Review article

Optimization algorithms as robust feedback controllers

Adrian Hauswirth, Zhiyu He, Saverio Bolognani *, Gabriela Hug, Florian Dörfler

Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland

ARTICLE INFO

ABSTRACT

Mathematical optimization is one of the cornerstones of modern engineering research and practice. Yet, throughout all application domains, mathematical optimization is, for the most part, considered to be a numerical discipline. Optimization problems are formulated to be solved numerically with specific algorithms running on microprocessors. An emerging alternative is to view optimization algorithms as dynamical systems. Besides being insightful in itself, this perspective liberates optimization methods from specific numerical and algorithmic aspects and opens up new possibilities to endow complex real-world systems with sophisticated self-optimizing behavior. Towards this goal, it is necessary to understand how numerical optimization algorithms can be converted into feedback controllers to enable robust "closed-loop optimization". In this article, we focus on recent control designs under the name of "feedback-based optimization" which implement optimization algorithms directly in closed loop with physical systems. In addition to a brief overview of selected continuous-time dynamical systems for optimization, our particular emphasis in this survey lies on closed-loop stability as well as the robust enforcement of physical and operational constraints in closed-loop implementations. To bypass accessing partial model information of physical systems, we further elaborate on fully data-driven and model-free operations. We highlight an emerging application in autonomous reserve dispatch in power systems, where the theory has transitioned to practice by now. We also provide short expository reviews of pioneering applications in communication networks and electricity grids, as well as related research streams, including extremum seeking and pertinent methods from model predictive and process control, to facilitate high-level comparisons with the main topic of this survey.

1. Introduction

Most advances in mathematical optimization in the past decades have been geared towards numerical implementations of iterative algorithms. The common viewpoint is that an optimization problem can be formulated, transformed, reduced, and relaxed, but ultimately the necessary steps to solve the problem rely purely on numerical linear algebra, which can be implemented and run on microprocessors. This offline computed solution is then used to reach and realize some decision.

This paradigm of optimization as a computational problem is almost synonymous with the field of management science and operations research (Bertsimas & Freund, 2004; Dantzig, 1998; Hillier & Lieberman, 2001), which has flourished ever since the inception of linear programming in the mid-20th century. Today, this kind of offline optimization is applied in various disciplines ranging from econometrics over statistical and machine learning (Bishop, 2009) to optimal control (Bertsekas, 2017).

From a control perspective, solving an optimization problem offline (with known data) and implementing its output as a decision is a *feedforward* approach. More precisely, the solution of an optimization problem is used as a set-point for a physical system, and feedback controllers are merely required to steer the physical system to this pre-computed state.[1] However, due to discrepancies between the model used for set-point optimization and the real system, the realized system state is generally suboptimal. Since set-point optimization happens offline and without recourse to real-time measurements, this traditional separation may be interpreted as an *open-loop* setup.

In contrast, in this article, we review a *feedback* approach to constrained nonlinear optimization that drives a physical system towards an optimal steady state by using real-time measurements in the set-point optimization itself. More concretely, we review ideas that consider optimization algorithms as dynamical systems and cast them as closed feedback loops. Equivalently, these methods may be interpreted

---

* Corresponding author.
  E-mail address: bsaverio@ethz.ch (S. Bolognani).

[1] Beyond computing optimal setpoints, optimization has many more uses in control, such as the optimal design and tuning of controllers, but those topics are not the subject of this article.

as "optimization algorithms that incorporate real-time measurements" or as "feedback controllers that mimic optimization algorithms".[2]

The study of feedback-based optimization of a domain-independent concept is relatively recent, although it is historically rooted in applications such as congestion control in communication networks (Low & Lapsley, 1999) and frequency control in power systems (Li et al., 2016; Molzahn et al., 2017). In these contexts, existing control systems have been interpreted from an optimization perspective that allows for improved redesigns and retrofitting. For educational reasons, we review these two applications in Appendix A, which have inspired much of the presented theory in this article.

On a more abstract level, this type of *closed-loop optimization* has been pursued mainly for three reasons:

(i) to approximately solve problems with inaccurate data and time-varying parameters,
(ii) to achieve constraint satisfaction with minimal model-dependence, and
(iii) to eliminate the need for exogenous setpoints and reference signals, especially in time-varying systems.

These reasons resonate well with the general feedback and feedforward paradigms advocated in control textbooks (Doyle et al., 2009; Franklin et al., 2010), and we further dwell on them hereafter.

Inaccurate problem data. In practice, optimization problems often lack precise data. Parameters and states based on measurements and statistical inference are inherently inaccurate, and so is the solution of an optimization problem based on such data. The earliest attempts at addressing this issue have resulted in sensitivity analysis for optimization problems (Shapiro, 1988) which asks how a solution changes as problem parameters vary. From a more practical perspective, robust optimization (Ben-Tal et al., 2009) and stochastic programming (Bonnans, 2019) offer ways to incorporate uncertainty in the problem. However, these approaches (which are primarily used offline and ahead of time) are conservative in the sense that they need to take into account the full set of possible problem instances even though, eventually, only one scenario will be realized. Often, they also entail massive computational costs. In contrast, by optimizing a system in real-time and using feedback, one needs to react only to the actual realization of the underlying disturbance process.

Constraint satisfaction. Beyond the mitigation of the effect of modeling inaccuracies on the accuracy of the solution, closed-loop setups render constraint satisfaction partially *model-free*. A physical system naturally defines and enforces a set of constraints. These constraints may be of a different nature. For instance, *input saturation* may be due to limited actuator capabilities or to the actions of a low-level controller. The violation of *hard physical limits* may trigger the immediate instability, failure, or destruction of the entire system, hence these constraints need to be satisfied at all times. For offline optimization, these limits need to be accurately modeled. In a feedback approach, on the other hand, the closed-loop optimization algorithm automatically operates on a reduced set, e.g., a manifold, defined by the physical constraints (e.g., input–output relations, saturation limits, and hard physical limits). Other constraints (e.g. thermal limits of physical components) are often fairly benign since they can be violated temporarily, but should be satisfied in the long run. These constraints are not enforced automatically and need to be addressed with proper control design. In a

feedback setting, measurements provide unequivocal information about the violation of such constraints, and this information can be exploited during runtime.

Autonomous operation. Finally, compared to standard feedback loops, closed-loop optimization setups can run without external setpoints or references. Instead, an economic objective can be directly optimized as long as a cost function can be specified and evaluated. This feature is particularly powerful in combination with the inherent constraint enforcement: whereas in classical control setups pre-computed setpoints have to be feasible (e.g., lie within actuator limits and be compatible with the physics of the plant), convergence to a feasible and optimal state is a defining aspect of feedback-based optimization. Additionally, in the presence of time-varying problem parameters that are hard to estimate and forecast, feedback-based optimization automatically results in the closed-loop system tracking the solution of the time-varying optimization problem. Furthermore, closed-loop setups open the possibility of optimizing a physical system in a *data-driven* and *model-free* manner. The input–output data collected in closed loop offers a rich source for representing, learning, and modulating the behaviors of a system.

In the remainder of this introduction, we discuss a simple, yet insightful, academic example of feedback-based optimization to offer a more concrete perspective on the points above and the technical issues that have been addressed by the research community in recent years. In particular, we illustrate a distinctive challenge in feedback-based optimization, which is the certification of the stability of the closed-loop interconnection between an optimization algorithm and the dynamical system. This topic has been the subject of substantial recent work which we review later in this article.

For the sake of a concise presentation, in our pedagogical examples and in the tutorial sections of this article, we restrict ourselves to continuous-time systems. In many cases, analogous discrete-time or sampled-data results exist or can be derived similarly to the continuous-time case. Discrete-time models also raise questions about the efficient implementation of feedback-based optimization. Those issues often require domain-specific solutions which are not the main topic of this article.

1.1. Illustrative examples

The following idealized yet general example of a gradient system interconnected with a physical plant illustrates the idea at the center of this review and some of the associated challenges. Many of the assumptions made in the following are for illustrative purposes and are not needed in general.

Example 1.1. Consider a dynamic nonlinear plant

$$\dot{\zeta} = f(\zeta, u) \qquad y = g(\zeta) + d \,, \tag{1}$$

where ζ, u and y are the state, input, and output, and d denotes an additive disturbance. The vector field $f(\cdot, \cdot)$ and the map $g(\cdot)$ describe the process and output measurement, respectively.

We assume that, for any fixed u, the plant is asymptotically stable with fast-decaying transients such that, for every u, there exists a unique steady state $\hat{h}(u)$ such that $0 = f(\hat{h}(u), u)$. Consequently, there also exists a *steady-state map* $h(u) := g(\hat{h}(u))$. We assume h to be continuously differentiable in u.[3]

We wish to drive the system to a steady state that minimizes a cost $\Phi(y)$ which is a function of the plant output y, hence to the solution of

$$\min_{u,y} \quad \Phi(y)$$

---

[3] Existence of a (differentiable) steady-state map is generally shown using implicit function theorems (Dontchev & Rockafellar, 2014).
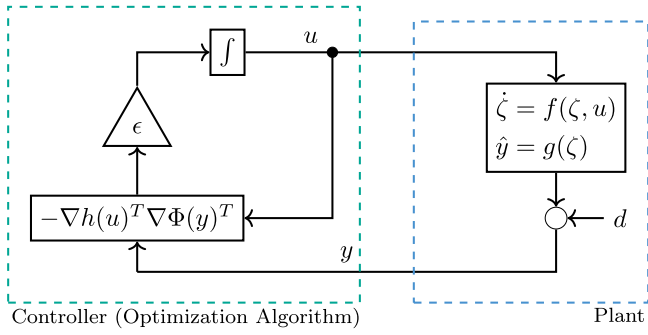
Fig. 1. Simple feedback-based gradient flow.

Fig. 2. Illustrations for Example 1.2 (top left: objective function; remaining panels: system trajectories for different control gains ϵ).

subject to $\quad y = h(u) + d$.

Given h and d, we may equivalently minimize the *reduced cost* $\tilde{\Phi}(u) := \Phi(h(u)+d)$ instead. For this purpose, we consider a simple gradient flow

$$\dot{u} = -\nabla\tilde{\Phi}(u)^T = -\nabla h(u)^T \nabla\Phi(h(u)+d)^T, \tag{2}$$

where $\nabla h(u)$ is due to the chain rule applied to $\Phi(h(u)+d)$.

The gradient flow (2) is a closed system (in the sense that it does not require an exogenous input or define an output). However, by identifying $h(u)+d$ as the measurable output y and u as the input, (2) can be easily transformed into an open system and interconnected with the plant (1), as shown in Fig. 1. This yields the closed-loop dynamics

$$\text{plant} \begin{cases} \dot{\zeta} &= f(\zeta, u) \\ y &= g(\zeta) + d \end{cases} \tag{3}$$

$$\text{controller} \begin{cases} \dot{u} &= -\epsilon \nabla h(u)^T \nabla\Phi(y)^T, \end{cases}$$

where $\epsilon > 0$ is a scalar control gain.

Crucially, the controller in (3) does not require explicit knowledge of h (nor of f or g). Instead, only the cost function gradient $\nabla\Phi(y)$ as well as steady-state input–output sensitivities $\nabla h(u)$ are required. Moreover, the additive disturbance d does not need to be known or explicitly estimated and is fully rejected, i.e., an equilibrium is a critical point of $\Phi(h(u)+d)$, independently of the value of d. Notice that this is partly possible because the disturbance is additive and therefore it does not affect the input–output sensitivity of the system. We will discuss how to deal with uncertain sensitivities in Section 3.3.

Thanks to the integral control structure of Eq. (3), it can be easily seen that any equilibrium point $(\zeta^\star, u^\star)$ of (3) is a steady state of the plant and satisfies $\nabla\tilde{\Phi}(u^\star)^T = \nabla h(u^\star)^T \nabla\Phi(h(u^\star)+d)^T = 0$. Therefore, $u^\star$ is a critical point of $\tilde{\Phi}$ (and a minimizer if $\tilde{\Phi}$ is convex).

The systems (1), (2), and (3) can be understood from a *singular perturbation* viewpoint (Khalil, 2002, Chap. 11): As $\epsilon \to 0^+$, the plant behavior is replaced by the algebraic map h, and the remaining dynamics (2) are the "slow" *reduced system*. Conversely, on a fast timescale, on which u and d can be assumed to be constant, the plant dynamics (1) are referred to as the "fast" *boundary-layer system*. ∎

One of the fundamental points left open by Example 1.1 is *closed-loop stability*. The idea that plant dynamics in Eq. (3) need to be fast-decaying relative to the controller is indeed crucial as the following numerical example shows.

Example 1.2. Consider the objective $\Phi(y) = (y^2 - 1)^2$ which is illustrated in the top left panel of Fig. 2 and has two isolated minima $\{-1, 1\}$. As plant (1), consider a single-input-single-output second-order plant governed by

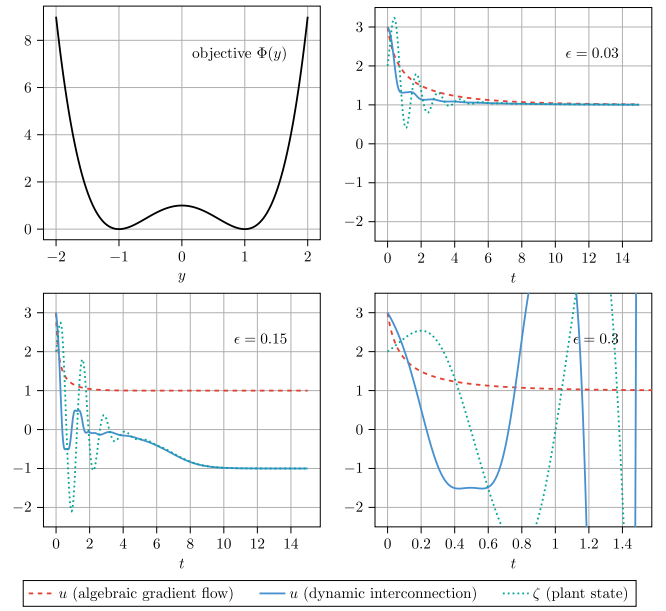$$\ddot{\zeta} + a\dot{\zeta} + b(\zeta - u) = 0$$

with $a = 2$, $b = 25$, and $y = \zeta$. There is no disturbance acting on the system. The plant is asymptotically stable, under-damped, and, at steady state, we have $y = \zeta = u$. Hence, the controller in (3) takes the form

$$\dot{u} = -\epsilon \nabla\Phi(y)^T = -4\epsilon y(y^2 - 1).$$

Fig. 2 shows trajectories of the closed-loop system Eq. (3) for the same initial condition, but different values of the gain ϵ, and comparing it to the "algebraic" gradient flow (2) given by $\dot{u} = -\epsilon\nabla\Phi(h(u))^T = -4\epsilon u(u^2 - 1)$.

We observe, that for the given initial condition the algebraic gradient trajectory converges to the minimizer at 1. In contrast, the trajectories of the closed-loop system (3) converge to either one of the two minimizers or diverge, depending on ϵ. In other words, closed-loop stability of (3) is not guaranteed, and even if it is, convergence may not be to the same minimizer as for (2). ∎

The issue of stability is central to many closed-loop optimization setups and work on this topic, including quantitative stability and robustness requirements, will be discussed in more detail in Section 3.1.

1.2. Organization

The remainder of this article is structured as follows: Section 2 provides a tutorial-like review of various optimization algorithms from a dynamical systems perspective. This point of view is crucial for closed-loop optimization. In Section 3, we then review recent works on constrained feedback-based optimization. We pay particular attention to different mechanisms to ensure closed-loop stability and to enforce constraints. We further clarify how to leverage data and exploration to facilitate model-free implementations in closed loop. Section 4 presents a case study on the timely problem of optimal autonomous redispatch in electricity grid. This problem requires solving a nonlinear, nonconvex AC optimal power flow problem in closed-loop with the physical power grid and has motivated many implementations and recent advances in closed-loop optimization. Finally, Section 5 concludes the article, highlights the unresolved problems in this domain, and presents exciting avenues for future research.

As mentioned before, in Appendix A we revisit some historical application examples from communication networks and power systems. In Appendix B, for further contextualization, we give a high-level description of other related approaches, such as *extremum seeking*, real-time iterations, and *modifier adaptation*, which perform optimization tasks at a system level within a closed-loop setup. We quickly discuss the main ideas of these well-established domains in order to provide historical context and a very broad overview of the adjacent research landscape.

2. Optimization algorithms as dynamical systems

In recent years, renewed attention has been paid to the fact that many numerical optimization algorithms, and specifically first-order iterative algorithms, can be interpreted as dynamical systems. In this section, we therefore provide an overview of different *optimization dynamics* and review the main properties of these flows. This perspective is essential to bridge the gap between algorithms and their implementation as feedback systems, which will be the focus of Section 3. Apart from continuous-time optimization dynamics, discrete-time methods (for optimization or games) are also amenable to closed-loop implementations in a sampled-data setting, see the feedback equilibrium seeking in Belgioioso et al. (2023, 2021, 2022).

2.1. Gradient flows

Arguably as old as differential calculus itself are methods that seek out (local) minima of a function $\Phi(x)$ over an unconstrained domain $x \in \mathbb{R}^n$ by following a descent direction. The most prototypical class of methods are gradient (or "steepest descent") schemes, e.g., $\dot{x} = -\nabla\Phi(x)$. Gradient methods exist in a wide variety of forms and contexts. Apart from $\mathbb{R}^n$, they can be defined on non-Euclidean manifolds (Absil et al., 2008; Brockett, 1988; Helmke & Moore, 1996; Zehnder, 2010) and on infinite-dimensional spaces (Ambrosio et al., 2005; Luenberger, 1969). In the absence of smoothness they can also be generalized to subgradient methods (Beck, 2017; Clarke, 1990). On top of that, steepest descent methods are generally available as continuous-time flows, discrete-time algorithms, or stochastic processes (Borkar, 2008). For ease of exposition and in line with the rest of this article, we will limit ourselves to continuous-time deterministic gradient flows.

At least for continuous-time negative gradient flows, convergence to minimizers appears plausible, if not tautological. However, closer inspection reveals important technical details, summarized in the following theorem which combines classical results available in most textbooks on calculus, dynamical systems, or optimization.

Theorem 1. *Let $\Phi : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable with locally Lipschitz derivative $\nabla\Phi$ such that, for some $c \in \mathbb{R}$, the sublevel set $\Phi^{-1}(c) = \{x \in \mathbb{R}^n \mid \Phi(x) \leq c\}$ is compact. Then, the following statements hold for the gradient flow $\dot{x} = -\nabla\Phi(x)^T$:*

(i) Trajectories starting in $\Phi^{-1}(c)$ converge to the set of critical points, i.e., points $x^\star$ such that $\nabla\Phi(x^\star) = 0$.

(ii) If Φ is analytic or convex, then every solution starting in $\Phi^{-1}(c)$ converges to a single point. If Φ is strongly convex or satisfies the Polyak-Lojasiewicz inequality, convergence is exponential.

(iii) Every asymptotically stable equilibrium is a strict and isolated minimizer, and every local minimizer is stable. If Φ is analytic or convex, then the set of all (strict) minimizers is equivalent to the set of (asymptotically) stable equilibria.

(iv) If Φ is twice continuously differentiable, the stability of a critical point is partially governed by its Hessian: if $\nabla^2\Phi(x^\star)$ is positive definite, $x^\star$ is a local minimizer and locally exponentially stable. If $\nabla^2\Phi(x^\star)$ has at least one negative eigenvalue, $x^\star$ is unstable.

In the absence of local Lipschitz continuity of $\nabla\Phi$, uniqueness of trajectories is not guaranteed and a distinction between *weak* and *strong equilibria* has to be made (Cortés, 2008; Hauswirth, Bolognani, & Dörfler, 2020).

Point (i) in Theorem 1 can be seen as a consequence of LaSalle's invariance theorem (Khalil, 2002). Namely, it is immediate that Φ is non-increasing along gradient trajectories. Compactness of $\Phi^{-1}(c)$ is sufficient to preclude unbounded trajectories that escape to the horizon. Further, compactness of $\Phi^{-1}(c)$ together with the continuity of Φ is sufficient to guarantee lower boundedness and thus the existence of a minimizer.

In general, however, trajectories may not converge to a single point but to the entire set of critical points (Palis & De Melo, 1982). This pathological behavior is ruled out in (ii) if Φ is analytic, which guarantees the finite length of trajectories due to Lojasiewicz's inequality (Absil et al., 2005), or if Φ is convex. The conditions that ensure exponential convergence are studied in Bianchin, Poveda, and Dall'Anese (2022), Frasconi et al. (2016).

As indicated in (iii), the stability of equilibria is related to their optimality, but an equivalence between the two requires additional assumptions. See Absil and Kurdyka (2006) for a proof and counterexamples. Finally, (iv) follows since, if Φ is smooth enough, the stability of an equilibrium $x^\star$ can be analyzed by investigating the linearized dynamics $\dot{x} = -\nabla^2\Phi(x^\star)x$.

Example 2.1 (*Variable Metric and Newton Gradient Flow*). A straightforward degree of freedom for gradient flows is the use of a *metric* $Q(\cdot)$ that maps every point $x \in \mathbb{R}^n$ to a square symmetric positive-definite matrix $Q(x)$. Under minor technical conditions (e.g., that Q has a uniformly bounded condition number, see Hauswirth, Bolognani, and Dörfler 2020) trajectories of the generalized gradient flow

$$\dot{x} = -Q(x)\nabla\Phi(x)^T \tag{4}$$

converge to the set of critical points of Φ. Namely, the use of Q, does neither alter the equilibrium points nor their attractivity and stability, but only the trajectories. This feature is illustrated in Fig. 3 which shows gradient trajectories for the same nonconvex potential function but for different metrics.

If Q is constant, (4) is equivalent to a Euclidean gradient flow in linearly transformed coordinates. Namely, if $Q = V^T \Lambda V$ is the eigenvalue decomposition of Q, we can define the coordinate transformation $y := \sqrt{\Lambda}^{-1} V x$ where $\sqrt{\Lambda}$ denotes the diagonal matrix of square root eigenvalues. Then, it can be easily shown that (4) is equivalent to $\dot{y} = -\nabla\hat{\Phi}(y)^T$ with $\hat{\Phi}(y) := \Phi(V^T \sqrt{\Lambda} y)$.

If Q is not constant, then, from a differential–geometric viewpoint, Q can be interpreted as a matrix representation of a non-Euclidean *Riemannian metric* on $\mathbb{R}^n$. Thus, Q endows $\mathbb{R}^n$ with a non-flat geometry (Lee, 1997).

As a special case of a non-constant metric, assume that Φ is twice differentiable and strongly convex. Then, Q can be chosen to be the inverse of the Hessian $\nabla^2\Phi$. This results in a continuous-time version of the classic *Newton method*, also referred to as "Newton gradient flow" (Jongen et al., 2001, Chap. 9.3). However, unlike the iterative Newton method, the continuous-time flow does not exhibit an inherently faster convergence rate compared to other gradient flows. For the inverted Hessian metric, convergence is *isotropic*, i.e., the same from all directions. This property counteracts ill-conditioning of the objective function as illustrated in Fig. 4.

Finally, if Q is sparse, then the *sparsity* pattern induces algebraic structure that can often be exploited for the purpose of a distributed implementation (Bullo, 2022). ∎

2.2. Projected gradient flows

When unilateral (i.e., inequality) constraints restrict the search domain of the optimization problem, it is often possible to rely on
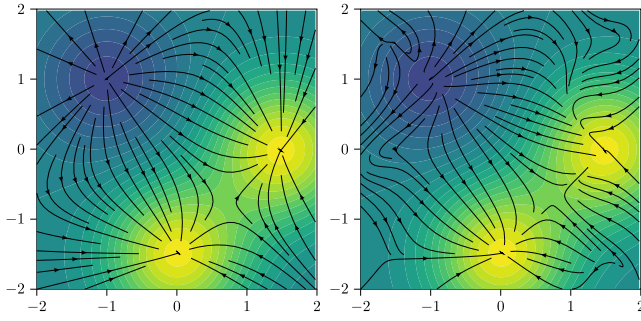
Fig. 3. Gradient trajectories for a non-convex objective function. Trajectories under the Euclidean metric (left) and a generic variable metric (right) differ significantly. The critical points (i.e., the minima, maximum, and saddle-point) and their stability properties are unaffected by the choice of metric.
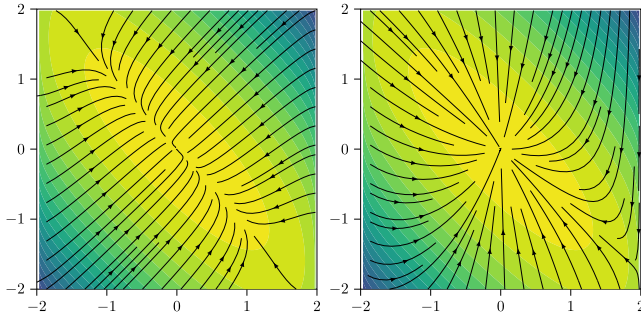

Fig. 4. Gradient trajectories for strongly convex, but ill-conditioned objective. The trajectories under the Euclidean metric (left) quickly approach a subspace on which the objective is almost flat, and then converge only slowly to the global optimizer. Trajectories under the "Newton metric" (right), approach the global optimizer isotropically, unaffected by the ill-conditioning of the objective.

gradient flows by introducing suitable projection mechanisms. In a computational context, this is particularly true if the projection onto a given constraint set is easy to evaluate numerically.

Consider the constrained optimization problem

$$\text{minimize} \quad \Phi(x) \qquad \text{subject to} \quad x \in \mathcal{X}, \qquad (5)$$

where $\mathcal{X} \subset \mathbb{R}^n$ is closed convex and non-empty, and Φ is continuously differentiable. The classical *projected gradient descent* iteration to solve this problem takes the form

$$x^{k+1} = P_{\mathcal{X}} \left(x^k - \alpha^k \nabla \Phi(x^k)^T \right), \qquad (6)$$

where $P_{\mathcal{X}(y)} := \arg\min_{x \in \mathcal{X}} \|x - y\|$ denotes the Euclidean minimum norm projection onto $\mathcal{X}$, and $\{\alpha^k\}$ is a sequence of step sizes.

By choosing infinitesimally small step-sizes, the continuous-time limit of (6) is a *projected gradient flow*

$$\dot{x} = \Pi_{\mathcal{X}} \left[-\nabla \Phi(x)^T \right] (x), \qquad (7)$$

where $\Pi_{\mathcal{X}[v]}(x)$ denotes the projection of the vector v onto the tangent cone $T_{\mathcal{X}(x)}$ of $\mathcal{X}$ at x, i.e., $\Pi_{\mathcal{X}[v]}(x) = \arg\min_{w \in T_{\mathcal{X}(x)}} \|v - w\|$. As illustrated in Fig. 5, the tangent cone is the set of all directions starting at x which point inward into the set $\mathcal{X}$. For convex sets, it takes the general form $T_{\mathcal{X}(x)} = \text{cl}\{d \mid d = \alpha(y - x), y \in \mathcal{X}, \alpha \geq 0\}$.

In contrast to the discrete-time system (6), continuous-time projected gradient flows can be generalized extensively. In particular, convexity of $\mathcal{X}$ is not required. More precisely, whereas $P_{\mathcal{X}}$ requires $\mathcal{X}$ to be convex to be well-defined, $\Pi_{\mathcal{X}}$ is generally well-defined since $T_{\mathcal{X}(x)}$ is non-empty and closed convex for a large class of non-convex sets called *(Clarke) regular* (see Rockafellar and Wets 2009).

Fig. 6(e) illustrates the qualitative behavior of projected gradient flows. Namely, in the interior of the feasible set, trajectories follow
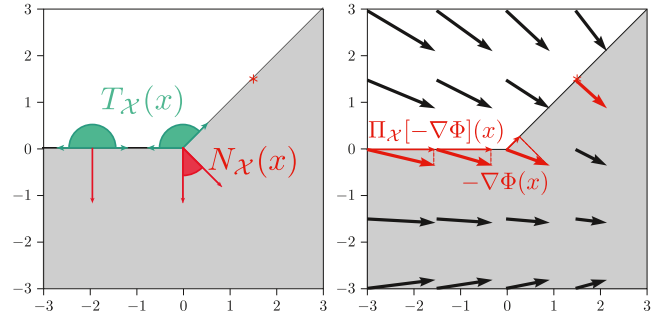

Fig. 5. Left: Examples of tangent and normal cones of a set $\mathcal{X}$ (gray area is infeasible). Right: Projected vector field onto the tangent cone. See Fig. 6(e) for the resulting projected gradient trajectory.

the gradient direction, whereas at the boundary, trajectories follow the steepest *feasible* descent direction. Projected gradient flows are therefore inherently discontinuous systems, and their study requires tools from non-smooth analysis (Aubin & Cellina, 1984; Hauswirth, Bolognani, & Dörfler, 2020). It is worth remarking that projected gradient flows can be modified in order to recover continuity (Xia & Wang, 2000) and in order to ensure that the flow is well defined outside of the feasible region (Muehlebach & Jordan, 2022). These properties often come with additional convexity assumptions. An exception is the modification presented in Allibhoy and Cortés (2023), where control barrier function theory is employed to achieve both these properties in a non-convex setting. Another strategy that handles non-convexity is to leverage anti-windup schemes to smoothly and robustly approximate projected gradient flows (Hauswirth et al., 2020a, 2020b, 2020c), see Example 3.3.

Projected gradient flows extend properties from their unconstrained counterparts. For instance, similarly to Theorem 1, trajectories of (7) converge to the set of critical points (in this case, Karush-Kuhn–Tucker points of (5)), and stability and optimality can be related analogously to (iii) in Theorem 1 (Clarke, 1990; Hauswirth, Bolognani, & Dörfler, 2020).

Remark 2.1 (*Discretization & Proximal-Point Algorithm*). As mentioned above, the continuous-time model (7) can be obtained from (6) in the limit as $\alpha \to 0^+$ where α is a constant step size. Conversely, (6) can be interpreted as a *forward Euler* discretization of (7). Analogously, a *backward Euler* scheme for (7) yields the implicit form $x^{k+1} = P_{\mathcal{X}} \left(x^k - \alpha^k \nabla \Phi(x^{k+1})^T \right)$. Comparing optimality conditions, it is easy to show that this discretization is a special case of the more general *proximal-point algorithm* (Beck, 2017, Chap. 27.1). ∎

Remark 2.2 (*Non-differentiable objective & subgradients*). Convex projected gradient flows also fall into the category of *subgradient flows*: if $\mathcal{X}$ is convex, we may consider the minimization of $\Phi(x) + I_{\mathcal{X}(x)}$ where $I_{\mathcal{X}}$ denotes the indicator function of the set $\mathcal{X}$. Since $I_{\mathcal{X}}$ is not differentiable, instead of a gradient flow, we need to resort to the *subgradient inclusion* $\dot{x} \in -\nabla \Phi(x)^T - N_{\mathcal{X}(x)}$. (In particular, the subgradient of $I_{\mathcal{X}}$ is given by the normal cone $N_{\mathcal{X}(x)}$ of $\mathcal{X}$ at x, i.e., $\partial I_{\mathcal{X}(x)} = N_{\mathcal{X}(x)}$.) This differential inclusion is also referred to as *differential variational inequality* (Aubin & Cellina, 1984) and can be shown to be equivalent to (7) (i.e., admit the same trajectories). ∎

Remark 2.3 (*Variable Metric and Projected Newton Gradient Flow*). Analogously to unconstrained gradient flows discussed in Example 2.1, projected gradient flows can also be defined using a variable metric $Q(\cdot)$. This degree of freedom allows, in particular, the definition of *projected Newton gradient flows*. This modification, however, requires a generalization of $\Pi_{\mathcal{X}}$ in (7) to take into account the effects of the

metric (see Hauswirth, Bolognani, and Dörfler 2020). Namely, the variable-metric projected gradient flow takes the form

$$\dot{x} = \Pi_{\mathcal{X}}^{Q}\left[-Q(x)\nabla\Phi(x)^T\right],$$

where the "oblique" projection operator

$$\Pi_{\mathcal{X}}^{Q}[v](x) := \arg\min_{w \in T_{\mathcal{X}}(x)} (w - v)^T Q(x)(w - v) \qquad (8)$$

has to be employed. ∎

2.3. Primal-dual saddle-point flows

Simply speaking, under weak technical assumptions, solutions of a constrained optimization problem are saddle-points of the associated *Lagrangian*. For this reason, dynamical systems that seek out saddle-points rather than extrema of a function are of particular interest for constrained optimization.

As a general yet basic setup, consider a differentiable function $L(x, \mu)$ that is convex in x for every μ, concave in μ for all x, and either strictly convex in x or strictly concave in μ. Then trajectories of the system

$$\dot{x} = -\nabla_x L(x, \mu)^T \qquad \dot{\mu} = \nabla_\mu L(x, \mu)^T \qquad (9)$$

converge to a saddle-point of L, i.e., a point $(x^\star, \mu^\star)$ such that $L(x, \mu^\star) \geq L(x^\star, \mu^\star) \geq L(x^\star, \mu)$ for all x and all μ. In particular, Eq. (9) consists of a gradient descent in the *primal* variables x and a gradient ascent in the *dual* variables μ.

Historically, *saddle-point flows* have primarily been considered in the context of nonlinear circuit analysis (Brayton & Moser, 1964; Smale, 1972), but their potential for optimization has been observed even before that Arrow et al. (1958), Kose (1956). Although saddle-point flows have been studied throughout the years (Bloch et al., 1992; Venets, 1985), they have recently become a topic of intense interest due their importance for distributed network optimization (see Cortés and Niederländer 2019, Feijer and Paganini 2010 and the example in Appendix A.1 on control of communication networks).

Example 2.2 (*Saddle-Point Flow for Optimization with Equality Constraints*). Consider the following linearly constrained problem

$$\text{minimize} \quad \Phi(x) \qquad \text{subject to} \quad Ax = b \qquad (10)$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, and Φ is strictly convex. The Lagrangian of (10), given by

$$L(x, \mu) := \Phi(x) + \mu^T(Ax - b)$$

is strictly convex in x (since Φ is strictly convex) and linear (thus concave) in μ. Consequently, the saddle-point flow (9) is globally convergent and takes the form

$$\dot{x} = -\nabla\Phi(x)^T - A^T\mu \qquad \dot{\mu} = Ax - b. \qquad (11)$$

Notice that any equilibrium $(x^\star, \mu^\star)$ of (11) satisfies $0 = \nabla\Phi(x^\star)^T + A^T\mu^\star$ and $0 = Ax^\star - b$ and thus the Karush-Kuhn–Tucker condition which are (for convex problems) necessary and sufficient for optimality of $x^\star$. ∎

Like gradient flows, saddle-point flows can be modified through the use of positive definite metrics Q_p, Q_d as

$$\dot{x} = -Q_p\nabla_x L(x, \mu)^T \qquad \dot{\mu} = Q_d\nabla_\mu L(x, \mu)^T. \qquad (12)$$

Often, Q_p and Q_d are chosen to be constant diagonal matrices that speed up or slow down convergence in specific directions. Interestingly, in the limit case $Q_p \gg Q_d$, one recovers a differential–algebraic system

$$x^\star(\mu) = \arg\min_x L(x, \mu) \qquad \dot{\mu} = Q_d\nabla_\mu L(x^\star(\mu), \mu)^T,$$

which is a continuous-time *dual ascent*. This idea of replacing the primal gradient flow with an explicit (algebraic) minimization can also be applied partially to a subset of variables. This technique was applied, e.g., in Li et al. (2016) and is illustrated in the example in Appendix A.2 for frequency control in power systems.

Convergence proofs for (9) and its generalization usually rely on the monotonicity of the vector field $[-\nabla_x L(x, \mu) \ \nabla_\mu L(x, \mu)]$. This type of argument, however, does not generalize beyond convex-concave saddle-point flows (see the discussion in Cherukuri, Gharesifard, and Cortes 2017). Even relaxing strict convexity or concavity requirement of either x or μ, respectively, is problematic since this may lead to oscillations (Holding & Lestas, 2014). We will see in Example 2.3 that augmentation of the Lagrangian is instrumental in recovering stability in these cases.

Differentiability of the saddle-function L is not generally required. In fact, (9) can be formulated in terms of subgradients if L is not differentiable (Goebel, 2017; Venets, 1985). In particular, (9) can be generalized to include projections on both x and/or μ (Cherukuri, Gharesifard, & Cortes, 2017; Cherukuri et al., 2016; Cherukuri, Mallada, Low, & Cortés, 2017; Hauswirth, Ortmann, Bolognani, & Dörfler, 2020; Stegink et al., 2018). This possibility is particularly important to deal with inequality constraints, as the following example shows.

Example 2.3 (*Projected Saddle-Point Flow for Optimization with Inequality Constraints*). We revisit Example 2.2 to show how projected saddle-point flows can be used to deal with inequality constraints and to discuss augmentations based on penalty terms to improve convergence. Instead of (10), consider the problem

$$\begin{aligned} \text{minimize} \quad & \Phi(x) \\ \text{subject to} \quad & x \in \mathcal{X} \\ & g(x) \leq 0, \end{aligned} \qquad (13)$$

where Φ and g are convex (but not necessarily strictly convex) and continuously differentiable. Further, let $\mathcal{X} \subset \mathbb{R}^n$ be non-empty and closed convex. We define the *partial* Lagrangian $L : \mathcal{X} \times \mathbb{R}_{\geq 0}^m \to \mathbb{R}$ of (13) as

$$L(x, \mu) := \Phi(x) + \mu^T g(x), \qquad (14)$$

and note that μ must lie in the non-negative orthant $\mathbb{R}_{\geq 0}^m$ because it is associated with an inequality constraint.

To find a saddle-point of L on the set $\mathcal{X} \times \mathbb{R}_{\geq 0}^m$ we use the continuous-time projection formalism introduced in Section 2.2 for projected gradient flows. Namely, we consider the *projected saddle-point flow*

$$\begin{aligned} \dot{x} &= \Pi_{\mathcal{X}}\big[\ \overbrace{-\nabla_x L(x, \mu)^T}^{-\nabla\Phi(x)^T - \nabla g(x)^T\mu}\ \big](x) \\ \dot{\mu} &= \Pi_{\mathbb{R}_{\geq 0}^m}\big[\underbrace{\nabla_\mu L(x, \mu)^T}_{g(x)}\big](\mu) \end{aligned} \qquad (15)$$

where $\Pi_{\mathcal{X}}[w](x)$ and $\Pi_{\mathbb{R}_{\geq 0}^m}[v](x)$ project w and v onto the tangent cone of $\mathcal{X}$ and on the non-negative orthant $\mathbb{R}_{\geq 0}^m$ at x and μ, respectively. Consequently, trajectories of (15) cannot leave $\mathcal{X} \times \mathbb{R}_{\geq 0}^m$.

Importantly, two different constraint enforcement mechanisms are at play: on one hand, the constraint $x \in \mathcal{X}$ is enforced directly by projection, similarly to the projected gradient flow in Section 2.2; on the other hand, the constraint $g(x) \leq 0$ is enforced by dualization. Namely, the dual variable μ is updated in response to a constraint violation $g(x) > 0$ and converges to a dual solution of (13).

As shown, e.g., by Goebel (2017), under weak technical assumptions and if Φ is strictly convex, trajectories of (15) are guaranteed to converge to a Karush-Kuhn–Tucker point (and thereby to a global optimizer) of (13).

If Φ is non-strictly convex, then trajectories of (15) converge to the optimizer of (13) if, instead of L, a penalty augmented Lagrangian of the form

$$L^a(x, \mu) := \Phi(x) + \mu^T g(x) + \phi(x) \qquad (16)$$

is used, where ϕ is a penalty function. In practice, ϕ is often chosen to be of the form $\phi(x) := \frac{\rho}{2}\|\max\{g(x), 0\}\|^2$, but other constructions are possible to guarantee asymptotic convergence (Hauswirth, Ortmann, Bolognani, & Dörfler, 2020).

If Φ or g are non-convex, convergence of trajectories to a saddle point is generally not guaranteed. For this reason, Bernstein et al. (2019), Dall'Anese and Simonetto (2018), Tang et al. (2018, 2022) have used an additional regularization on the dual variables of the form

$$L^b(x, \mu) := \Phi(x) + \mu^T g(x) + \phi(x) - \frac{\hat{\rho}}{2}\|\mu\|^2, \tag{17}$$

which renders L^b strongly concave in μ. Using this modification and a large enough $\hat{\rho}$, the corresponding saddle-point flow is convergent, even with exponential stability guarantees. However, saddle-points of L^b do generally not coincide with Karush-Kuhn–Tucker points of (13) anymore (unless the KKT point lies in the interior of the feasible set, in which case μ is zero and the dual regularization term disappears, see Bianchin, Cortés, Poveda and and Dall'Anese 2022). In particular, notice that the dual update takes the form $\dot{\mu} = \Pi_{\mathbb{R}^n_{\geq 0}}[g(x) + \hat{\rho}\mu]$ and thus, at an equilibrium, it holds that $\mu_i^\star(g_i(x^\star) + \hat{\rho}\mu_i^\star) = 0$. In the absence of dual regularization, the same equilibrium condition simplifies to $\mu_i^\star g_i(x^\star) = 0$ which is the *complementary slackness* part of the KKT optimality conditions of (13). The error between the saddle points of the augmented Lagrangian and the solution of the constrained optimization problem can be bounded as a function of the augmentation term (Koshal et al., 2011). ■

2.4. Comparison of constraint-enforcement mechanisms

In the previous subsections we have presented several flows that correspond to optimization dynamics. Here, we compare and contrast the behavior of these different mechanisms based on how they handle constraints that define the feasible region of the optimization problem, as this is a crucial aspect in online and closed-loop applications.

Gradient flows, as defined in Section 2.1, do not naturally allow for unilateral (i.e., inequality) constraints. A relatively easy and widely applicable way to incorporate them is the addition of *penalty* or *barrier* terms to the objective. However, both approaches by themselves can enforce constraints only approximately.

More precisely, for a constraint of the form $g(x) \leq 0$ where $g : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable, a common penalty function is the squared 2-norm of the constraint violation vector, i.e., $\phi(x) = \frac{\rho}{2}\|\max\{g(x), 0\}\|^2$ where $\rho > 0$ denotes a scaling parameter. Many variations, including different norms on constraint violations are possible. The common feature of penalty function lies in the fact that they technically allow for constraint violations and thus produce an "outer" approximation of the feasible region, i.e., minimizers of a penalty-augmented cost function $\Phi(x) + \phi(x)$ do not generally satisfy $g(x) \leq 0$.[4] This is illustrated in Fig. 6(a).

Barrier functions, on the other hand, can be used to enforce constraints strictly, i.e., without allowing for any violation. For this purpose, a barrier function $\psi(\cdot)$ for the constraint $g(x) \leq 0$ needs to be such that for $x \rightarrow x^\star$ with $g(x^\star) = 0$ we have $\psi(x) \rightarrow \infty$. A common example satisfying this condition are negative log-barriers of the form $\psi(x) = -\frac{1}{\mu}\log(g(x))$ which are important for interior-point methods for constrained convex programming (Nesterov & Nemirovskii, 1994). Barrier functions therefore achieve an "inner" (i.e., conservative) approximation of the feasible region, as illustrated in Fig. 6(b).

Both approaches are widely applicable (under minor technical assumptions) and do not require convexity of the constraints. In general,

---

[4] A notable exception are so-called *exact penalty methods* that transform a constrained optimization problem into an unconstrained one without changing the location of minimizers, albeit at the expense of smoothness or other technical drawbacks (Di Pillo & Grippo, 1989).
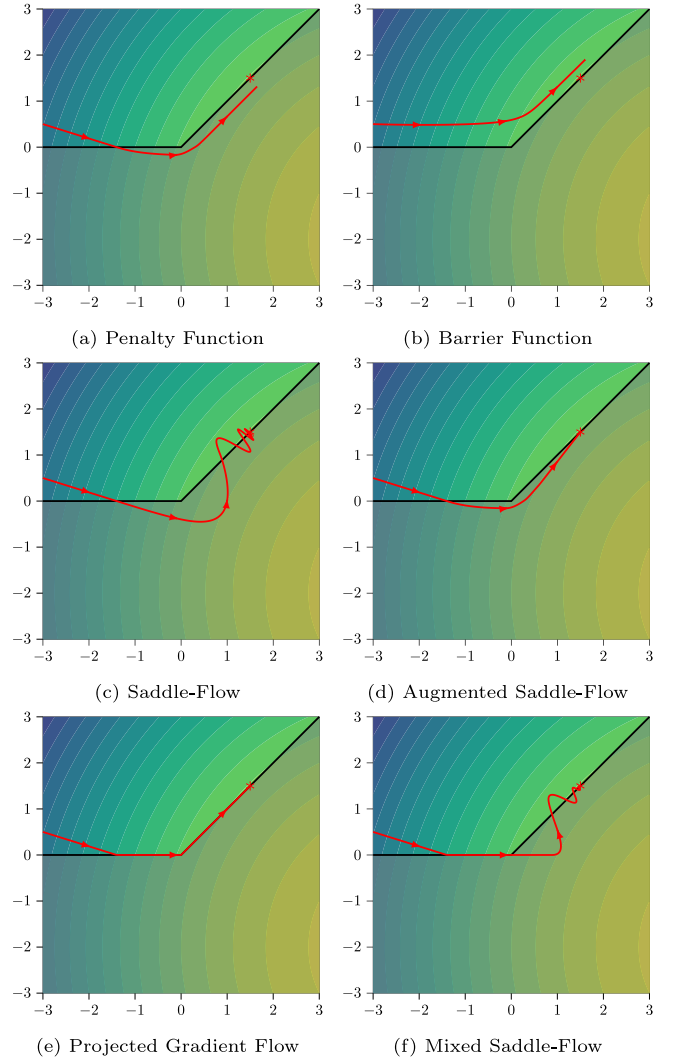

(a) Penalty Function   (b) Barrier Function

(c) Saddle-Flow   (d) Augmented Saddle-Flow

(e) Projected Gradient Flow   (f) Mixed Saddle-Flow

Fig. 6. Behavior of different constraint enforcement mechanisms All panels show the minimization of a quadratic function subject to two constraints $x_2 \geq 0$ and $x_2 \geq x_1$ (the grayed out area is infeasible). Penalty (a) and barrier (b) functions allow for smooth outer and inner approximations of constraints with an unconstrained gradient flow. Saddle-point flows (c) enforce constraints only asymptotically by integrating constraint violation over time. Augmenting saddle-point flows with a penalty term can improve convergence (d) as explained in Example 2.3. Projected gradient flows (e) enforce constraints directly by projection, which results in non-smooth trajectories. Individual constraints can also be enforced with a combination of these mechanisms, e.g., as in (f) with a projection for $x_2 \geq 0$ and dualization (saddle-flow) for $x_2 \geq x_1$, as in Example 2.3.

steeper penalties or barriers (i.e., $\rho \rightarrow \infty$ or $\mu \rightarrow \infty$) will lead to more precise results. However, excessively steep augmentations can lead to stability issues when implemented numerically or in closed loop with a dynamical system (see Section 3.1).

Constraint enforcement by (infinitesimal) projection, as for continuous-time projected gradient flows presented in Section 2.2, is mathematically well-posed and works in very general settings. Furthermore, constraints are represented exactly and they are satisfied at all times (see Fig. 6(e)). However, we will see in Section 3.2 that producing these continuous-time projected flows on a physical system is not straightforward. In some instances, the projection behavior emerges naturally from physical saturation and anti-windup control. When the constraint is not enforced by the plant, careful design of the discrete-time implementation is required in order to maintain computational tractability and preserve the stability properties without introducing additional convexity assumptions.

Dualization of constraints leads to saddle-point flows where dual variables are computed by integrating the constraint violation over time. Hence, transient constraint violations are generally unavoidable, even in the presence of augmentation terms (see Figs. 6(c) and 6(d)). Tuning can be difficult, especially for nonlinear problems. Suboptimal parameter choices can lead to severely under- or over-damped transients that may venture far outside the feasible domain, which is undesirable in online and closed-loop applications. This problem gets only more challenging for high-dimensional and ill-conditioned problems.

In theory, each constraint (in functional form) can be enforced with one of these mechanisms independently of the other constraints. For example, in Fig. 6(f), the constraint $x_2 \geq 0$ is enforced by projection whereas $x_2 \geq x_1$ is dualized. As we will see in Section 3.2, all these constraint enforcement methods can be applied in an online feedback setup. This design freedom is particularly useful in control setups where the real-world nature of constraints can dictate the appropriate enforcement mechanism. For instance, barrier functions may be considered for constraints that may not be violated under any circumstances. Constraints that are naturally enforced by physical saturation, mechanical constraints or similar are best represented by projections. Dualization in combination with a penalty term is particularly helpful to enforce constraints asymptotically and often allows for distributed implementations.

2.5. Time-varying online optimization

A topic that is central to online optimization, in open or closed loop, is the study of problems that vary over time. In recent years, this topic has garnered significant interest because of its relevance to many applications in control, robotics, machine learning, and others (Dall'Anese et al., 2020; Hazan, 2022; Simonetto et al., 2020). The focus has been the development of algorithms that can track the solution of a non-stationary optimization problem with performance guarantees. Historically, two perspectives can be distinguished.

On the one hand, Besbes et al. (2015), Hall and Willett (2015), Jadbabaie et al. (2015), Lesage-Landry et al. (2020), Nonhoff and Müller (2020), Shi et al. (2020), Zinkevich (2003) and others frame time-varying optimization as an *iterative learning problem*: at every iteration k an agent chooses an action x_k and subsequently a convex function Φ_k is revealed. The agent's goal is to minimize her *regret*, i.e., some measure of accumulated suboptimality.

On the other hand, Fattahi et al. (2020), Rahili and Ren (2017), Rahili et al. (2015), Simonetto et al. (2017), Tang et al. (2018, 2022) are inspired more by control theory and describe time-varying optimization as a *tracking problem* whereby an optimization algorithm defines a time-varying solution map $t \mapsto x^\star(t)$ that needs to be followed as closely as possible by the online optimization scheme.

Both viewpoints share a common base.

First, additional assumptions need to be introduced in order to guarantee well-posedness of the tracking problem. In many cases, convexity is assumed to guarantee the existence of a unique minimum and, in the case of strong convexity, a unique minimizer. Exceptions are (Dontchev et al., 2013; Kungurtsev & Jäschke, 2017; Subotić et al., 2022; Suwartadi et al., 2017; Tang et al., 2018, 2022; Zavala & Anitescu, 2010), where results for non-convex optimization problems are presented.

Second, to give meaningful performance guarantees, some sort of "bounded variation" in the optimization problem has to be assumed. A common assumption is that the rate of change of the optimizers is bounded by a known constant (see for example Bernstein et al. 2019, Bianchin, Cortés, Poveda and and Dall'Anese 2022, Madden et al. 2021). This assumption is relaxed in Subotić et al. (2022) which shows how, in special cases, the rate of change of the optimizer can be bounded using information about the objective and the constraint functions only.

Depending on the application, the time-varying optimization problem might be unconstrained (Popkov, 2005), constrained to a stationary set (Hall & Willett, 2015; Jadbabaie et al., 2015; Mokhtari et al., 2016; Zinkevich, 2003), or have time-varying constraints (Bernstein et al., 2019; Bianchin, Cortés, Poveda & and Dall'Anese, 2022; Fazlyab et al., 2016; Fazlyab, Paternain, Preciado & and Ribeiro, 2018; Hauswirth et al., 2018; Rahili & Ren, 2017; Rahili et al., 2015; Subotić et al., 2022; Tang et al., 2018, 2022; Zhang et al., 2021).

Roughly speaking, algorithms for time-varying optimization can be divided into *running* algorithms that do not incorporate any information about the evolution of the problem (Bastianello & Dall'Anese, 2021; Bernstein et al., 2019; Bianchin, Cortés, Poveda & and Dall'Anese, 2022; Colombino et al., 2020; Popkov, 2005; Simonetto, 2017; Simonetto & Leus, 2014; Tang et al., 2018, 2022) and *predictive* schemes that exploit some knowledge or estimate about the change in the optimization problem (Bastianello et al., 2020; Fazlyab, Paternain, Preciado & and Ribeiro, 2018; Lesage-Landry et al., 2020; Simonetto & Dall'Anese, 2017; Simonetto et al., 2017; Zheng et al., 2020).

The following simple examples generalize Example 1.1 to a time-varying setting and present, just for the sake of illustration, a running and a predictive scheme.

Example 2.4. Consider the same setup as in Example 1.1. Namely, let a physical plant be characterized by the steady-state input–output map $y = h(u) + d(t)$. We now assume that $d(t)$ is time-varying and (Lebesgue) measurable. Consequently, we wish to track the solution of

$$\underset{u}{\text{minimize}} \quad \tilde{\Phi}(u, t) := \Phi(h(u) + d(t)). \tag{18}$$

Without modifying the controller we get the (non-autonomous) closed-loop system

$$\dot{u} = -\nabla_u \tilde{\Phi}(u, t)^T \tag{19}$$

$$= -\nabla h(u)^T \nabla \Phi(y)^T \qquad y = h(u) + d(t) \tag{20}$$

which is a running algorithm to solve (18).

Assume that $\tilde{\Phi}$ is β-strongly convex for every t and consequently has a unique global minimizer $u^\star(t)$ for every t. Further, assume that $\|u^\star(t) - u^\star(t')\| \leq \ell \|t - t'\|$ for all t, t'. In other words, the unique optimizer is ℓ-Lipschitz.

The quantity ℓ can sometimes be bounded from problem parameters (Subotić et al., 2022). For instance, if it is known that d is ℓ_d-Lipschitz in t and Φ is β'-strongly convex, then the estimate $\ell \leq \beta' \ell_d$ holds.

Exploiting strong convexity and Cauchy–Schwarz inequality, the distance between $u(t)$ of (19) and $u^\star(t)$ can be shown to differentiable for almost all t and satisfy

$$\frac{d}{dt} \frac{1}{2} \|u(t) - u^\star(t)\|^2$$
$$\leq \langle \dot{u}(t), u(t) - u^\star(t) \rangle + \ell \|u(t) - u^\star(t)\|$$
$$\leq -\beta \|u(t) - u^\star(t)\|^2 + \ell \|u(t) - u^\star(t)\|.$$

Consequently, $\|u(t) - u^\star(t)\|$ is decreasing as long as $\|u(t) - u^\star(t)\| > \ell/\beta$. It follows from standard invariance arguments that, as $t \to \infty$, $u(t)$ will be ℓ/β-close to $u^\star(t)$. ∎

Example 2.5. To illustrate a continuous-time predictive algorithm, consider the same setup as in Example 2.4 and assume, in addition, that Φ is twice continuously differentiable, and the time derivative $\dot{d}(t)$ is available (e.g., can be estimated using finite differences). Then, we may consider the following control law derived from the sensitivity-conditioning approach (Picallo et al., 2023)

$$\dot{u} = -\nabla h(u)^T \nabla_u \Phi(y)^T - \nabla h(u)^T \nabla_{yy}^2 \Phi(y) \dot{d}(t)$$

$$y = h(u) + d(t)$$

which reduces to

$$\dot{u} = -\nabla_u \tilde{\Phi}(u, t)^T - \frac{d}{dt} (\nabla_u \tilde{\Phi})^T (u, t).$$

One may consider the time-varying Lyapunov function $W(u,t) := \frac{1}{2}\|\nabla_u \tilde{\Phi}(u,t)\|^2$. It holds that

$$\frac{d}{dt}W(u(t),t) = \nabla_u W(u(t),t)\dot{u}(t) + \nabla_t W(u(t),t)$$
$$= \|\nabla_u \tilde{\Phi}(u(t),t)\|^2 < 0.$$

for all t and all $u(t) \neq u^\star(t)$. This fact can be used to guarantee zero tracking error as $t \to \infty$. ∎

3. Online feedback-based optimization

We now present the key idea and main topic of this review article—the implementation of optimization algorithms in closed loop with physical systems. The unconstrained feedback gradient flow from Example 1.1 in the introduction can be considered as a minimal viable example for this approach. A more general formulation is the problem of driving an asymptotically stable plant with the steady-state map $y = h(u) + d$ to the solution of the optimization problem

$$\min_{u,y} \quad \Phi(y) \tag{21a}$$

$$\text{subject to} \quad y = h(u) + d \tag{21b}$$

$$u \in \mathcal{U} \tag{21c}$$

$$y \in \mathcal{Y}, \tag{21d}$$

where $\mathcal{U}$ and $\mathcal{Y}$ are the feasible regions of the input u and the output y, respectively. This formulation can be generalized to include a dependence of the cost Φ on the input u directly, without any technical complication.

Almost any optimization algorithm in the form of a dynamical system can be defined as open systems and interconnected with the physical plant, but not every such control design makes sense in the real world and achieves the aforementioned goal. The resulting interconnection needs to be robust, satisfy the plant operational constraints, and be implementable given the limited model knowledge.

The seminal papers on network congestion control (Kelly et al., 1998; Low & Lapsley, 1999) have demonstrated the potential of studying optimization dynamics in closed-loop implementations, see Appendix A.1 for a self-contained review. Afterward, Jokic et al. (2009) and Brunner et al. (2012) consider a more abstract formulation of steering the outputs of a nonlinear continuous-time dynamical plant to a steady-state that solves a constrained convex optimization problem. These pioneering works inspired a plethora of contributions in different directions.

In this section, we review three fundamental aspects that constitute the core challenges when aiming to achieve a *robust* interconnection of an optimization algorithm with a physical plant:

- In Section 3.1, we present approaches to guarantee *closed-loop stability* in the presence of plant dynamics, either via singular perturbation analysis or linear matrix inequality certificates.
- In Section 3.2, we show how partial model information and feedback measurements can be used to *robustly enforce constraints* in closed-loop optimization.
- In Section 3.3, we discuss strategies for bypassing the need of model information and achieving fully *data-driven and model-free operations*.

For each of these themes, we provide a tutorial exposition and review the relevant literature.

3.1. Closed-loop stability

Example 1.2 in the introduction has illustrated that a simple gradient-based controller interconnected with a dynamical system is not necessarily stable, unless the control gain ϵ is small enough (Fig. 2). In other words, *sufficient timescale separation* between the fast plant behavior and the slow optimization dynamics is generally required.

For control design, it is convenient to reduce the fast transient plant dynamics to their algebraic steady-state map. We have explicitly applied this design step in Example 1.1 and will do so implicitly in the rest of the article. In this section, however, we will investigate the problems of timescale separation and closed-loop stability in detail, and survey works that have tackled this issue.

We identify and present two main research streams: First, we discuss stability results inspired by singular perturbation analysis, which formalize and quantify stability in terms of timescale separation. This approach is very general and applicable to nonlinear (but asymptotically stable) plant dynamics and non-convex optimization dynamics, but potentially very conservative. A second line of research taps into robust control and provides computational stability certificates in the form of linear matrix inequalities. While these stability guarantees are less conservative, they apply only to linear time-invariant plants and convex optimization dynamics.

Remark 3.1. Exploiting *passivity* is sometimes a third possibility to certify closed-loop stability (Khalil, 2002, Chap. 6). This approach requires that the control loop can be identified as a feedback interconnection of passive systems, which is not always possible. Saddle-point flows (discussed in Section 2.3) are one class of systems that are amenable to passivity arguments to study stability and robustness (Simpson-Porco, 2016; van der Schaft, 2011). These ideas have been applied to congestion avoidance in communication networks (Appendix A.1) in Wen and Arcak (2004) and in Chapter 4 of Low (2017). Similarly, passivity has been exploited in power systems applications (like Appendix A.2) in Stegink et al. (2017), Trip et al. (2019). ∎

Remark 3.2. In the following, for the sake of keeping the presentation consistent, we maintain the same modeling framework where both the plant and the optimization algorithm are described as continuous time flows. In some applications, for example when the presence of communication delays requires less frequent algorithms iterations, a sampled-data approach may be more appropriate. Belgioioso et al. (2021, 2022) provide stability results in this hybrid setting. ∎

3.1.1. Singular perturbation analysis

For the purpose of providing explicit bounds on ϵ, Hauswirth, Bolognani, Hug, and Dörfler (2020), Menta et al. (2018) pursue a singular perturbation approach and arrive at easy-to-compute sufficient conditions for closed-loop stability. In particular, Hauswirth, Bolognani, Hug, and Dörfler (2020) considers general nonlinear (but stable) plant dynamics and investigates a variety of optimization dynamics for convex and non-convex problems including (projected) gradient, saddle-point, and momentum methods. Using similar techniques, Simpson-Porco (2021a) provides stability guarantees for general low-gain integral controllers that satisfy an infinitesimal contraction property, but with a discussion of feedback-based optimization limited to LTI plant dynamics. The following example illustrates the main idea behind these stability conditions.

Example 3.1. We wish to characterize the stability of the feedback loop introduced in Example 1.1. In particular, we want to formulate conditions on the gain ϵ in Fig. 1 that guarantee closed-loop stability. For this purpose we pass to the singular perturbation decomposition into reduced and boundary-layer error dynamics illustrated in Fig. 7 (Khalil, 2002; Kokotovic et al., 1999). In particular, $\hat{h}$ is defined such that $f(\hat{h}(u), u) = 0$ for all u.

The resulting reduced dynamics correspond exactly to the simplified model that we have already used in the design of the optimizing controller, where the plant is replaced by its algebraic steady-state map. The boundary-layer error dynamics $z := \zeta - \hat{h}(u)$ evolve as $\dot{z} = f(\zeta, u)$ for any fixed u. If these error dynamics are exponentially stable (and
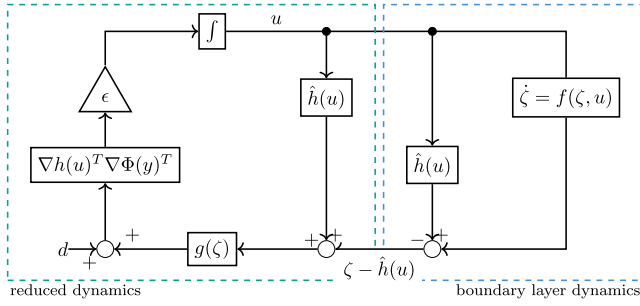
Fig. 7. Feedback-based gradient flow from Fig. 1 rearranged and decomposed into reduced and boundary-layer error dynamics.

Fig. 8. Feedback-based gradient flow from Fig. 1 rewritten as a feedback interconnection of an LTI system with a nonlinearity Δ.

other technical assumptions are satisfied), standard converse results guarantee the existence of a Lyapunov function W and parameters $\gamma, \omega > 0$ such that, for any fixed u, it holds that

$$
\begin{aligned}
&\dot{W}(\zeta - \hat{h}(u)) \le -\gamma \|\zeta - \hat{h}(u)\|^2 \\
&\|\nabla_u W(\zeta - \hat{h}(u))\| \le \omega \|\zeta - \hat{h}(u)\| .
\end{aligned}
\tag{22}
$$

A class of Lyapunov function candidates to certify stability of the closed system in Fig. 7 is given by

$$
V_\delta(u, \zeta) = \delta \tilde{\Phi}(u) + (1 - \delta) W(\zeta - \hat{h}(u))
$$

where $\delta \in (0, 1)$ is a convex combination parameter.

Let L be the Lipschitz constant of $\nabla \hat{\Phi}$. Hauswirth, Bolognani, Hug, and Dörfler (2020) have shown that for all

$$
\epsilon < \epsilon^\star := \frac{\gamma}{\omega L}
\tag{23}
$$

the parameter δ can be chosen such that V_δ is non-increasing and thus a LaSalle invariance argument guarantees (asymptotic) stability. ∎

Remark 3.3. Recall from Section 2.4 that constraints can be incorporated into an optimization problem through the combination of gradient flows with penalty or barrier functions. The bound (23) on ϵ in Example 3.1 indicates that there is a natural limitation to this constraint enforcement mechanism. Notably, penalty functions cannot be made arbitrarily steep, thus increasing L, without at the same time reducing ϵ by the same fraction. For the same reason log-barrier functions that do not have a Lipschitz gradient have to be applied with great care, as closed-loop stability cannot be guaranteed with these singular perturbation bounds on ϵ. ∎

The timescale separation argument inspired by singular perturbation analysis, as presented in Example 3.1, works under very general conditions. In particular, for the case of a gradient flow convexity of Φ is not generally required. The type of stability proof can also be established for various optimization algorithms interconnected with exponentially stable plants. Examples include but are not limited to Newton flows, projected gradient flows, and saddle-point flows encountered in the previous section. Moreover, the above timescale separation argument can also be applied in the sampled-data setting that involves the interconnection of continuous-time plants and discrete-time equilibrium-seeking algorithms, see the feedback equilibrium seeking in Belgioioso et al. (2023, 2021, 2022). These conditions can be very conservative, but they are qualitatively tight. Namely, nonexamples in Hauswirth, Bolognani, Hug, and Dörfler (2020) show how subgradient flows and continuous-time accelerated gradient flows interconnected with dynamical plants are not generally stable. These setups are not amenable to the same type of timescale separation argument as above because important assumptions such as uniform asymptotic stability of the reduced dynamics are not satisfied (Poveda & Li, 2019).
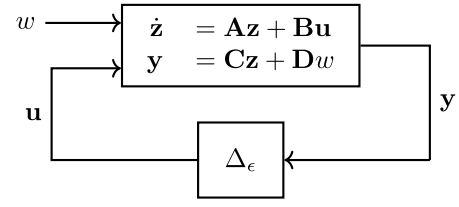
3.1.2. LMI stability certificates

Alternatively, closed-loop stability can be certified by applying tools from robust control, namely linear matrix inequalities. Recent results on integral quadratic constraints and their use for the analysis of optimization algorithms (Fazlyab, Ribeiro, Morari & and Preciado, 2018; Lessard et al., 2016) have proven very useful for this purpose and have been applied in Colombino et al. (2020, 2019), Nelson and Mallada (2018). In particular, Nelson and Mallada (2018) studies the joint design of stabilizing control, estimator, and optimization dynamics. Similarly, Lawrence et al. (2018, 2021) consider an output regulation framework and reduce the control design to a stabilization problem for the case of convex problems and LTI systems.

Limiting the applicability of these techniques from robust control is the fact that plant dynamics are generally required to be LTI, and objective functions need to be (strongly) convex. Moreover, these LMI-based conditions are in the form of computational stability certificates and do not directly translate into control design procedures or tuning recommendations.

The following example presents a LMI-IQC based stability test for the feedback-based gradient flow.

Example 3.2. Consider the same setup as in Example 1.1, and, in addition, assume that Φ is m-strongly convex and has a L-Lipschitz gradient. Further, assume that

$$
\dot{\zeta} = f(\zeta, u) = Ax + Bu \qquad y = g(\zeta) + d = C\zeta + d
\tag{24}
$$

is an LTI system. Assuming that A is invertible, the steady-state input–output map is given by $h(u) = Hu$ with $H = -CA^{-1}B$. Furthermore, (3) has a unique equilibrium $(\zeta^\star, u^\star)$ such that $\nabla \tilde{\Phi}(u^\star) = 0$ and $0 = A\zeta^\star + Bu^\star$.

The closed-loop system Eq. (3) can be written in a standard robust control setup as an interconnection of a nonlinearity with an LTI system. This is shown in Fig. 8, where

$$
\mathbf{z} := \begin{bmatrix} \zeta \\ u \end{bmatrix} \qquad \mathbf{A} := \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \qquad \mathbf{B} := \begin{bmatrix} 0 & 0 \\ 0 & -H^T \end{bmatrix}
$$

$$
\mathbf{y} := \begin{bmatrix} y \\ u \end{bmatrix} \qquad \mathbf{C} := \begin{bmatrix} 0 & \mathbb{I} \\ C & 0 \end{bmatrix} \qquad \mathbf{D} := \begin{bmatrix} \mathbb{I} \\ 0 \end{bmatrix}
$$

and

$$
\mathbf{u} = \Delta_\epsilon(\mathbf{y}) = \begin{bmatrix} 0 \\ \epsilon \nabla \Phi(y) \end{bmatrix} .
\tag{25}
$$

Crucially, $\Delta_\epsilon(\cdot)$ satisfies the IQC defined by

$$
\begin{bmatrix} \mathbf{y} - \mathbf{y}^\star \\ \mathbf{u} - \mathbf{u}^\star \end{bmatrix}^T \underbrace{\begin{bmatrix} -2\epsilon^2 mL\mathbb{I} & \epsilon(L+m)\mathbb{I} \\ \epsilon(L+m)\mathbb{I} & -2\mathbb{I} \end{bmatrix}}_{\Xi_\epsilon} \begin{bmatrix} \mathbf{y} - \mathbf{y}^\star \\ \mathbf{u} - \mathbf{u}^\star \end{bmatrix} \ge 0
\tag{26}
$$

for all (y, u) such that $y = \Delta(y)$ (Lessard et al., 2016, Lemma 6). Equivalently, we have

$$
\begin{bmatrix} \mathbf{z} - \mathbf{z}^\star \\ \mathbf{u} - \mathbf{u}^\star \end{bmatrix}^T \mathbf{C}^T \Xi_\epsilon \mathbf{C} \begin{bmatrix} \mathbf{z} - \mathbf{z}^\star \\ \mathbf{u} - \mathbf{u}^\star \end{bmatrix} \ge 0
\tag{27}
$$

for all $(\mathbf{z}, \mathbf{u})$ such that $\mathbf{u} = \Delta(\mathbf{Cz})$.

Next, consider a Lyapunov function of the form $V(\mathbf{z}) = (\mathbf{z} - \mathbf{z}^\star)^T \mathbf{P}(\mathbf{z} - \mathbf{z}^\star)$ where $\mathbf{P} \succeq 0$ remains to be determined. The time derivative of V along system trajectories can be written as a quadratic form

$$\dot{V}(\mathbf{z}) = \begin{bmatrix} \mathbf{z} - \mathbf{z}^\star \\ \mathbf{u} - \mathbf{u}^\star \end{bmatrix}^T \begin{bmatrix} \mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} & \mathbf{P}\mathbf{B} \\ \mathbf{B}^T \mathbf{P} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z} - \mathbf{z}^\star \\ \mathbf{u} - \mathbf{u}^\star \end{bmatrix}. \tag{28}$$

Combining (27) and Eq. (28), it follows that $\dot{V}(\mathbf{z}) < 0$ for $\mathbf{z} \neq \mathbf{z}^\star$ (and thus the closed-loop is stable) if the LMI

$$\begin{bmatrix} \mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} & \mathbf{P}\mathbf{B} \\ \mathbf{B}^T \mathbf{P} & 0 \end{bmatrix} \prec -\mathbf{C}^T \Xi_\epsilon \mathbf{C} \tag{29}$$

holds for some $\mathbf{P} \succeq 0$. ∎

3.2. Constraint enforcement in closed loop

The capacity and various possibilities to robustly enforce complicated constraints despite model uncertainty is one of the distinguishing features of feedback-based optimization.

In Section 2 we have seen different mechanisms used in continuous-time optimization algorithms to deal with constraints. Section 2.4 summarizes these options and highlights how they can be combined. In this subsection, we will explore the closed-loop implementations of these mechanisms. In particular, in Section 3.2.1 we will discuss the use of saturation to enforce constraints on plant inputs without major computational effort. In Section 3.2.2, we will review strategies to handle general output constraints.

3.2.1. Input saturation via projection and anti-windup

Physical plants are generally subject to limited actuator capabilities, simply due to the fact that any realistic system is bound by the laws of physics and can handle only signals of finite power. Such actuation limits can often be modeled as input saturation of control signal. At a physical level, such saturation is, for example, the result of mechanical constraints, or it can be observed in electric circuits involving diodes and other semi-conducting devices. From a systems perspective, low-level controllers that protect devices and subsystems from operating outside of a safe zone of conditions can also be modeled as saturation. The nature of these constraints implies that they cannot be violated at any point in time. Given the constraint enforcement possibilities from Section 2.4, it makes sense to model saturation as a projection onto the set of feasible inputs with the key property that the projection is naturally "evaluated" and applied by the physical system.

This property creates the possibility of outsourcing the handling of these constraints from the controller to the plant and thus reduces computational requirements and the need for exact modeling of these constraints, which can be time-varying and/or unknown.

For discrete-time feedback-based optimization schemes such as those in Bernstein et al. (2019), Dall'Anese and Simonetto (2018), Gan and Low (2016), Simpson-Porco (2022), this model-free constraint handling by saturation is relatively straightforward (at least for gradient descent with standard Euclidean metric), under the assumption that the saturated control signal is measured and available within one sampling interval. In this case, it is possible to simply add the control increment to the measured saturated control signal of the preceding sampling interval.

For continuous-time methods, as in Chang et al. (2019), Hauswirth et al. (2016), exploiting input saturation is trickier because a continuous-time integrator in cascade with a saturation element will generally lead to *integrator windup*. For this reason Hauswirth et al. (2020a, 2020b, 2020c) study the use of anti-windup compensators for feedback-based optimization and rigorously show how these control designs can be used to smoothly and robustly approximate continuous-time projected gradient flows like Eq. (7). The following example illustrates this idea.

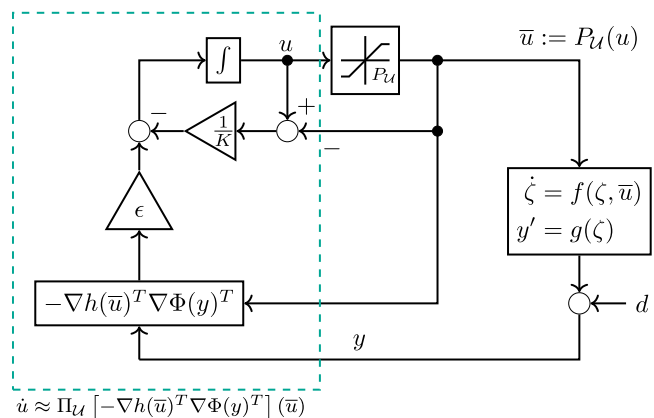**Fig. 9.** Anti-windup system (30) which approximates a projected gradient flow (33) as $K \to 0^+$.

Example 3.3.

Consider the same setup as in Example 1.1. Namely, we want to drive a plant (with fast decaying dynamics and steady-state map $y = h(u) + d$) to an optimal steady state minimizing the cost function $\tilde{\Phi}(u)$.

In addition, we assume that the plant is subject to input saturation that acts like a projection onto a non-empty set $\mathcal{U}$ of admissible inputs. For simplicity, assume that $\mathcal{U}$ is convex. In order to mitigate the effects of integrator windup, a simple anti-windup scheme with a tuneable gain $\frac{1}{K}$ is in place as illustrated in Fig. 9.

A state-space representation of this system is given, analogously to Eq. (3), by

$$\text{plant} \begin{cases} \overline{u} &= P_{\mathcal{U}(u)} \\ \dot{\zeta} &= f(\zeta, \overline{u}) \\ y &= g(\zeta) + d \end{cases} \tag{30}$$

$$\text{controller} \left\{ \dot{u} = -\epsilon \nabla h(u)^T \nabla \Phi(y)^T - \frac{1}{K}(u - \overline{u}) \right..$$

Assuming, as before, that the plant dynamics can be approximated by the algebraic map $y = h(u) + d$ and with $\tilde{\Phi}(u) = \Phi(h(u) + d)$, the system (30) reduces to

$$\dot{u} = -\epsilon \nabla \tilde{\Phi}(P_{\mathcal{U}(u)})^T - \frac{1}{K}(u - P_{\mathcal{U}(u)}). \tag{31}$$

The key aspect in Eq. (31) is the fact that $\nabla \tilde{\Phi}$ is evaluated at $P_{\mathcal{U}(u)}$ rather than at u. In Hauswirth et al. (2020b) it was shown that trajectories $u(t)$ of (31) converge in the sense that $P_{\mathcal{U}}(u(t))$ converges to the set of KKT points of the optimization problem

$$\text{minimize} \quad \tilde{\Phi}(u) \qquad \text{subject to} \quad u \in \mathcal{U}. \tag{32}$$

In particular, if (32) is convex, then convergence of $P_{\mathcal{U}}(u(t))$ is to the global minimizer.

Importantly, this means that upon convergence, the plant in Fig. 9 is at an optimal steady state even though the (unsaturated) control u is not optimal (only the saturated control $P_{\mathcal{U}(u)}$ is optimal).

Furthermore, Hauswirth et al. (2020a, 2020c) have shown that (31) approximates the projected gradient flow

$$\dot{u} = \Pi_{\mathcal{U}} \left[-\nabla h(\overline{u})^T \nabla \Phi(y)^T \right] (\overline{u}) \tag{33}$$

as $K \to 0^+$. This insight extends to other anti-windup schemes which can be shown to approximate more general projected dynamical systems. ∎

3.2.2. Output constraints via dualization and approximate projections

We now turn to mechanisms that allow us to enforce more general constraints, and in particular those that apply to plant outputs.

In the following, we focus on two approaches. First, we illustrate how augmented and projected saddle-point flows, as in Example 2.3,

can be implemented as feedback controllers. This approach is particularly suited for solving convex problems in closed loop and distributed over a network (assuming the problem exhibits a suitable sparsity structure). For non-convex problems, these algorithms can still be applied, but theoretical global convergence guarantees are not generally available.

As a second possibility to enforce output constraints, we discuss a special discretization of projected gradient flows that can be implemented as a feedback controller and comes with strong global convergence guarantees, even for non-convex setups. However, it is less easily amenable to a distributed implementation and, instead, requires the solution of a simple quadratic program at every iteration.

Projected saddle-flows as feedback controllers. Consider the optimal steady-state problem Eq. (21). Assume that the disturbance d is fixed and that the engineering constraints $(u, y) \in \mathcal{X}$ can be expressed as $\mathcal{X} := \{(u, y) \mid g(y, u) \leq 0\}$, where $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ denotes a continuously differentiable constraint function. After eliminating y from Eq. (21), we are left with the reduced problem

$$\underset{u}{\text{minimize}} \quad \hat{\Phi}(u) := \Phi(u, h(u) + d) \tag{34a}$$

$$\text{subject to} \quad u \in \mathcal{U} \tag{34b}$$

$$u \in \hat{\mathcal{X}} := \{u \mid \hat{g}(u) \leq 0\} \tag{34c}$$

with $\hat{g}(u) := g(u, h(u))$.

This problem can be tackled with a projected saddle-point flow with a primal augmentation term $\phi(u) = \frac{\rho}{2} \| \max\{\hat{g}(u), 0\} \|^2$ as presented in Example 2.3. Namely, we have

$$\dot{u} = \Pi_{\mathcal{U}} \left[-\nabla \hat{\Phi}(u)^T - \nabla \hat{g}(u)^T (\mu + \rho \max\{\hat{g}(u), 0\}) \right] (u) \tag{35a}$$

$$\dot{\mu} = \Pi_{\mathbb{R}^m_{\geq 0}} \left[\hat{g}(u) \right] (\mu). \tag{35b}$$

Recall from Example 2.3 that, in the special case where (34) is convex, trajectories of (35) converge to a global minimizer. In particular thanks to the primal augmentation, as in Eq. (16). The primal augmentation term reduces oscillations, especially with non-strictly convex objective functions. If the problem is non-convex, a dual augmentation term, as in Eq. (17), can be added to guarantee global convergence at the expense of changing equilibrium points away from the true KKT points of (34).

We wish to realize (35) as the closed-loop dynamics of a feedback loop incorporating the physical plant. To achieve this, we replace any evaluation of $h(u) + d$ with the plant output y. This yields the controller

$$\dot{u} = \Pi_{\mathcal{U}}^Q \left[-\epsilon Q(u) H(u) G(u, y) \right] (u) \tag{36a}$$

$$\dot{\mu} = \Pi_{\mathbb{R}^m_{\geq 0}} \left[g(u, y) \right] (\mu) \tag{36b}$$

where $H(u) := \begin{bmatrix} \mathbb{I} & \nabla h(u)^T \end{bmatrix}$ and

$$G(u, y) := \begin{bmatrix} \nabla_u \Phi(u, y)^T - \nabla_u g(u, y)^T (\mu + \rho \max\{g(u, y), 0\}) \\ \nabla_y \Phi(u, y)^T - \nabla_y g(u, y)^T (\mu + \rho \max\{g(u, y), 0\}) \end{bmatrix}.$$

We have also included in Eq. (36) a variable metric $Q(u)$ on the primal variables. This degree of freedom will be exploited in the forthcoming Examples 4.1 and 4.2.

In practice, one can build an estimate $\hat{H}$ of $H(u)$ by using y and u, i.e., $\hat{H}(u, y) \approx H(u)$. We defer the discussion on how to perform this estimation to Section 3.3.1.

Furthermore, if the plant (1) is an asymptotically stable LTI system subject to constant disturbances

$$\dot{\xi} = A\xi + Bu + d_1 \qquad y = C\xi + Du + d_2, \tag{37}$$

then the steady-state sensitivity matrix $\nabla h(u) = -CA^{-1}B + D$ is constant and independent of d, which renders its estimation much easier.

Despite a lack of theoretical convergence guarantees for non-convex problems and closed-loop stability certificates, primal–dual saddle-point algorithms (and their projected, proximal, and augmented variations) have enjoyed great popularity. In particular, saddle-point flows

often lend themselves to a distributed implementation. Appendices A.1 and A.2 are prime examples. In the context of (36), a distributed implementation is possible if $\nabla h(u)$ has a fixed sparsity pattern, the cost function is separable, and the constraints $u \in \mathcal{U}$ and $g(u, y) \leq 0$ are appropriately localized. Moreover, selecting a sparsity-inducing metric $Q(u)$ can also facilitate distributed operations. In this case, the computation of a given component u_i requires only the quantity $\mu_j + \rho \max\{g_j(u, y), 0\}$ to be communicated from all neighboring nodes j in a network.

Linearized output projections. As a second possibility to solve (21) in closed loop (and thus tackle the reserve dispatch problem from Section 4), we highlight the method proposed in Häberle et al. (2021) and extended in Hauswirth (2020). This method is conceptually related to Torrisi et al. (2018), in that we seek a descent direction that ensures satisfying the linearized constraint.

We have seen how projected (and augmented) saddle-point flows can be easily turned into feedback controllers. The key mechanism for enforcing the engineering constraints $(u, y) \in \mathcal{X}$ thereby lies in integrating the constraint violation $g(u, y)$ that is based on plant output measurements. In Example 4.1 we have noted the ease of deriving a discrete controller by simply applying an explicit Euler discretization.

Now, instead, we present a discrete controller that approximates the projected gradient flow

$$\dot{u} = \Pi_{\mathcal{U} \cap \hat{\mathcal{X}}} [-\nabla \hat{\Phi}(u)] (u). \tag{38}$$

The trajectories of (38) are guaranteed to converge to the KKT points of the reduced problem Eq. (34), even under non-convexity (but under technical assumptions similar to those in Theorem 1). We refer to Hauswirth, Bolognani, and Dörfler (2020) for a discussion of the available convergence guarantees for projected gradient flows.

We have previously seen in Section 3.2.1 that input constraints $u \in \mathcal{U}$ are often enforced by input saturation (possibly requiring an anti-windup compensator to avoid integrator windup). The challenge in (38) lies in the fact that also the engineering constraints $\mathcal{X}$ need to be enforced "by projection" and in feedback using measurements of the plant output, rather than using a model-based evaluation of $h(u) + d$.

To address this problem, rather than pondering on the continuous-time dynamics (38), we directly consider a discrete-time controller of the form

$$u^+ = u + \alpha \, \Sigma_\alpha(u, y), \tag{39}$$

where $\alpha > 0$ is a fixed step-size, $y = h(u) + d$ is the measured system output, and $\Sigma_\alpha(u, y)$ is defined as the solution of

$$\underset{w \in \mathbb{R}^p}{\text{minimize}} \| w + Q(u) H(u) \nabla \Phi(u, y)^T \|_{Q(u)}^2 \tag{40a}$$

$$\text{subject to} \quad u + \alpha w \leq \mathcal{U} \tag{40b}$$

$$g(u + \alpha w, y + \alpha \nabla h(u)w) \leq 0, \tag{40c}$$

where, as before, $H(u) := \begin{bmatrix} \mathbb{I}_p & \nabla h(u)^T \end{bmatrix}$ and $Q(u)$ is a metric that assigns to every $u \in \mathcal{U}$ a positive definite matrix.

The feedback law (39) approximates a projected gradient descent, by computing a descent direction w that is feasible with respect to $\mathcal{U}$ (see Eq. (40b)) and approximately feasible (up to first order) with respect to $\mathcal{X}$ (see Eq. (40c)). In particular, it can be rigorously shown that (39) approximates (38) as $\alpha \rightarrow 0^+$ (Hauswirth, 2020).

Any equilibrium point of (39) is feasible and a KKT point of (21). Global convergence of (39) to KKT points of (21) is guaranteed under weak technical assumptions and without convexity, and for a small enough step size α (Häberle et al., 2021).

Note that, computing $\Sigma_\alpha(u, y)$ requires the solution of a quadratic program. However, in comparison to real-time iteration (RTI) schemes discussed in Appendix B.3, the computational effort does not scale with a prediction horizon and no explicit model of the plant dynamics is required. Instead, it is enough to estimate $\nabla h(u)$. On the other hand, in general, (39) does not lend itself to a natural distributed
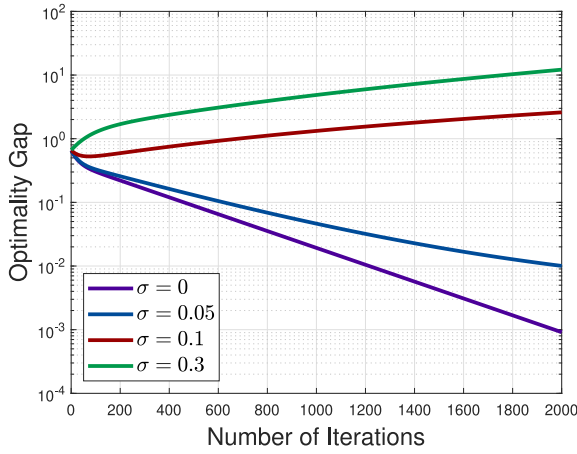
Fig. 10. Performance of feedback-based optimization schemes that use inexact sensitivity estimations.

implementation. Although, depending on the problem structure (e.g., a diagonally dominant $\nabla h(u)$ or a sparsity-inducing metric $Q(u)$), one can solve (40) distributedly at every iteration.

3.3. Data-driven and model-free operation

The appealing benefits of the aforementioned feedback-based optimization schemes are built on limited model information, i.e., the sensitivity $\nabla h(u)$ of the steady-state input–output map of a physical plant. Due to the chain rule, this information is required during the gradient-based (i.e., first-order) updates of those schemes. In some cases, we can derive a quite accurate sensitivity estimate based on first-principle models and parameters (Ortmann, Hauswirth, et al., 2020; Ortmann, Prostejovsky, et al., 2020). In a broader context, however, the sensitivity may be elusive because of the complex structure of a plant and the lack of key model parameters. As a result, the effective functioning (e.g., stability and optimality) of feedback-based optimization is affected.

Example 3.4. Consider a similar setup as in Example 3.2. The objective $\Phi(u, y) = \frac{1}{2}(u^\top R u + y^\top Q y)$ is strongly convex and has Lipschitz continuous gradients with respect to u and y. A discrete-time linear plant is given by

$$\zeta_{k+1} = A\zeta_k + Bu_k, \qquad y_k = C\zeta_k + d. \qquad (41)$$

The steady-state input–output map is $y = Hu + d$, where $H = C(I - A)^{-1}B$ is the sensitivity matrix. Let $\hat{H}$ be an estimation of H. We synthetically generate $\hat{H}$ by $\hat{H} = H + \sigma M$, where $\sigma \geq 0$, and the elements of M are drawn from the uniform distribution. Fig. 10 illustrates the performance of the closed-loop interconnection of the plant (41) and the following feedback-based optimization scheme using $\hat{H}$

$$u_{k+1} = u_k - \eta(Ru_k + \hat{H}^\top Q y_{k+1}).$$

We use the same step size η for different cases of σ. We observe that feedback-based optimization is robust to small inaccuracies (Colombino et al., 2019). However, as σ further increases (i.e., the error level of the sensitivity estimation increases), the closed-loop behavior exhibits sub-optimality and even instability. Hence, the accuracy of the sensitivity plays an important role in guaranteeing effective closed-loop performance. ∎

Faced with elusive sensitivities, fortunately, we may possess sufficiently rich input–output data collected during historical operations, or we can perform strategic exploration and obtain helpful data on the fly.

It is meaningful to tackle the challenges posed by unknown sensitivities by extracting useful patterns from data and exploiting such patterns in the updates.

We review two main strategies to achieve data-driven and model-free operations. The first strategy consists in learning sensitivities from offline or online data and then plugging them into the first-order updates of the controller. Nonetheless, there also exist scenarios where sensitivities are highly state-dependent and hence hard to learn (e.g., for nonlinear dynamic plants) or even non-existent (e.g., for a non-differentiable steady-state map $h(u)+d$). To handle these issues, the second strategy circumvents learning sensitivities and instead utilizes derivative-free optimization techniques.

3.3.1. Online sensitivity learning

The knowledge of steady-state input–output sensitivities is a key requirement for the above-mentioned feedback-based optimization schemes. However, such knowledge may not be easily accessible in some applications because of the lack of exact models and accurate parameters. To address this issue, we can utilize offline data or online interactions to perform sensitivity learning. Afterward, we plug these sensitivity surrogates in the gradient-based updates of feedback-based optimization schemes.

We review two approaches to sensitivity learning. The first strategy leverages Willems' fundamental lemma (Willems et al., 2005) and uses open-loop input–output trajectories to parametrize the sensitivity matrix of an LTI plant (Bianchin et al., 2023; Nonhoff & Müller, 2021, 2022). Provided that finite-length trajectories are informative enough (i.e., they satisfy a persistence-of-excitation condition), the Hankel matrices constructed from such trajectories characterize the whole input–output behavior of an LTI plant (see the review in Markovsky and Dörfler (2021), Markovsky et al. (2023)). Remarkably, we bypass the step of identifying system matrices and perform control from raw data in a direct manner. The extension to handle general nonlinear plants remains to be explored.

An alternative strategy learns the sensitivities of nonlinear algebraic maps through recursive least squares estimation. The key idea is to feed the data collected during online implementations to a recursive estimator, thus gradually improving the learning accuracy (Dominguez-Garcia et al., 2023; Picallo et al., 2022). With the iterative adjustments of feedback-based optimization, we collect a series of inputs and the corresponding measured outputs. The differences between the input–output pairs at neighboring time steps provide rich information on the rates of change, i.e., sensitivities. Since such data arrives in an online fashion, we perform recursive estimation to learn more and more accurate sensitivities. We illustrate this idea through the following example with nonlinear algebraic maps.

Example 3.5. We revisit Example 1.1 and consider a discrete-time plant described by a differentiable steady-state map $y = h(u) + d$. The objective $\Phi(y)$ is a function of the steady-state output y and is equivalent to the reduced cost $\tilde{\Phi}(u) := \Phi(h(u) + d)$. For the output y_{k+1} at time $k + 1$, we analyze its first-order Taylor approximation

$$y_{k+1} = h(u_{k+1}) + d \approx \underbrace{h(u_k) + d}_{=y_k} + \nabla h(u_k)(u_{k+1} - u_k),$$

where we consider a small change of the input $\Delta u_k = u_{k+1} - u_k$. Hence, the map from $\Delta u_k = u_{k+1} - u_k$ to the change of the output $\Delta y_k = y_{k+1} - y_k$ is dominated by the linear sensitivity, i.e., $\Delta y_k \approx \nabla h(u_k)\Delta u_k$, subject to errors due to approximation. Let $s_k = \text{vec}(\nabla h(u_k))$ and $U_{\Delta,k} = \Delta u_k^\top \otimes I$, where vec and $\otimes$ denote the column-wise vector representation and the Kronecker product, respectively. The dynamics of the sensitivity s_k and the measurement equation are given by

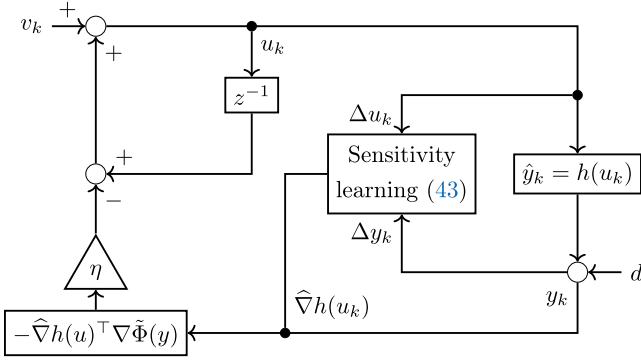$$\begin{aligned} s_{k+1} &= s_k + w_{p,k}, \\ \Delta y_k &= U_{\Delta,k} s_k + w_{m,k}, \end{aligned} \qquad (42)$$

Fig. 11. Feedback-based optimization with sensitivity learning via recursive least squares (44).

where $w_{p,k}$ and $w_{m,k}$ represent the change and the approximation error, respectively. Picallo et al. (2022) assume that they are modeled by Gaussian noises with covariance matrices $\Sigma_{p,k}$ and $\Sigma_{m,k}$. To reflect the dependence of s_k on u_k, it is further assumed that $\Sigma_{p,k}$ is proportional to $\|\Delta u_k\|^2$. Then, learning sensitivity based on (42) amounts to the classical problem of state estimation. The online sensitivity learning strategy (Picallo et al., 2022) based on the Kalman filter is as follows

$$\hat{s}_{k+1} = \hat{s}_k + K_k \left(\Delta y_k - U_{\Delta,k} \hat{s}_k \right),$$
$$\Sigma_{k+1} = \left(I - K_k U_{\Delta,k} \right) \Sigma_k + \Sigma_{p,k},$$
$$K_k = \Sigma_k U_{\Delta,k}^\top \left(\Sigma_{m,k} + U_{\Delta k} \Sigma_k U_{\Delta k}^\top \right)^{-1}.$$
(43)

Afterward, we convert the vector representation $\hat{s}_k$ to the matrix form $\widehat{\nabla} h(u_k)$ and plug it into the updates of the controller. As illustrated by Fig. 11, the closed-loop interconnection is

$$\text{plant} \begin{cases} y_k &= h(u_k) + d \\ \text{controller} \begin{cases} u_{k+1} &= u_k - \eta \widehat{\nabla} h(u_k)^\top \nabla \tilde{\Phi}(y_k) + v_k, \end{cases} \end{cases}$$
(44)

where $\eta > 0$ is the step size, v_k is a random excitation noise, and $\widehat{\nabla} h(u_k)$ is obtained from (43). Picallo et al. (2022) show that the interconnection of (43) and the plant ensures that $\widehat{\nabla} h(u_k)$ is an asymptotically unbiased estimate of the sensitivity $\nabla h(u_k)$ with a bounded variance. The bound on the solution accuracy (i.e., the distance of u_k to the optimal point) depends on this variance.

The strategy (43) uses one previous measurement. By contrast, the recursive sensitivity learning strategy in Dominguez-Garcia et al. (2023) exploits all the historical data by assigning decaying weights to old measurements.

For the extension to nonlinear dynamic plants, the central requirement of (44) is to achieve sufficiently accurate and fast learning of sensitivities. Otherwise, the controller will suffer from sub-optimality due to cumulative errors in gradient steps. ∎

3.3.2. Model-free control via derivative-free optimization

Confronted with general cases involving nonlinear dynamic plants or non-differentiable steady-state maps, we may not be able to learn sensitivities accurately enough. This restriction may jeopardize the overall performance of feedback-based optimization.

A natural angle of attack is to give up learning sensitivities altogether and pursue entirely model-free operations. To achieve this goal, we consider feedback-based optimization without gradient evaluations deeply rooted in derivative-free optimization. The precursors include the Nelder–Mead simplex method (Nelder & Mead, 1965) and evolutionary optimization algorithms (Simon, 2013) (see Conn et al. (2009) for an extensive review). The paradigms more related to this review are simultaneous perturbation stochastic approximation (Spall, 1992) and its modern variant, i.e., zeroth-order optimization (ZO) (Liu

et al., 2020; Nesterov & Spokoiny, 2017), as well as Bayesian optimization (Frazier, 2018; Shahriari et al., 2016). We provide a more detailed account of model-free feedback-based optimization schemes based on ZO, because they i) handle more general problems involving high-dimensional systems, and (ii) ensure closed-loop stability when interconnected with dynamic plants (similar to the timescale separation argument in Section 3.1.1).

Zeroth-order optimization-based schemes. The key idea of ZO is to construct stochastic gradient estimates from the evaluations of objective values, and then set update directions as negative gradient estimates. Zeroth-order optimization algorithms enjoy comparable convergence rates as their first-order counterparts, albeit further exhibiting polynomial dependence on the dimension of the problem (Chen et al., 2022; Duchi et al., 2015; Flaxman et al., 2005; Shamir, 2017; Zhang et al., 2022).

Recall that we aim to realize fully model-free feedback-based optimization. The insight from ZO is that we can utilize inputs and real-time outputs to evaluate objective values, construct gradient estimates, and then iteratively update control inputs (Chen, Bernstein, Devraj & Meyn, 2020; He et al., 2023; Poveda & Teel, 2017b; Tang et al., 2023). While constructing gradient estimates, however, we cannot obtain numerous independent and identically distributed samples from the plant, because every input actuation changes the operating condition. Hence, we need to judiciously use samples and select inputs. In the following example, we illustrate the model-free designs of feedback-based optimization.

Example 3.6. Similar to Example 3.5, we consider a discrete-time plant with the steady-state map $y = h(u) + d$. Let $\tilde{\Phi}(u) := \Phi(h(u) + d)$ denote the reduced cost function. We assume that $\tilde{\Phi}(u)$ is smooth and Lipschitz continuous.

We first discuss how to construct a stochastic gradient estimate. The key idea is to explore $\tilde{\Phi}(u)$ at perturbed points and scale a random vector with quantities related to objective values to form the estimate. Let $v \sim \mathcal{N}(0, \frac{1}{p}I)$ be a p-dimensional vector sampled from the normal distribution. Consider the directional derivative $\langle \nabla \tilde{\Phi}(u), v \rangle$ of $\tilde{\Phi}(u)$ along v. We use the finite differences of function values to approximate it as follows

$$\langle \nabla \tilde{\Phi}(u), v \rangle \approx \frac{\tilde{\Phi}(u + \delta v) - \tilde{\Phi}(u)}{\delta},$$

where $\delta > 0$ is the smoothing radius and is typically chosen sufficiently small. Since v is stochastic and satisfies $\mathbb{E}_v \left[vv^\top \right] = \frac{1}{p}I$, we have

$$\mathbb{E}_v \left[p \langle \nabla \tilde{\Phi}(u), v \rangle v \right] = \mathbb{E}_v \left[pvv^\top \nabla \tilde{\Phi}(u) \right] = \nabla \tilde{\Phi}(u),$$

which implies that $p \langle \nabla \tilde{\Phi}(u), v \rangle v$ is, in expectation, an unbiased estimate of the gradient $\nabla \tilde{\Phi}(u)$. Thus, we replace $\langle \nabla \tilde{\Phi}(u), v \rangle$ with the aforementioned finite differences and obtain the following two-point gradient estimates (Duchi et al., 2015)

$$\phi^{(2)}(u) = \frac{p}{\delta} (\tilde{\Phi}(u + \delta v) - \tilde{\Phi}(u)) v.$$
(45)

The above estimates are still useful in the sense of expectation. In fact, $\mathbb{E}_v[\phi^{(2)}(u)]$ and $\mathbb{E}_v[\tilde{\phi}^{(2)}(u)]$ equal to the gradient of the smooth approximation $\tilde{\Phi}_\delta(u)$ of the objective function $\tilde{\Phi}(u)$ (Nesterov & Spokoiny, 2017), where

$$\tilde{\Phi}_\delta(u) = \mathbb{E}_v \left[\tilde{\Phi}(u + \delta v) \right].$$

Moreover, the gradients and the values of $\tilde{\Phi}_\delta(u)$ and $\tilde{\Phi}(u)$ can be made arbitrarily close by selecting a δ sufficiently close to 0.

The gradient estimates in (45) require two evaluations of objective values. We need to actuate the plant twice with $u + \delta v$ and u, measure the corresponding steady-state outputs y and y', and then evaluate the objective values $\tilde{\Phi}(y)$ and $\tilde{\Phi}(y')$. Such designs work well for algebraic maps, where steady-state outputs are directly available (Chen, Bernstein, Devraj & Meyn, 2020; Poveda & Teel, 2017b; Tang et al.,
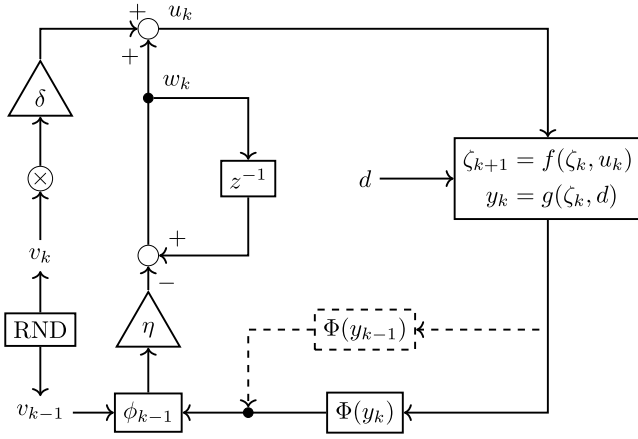
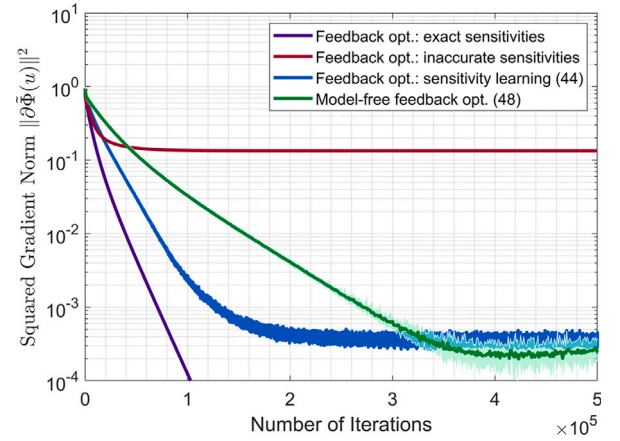**Fig. 12.** Model-free feedback-based optimization (48) in a closed loop.

Fig. 13. Comparison of the model-free feedback controller in Eq. (48) and the feedback controllers with exact sensitivities, inaccurate sensitivities, or sensitivity learning as in Eq. (44).

2023). Nonetheless, they may not be suitable for real-time decision-making with dynamic plants. In that case, within a short interval, it would be challenging to inject two different inputs and measure the corresponding steady-state outputs. That is, every time interval may only allow one actuation step and return one dynamic measurement.

The key to real-time operations lies in utilizing one new evaluation of objective values. A close look at $\mathbb{E}_v[\phi^{(2)}(u)]$ in (45) helps to achieve this goal. Indeed,

$$\mathbb{E}_v[\phi^{(2)}(u)] = \mathbb{E}_v\left[\frac{p}{\delta}\tilde{\Phi}(u+\delta v)v - \frac{p}{\delta}\tilde{\Phi}(u)v\right]$$
$$= \mathbb{E}_v\left[\frac{p}{\delta}\tilde{\Phi}(u+\delta v)v\right],$$

where we use the fact that $\mathbb{E}_v[v] = 0$ for $v \sim \mathcal{N}(0, \frac{1}{p}I)$, and we shall design the iterative updates such that u is independent of v (see (48) below). The implication is that the one-point estimate

$$\phi^{(1)}(u) = \frac{p}{\delta}\tilde{\Phi}(u+\delta v)v \tag{46}$$

suffices to be a sensible estimate of the gradient (Flaxman et al., 2005), because its expectation equals that of the two-point estimate $\phi^{(2)}(u)$. Nonetheless, one issue of $\phi^{(1)}(u)$ is its potentially high variance. The main reason is that the magnitude of $\phi^{(1)}(u)$ depends on the objective value $\tilde{\Phi}(u+\delta u)$, whose range of change may be large. To solve this issue, we can subtract the term $\tilde{\Phi}(u+\delta u)$ in $\phi^{(1)}(u)$ with an appropriate value that is independent of v, thus reducing the variance of the gradient estimate. Notably, Zhang et al. (2022) proposes the following one-point residual-feedback estimate

$$\tilde{\phi}^{(1)}(u) = \frac{p}{\delta}\left(\tilde{\Phi}(u+\delta v) - \tilde{\Phi}(u'+\delta v')\right)v, \tag{47}$$

which subtracts (46) with the objective value at the previous point $u'+\delta v'$ during iterations, and $v' \sim \mathcal{N}(0, \frac{1}{p}I)$ is independent of v. Furthermore, Chen et al. (2022) exploits insights into high-pass and low-pass filters in extremum seeking (Ariyur & Krstic, 2003) and presents a unifying structure for one-point estimates. This structure encompasses residual feedback for variance reduction and a momentum term for acceleration.

Motivated by the above observations, we consider the model-free feedback-based optimization scheme (He et al., 2023) illustrated by Fig. 12. The closed-loop dynamics are as follows

$$\text{plant}\begin{cases} \zeta_{k+1} &= f(\zeta_k, u_k) \\ y_k &= g(\zeta_k, d) \end{cases}$$

$$\text{controller}\begin{cases} w_{k+1} &= w_k - \eta\phi_k \\ \phi_k &= \frac{p}{\delta}\left(\Phi(y_{k+1}) - \Phi(y_k)\right)v_k \\ u_{k+1} &= w_{k+1} + \delta v_{k+1}, \end{cases} \tag{48}$$

where $\eta > 0$ is the step size, $k = 0, 1, \ldots$, and $v_0, v_1, \ldots$ are independent and identically distributed random vectors drawn from $\mathcal{N}(0, \frac{1}{p}I)$. In (48), w_{k+1} is the candidate solution and is iteratively updated based on the gradient estimate ϕ_k at the previous solution w_k. The estimate ϕ_k is similar to (47). The difference is that due to real-time operations, we substitute the steady-state output $h(u_k, d)$ with the measurement output y_{k+1} after applying the input u_k. Consequently, $\Phi(y_{k+1})$ is an approximation of the true objective value $\tilde{\Phi}(u_k) = \Phi(h(u_k, d))$. Afterward, the controller adds an exploration noise δv_{k+1} to w_{k+1} and applies the input u_{k+1} to the plant. To establish the stability of the closed-loop interconnection (48), we can extend the timescale separation argument discussed in Section 3.1.1. A further extension to handle input constraints via Frank-Wolfe type updates is also available (He et al., 2023). ■

By and large, model-free methods in Example 3.6 are less sample-efficient than feedback-based optimization schemes with perfect sensitivities (Bernstein et al., 2019; Colombino et al., 2020; Hauswirth, Bolognani, Hug, & Dörfler, 2020). In other words, they require relatively more actuation steps to reach a certain solution accuracy, see Fig. 13 for a detailed comparison. One intuition is that model-free methods drop the useful structural information contained in prior knowledge or approximate models and instead rely on stochastic exploration. This stochasticity may influence the overall convergence rates and cause an increased number of actuations. Another interpretation is that not all history is taken into account at every iteration, which differs from the recursive estimation of sensitivity in Section 3.3.1. This issue may be a disadvantage when the sensitivity is almost constant.

Bayesian optimization-based schemes. Another line of model-free feedback-based optimization schemes is built on Bayesian optimization (Frazier, 2018; Shahriari et al., 2016). The key idea is to maintain probabilistic surrogate models of unknown objectives and constraints based on samples, and then set the next query as the optimal point of an acquisition function. This update mechanism is shown to be sample-efficient and hence attractive for applications where data collection is costly (Neumann-Brosig et al., 2020; Simonetto et al., 2021). Another appealing part is that black-box constraints are satisfied with a high probability (Krishnamoorthy & Doyle III, 2023). It benefits from the design that balances exploration and exploitation by judiciously expanding a safe region and selecting samples from this region (Krishnamoorthy & Doyle, 2022; Sui et al., 2015). The closed-loop implementation requires the detection of steady states (Krishnamoorthy & Doyle III, 2023), which implies a complete time-scale separation. Furthermore, it is possible to incorporate contextual information to better adapt to time-varying external conditions (Xu et al., 2023).

In general, Bayesian optimization-based methods are mostly suitable for low-dimensional systems. Otherwise, the computational cost of using Gaussian processes as surrogate models of objectives and constraints is significant (Shahriari et al., 2016).

Remark 3.4. The aforementioned model-free schemes utilize exact function evaluations at perturbed candidate solutions. In some applications (e.g., optimization of personalized utility functions with user's feedback (Notarnicola et al., 2023; Simonetto et al., 2021)), only infrequent and noisy evaluations are available. To address this issue, we can construct approximations based on parametric estimation with basis functions (Cothren et al., 2022a) or neural networks (Cothren et al., 2022b). Then, we optimize these approximations via iterative descent algorithms. In this line of work, the considered objective function is decoupled and contains two parts that depend on the input and the output. Approximations are separately constructed for these two parts. Therefore, these schemes still require knowing steady-state input–output sensitivities. ∎

4. Application example: Optimal reserve dispatch in electricity grids

So far, we have explored small educational examples illustrating various individual concepts and special cases. In this section, we illustrate the potential of these methods for the emerging problem of optimal real-time operation of future power systems in the face of increasing intermittent renewable energy generation.

Future power systems will face less predictable and more volatile conditions, which require new control and decision protocols to continue operating power grids safely and efficiently. In a simplified and idealized fashion, this task of optimal real-time grid operation can be cast as tracking the solution of a (time-varying) ACOPF problem (Frank et al., 2012a, 2012b; Huneault & Galiana, 1991). This perspective on the problem has motivated and guided an essential part of the recent research on online feedback optimization and has contributed to defining many of the methodological questions that we addressed earlier in this article.

Although solving ACOPF problems in real-time has been proposed for several decades, the idea of driving the state of a power grid to the solution of an ACOPF by interconnecting an optimization dynamic in closed loop with the physical grid is much more recent. In the context of distribution grids, the idea of using iterative measurements (rather than a model) to determine a gradient direction and ultimately implement an iterative optimization algorithm was first explored in Bolognani and Zampieri (2013) and in Farivar et al. (2013), with a focus on deriving distributed and decentralized control strategies. The idea of employing gradient-based online algorithms in closed loop with the power grid to drive the system to the solution of a given ACOPF was further explored (with the addition of constraints, time-varying parameters, and more general problem formulations) in Bernstein and Dall'Anese (2019), Bernstein et al. (2015), Bolognani et al. (2015), Gan and Low (2016), Mazzi et al. (2018), Reyes-Chamorro et al. (2015), Tang et al. (2017) among others.

A connected line of research, also motivated by power flow optimizations in the electricity grid, is based on the insights of Bolognani and Dörfler (2015) that identifies the solutions space as a smooth manifold. Using this perspective, Hauswirth et al. (2016) proposes a (continuous-time) gradient-based controller that results in a differential–algebraic system. This algorithm (in a discretized form) was applied to ACOPF problems in Hauswirth et al. (2017), and formal convergence guarantees of the discretization were established in Häberle et al. (2021). While considering the solution space of the AC power flow equations as a smooth manifold (rather than assuming an input–output map) adds to the conceptual complexity, it also highlights underlying challenges relating to solvability, controllability, and the possibility for bifurcations.
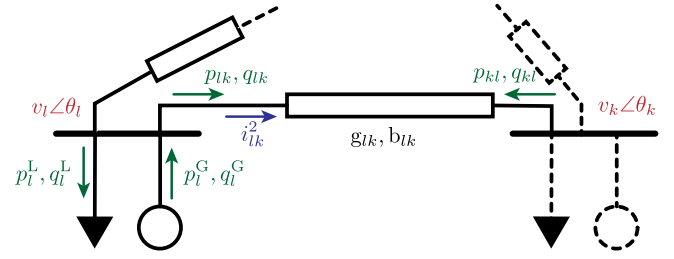

Fig. 14. Physical AC power flow quantities.

Besides purely gradient-based schemes, saddle-point algorithms (as defined in Section 2.3) have also been extensively studied in the context of real-time feedback-based power system optimization. However, the AC OPF problem's non-convexity makes formal convergence guarantees difficult to establish. For this reason, Dall'Anese and Simonetto (2018) resorts to a dual augmentation term, which improves convergence but perturbs the equilibria such that they do not necessarily correspond to critical points of the optimization problem anymore.

Further research on this topic has then produced several contributions, and we refer to Section 4 in Molzahn et al. (2017) for a review of these works. This feedback-optimization approach to the real-time operation of power grids has reached a remarkable level of technological maturity (Kroposki et al., 2020) and has been validated both in laboratory experiments (Ortmann, Hauswirth, et al., 2020; Rodríguez-Gonzalez et al., 2022) and in real-world deployments (Ortmann et al., 2023).

In the following, we do not delve into the peculiarities of any of these implementations; instead, we briefly describe one possible version of this problem, we show how the problem can be brought into the general form we studied in the previous sections, we apply the results and algorithms we reviewed, and we discuss why a feedback-based optimization solution is so successful in this context.

4.1. Formulation as a real-time optimal power flow problem

Consider an AC power transmission network as an undirected graph with a set $\mathcal{N}$ of buses and a set $\mathcal{M}$ of transmission lines. We denote by $(l, k) \in \mathcal{M}$ a line connecting bus l to bus k. As decision variables, each bus $l \in \mathcal{N}$ has an associated voltage magnitude v_l, voltage angle θ_l, and active and reactive power generation $p_l^{\mathrm{G}}, q_l^{\mathrm{G}}$. By $p^{\mathrm{G}}, q^{\mathrm{G}}$, etc. we denote the vectors obtained from stacking all respective components p_l^{G} for all $l \in \mathcal{N}$.

For simplicity, we assume that the objective is given as the cost $\Phi(p^{\mathrm{G}}) = \sum_{l \in \mathcal{N}} \Phi_l(p_l^{\mathrm{G}})$ for active power generation as the minimization objective.

Hence, a basic ACOPF problem is given by

$$\underset{v, \theta, p^{\mathrm{G}}, q^{\mathrm{G}}}{\text{minimize}} \quad \Phi(p^{\mathrm{G}}) \tag{49a}$$

$$\text{subject to} \quad 0 = p_l^{\mathrm{G}} - p_l^{\mathrm{L}} - \sum_{l \to k} p_{lk}(v_l, v_k, \theta_l, \theta_k) \tag{49b}$$
$$\forall l \in \mathcal{N}$$

$$0 = q_l^{\mathrm{G}} - q_l^{\mathrm{L}} - \sum_{l \to k} q_{lk}(v_l, v_k, \theta_l, \theta_k) \tag{49c}$$

$$\underline{p}_l \leq p_l^{\mathrm{G}} \leq \overline{p}_l \tag{49d}$$

$$\underline{q}_l \leq q_l^{\mathrm{G}} \leq \overline{q}_l \tag{49e}$$

$$\underline{v}_l \leq v_l \leq \overline{v}_l \tag{49f}$$

$$i_{lk}^2(v_l, v_k, \theta_l, \theta_k) \leq \overline{i}_{kl}^2 \quad \forall (l, k) \in \mathcal{M}, \tag{49g}$$

The inequalities (49d)–(f) denote box constraints on power generation and voltages at each bus, and (49g) limits the current that flows through the line from k to ℓ.

The functions $p_{lk}(\cdot), q_{lk}(\cdot)$ and $i_{lk}^2(\cdot)$ denote active and reactive power flow and squared current magnitude from bus l in direction of bus k, respectively, as illustrated in Fig. 14. These nonlinear terms depend on line parameters and constitute the main source of complexity in ACOPF problems. Their explicit expressions (neglecting the so-called *shunt admittances* of the buses) are given by

$$p_{lk}(v_l, v_k, \theta_l, \theta_k) := v_l v_k \left(\mathrm{g}_{lk} \cos(\theta_l - \theta_k) + \mathrm{b}_{lk} \sin(\theta_l - \theta_k) \right)$$
$$q_{lk}(v_l, v_k, \theta_l, \theta_k) := v_l v_k \left(\mathrm{g}_{lk} \sin(\theta_l - \theta_k) - \mathrm{b}_{lk} \cos(\theta_l - \theta_k) \right)$$
$$i_{lk}^2(v_l, v_k, \theta_l, \theta_k) := (\mathrm{g}_{lk}^2 + \mathrm{b}_{lk}^2) \left(v_k^2 + v_l^2 - 4 v_k v_l \cos(\theta_k - \theta_l) \right)$$

where g_{lk} and b_{lk} denote the *conductance* and *susceptance* of the transmission line connecting buses l and k. Note that $i_{lk}^2 = i_{kl}^2$ holds (without shunt elements), but $p_{lk} = -p_{kl}$ and $p_{lk} = -p_{kl}$ are not generally true. For a more comprehensive introduction to AC power flow including alternative formulations the reader is referred to Frank and Rebennack (2016), Molzahn and Hiskens (2019).

ACOPF problems like (49) are well-studied and routinely solved (numerically) in practice. Yet, they remain computationally demanding, because of the nonlinear power flow equations (49b)–(c) which render the problem non-convex (although convex relaxations can yield global optimality certificates in some cases; Low 2014, Molzahn and Hiskens 2019).

4.2. Real-time operation as closed-loop optimization

In an online setting, (49) defines the most desirable operating point of the grid (its power flows and its power injections p^G, q^G) given the actual power consumption p^L, q^L (as opposed to predictions used in offline day/hour-ahead calculations). The power consumption p^L, q^L and other parameters in Eq. (49) are generally time-varying. Hence, rather than solving (49) once for a given configuration, a real-time-operation strategy needs to track the ACOPF problem's solution across time.

From a theoretical viewpoint, robustly tracking solutions of (49) presents important challenges:

(i) Equations (49b)–(c) are the steady-state equations of the complicated, interconnected nonlinear dynamics of transmission lines, generation units, loads, and their low-level controllers. Although these dynamics can be assumed to be asymptotically stable, attempts at steering the physical system too aggressively towards a solution of (49) can destabilize them.

(ii) In general, only an approximate model in the form of the steady-state AC power flow equations (49b)–(c) can be employed. A dynamical model for a large-scale power system is not available, as different stakeholders own parts of the system, models are proprietary, operating conditions (e.g., which generation units are online) change over time, and internal states are typically inaccessible. Steady-state models have significant parameter uncertainty and are affected by disturbances.

(iii) Many physical and engineering constraints of different nature have to be satisfied. Some of these limits, like the generation constraints (49b)–(c), are physical constraints that are strictly enforced by lower-level controllers or through saturation. Other constraints, such as the line flow limits Eq. (49g), are thermal limits that can be violated temporarily. Yet other constraints are hard limits that must not be violated at any time. This mainly concerns dynamic and voltage stability limits, which we will not consider in this simplified setting. Simply note that the voltage constraints Eq. (49f) can be understood as guarding against voltage instability (Van Cutsem & Vournas, 1998).

(iv) The nonlinear nature, especially under critical operating conditions, of the power flow equations (49b)–(c) call for methods that work in the absence of convexity and for ill-conditioned problems.
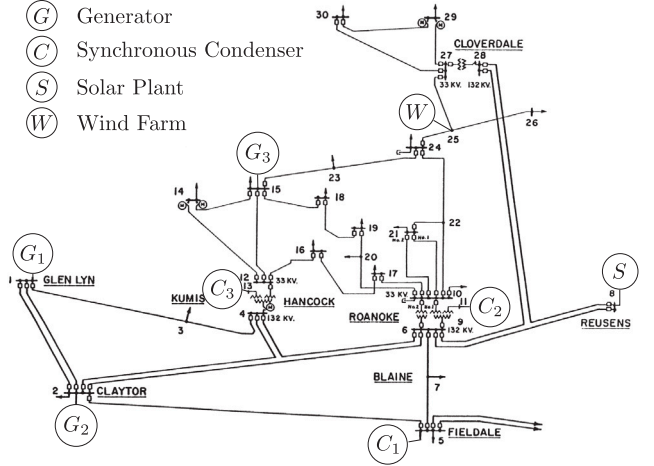

Fig. 15. Grid diagram of adapted 30-bus power systems test case used in Examples 4.1 and 4.2.

With this in mind, we argue that the online feedback-based optimization methods that we reviewed are very well suited for this task. From a practical perspective, "closing the loop" on the ACOPF problem (49) offers an opportunity to more closely integrate and combine different power system control tasks and thereby increase economic efficiency, resilience, and autonomy. More concretely, a controller tracking the solution of (49) makes complex yet economically efficient re-dispatch decisions in response to unscheduled events to guarantee voltage and line flow limits are respected. Thus, (49) should be interpreted as a "residual" optimization problem around a pre-planned generation schedule, and constraints (49d)–(e) do not necessarily quantify the full production capacity of a generation unit, but rather the amount of dispatchable reserves.

In order to express (49) as a problem of the form (21), we need to identify inputs, outputs, and disturbances. For this purpose, for each generation unit, the active power output and either the reactive power or voltage magnitude are assumed to be controllable[5] and thus make up the control input u. The loads p^L, q^L define the disturbance d. All remaining quantities form the output y.

Consequently, (49) can be brought into the form (21) where the constraints (49d)–(g) are assigned to either $\mathcal{U}$ or $\mathcal{X}$ according to whether they apply only to controllable variables or not. Under normal operating conditions, the local existence and differentiability of the steady-state map $h(\cdot)$ derived from (49b)–(c) is guaranteed by the implicit function theorem (Bolognani & Dörfler, 2015).

The following two examples illustrate how feedback-based optimization can be applied to the optimal reserve dispatch problem.

Example 4.1. To track the time-varying solution of Eq. (49), we adopt the feedback controller (36) that implements a projected and primal augmented saddle-point flow.

As a numerical testbed, we use a modified version of the IEEE 30-bus power systems test case adopted from Hauswirth (2020), Hauswirth et al. (2017) and illustrated in Fig. 15. In particular, a wind and a solar plant provide intermittent generation capacity in addition to three conventional generators. The available solar and wind generation capacity profiles are illustrated in Fig. 16. One of the conventional generators suffers an outage at 4:00 upon which its generation capacity is set to

---

[5] One distinguishes between so-called *PQ-* and *PV*-buses. See Hauswirth et al. (2017) for a more detailed discussion including the role of the *slack bus* in numerical simulations).
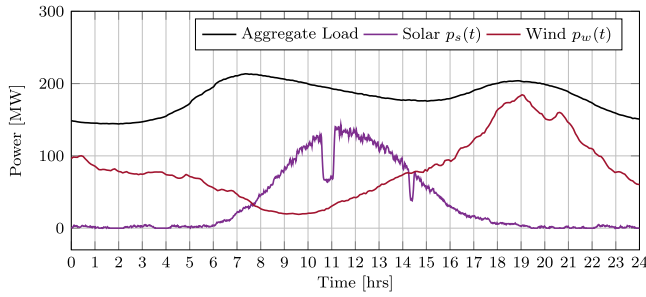
Fig. 16. Load and available generation profiles for the modified 30-bus test case in Examples 4.1 and 4.2 over a 24 h horizon.

zero for the remainder of the simulation. The goal is to optimally use renewable resources without violating engineering constraints such as voltage limits and thermal line ratings.

We implement (36) as a discretized controller using a simple explicit Euler discretization with a step size corresponding to 1-minute intervals between control actions. The resulting control law is similar to the designs proposed in Bernstein et al. (2019), Dall'Anese and Simonetto (2018), Tang et al. (2018, 2022). The physical system is simulated without dynamics, and an off-the-shelf AC power flow solver (Zimmerman et al., 2011) is used to compute a solution to the ACPF equations (49b)–(c) and thus evaluate the input–output map $h(u)+d$. We further compute $\nabla h(u)$ based on input and output measurements and the power flow model in (49) (rather than an explicit estimate of d). Finally, we choose $Q(u) := H(u)^T H(u)$ as a metric. This choice of "implicit" metric has proven to increase numerical stability in the face of an ill-conditioned input–output map. For a more detailed discussion and motivation, see Chapter 14 in Hauswirth (2020).

The left panel in Fig. 17 illustrates the (almost perfect) performance achieved by this feedback-based optimization approach in terms of cost compared to the a-posteriori sequential solution of the ACOPF problem (which is based on omniscient and perfect information).

More importantly, however, the left panel in Fig. 18 shows that, over the entire simulation horizon, constraint violations are very minor and only temporary. The controller achieves this by jointly managing active and reactive power infeed to manage voltage magnitudes and line currents. In particular, both solar and wind generation have to be curtailed to prevent line overloads. ∎

Example 4.2. We reconsider the numerical test case from Example 4.1, but apply the feedback control strategy Eq. (39) instead of (36). Again, we choose a sampling period (corresponding to the step size α) of 1 min. As before we approximate $\nabla h(u)$ based on input and output measurements and the power flow model, and we choose $Q(u) := H(u)^T H(u)$ as the metric.

The right-hand side panels in Figs. 17 and 18 show how this linearized output projection method achieves similar performance to the discretized projected saddle-point flow controller Eq. (36).

In fact, the constraint satisfaction is even superior, as output constraints such as line flow limits are directly enforced. Whereas for the saddle-point transient constraint violation is generally unavoidable to achieve convergence to non-zero dual variables at an equilibrium, the linearized output projection method predicts (and avoids) constraint violations based on first-order model information. This effect is illustrated in Fig. 19 where the enforcement of a line flow constraint from the previous simulations is compared. It is readily noticeable that the augmented projected saddle-point flow exhibits transient constraint violation and only asymptotically satisfies the constraint. This observation corresponds to the mechanism illustrated in Fig. 6(d). In contrast, the linearized output projection exhibits almost no transient constraint violation and this close in behavior to the projected gradient flow it approximates (see also Fig. 6(e)), at the cost of slightly higher

computational complexity (a quadratic program needs to be solved at each iteration).

Finally, we highlight that the feedback controller (39) has gone beyond numerical tests and reached real-world industrial deployment. Ortmann et al. (2023) apply the controller (39) to optimize reactive power flows of a real distribution grid subject to voltage constraints. The deployed controller features compatibility with the existing grid infrastructure and 24/7 uninterruptible operation. ∎

5. Conclusions and outlook

In this article, we have surveyed various approaches for solving optimization problems in closed loop with a physical system. As a particular focus, we have interpreted classical numerical optimization algorithms as dynamical systems, and we have studied the conceptually simple idea of interconnecting these algorithms with asymptotically stable plants with well-defined input–output behavior.

Solving optimization problems online and with feedback, rather than offline and in open loop, leads to greater robustness against uncertain problem data and reduces required model information and computational effort because the physical system inherently enforces certain constraints. The design of such controllers is aided by the recent rediscovery of and interest in the dynamical systems perspective of optimization algorithms. However, in contrast to analyzing existing algorithms with tools from dynamical systems theory, implementing these algorithms as feedback controllers raises novel and unique challenges, such as their closed-loop stability and robustness. Feedback-based optimization calls for smart control designs that meaningfully exploit the physical properties of a system, such as its steady-state response and saturation effects. We have particularly delved into the available guarantees of robustness and constraint satisfaction.

5.1. Ongoing and future research avenues

Even though feedback-based optimization has been an active research area for some time, many questions remain unanswered. To complete this article, we propose two worthwhile avenues for future inquiry.

General equilibrium seeking in closed loop

In this article, we have focused on solving optimization problems and thus steering a physical plant to a state that solves the corresponding KKT conditions. This setup is motivated by many applications that come with steady-state specifications regarding global plant efficiency, running cost, yield, etc. Instead of optimality conditions, it is also possible to design controllers that seek states that satisfy more general equilibrium conditions, such as solving variational inequalities or generalized equalities. It thus becomes possible, for example, to design feedback controllers in a non-cooperative setting that allow players to converge to a Nash equilibrium (De Persis & Monshizadeh, 2019). The closed-loop structure and the use of online measurements also help to achieve robust tracking of time-varying Nash equilibria due to changing disturbances and problem parameters (Belgioioso et al., 2021, 2022).

Time-varying non-convex optimization

As discussed in Section 2.5, the study of time-varying optimization problems has so far focused on convex problems (often with a strongly convex objective). This offers the convenience of a unique global optimizer that forms a well-defined trajectory over time and can be tracked by appropriate controllers and algorithms.

Solving non-convex problems, such as the optimal reserve dispatch problem from Section 4, in closed loop poses a much more challenging problem. Stationary non-convex problems are generally hard to solve due to the existence of multiple local minimizers. Time-varying non-convex problems pose additional problems since the possible appearance and disappearance of minimizers make it generally impossible
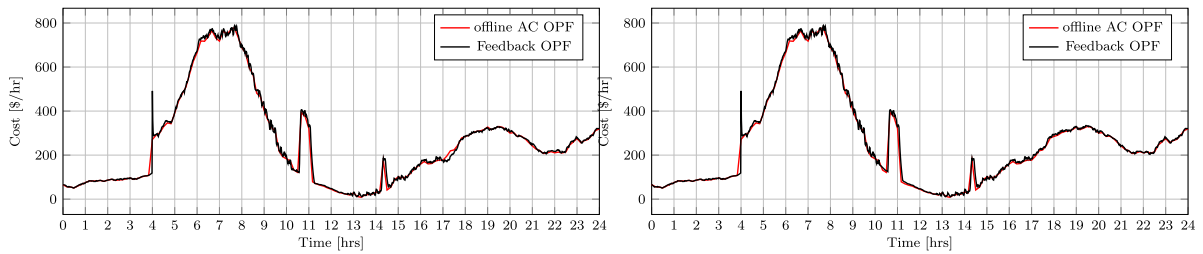
Fig. 17. Cost realized by feedback-based optimization in comparison to a-posteriori ACOPF solutions using perfect information. Left: cost achieved by the projected augmented saddle-point flow (36); right: cost achieved by the linearized output projection method Eq. (39).
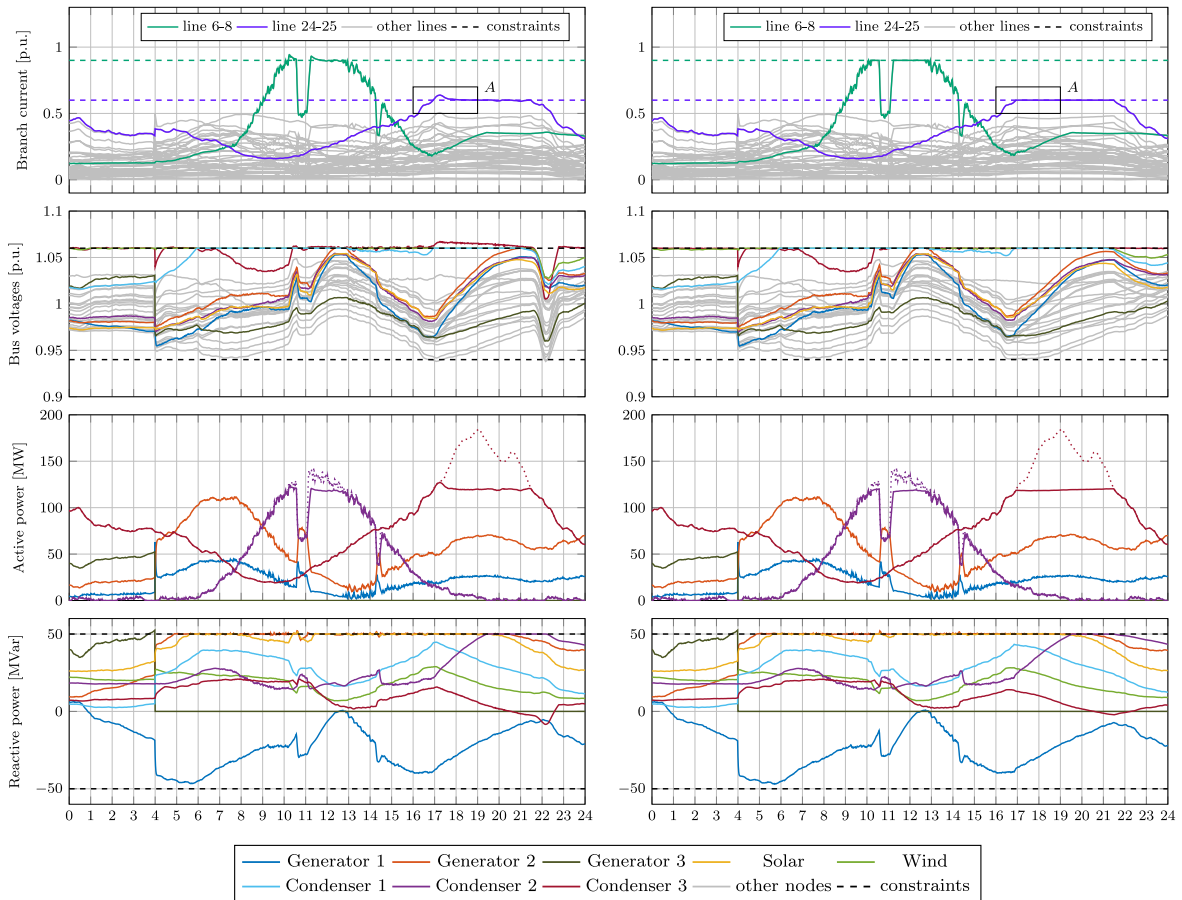

Fig. 18. State trajectories realized by feedback-based optimization. Left: trajectories of the projected augmented saddle-point flow (36); right: trajectories of the linearized output projection method Eq. (39). See Fig. 19 for a discussion of the inset A.
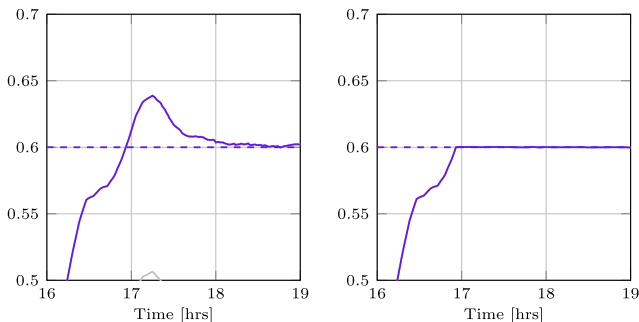

Fig. 19. Comparison of line flow limit violation and enforcement (Detail A in Fig. 18); Left: projected saddle-point flow; Right: linearized output projection;.

to track a single global minimizer continuously. Nevertheless, it is a practical necessity to understand how feedback-based optimization performs in non-convex settings and to develop controllers that can cope with such pathologies. Recent papers exploring this direction include Ding et al. (2021), Lin et al. (2020).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Historical/pioneering applications

A.1. Network congestion control

Arguably one of the largest man-made distributed feedback systems is formed by the congestion management mechanisms at the heart of the Internet (Low et al., 2002). These protocols manage the allocation of link capacities to individual connections in highly dynamic environments, for heterogeneous agents, and subject to real-world imperfections such as delays, among others (Paganini et al., 2005; Tang et al., 2007; Vinnicombe, 2002; Wen & Arcak, 2004). Deterministic, continuous-time flow models have been particularly successful in explaining these protocols, by providing an optimization-based perspective in terms of network utility maximization (Kelly et al., 1998; Low, 2017; Low & Lapsley, 1999; Shakkottai & Srikant, 2007), and enabling improved designs (Wei et al., 2006, 2013; Zargham et al., 2013).

Given a communication network, let a set of N sources share M links. The set of links used by each to communicate to its destination is collected in the routing matrix $R \in \mathbb{R}^{N \times M}$ in the sense that $R_{ij} = 1$ if link j is used by source i and $R_{ij} = 0$ otherwise. Each link j in the network has an associated congestion measure μ_j (also referred to as *price*) which may describe queuing delays or packet loss probabilities. Each link j has a finite capacity c_j. Each source i has a controllable *source rate* x_i, e.g., in the form of its *transmission window size*. The different source rates define a vector of aggregate flows for each line given by $y = Rx$ where each row of R corresponds to a link. Conversely, sources are assumed to have access to their respective *aggregate price* $p = R^T \mu$. In practice, each source can estimate the total price over its own path by measuring delays, estimating the packet loss probability, or detecting the presence of specific "congestion markers" on its packets.

The simplest model to describe the *link dynamics* is given by the *projected gradient* flow

$$\dot{\mu}_j = \Pi_{\mathbb{R}_{\geq 0}^M}[y_j - c_j](\mu_j) \tag{A.1}$$

where

$$\Pi_{\mathbb{R}_{\geq 0}^M}[y_j - c_j](\mu_j) = \begin{cases} y_j - c_j & \mu_j = 0 \text{ and } y_j > c_j, \\ 0 & \text{otherwise.} \end{cases}$$

In other words, the price increases when the link is overloaded (and packets are lost) and decreases whenever it is not overloaded, but the price never drops below zero.

Source controllers can often be modeled as $\dot{x}_i = -\nabla \Phi_i(x_i) - p_i$, where $\Phi_i(x_i)$ is a specific type of cost function, reverse-engineered and depending on the particular protocol. Namely, sources adapt their rate to minimize a given cost function (equivalently, maximize their utility) corrected by the aggregate price.

Hence, congestion control mechanisms interconnect source controllers and link dynamics into a feedback loop as illustrated in Fig. A.20, where $\Phi(x) := \sum_i \Phi_i(x_i)$. Importantly, each controller is fully distributed and requires only locally available information. Written more compactly, the closed-loop dynamics are defined as

$$\dot{x} = -\nabla \Phi(x) - R^T \mu \qquad \dot{\mu} = \Pi_{\mathbb{R}_{\geq 0}^M}[Rx - c](\mu).$$

This system defines a *projected saddle-point flow*, as in Example 2.3, whose trajectories converge to a solution of

$$\text{minimize} \quad \Phi(x) \qquad \text{subject to} \quad Rx \leq c, \tag{A.2}$$
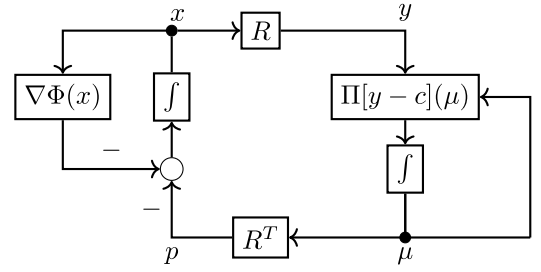

Fig. A.20. Simple network utility maximization.

whenever Φ is strictly convex.

From a control and optimization perspective, source controllers and link dynamics form a closed feedback loop that implicitly tracks the solution of the underlying utility maximization problem (Wang & Elia, 2011).

From the viewpoint of closed-loop optimization advocated in this article, one can argue that the links implement the dual dynamics that have to be complemented by controllers that form the primal dynamics. Together they optimize Eq. (A.2).

A.2. Optimal frequency control

Recent years have seen renewed interest in the control of power systems because of the new challenges associated with an increasingly dominating number of power electronic devices connected to the electricity grid and the growth of highly intermittent infeed from new renewable energy sources.

Frequency control in AC grids is one control task that has been revisited in this context. In this example, we illustrate how recent work has framed this challenge as a closed-loop optimization problem in order to improve upon classical frequency control schemes.

The reader is referred to Molzahn et al. (2017) for a recent survey on this topic and to Chen, Zhao, and Li (2020), Simpson-Porco (2021b), Zhao et al. (2016) and references therein for latest results. The ensuing model is based on Li et al. (2016).

In contrast to the optimal reserve dispatch problem outlined in Section 4, we assume a *lossless* AC power transmission network with a set $\mathcal{N}$ of buses and a set $\mathcal{M}$ of transmission lines. We consider the power system around an operating point and all quantities are to be interpreted as deviations from their nominal values.

The frequency deviation ω_j at every bus j is governed by the *swing equation*

$$\dot{\omega}_j = \frac{1}{M_j}\left(p_j^M - D_j\omega_j - p_j^L - \sum_{k:j\to k} p_{jk}\right) \tag{A.3}$$

where M_j is the inertia of the generator and D_j is the damping constant at bus j. The mechanical power output of the generator and the power consumed at bus j are denoted by p_j^M and p_j^L, respectively. The power flowing out of bus j toward an adjacent bus k is written as p_{jk}. The notation $j \to k$ indicates a transmission line between buses j and k, i.e., $(j, k) \in \mathcal{M}$.

For every $j \to k$, *line flow dynamics* are linearized and modeled as

$$\dot{p}_{jk} = b_{jk}(\omega_j - \omega_k) \tag{A.4}$$

where b_{jk} is the line susceptance. Note that transmission lines are undirected, and, throughout, the constraint $p_{jk} = -p_{kj}$ holds for every line implicitly.

At steady state and nominal frequency $\omega_j = 0$, the swing Eq. (A.3) expresses the power balance at each node, and (A.4) yields the so-called *DC-flow* approximation of the AC power flow equations (assuming nominal voltages, lossless lines, and small angle differences).

The mechanical power output p_j^M is described by a simplified *governor-turbine control* model of the form

$$\dot{p}_j^M = -\frac{1}{T_j}\left(p_j^M - p_j^C + \frac{1}{R_j}\omega_j\right) \tag{A.5}$$

where p_j^C is a power adjustment signal and T_j and R_j are constants. In particular, R_j is understood as a generator's participation factor in primary frequency control.

The power adjustment itself is evaluated using the so-called *area control error* (ACE) which combines frequency and line flow deviations. More specifically, we have

$$\dot{p}_j^C = -K_j\left(B_j\omega_j + \sum_{k:j\to k} p_{jk}\right). \tag{A.6}$$

with B_j and K_j being constant control parameters.

The role of this type of *automatic generation control* (or *secondary frequency control*) is to react to changes in p_j^L, driving frequency deviation to zero, and to make sure that deviations in p_j^L are compensated by the local generator through adjustments of p_j^M. Consequently, at steady state, the deviations P_{ij} of the line flows from their nominal values are zero.[6]

Although not immediately obvious, the system (A.3)–(6) realizes a (partial) saddle-point flow, as introduced in Section 2.3. To see this, we consider the optimization problem

$$\underset{p^M,\omega,\{p_{jk}\}_{jk}}{\text{minimize}} \quad \sum_{j\in\mathcal{N}}\left(\frac{\beta_j}{2}\left(p_j^M\right)^2 + \frac{D_j}{2}\omega_j^2\right)$$
$$\text{subject to} \quad 0 = p_j^M - p_j^L$$
$$\phantom{\text{subject to}}_{\forall j\in\mathcal{N}} \quad 0 = p_j^M - D_j\omega_j - p_j^L - \sum_{k:j\to k} p_{jk}. \tag{A.7}$$

where β_j is a positive parameter to be determined. The solution of this problem guarantees power balance and exact frequency regulation. We can write out the corresponding (unprojected) primal–dual saddle-point flow from Example 2.2 as

$$\text{primal}\begin{cases} \dot{\omega}_j &= -\epsilon_{\omega_j}\left(D_j\omega_j - D_j\lambda_j\right) \\ \dot{p}_j^M &= -\epsilon_{P^M}(\beta_j p_j^M - \mu_j - \lambda_j) \\ \dot{p}_{jk} &= -\epsilon_{p_{jk}}(\lambda_j - \lambda_k) \end{cases}$$
$$\text{dual}\begin{cases} \dot{\mu}_j &= \epsilon_{\mu_j}(p_j^M - p_j^L) \\ \dot{\lambda}_j &= \epsilon_{\lambda_j}\left(p_j^M - D_j\omega_j - p_j^L - \sum_{k:j\to k} p_{jk}\right) \end{cases} \tag{A.8}$$

for all $j,k \in \mathcal{N}$ and $j\to k$, and where we use separate positive gains $\epsilon_{(\cdot)}$ for every variable.

By taking the limit $\epsilon_{\omega_j}\to\infty$, we can replace the corresponding ODE with the algebraic expression $0 = D_j(\omega_j - \lambda_j)$ and thus substitute $\omega_j = \lambda_j$. This results in a so-called *partial* saddle-point flow. In this special case, the ω_j can hence be interpreted as a primal or dual variable.

We further apply a transformation to μ_j by defining $p_j^C := K_j\left(M_j\omega_j - \frac{1}{\epsilon_{\mu_j}}\mu_j\right)$. Consequently, we have

$$\dot{p}_j^C = K_j\left(M_j\dot{\omega}_j - \frac{1}{\epsilon_{\mu_j}}\dot{\mu}_j\right) = K_j\left(M_j\dot{\omega}_j - (p_j^M - p_j^L)\right).$$

Combining these insights, we can rewrite the partial saddle-point flow derived from (A.8) as

$$\text{primal}\begin{cases} \dot{p}_j^M &= -\epsilon_{P_i}(\beta_j p_j^M - \mu_j - \omega_j) \\ \dot{p}_{ij} &= -\epsilon_{P_{ij}}(\omega_k - \omega_j) \end{cases}$$
$$\text{primal or dual}\begin{cases} \dot{\omega}_j &= \epsilon_{\lambda_j}\left(p_j^M - D_j\omega_j - p_j^L - \sum_{k:j\to k} p_{jk}\right) \end{cases}$$
$$\text{dual}\begin{cases} \dot{p}_j^C &= -K_j\left(D_j\omega_j + \sum_{k:j\to k} p_{jk}\right). \end{cases} \tag{A.9}$$

By carefully choosing the remaining gains $\epsilon_{(\cdot)}$ and β_j (e.g., $\epsilon_{P_{ij}} := b_{ij}$ or $\omega_{\lambda_j} = \frac{1}{M_j}$) as well as, under $B_j = D_j$, (A.9) is equal to (A.3)–(6). The choice of $B_j = D_j$ is fragile, since it means that the controller (A.6) requires an accurate estimate of the damping D_j at every bus. However, numerical experiments in Li et al. (2016) show that (A.3)–(6) is stable even if $B_j = D_j$ does not hold exactly.

Starting from this observation that automatic generation control can be interpreted from a closed-loop optimization perspective, various works have proposed optimal frequency control schemes that allow for more general objective functions than in Eq. (A.7), relax the constraint $p_j^M = p_j^L$, or incorporate additional constraints on generation and transmission capacity.

Appendix B. Related research streams

The key idea studied in this article is the interconnection of optimization algorithms with physical systems. The architecture that we discussed (exemplified in Example 1.1) is by no means the only meaningful way to combine optimization with feedback control. This section provides a pedagogical overview of various other well-established approaches in this direction. Our treatment is deliberately simplified to highlight the fundamental differences. The reader is referred to respective survey papers for exhaustive reviews of the topics treated in this section.

We focus our attention on the following three methods.

(i) *Extremum seeking*, which is rooted in adaptive control, emphasizes being completely *model-free* and probing the system with the help of a perturbation signal. However, the approach is best suited for systems with low-dimensional inputs and without complicated engineering constraints.

(ii) Emerging from process engineering, the primary merit of the *modifier adaptation* method is to mitigate the effects of model bias when solving successive optimization problems. This method does not reduce the computational requirements compared to feedforward optimization.

(iii) *Real-time iteration* schemes, rooted in model predictive control, aim to solve classical receding horizon problems with limited resources. The focus is the stabilization of a plant under state and input constraints. For this purpose, a full model of the plant dynamics is generally required, and the computational burden scales with the planning horizon.

Many more approaches can be interpreted as optimization in conjunction with feedback control. Several topics such as *iterative feedback tuning* (Hjalmarsson, 2002) or *iterative learning control* (Bristow et al., 2006) are concerned with the optimal tuning of controllers by either requiring a sequence of experiments or repetitive operation. Although these methods use measurement data to optimize subsequent control actions, because of their "episodic" nature, they are related only remotely to the key idea in this article.

For similar reasons, we do not review recent work on *reinforcement learning*, even though it has produced staggering results, especially in sequential decision-making for games and in robotics. Even though techniques such as *policy gradient* admit a dynamical perspective, RL is generally (and traditionally) framed as optimal control over Markov decision processes (Bertsekas, 2017; Lewis & Liu, 2012).

---

[6] In this example, the area control error is evaluated and regulated on a per-bus basis. In more general settings, multiple buses belong to the same area, and therefore secondary frequency control regulates the aggregate power balance in each area and the power flows between areas.
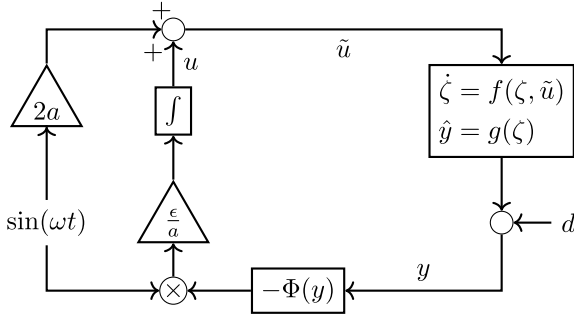
Fig. B.21. Simple extremum seeking to minimize $\Phi(h(u))$.

B.1. Extremum seeking

Arguably, one of the oldest control methods to steer a plant to an extremum of a function rather than tracking a setpoint is extremum seeking (ES) (see Ariyur and Krstic 2003, Tan et al. 2010 for historical accounts). Its popularity rose in the 1950's and 60's as part of adaptive control (Åström & Wittenmark, 2008) and later regained momentum with the rigorous theoretical results derived in Krstic and Wang (2000), Tan et al. (2006), Wittenmark and Urquhart (1995).

The main idea behind ES is to inject a *dither signal* to locally explore the objective function and "learn" its gradient. This dither signal is generally sinusoidal, but other perturbations have been proposed (Teel & Popovic, 2001). Consequently, the objective can be optimized without recourse to any model information about the plant (and objective) and without any computation aside from the addition and multiplication of the dither signal. The following example illustrates this fact.

Example B.1. We consider the same problem as in Example 1.1, i.e., the minimization of $\Phi(y)$ where y is the output of a plant of the form (1). In addition, we assume that the plant is single-input-single-output (and thus all signals are scalar). Instead of the gradient scheme in Fig. 1, we apply the ES setup in Fig. B.21 where a sinusoid perturbs the signal u, yielding $\tilde{u}$, which is then fed to the plant.

As in Example 1.1, we replace the fast plant dynamics by the steady-state map $y = h(u) + d$, and we define the reduced cost function $\tilde{\Phi}(u) := \Phi(h(u) + d)$. Consequently, the *reduced* ES dynamics of the system in Fig. B.21 can be expressed in terms of $\dot{u}$ as

$$\dot{u} = F(u,t) := -\frac{\epsilon}{a} \underbrace{\Phi\big(h\left(u + 2a\sin(\omega t)\right) + d\big)}_{\tilde{\Phi}(u + 2a\sin(\omega t))} \sin(\omega t).$$

The *averaged* dynamics are obtained by integrating $F(u,t)$ from 0 to $T = \frac{2\pi}{\omega}$. Namely, using the Taylor expansion of $\tilde{\Phi}$ around u, the average control signal is

$$\frac{1}{T}\int_0^T F(u,t)dt = -\frac{\epsilon}{aT}\int_0^T \tilde{\Phi}\big(u + 2a\sin(\omega t)\big)\sin(\omega t)dt$$

$$\approx -\frac{\epsilon}{aT}\int_0^T \big(\tilde{\Phi}(u) + 2a\sin(\omega t)\nabla\tilde{\Phi}(u)\big)\sin(\omega t)dt$$

$$= -\frac{\epsilon}{a}\frac{\omega}{2\pi}2a\nabla\tilde{\Phi}(u)\int_0^T \sin(\omega t)^2 dt = -\epsilon\nabla\tilde{\Phi}(u),$$

where we have neglected higher-order terms in ϵ. Thus, the ES scheme approximates the gradient flow (2) from Example 1.1. However, ES merely requires measurements of $\Phi(y)$ and neither an estimate of ∇h nor of $\nabla\Phi$.

In contrast to the design in Example 1.1, ES systems generally evolve on three (rather than two) different timescales: the plant dynamics (which have been ignored for this example), the frequency range of the probing signal, and the slow averaged optimization dynamics. ∎

Classically, averaging theory and singular perturbation analysis (for dynamic plants) are used to render the insights from Example B.1 rigorous (Guay & Zhang, 2003; Krstic & Wang, 2000; Tan et al., 2006). More recently, ES schemes have also been studied with the help of Lie bracket approximations which offer an alternative perspective (Dürr et al., 2017; Dürr, Stanković, Ebenbauer, & Johansson, 2013; Grushkovskaya et al., 2018).

While original work only considered finding extrema (i.e. minima or maxima) of unconstrained problems, constraints have been incorporated by submanifold constraints (Dürr et al., 2014), barrier function (DeHaan & Guay, 2005; Guay et al., 2015), and saddle-point formulations (Dürr, Zeng, & Ebenbauer, 2013). Beyond asymptotic constraint satisfaction, Chen et al. (2021, 2023) introduce projection maps into ES to robustly handle constraints while achieving model-free optimization. ES has also showcased its power in Nash-equilibrium seeking (Frihauf et al., 2012; Stankovic et al., 2012). Further, ES has been studied for stochastic (Coito et al., 2005; Stanković & Stipanović, 2010) and discrete-time setups (Feiling et al., 2018; Frihauf et al., 2013; Stankovic & Stipanovic, 2009), and hybrid extensions have been proposed (Bianchin, Cortés, Poveda & and Dall'Anese, 2022; Poveda et al., 2017; Poveda & Li, 2021; Poveda & Teel, 2017a).

ES has been applied in the automotive sector (Killingsworth et al., 2009), process engineering (Guay et al., 2004), formation flight and obstacle avoidance (Binetti et al., 2003; Montenbruck et al., 2014) and others. Further applications concern problems in renewable energy such as maximum power point tracking in photovoltaic (Ghaffari et al., 2015) or wind energy systems (Ghaffari et al., 2014; Krstic et al., 2014), and Volt-VAR control in power systems (Arnold et al., 2016).

Despite strong theoretical guarantees and being model-free, ES has been confined to relatively low-dimensional, mostly unconstrained, systems. This is due to the fact that plants with multidimensional input require probing signals at different, carefully chosen, frequencies that do not interfere with each other.

B.2. Modifier adaptation

In the context of *real-time optimization* in process engineering, the notion of *measurement-based optimization* (Chachuat et al., 2009; Francois & Bonvin, 2013) has been used to collect several approaches towards mitigating the effects of model bias in repetitive optimization applications. In the following, we showcase modifier adaptation (MA) methods by François et al. (2005), Gao and Engell (2005), Marchetti et al. (2009a, 2009b).

Given a model of a physical system, assume that we can solve an optimal steady-state problem like (21) numerically. When implementing this solution by setting the appropriate inputs of the stable plant to their pre-computed optimal setpoints, the mismatch between the model estimate (used for computing an optimal state) and the actual plant will invariably lead to a system state that is suboptimal, and possibly violating constraints.

If the optimization of the optimal plant state is performed repeatedly, and at each step the solution is implemented on the physical system, MA provides a method to steadily reduce the discrepancy between model-based solution and physical plant by *modifying* the optimal steady-state problem at every iteration by incorporating plant measurements from the previous iteration. MA does not directly "learn" or identify a better model of the plant. Instead, MA corrects the optimization problem by adding adaption terms to the cost and constraint functions. The following simple example demonstrates one possible adaptation mechanism.

Example B.2. Consider the same setup as in Example 1.1. Namely, we wish to minimize the function $\tilde{\Phi}(u) := \Phi(h(u) + d)$ where $y = h(u) + d$ is the steady-state input–output map of a plant with fast-decaying dynamics. However, only an approximate model $\tilde{h}$ of h and an estimate $\tilde{d}$ of d is available.

Therefore, instead of minimizing $\tilde{\Phi}$, we repeatedly solve

$$\underset{u}{\text{minimize}} \quad \tilde{\Phi}(\tilde{h}(u) + \tilde{d}) + \lambda_k u, \tag{B.1}$$

where λ_k is a *modifier* at iteration k that is adapted at every iteration based on the outcome of the previous iteration $u_k^\star$. In particular, λ is updated according to

$$\lambda_{k+1} = \nabla \tilde{\Phi}(u_k^\star) - \nabla \tilde{\Phi}'(u_k^\star),$$

where $\nabla \tilde{\Phi}(u_k^\star)$ needs to be estimated, and $\nabla \tilde{\Phi}'(u_k^\star) := \nabla(\Phi(\tilde{h}(u_k^\star) + \tilde{d}))$ is model-based. The particular structure of $\tilde{\Phi}$, however, lets us write

$$\nabla \tilde{\Phi}(u_k^\star) = \nabla \Phi(h(u_k^\star) + d) \nabla h(u_k^\star), \tag{B.2}$$

where $h(u_k^\star) + d$ is the measured output of the plant. Thus, essentially, only $\nabla h(u_k^\star)$ needs to be estimated. If the scheme converges to some $u^\star$, we can easily verify that $u^\star$ is a critical point of $\tilde{\Phi}(u)$. ∎

Example B.2 is simplified to the point that a comparison with Example 1.1 is easily possible. However, MA methods are easily applied to constrained problems, where modifiers on constraints are introduced analogously (Costello et al., 2014; Faulwasser et al., 2018; Grégory et al., 2014).

Clearly, the tricky part about MA is the estimation of the (true) plant sensitivities $\nabla \tilde{\Phi}(u_k^\star)$. This can be achieved with finite differences (Mansour & Ellis, 2003). Moreover, MA does not reduce the computation burden nor does it aim to reduce the amount of model information required.

B.3. Model predictive control with incomplete optimization and real-time iterations

Historically, model predictive control (MPC) is an approach to control and stabilize a plant that is subject to input and state constraints. This is achieved by numerically solving an optimal control problem with a finite receding horizon at every sampling time (or, every few sampling instants), but implementing only the first (respectively, few) input(s) of the computed optimal policy before solving the next problem with shifted horizon and based on an updated state measurement.

The high computational requirements have long restricted the application of MPC to relatively slow and low-dimensional plants in process engineering. For standard (linear) MPC, this issue has led to *explicit* MPC (Alessio & Bemporad, 2009; Bemporad et al., 2002) which exploits multi-parametric optimization (Tondel et al., 2003) to solve the receding horizon problem ahead of time and implement the controller as a simple lookup table.

More interesting from our perspective are RTIs for *nonlinear MPC* (Bock et al., 2000; Diehl et al., 2002). These methods have emerged as an approximation of *multiple shooting methods* (Diehl et al., 2006) and have been proposed for various applications in process engineering (Diehl et al., 2003), robotics (Diehl et al., 2006), and for airborne kites (Diehl, Bock, & Schlöder, 2005). The main idea of RTIs is to solve the optimal control problem only approximately at every iteration by performing only a single iteration of the underlying optimization algorithm (which is usually a sequential quadratic programming (SQP) scheme; Nocedal and Wright 2006, Quirynen et al. 2018, Zavala and Biegler 2009). The first input of the approximate control policy is implemented and the optimization problem for the next sampling period is warm-started at a shifted version of the previous (approximate) solution. In other words, RTIs are a special case of MPC methods with incomplete optimization (Figura et al., 2020; Graichen & Kugi, 2010; Grüne & Pannek, 2010; Liao-McPherson et al., 2020, 2022). Example B.3 below illustrates this procedure.

The underlying idea of RTIs is that the approximation error committed by performing only a single optimization iteration is offset by savings in computation time. In particular, because the receding horizon problem is solved more often, it changes less between samples.

This feature allows one to prove stability and convergence of RTI schemes (Diehl, Bock, & Schlöder, 2005; Diehl et al., 2007; Diehl, Findeisen, et al., 2005; Zanelli et al., 2020).

However, although they interleave optimization iterations with physical dynamics, RTIs have been developed for stabilization and require an exogenous setpoint as well as a dynamic model of the plant. This is particularly reflected in the assumptions on the state cost function, which are, roughly speaking, required to be quadratic functions centered at the origin (see also Remark B.1 further below on economic MPC).

Example B.3. We consider a discrete-time plant $\zeta^+ = f(\zeta, u)$ for which the origin is a steady state, i.e., $0 = f(0, 0)$. For simplicity, we do not model any constraints, although RTIs can incorporate them naturally.

Consider the receding horizon problem at time l

$$\underset{\{r\}_1^{K-1}, \{s\}_1^K}{\text{minimize}} \quad \sum_{k=1}^{K-1} \begin{bmatrix} s_k \\ r_k \end{bmatrix}^T Q \begin{bmatrix} s_k \\ r_k \end{bmatrix} + s_K^T R s_K$$

$$\text{subject to} \quad s_1 = \overline{\zeta}_l \tag{B.3}$$

$$s_{k+1} = f(s_k, r_k) \quad \forall k \in \{1, \ldots, K-1\}$$

where K denotes the horizon length, Q, R are positive definite stage and terminal cost matrices, and $\overline{\zeta}_l$ denotes the measured plant state at time l. Let $(\hat{r}^l, \hat{s}^l)$ denote the solution of (B.3) for the sampling instant l. Then, the feedback law at l is given by $u[l] = \hat{r}_1^l$. In other words, upon solving (B.3), the first control of the optimal policy is implemented at l. This is the key mechanism behind standard MPC.

RTI schemes approximate the solution of (B.3) by performing only a single iteration of an SQP method. Namely, let $z = (s, r, \lambda)$ and consider the Lagrangian of (B.3) at time l defined as

$$L^l(z) := \sum_{k=1}^{K-1} \begin{bmatrix} s_k \\ r_k \end{bmatrix}^T Q \begin{bmatrix} s_k \\ r_k \end{bmatrix} + s_K^T R s_K + \lambda_1(s_1 - \overline{\zeta}_l)$$

$$+ \sum_{k=1}^{K-1} \lambda_k \left(s_{k+1} - f(s_k, r_k) \right).$$

An SQP iteration then takes the form

$$z^+ = z + \Delta z \tag{B.4}$$

where Δz solves the first-order condition

$$\nabla L^l(z) + \nabla_{zz}^2 L^l(z) \Delta z = 0.$$

where, in practice, the inverse of the Hessian $\nabla_{zz}^2 L^l$ is often approximated.

Crucially, the receding horizon problem (B.3) at the next sample $l+1$ is warm-started with the shifted approximate of the previous sample. This procedure leads, under additional assumptions, to local stability of the scheme. ∎

Example B.3 elucidates several differences with respect to the optimal steady-state control problem Eq. (21) we wish to solve: First, the purpose of RTIs is primarily to drive a plant to the steady state at the origin, not seeking out an operating point with minimal cost (see Remark B.1 below). Further, a full dynamic model f of the plant dynamics is required, and finally, the computational burden of solving the SQP iteration scales not only with the system dimension but also with the prediction horizon. Conversely, compared to Example 1.1, while stabilizing the system, RTI also seeks to minimize a quadratic running stage cost.

Remark B.1. Traditionally, linear and nonlinear MPC have been considered with the goal of stabilizing a plant and tracking a precomputed setpoint or trajectory. The modern variation of *economic MPC* (Ellis et al., 2014; Faulwasser et al., 2018; Rawlings et al., 2012) studies the effects of incorporating an economic objective directly and thus not requiring an exogenous setpoint. This idea is very much in line with the topic of this article.

Economic MPC, however, still requires the solution of an optimal control problem at every iteration. Thus, it is computationally very expensive. Moreover, the optimal solution is not a-priori guaranteed to be a steady state of the plant. This makes the stability analysis of economic MPC more involved and, in particular, still requires a full model of the plant dynamics. ∎

References

Absil, P.-A., & Kurdyka, K. (2006). On the stable equilibrium points of gradient systems. *Systems & Control Letters*, *55*(7), 573–577.

Absil, P., Mahony, R., & Andrews, B. (2005). Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, *16*(2), 531–547.

Absil, P.-A., Mahony, R., & Sepulchre, R. (2008). *Optimization algorithms on matrix manifolds*. Princeton, NJ: Princeton University Press.

Alessio, A., & Bemporad, A. (2009). A survey on explicit model predictive control. In *Lecture notes in control and information sciences*: *vol. 384, Nonlinear model predictive control* (pp. 345–369). Berlin Heidelberg, Germany: Springer.

Allibhoy, A., & Cortés, J. (2023). Control barrier function-based design of gradient flows for constrained nonlinear programming. *IEEE Transactions on Automatic Control*, 1–16.

Ambrosio, L., Gigli, N., & Savaré, G. (2005). Gradient flows: In metric spaces and in the space of probability measures. In *Lectures in mathematics ETH zürich*, Boston: Birkhäuser.

Ariyur, K. B., & Krstic, M. (2003). *Real time optimization by extremum seeking control* (1st ed.). Hoboken, NJ: Wiley Interscience.

Arnold, D. B., Negrete-Pincetic, M., Sankur, M. D., Auslander, D. M., & Callaway, D. S. (2016). Model-free optimal control of VAR resources in distribution systems: An extremum seeking approach. *IEEE Transactions on Power Systems*, *31*(5), 3583–3593.

Arrow, K. J., Hurwicz, L., & Uzawa, H. (1958). *Studies in linear and nonlinear programming*. Stanford, CA: Stanford University Press.

Åström, K. J., & Wittenmark, B. (2008). Adaptive control. In *Dover books on engineering* (2nd ed., Dover ed). Mineola, N.Y: Dover Publications, OCLC: ocn196313137.

Aubin, J.-P., & Cellina, A. (1984). *Grundlehren der mathematischen wissenschaften*, *Differential inclusions: set-valued maps and viability theory*. (264), Berlin Heidelberg, Germany: Springer.

Bastianello, N., & Dall'Anese, E. (2021). Distributed and inexact proximal gradient method for online convex optimization. In *European control conference* (pp. 2432–2437).

Bastianello, N., Simonetto, A., & Carli, R. (2020). Primal and dual prediction-correction methods for time-varying convex optimization. arXiv:2004.11709 [math.OC].

Beck, A. (2017). *First-order methods in optimization*. Philadelphia, PA: SIAM.

Belgioioso, G., Bolognani, S., Pejrani, G., & Dörfler, F. (2023). Tutorial on congestion control in multi-area transmission grids via online feedback equilibrium seeking. In *IEEE conference on decision and control*.

Belgioioso, G., Liao-McPherson, D., de Badyn, M. H., Bolognani, S., Lygeros, J., & Dörfler, F. (2021). Sampled-data online feedback equilibrium seeking: Stability and tracking. In *IEEE conference on decision and control* (pp. 2702–2708).

Belgioioso, G., Liao-McPherson, D., de Badyn, M. H., Bolognani, S., Smith, R. S., Lygeros, J., & Dörfler, F. (2022). Online feedback equilibrium seeking. arXiv:2210.12088 [math.OC].

Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, *38*.

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). Robust optimization. In *Princeton series in applied mathematics*, Princeton, NJ: Princeton University Press.

Bernstein, A., & Dall'Anese, E. (2019). Real-time feedback-based optimization of distribution grids: A unified approach. *IEEE Transactions on Control of Network Systems*, *6*(3), 1197–1209.

Bernstein, A., Dall'Anese, E., & Simonetto, A. (2019). Online primal-dual methods with measurement feedback for time-varying convex optimization. *IEEE Transactions on Signal Processing*, *67*(8), 1978–1991.

Bernstein, A., Reyes-Chamorro, L., Le Boudec, J.-Y., & Paolone, M. (2015). A composable method for real-time control of active distribution networks with explicit power setpoints. Part I: Framework. *Electric Power Systems Research*, *125*, 254–264.

Bertsekas, D. P. (2017). *vol. 1, Dynamic Programming and Optimal Control* (4th ed.). Belmont, MA: Athena Scientific.

Bertsimas, D., & Freund, R. (2004). Data, models, and decisions: The fundamentals of management science. Charlestown, MA: Dynamic Ideas.

Besbes, O., Gur, Y., & Zeevi, A. (2015). Non-stationary stochastic optimization. *Operations Research*, *63*(5), 1227–1244.

Bianchin, G., Cortés, J., Poveda, J. I., & Dall'Anese, E. (2022). Time-varying optimization of LTI systems via projected primal-dual gradient flows. *IEEE Transactions on Control of Network Systems*, *9*(1), 474–486.

Bianchin, G., Poveda, J. I., & Dall'Anese, E. (2022). Online optimization of switched LTI systems using continuous-time and hybrid accelerated gradient flows. *Automatica*, *146*, Article 110579.

Bianchin, G., Vaquero, M., Cortés, J., & Dall'Anese, E. (2023). Online stochastic optimization for unknown linear systems: Data-driven controller synthesis and analysis. *IEEE Transactions on Automatic Control*, http://dx.doi.org/10.1109/TAC.2023.3323581 (in press).

Binetti, P., Ariyur, K. B., Krstic, M., & Bernelli, F. (2003). Formation flight optimization using extremum seeking feedback. *Journal of Guidance, Control, and Dynamics*, *26*(1), 132–142.

Bishop, C. (2009). Pattern recognition and machine learning. In *Information science and statistics*. New York, NY: Springer.

Bloch, A. M., Brockett, R. W., & Ratiu, T. S. (1992). On the geometry of saddle point algorithms. In *IEEE conference on decision and control* (pp. 1482–1487). Tucson, AZ.

Bock, H. G., Diehl, M., Leineweber, D., & Schlöder, J. P. (2000). A direct multiple shooting method for real-time optimization of nonlinear DAE processes. In *Progress in systems and control theory*: *vol. 26, Nonlinear model predictive control* (pp. 245–268). Basel, Switzerland: Springer.

Bolognani, S., Carli, R., Cavraro, G., & Zampieri, S. (2015). Distributed reactive power feedback control for voltage regulation and loss minimization. *IEEE Transactions on Automatic Control*, *60*(4), 966–981.

Bolognani, S., & Dörfler, F. (2015). Fast power system analysis via implicit linearization of the power flow manifold. In *Annual allerton conference on communication, control, and computing* (pp. 402–409). Monticello, IL.

Bolognani, S., & Zampieri, S. (2013). A distributed control strategy for reactive power compensation in smart microgrids. *IEEE Transactions on Automatic Control*, *58*(11), 2818–2833.

Bonnans, J. F. (2019). Convex and stochastic optimization. In *Universitext*. Cham, Switzerland: Springer.

Borkar, V. S. (2008). Stochastic approximation: A dynamical systems viewpoint. *Texts and readings in mathematics*, (48), New Delhi: Hindustan Book Agency, OCLC:231580915.

Brayton, R. K., & Moser, J. K. (1964). A theory of nonlinear networks - I. *Quarterly of Applied Mathematics*, *22*(1).

Bristow, D. A., Tharayil, M., & Alleyne, A. G. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, *26*(3), 96–114.

Brockett, R. W. (1988). Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. *Vol. 1*, In *IEEE conference on decision and control* (pp. 799–803). Austin, TX.

Brunner, F. D., Dürr, H., & Ebenbauer, C. (2012). Feedback design for multi-agent systems: A saddle point approach. In *IEEE conference on decision and control* (pp. 3783–3789). Maui, HI.

Bullo, F. (2022). *Lectures on network systems* (1st ed.). Kindle Direct Publishing.

Chachuat, B., Srinivasan, B., & Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Computers & Chemical Engineering*, *33*(10), 1557–1567.

Chang, C.-Y., Colombino, M., Cortés, J., & Dall'Anese, E. (2019). Saddle-flow dynamics for distributed feedback-based optimization. *IEEE Control Systems Letters*, *3*(4), 948–953.

Chen, Y., Bernstein, A., Devraj, A., & Meyn, S. (2020). Model-free primal-dual methods for network optimization with application to real-time optimal power flow. In *American control conference* (pp. 3140–3147).

Chen, X., Poveda, J. I., & Li, N. (2021). Safe model-free optimal voltage control via continuous-time zeroth-order methods. In *IEEE conference on decision and control*.

Chen, X., Poveda, J. I., & Li, N. (2023). Continuous-time zeroth-order dynamics with projection maps: Model-free feedback optimization with safety guarantees. arXiv:2303.06858 [math.OC].

Chen, X., Tang, Y., & Li, N. (2022). Improve single-point zeroth-order optimization using high-pass and low-pass filters. In *International conference on machine learning* (pp. 3603–3620).

Chen, X., Zhao, C., & Li, N. (2020). Distributed automatic load frequency control with optimality in power systems. *IEEE Transactions on Control of Network Systems*, *8*(1), 307–318.

Cherukuri, A., Gharesifard, B., & Cortes, J. (2017). Saddle-point dynamics: conditions for asymptotic stability of saddle points. *SIAM Journal on Control and Optimization*, *55*(1), 486–511.

Cherukuri, A., Mallada, E., & Cortés, J. (2016). Asymptotic convergence of constrained primal–dual dynamics. *Systems & Control Letters*, *87*, 10–15.

Cherukuri, A., Mallada, E., Low, S., & Cortés, J. (2017). The role of convexity on saddle-point dynamics: Lyapunov function and robustness. *IEEE Transactions on Automatic Control*, *63*(8), 2449–2464.

Clarke, F. (1990). Optimization and nonsmooth analysis. In *Classics in applied mathematics*. Philadelphia, PA: SIAM.

Coito, F., Lemos, J., & Alves, S. (2005). Stochastic extremum seeking in the presence of constraints. In *IFAC world congress*. Prague, Czech Republic.

Colombino, M., Dall'Anese, E., & Bernstein, A. (2020). Online optimization as a feedback controller: Stability and tracking. *IEEE Transactions on Control of Network Systems*, *7*(1), 422–432.

Colombino, M., Simpson-Porco, J. W., & Bernstein, A. (2019). Towards robustness guarantees for feedback-based optimization. In *IEEE conference on decision and control* (pp. 6207–6214).

Conn, A. R., Scheinberg, K., & Vicente, L. N. (2009). *Introduction to derivative-free optimization*. SIAM.

Cortés, J. (2008). Discontinuous dynamical systems. *IEEE Control Systems Magazine*, *28*(3), 36–73.

Cortés, J., & Niederländer, S. K. (2019). Distributed coordination for nonsmooth convex optimization via saddle-point dynamics. *Journal of Nonlinear Science*, *29*(4), 1247–1272.

Costello, S., François, G., Bonvin, D., & Marchetti, A. (2014). Modifier adaptation for constrained closed-loop systems. In *IFAC world congress*. Cape Town, South Africa.

Cothren, L., Bianchin, G., & Dall'Anese, E. (2022a). Data-enabled gradient flow as feedback controller: Regulation of linear dynamical systems to minimizers of unknown functions. In *Learning for dynamics and control conference* (pp. 234–247).

Cothren, L., Bianchin, G., & Dall'Anese, E. (2022b). Online optimization of dynamical systems with deep learning perception. *IEEE Open Journal of Control Systems*, *1*, 306–321.

Dall'Anese, E., & Simonetto, A. (2018). Optimal power flow pursuit. *IEEE Transactions on Smart Grid*, *9*(2), 942–952.

Dall'Anese, E., Simonetto, A., Becker, S., & Madden, L. (2020). Optimization and learning with information streams: Time-varying algorithms and applications. *IEEE Signal Processing Magazine*, *37*(3), 71–83.

Dantzig, G. B. (1998). Linear programming and extensions. In *Princeton landmarks in mathematics and physics*. Princeton, NJ: Princeton Univ. Press, 11. printing, 1. paperback printing.

De Persis, C., & Monshizadeh, N. (2019). A feedback control algorithm to steer networks to a cournot-Nash equilibrium. *IEEE Transactions on Control of Network Systems*, *6*(4), 1486–1497.

DeHaan, D., & Guay, M. (2005). Extremum-seeking control of state-constrained nonlinear systems. *Automatica*, *41*(9), 1567–1574.

Di Pillo, G., & Grippo, L. (1989). Exact penalty functions in constrained optimization. *SIAM Journal on Control and Optimization*, *27*(6), 1333–1360.

Diehl, M., Bock, H., Diedam, H., & Wieber, P.-B. (2006). Fast direct multiple shooting algorithms for optimal robot control. *vol. 340*, In *Fast motions in biomechanics and robotics* (pp. 65–93). Berlin Heidelberg, Germany: Springer.

Diehl, M., Bock, H., & Schlöder, J. (2005). A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, *43*(5), 1714–1736.

Diehl, M., Bock, H. G., Schlöder, J. P., Findeisen, R., Nagy, Z., & Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, *12*(4), 577–585.

Diehl, M., Findeisen, R., & Allgöwer, F. (2007). A stabilizing real-time implementation of nonlinear model predictive control. In *Real-time PDE-constrained optimization* (pp. 25–52). Philadelphia, PA: SIAM.

Diehl, M., Findeisen, R., Allgöwer, F., Bock, H., & Schlöder, J. (2005). Nominal stability of real-time iteration scheme for nonlinear model predictive control. *IEE Proceedings D (Control Theory and Applications)*, *152*(3), 296–308.

Diehl, M., Findeisen, R., Schwarzkopf, S., Uslu, I., Allgöwer, F., Bock, H. G., Gilles, E. D., & Schlöder, J. P. (2003). An efficient algorithm for nonlinear model predictive control of large-scale systems Part II: Experimental evaluation for a distillation column. *At - Automatisierungstechnik*, *51*(1), 22–29.

Ding, Y., Lavaei, J., & Arcak, M. (2021). Escaping spurious local minimum trajectories in online time-varying nonconvex optimization. In *American control conference* (pp. 454–461). New Orleans, LA, USA: IEEE.

Dominguez-Garcia, A. D., Zholbaryssov, M., Amuda, T., & Ajala, O. (2023). An online feedback optimization approach to voltage regulation in inverter-based power distribution networks. In *American control conference* (pp. 1868–1873). San Diego, CA, USA: IEEE.

Dontchev, A. L., Krastanov, M. I., Rockafellar, R. T., & Veliov, V. M. (2013). An Euler–Newton continuation method for tracking solution trajectories of parametric variational inequalities. *SIAM Journal on Control and Optimization*, *51*(3), 1823–1840.

Dontchev, A. L., & Rockafellar, R. T. (2014). Implicit functions and solution mappings. In *Springer series in operations research and financial engineering*. New York, NY: Springer New York.

Doyle, J. C., Francis, B., & Tannenbaum, A. (2009). *Feedback control theory*. New York, NY: Dover.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., & Wibisono, A. (2015). Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, *61*(5), 2788–2806.

Dürr, H.-B., Krstić, M., Scheinker, A., & Ebenbauer, C. (2017). Extremum seeking for dynamic maps using Lie brackets and singular perturbations. *Automatica*, *83*, 91–99.

Dürr, H.-B., Stanković, M. S., Ebenbauer, C., & Johansson, K. H. (2013). Lie bracket approximation of extremum seeking systems. *Automatica*, *49*(6), 1538–1552.

Dürr, H.-B., Stanković, M. S., Johansson, K. H., & Ebenbauer, C. (2014). Extremum seeking on submanifolds in the euclidian space. *Automatica*, *50*(10), 2591–2596.

Dürr, H.-B., Zeng, C., & Ebenbauer, C. (2013). Saddle point seeking for convex optimization problems. In *IFAC symposium on nonlinear control systems* (pp. 540–545). Tolouse, France.

Ellis, M., Durand, H., & Christofides, P. D. (2014). A tutorial review of economic model predictive control methods. *Journal of Process Control*, *24*(8), 1156–1178.

Farivar, M., Chen, L., & Low, S. (2013). Equilibrium and dynamics of local voltage control in distribution systems. In *IEEE conference on decision and control* (pp. 4329–4334).

Fattahi, S., Josz, C., Mohammadi, R., Lavaei, J., & Sojoudi, S. (2020). Absence of spurious local trajectories in time-varying optimization: A control-theoretic perspective. In *IEEE conference on control technology and applications* (pp. 140–147). Montreal, QC, Canada: IEEE.

Faulwasser, T., Grüne, L., & Müller, M. A. (2018). Economic nonlinear model predictive control. *Vol. 5*, In *Foundations and trends in systems and control*. (1).

Fazlyab, M., Paternain, S., Preciado, V. M., & Ribeiro, A. (2016). Interior point method for dynamic constrained optimization in continuous time. In *American control conference* (pp. 5612–5618). Boston, MA.

Fazlyab, M., Paternain, S., Preciado, V. M., & Ribeiro, A. (2018). Prediction-correction interior-point method for time-varying convex optimization. *IEEE Transactions on Automatic Control*, *63*(7), 1973–1986.

Fazlyab, M., Ribeiro, A., Morari, M., & Preciado, V. M. (2018). Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems. *SIAM Journal on Optimization*, *28*(3), 2654–2689.

Feijer, D., & Paganini, F. (2010). Stability of primal–dual gradient dynamics and applications to network optimization. *Automatica*, *46*(12), 1974–1981.

Feiling, J., Zeller, A., & Ebenbauer, C. (2018). Derivative-free optimization algorithms based on non-commutative maps. *IEEE Control Systems Letters*, *2*(4), 743–748.

Figura, M., Su, L., Gupta, V., & Inoue, M. (2020). Instant distributed model predictive control for constrained linear systems. In *American control conference* (pp. 4582–4587). Denver, CO, USA.

Flaxman, A. D., Kalai, A. T., & McMahan, H. B. (2005). Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Annual ACM-SIAM symposium on discrete algorithms* (pp. 385–394).

Francois, G., & Bonvin, D. (2013). Measurement-based real-time optimization of chemical processes. In *Advances in chemical engineering: vol. 43*.

François, G., Srinivasan, B., & Bonvin, D. (2005). Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. *Journal of Process Control*, *15*(6), 701–712.

Frank, S., & Rebennack, S. (2016). An introduction to optimal power flow: Theory, formulation, and examples. *IIE Transactions*, *48*(12), 1172–1197.

Frank, S., Steponavice, I., & Rebennack, S. (2012a). Optimal power flow: A bibliographic survey I: Formulations and deterministic methods. *Energy Systems*, *3*(3), 221–258.

Frank, S., Steponavice, I., & Rebennack, S. (2012b). Optimal power flow: A bibliographic survey II: Non-deterministic and hybrid methods. *Energy Systems*, *3*(3), 259–289.

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2010). *Feedback control of dynamic systems* (6th ed.). Upper Saddle River, NJ: Pearson.

Frasconi, P., Landwehr, N., Manco, G., & Vreeken, J. (Eds.), (2016). *Lecture notes in computer science: vol. 9851, Machine learning and knowledge discovery in databases: European conference, ECML PKDD 2016, Riva Del Garda, Italy, September 19-23, 2016, Proceedings, Part I*. Cham: Springer International Publishing.

Frazier, P. I. (2018). A tutorial on Bayesian optimization. arXiv:1807.02811 [stat.ML].

Frihauf, P., Krstic, M., & Basar, T. (2012). Nash equilibrium seeking in noncooperative games. *IEEE Transactions on Automatic Control*, *57*(5), 1192–1207.

Frihauf, P., Krstic, M., & Başar, T. (2013). Finite-horizon LQ control for unknown discrete-time linear systems via extremum seeking. *European Journal of Control*, *19*(5), 399–407.

Gan, L., & Low, S. H. (2016). An online gradient algorithm for optimal power flow on radial networks. *IEEE Journal on Selected Areas in Communications*, *34*(3), 625–638.

Gao, W., & Engell, S. (2005). Iterative set-point optimization of batch chromatography. *Computers & Chemical Engineering*, *29*(6), 1401–1409.

Ghaffari, A., Krstic, M., & Seshagiri, S. (2014). Power optimization and control in wind energy conversion systems using extremum seeking. *IEEE Transactions on Control Systems Technology*, *22*(5), 1684–1695.

Ghaffari, A., Seshagiri, S., & Krstić, M. (2015). Multivariable maximum power point tracking for photovoltaic micro-converters using extremum seeking. *Control Engineering Practice*, *35*, 83–91.

Goebel, R. (2017). Stability and robustness for saddle-point dynamics through monotone mappings. *Systems and Control Letters*, *108*, 16–22.

Graichen, K., & Kugi, A. (2010). Stability and incremental improvement of suboptimal MPC without terminal constraints. *IEEE Transactions on Automatic Control*, *55*(11), 2576–2580.

Grégory, G., Ois, F., & Bonvin, D. (2014). Use of transient measurements for the optimization of steady-state performance via modifier adaptation. *Industrial Engineering Chemistry Research*, *53*, 5148–5159.

Grüne, L., & Pannek, J. (2010). Analysis of unconstrained NMPC schemes with incomplete optimization. *IFAC Proceedings Volumes*, *43*(14), 238–243.

Grushkovskaya, V., Zuyev, A., & Ebenbauer, C. (2018). On a class of generating vector fields for the extremum seeking problem: Lie bracket approximation and stability properties. *Automatica*, *94*, 151–160.

Guay, M., Dochain, D., & Perrier, M. (2004). Adaptive extremum seeking control of continuous stirred tank bioreactors with unknown growth kinetics. *Automatica*, *40*(5), 881–888.

Guay, M., Moshksar, E., & Dochain, D. (2015). A constrained extremum-seeking control approach. *International Journal of Robust and Nonlinear Control*, *25*(16), 3132–3153.

Guay, M., & Zhang, T. (2003). Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties. *Automatica*, *39*(7), 1283–1293.

Häberle, V., Hauswirth, A., Ortmann, L., Bolognani, S., & Dörfler, F. (2021). Non-convex feedback optimization with input and output constraints. *IEEE Control Systems Letters*, *5*(1), 343–348.

Hall, E. C., & Willett, R. M. (2015). Online convex optimization in dynamic environments. *IEEE Journal of Selected Topics in Signal Processing*, *9*(4), 647–662.

Hauswirth, A. (2020). *Optimization algorithms as feedback controllers for power system operations* (Ph.D. thesis), Zürich, Switzerland: ETH Zürich.

Hauswirth, A., Bolognani, S., & Dörfler, F. (2020). Projected dynamical systems on irregular, non-euclidean domains for nonlinear optimization. *SIAM Journal on Control and Optimization*, *59*(1), 635–668.

Hauswirth, A., Bolognani, S., Hug, G., & Dörfler, F. (2016). Projected gradient descent on Riemannian manifolds with applications to online power system optimization. In *Annual allerton conference on communication, control, and computing* (pp. 225–232). Monticello, IL.

Hauswirth, A., Bolognani, S., Hug, G., & Dörfler, F. (2020). Timescale separation in autonomous optimization. *IEEE Transactions on Automatic Control*, *66*(2), 611–624.

Hauswirth, A., Dörfler, F., & Teel, A. R. (2020a). Anti-windup approximations of oblique projected dynamical systems for feedback-based optimization. arXiv:2003.00478 [math.OC].

Hauswirth, A., Dörfler, F., & Teel, A. R. (2020b). On the differentiability of projected trajectories and the robust convergence of non-convex anti-windup gradient flows. *IEEE Control Systems Letters*, *4*(3), 620–625.

Hauswirth, A., Dörfler, F., & Teel, A. R. (2020c). On the robust implementation of projected dynamical systems with anti-windup controllers. In *American control conference* (pp. 1286–1291). Denver, CO, USA.

Hauswirth, A., Ortmann, L., Bolognani, S., & Dörfler, F. (2020). Limit behavior and the role of augmentation in projected saddle flows for convex optimization. In *IFAC world congress*. Berlin, Germany.

Hauswirth, A., Subotić, I., Bolognani, S., Hug, G., & Dörfler, F. (2018). Time-varying projected dynamical systems with applications to feedback optimization of power systems. In *IEEE conference on decision and control* (pp. 3258–3263). Miami Beach, FL.

Hauswirth, A., Zanardi, A., Bolognani, S., Dörfler, F., & Hug, G. (2017). Online optimization in closed loop on the power flow manifold. In *IEEE powerTech conference*. Manchester, UK.

Hazan, E. (2022). *Introduction to online convex optimization*. Princeton, NJ: MIT Press.

He, Z., Bolognani, S., He, J., Dörfler, F., & Guan, X. (2023). Model-free nonlinear feedback optimization. *IEEE Transactions on Automatic Control*, http://dx.doi.org/10.1109/TAC.2023.3341752 (in press).

Helmke, U., & Moore, J. B. (1996). Optimization and dynamical systems. In *Communications and control engineering* (second ed.). London, UK: Springer.

Hillier, F. S., & Lieberman, G. J. (2001). *Introduction to operations research* (7th ed.). New York, NY: McGraw-Hill.

Hjalmarsson, H. (2002). Iterative feedback tuning—an overview. *International Journal of Adaptive Control and Signal Processing*, *16*, 373–395.

Holding, T., & Lestas, I. (2014). On the convergence to saddle points of concave-convex functions, the gradient method and emergence of oscillations. In *IEEE conference on decision and control* (pp. 1143–1148). Los Angeles, CA, USA.

Huneault, M., & Galiana, F. D. (1991). A survey of the optimal power flow literature. *IEEE Transactions on Power Systems*, *6*(2), 762–770.

Jadbabaie, A., Rakhlin, A., Shahrampour, S., & Sridharan, K. (2015). Online optimization : Competing with dynamic comparators. In *International conference on artificial intelligence and statistics* (pp. 398–406). San Diego, CA, USA.

Jokic, A., Lazar, M., & van den P. P. J. Bosch (2009). On constrained steady-state regulation : Dynamic KKT controllers. *IEEE Transactions on Automatic Control*, *54*(9), 2250–2254.

Jongen, H. T., Jonker, P., & Twilt, F. (2001). Nonlinear optimization in finite dimensions. *vol. 47*, In *Nonconvex optimization and its applications*. Boston, MA: Springer US.

Kelly, F. P., Maulloo, A. K., & Tan, D. K. H. (1998). Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, *49*(3), 237–252.

Khalil, H. K. (2002). *Nonlinear systems* (third ed.). Upper Saddle River, NJ: Prentice Hall.

Killingsworth, N., Aceves, S., Flowers, D., Espinosa-Loza, F., & Krstic, M. (2009). HCCI engine combustion-timing control: Optimizing gains and fuel consumption via extremum seeking. *IEEE Transactions on Control Systems Technology*, *17*(6), 1350–1361.

Kokotovic, P., Khalil, H., & O'Reilly, J. (1999). Singular perturbation methods in control: Analysis and design. In *Classics in applied mathematics*. (25), Philadelphia, PA: SIAM.

Kose, T. (1956). Solutions of saddle value problems by differential equations. *Econometrica*, *24*(1), 59–70.

Koshal, J., Nedić, A., & Shanbhag, U. (2011). Multiuser optimization: Distributed algorithms and error analysis. *SIAM Journal on Optimization*, *21*(3), 1046–1081.

Krishnamoorthy, D., & Doyle, F. J. (2022). Safe Bayesian optimization using interior-point methods—Applied to personalized insulin dose guidance. *IEEE Control Systems Letters*, *6*, 2834–2839.

Krishnamoorthy, D., & Doyle III, F. J. (2023). Model-free real-time optimization of process systems using safe Bayesian optimization. *AIChE Journal*, *69*(4), Article e17993.

Kroposki, B., Bernstein, A., King, J., Vaidhynathan, D., Zhou, X., Chang, C.-Y., & Dall'Anese, E. (2020). Autonomous energy grids: Controlling the future grid with large amounts of distributed energy resources. *IEEE Power and Energy Magazine*, *18*(6), 37–46.

Krstic, M., Ghaffari, A., & Seshagiri, S. (2014). Extremum seeking for wind and solar energy applications. In *World congress on intelligent control and automation* (pp. 6184–6193). Shenyang, China.

Krstic, M., & Wang, H.-H. (2000). Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, *36*, 595–601.

Kungurtsev, V., & Jäschke, J. (2017). A predictor-corrector path-following algorithm for dual-degenerate parametric optimization problems. *SIAM Journal on Optimization*, *27*(1), 538–564.

Lawrence, L. S. P., Nelson, Z. E., Mallada, E., & Simpson-Porco, J. W. (2018). Optimal steady-state control for linear time-invariant systems. In *IEEE conference on decision and control* (pp. 3251–3257). Miami Beach, FL, USA.

Lawrence, L. S. P., Simpson-Porco, J. W., & Mallada, E. (2021). Linear-convex optimal steady-state control. *IEEE Transactions on Automatic Control*, *66*(11), 5377–5384.

Lee, J. M. (1997). Riemannian manifolds - an introduction to curvature. In *Graduate texts in mathematics* (first ed.). (176), New York, NY: Springer.

Lesage-Landry, A., Shames, I., & Taylor, J. A. (2020). Predictive online convex optimization. *Automatica*, *113*, Article 108771.

Lessard, L., Recht, B., & Packard, A. (2016). Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, *26*(1), 57–95.

Lewis, F. L., & Liu, D. (Eds.), (2012). *Reinforcement learning and approximate dynamic programming for feedback control*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Li, N., Zhao, C., & Chen, L. (2016). Connecting automatic generation control and economic dispatch from an optimization view. *IEEE Transactions on Control of Network Systems*, *3*(3), 254–264.

Liao-McPherson, D., Nicotra, M. M., & Kolmanovsky, I. (2020). Time-distributed optimization for real-time model predictive control: Stability, robustness, and constraint satisfaction. *Automatica*, *117*, Article 108973.

Liao-McPherson, D., Skibik, T., Leung, J., Kolmanovsky, I., & Nicotra, M. M. (2022). An analysis of closed-loop stability for linear model predictive control based on time-distributed optimization. *IEEE Transactions on Automatic Control*, *67*(5), 2618–2625.

Lin, Y., Goel, G., & Wierman, A. (2020). Online optimization with predictions and non-convex losses. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, *4*(1), 1–32.

Liu, S., Chen, P.-Y., Kailkhura, B., Zhang, G., Hero III, A. O., & Varshney, P. K. (2020). A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, *37*(5), 43–54.

Low, S. H. (2014). Convex relaxation of optimal power flow; Part I: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, *1*(1), 15–27.

Low, S. H. (2017). Analytical methods for network congestion control. In *Synthesis lectures on communication networks*, (18), Williston, VT: Morgan & Claypool.

Low, S. H., & Lapsley, D. E. (1999). Optimization flow control. I. Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, *7*(6), 861–874.

Low, S. H., Paganini, F., & Doyle, J. C. (2002). Internet congestion control. *IEEE Control Systems Magazine*, *22*(1), 28–43.

Luenberger, D. G. (1969). Optimization by vector space methods. In *Series in decision and control*, New York: J. Wiley.

Madden, L., Becker, S., & Dall'Anese, E. (2021). Bounds for the tracking error of first-order online optimization methods. *Journal of Optimization Theory and Applications*, *189*(2), 437–457.

Mansour, M., & Ellis, J. (2003). Comparison of methods for estimating real process derivatives in on-line optimization. *Applied Mathematical Modelling*, *27*(4), 275–291.

Marchetti, A., Chachuat, B., & Bonvin, D. (2009a). Modifier-adaptation methodology for real-time optimization. *Industrial and Engineering Chemistry Research*, *48*(13), 6022–6033.

Marchetti, A., Chachuat, B., & Bonvin, D. (2009b). Real-time optimization with estimation of experimental gradients. In *IFAC symposium on advanced control of chemical processes* (pp. 524–529). Tolouse, France.

Markovsky, I., & Dörfler, F. (2021). Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, *52*, 42–64.

Markovsky, I., Huang, L., & Dörfler, F. (2023). Data-driven control based on the behavioral approach: From theory to applications in power systems. *IEEE Control Systems*, *43*(5), 28–68.

Mazzi, N., Zhang, B., & Kirschen, D. S. (2018). An online optimization algorithm for alleviating contingencies in transmission networks. *IEEE Transactions on Power Systems*, *33*(5), 5572–5582.

Menta, S., Hauswirth, A., Bolognani, S., Hug, G., & Dörfler, F. (2018). Stability of dynamic feedback optimization with applications to power systems. In *Annual allerton conference on communication, control, and computing* (pp. 136–143). Monticello, IL.

Mokhtari, A., Shahrampour, S., Jadbabaie, A., & Ribeiro, A. (2016). Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *IEEE conference on decision and control* (pp. 7195–7201). Las Vegas, NV.

Molzahn, D. K., Dörfler, F., Sandberg, H., Low, S. H., Chakrabarti, S., Baldick, R., & Lavaei, J. (2017). A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Smart Grid*, *8*(6), 2941–2962.

Molzahn, D. K., & Hiskens, I. A. (2019). A survey of relaxations and approximations of the power flow equations. *FNT in Electric Energy Systems*, *4*(1–2), 1–221.

Montenbruck, J. M., Dürr, H.-B., Ebenbauer, C., & Allgöwer, F. (2014). Extremum seeking and obstacle avoidance on the special orthogonal group. In *IFAC world congress*. Cape Town, South Africa.

Muehlebach, M., & Jordan, M. I. (2022). On constraints in first-order optimization: A view from non-smooth dynamical systems. *Journal of Machine Learning Research*, *23*(256), 1–47.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, *7*(4), 308–313.

Nelson, Z. E., & Mallada, E. (2018). An integral quadratic constraint framework for real-time steady-state optimization of linear time-invariant systems. In *American control conference* (pp. 597–603). Milwaukee, WI.

Nesterov, Y., & Nemirovskii, A. S. (1994). Interior-point polynomial algorithms in convex programming. *Vol. 13*, In *SIAM studies in applied mathematics*. Philadelphia: Society for Industrial and Applied Mathematics.

Nesterov, Y., & Spokoiny, V. (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, *17*, 527–566.

Neumann-Brosig, M., Marco, A., Schwarzmann, D., & Trimpe, S. (2020). Data-efficient autotuning with Bayesian optimization: An industrial control study. *IEEE Transactions on Control Systems Technology*, *28*(3), 730–740.

Nocedal, J., & Wright, S. J. (2006). Numerical optimization, In *Series in operations research*, (second ed.). New York, NY: Springer.

Nonhoff, M., & Müller, M. A. (2020). Online gradient descent for linear dynamical systems. In *IFAC world congress*. Berlin, Germany.

Nonhoff, M., & Müller, M. A. (2021). Data-driven online convex optimization for control of dynamical systems. In *IEEE conference on decision and control* (pp. 3640–3645).

Nonhoff, M., & Müller, M. A. (2022). Online convex optimization for data-driven control of dynamical systems. *IEEE Open Journal of Control Systems*, *1*, 180–193.

Notarnicola, I., Simonetto, A., Farina, F., & Notarstefano, G. (2023). Distributed personalized gradient tracking with convex parametric models. *IEEE Transactions on Automatic Control*, *68*(1), 588–595.

Ortmann, L., Hauswirth, A., Caduff, I., Dörfler, F., & Bolognani, S. (2020). Experimental validation of feedback optimization in power distribution grids. *Electric Power Systems Research*, *189*, Article 106782.

Ortmann, L., Prostejovsky, A., Heussen, K., & Bolognani, S. (2020). Fully distributed peer-to-peer optimal voltage control with minimal model requirements. *Electric Power Systems Research*, *189*, Article 106717.

Ortmann, L., Rubin, C., Scozzafava, A., Lehmann, J., Bolognani, S., & Dörfler, F. (2023). Deployment of an online feedback optimization controller for reactive power flow optimization in a distribution grid. In *IEEE PES ISGT Europe*.

Paganini, Z. W., Doyle, J., & Low, S. (2005). Congestion control for high performance, stability, and fairness in general networks. *IEEE/ACM Transactions on Networking*, *13*(1), 43–56.

Palis, J. J., & De Melo, W. (1982). *Geometric theory of dynamical systems: an introduction*. New York, NY: Springer.

Picallo, M., Bolognani, S., & Dörfler, F. (2023). Sensitivity-conditioning: Beyond singular perturbation for control design on multiple time scales. *IEEE Transactions on Automatic Control*, *68*(4), 2309–2324.

Picallo, M., Ortmann, L., Bolognani, S., & Dörfler, F. (2022). Adaptive real-time grid operation via online feedback optimization with sensitivity estimation. *Electric Power Systems Research*, *212*, Article 108405.

Popkov, A. Y. (2005). Gradient methods for nonstationary unconstrained optimization problems. *Automation and Remote Control*, *66*(6), 883–891.

Poveda, J., Benosman, M., & Teel, A. (2017). Distributed extremum seeking in multi-agent systems with arbitrary switching graphs. In *IFAC world congress*. Toulouse, France.

Poveda, J. I., & Li, N. (2019). Inducing uniform asymptotic stability in non-autonomous accelerated optimization dynamics via hybrid regularization. In *IEEE conference on decision and control* (pp. 3000–3005). Nice, France.

Poveda, J. I., & Li, N. (2021). Robust hybrid zero-order optimization algorithms with acceleration via averaging in time. *Automatica*, *123*, Article 109361.

Poveda, J. I., & Teel, A. R. (2017a). A framework for a class of hybrid extremum seeking controllers with dynamic inclusions. *Automatica*, *76*, 113–126.

Poveda, J. I., & Teel, A. R. (2017b). A robust event-triggered approach for fast sampled-data extremization and learning. *IEEE Transactions on Automatic Control*, *62*(10), 4949–4964.

Quirynen, R., Gros, S., & Diehl, M. (2018). Inexact Newton-type optimization with iterated sensitivities. *SIAM Journal on Optimization*, *28*(1), 74–95.

Rahili, S., & Ren, W. (2017). Distributed continuous-time convex optimization with time-varying cost functions. *IEEE Transactions on Automatic Control*, *62*(4), 1590–1605.

Rahili, S., Ren, W., & Ghapani, S. (2015). Distributed convex optimization of time-varying cost functions with swarm tracking behavior for continuous-time dynamics. In *IEEE conference on decision and control*.

Rawlings, J. B., Angeli, D., & Bates, C. N. (2012). Fundamentals of economic model predictive control. In *IEEE conference on decision and control* (pp. 3851–3861). Maui, HI.

Reyes-Chamorro, L., Bernstein, A., Le Boudec, J.-Y., & Paolone, M. (2015). A composable method for real-time control of active distribution networks with explicit power setpoints. Part II: Implementation and validation. *Electric Power Systems Research*, *125*, 265–280.

Rockafellar, R. T., & Wets, R. J.-B. (2009). Variational analysis, In *Grundlehren der mathematischen wissenschaften*, (third ed.). (317), Berlin Heidelberg, Germany: Springer.

Rodríguez-Gonzalez, E., Olives-Camps, J., García-Lopez, F., Del Nozal, A. R., Mauricio, J., & Maza-Ortega, J. (2022). Experimental validation of a real-time distributed model-less control for DC microgrids. In *International conference on smart energy systems and technologies* (pp. 1–6).

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, *104*(1), 148–175.

Shakkottai, S., & Srikant, R. (2007). Network optimization and control. *FNT in Networking*, *2*(3), 271–379.

Shamir, O. (2017). An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, *18*(1), 1703–1713.

Shapiro, A. (1988). Sensitivity analysis of nonlinear programs and differentiability properties of metric projections. *SIAM Journal on Control and Optimization*, *26*(3), 628–645.

Shi, G., Chung, S.-J., Lin, Y., Yue, Y., & Wierman, A. (2020). Online optimization with memory and competitive control. In *Conference on neural information processing systems* (p. 12). Vancouver, Canada.

Simon, D. (2013). *Evolutionary optimization algorithms*. John Wiley & Sons.

Simonetto, A. (2017). Time-varying convex optimization via time-varying averaged operators. arXiv:1704.07338 [math].

Simonetto, A., & Dall'Anese, E. (2017). Prediction-correction algorithms for time-varying constrained optimization. *IEEE Transactions on Signal Processing*, *65*(20), 5481–5494.

Simonetto, A., Dall'Anese, E., Monteil, J., & Bernstein, A. (2021). Personalized optimization with user's feedback. *Automatica*, *131*, Article 109767.

Simonetto, A., Dall'Anese, E., Paternain, S., Leus, G., & Giannakis, G. B. (2020). Time-varying convex optimization: Time-structured algorithms and applications. *Proceedings of the IEEE*, *108*(11), 2032–2048.

Simonetto, A., Koppel, A., Mokhtari, A., Leus, G., & Ribeiro, A. (2017). Decentralized prediction-correction methods for networked time-varying convex optimization. *IEEE Transactions on Automatic Control*, *62*(11), 5724–5738.

Simonetto, A., & Leus, G. (2014). Distributed asynchronous time-varying constrained optimization. In *Asilomar conference on signals, systems and computers* (pp. 2142–2146). Pacific Grove, CA.

Simpson-Porco, J. W. (2016). Input/output analysis of primal-dual gradient algorithms. In *Annual allerton conference on communication, control, and computing* (pp. 219–224). Monticello, IL.

Simpson-Porco, J. W. (2021a). Analysis and synthesis of low-gain integral controllers for nonlinear systems. *IEEE Transactions on Automatic Control*, *66*(9), 4148–4159.

Simpson-Porco, J. W. (2021b). On stability of distributed-averaging proportional-integral frequency control in power systems. *IEEE Control Systems Letters*, *5*(2), 677–682.

Simpson-Porco, J. W. (2022). Low-gain stability of projected integral control for input-constrained discrete-time nonlinear systems. *IEEE Control Systems Letters*, *6*, 788–793.

Smale, S. (1972). On the mathematical foundations of electrical circuit theory. *Journal of Differential Geometry*, *7*(1–2), 193–210.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, *37*(3), 332–341.

Stankovic, M. S., Johansson, K. H., & Stipanovic, D. M. (2012). Distributed seeking of Nash equilibria with applications to mobile sensor networks. *IEEE Transactions on Automatic Control*, *57*(4), 904–919.

Stankovic, M. S., & Stipanovic, D. M. (2009). Discrete time extremum seeking by autonomous vehicles in a stochastic environment. In *IEEE conference on decision and control (CDC) and Chinese control conference* (pp. 4541–4546). Shanghai, China.

Stanković, M. S., & Stipanović, D. M. (2010). Extremum seeking under stochastic noise and applications to mobile sensors. *Automatica*, *46*(8), 1243–1251.

Stegink, T. W., Damme, T. V., & Persis, C. D. (2018). Convergence of projected primal-dual dynamics with applications in data centers. In *IFAC workshop on distributed estimation and control in networked systems* (pp. 88–93). Groningen, The Netherlands.

Stegink, T., De Persis, C., & van der Schaft, A. (2017). A unifying energy-based approach to stability of power grids with market dynamics. *IEEE Transactions on Automatic Control*, *62*(6), 2612–2622.

Subotić, I., Hauswirth, A., & Dörfler, F. (2022). Quantitative sensitivity bounds for nonlinear programming and time-varying optimization. *IEEE Transactions on Automatic Control*, *67*(6), 2829–2842.

Sui, Y., Gotovos, A., Burdick, J., & Krause, A. (2015). Safe exploration for optimization with Gaussian processes. In *International conference on machine learning* (pp. 997–1005).

Suwartadi, E., Kungurtsev, V., & Jäschke, J. (2017). Sensitivity-based economic NMPC with a path-following approach. *Processes*, *5*(4), 8.

Tan, Y., Moase, W. H., Manzie, C., Nesic, D., & Mareels, I. M. Y. (2010). Extremum seeking from 1922 to 2010. In *Chinese control conference* (pp. 14–26). Beijing, China.

Tan, Y., Nešić, D., & Mareels, I. (2006). On non-local stability properties of extremum seeking control. *Automatica, 42*(6), 889–903.

Tang, Y., Dall'Anese, E., Bernstein, A., & Low, S. H. (2018). A feedback-based regularized primal-dual gradient method for time-varying nonconvex optimization. In *IEEE conference on decision and control* (pp. 3244–3250). Miami Beach, FL.

Tang, Y., Dall'Anese, E., Bernstein, A., & Low, S. (2022). Running primal-dual gradient method for time-varying nonconvex problems. *SIAM Journal on Control and Optimization, 60*(4), 1970–1990.

Tang, Y., Dvijotham, K., & Low, S. (2017). Real-time optimal power flow. *IEEE Transactions on Smart Grid, 8*(6), 2963–2973.

Tang, Y., Ren, Z., & Li, N. (2023). Zeroth-order feedback optimization for cooperative multi-agent systems. *Automatica, 148*, Article 110741.

Tang, A., Wang, J., Low, S. H., & Chiang, M. (2007). Equilibrium of heterogeneous congestion control: Existence and uniqueness. *IEEE/ACM Transactions on Networking, 15*(4), 824–837.

Teel, A., & Popovic, D. (2001). Solving smooth and nonsmooth multivariable extremum seeking problems by the methods of nonlinear programming. In *American control conference* (pp. 2394–2399). Arlington, VA.

Tondel, P., Johansen, T. A., & Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica, 39*(3), 489–497.

Torrisi, G., Grammatico, S., Smith, R. S., & Morari, M. (2018). A projected gradient and constraint linearization method for nonlinear model predictive control. *SIAM Journal on Control and Optimization, 56*(3), 1968–1999.

Trip, S., Cucuzzella, M., De Persis, C., van der Schaft, A., & Ferrara, A. (2019). Passivity-based design of sliding modes for optimal load frequency control. *IEEE Transactions on Control Systems Technology, 27*(5), 1893–1906.

Van Cutsem, T., & Vournas, C. (1998). *Voltage stability of electric power systems*. Dordrecht, The Netherlands: Springer.

van der Schaft, A. J. (2011). On the relation between port-Hamiltonian and gradient systems. In *IFAC world congress*. Milano, Italy.

Venets, V. (1985). Continuous algorithms for solution of convex optimization problems and finding saddle points of contex-coneave functions with the use of projection operations. *Optimization, 16*(4), 519–533.

Vinnicombe, G. (2002). On the stability of networks operating TCP-like congestion control. In *IFAC world congress*. Barcelona, Spain.

Wang, J., & Elia, N. (2011). A control perspective for centralized and distributed convex optimization. In *IEEE conference on decision and control and European control conference* (pp. 3800–3805). Orlando, FL.

Wei, D. X., Jin, C., Low, S. H., & Hegde, S. (2006). FAST TCP: Motivation, architecture, algorithms, performance. *IEEE/ACM Transactions on Networking, 14*(6), 1246–1259.

Wei, E., Ozdaglar, A., & Jadbabaie, A. (2013). A distributed Newton method for network utility maximization—Part II: Convergence. *IEEE Transactions on Automatic Control, 58*(9), 2176–2188.

Wen, J. T., & Arcak, M. (2004). A unifying passivity framework for network flow control. *IEEE Transactions on Automatic Control, 49*(2), 162–174.

Willems, J. C., Rapisarda, P., Markovsky, I., & De Moor, B. L. (2005). A note on persistency of excitation. *Systems & Control Letters, 54*(4), 325–329.

Wittenmark, B., & Urquhart, A. (1995). Adaptive extremal control. *Vol. 2, In IEEE conference on decision and control* (pp. 1639–1644). New Orleans, LA, USA: IEEE.

Xia, Y. S., & Wang, J. (2000). On the stability of globally projected dynamical systems. *Journal of Optimization Theory and Applications, 106*, 129–150.

Xu, W., Jones, C. N., Svetozarevic, B., Laughman, C. R., & Chakrabarty, A. (2023). Violation-aware contextual Bayesian optimization for controller performance optimization with unmodeled constraints. arXiv:2301.12099 [cs.LG].

Zanelli, A., Dinh, Q. T., & Diehl, M. (2020). A Lyapunov function for the combined system-optimizer dynamics in nonlinear model predictive control. arXiv:2004.08578 [math].

Zargham, M., Ribeiroy, A., & Jadbabaie, A. (2013). Accelerated dual descent for constrained convex network flow optimization. In *IEEE conference on decision and control* (pp. 1037–1042). Florence, Italy.

Zavala, V., & Anitescu, M. (2010). Real-time nonlinear optimization as a generalized equation. *SIAM Journal on Control and Optimization, 48*(8), 5444–5467.

Zavala, V. M., & Biegler, L. T. (2009). Nonlinear programming strategies for state estimation and model predictive control. *vol. 384, In Nonlinear model predictive control* (pp. 419–432). Berlin Heidelberg, Germany: Springer.

Zehnder, E. (2010). Lectures on dynamical systems: Hamiltonian vector fields and symplectic capacities. In *EMS textbooks in mathematics*, Zürich: European Math. Soc.

Zhang, Y., Dall'Anese, E., & Hong, M. (2021). Online proximal-ADMM for time-varying constrained convex optimization. *IEEE Transactions on Signal and Information Processing over Networks, 7*, 144–155.

Zhang, Y., Zhou, Y., Ji, K., & Zavlanos, M. M. (2022). A new one-point residual-feedback oracle for black-box learning and control. *Automatica, 136*, Article 110006.

Zhao, C., Mallada, E., Low, S., & Bialek, J. (2016). A unified framework for frequency control and congestion management. In *Power systems computation conference* (pp. 1–7). Genoa, Italy: IEEE.

Zheng, T., Simpson-Porco, J., & Mallada, E. (2020). Implicit trajectory planning for feedback linearizable systems: A time-varying optimization approach. In *American control conference* (pp. 4677–4682). Denver, CO, USA.

Zimmerman, R. D., Murillo-Sanchez, C. E., & Thomas, R. J. (2011). MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems, 26*(1), 12–19.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *International conference on machine learning*. Washington DC.