IEEE CSS

Received 30 April 2022; revised 4 May 2022 and 27 July 2022; accepted 21 August 2022. Date of publication 14 September 2022; date of current version 24 October 2022. Recommended by Senior Editor Sonia Martinez.

Digital Object Identifier 10.1109/OJCSYS.2022.3205871

Online Optimization of Dynamical Systems With Deep Learning Perception

LILIAOKEAWAWA COTHREN, GIANLUCA BIANCHIN, AND EMILIANO DALL'ANESE

Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, CO 80309 USA

CORRESPONDING AUTHOR: E. Dall'Anese (e-mail: emiliano.dallanese@colorado.edu).

This work was supported by the National Science Foundation (NSF) through the awards CMMI 2044946 and CAREER 1941896.

ABSTRACT This paper considers the problem of controlling a dynamical system when the state cannot be directly measured and the control performance metrics are unknown or only partially known. In particular, we focus on the design of data-driven controllers to regulate a dynamical system to the solution of a constrained convex optimization problem where: i) the state must be estimated from nonlinear and possibly high-dimensional data; and ii) the cost of the optimization problem – which models control objectives associated with inputs and states of the system – is not available and must be learned from data. We propose a data-driven feedback controller that is based on adaptations of a projected gradient-flow method; the controller includes neural networks as integral components for the estimation of the unknown functions. Leveraging stability theory for perturbed systems, we derive sufficient conditions to guarantee exponential input-to-state stability (ISS) of the control loop. In particular, we show that the interconnected system is ISS with respect to the approximation errors of the neural network and unknown disturbances affecting the system. The transient bounds combine the universal approximation property of deep neural networks with the ISS characterization. Illustrative numerical results are presented in the context of robotics and control of epidemics.

INDEX TERMS Gradient methods, neural networks, optimization, perception-based control, regulation.

I. INTRODUCTION

Control frameworks for modern engineering and societal systems critically rely on the use of perceptual information from sensing and estimation mechanisms. Extraction of critical information for feedback control increasingly requires the processing of high-dimensional sensory data obtained from nonlinear sensory systems [1], [2], [3], [4], and the interpretation of information received from humans interacting with the system regarding the end-user perception of safety, comfort, or (dis)satisfaction [5], [6]. For example, control systems in autonomous driving rely on positioning information extracted from camera images [1] and must account for the perception of the safety of the vehicle occupants [7]. In power grids, state feedback is derived from nonlinear state estimators or pseudo-measurements [8], and control goals must account for comfort and satisfaction objectives of the end-users that are difficult to model [9].

Within this broad context, this paper considers the problem of developing feedback controllers for dynamical systems where the acquisition of information on the system state and on the control performance metrics requires a systematic integration of supervised learning methods in the controller design process. Further, our problem pertains to the design of feedback controllers to steer a dynamical system towards the solution of a constrained convex optimization problem, where the cost models objectives that are associated with the state and the controllable inputs. The design of feedback controllers inspired by first-order optimization methods has received significant attention during the last decade [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]; see also the recent line of work on using online optimization methods for discrete-time linear time-invariant (LTI) systems [20], [21], [22], [23]. However, open research questions remain on how it is possible to systematically integrate learning methods in the control loop when information on the system and on the optimization model is not directly available, and on how to analyze the robustness and safety of optimization-based controllers in the presence of learning and estimation errors.

In this work, we investigate the design of feedback controllers based on an adaptation of the projected gradient flow method [18], [24] combined with learning components, where: i) estimates of the state of the system are provided by feedforward neural networks [25], [26] and residual neural networks [27], [28], and ii) the gradient information is acquired via finite-differences based on a deep neural network approximation of the costs. When the neural network-based controller is interconnected with the dynamical system, we establish conditions that guarantee input-to-state stability [29], [30] by leveraging tools from the theory of perturbed systems [31], Ch. 9] and singular perturbation theory [31], Ch. 11]. In particular, the ISS bounds show how the transient and asymptotic behaviors of the interconnected system are related to the neural network approximation errors. When the system is subject to unknown disturbances, the ISS bounds also account for the time-variability of the disturbances.

Prior works: Perception-based control of discrete-time linear time-invariant systems is considered in, e.g., [1], [2], where the authors study the effect of state estimation errors on controllers designed via system-level synthesis. Further insights on the tradeoffs between learning accuracy and performance are offered in [4]. For continuous-time systems, ISS results for dynamical systems with deep neural network approximations of state observers and controllers are provided in [3]. Differently from [3], we consider the estimation of states and cost functions and the interconnection of optimization-based controllers with dynamic plants. Optimization methods with learning of the cost function are considered in, e.g., [6], [32], [33] (see also references therein); however, these optimization algorithms are not implemented in closed-loop with a dynamic plant. Regarding control problems for dynamical systems, existing approaches leveraged gradient-flow controllers [17], [34], proximal-methods in [14], prediction-correction methods [15], and hybrid accelerated methods [17]. Plants with nonlinear dynamics were considered in [11], [16], and switched LTI systems in [18]. A joint stabilization and regulation problem was considered in [13], [35]. See also the recent survey by [19]. In all of these works, the states and outputs are assumed to be observable and cost functions are known.

We also acknowledge works where controllers are learned using neural networks; see, for example, [3], [36], [37], and the work on reinforcement learning in [38]. Similarly to this literature, we leverage neural networks to supply state and gradient estimates to a projected gradient-flow controller. By analogy with dynamical systems, optimization has been applied to Markov decision processes in, e.g., [39].

Finally, we note that ISS applied to perturbed gradient flows was investigated in [40]. In this work, we consider interconnections between a perturbed, projected gradient-flow and a dynamical system, and combine the theory of perturbed systems [31], Ch. 9] with singular perturbation [31], Ch. 11]. Our ISS bounds are then customized for feedforward neural networks [25], [26] and residual neural networks [27], [28]. We also acknowledge [41], where basis expansions are utilized to learn a function, which is subsequently minimized via extremum seeking.

Finally, the preliminary work [42] used a gradient-flow controller in cases where the optimization cost is learned via least-squares methods. Here, we extend [42] by accounting for systems with nonlinear dynamics, by using neural networks instead of parametric estimation techniques, by considering errors in the state estimates, and by combining ISS estimates with neural network approximation results.

Contributions: The contribution of this work is threefold. First, we characterize the transient performance of a projected gradient-based controller applied to a nonlinear dynamical system while operating with errors in the gradient. Our analysis is based on tools from ISS analysis of nonlinear dynamical systems. More precisely, we leverage Lyapunov-based singular perturbation reasonings to prove that the proposed control method guarantees that the controlled system is ISS with respect to the variation of exogenous disturbances affecting the system and the error in the gradient. This fact is remarkable because unknown exogenous disturbances introduce shifts in the equilibrium point of the system to control. Second, we propose a framework where optimization-based controllers are used in combination with deep neural networks. We tailor our results to two types of deep neural networks that can be used for this purpose: deep residual networks and deep feedforward networks. We then combine the universal approximation property of deep neural networks with the ISS characterization and provide an explicit transient bound for feedback-based optimizing controllers with neural-network state estimators. Third, we propose a novel framework where deep neural networks are used to estimate the gradients of the cost functions characterizing the control goal based on training data. Analogously to the case above, we tailor our results to two cases: deep residual networks and feedforward networks. In this case, we leverage our ISS analysis to show how it is possible to design optimization-based controllers to accomplish the target control task, and we provide an explicit transient bound for these methods. Finally, we illustrate the benefits of the methods in: (i) an application in robotic control and (ii) the problem of controlling the outbreak of an epidemic modeled by using a susceptible-infected-susceptible model. Overall, our results show for the first time that the universal approximation properties of deep neural networks can be harnessed, in combination with the robustness properties of feedback-based optimization algorithms, to provide guarantees in perception-based control.

In conclusion, we highlight that the assumptions and control frameworks outlined in this paper find applications in, for example, power systems [13], [14], [16], [34], traffic flow control in transportation networks [18], epidemic control [43], and in neuroscience [44]. When the dynamical model for the plant does not include exogenous disturbances, our optimization-based controllers can also be utilized in the context of autonomous driving [1], [2] and robotics [45].

Organization: The remainder of this paper is organized as follows. Section II describes the problem formulation and introduces some key preliminaries used in our analysis. Section III presents a main technical result that characterizes an error bound for gradient-type controllers with arbitrary gradient error. In Sections IV and V, we present our main control algorithms corresponding to the case of state perception and cost perception, respectively. Section VI and Section VII illustrate our simulation results and Section VIII concludes the paper.

II. PRELIMINARIES AND PROBLEM FORMULATION

We first outline the notation used throughout the paper and provide relevant definitions.

Notation: We denote by $\mathbb{N}, \mathbb{N}_{>0}, \mathbb{R}, \mathbb{R}_{>0},$ and $\mathbb{R}_{\geq 0}$ the set of natural numbers, the set of positive natural numbers, the set of real numbers, the set of positive real numbers, and the set of non-negative real numbers. For vectors $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, $\|x\|$ denotes the Euclidean norm of x, $\|x\|_\infty$ denotes the supremum norm, and $(x, u) \in \mathbb{R}^{n+m}$ denotes their vector concatenation; $x^\top$ denotes transposition, and x_i denotes the i-th element of x. For a matrix $A \in \mathbb{R}^{n\times m}$, $\|A\|$ is the induced 2-norm and $\|A\|_\infty$ the supremum norm.

The set $\mathcal{B}_n(r) := \{z \in \mathbb{R}^n : \|z\| < r\}$ is the open ball in $\mathbb{R}^n$ with radius $r > 0$; $\mathcal{B}_n[r] := \{z \in \mathbb{R}^n : \|z\| \leq r\}$ is the closed ball. Given two sets $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$, $\mathcal{X} \times \mathcal{Y}$ denotes their Cartesian product; moreover, $\mathcal{X} + \mathcal{B}_n(r)$ is the open set defined as $\mathcal{X} + \mathcal{B}_n(r) = \{x + y : x \in \mathcal{X}, y \in \mathcal{B}_n(r)\}$. Given a closed and convex set $\mathcal{C} \subset \mathbb{R}^n$, $\Pi_{\mathcal{C}}\{\cdot\}$ denotes the Euclidean projection onto the closed and convex set; i.e., $\Pi_{\mathcal{C}}\{y\} := \arg\min_{x\in\mathcal{C}} \|x - y\|^2$. For a continuously differentiable function $\phi : \mathbb{R}^n \to \mathbb{R}$, $\nabla\phi(x) \in \mathbb{R}^n$ denotes its gradient. If the function is not differentiable at a point x, $\partial\phi(x)$ denotes its subdifferential.

Partial ordering: The first orthant partial order on $\mathbb{R}^n$ is denoted as $\preceq$ and it is defined as follows: for any $x, z \in \mathbb{R}^n$, we say that $x \preceq z$ if $x_i \leq z_i$ for $i = 1, \ldots, n$. We say that a function $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is monotone if for any $x, z \in \mathbb{R}^n$ such that $x \preceq z$, we have that $\phi(x) \preceq \phi(z)$. Finally, the interval $[x, z]$, for some $x, z \in \mathbb{R}^n$, is defined as $[x, z] = \{w \in \mathbb{R}^n : x \preceq w \preceq z\}$.

Set covering: Let $\mathcal{Q}, \mathcal{Q}_s \subset \mathbb{R}^n$, with $\mathcal{Q}$ compact. We say that $\mathcal{Q}_s$ is an ϱ-cover of $\mathcal{Q}$, for some $\varrho > 0$, if for any $x \in \mathcal{Q}$ there exists a $z \in \mathcal{Q}_s$ such that $\|x - z\|_\infty \leq \varrho$. We say that $\mathcal{Q}_s$ is an ϱ-cover of $\mathcal{Q}$ "with respect to the partial order $\preceq$," for some $\varrho > 0$, if for any $x \in \mathcal{Q}$ there exists $w, z \in \mathcal{Q}_s$ such that $x \in [w, z]$ and $\|w - z\|_\infty \leq \varrho$ [28].

A. MODEL OF THE PLANT

We consider systems that can be modeled using continuous-time nonlinear dynamics:

$$\dot{x} = f(x, u, w_t), \qquad x(t_0) = x_0, \tag{1}$$

where $f : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \to \mathbb{R}^n$, with $\mathcal{X} \subseteq \mathbb{R}^n, \mathcal{U} \subseteq \mathbb{R}^{n_u}, \mathcal{W} \subseteq \mathbb{R}^{n_w}$ open and connected sets. In (1), $x : \mathbb{R}_{\geq 0} \to \mathcal{X}$ denotes the state, $x_0 \in \mathcal{X}$ is the initial condition, $u : \mathbb{R}_{\geq 0} \to \mathcal{U}$ is the control input, and $w_t : \mathbb{R}_{\geq 0} \to \mathcal{W}$ is a time-varying exogenous disturbance (the notation w_t emphasizes the dependence on time). In the remainder, we restrict our attention to cases where $u \in \mathcal{U}_c$ at all times, where $\mathcal{U}_c \subset \mathcal{U}$ is compact[1]. Additionally, we assume that the vector field $f(x, u, w)$ is continuously-differentiable and Lipschitz-continuous, with constants L_x, L_u, L_w, respectively, in its variables. We make the following assumptions on (1).

Assumption 1 (Steady-state map): There exists a (unique) continuously differentiable function $h : \mathcal{U} \times \mathcal{W} \to \mathcal{X}$ such that, for any fixed $\bar{u} \in \mathcal{U}, \bar{w} \in \mathcal{W}$, $f(h(\bar{u}, \bar{w}), \bar{u}, \bar{w}) = 0$. Moreover, $h(u, w)$ admits the decomposition $h(u, w) = h_u(u) + h_w(w)$, where h_u and h_w are Lipschitz continuous with constants ℓ_{h_u} and ℓ_{h_w}, respectively. □

Assumption 1 guarantees that, with constant inputs $\bar{u}, \bar{w}$, system (1) admits a unique equilibrium point $\bar{x} := h(\bar{u}, \bar{w})$. Notice that existence of $h(u, w)$ is always guaranteed in cases where, in addition, $\nabla_x f(x, \bar{u}, \bar{w})$ is invertible for any $\bar{u}, \bar{w}$. Indeed, in these cases, the implicit function theorem [16] guarantees that $h(u, w)$ exists and is differentiable, since $f(x, u, w)$ is continuously differentiable.

In this work, we interpret w_t as an unknown exogenous input modeling disturbances affecting the system. We make the following assumption on w_t.

Assumption 2 (Properties of exogenous inputs): For all $t \in \mathbb{R}_{\geq 0}$, $w_t \in \mathcal{W}_c$, where $\mathcal{W}_c \subset \mathcal{W}$ is compact. Moreover, $t \mapsto w_t$ is locally absolutely continuous on $\mathcal{W}_c$. □

Assumption 2 imposes basic continuity and compactness requirements on the exogenous disturbances affecting (1). Following Assumption 2, in the remainder of this paper we denote by $\mathcal{X}_{eq} := h(\mathcal{U}_c \times \mathcal{W}_c)$ the set of admissible equilibrium points of the system (1). We note that in Assumption 1 we consider a decomposition $h(u, w) = h_u(u) + h_w(w)$ so that the Jacobian of $h(u, w)$ with respect to u does not depend on the unknown disturbance w; this property will be leveraged in the implementation of our gradient-based controller. Notably, this assumption is satisfied in, e.g., power systems [13], [14], [16], [34], transportation networks [18], and in neuroscience [44]. Our model clearly subsumes the case where no disturbance w is present, as in the models for, e.g., autonomous driving [1], [2] and robotics [45]. We also emphasize that the dynamics (1) can model both the dynamics of the physical system and of the stabilizing controllers; see, for example, [13], our previous work on LTI systems in [47], and the recent survey [19].

Remark 1 (Compactness of the equilibrium set): Notice that the equilibrium set $\mathcal{X}_{eq}$ is compact. This follows by noting that $\mathcal{U}_c \times \mathcal{W}_c$ is compact, $h(u, w)$ is continuously differentiable, and by application of [48], Theorem 4.14]. Moreover, notice that $\|\nabla_u h(u, \bar{w})\| \leq \ell_{h_u}$ for all $u \in \mathcal{U}_c$, which follows from compactness of $\mathcal{U}_c$, see [48], Ch. 4]. □

Before proceeding, we let r denote the largest positive constant such that $\mathcal{X}_r := \mathcal{X}_{eq} + \mathcal{B}_n(r)$ satisfies $\mathcal{X}_r \subseteq \mathcal{X}$ (see Fig. 1

[1]The sets $\mathcal{U}_c$ and $\mathcal{W}_c$ are assumed compact to reflect hardware and operational constraints in applications such as autonomous driving [1], power systems [14], [16], neuroscience [44], control of epidemics [46].

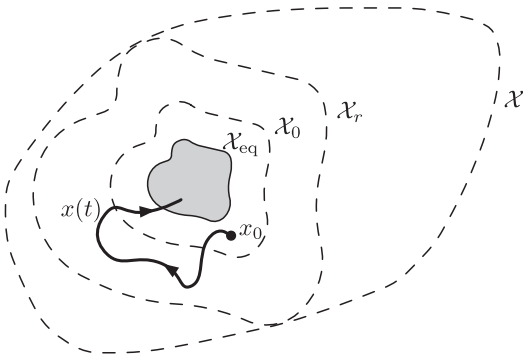


FIGURE. 1. Illustration of notation used for sets and Assumption 3. Continuous lines denote compact sets, dashed lines denote open sets. Assumption 3 guarantees that trajectories that start from the set of initial conditions $\mathcal{X}_0$ do not leave $\mathcal{X}_r$.

for an illustration). For instance, if $\mathcal{X} = \{x \in \mathbb{R}^n : \|x\| < \varrho\}$ and $\mathcal{X}^{\text{eq}} = \{0\}$, then $r = \varrho$.

Assumption 3 (Exponential stability): There exist $a, k > 0$ such that for any fixed $\bar{u} \in \mathcal{U}_c$, $\bar{w} \in \mathcal{W}_c$, the bound

$$\|x(t) - h(\bar{u}, \bar{w})\| \leq k\|x_0 - h(\bar{u}, \bar{w})\|e^{-a(t-t_0)}, \tag{2}$$

holds for all $t \geq t_0$ and for every initial condition $x_0 \in \mathcal{X}_0 := \mathcal{X}_{\text{eq}} + \mathcal{B}_n(r_0)$, $r_0 < r/k$, where $x(t)$ is the solution of (1) with $u(t) = \bar{u}$ and $w(t) = \bar{w}$. □

Assumption 3 guarantees that $\bar{x} = h(\bar{u}, \bar{w})$ is exponentially stable, uniformly in time. This, in turn, implies the existence of a Lyapunov function as formalized in the following result, which is a direct application of [31, Thm. 4.14].

Lemma 1 (Existence of a Lyapunov function for (1)): Let Assumptions 1–3 hold and let $\mathcal{X}_0$ be the set of initial conditions as in Assumption 3. Then, there exists a function $W : \mathcal{X}_0 \times \mathcal{U} \times \mathcal{W} \to \mathbb{R}$ that satisfies the inequalities:

$$d_1\|x - h(u, w)\|^2 \leq W(x, u, w) \leq d_2\|x - h(u, w)\|^2,$$

$$\frac{\partial W}{\partial x} f(x - h(u, w), u, w) \leq -d_3\|x - h(u, w)\|^2,$$

$$\left\|\frac{\partial W}{\partial x}\right\| \leq d_4\|x - h(u, w)\|, \quad \left\|\frac{\partial W}{\partial u}\right\| \leq d_5\|x - h(u, w)\|,$$

$$\left\|\frac{\partial W}{\partial w}\right\| \leq d_6\|x - h(u, w)\|, \tag{3}$$

for some positive constants $d_1 \leq d_2, d_3, d_4, d_5, d_6$. □

Proof: We begin by noting that, under our assumptions, the vector field $f(x, u, w)$ is Lipschitz in $\mathcal{X}_r \times \mathcal{U}_c \times \mathcal{W}_c$, and thus its Jacobian $\frac{\partial f}{\partial x}$ is bounded on $\mathcal{X}_r$, uniformly with respect to u and w. The proof thus follows by iterating the steps in [31], Thm. 4.14] for fixed $u \in \mathcal{U}_c$ and $w \in \mathcal{W}_c$, by noting that Assumption 3 implies that solutions that start in $\mathcal{X}_0$ do not leave $\mathcal{X}_r$, and thus (2) holds. Then, sensitivity with respect to u, w follows from [31], Lemma 9.8] and [49]. ■

In the following, we state the main optimization problem associated with (1) and formalize the problem statements.

B. TARGET CONTROL PROBLEM

In this work, we focus on the problem of controlling, at every time t, the system (1) to a solution of the following time-dependent optimization problem:

$$(u_t^*, x_t^*) \in \arg\min_{\bar{u}, \bar{x}} \ \phi(\bar{u}) + \psi(\bar{x}) \tag{4a}$$

$$\text{s.t.} \ \ \bar{x} = h(\bar{u}, w_t), \quad \bar{u} \in \mathcal{C}, \tag{4b}$$

where $\phi : \mathcal{U} \to \mathbb{R}$ and $\psi : \mathcal{X} \to \mathbb{R}$ describe costs associated with the system's inputs and states, respectively, and $\mathcal{C} \subset \mathcal{U}_c$ is a closed and convex set representing constraints on the input at optimality.

Remark 2 (Interpretation of the control objective): The optimization problem (4) formalizes an *optimal equilibrium selection problem*, where the objective is to select an optimal input-state pair (u_t^*, x_t^*) that, at equilibrium, minimizes the cost specified by $\phi(\cdot)$ and $\psi(\cdot)$. It is worth noting that - differently from stabilization problems, where the objective is to guarantee that the trajectories of (1) converge to *some* equilibrium point - the control objective here is to *select, among all equilibrium points of (1), an equilibrium point that is optimal* as described by the function $\phi(u) + \psi(x)$. In this sense, (4) can be interpreted as a high-level control objective that can be nested with a stabilizing controller (where the latter is used to guarantee the satisfaction of Assumption 3).□

Two important observations are in order. First, the constraint (4b) is parametrized by the disturbance w_t, and thus the solutions of (4) are parametrized by w_t (or, equivalently, by time). In this sense, the pairs (u_t^*, x_t^*) are time-dependent and characterize optimal trajectories [50]. Secondly, by recalling that w_t is assumed to be unknown and unmeasurable, solutions of (4) cannot be computed explicitly.

By recalling that $h(u, w)$ is unique for any fixed u, w, problem (4) can be rewritten as an unconstrained problem:

$$u_t^* \in \arg\min_{\bar{u} \in \mathcal{C}} \ \phi(\bar{u}) + \psi(h(\bar{u}, w_t)). \tag{5}$$

We make the following assumptions on the costs of (5).

Assumption 4 (Smoothness and strong convexity): The following conditions hold:

a) The function $u \mapsto \phi(u)$ is continuously-differentiable and ℓ_u-smooth, $\ell_u \geq 0$. The function $x \mapsto \psi(x)$ is continuously-differentiable and ℓ_x-smooth, $\ell_x \geq 0$.
b) For any $\bar{w} \in \mathcal{W}_c$ fixed, the composite cost $u \mapsto \phi(u) + \psi(h(u, \bar{w}))$ is μ_u-strongly convex, $\mu_u > 0$. □

It follows from Assumption 4(a) that the composite cost $u \mapsto \phi(u) + \psi(h(u, w_t))$ is ℓ-smooth with $\ell := \ell_u + \ell_{h_u}^2 \ell_x$; it follows from Assumption 4(b) that the optimizer (u_t^*, x_t^*) of (5) is unique for any $w_t \in \mathcal{W}$.

Assumption 5 (Regularity of optimal trajectory map): There exists a continuous function $J : \mathcal{W}_c \to \mathcal{U}_c$ such that $u_t^* = J(w_t)$. Moreover, there exists $\ell_J < \infty$ such that $\|\partial J(w_t)\| \leq \ell_J$ for all $w_t \in \mathcal{W}_c$. □

Assumption 5 imposes regularity assumptions on the function that maps w_t (which parametrizes the problem (5)) into the optimal solution u_t^* [51], Ch. 2]; conditions can be

obtained from standard arguments in parametric convex programming.

C. OPTIMAL REGULATION WITH PERCEPTION IN-THE-LOOP

Feedback-based optimizing controllers for (1)–(4) were studied in [18] when (1) has linear dynamics and in [16] when (4) is unconstrained and w_t is constant. The authors consider low-gain gradient-type controllers of the form:

$$\dot{u} = \Pi_{\mathcal{C}} \left\{ u - \eta \left(\nabla \phi(u) + H(u)^\top \nabla \psi(x) \right) \right\} - u, \qquad (6)$$

where $H(u)$ denotes the Jacobian of $h_u(u)$ and $\eta > 0$ is a tunable controller parameter. The controller (6) is of the form of a projected gradient-flow algorithm, often adopted to solve problems of the form (4), yet modified by replacing the true gradient $\nabla \psi(h(u, w_t))$ with the gradient $\nabla \psi(x)$ evaluated at the instantaneous system state, thus making the iteration (6) independent of the unknown disturbance w_t.

Implementations of the controller (6) critically rely on the exact knowledge of the system state x as well as of the gradients $\nabla \phi(u)$ and $\nabla \psi(x)$. In this work, we consider two scenarios. In the first, the controller is used with an estimate $\hat{x}$ of x provided by a deep neural network. More precisely, we focus on cases where x is not directly measurable; instead, we have access only to nonlinear and possibly high-dimensional observations of the state $\xi = q(x)$, where $q : \mathcal{X} \to \mathbb{R}^{n_\xi}$ is an *unknown* map. In the second case, the controller is used with estimates of the gradients $\nabla \phi(u)$, $\nabla \psi(x)$, obtained by using a deep neural network. Similarly to before, we consider cases where the analytic expressions of the gradients are unknown, and instead, we have only access to functional evaluations $\{u_i, \phi(u_i)\}$, $\{x_i, \psi(x_i)\}$ of the cost functions. We formalize these two cases next.

Problem 1 (Optimization with state perception): Design a feedback controller to regulate inputs and states of (1) to the time-varying solution of (4) when x is unmeasurable and, instead, we have access only to state estimates $\hat{x} = \hat{p}(\xi)$ produced by a deep neural network $\hat{p}(\cdot)$ trained as a state observer. □

Problem 2 (Optimization with cost perception): Design a feedback controller to regulate inputs and states of (1) to the time-varying solution of (4) when $\nabla \phi(u)$, $\nabla \psi(x)$ are unknown and, instead, we have access only to estimates $\hat{\phi}(u)$, $\hat{\psi}(x)$ of $\phi(u)$, $\psi(x)$ produced by a deep neural network trained as a function estimator. □

We conclude by discussing in the following remarks the relevance of Problems 1–2 in the applications.

Remark 3 (Motivating applications for Problem 1): In applications in autonomous driving, vehicles states are often reconstructed from perception-based maps $\xi = q(x)$ where q describes images generated by cameras. In a power systems context, $\xi = q(x)$ describes the nonlinear power flow equations describing the relationships between net powers and voltages at the buses (described by ξ) and generators' phase angles and frequencies (described by x). Finally, we note that a related observer design problem was considered in [3]. □

Remark 4 (Motivating applications for Problem 2): When systems interact with humans, $\phi(u)$ is often used to model end-users' perception regarding safety, comfort, or (dis)satisfaction of the adopted control policy [6], [32], [33], [42], [52]. Due to the complexity of modeling humans, $\phi(u)$ is often unknown and learned from available historical data. In robotic trajectory tracking problems, $\psi(x) = \|x - x^r\|^2$ where $x^r \in \mathbb{R}^n$ models an unknown target to be tracked. In these cases, we have access only to measurements of the relative distance $\|x - x^r\|^2$ between the robot the target. Additional examples include cases where $\psi(x)$ represents a barrier function associated with unknown sets [53], [54]. □

III. GENERAL ANALYSIS OF GRADIENT-FLOW CONTROLLERS WITH GRADIENT ERROR

In this section, we take a holistic approach to address Problems 1–2 and we provide a general result characterizing gradient-type controllers of the form (6) that operate with general errors. More precisely, in this section we study the following plant-controller interconnection:

$$\dot{x} = f(x, u, w_t), \qquad (7a)$$

$$\dot{u} = \Pi_{\mathcal{C}} \{ u - \eta (F(x, u) + e(x, u)) \} - u, \qquad (7b)$$

with $x(t_0) = x_0$ and $u(t_0) = u_0$, where $F(x, u) := \nabla \phi(u) + H(u)^\top \nabla \psi(x)$ is the nominal gradient as in (6), and $e : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^{n_u}$ models any state- or input-dependent error.

It is worth noting three important features of the controller (7b). First, (7b) can be implemented without knowledge of w_t (similarly to (6), the true gradient $\nabla \psi(h(u, w_t))$ is replaced by evaluations of the gradient at the instantaneous state $\nabla \psi(x)$). Second, since the vector field in (7b) is Lipschitz-continuous, for any (x_0, u_0) the initial value problem (7) admits a unique solution that is continuously differentiable [18], Lemma 3.2], [24]. Third, the set $\mathcal{C}$ is attractive and forward-invariant for the dynamics (7b); namely, if $u(t_0) \notin \mathcal{U}$, then $u(t)$ approaches $\mathcal{C}$ exponentially, and if $u(t_0) \in \mathcal{C}$, then $u(t) \in \mathcal{U}$ for all $t \geq t_0$ [18], Lemma 3.3].

To state our results, we let $z := (x - x_t^*, u - u_t^*)$ be the tracking error between the state of (7) and the optimizer of (4). Moreover, for fixed $s \in (0, 1)$, define:

$$c_0 := \min \left\{ s\mu\eta, s\frac{d_3}{d_2} \right\}, \qquad c_1 := \frac{1}{\eta} \min \left\{ \frac{(1-\theta)}{2}, \theta d_1 \right\},$$

$$c_2 := \frac{1}{\eta} \max \left\{ \frac{(1-\theta)}{2}, \theta d_2 \right\}, \quad c_3 := \sqrt{2} c_1^{-1},$$

$$c_4 := \sqrt{2\eta} \max \left\{ 1, d_1^{-\frac{1}{2}} \right\}, \qquad c_5 := \frac{\sqrt{2}}{\sqrt{\eta}} \ell_J + \frac{d_4 \ell_{h_w}}{\sqrt{\eta}\sqrt{d_1}}, \qquad (8)$$

and $\theta = (1 + d_4 + \frac{d5}{\ell \ell_{h_u}})^{-1}$, where we recall that $(d_1, d_2, d_3, d_4, d_5)$ are as in Lemma 1.

Theorem 1 (Transient bound for gradient flows with error): Consider the closed-loop system (7) and let Assumptions 1–5 be satisfied. Suppose that, for any $x \in \mathcal{X}_0$ and $u \in \mathcal{U}$, the

gradient error satisfies the condition

$$\|e(x,u)\| \leq \gamma\|z\| + \delta, \tag{9}$$

for some $\delta > 0$ and $\gamma \in [0, c_0/c_3)$. If $\eta \in (0, \eta^*)$, with

$$\eta^* := \min\left\{\frac{2\mu}{\ell^2}, \frac{(1-s)^2 d_3 \mu}{\ell_{h_u}(d_4\ell_{h_u} + d_5)((1-s)\mu\ell_y + 2\ell^2)}\right\}, \tag{10}$$

then, the tracking error satisfies

$$\|z(t)\| \leq \kappa_1 e^{-\frac{\alpha}{2}(t-t_0)}\|z(t_0)\| + \kappa_2 \text{ess sup}_{t_0\leq\tau\leq t}\|\dot{w}_\tau\| + \kappa_3\delta, \tag{11}$$

for all $t \geq t_0$, where $\alpha = c_0 - \gamma c_3$ and

$$\kappa_1 = \sqrt{\frac{c_2}{c_1}}, \qquad \kappa_2 = \frac{c_5}{c_0\sqrt{c_1}}, \qquad \kappa_3 = \frac{c_4}{c_0\sqrt{c_1}},$$

for any $x(t_0) \in \mathcal{D}_0 := \mathcal{X}_{\text{eq}} + \mathcal{B}_n(r')$ where r' is such that $0 < r' < r_o/(\sqrt{2}\kappa_1) - \sqrt{2}\kappa_3\delta - \sqrt{2}\kappa_2 \text{ess sup}_{\tau\geq t_0}\|\dot{w}_\tau\| - \sqrt{2}\kappa_1 \text{diam}(\mathcal{U})$, and for any $u(t_0) \in \mathcal{U}$. □

The proof of this claim is postponed to the Appendix.

Theorem 1 asserts that if the worst-case estimation error $e(x,u)$ is bounded by a term $\gamma\|z\|$ that vanishes at the optimizer and by a nonvanishing but constant term δ, then a sufficiently-small choice of the gain η guarantees exponential convergence of the tracking error to a neighborhood of zero. More precisely, the tracking error z is ultimately bounded by two terms: the first $\kappa_2 \text{ess sup}_{t_0\leq\tau\leq t}\|\dot{w}_\tau\|$ accounts for the effects of the time-variability of w_t on the optimizer (u_t^*, x_t^*), and the second $\kappa_3\delta$ accounts for the effects of a nonvanishing error in the utilized gradient function. It follows that the bound (11) guarantees input-to state stability (ISS) of (7) (in the sense of [30], [40], [55]) with respect to $\dot{w}_t$ and δ.

IV. OPTIMIZATION WITH NEURAL NETWORK STATE PERCEPTION

In this section, we propose an algorithm to address Problem 1 and tailor the conclusions drawn in Theorem 1 to characterize the performance of the proposed algorithm.

A. ALGORITHM DESCRIPTION

To produce estimates of the system state $\hat{x} = p(\xi)$, we assume that a set of training points $\{(\xi^{(i)}, x^{(i)})\}_{i=1}^N$ is utilized to train a neural network via empirical risk minimization. More precisely, in the remainder, we will study two types of neural networks that can be used for this purpose: (i) feedforward neural networks and (ii) residual neural networks[2]. We thus propose to train a neural network to produce a map $\hat{x} = \hat{p}(\xi)$ that yields estimates of the system state given nonlinear and high-dimensional observations ξ. Accordingly, we modify the controller (6) to operate with estimates of the system state $\hat{x}$ produced by the neural network. The proposed framework is described in Algorithm 1 and illustrated in Fig. 2.

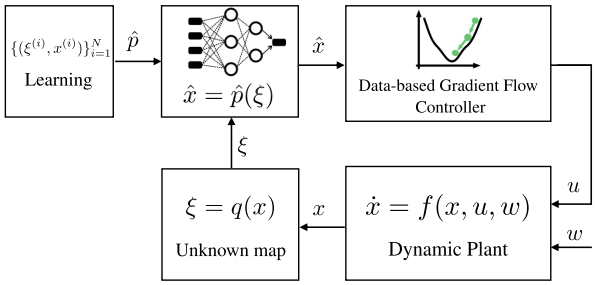


FIGURE. 2. Schematics of the control framework proposed in Algorithm 1. In this setting, the state x of the system is not measurable directly, instead, only high-dimensional estimations (e.g., camera images) $\xi = q(x)$ are available. A deep neural network is then used to compute state estimates $\hat{x} = \hat{p}(\xi)$, which are used to feed a gradient-based feedback controller).

Algorithm 1: Optimization with NN State Perception.

 # Training

Given: training set $\{(x^{(i)}, \xi^{(i)})\}_{i=1}^N$

Obtain: $\hat{p} \leftarrow$ NN-learning$(\{(x^{(i)}, \xi^{(i)})\}_{i=1}^N)$

 # Gradient-based Feedback Control

Given: set $\mathcal{U}$, funct.s $\nabla\phi$, $\nabla\psi$, $H(u)$, neural net $\hat{p}$, gain η

Initial conditions: $x(t_0) \in \mathcal{X}_0$, $u(t_0) \in \mathcal{U}$

For $t \geq t_0$:

$$\dot{x} = f(x, u, w_t) \tag{12a}$$

$$\xi = q(x) \tag{12b}$$

$$\dot{u} = \Pi_{\mathcal{C}}\{u - \eta\left(\nabla\phi(u) + H(u)^\top\nabla\psi(\hat{p}(\xi))\right)\} - u \tag{12c}$$

B. ANALYSIS OF ALGORITHM 1

In what follows, we analyze the tracking properties of Algorithm 1. To this end, we introduce the following.

Assumption 6 (Generative and Perception Maps): The generative map $x \mapsto q(x) = \xi$ is such that, for any compact set $\mathcal{X}' \subseteq \mathcal{X}_r$, the image $q(\mathcal{X}')$ is compact. Moreover, there exists a continuous $p : \mathbb{R}^{n_\xi} \to \mathbb{R}^n$ such that $p(\xi) = x$ for any $x \in \mathcal{X}_r$, where $\xi = q(x)$. □

[2]We refer the reader to the representative papers [25], [28] for an overview of feedforward and residual networks. Briefly, a neural network consists of inputs, various hidden layers, activation functions, and output layers, and can be trained for, e.g., functional estimation and classification. When the layers are sequential and the architecture is described by a directed acyclic graph, the underlying network is called *feedforward*; when some of these layers are bypassed, then the underlying network is called *residual*.

Remark 5 (Relationship with System Observability): We note that a standard approach in the literature for the state observer design problem is to leverage the concept of ℓ-observability, where the state is estimated based on $\ell+1$ samples of ξ and ℓ samples of the inputs u, w [3]. However, in our setup, we do not have access to measurements of the exogenous input w_t. Therefore, we rely on an approach similar to [1], [2], where x is estimated from the observation ξ. □

To guarantee that network training is well-posed, we assume that the N training points $\{x^{(i)}\}_{i=1}^{N}$ for the state are drawn from a compact set $\mathcal{X}_{\text{train}} := \mathcal{X}_{\text{eq}} + \mathcal{B}_n[r_{\text{train}}]$, where r_{train} is such that $r_0 \le r_{\text{train}} < r$. Moreover, we let $\mathcal{Q}_{\text{train}} := q(\mathcal{X}_{\text{train}})$ denote the perception set associated with the training set $\mathcal{X}_{\text{train}}$, and we denote by $\mathcal{Q}_{\text{train},s} := \{\xi^{(i)} = q(x^{(i)}), i = 1,\dots,N\} \subset \mathcal{Q}_{\text{train}}$ the set of available perception samples. Notice that the set $\mathcal{Q}_{\text{train}}$ is compact by Assumption 6. Compactess of $\mathcal{Q}_{\text{train}}$ will allow us to build on the results of [25] and [3], [27] to bound the perception error $\|p(\xi)-\hat{p}(\xi)\|$ on the compact set $\mathcal{Q}_{\text{train}}$. With this background, we let $\sup_{\xi\in\mathcal{Q}_{\text{train,s}}}\|p(\xi)-\hat{p}(\xi)\|_\infty$ denote the supremum norm of the approximation error over $\mathcal{Q}_{\text{train,s}}$.

Remark 6 (Properties of the Training Set) Notice that the set of training data $\mathcal{X}_{\text{train}}$ is assumed to contain the set of initial conditions $\mathcal{X}_0$. This allows us to guarantee that the neural network can be trained over the domain of definition of the Lyapunov function W in Lemma 1 (see Fig. 1 for an illustration). By contrast, if the set $\mathcal{X}_{\text{train}}$ were contained in $\mathcal{X}_0$, then set of initial conditions of (12) must be modified so that the trajectories do not leave the set $\mathcal{X}_{\text{train}}$. □

We begin by characterizing the performance of (12) when residual networks are utilized to reconstruct the system state. For simplicity of exposition, we outline the main result for the case where $n = n_\xi$, and then discuss how to consider the case $n < n_\xi$ in Remark 7.

Proposition 1 (Transient Performance of Algorithm 1 with Residual Neural Network): Consider the closed-loop system (12), let Assumptions 1–6 be satisfied, and assume $n = n_\xi$. Assume that the training set $\mathcal{Q}_{\text{train},s}$ is a ϱ-cover of $\mathcal{Q}_{\text{train}}$ with respect to the partial order $\preceq$, for some $\varrho > 0$. Let $p_{\text{resNet}} : \mathbb{R}^{n_\xi} \to R^{n_\xi}$ describe a residual network, and assume that it can be decomposed as $p_{\text{resNet}} = m + A$, where $m : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\xi}$ is monotone and $A : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\xi}$ is a linear function. If Algorithm 1 is implemented with $\hat{p} = p_{\text{resNet}}$ and $\eta \in (0, \eta^*)$, then the error $z(t) = (x - x_t^*, u - u_t^*)$ of (12) satisfies (11) with $\kappa_1, \kappa_2, \kappa_3$ as in Theorem 1, $\gamma = 0$, and

$$\delta = \ell_{h_u}\ell_x\sqrt{n_\xi}\Big(3\sup_{\xi\in\mathcal{Q}_{\text{train},s}}\|p(\xi)-p_{\text{resNet}}(\xi)\|_\infty + 2\,\omega_p(\varrho) + 2\|A\|_\infty\varrho\Big), \tag{13}$$

where ω_p is a modulus of continuity of p on $\mathcal{Q}_{\text{train}}$. □

Proof: Start by noticing that (12c) can be written in the generic form (7b) with the error $e(x,u)$ given by $e(u,x) = H(u)^\top\nabla\psi(x) - H(u)^\top\nabla\psi(\hat{x}) = H(u)^\top\nabla\psi(p(\xi)) - H(u)^\top\nabla\psi(\hat{p}(\xi))$, by simply adding and subtracting the true gradient $H(u)^\top\nabla\psi(x)$ in (12c). A uniform bound on the norm of $e(u,x)$ over the compact set $\mathcal{Q}_{\text{train}}$ is given by:

$$\begin{aligned}\|e(u,x)\| &\le \ell_{h_u}\|\nabla\psi(p(\xi)) - \nabla\psi(\hat{p}(\xi))\| \\ &\le \ell_{h_u}\ell_x\|p(\xi) - p_{\text{resNet}}(\xi)\| \\ &\le \ell_{h_u}\ell_x\sup_{\xi\in\mathcal{Q}_{\text{train}}}\|p(\xi)-p_{\text{resNet}}(\xi)\|,\end{aligned} \tag{14}$$

where we have used Assumption 1 and the fact that the norm of the Jacobian of h is bounded over the compact set $\mathcal{U}$. Next, notice first that $\|p(\xi)-\hat{p}(\xi)\| \le \sqrt{n_\xi}\|p(\xi)-p_{\text{resNet}}(\xi)\|_\infty$. Since $\mathcal{Q}_{\text{train},s}$ is a ϱ-cover of $\mathcal{Q}_d$ with respect to the partial order $\preceq$, for some $\varrho > 0$, and $p_{\text{resNet}} = m + A$, the infimum norm of the estimation error can be upper bounded as $\sup_{\xi\in\mathcal{Q}_{\text{train}}}\|p(\xi)-p_{\text{resNet}}(\xi)\|_\infty \le 3\sup_{\xi\in\mathcal{Q}_{\text{train},s}}\|p(\xi)-p_{\text{resNet}}(\xi)\|_\infty + 2\,\omega_p(\varrho) + 2\|A_p\|_\infty$ as shown in [28], Theorem 7]. The result then follows from Theorem 1 by setting $\gamma = 0$ and δ as in (13). ■

Proposition 1 shows that the control method in Algorithm 1 guarantees convergence to the optimizer of (4) up to an error that depends only on the uniform approximation error of the adopted neural network. Notice that $\sup_{\xi\in\mathcal{Q}_{\text{train},s}}\|p(\xi)-p_{\text{resNet}}(\xi)\|_\infty$ is a constant that denotes the worse-case approximation error of the training data over the compact set $\mathcal{Q}_{\text{train},s}$. More precisely, the result characterizes the role of the approximation errors due to the use of a neural network in the transient and asymptotic performance of the interconnected system (12).

Remark 7 (Perception in High-Dimensional Spaces): When $n_\xi > n$, one can consider training a neural network to approximate a lifted map $\tilde{p} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\xi}$ defined as $\tilde{p} = \iota \circ p$, where $\iota : \mathbb{R}^n \to \mathbb{R}^{n_\xi}$ is the injection: $\iota(x) = (x_1,\dots,x_n,0,\dots,0)$ for any $x \in \mathbb{R}^n$. From Assumption 6, it follows that $\tilde{p}(\cdot)$ is such that $\tilde{p}(\xi) = (x, 0_{q-n})$ for any $x \in \mathcal{X}_r$, with ξ an observation generated by the map $q(\cdot)$.

In this case, we use the training set $\{(x^{(i)}, 0_{q-n}), \xi^{(i)}\}_{i=1}^{N}$ to train the neural network implementing a map $p_{\text{resNet}} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\xi}$ [3]. Subsequently, the perception map $\hat{p}$ that will be used in (12c) is given by $\hat{p} = \pi \circ p_{\text{resNet}}$, where $\pi : \mathbb{R}^{n_\xi} \to \mathbb{R}^n$ is a projection map that returns the first n entries of its argument, namely, $\pi(y) = (y_1,\dots,y_n)$, for any $y \in \mathbb{R}^q$. In short, the training step NN-learning$(\cdot)$ in Algorithm 1 for this case involves the training of the map p_{resNet}, followed by the projection $\hat{x} = \hat{p}(\xi) = \pi(p_{\text{resNet}}(\xi))$. Finally, we notice that, for this case, the claim in Proposition 14 holds unchanged by replacing $p(\xi)$ with $\tilde{p}(\xi)$. This follows by noting that $\|p(\xi)-\hat{p}(\xi)\| \le \sqrt{n_\xi}\|p(\xi)-p_{\text{resNet}}(\xi)\|_\infty$. □

Remark 8 (Density of Training Set): Proposition 1 requires the training set $\mathcal{Q}_{\text{train},s}$ is a ϱ-cover of $\mathcal{Q}_{\text{train}}$, with respect to the partial order $\preceq$. As pointed out in [3], verifying this condition often involves computing the relative position of the training points and the points in the set $\mathcal{Q}_{\text{train}}$. When this is not possible, [3], Lemma 2] shows that there exists a relationship between ϱ-covering of the set $\mathcal{Q}_{\text{train}}$ with respect to $\preceq$ and the density of the training points. In particular, the authors show

that if the set of training points is a ϱ'-cover of $\mathcal{Q}_{\text{train}}$, then it is also a ϱ-covering for a set the set $\mathcal{Q}'_{\text{train}}$, $\mathcal{Q}_{\text{train}} \subset \mathcal{Q}'_{\text{train}}$, with respect to $\preceq$, for some $\varrho > \varrho'$; see [3, Lemma 2]. □

In the remainder of this section, we focus on characterizing the performance of (12) when a feedforward network is utilized to reconstruct the system state. More precisely, we consider cases where the training set $\{\xi^{(i)}, x^{(i)}\}_{i=1}^N$ is utilized to train n multilayer feedforward networks, each of them implementing a map $p_{\text{feedNet},i} : \mathbb{R}^{n_\xi} \to \mathbb{R}$ that estimates the i-th component of the system state $\hat{x}_i = p_{\text{feedNet},i}(\xi)$. In this case, we assume that Algorithm 1 is implemented with $\hat{p} = p_{\text{feedNet}}$, where $p_{\text{feedNet}}(\xi) := (p_{\text{feedNet},1}(\xi), \ldots, p_{\text{feedNet},n}(\xi))$ in (12). Next, we recall that feedforward neural networks are capable of approximating any measurable function on compact sets with any desired degree of accuracy (see, for instance, [25], [26] and the bounds in [56], [57]).

Proposition 2 (Transient Performance of Algorithm 1 with Feedforward Neural Network): Consider the closed-loop system (12) and let Assumptions 1–6 be satisfied. Suppose that $\hat{p}(\xi) = (p_{\text{feedNet},1}(\xi), \ldots, p_{\text{feedNet},n}(\xi))$, with $p_{\text{feedNet},i}$ approximating the map $p_i(\xi)$. Then, if $\eta \in (0, \eta^*)$, the error $z(t) = (x - x_t^*, u - u_t^*)$ of (12) satisfies (11) with $\gamma = 0$,

$$\delta = \ell_{h_u} \ell_x \sqrt{n} \sup_{\xi \in \mathcal{Q}_{\text{train}}} \|p(\xi) - p_{\text{feedNet}}(\xi)\|_\infty , \qquad (15)$$

and $\kappa_1, \kappa_2, \kappa_3$ as in Theorem 1. □

The proof follows similar steps as in Proposition 1, and it is omitted. Proposition 2 shows that the control method in Algorithm 1 guarantees convergence to the optimizer of (4), up to an error that depends only on the uniform approximation error $\sup_{\xi \in \mathcal{Q}_{\text{train}}} \|p(\xi) - p_{\text{feedNet}}(\xi)\|_\infty$ (computed over the entire training set $\mathcal{Q}_{\text{train}}$). Notice that, with respect to Proposition 1, the adoption of a feedforward network allows us to provide tighter guarantees in terms of the entire set $\mathcal{Q}_{\text{train}}$ (as opposed to the set of available samples $\mathcal{Q}_{\text{train},s}$). We conclude by noting that the bound (15) can be further customized for specific error bounds, given the architecture of the feedforward network [56], [57].

Remark 9 (Noisy generative and perception maps): Assumption 6 is borrowed from [2] and it holds when, for example, q is injective. Although the model in Assumption 6 is used for simplicity, the subsequent analysis of our perception-based controllers can be readily extended to the case where: (i) the perception map imperfectly estimates the state; that is, one has that $p(\xi) = x + \nu$, with $\xi = q(x)$, and where $\nu \in \mathbb{R}^n$ is a bounded error [1]. (ii) When unknown externalities enter the generative map. One way to collectively account for both externalities entering q and for approximate perception map is to use the noisy model $p(q(x)) = x + \nu'$, with $\nu' \in \mathbb{R}^n$ a given error (bounded in norm). The results presented in this section can be readily modified to account for this additional error by adding a term proportional to the norm of ν' in the parameter δ. □

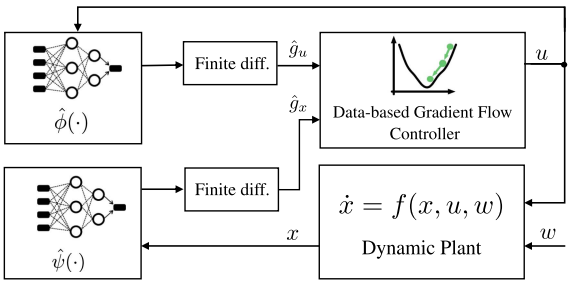


FIGURE. 3. **Schematics of the control framework proposed in Algorithm 2. A gradient-based controller is utilized to control the system to the optimizer of an unknown cost function. The underlying cost function is estimated from data samples through a neural network.**

Algorithm 2: Optimization with NN Cost Perception.

 # Training

Given: $\{u^{(i)}, \phi(u^{(i)})\}_{i=1}^N$, $\{x^{(i)}, \psi(x^{(i)})\}_{i=1}^M$

Obtain:

$$\hat{\phi} \leftarrow \text{NN-learning}(\{(u^{(i)}, \phi(u^{(i)}))\}_{i=1}^N)$$

$$\hat{\psi} \leftarrow \text{NN-learning}(\{(x^{(i)}, \psi(x^{(i)}))\}_{i=1}^M)$$

 # Gradient-based Feedback Control

Given: $x(t_0) \in \mathcal{D}_0$, $u(t_0) \in \mathcal{C}$, NN maps $\hat{\phi}$, $\hat{\psi}$.

For $t \geq t_0$:

$$\dot{x} = f(x, u, w_t) \qquad (16a)$$

$$\hat{g}_u(u) = \sum_{i=1}^{n_u} \frac{1}{2\varepsilon} \left(\hat{\phi}(u + \varepsilon b_i) - \hat{\phi}(u - \varepsilon b_i)\right) b_i, \qquad (16b)$$

$$\hat{g}_x(x) = \sum_{i=1}^{n} \frac{1}{2\varepsilon} \left(\hat{\psi}(x + \varepsilon d_i) - \hat{\psi}(x - \varepsilon d_i)\right) d_i, \qquad (16c)$$

$$\dot{u} = \Pi_{\mathcal{C}}\left\{u - \eta \left(\hat{g}_u(u) + H(u)^\top \hat{g}_x(x)\right)\right\} - u \qquad (16d)$$

V. OPTIMIZATION WITH COST-FUNCTION PERCEPTION

In this section, we propose an algorithm to address Problem 2, and we tailor the conclusions drawn in Theorem 1 to characterize the performance of the proposed algorithm.

A. ALGORITHM DESCRIPTION

To determine estimates of the gradient functions $\nabla\phi(u), \nabla\psi(x)$, we assume the availability of functional evaluations $\{(u^{(i)}, \phi(u^{(i)}))\}_{i=1}^N$ and $\{(x^{(i)}, \psi(x^{(i)}))\}_{i=1}^M$, with $u^{(i)} \in \mathcal{C}$ and $x^{(i)} \in \mathcal{X}_{\text{train}}$. We then consider the training of two neural networks that approximate the functions $u \mapsto \phi(u)$ and $x \mapsto \psi(x)$, respectively, to determine $\hat{\phi}(u)$, $\hat{\psi}(x)$. Accordingly, we modify the controller (6) to operate with estimates of the system state $\hat{x}$ produced by the neural network. The proposed framework is described in Algorithm 2 and illustrated in Fig. 3.

In Algorithm 2, the gradients of the costs are obtained via centered difference, applied to the estimated maps $\hat{\phi}$ and $\hat{\psi}$,

where $\varepsilon > 0$, b_i denotes the i-th canonical vector of $\mathbb{R}^{n_u}$, and d_i is the the i-th canonical vector of $\mathbb{R}^n$. The computation of the approximate gradient $\hat{g}_u$ (respectively, $\hat{g}_x$) thus requires $2n_u$ functional evaluations (respectively, $2n$) of the neural network map $\hat{\phi}$ (respectively, $\hat{\psi}$). The gradient estimates $\hat{g}_u$ and $\hat{g}_x$ are then utilized in the gradient-based feedback controller (16d).

B. ANALYSIS OF ALGORITHM 2

We begin by characterizing the performance of (16) when feedforward networks are utilized to estimate the costs.

Proposition 3 (Transient Performance of Algorithm 2 with Feedforward Neural Network): Suppose that feedforward network maps $\hat{\phi}_{\text{feedNet}}$ and $\hat{\psi}_{\text{feedNet}}$ approximate the costs ϕ and ψ over the compact sets $\mathcal{X}_{\text{train}}$ and $\mathcal{C}_{\text{train}} := \mathcal{C} + \mathcal{B}[\varepsilon]$, respectively. Consider the interconnected system (16), with $\hat{\phi} = \hat{\phi}_{\text{feedNet}}$ and $\hat{\psi} = \hat{\psi}_{\text{feedNet}}$, and let Assumptions 1–5 be satisfied. If $\eta \in (0, \eta^*)$, then error $z(t) = (x - x_t^*, u - u_t^*)$ satisfies (11) with $\kappa_1, \kappa_2, \kappa_3$ as in Theorem 1, $\gamma = 0$, and

$$\delta = e_{u,\text{fd}} + n_u \varepsilon^{-1} \sup_{u \in \mathcal{C}_{\text{train}}} |\phi(u) - \hat{\phi}_{\text{feedNet}}(u)| + \ell_{h_u} e_{x,\text{fd}} + n\varepsilon^{-1} \ell_{h_u} \sup_{u \in \mathcal{X}_{\text{train}}} |\psi(x) - \hat{\psi}_{\text{feedNet}}(x)|, \quad (17)$$

where $e_{x,\text{fd}}$ and $e_{x,\text{fd}}$ are bounds on the centered difference approximation error for the functions ϕ and ψ, respectively; namely, $e_{u,\text{fd}} = \sup_{u \in \mathcal{C}_{\text{train}}} \|\nabla\phi(u) - \sum_{i=1}^{n_u} \frac{1}{2\varepsilon}(\phi(u + \varepsilon b_i) - \phi(u - \varepsilon b_i))b_i\|$ and $e_{x,\text{fd}} = \sup_{x \in \mathcal{X}_{\text{train}}} \|\nabla\psi(x) - \sum_{i=1}^{n} \frac{1}{2\varepsilon}(\psi(x + \varepsilon d_i) - \psi(x - \varepsilon d_i))d_i\|$. □

Proof: We start rewriting (16d) as (7b) by setting $e(x, u) = \nabla\phi(u) - \hat{g}_u(u) + H^\top(\nabla\psi(x) - \hat{g}_x(x))$. Let $g_u(u) := \sum_{i=1}^{n_u} \frac{1}{2\varepsilon}(\phi(u + \varepsilon b_i) - \phi(u - \varepsilon b_i))b_i$ and $g_x(x) := \sum_{i=1}^{n} \frac{1}{2\varepsilon}(\psi(x + \varepsilon d_i) - \psi(x - \varepsilon d_i))d_i$ be the finite difference approximations of the true gradients for brevity. Adding and subtracting $g_u(u)$ and $g_x(x)$ using the triangle inequality, and Assumption 1, we get $\|e(x, u)\| \le \|\nabla\phi(u) - g_u(u)\| + \|g_u(u) - \hat{g}_u(u)\| + \ell_{h_u}\|\nabla\psi(x) - g_x(x)\| + \ell_{h_u}\|g_x(x) - \hat{g}_x(x)\|$. The terms $\|\nabla\phi(u) - g_u(u)\|$ and $\|\nabla\psi(x) - g_x(x)\|$ are errors due to a finite difference approximation of the true gradients, and are bounded by $e_{u,\text{fd}}$ and $e_{x,\text{fd}}$, respectively. On the other hand, $\|g_u(u) - \hat{g}_u(u)\|$ can be bounded as:

$$\begin{aligned}
\|g_u(u) - \hat{g}_u(u)\| &= \Bigg\| \sum_{i=1}^{n_u} \frac{1}{2\varepsilon} (\phi(u + \varepsilon b_i) - \phi(u - \varepsilon b_i))\, b_i \\
&\quad - \sum_{i=1}^{n_u} \frac{1}{2\varepsilon} \left(\hat{\phi}_{\text{feedNet}}(u + \varepsilon b_i) - \hat{\phi}_{\text{feedNet}}(u - \varepsilon b_i)\right) b_i \Bigg\| \\
&\le \sum_{i=1}^{n_u} \frac{1}{2\varepsilon} |\phi(u + \varepsilon b_i) - \hat{\phi}_{\text{feedNet}}(u + \varepsilon b_i)| \|b_i\| \\
&\quad + \sum_{i=1}^{n_u} \frac{1}{2\varepsilon} |\hat{\phi}_{\text{feedNet}}(u - \varepsilon b_i) - \phi(u - \varepsilon b_i)| \|b_i\| \\
&\le \frac{n_u}{2\varepsilon} \sup_{u \in \mathcal{C} + \mathcal{B}[\varepsilon]} |\phi(u) - \hat{\phi}_{\text{feedNet}}(u)|, \quad (18)
\end{aligned}$$

where we used the fact that $\|b_i\| = 1$. Similar steps can be used to bound the error term $\|g_x(x) - \hat{g}_x(x)\|$ to get the final expression for δ in (19). ■

Proposition 3 shows that the control method in Algorithm 2 guarantees convergence to the optimizer of (4), up to an error that depends on the uniform approximation error of the neural networks and on the accuracy of the centered approximation method. More precisely, the result characterizes the role of the approximation errors due to the use of a feedforward neural network in the transient and asymptotic performance of the interconnected system (16).

In the remainder of this section, we focus on characterizing the performance of (16) when a residual network is utilized to reconstruct the system state. To provide guarantees for residual networks, it is necessary to replace $\phi(\cdot)$ by its the lifted counterpart $\tilde{\phi} : \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$, defined as $\tilde{\phi} = \iota_\phi \circ \phi$, where $\iota_\phi : \mathbb{R} \to \mathbb{R}^{n_u}$ is the injection $\iota_\phi(z) = (z, 0, \ldots, 0)$ for any $z \in \mathbb{R}$. Following [28], we consider a residual network map $\hat{\phi}_{\text{resNet}} : \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$ approximating the lifted map $\tilde{\phi}$; the function $\hat{\phi}$ used in (16) is then given by $\hat{\phi}(u) = \hat{\phi}_{\text{resNet}}(u)^\top b_1$, where we recall that b_1 is the first canonical vector of $\mathbb{R}^{n_u}$. Similarly, consider the lifted map $\tilde{\psi} : \mathbb{R}^n \to \mathbb{R}^n$ defined as $\tilde{\psi} = \iota_\psi \circ \psi$, where $\iota_\phi : \mathbb{R} \to \mathbb{R}^n$ is such that $\iota_\psi(z) = (z, 0, \ldots, 0)$ for any $z \in \mathbb{R}$, and consider a residual network map $\hat{\psi}_{\text{resNet}} : \mathbb{R}^n \to \mathbb{R}^n$ approximating the lifted map $\tilde{\psi}$. Accordingly, it follows that $\hat{\psi}(x) = \hat{\psi}_{\text{resNet}}(x)^\top d_1$. With this setup, we have the following.

Proposition 4 (Transient Performance of Algorithm 2 with Residual Neural Network): Suppose that the residual network maps $\hat{\phi}_{\text{resNet}}$ and $\hat{\psi}_{\text{resNet}}$ approximate the functions $\tilde{\phi}$ and $\tilde{\phi}$ over the compact sets $\mathcal{X}_{\text{train}}$ and $\mathcal{C}_{\text{train}}$, respectively. Suppose that the set of training points $\mathcal{C}_{\text{train,s}} := \{u_i^{(i)}\}$ is a ϱ_u-cover of $\mathcal{C}_{\text{train}}$ with respect to the partial order $\preceq$, for some $\varrho_u > 0$, and $\mathcal{X}_{\text{train,s}} = \{x_i^{(i)}\}$ is a ϱ_x-cover of $\mathcal{X}_{\text{train}}$ with respect to the partial order $\preceq$, for some $\varrho_x > 0$. Moreover, suppose that the residual network maps can be decomposed as $\hat{\phi}_{\text{resNet}} = m_u + A_u$ and $\hat{\psi}_{\text{resNet}} = m_x + A_x$, where $m_u : \mathbb{R}^{n_u} \to \mathbb{R}^{n_u}$ and $m_x : \mathbb{R}^n \to \mathbb{R}^n$ are monotone, and A_u, A_x are a linear functions. Consider the interconnected system (16), with $\hat{\phi}(u) = \hat{\phi}_{\text{resNet}}(u)^\top b_1$ and $\hat{\psi}(x) = \hat{\psi}_{\text{resNet}}(x)^\top d_1$, and let Assumptions 1–5 be satisfied. If $\eta \in (0, \eta^*)$, the error $z(t) = (x - x_t^*, u - u_t^*)$ satisfies (11) with $\kappa_1, \kappa_2, \kappa_3$ as in Theorem 1, $\gamma = 0$, and

$$\begin{aligned}
\delta = e_{u,\text{fd}} + n_u^{3/2}\varepsilon^{-1} \left(3e_{u,\text{train}} + 2\,\omega_\phi(\varrho_u) + 2\|A_u\|_\infty\right) \\
+ \ell_{h_u} e_{x,\text{fd}} + n^{3/2}\varepsilon^{-1}\ell_{h_u} \left(3e_{x,\text{train}} + 2\,\omega_\psi(\varrho_x) + 2\|A_x\|_\infty\right) \quad (19)
\end{aligned}$$

where $e_{x,\text{fd}}$ and $e_{x,\text{fd}}$ are defined as in Proposition 3, ω_u, ω_x are the moduli of continuity of $\tilde{\phi}$ and $\tilde{\psi}$, respectively, and

$$e_{u,\text{train}} := \sup_{u \in \mathcal{C}_{\text{train,s}}} \|\tilde{\phi}(u) - \hat{\phi}_{\text{feedNet}}(u)\|_\infty,$$

$$e_{x,\text{train}} := \sup_{x \in \mathcal{X}_{\text{train,s}}} \|\tilde{\psi}(x) - \hat{\psi}_{\text{feedNet}}(x)\|_\infty.$$

□

The proof of Proposition 4 follows similar steps as the proof of Proposition 3 and is omitted for space limitations. Proposition 4 shows that the control method in Algorithm 2 guarantees convergence to the optimizer of (4) up to an error that depends on the uniform approximation error of the neural networks and on the accuracy of the centered approximation method. Notice that, with respect to the characterization in Proposition 3, the use of a residual neural network allows us to characterize the error with respect to accuracy of the available set of samples $\mathcal{C}_{\text{train,s}}$ and $\mathcal{X}_{\text{train,s}}$.

VI. APPLICATION TO ROBOTIC CONTROL

In this section, we illustrate how to apply the proposed framework to control a unicycle robot to track an optimal equilibrium point and whose position is accessible only through camera images. We consider a robot described by unicycle dynamics with state $x = (a, b, \theta)$, where $r := (a, b)^\mathsf{T} \in \mathbb{R}^2$ denotes the position of the robot in a 2-dimensional plane, and $\theta \in (-\pi, \pi]$ denotes its orientation with respect to the $a-$axis [45]. The unicycle dynamics are:

$$\dot{a} = v\cos(\theta), \qquad \dot{b} = v\sin(\theta), \qquad \dot{\theta} = \omega, \tag{20}$$

where $v, \omega \in \mathbb{R}$ are the controllable inputs. Given the dynamics (20), we assume that its state $x = (a, b, \theta)$ is not directly measurable for control purposes, instead, at every time x can be observed only through a noisy camera image, denoted by $\xi = q(x)$; see, e.g. [1], [2]. To tackle the desired problem, we consider an instance of (4) with:

$$\phi(u) = \|u\|^2, \qquad \psi(r) = \|r - r^f\|^2,$$

where $r^f \in \mathbb{R}^2$ denotes the desired final position of the robot.

To address this problem, we consider a two-level control architecture, in which an onboard (low-level) stabilizing controller is first used to stabilize the unicycle dynamics (20) (to guarantee satisfaction of Assumption 3) and, subsequently, the control framework outlined in Algorithm 1 is utilized to select an optimal high-level control references. To design a stabilizing controller, let $u = (u_a, u_b) \in \mathbb{R}^2$ denote the instantaneous high-level control input, and consider the standard change of variables from rectangular to polar coordinates (ξ, ϕ), given by:

$$\xi := \|u - x\|, \qquad \phi := \text{atan2}\left(\frac{u_b - b}{u_a - a}\right) - \theta. \tag{21}$$

In the new variables, the dynamics read as:

$$\dot{\xi} = -v\cos(\phi), \qquad \dot{\phi} = \frac{v}{\xi}\sin(\phi) - \omega. \tag{22}$$

The following lemma provides a stabilizing control law for (22).

Lemma 2: (Stability of unicycle dynamics) The unicycle dynamics (22) with the following control law:

$$v = k\xi\cos(\phi), \qquad \omega = k(\cos(\phi) + 1)\sin(\phi) + k\phi, \tag{23}$$

$k > 0$, admit a unique equilibrium point $(\xi, \phi) = (0, 0)$ that is globally exponentially stable.

Proof: Noticing that the dynamics of ξ and ϕ are decoupled, consider the Lyapunov function $V(\phi) = \frac{1}{2}\phi^2$. We have:

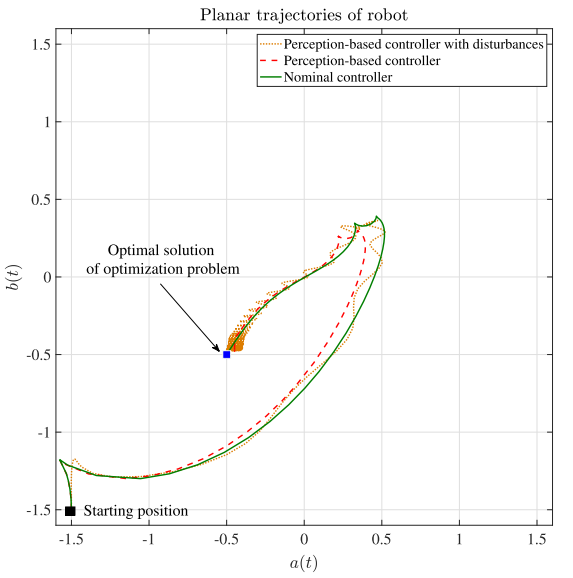
$$\dot{V}(\phi) = -k\phi\sin(\phi) - k\phi^2 \leq -k\phi^2 = -2\,kV(\phi),$$

with $V(\phi) = 0$ if and only if $\phi = 0$. The proof of exponential stability of ξ follows immediately from [45], Lem. 2.1]. ■

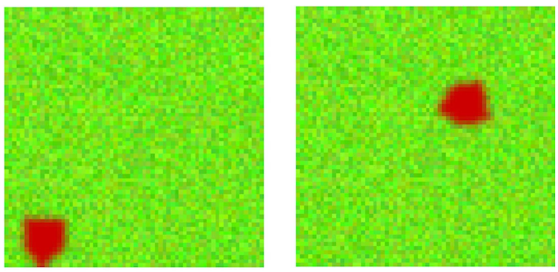
According to Lemma 2, the dynamics (22) with the onboard control law (23) satisfy Assumption 3. We next apply the perception-based control framework outlined in Algorithm 2 to design the reference input u to be utilized in (21).

For our perception-based controller, we use a residual neural network to estimate the state perception map; in particular, the neural network returns estimated state coordinates from aerial images following the procedure of Algorithm 1. To prepare for the execution of Algorithm 1, we generate 122,880 square, blurry, RGB images of size 64×64 pixels to represent the location and orientation of the robot on a two-dimensional plane. These images were built using the *MATLAB Image Processing Toolbox*. First, we build a base image of a maroon pentagon representing the robot over a grassy background (using ranges of RGB values for different green colors), and then add blurriness to the image using the MATLAB function *imgaussfilt*; see, for example, the sample images in Fig. 4(b). This function filters the base image with a 2-D Gaussian smoothing kernel with a standard deviation of 0.5. For our residual network, we used the *resnet50* network structure given in the *MATLAB Deep Learning Toolbox* and tailored the input output sizes for our particular setting. Specifically, we set the input layer to read in images of size 64×64 pixels and set the number of possible output labels as $64^2 = 4996$ to account for all possible locations or coordinates of the pixelized image. The number of training images for the network was chosen so that all possible locations or coordinates of the pixelized image are generated for 30 different orientations, totaling 122,880 training images. After training, the residual network returns estimated locations in the 2D plane by identifying the center pixel of the robot. Of course, during the time-evolution of the closed loop dynamics with the residual network, the returned labels are converted into (a, b) coordinates. We note that, while the MATLAB Image Processing Toolbox allows for classification of images into output labels, future efforts will look at implementations of residual networks for regression.

Simulation results are presented in Fig. 4, when using the initial conditions $(a_0, b_0, \theta_0, u_{a_0}, u_{b_0}) = (-1.5, -1.5, -\pi/2, 1, 1)$; moreover, the optimal solution of the problem is $r^* = (-0.5, -0.5)$. As evidenced in Figure 4(a), differences persist between the trajectories produced by the nominal controller, which utilizes perfect state information, and by the perception-based controller. Importantly, imperfect estimates of (a, b) are also due to the fact that the labels returned from our trained network correspond to the pixels of the image, which can limit



(a)



(b)

FIGURE. 4. Numerical validation of Algorithm 1. The technique is used to control a robot whose state (a, b, θ) cannot be directly measured and is instead estimated using a residual neural network from aerial images. (a) Trajectory of the robot (in the coordinates (a, b)) with the perception-based controller and with a nominal controller using perfect state information. (b) Images that the perception-based controller uses at $t = 4.32$ seconds (left image) and $t = 70.36$ seconds (right image).

how well the (a, b) values are represented. Fig. 4(b) shows sample images at times $t = 4.32$ and $t = 70.36$; these images are used as inputs to the neural network. As expected, the ideal controller converges to the reference state arbitrarily well, while the perception-based controller converges to the reference state to within a neighborhood dependent on the error associated with the perceived state. Overall, these simulations demonstrate the claims made in Proposition 1.

VII. APPLICATION TO EPIDEMIC CONTROL

In this section, we apply the proposed framework to regulate the spread of an epidemic by controlling contact-relevant transmissions. The latter is achieved by e.g. selecting the intensity of restrictions such as mask-wearing, social restrictions, school closures, and stay-at-home orders, etc. To describe the epidemic evolution, we adopt a susceptible-infected-susceptible (SIS) model [58], described by:

$$\dot{s} = \mu - \mu s - u\beta s x + \gamma x, \quad \dot{x} = u\beta s x - (\gamma + \mu)x, \quad (24)$$

where $s \in \mathbb{R}, x \in \mathbb{R}$ describe the fraction of susceptible and infected population, respectively, with $s + x = 1$ at all times, $u \in (0, 1]$ is an input modeling the reduction in contact-relevant transmissions, $\beta > 0$ is the transmission rate, $\mu > 0$ is the death/birth rate, and $\gamma > 0$ is the recovery rate. Model parameters of (24) are chosen as follows: $\beta = 4$, $\gamma = 1/9$, $\mu = 10^{-4}$, cost function parameters: $u^{\text{ref}} = 0.36$, $x^{\text{ref}} = 0.85$, $w_\psi = w_\psi = 1$. As characterized in [58], Thm. 2.1 and Lem. 2.1], (24) admits a unique (unstable) equilibrium point described by $x = 0$ (called disease-free equilibrium) and a unique equilibrium point with $x \neq 0$ (called endemic equilibrium) that is exponentially stable [58], Thm. 2.4], thus satisfying Assumption 3. We utilize the control problem (4) to determine an optimal balance between a desired fraction of infections x^{ref} and a desired level of restrictions u^{ref}. More formally, we consider an instance of (4) with $\phi(u) = w_\phi(u - u^{\text{ref}})^2$ and $\psi(x) = w_\psi(x - x^{\text{ref}})^2$, where $u^{\text{ref}}, x^{\text{ref}} \in [0, 1]$ are desired reference inputs and states and $w_\phi, w_\psi \in \mathbb{R}_{\geq 0}$ are weighting factors. For our simulations, we perform the change of variables $(\tilde{x}, \tilde{u}) = (x, \frac{1}{u})$; in the new variables, $h(\tilde{u}) = 1 - \frac{\mu+\gamma}{\beta}\tilde{u}$ satisfies Assumptions 1 and 4. In order to illustrate the effects of perception on the control loops, in what follows we assume $w_t = 0$ so all the tracking error can be associated to perception errors.

A. OPTIMIZATION WITH STATE PERCEPTION

We begin by illustrating the case of state perception (Section IV). One of the main challenges in predicting and controlling the outbreak of an epidemic is related to the difficulty in estimating the true number of infections from incomplete information describing documented infected individuals with symptoms severe enough to be confirmed. During the outbreak of COVID-19, researchers have proposed several methods to overcome these challenges and by using several sources of data including [59] detected cases, recovered cases, deaths, test positivity [60], and mobility data [46], [61]. In our simulations, we adopted the approach in Algorithm 1 to achieve this task. For the purpose of illustration, we utilized a map $q(\cdot)$ composed of a set of 4 Gaussian basis functions with mean $\mu_b = (1, 5, 9, 13)$ and variance $\sigma = I$ to determine the perception signal ξ. The training phase of Algorithm 1 has then been performed using a feedforward neural network to determine the map $\hat{x} = \hat{p}(\xi)$ to reconstruct the state of (24). Simulation results are illustrated in Fig. 5. As illustrated by the state trajectories in Fig. 5(a)-(b), the use of a neural network originates discrepancies between the true state and the estimated state, especially during transient phases. Fig. 5(c) provides a comparison of the tracking error between the use of the ideal controller (6) (which uses the exact state) and the controller in Algorithm 1. The numerics illustrate that while the exact controller (6) is capable of

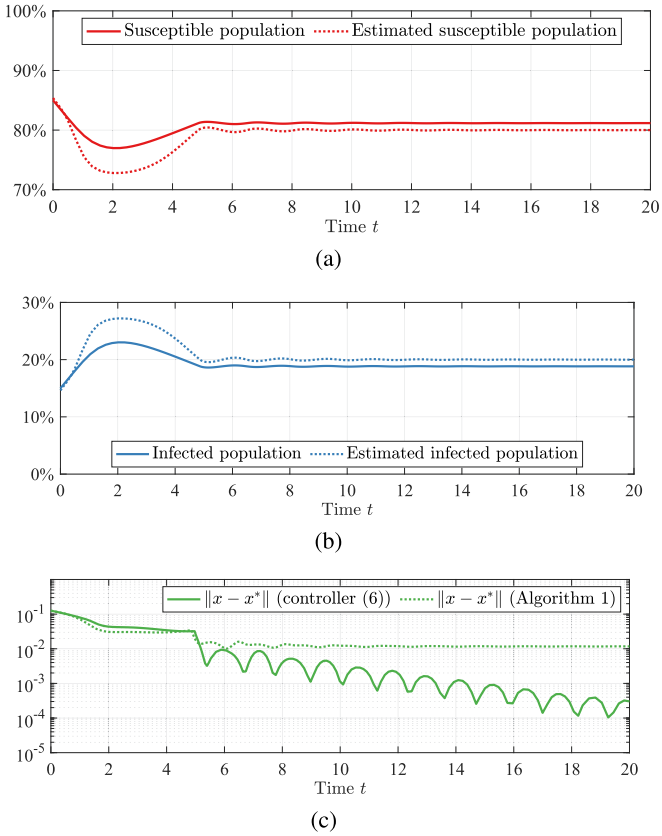


FIGURE. 5. **Numerical validation of Algorithm 1. The technique is used to control an epidemic SIS model whose state cannot be directly measured and is instead reconstructed using a neural network. Panels (a)–(b) compare the time-evolution of the true states and of the estimated states. Panel (c) compares the performance of the ideal controller (6) with that of the perception-based counterpart described in Algorithm 1. Steady-state errors in Algorithm 1 are associated to errors originated by the use of a neural network (see Proposition 2).**

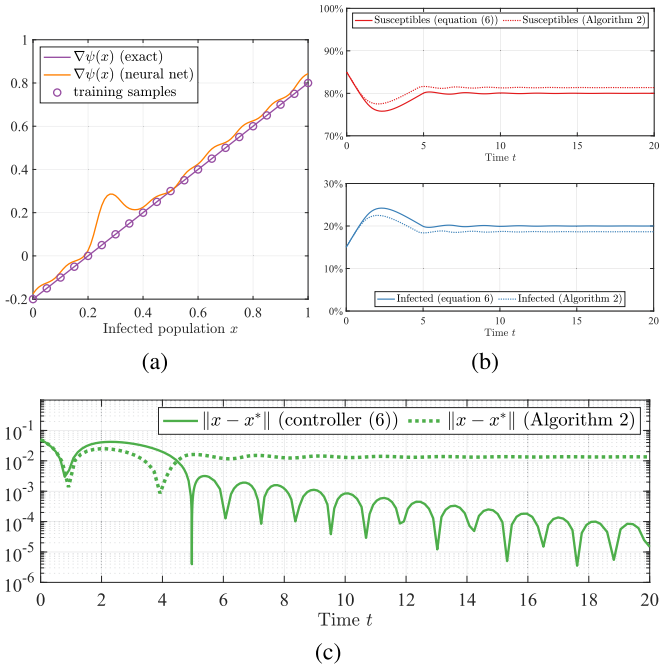


FIGURE. 6. **Numerical validation of Algorithm 2. The technique is used to control an epidemic SIS model, where the economic impact of infections is unknown and must be estimated from samples using a neural network. Panel (a) illustrates the approximation accuracy of the neural network (used to approximate the gradient of the function). Panel (b) compares the time-evolution of the true states obtained by using Algorithm 2 and those obtained by using the ideal controller (6). Panel (c) compares the tracking error of the ideal and perception-based control methods. Steady-state errors in Algorithm 2 are associated to errors originated by the use of a neural network to compute the gradients (see Proposition 3).**

converging to the desired optimizer with arbitrary accuracy, the controller in Algorithm 1 yields an error of the order 10^{-2}, due to uncertainties in the reconstructed system state. Overall, the simulations validate the convergence claim made in Proposition 2.

B. OPTIMIZATION WITH COST-FUNCTION PERCEPTION

Next, we illustrate the case of cost perception (Section V). For illustrative purposes, we focus on cases where the analytic expression of $\phi(u)$ in (4) is known, while $\psi(x)$ is unknown. As described in Algorithm 2, we utilized a set of samples $\{(x_i, \psi(x_i))\}_{i=1}^{M}$ to train a feedforward neural network to approximate the function $\psi(x)$. Simulation results are illustrated in Fig. 6. Fig. 6(a) illustrates the set of samples used for training and provides a comparison between the true gradient $\nabla\psi(x)$ and the approximate gradient $\hat{g}_x(x)$, obtained through the neural network. Fig. 6(b) provides a comparison between the state trajectories obtained by using the ideal controller (6) and those obtained through the perception-based controller in Algorithm 2. Fig. 5(c) provides a comparison of the tracking error between the use of the ideal controller (6) (which uses the exact gradients of the cost) and the controller in Algorithm 2. The simulations illustrate that while the exact controller (6) is capable of converging to the desired optimizer with arbitrary accuracy, the controller in Algorithm 2 yields a nontrivial steady-state error due to uncertainties in the gradient function. Overall, the simulations validate the convergence claim made in Proposition 3.

VIII. CONCLUSION

We proposed algorithms to control and optimize dynamical systems when the system state cannot be measured and the cost functions in the optimization are unknown, but instead, can only be sensed via neural network-based perception. Our results show for the first time how feedback-based optimizing controllers can be adapted to operate with perception in the control loop. Our findings crucially hinge on recent results on the uniform approximation properties of deep neural networks. We believe that the results can be further refined to account for cases where the training of the neural networks is performed online and we are currently investigating such possibility. While this paper provided conditions to obtain exponential ISS results for the interconnection of a plant with

a projected gradient-flow controller, future efforts will analyze the interconnection of plants and controllers that are (locally) asymptotically stable when considered individually.

APPENDIX A
PROOF OF THEOREM 1

The proof of this claim uses Lyapunov-based singular perturbation reasonings, and is organized into seven main steps.

(1 – Change of variables) We consider the change of variables $\tilde{x} = x - h(u, w)$, which shift the equilibrium of (1) to the origin. In the new variables, (7) reads as:

$$\dot{\tilde{x}} = f(\tilde{x} + h(u, w), u, w) - \frac{d}{dt}h(u, w), \tag{25a}$$

$$\dot{u} = \Pi_{\mathcal{C}}\{u - \eta(\tilde{F}(\tilde{x}, u) + e(\tilde{x} + h(u, w_t), u)\} - u, \tag{25b}$$

where we denoted, in compact form, $\tilde{F}(\tilde{x}, u) := F(\tilde{x} + h(u, w), u)$, $\tilde{e}(\tilde{x}, u) := e(\tilde{x} + h(u, w_t), u)$. Before proceeding, we note that:

$$\begin{aligned}\|\tilde{F}(\tilde{x}, u) - \tilde{F}(0, u_t^*)\| &\leq \|\tilde{F}(\tilde{x}, u) - \tilde{F}(0, u)\| \\ &\quad + \|\tilde{F}(0, u) - \tilde{F}(0, u_t^*)\| \\ &\leq \ell_{h_u}\ell_y\|\tilde{x}\| + \ell\|u - u_t^*\|, \end{aligned} \tag{26}$$

where we recall that $(\ell_{h_u}, \ell_y, \ell)$ are as in Assump. 1 and 4(a).

(2 – Lyapunov functions) Inspired by singular perturbation reasonings [31], Ch. 11], we adopt the following composite Lyapunov function for (25):

$$v(\tilde{x}, u, t) := \theta\frac{1}{\eta}W(\tilde{x}, u, w_t) + (1 - \theta)\frac{1}{\eta}V(u, u_t^*), \tag{27}$$

where θ is as defined in (8), $W(\tilde{x}, u, w)$ describes a Lyapunov function for (25a) and is given in Lemma 1, and $V(u, u_t^*)$ describes a Lyapunov function for (25b) and is given by: $V(u, u_t^*) = \frac{1}{2}\|u - u_t^*\|^2$.

Before proceeding, we notice that $v(\tilde{x}, u, t)$ satisfies

$$c_1\|(\tilde{x}, u - u_t^*)\|^2 \leq v(\tilde{x}, u, t) \leq c_2\|(\tilde{x}, u - u_t^*)\|^2,$$

where c_1, c_2 are as in (8).

(3 – Bound for the Time-Derivative of W) We begin by bounding the time-derivative $\frac{d}{dt}W$. To this end, notice that Lemma 1 guarantees the following estimates:

$$\begin{aligned}\frac{\partial W}{\partial \tilde{x}}f(\tilde{x} + h(u, w), u, w) &\leq -d_3\|\tilde{x}\|^2, \\ \frac{\partial W}{\partial \tilde{x}}\frac{d}{dt}h(u, w) &= \frac{\partial W}{\partial \tilde{x}}(\nabla_u h(u, w)\dot{u} + \nabla_w h(u, w)\dot{w}) \\ &\leq d_4\|\tilde{x}\|(\ell_{h_u}\|\dot{u}\| + \ell_{h_w}\|\dot{w}\|).\end{aligned}$$

By using the above estimates and Lemma 1, we have:

$$\begin{aligned}\frac{d}{dt}W(\tilde{x}, u, w) \leq\ &-d_3\|\tilde{x}\|^2 + (d_4\ell_{h_u} + d_5)\|\dot{u}\|\|\tilde{x}\| \\ &+ (d_4\ell_{h_w} + d_6)\|\dot{w}_t\|\|\tilde{x}\|.\end{aligned}$$

Next, refine the bound on the second term. Notice that

$$\begin{aligned}\|\dot{u}\| &= \|\Pi_{\mathcal{C}}\left\{u - \eta\tilde{F}(\tilde{x}, u) + \eta\tilde{e}(\tilde{x}, u)\right\} - u\| \\ &\leq \eta\|\tilde{F}(\tilde{x}, u) - \tilde{F}(0, u_t^*)\| + \eta\|\tilde{e}(\tilde{x}, u)\| \\ &= \eta\ell_{h_u}\ell_y\|\tilde{x}\| + \eta\ell\|u - u_t^*\| + \eta\|\tilde{e}(\tilde{x}, u)\|.\end{aligned}$$

To obtain the first inequality, add $u^* - \Pi_{\mathcal{C}}\{u^* - \eta\tilde{F}(0, u^*)\} = 0$ and apply the triangle inequality. Then, apply the bounds obtained in (26). It follows that

$$\begin{aligned}\frac{d}{dt}W(\tilde{x}, u, w) &\leq -d_3\|x'\|^2 + \eta\ell_{h_u}\ell_y(d_4\ell_{h_u} + d_5)\|\tilde{x}\|^2 \\ &\quad + \eta\ell(d_4\ell_{h_u} + d_5)\|u - u^*\|\|\tilde{x}\| \\ &\quad + \eta(d_4\ell_{h_u} + d_5)\|e(\tilde{x}, u)\|\|\tilde{x}\| + (d_4\ell_{h_w} + d_6)\|\dot{w}_t\|\|x\|.\end{aligned} \tag{28}$$

(4 – Bound for the Time-Derivative of V) Next, we bound the time-derivative $\frac{d}{dt}V$. In what follows, we use the compact notation $\tilde{u} := u - u_t^*$. By expanding:

$$\frac{d}{dt}V(u, u_t^*) = \tilde{u}^\top\dot{u} - \tilde{u}^\top\dot{u}_t^* \leq \tilde{u}^\top\dot{u} + \ell_J\|\dot{w}\|\|\tilde{u}\|, \tag{29}$$

where we used Assumption 5. The first term above satisfies:

$$\begin{aligned}\tilde{u}^\top\dot{u} \leq\ &\tilde{u}^\mathsf{T}(\Pi_{\mathcal{C}}\{u - \eta\tilde{F}(0, u)\} - u) \\ &+ \eta\ell_{h_u}\ell\|\tilde{x}\|\|\tilde{u}\| + \eta\|\tilde{u}\|\|\tilde{e}(\tilde{x}, u)\|.\end{aligned} \tag{30}$$

To obtain (30), we added and subtracted $\Pi_{\mathcal{C}}\{u - \eta F(0, u)\}$, we applied the triangle inequality, and we used Assumptions 1 and 4(a). Next, we use the fact that the first term on the right hand side of (30) satisfies:

$$\tilde{u}^\top(P_u - u) \leq -\eta\left(\mu - \eta\ell^2/4\right)\|\tilde{u}\|^2. \tag{31}$$

(This fact will be proven in Step 5, shortly below.) By combining (29)–(30)–(31) we conclude that:

$$\begin{aligned}\frac{d}{dt}V \leq\ &-\eta\left(\mu - \eta\frac{\ell^2}{4}\right)\|\tilde{u}\|^2 + \eta\ell_{h_u}\ell\|\tilde{x}\|\|\tilde{u}\| \\ &+ \eta\|\tilde{u}\|\|e(\tilde{x}, u)\| + \ell_J\|\dot{w}\|\|\tilde{u}\|.\end{aligned} \tag{32}$$

(5 – Proof of (31)) For brevity, in what follows we use the compact notation $P_u := \Pi_{\mathcal{C}}\{u - \eta\tilde{F}(0, u)\}$. To prove (31), we recall the following property of the projection operator:

$$\left(u' - \Pi_{\mathcal{C}}(v)\right)^\top(\Pi_{\mathcal{C}}(v) - v) \geq 0, \ \forall\, u' \in \mathcal{C},$$

which holds for any $v \in R^{n_u}$. By applying this property with $u' := u_t^*$ and $v = u - \eta\tilde{F}(0, u)$, we have:

$$\left(u_t^* - P_u\right)^\top(P_u - u + \eta\tilde{F}(0, u)) \geq 0.$$

By expanding and by adding $u^\top(u - P_u + \eta\tilde{F}(0, u))$ to both sides of the inequality, the left hand side reads as:

$$u^\top u - u^\top P_u - u_t^{*\top}u + u_t^{*\top}P_u + \eta u^\top\tilde{F}(0, u) - \eta\tilde{F}(0, u)^\top P_u,$$

while the right hand side reads as:

$$u^\top u - P_u^\top u - u^\top P_u^\top + P_u^\top P_u + \eta u^\top\tilde{F}(0, u) - \eta u_t^{*\top}\tilde{F}(u, 0),$$

and, after regrouping,

$$\tilde{u}^\top (u - P_u) \geq \|u - P_u\|^2 + \eta(P_u - u_t^*)^\top \tilde{F}(u, 0).$$

By adding and subtracting $\eta(P_u - u_t^*)^\top \tilde{F}(0, u_t^*)$:

$$\begin{aligned}\tilde{u}^\top (P_u - u) \leq &-\|u - P_u\|^2 - \eta(P_u - u_t^*)^\top \tilde{F}(u_t^*, 0) \\ &- \eta(P_u - u_t^*)^\top (\tilde{F}(0, u) - \tilde{F}(0, u_t^*)).\end{aligned}$$

From this, expand $-(P_u - u_t^*) = u - P_u - \tilde{u}$ and apply $(P_u - u_t^*)^\top \tilde{F}(0, u) \geq 0$. Use strong convexity and Lipschitz-continuity to obtain,

$$\begin{aligned}\tilde{u}^\top (P_u - u) \leq &-\|u - P_u\|^2 - \eta\mu\|\tilde{u}\|^2 + \eta\ell\|u - P_u\|\|\tilde{u}\| \\ \leq &-\eta\left(\mu - \eta\ell^2/4\right)\|\tilde{u}\|^2,\end{aligned}$$

where the second inequality follows by recalling that, for real numbers a, b, it holds $2ab < a^2 + b^2$, and by letting $a = \|u - P_u\|$ and $b = \frac{1}{2}\eta\ell\|\tilde{u}\|$. Hence, (31) is proved.

(6 – Conditions for Exponential Convergence) By combining (28) and (32):

$$\begin{aligned}\frac{d}{dt}\nu \leq (1-\theta)\{&-\left(\frac{\mu - \eta\ell^2}{4}\right)\|\tilde{u}\|^2 + \|\tilde{e}(\tilde{x}, u)\|\|\tilde{u}\| \\ &+ \frac{\ell_J}{\eta}\|\dot{w}_t\|\|\tilde{u}\| + \ell_{h_u}\ell\|\tilde{x}\|\|\tilde{u}\|\} \\ &+ \theta\{-\frac{d_3}{\eta}\|\tilde{x}\|^2 + \ell_{h_u}\ell_y(d_4\ell_{h_u} + d_5)\|\tilde{x}\|^2 \\ &+ \ell(d_4\ell_{h_u} + d_5)\|\tilde{u}\|\|\tilde{x}\| \\ &+ (d_4\ell_{h_u} + d_5)\|\tilde{e}(\tilde{x}, u')\| + \frac{(d_4\ell_{h_w} + d_6)}{\eta}\|\dot{w}_t\|\|x'\|\}.\end{aligned} \tag{33}$$

By using (10), we have $\eta < \frac{2\mu}{\ell^2}$, and thus the first term satisfies $(\frac{\eta\ell^2}{4} - \mu)\|\tilde{u}\|^2 \leq -\frac{\mu}{2}\|\tilde{u}\|^2$. Next, let $s \in (0, 1)$ be a fixed constant. Then, (33) can be rewritten as:

$$\begin{aligned}\frac{d}{dt}\nu \leq (1-\theta)\{&-\frac{(1-s)\mu}{2}\|\tilde{u}\|^2 + \|\tilde{e}(\tilde{x}, u)\|\|\tilde{u}\| \\ &+ \frac{\ell_J}{\eta}\|\dot{w}_t\|\|\tilde{u}\| + \ell_{h_u}\ell\|\tilde{x}\|\|\tilde{u}\|\} \\ &+ \theta\{-\frac{(1-s)d_3}{\eta}\|\tilde{x}\|^2 + \ell_{h_u}\ell_y(d_4\ell_{h_u} + d_5)\|\tilde{x}\|^2 \\ &+ \ell(d_4\ell_{h_u} + d_5)\|\tilde{u}\|\|\tilde{x}\| \\ &+ (d_4\ell_{h_u} + d_5)\|\tilde{e}(\tilde{x}, u)\|\|\tilde{x}\| + \frac{d_4\ell_{h_w} + d_6}{\eta}\|\dot{w}_t\|\|\tilde{x}\|\} \\ &- \underbrace{\min\{s\mu\eta, s\frac{d_3}{d_2}\}}_{=c_0}\nu,\end{aligned} \tag{34}$$

where we used $-(1-\theta)s\frac{\mu}{2}\|\tilde{u}\|^2 = -(1-\theta)s\mu V(u, u_t^*)$ and $-\theta s\frac{d_3}{\eta}\|x'\|^2 \leq -\frac{\theta s}{\eta}\frac{d_3}{d_2}\|x'\|^2 \leq -\frac{\theta s}{\eta}\frac{d_3}{d_2}W$. The bound (34) can be rewritten as:

$$\begin{aligned}\frac{d}{dt}\nu \leq &-c_0\nu - \xi^\top\Lambda\xi + (1-\theta)\{\|\tilde{e}(\tilde{x}, u)\|\|\tilde{u}\| + \frac{\ell_J}{\eta}\|\dot{w}_t\|\|\tilde{u}\|\} \\ &+ \theta\{(d_4\ell_{h_u} + d_5))\|\tilde{e}(\tilde{x}, u)\|\|\tilde{x}\| + \frac{d_4\ell_{h_w} + d_6}{\eta}\|\dot{w}_t\|\|\tilde{x}\|\},\end{aligned} \tag{35}$$

where $\xi = (\|\tilde{u}\|, \|\tilde{x}\|)$ and

$$\Lambda = \begin{pmatrix}(1-\theta)\alpha_1 & -\frac{1}{2}[(1-\theta)\beta_1 + \theta\beta_2] \\ -\frac{1}{2}[(1-\theta)\beta_1 + \theta\beta_2] & \theta[\frac{\alpha_2}{\eta} - \beta_1\beta_2]\end{pmatrix},$$

with $\alpha_1 := (1-s)\frac{\mu}{2}$, $\alpha_2 := (1-s)d_3$, $\beta_1 := \ell\ell_{h_u}$, $\beta_2 := \ell(d_4\ell_{h_u} + d_5)$, and $\gamma := \ell_y\ell_{h_u}(d_4\ell_{h_u} + d_5)$. Finally, we will show that $\xi^\top\Lambda\xi \leq 0$. Equivalently, Λ is positive definite if and only if its principal minors are positive, namely,

$$[(1-\theta)\alpha_1][\theta\left(\frac{\alpha_2}{\eta} - \beta_1\beta_2\right)] > \frac{1}{4}[(1-\theta)\beta_1 + \theta\beta_2]^2,$$

or, equivalently,

$$\eta < \frac{\alpha_1\alpha_2}{\frac{1}{4\theta(1-\theta)}[(1-\theta)\beta_1 + \theta\beta_2]^2 + \alpha_1\beta_1\beta_2}.$$

The right hand side is a concave function of θ and, by maximizing it with respect to θ, we have that the maximum is obtained at $\theta^* = \frac{\beta_1}{\beta_1+\beta_2}$, which gives

$$\eta < \frac{\alpha_1\alpha_2}{\alpha_1\gamma + \beta_1\beta_2},$$

which holds under (10).

(7 – Derivation of Convergence Bound) We begin by showing that (35) can be further refined as follows:

$$\frac{d}{dt}\nu \leq -c_0\nu + c_3\gamma\nu + (c_4\delta + c_5\|\dot{w}_t\|)\sqrt{\nu}. \tag{36}$$

To this end, we first notice that the following inequalities are true: $\|\tilde{u}\| \leq \sqrt{2}\sqrt{V} \leq \sqrt{2}\frac{\sqrt{\eta}}{\sqrt{(1-\theta)}}\sqrt{\nu}$, $\|\tilde{x}\| \leq \frac{\sqrt{W}}{\sqrt{d_1}} \leq \frac{\sqrt{\eta}}{\sqrt{\theta}\sqrt{d_1}}\sqrt{\nu}$ and that $\sqrt{\theta}, \sqrt{1-\theta} \in (0, 1)$ since $\theta \in (0, 1)$. Using these facts, the two positive terms in (35) satisfy:

$$\frac{(1-\theta)}{\eta}\ell_J\|\dot{w}_t\|\|\tilde{u}\| \leq \eta^{1/2}\sqrt{2}\ell_J\|\dot{w}_t\|\sqrt{\nu}$$

$$\theta\frac{d_4\|\nabla_w h(u, w)\|}{\eta}\|\dot{w}_t\|\|\tilde{x}\| \leq d_4\ell_{h_w}\frac{\|\dot{w}_t\|}{\sqrt{\eta}\sqrt{d_1}}\sqrt{\nu},$$

from which (36) follows with c_3, c_4, c_5 as in (8).

To conclude, define $\nu' := \sqrt{\nu}$. Then,

$$\dot{\nu}' \leq -\frac{1}{2}(c_0 - c_3\gamma)\nu' + \frac{1}{2}(c_4\delta + c_5\|\dot{w}_t\|).$$

Define $\Phi(t, t_0) := \exp(-\frac{1}{2}(c_0 - c_3)(t - t_0))$ for $t \geq t_0$. By the comparison lemma [31]:

$$\dot{\nu}' \leq \Phi(t, t_0)\dot{\nu}'(t_0) + \int_{t_0}^{t}\Phi(t, \tau)\left(\frac{c_4}{2}\delta + \frac{c_5}{2}\|\dot{w}_\tau\|\right)d\tau,$$

which implies, by recalling $c_1\|z\|^2 \leq v \leq c_2\|z\|^2$, that

$$\|z(t)\| \leq \sqrt{\frac{c_2}{c_1}}\Phi(t,t_0)\|z(t_0)\| + \frac{c_4}{2\sqrt{c_1}}\int_{t_0}^{t}\Phi(t,\tau)\delta d\tau + \frac{c_5}{2\sqrt{c_1}}\int_{t_0}^{t}\Phi(t,\tau)\|\dot{w}_\tau\|d\tau,$$

from which (11) follows.

REFERENCES

[1] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust guarantees for perception-based control," in *Proc. Learn. for Dyn. Control*, 2020, pp. 350–360.

[2] S. Dean and B. Recht, "Certainty equivalent perception-based control," in *Proc. Learn. for Dyn. Control*, 2021, pp. 399–411.

[3] M. Marchi, J. Bunton, B. Gharesifard, and P. Tabuada, "Safety and stability guarantees for control loops with deep learning perception," *IEEE Control Syst. Lett.*, vol. 6, pp. 1286–1291, 2022.

[4] A. A. Al Makdah, V. Katewa, and F. Pasqualetti, "Accuracy prevents robustness in perception-based control," in *Proc. Amer. Control Conf.*, 2020, pp. 3940–3946.

[5] A. Bajcsy, A. Siththaranjan, C. J. Tomlin, and A. D. Dragan, "Analyzing human models that adapt online," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 2754–2760.

[6] A. Simonetto, E. Dall'Anese, J. Monteil, and A. Bernstein, "Personalized optimization with user's feedback," *Automatica*, vol. 131, 2021, Art. no. 109767.

[7] E. van Nunen, F. Esposto, A. K. Saberi, and J.-P. Paardekooper, "Evaluation of safety indicators for truck platooning," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1013–1018.

[8] E. Manitsas, R. Singh, B. Pal, and G. Strbac, "Modelling of pseudo-measurements for distribution system state estimation," in *Proc. SmartGrids for Distribution*, CIRED, 2008.

[9] N. Lassen, F. Goia, S. Schiavon, and J. Pantelic, "Field investigations of a smiley-face polling station for recording occupant satisfaction with indoor climate," *Building Environ.*, vol. 185, 2020, Art. no. 107266.

[10] A. Jokic, M. Lazar, and P. P.-J. Van Den Bosch, "On constrained steady-state regulation: Dynamic KKT controllers," *IEEE Trans. Autom. Control*, vol. 54, no. 9, pp. 2250–2254, Sep. 2009.

[11] F. D. Brunner, H.-B. Dürr, and C. Ebenbauer, "Feedback design for multi-agent systems: A saddle point approach," in *Proc. IEEE Conf. Decis. Control*, 2012, pp. 3783–3789.

[12] K. Hirata, J. P. Hespanha, and K. Uchida, "Real-time pricing leading to optimal operation under distributed decision makings," in *Proc. Amer. Control Conf.*, Portland, OR, USA, 2014, pp. 1925–1932.

[13] L. S. P. Lawrence, Z. E. Nelson, E. Mallada, and J. W. Simpson-Porco, "Optimal steady-state control for linear time-invariant systems," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 3251–3257.

[14] M. Colombino, E. Dall'Anese, and A. Bernstein, "Online optimization as a feedback controller: Stability and tracking," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 422–432, Mar. 2020.

[15] T. Zheng, J. Simpson-Porco, and E. Mallada, "Implicit trajectory planning for feedback linearizable systems: A time-varying optimization approach," in *Proc. Amer. Control Conf.*, 2020, pp. 4677–4682.

[16] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Timescale separation in autonomous optimization," *IEEE Trans. Autom. Control*, vol. 66, no. 2, pp. 611–624, Feb. 2021.

[17] G. Bianchin, J. I. Poveda, and E. Dall'Anese, "Online optimization of switched LTI systems using continuous-time and hybrid accelerated gradient flows," *Automatica*, vol. 146, 2022, Art. no. 110579.

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

[19] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Optimization algorithms as robust feedback controllers," 2021, *arXiv:2103.11329*.

[20] M. Nonhoff and M. A. Müller, "Online convex optimization for data-driven control of dynamical systems," 2022, *arXiv:2204.13680*.

[21] M. Nonhoff and M. A. Müller, "An online convex optimization algorithm for controlling linear systems with state and input constraints," in *Proc. Amer. Control Conf.*, 2021, pp. 2523–2528.

[22] E. Minasyan, P. Gradu, M. Simchowitz, and E. Hazan, "Online control of unknown time-varying dynamical systems," *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021.

[23] Y. Lin, Y. Hu, G. Shi, H. Sun, G. Qu, and A. Wierman, "Perturbation-based regret analysis of predictive control in linear time varying systems," *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021.

[24] Y. S. Xia and J. Wang, "On the stability of globally projected dynamical systems," *J. Optim. Theory Appl.*, vol. 106, no. 1, pp. 129–150, 2000.

[25] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[26] A. R. Barron, "Approximation and estimation bounds for artificial neural networks," *Mach. Learn.*, vol. 14, no. 1, pp. 115–133, 1994.

[27] P. Tabuada and B. Gharesifard, "Universal approximation power of deep residual neural networks via nonlinear control theory," 2020, *arXiv:2007.06007*.

[28] M. Marchi, B. Gharesifard, and P. Tabuada, "Training deep residual networks for uniform approximation guarantees," in *Proc. Conf. Learn. Dyn. Control*, ser. Proceedings of Machine Learning Research, 2021, vol. 144, pp. 677–688.

[29] E. D. Sontag, "Input to state stability: Basic concepts and results," in *Nonlinear and Optimal Control Theory*. Berlin, Germany: Springer, 2008, pp. 163–220.

[30] D. Angeli, E. D. Sontag, and Y. Wang, "Input-to-state stability with respect to inputs and their derivatives," *Int. J. Robust Nonlinear Control: IFAC-Affiliated J.*, vol. 13, no. 11, pp. 1035–1056, 2003.

[31] H. K. Khalil, *Nonlinear Systems*; 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[32] F. Fabiani, A. Simonetto, and P. J. Goulart, "Learning equilibria with personalized incentives in a class of nonmonotone games," in *Proc. Eur. Control Conf.*, 2022, pp. 2179–2184.

[33] I. Notarnicola, A. Simonetto, F. Farina, and G. Notarstefano, "Distributed personalized gradient tracking with convex parametric models," *IEEE Trans. Autom. Control*, early access, Jan. 31, 2022, doi: 10.1109/TAC.2022.3147007.

[34] S. Menta, A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Stability of dynamic feedback optimization with applications to power systems," in *Proc. Annu. Conf. Commun., Control, Comput.*, 2018, pp. 136–143.

[35] L. S. P. Lawrence, J. W. Simpson-Porco, and E. Mallada, "Linear-convex optimal steady-state control," *IEEE Trans. Autom. Control*, vol. 66, no. 11, pp. 5377–5384, Nov. 2021.

[36] B. Karg and S. Lucia, "Stability and feasibility of neural network-based controllers via output range analysis," in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 4947–4954.

[37] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," *IEEE Trans. Autom. Control*, vol. 67, no. 4, pp. 1980–1987, Apr. 2022.

[38] M. Jin and J. Lavaei, "Stability-certified reinforcement learning: A control-theoretic perspective," *IEEE Access*, vol. 8, pp. 229086–229100, 2020.

[39] P. Marbach and J. N. Tsitsiklis, "Simulation-based optimization of markov reward processes," *IEEE Trans. Autom. Control*, vol. 46, no. 2, pp. 191–209, Feb. 2001.

[40] E. D. Sontag, "Remarks on input to state stability on open sets, motivated by perturbed gradient flows in model-free learning," *Syst. Control Lett.*, vol. 161, 2022, Art. no. 105138.

[41] J. I. Poveda, M. Benosman, and K. G. Vamvoudakis, "Data-enabled extremum seeking: A cooperative concurrent learning-based approach," *Int. J. Adaptive Control Signal Process.*, vol. 35, no. 7, pp. 1256–1284, 2021.

[42] L. Cothren, G. Bianchin, and E. Dall'Anese, "Data-enabled gradient flow as feedback controller: Regulation of linear dynamical systems to minimizers of unknown functions," in *Proc. Conf. Learn. Dyn. Control*, 2022, pp. 234–247.

[43] G. Bianchin, E. Dall'Anese, J. I. Poveda, D. Jacobson, E. J. Carlton, and A. G. Buchwald, "Planning a return to normal after the COVID-19 pandemic: Identifying safe contact levels via online optimization," 2021, *arXiv:2109.06025*.

[44] Y. Yang et al., "Modelling and prediction of the dynamic responses of large-scale brain networks during direct electrical stimulation," *Nature Biomed. Eng.*, vol. 5, no. 4, pp. 324–345, 2021.

[45] A. Terpin, S. Fricker, M. Perez, M. H. de Badyn, and F. Dörfler, "Distributed feedback optimisation for robotic coordination," in *Proc. Amer. Controls Conf.*, 2022, pp. 3710–3715.

[46] G. Bianchin, E. Dall'Anese, J. I. Poveda, D. Jacobson, E. J. Carlton, and A. Buchwald, "Novel use of online optimization in a mathematical model of COVID-19 to guide the relaxation of pandemic mitigation measures," *Sci. Rep.*, vol. 4731, no. 12, Mar. 2022.
[47] F. Galarza-Jimenez, G. Bianchin, J. I. Poveda, and E. Dall'Anese, "Online optimization of LTI systems under persistent attacks: Stability, tracking, and robustness," *Nonlinear Anal. Hybrid Syst.*, vol. 44, 2022, Art. no. 101152.
[48] W. Rudin, *Principles of Mathematical Analysis*, vol. 3. New York, NY, USA: McGraw-Hill, 1976.
[49] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Corrigendum to: "timescale separation in autonomous optimization" [feb 21 611-624]," *IEEE Trans. Autom. Control*, vol. 66, no. 12, pp. 6197–6198, Dec. 2021.
[50] A. Y. Popkov, "Gradient methods for nonstationary unconstrained optimization problems," *Automat. Remote Control*, vol. 66, no. 6, pp. 883–891, 2005.
[51] A. L. Dontchev and R. T. Rockafellar, *Implicit Functions and Solution Mappings*, vol. 543, Berlin, Germany: Springer, 2009.
[52] X. Luo, Y. Zhang, and M. M. Zavlanos, "Socially-aware robot planning via bandit human feedback," in *Proc. Int. Conf. Cyber-Phys. Syst.*, 2020, pp. 216–225.
[53] A. Robey et al., "Learning control barrier functions from expert demonstrations," in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 3717–3724.
[54] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Proc. Learn. for Dyn. Control*, 2020, pp. 708–717.
[55] E. D. Sontag and Y. Wang, "Output-to-state stability and detectability of nonlinear systems," *Syst. Control Lett.*, vol. 29, no. 5, pp. 279–290, 1997.
[56] M. Mehrabi, A. Tchamkerten, and M. Yousefi, "Bounds on the approximation power of feedforward neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3453–3461.
[57] S. Goebbels, "On sharpness of an error bound for deep relu network approximation," *Sampling Theory, Signal Process., Data Anal.*, vol. 20, no. 1, pp. 1–15, 2022.
[58] A. Fall, A. Iggidr, G. Sallet, and J.-J. Tewa, "Epidemiological models and lyapunov functions," *Math. Modelling Natural Phenomena*, vol. 2, no. 1, pp. 62–83, 2007.
[59] E. Armstrong, M. Runge, and J. Gerardin, "Identifying the measurements required to estimate rates of COVID-19 transmission, infection, and detection, using variational data assimilation," *Infect. Dis. Modelling*, vol. 6, pp. 133–147, 2021.
[60] A. Chiu and M. L. Ndeffo-Mbah, "Using test positivity and reported case rates to estimate state-level COVID-19 prevalence and seroprevalence in the USA," *PLoS Comput. Biol.*, vol. 17, no. 9, 2021, Art. no. e1009374.
[61] R. Li et al., "Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2)," *Science*, vol. 368, no. 6490, pp. 489–493, 2020.

LILIAOKEAWAWA COTHREN received the Bachelor of Science degree in mathematics in 2021 from Arizona State University, Tempe, AZ, USA. She is currently working toward the Ph.D. degree with the Department of Electrical Computer and Energy Engineering with the University of Colorado Boulder, Boulder, CO, USA. Her main research interests include online optimization and data-driven control.

GIANLUCA BIANCHIN received the Doctor of Philosophy degree in mechanical engineering from the University of California Riverside, Riverside, CA, USA, in 2020, the Laurea degree in information engineering and the Laurea Magistrale degree in controls engineering from the University of Padova, Padua, Italy, in 2012 and 2014, respectively. He is currently a Postdoctoral Scholar with the Department of Electrical Computer and Energy Engineering with the University of Colorado Boulder, Boulder, CO, USA. His main research interests include modeling, analysis, and control of large-scale interconnected systems, with a focus on transportation networks, and security of cyber-physical systems. He was the recipient of the Dissertation Year Award and of the Dean's Distinguished Fellowship Award from the University of California Riverside.

EMILIANO DALL'ANESE received the Ph.D. degree in information engineering from the Department of Information Engineering University of Padova, Padova, Italy, in 2011. He is currenly an Assistant Professor with the Department of Electrical Computer and Energy Engineering with the University of Colorado Boulder, Boulder, CO, USA, where he is also an Affiliate Faculty with the Department of Applied Mathematics. His research interests include optimization, control, and learning, current applications include power systems and healthcare. He was the recipient of the National Science Foundation CAREER Award in 2020 and the IEEE PES Prize Paper Award in 2021.