$,

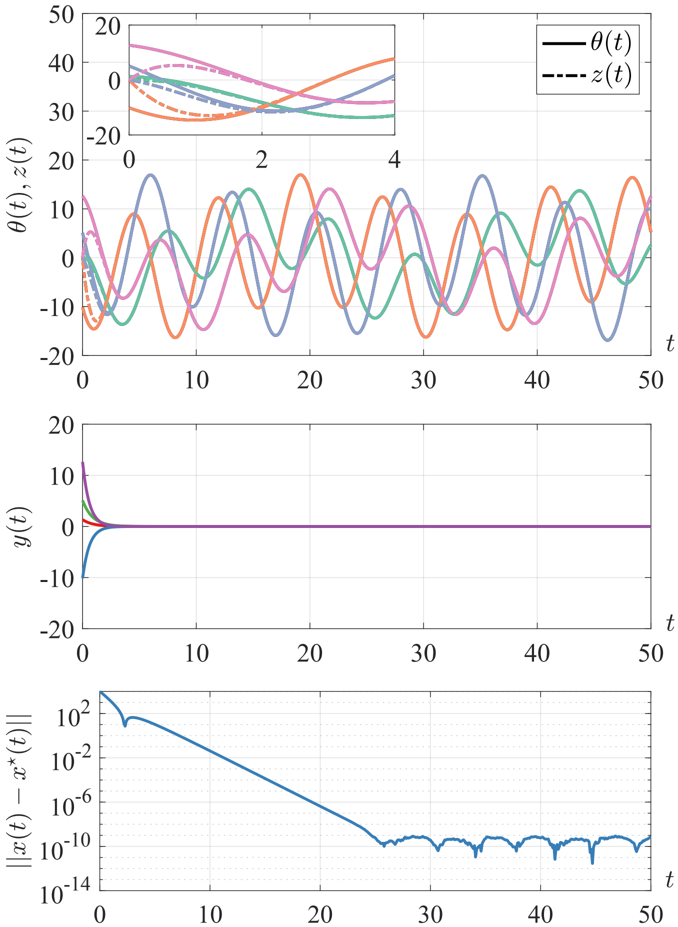


Fig. 3: Simulation results illustrating the performance of an optimization algorithm synthesized using Algorithm 1 for the quadratic instance (19) of (1). See Example 2 and Section VII-A for a discussion. (Top) Illustration of the temporal variability of the parameter $\theta(t)$ and of $z(t)$. (Second from top) $z(t)$ is an estimator for $\theta(t)$, and thus $z(t) \to \theta(t)$ as $t \to \infty$. (Third from top) The proposed control algorithm is successful in regulating the gradient feedback signal $y(t)$ to zero asymptotically. (Bottom) Illustration that $x(t) \to x^\star(t)$ as $t \to \infty$.

we chose the matrix $R \in \mathbb{S}^4$ with random entries such that its eigenvalues are uniformly distributed in the open real interval $(0, 1)$, and we set $Q = I \in \mathbb{R}^{4\times4}$. We let the exosystem be $\dot{\theta}(t) = S\theta(t)$, where $S \in \mathbb{R}^{4\times4}$ is given by $S = \tilde{S} - \tilde{S}^\mathsf{T}$, and $\tilde{S}$ is a matrix with random entries uniformly distributed in the open real interval $(0, 1)$. Notice that this choice ensures that the eigenvalues of S are on the imaginary axis. We chose $H_c(\theta) = H_c\theta$ with $H_c = -R^{-1}Q$, which ensures (16) holds (see Example 2). We applied Algorithm 1, choosing L such the eigenvalues of $S - LQ$ are uniformly distributed in the real interval $(-2, -1)$, as described in Example 3. Simulation results are presented in Fig. 3. The top plot illustrates the time-varying signal $\theta(t)$ and the state of the optimization algorithm $z(t)$; the figure illustrates that $z(t)$ converges to $\theta(t)$ asymptotically (see discussion immediately after Theorem 5). The middle plot illustrates the gradient signal, validating the claim that the algorithm converges to a critical point (where the gradient is zero). Finally, the bottom plot illustrates, in log scale, the error between the optimization state $x(t)$ and the optimizer $x^\star(t)$; this figure shows that convergence to the optimizer is exponential, up to numerical precision.

B. Nonlinear example

We now illustrate our algorithm design on a nonlinear time-varying optimization problem from [30], [34], and we compare the results with several prediction-correction algorithm variants. Consider minimizing the function:

$$f(x, \theta) = \tfrac{1}{2}(x - \theta)^2 + \kappa \log(1 + \exp(\mu x)),$$

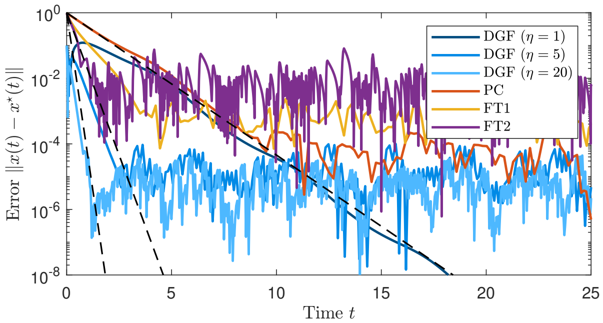
with parameters $\kappa = 7.5$ and $\mu = 1.5$, where the time-varying parameter $\theta(t)$ satisfies the exosystem dynamics $\dot{\theta} = S\theta$ where $S = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$. We compare the error $\|x(t) - x^\star(t)\|$ over time for several algorithms in Fig. 4. The objective function satisfies the conditions in Theorem 3; for our simulations, we computed numerically a mapping that zeros the gradient locally. In Fig. 4(a), we show the error for our algorithm with observer pole locations $(-\eta, -2\eta)$ for $\eta \in \{1, 5, 20\}$ (shown in dark to light blue) along with the corresponding rate $e^{-\eta t}$ (dashed black). For comparison, we also show the error of the prediction-correction (PC) algorithm (8) along with two of its finite-time (FT) variants [34, Equations (9) and (20)] using the parameter values in [34, Section VI.A].

Instead of computing $H_c(\theta)$ numerically, we can use its linear approximation in (20). Doing so yields the results in Fig. 4(b) for various size exosystem initial conditions $\theta(0) = \zeta\theta_0$ for $\zeta \in \{1, 0.1, 0.01\}$. The linear approximation does not zero the gradient exactly, which results in a finite steady-state error whose size depends on the size of the limit set of the exosystem (which corresponds to the size of the initial conditions since the exosystem dynamics are sinusoidal).

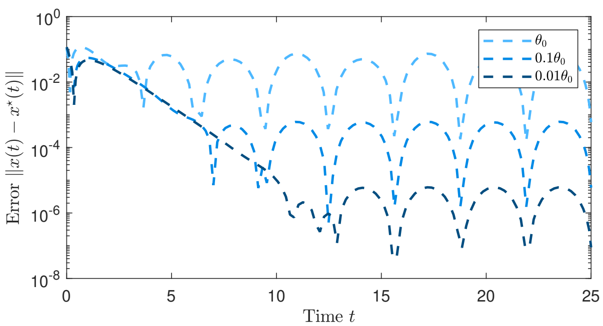
C. Application to solve the dynamic traffic assignment problem in transportation

We next illustrate the applicability of the framework in solving the dynamic traffic assignment problem in roadway transportation [58]; intuitively, the objective is to decide how traffic flows are split among the available paths of a network to minimize the drivers' travel time to destination. We model a roadway transportation network using a static flow model [58], described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with edges $i \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ (modeling traffic roads) and nodes $\mathcal{V}$ (modeling traffic junctions). For $i \in \mathcal{E}$, we denote by $i^+ \in \mathcal{V}$ and $i^- \in \mathcal{V}$ its origin and destination nodes, respectively. We assume that an exogenous, time-varying, inflow of traffic $\theta(t)$ enters the network at a certain origin node, denoted by $o \in \mathcal{V}$, and exits at a certain destination node, denoted by $d \in \mathcal{V}$; for simplicity, we assume that there is only one origin-destination pair, but this is without loss of generality[8]. We describe the network state using a vector $x \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$ (where $|\mathcal{E}|$ denotes the number of edges) whose entries x_i describe the amount of inflow θ routed through road i. To each link i, we associate a function $\ell_i(x_i)$ describing the latency (or travel time) of the road i. For our simulations, we consider the network topology in Fig. 5,

[8]When the network has multiple origin-destination pairs, an optimization problem of the form (35) will be associated with each origin-destination pair. Because the optimization problems are independent from one another [58], each of them can be solved independently using the methods illustrated here.



(a) Error over time for the proposed dynamic gradient-feedback (DGF) algorithm with observer pole locations $(-\eta, -2\eta)$ for $\eta \in \{1, 5, 20\}$ (dark to light blue) along with the prediction-correction (PC) algorithm and two finite-time (FT) variants.



(b) Error over time for the proposed dynamic gradient-feedback algorithm using the linear approximation to $H_c(\theta)$ in (20) with different size initial conditions $\theta(0) = \zeta\theta_0$ for $\zeta \in \{1, 0.1, 0.01\}$. When the initial conditions are small, the exosystem state stays near the equilibrium, in which case H_c approximately zeros the gradient on the limit set of the exosystem.

Fig. 4: Dynamic gradient-feedback algorithm applied to the nonlinear time-varying optimization problem from Section VII-B.

and choose the latency functions as follows:

$$\ell_1(x_1) = x_1, \qquad \ell_2(x_2) = 10x_2, \qquad \ell_3(x_3) = x_3,$$
$$\ell_4(x_4) = 5x_4, \qquad \ell_5(x_5) = x_5.$$

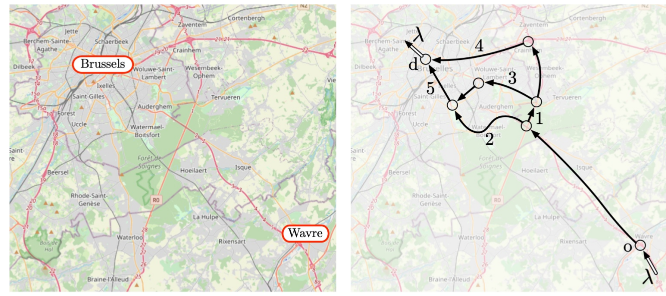
According to Wardrop's first principle [58, pp. 31], transportation networks operate at a condition where travelers select their path to minimize their travel time to their destination. Mathematically, Wardrop's equilibrium is the optimizer of the following optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^{|\mathcal{E}|}} \quad & \sum_{i \in \mathcal{E}} \int_0^{x_i} \ell_i(s) ds \\ \text{subject to:} \quad & \sum_{j \in \mathcal{E}: j^- = v} x_j - \sum_{j \in \mathcal{E}: j^+ = v} x_j = \delta_v(\theta(t)), \ \forall v \in \mathcal{V}, \\ & x_i \geq 0, \ \forall i \in \mathcal{E}, \end{aligned} \tag{35}$$

where $\delta_v(\theta)$ for $v \in \mathcal{V}$ is defined as $\delta_o(\theta) = \theta$ for the origin node, $\delta_d(\theta) = -\theta$ for the destination node, and $\delta_v(\theta) = 0$ for all other nodes. The loss function in (35) is used to model travelers who will switch to a different path if it has a shorter travel time to destination, while the first constraint in (35) describes the network topology, namely, that traffic flows are conserved at each node. Notice that (35) is a time-varying optimization problem, where the temporal variability originates from the dependence of the constraint on $\theta(t)$, which describes the inflow of vehicles at the origin and outflow at the destination, measured in vehicles per hour. For our simulations, we assume that the network inflow is sinusoidal:

$$\theta(t) = \theta_0 - \theta_1 \cos(\omega_1 t + \phi_1) - \theta_2 \cos(\omega_2 t + \phi_2), \tag{36}$$

where $\theta_0, \theta_1, \theta_2, \omega_1, \omega_2, \phi_1, \phi_2 \in \mathbb{R}_{>0}$, satisfy $\theta_0 > \theta_1$, $\theta_0 > \theta_2$, and $\omega_2 > \omega_1$. The model (36) states that the network inflow is the sum of a constant term, θ_0, a slowly-varying sinusoid with angular frequency ω_1 and a quickly-varying sinusoid with angular frequency ω_2. The low-frequency sinusoid is used here to describe slowly-varying (e.g., hourly) traffic demands, while the high-frequency sinusoid is used to model sudden (e.g., at the minute level) variations in traffic demand. For our simulations, we let $\theta_0 = 3$ veh/h, $\theta_1 = 1$ veh/h, $\theta_2 = 0.1$ veh/h, $\omega_1 = 0.1$ rad/hour, $\omega_2 = \sqrt{50}$ rad/hour, and $\phi_1 = \phi_2 = 0$, see Fig. 6 (top).



(a) (b)

Fig. 5: (Left) Areal view of the highway system between the cities of Wavre and Brussels, Belgium. (Right) Graph utilized to model the portion of traffic network of interest.

We applied Algorithm 1 to derive an optimization algorithm to solve the traffic assignment problem (35). For the synthesis, we utilized the internal model $s(z) = Sz$, where

$$S = \text{diag}\left(\begin{bmatrix} 0 & 1 \\ -\omega_1^2 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ -\omega_2^2 & 0 \end{bmatrix}, 0 \right).$$

Notice that knowledge of $\theta_0, \theta_1, \theta_2, \phi_1, \phi_2$ is not required to synthesize the optimization algorithm – only the frequencies ω_1, ω_2 are required to be known. To seek a solution to the constrained problem (35), consider the Lagrangian function:

$$\begin{aligned} L(x, \lambda, \theta(t)) := & \sum_{i \in \mathcal{E}} \int_0^{x_i} \ell_i(s) ds \\ & + \sum_{v \in \mathcal{V}} \lambda_v \Bigg(\delta_v(\theta(t)) - \sum_{j \in \mathcal{E}: j^- = v} x_j + \sum_{j \in \mathcal{E}: j^+ = v} x_j \Bigg), \end{aligned}$$

where $\lambda := (\lambda_1, \ldots, \lambda_{|\mathcal{V}|})$ is the vector of Lagrange multipliers. We applied Algorithm 1 to the optimization problem (1) with $f(\tilde{x}, \theta) = L(x, \lambda, \theta)$ (see Section VI); the inequality constraints in (35) have been accounted for by projecting $x(t)$ onto the feasible set. Here, matrix L has been chosen so that the eigenvalues of $S - LQ$ are uniformly distributed in the open real interval $(-1, -2)$. Notice that, since the latencies are strictly increasing, the Lagrangian is strongly convex-strongly concave [58], and thus the problem admits a unique critical point that is a saddle point. It follows that our algorithm is

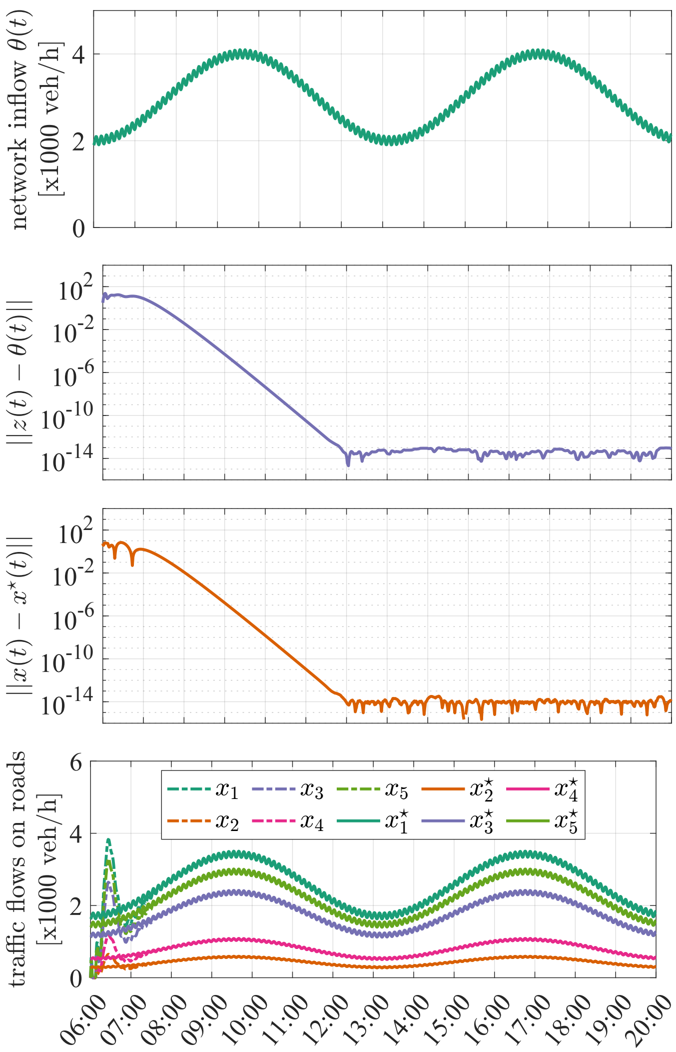


Fig. 6: Illustration of the performance of the optimization method synthesized using Algorithm 1 to solve the dynamic traffic assignment problem (35). See Section VII-C for a discussion. The bottom figure shows that, from a zero initial condition, the algorithm is capable of computing a Wardrop equilibrium in about 1.5 hours, and then of tracking this equilibrium. Notice that, a faster rate of convergence could be obtained by shifting the eigenvalues of $S - LQ$.

guaranteed to converge to a minimizer of (35). Simulations for this problem are presented in Fig. 6. The simulation shows that $z(t) \to \theta(t)$ as $t \to \infty$ (see Fig. 6-second figure) and $y(t) \to 0$ (up to machine precision) which implies that the algorithm successfully computes a critical point (see Fig. 6-third figure). Finally, the bottom figure of Fig. 6 illustrates the rate of convergence of the algorithm, which, as expected, is governed by the placement of the eigenvalues of the observer.

VIII. Conclusions

We showed that the problem of designing algorithms for time-varying optimization problems can be reformulated as a nonlinear multivariate output regulation problem. This connection allowed us to prove the internal model principle of time-varying optimization, which states that (when the algorithm has access only to functional evaluations of the gradient) asymptotic tracking can be achieved only if the algorithm incorporates a reduplicated model of the temporal variability of the problem. On the other hand, when the time-varying parameters are measurable, asymptotic tracking can be achieved by seeking a mapping that zero the gradient. Importantly, our results show that asymptotic tracking can be achieved under more relaxed assumptions than what is normally imposed in the literature. Moreover, our algorithm structure is novel in the literature, and it relies on the use of an observer for the temporal variability of the problem. This work opens the opportunity for several directions of future work, including the relaxation of the convexity and smoothness assumptions, an investigation of discrete-time problems, and applications to feedback optimization.

Appendix

We now summarize relevant facts in center manifold theory from [19]; see also [59]. Consider the nonlinear system:

$$\dot{x} = f(x) \tag{37}$$

where f is a C^k vector field defined on an open subset U of $\mathbb{R}^n$, and let $x_\circ \in U$ be an equilibrium point for f, i.e., $f(x_\circ) = 0$. Without loss of generality, suppose $x^\circ = 0$. Let $F = \left[\frac{\partial f}{\partial x}\right]_{x=0}$, denote the Jacobian matrix of f at $x = 0$. Suppose the matrix F has n° eigenvalues with zero real part, n^- eigenvalues with negative real part, and n^+ eigenvalues with positive real part. Let E^-, E°, and E^+ be the (generalized) real eigenspaces of F associated with eigenvalues of F lying on the open left half plane, the imaginary axis, and the open right half plane, respectively. Note that E°, E^-, E^+ have dimension n°, n^-, n^+, respectively and that each of these spaces is invariant under the flow of $\dot{x} = Fx$. If the linear mapping F is viewed as a representation of the differential (at $x = 0$) of the nonlinear mapping f, its domain is the tangent space T_0U to U at $x = 0$, and the three subspaces in question can be viewed as subspaces of T_0U satisfying $T_0U = E^\circ \oplus E^- \oplus E^+$. We refer to [60, Sec. A.II] for a precise definition of C^k manifolds; loosely speaking, a set $S \subset U$ is a C^k manifold it can be locally represented as the graph of a C^k function.

Definition 4 (Locally invariant manifold)**.** A C^k manifold S of U is locally invariant for (37) if, for each $x_\circ \in S$, there exists $t_1 < 0 < t_2$ such that the integral curve $x(t)$ of (37) satisfying $x(0) = x_\circ$ satisfiest $x(t) \in S$ for all $t \in (t_1, t_2)$. $\square$

Intuitively, by letting $x = (y, \theta)$ and expressing (37) as:

$$\dot{y} = f_y(\theta, y), \qquad \dot{\theta} = f_\theta(\theta, y), \tag{38}$$

a curve $y = \pi(\theta)$ is an invariant manifold for (38) if the solution of (38) with $\theta(0) = \theta_\circ$ and $y(0) = \pi(\theta_\circ)$ lies on the curve $y = \pi(\theta)$ for t in a neighborhood of 0. The notion of invariant manifold is useful as, under certain assumptions, it allows us to reduce the analysis of (37) to the study of a reduced system in the variable θ only. The remainder of this section is devoted to formalizing this fact.

Definition 5 (Center manifold)**.** Let $x = 0$ be an equilibrium of (37). A manifold S, passing through $x = 0$, is said to be a center manifold for (37) at $x = 0$ if it is locally invariant and the tangent space to S at 0 is exactly E°. $\square$

Intuitively, the invariant manifold $y = \pi(\theta)$ is a center manifold for (38) when all orbits of y decay to zero and those of θ neither decay nor grow exponentially.

In what follows, we will assume that all eigenvalues of F have nonpositive real part, i.e., $n^+ = 0$. When this holds, it is always possible to choose coordinates in U such that (37) is

$$\dot{y} = Ay + g(y, \theta), \tag{39a}$$
$$\dot{\theta} = B\theta + h(y, \theta), \tag{39b}$$

where A is an $n^- \times n^-$ matrix having all eigenvalues with negative real part, B is an $n^\circ \times n^\circ$ matrix having all eigenvalues with zero real part, and the functions g and h are C^k functions vanishing at $(y, \theta) = (0, 0)$, together with all their first-order derivatives. Because of their equivalence, any conclusion drawn for (39) will apply also to (37). The following result ensures the existence of a center manifold.

Theorem 9 (Center manifold existence theorem)**.** Assume that $n^+ = 0$. There exists a neighborhood $V \subset \mathbb{R}^{n^0}$ of 0 and a class C^{k-1} mapping $\pi : V \to \mathbb{R}^{n^-}$ such that the set

$$S = \{(y, \theta) \in \mathbb{R}^{n^-} \times V : y = \pi(\theta)\},$$

is a center manifold for (39). □

Some important observations are in order. By definition, a center manifold for (39) passes through $(0, 0)$ and is tangent to the subset of points whose y coordinate is zero. Namely,

$$\pi(0) = 0 \qquad \text{and} \qquad \frac{\partial \pi}{\partial \theta}(0) = 0. \tag{40}$$

Moreover, this manifold is locally invariant for (39): this imposes on the mapping π the constraint:

$$\frac{\partial \pi}{\partial \theta}(B\theta + h(\pi(\theta), \theta)) = A\pi(\theta) + g(\pi(\theta), \theta), \tag{41}$$

as deduced by differentiating with respect to time any solution $(y(t), \theta(t))$ of (39) on the manifold $y(t) = \pi(\theta(t))$. In other words, any center manifold for (39) can equivalently be described as the graph of a mapping $y = \pi(\theta)$ satisfying the partial differential equation (41), with the constraints (40).

Remark 14. Theorem 9 shows existence but not the uniqueness of a center manifold. Moreover, (i) if g and h are $C^k, k \in \mathbb{N}_{>0}$, then (39) admits a C^{k-1} center manifold; (ii) if g and h are C^∞ functions, then (39) has a C^k center manifold for any finite k, but not necessarily a C^∞ center manifold. □

The next result shows that any y-trajectory of (39), starting sufficiently close to the origin converges, as time tends to infinity, to a trajectory that belongs to the center manifold.

Theorem 10. Assume that $n^+ = 0$ and suppose $y = \pi(\theta)$ is a center manifold for (39) at $(0, 0)$. Let $(y(t), \theta(t))$ be a solution of (39). There exists a neighborhood U° of $(0, 0)$ and real numbers $M > 0$ and $K > 0$ such that, if $(y(0), \theta(0)) \in U^\circ$, then for all $t \geq 0$,

$$\|y(t) - \pi(\theta(t))\| \leq Me^{-Kt}\|y(0) - \pi(\theta(0))\|. \qquad \square$$

From the above discussion, any trajectory of (39) starting at a point $y^\circ = \pi(\theta^\circ)$ of a center manifold satisfies $y(t) = \pi(\zeta(t))$ and $\theta(t) = \zeta(t)$, where $\zeta(t)$ is any solution of

$$\dot{\zeta} = B\zeta + h(\pi(\zeta), \zeta), \qquad \zeta(0) = \theta_\circ. \tag{42}$$

This decomposition allows us to predict the asymptotic behavior of (39) by studying the asymptotic behavior of the reduced-order system (42). This is formalized in the following result.

Theorem 11. Suppose $\zeta = 0$ is a stable (respectively, asymptotically stable, unstable) equilibrium of (42). Then $(y, \theta) = (0, 0)$ is a stable (respectively, asymptotically stable, unstable) equilibrium of (39). □

[1] Y. Chen and Y. Zhou, "Machine learning based decision making for time varying systems: Parameter estimation and performance optimization," *Knowledge-Based Systems*, vol. 190, p. 105479, 2020.

[2] A. Rakhlin and K. Sridharan, "Online learning with predictable sequences," in *Conference on Learning Theory*, 2013, pp. 993–1019.

[3] F. Y. Jakubiec and A. Ribeiro, "D-map: Distributed maximum a posteriori probability estimation of dynamic systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 450–466, 2012.

[4] A. Davydov, V. Centorrino, A. Gokhale, G. Russo, and F. Bullo, "Contracting dynamics for time-varying convex optimization," *arXiv preprint*, 2023, arXiv:2305.15595.

[5] B. Chachuat, B. Srinivasan, and D. Bonvin, "Adaptation strategies for real-time optimization," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1557–1567, 2009.

[6] V. M. Zavala and M. Anitescu, "Real-time nonlinear optimization as a generalized equation," *SIAM Journal on Control and Optimization*, vol. 48, no. 8, pp. 5444–5467, 2010.

[7] W. Su, S. Boyd, and E. J. Candes, "A differential equation for modeling nesterov's accelerated gradient method: Theory and insights," *Journal of Machine Learning Research*, vol. 17, no. 153, pp. 1–43, 2016.

[8] A. Cherukuri, E. Mallada, and J. Cortés, "Asymptotic convergence of constrained primal–dual dynamics," *Systems & Control Letters*, vol. 87, pp. 10–15, 2016.

[9] M. Muehlebach and M. Jordan, "A dynamical systems perspective on nesterov acceleration," in *International Conference on Machine Learning*, 2019, pp. 4656–4662.

[10] Y. Zhao and M. N. S. Swamy, "A novel technique for tracking time-varying minimum and its applications," in *IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 2, 1998, pp. 910–913.

[11] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1973–1986, 2017.

[12] N. Bastianello, R. Carli, and S. Zampieri, "Internal model-based online optimization," *IEEE Transactions on Automatic Control*, vol. 69, no. 1, pp. 689–696, 2024.

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

[14] G. Bianchin, J. Cortés, J. I. Poveda, and E. Dall'Anese, "Time-varying optimization of LTI systems via projected primal-dual gradient flows," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 474–486, Mar. 2022.

[15] A. Simonetto, E. Dall'Anese, J. Monteil, and A. Bernstein, "Personalized optimization with user's feedback," *Automatica*, vol. 131, p. 109767, 2021.

[16] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[17] J. Hepburn and W. Wonham, "Error feedback and internal models on differentiable manifolds," *IEEE Transactions on Automatic Control*, vol. 29, no. 5, pp. 397–403, 1984.

[18] B. A. Francis, "The linear multivariable regulator problem," *SIAM Journal on Control and Optimization*, vol. 15, no. 3, pp. 486–505, 1977.

[19] J. Carr, *Applications of centre manifold theory*. Springer-Verlag, 1981, vol. 35.

[20] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[21] S. M. Fosson, “Centralized and distributed online learning for sparse time-varying optimization,” *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2542–2557, 2020.

[22] S. Shalev-Shwartz, “Online learning and online convex optimization,” *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[23] R. Dixit, A. S. Bedi, R. Tripathi, and K. Rajawat, “Online learning with inexact proximal online gradient descent algorithms,” *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1338–1352, 2019.

[24] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” in *International conference on machine learning*, 2003, pp. 928–936.

[25] E. Hazan, A. Agarwal, and S. Kale, “Logarithmic regret algorithms for online convex optimization,” *Machine Learning*, vol. 69, no. 2, pp. 169–192, 2007.

[26] E. Hazan, “Introduction to online convex optimization,” *Foundations and Trends in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[27] Y. Li, G. Qu, and N. Li, “Using predictions in online optimization with switching costs: A fast algorithm and a fundamental limit,” in *American Control Conference*, 2018, pp. 3008–3013.

[28] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, “Online optimization: Competing with dynamic comparators,” in *Artificial Intelligence and Statistics*, 2015, pp. 398–406.

[29] O. Dekel, N. Haghtalab, and P. Jaillet, “Online learning with a hint,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[30] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, “A class of prediction-correction methods for time-varying convex optimization,” *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4576–4591, 2016.

[31] M. Marchi, J. Bunton, J. P. Silvestre, and P. Tabuada, “A framework for time-varying optimization via derivative estimation,” *arXiv preprint*, 2024, arXiv:2403.19088.

[32] A. Hauswirth, Z. He, S. Bolognani, G. Hug, and F. Dörfler, “Optimization algorithms as robust feedback controllers,” *Annual Reviews in Control*, vol. 57, p. 100941, 2024.

[33] A. Simonetto and E. Dall’Anese, “Prediction-correction algorithms for time-varying constrained optimization,” *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5481–5494, 2017.

[34] R. Raveendran, A. D. Mahindrakar, and U. Vaidya, “Fixed-time dynamical system approach for solving time-varying convex optimization problems,” in *American Control Conference*, 2022, pp. 198–203.

[35] Y. Shen, C. Chen, X. Zeng, and W. Chen, “Distributed algorithm for time-varying convex optimization with fixed-time convergence,” in *Internat. Conference on Control & Automation*, 2024, pp. 314–318.

[36] R. Raveendran, A. D. Mahindrakar, and U. Vaidya, “Dynamical system approach for time-varying constrained convex optimization problems,” *IEEE Transactions on Automatic Control*, 2023.

[37] K. Garg and D. Panagou, “Fixed-time stable gradient flows: Applications to continuous-time optimization,” *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2002–2015, 2020.

[38] E. Davison, “The robust control of a servomechanism problem for linear time-invariant multivariable systems,” *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 25–34, 1976.

[39] A. Isidori and C. I. Byrnes, “Output regulation of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 35, no. 2, pp. 131–140, 1990.

[40] C. I. Byrnes, F. D. Priscoli, A. Isidori, and W. Kang, “Structurally stable output regulation of nonlinear systems,” *Automatica*, vol. 33, no. 3, pp. 369–385, 1997.

[41] C. I. Byrnes, A. Isidori, and L. Praly, “On the asymptotic properties of a system arising in non-equilibrium theory of output regulation,” *Preprint of the Mittag-Leffler Institute, Stockholm*, vol. 18, pp. 2002–2003, 2003.

[42] L. Wang, A. Isidori, H. Su, and L. Marconi, “Nonlinear output regulation for invertible nonlinear MIMO systems,” *International Journal of Robust and Nonlinear Control*, vol. 26, no. 11, pp. 2401–2417, 2016.

[43] A. Isidori, L. Marconi, and L. Praly, “Robust design of nonlinear internal models without adaptation,” *Automatica*, vol. 48, no. 10, pp. 2409–2419, 2012.

[44] J. Huang, *Nonlinear output regulation: theory and applications*. SIAM, 2004.

[45] J. Huang, A. Isidori, L. Marconi, M. Mischiati, E. Sontag, and W. M. Wonham, “Internal models in control, biology and neuroscience,” in *IEEE Conf. on Decision and Control*, 2018, pp. 5370–5390.

[46] R. K. Sundaram, *A first course in optimization theory*. Cambridge university press, 1996.

[47] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed., ser. International Series in Pure and Applied Mathematics. McGraw-Hill, 1976.

[48] P.-A. Absila and K. Kurdyka, “On the stable equilibrium points of gradient systems,” *Systems & Control Letters*, vol. 55, pp. 573–577, 2006.

[49] G. D. Birkhoff, *Dynamical systems*. American Mathematical Society: New York, 1927.

[50] H. D. Nguyen and F. Forbes, “Global implicit function theorems and the online expectation–maximisation algorithm,” *Australian & New Zealand Journal of Statistics*, vol. 64, no. 2, pp. 255–281, 2022.

[51] A. V. Arutyunov and S. E. Zhukovskiy, “Application of methods of ordinary differential equations to global inverse function theorems,” *Differential equations*, vol. 55, pp. 437–448, 2019.

[52] W. Lin and C. I. Byrnes, “Design of discrete-time nonlinear control systems via smooth feedback,” *IEEE Transactions on Automatic Control*, vol. 39, no. 11, pp. 2340–2346, 1994.

[53] R. Hermann and A. Krener, “Nonlinear controllability and observability,” *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

[54] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[55] D. Nesic, A. R. Teel, and D. Carnevale, “Explicit computation of the sampling period in emulation of controllers for nonlinear sampled-data systems,” *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 619–624, 2009.

[56] D. Nešić, A. R. Teel, and E. D. Sontag, “Formulas relating KL stability estimates of discrete-time and sampled-data nonlinear systems,” *Systems & Control Letters*, vol. 38, no. 1, pp. 49–60, 1999.

[57] P. Bogacki and L. F. Shampine, “An efficient runge-kutta (4, 5) pair,” *Computers & Mathematics with Applications*, vol. 32, no. 6, pp. 15–28, 1996.

[58] M. Patriksson, *The traffic assignment problem: models and methods*. Mineola, NY: Dover Publications, 2015.

[59] S. Wiggins, *Ordinary Differential Equations: A Dynamical Point of View*. World Scientific, 2023.

[60] M. C. Irwin, *Smooth dynamical systems*. World Scientific, 2001, vol. 17.