# Deep learning-based output tracking via regulation and contraction theory

Samuele Zoboli, Steeven Janny, Mattia Giaccagli

# Deep learning-based output tracking via regulation and contraction theory (Extended Version)

**Samuele Zoboli**[†] [*] **Steeven Janny**[†] [**] **Mattia Giaccagli**[†] [***]

[*] *LAGEPP, UCBL1 (samuele.zoboli@univ-lyon1.fr),*
[**] *INSA-Lyon, LIRIS UMR CNRS 5205 (steeven.janny@insa-lyon.fr),*
[***] *School of Electrical Engineering, Tel Aviv University (giaccagli.mattia@gmail.com).*

**Abstract:** In this paper, we deal with output tracking control problems for input-affine nonlinear systems. We propose a deep learning-based solution whose foundations lay in control theory. We design a two-step state-feedback controller: a contraction-based feedback stabilizer and a feedforward action. The first component guarantees convergence to the steady-state trajectory on which the tracking error is zero. The second one is inherited from output regulation theory and provides forward invariantness of such a trajectory along the solutions of the system. To alleviate the need for heavy analytical computations or online optimization, we rely on deep neural networks and link their approximation error to the tracking one. Mimicking the analytical control structure, we split the learning task into two separate modules. For the stabilizer module, we propose a switching objective function balancing feasibility of the solution and performance improvement. We test our solution in a challenging environment to validate the proposed design.

*Keywords:* Nonlinear systems, output tracking control, deep neural networks, contraction

## 1. INTRODUCTION

Output tracking is arguably among the most versatile applications of control theory, ranging from vertical take-off in aerospace (Martin et al., 1996) and naval ships trajectories (Wondergem et al., 2010) to electronics (Wu and Zou, 2009) and mechanics (Xie and Duan, 2010). The task consists in designing a control action leading the output of a dynamical system to track an arbitrary reference signal. Such a trajectory may be generated from manual design or any external source, depending on the control task. While being fairly simple to address on linear systems (Francis and Wonham, 1976), output tracking remains an open problem for most general nonlinear dynamical systems. In this case, existing approaches either rely on heavy online computation or demand dynamical model knowledge.

Existing solutions canonically address the output tracking problem by exploiting one of the following tools: (i) model inversion, (ii) regulation theory or (iii) optimization. (i) The first method looks for an inverse model mapping the current state-target couple to the input transporting the former to the latter. As an example, we point the reader to solutions based on feedback linearization (Isidori, 1995, Chapter 5.2) or (Devasia et al., 1996) and references therein. (ii) The second design generalizes the linear method: the controller is divided in a dynamical part (the internal model) and a stabilizer (Isidori and Byrnes, 1990). The internal model component includes a generator of the steady-state solution where the tracking error is zero. The stabilizer provides convergence to such a solution (see e.g. Giaccagli et al. (2022a) for the case of constant references). The overall control law guarantees stability, attractivity and forward-invariantness of a manifold where tracking is achieved. As for (i), the approach is strongly model-dependent. Related issues can be alleviated with possible countermeasures such as adaptive techniques (Serrani et al., 2001). Unfortunately, these tricks often leverage challenging analytical considerations, as they usually require a well-defined change of coordinates to bring the system in normal form and, in most cases, a minimum-phase assumption. (iii) Finally, the third method considers output tracking as an optimization problem, which motivates the use of the corresponding tools.

The latter is promising in many ways: it can cope with (small) model errors and it requires moderate theoretical analysis to be deployed, such as Model Predictive Control (MPC) (Limon and Alamo, 2021). The main drawback is the computational effort. MPC usually requires solving an online optimization problem for any point in the trajectory. The complexity of such a task drastically increases for systems presenting significant nonlinearities. To avoid the need of online computation, another class of optimization tools have emerged from the use of convex programming (Tsukamoto and Chung, 2020). Here, an offline optimization problem is solved on isolated points to produce a dataset. The solution is then interpolated by means of a Deep Neural Network (DNN). Deep learning-based controllers are fast, versatile, and very efficient in most situations, but are very data intensive, and often requires a preexisting expert agent to build a sufficiently informative dataset. A different option is to rely on the reinforcement learning philosophy (Bertsekas, 2019), which

[†] Equal contribution from all authors

leverages exploration guided by a weak reward signal. However, theoretical guarantees such as stability are challenging to obtain. Hence, existing results are frequently restricted to simpler classes of systems, e.g., linear ones (Chen et al., 2022).

In this work, we develop a solution to the output tracking problem which intertwines techniques from machine learning and control theory. To do so, we propose a DNN-based algorithm whose backbone comes from output-regulation theory. Formally, we propose a two-step controller. First, we estimate the solution of the regulator equations for a given reference signal, resulting in steady-state trajectory $\pi(t)$ and control action $\psi(t)$ minimizing the tracking error. Then, we design a stabilizer making trajectories asymptotically converge to the reference. To this aim, we rely on the concept of contractive dynamics. Hence, our stabilizer makes the closed-loop a contraction via the results presented in (Giaccagli et al., 2022b) and approximates its analytical solutions with another DNN.

Mixing contraction and DNNs has been recently studied, for instance in (Zhao et al., 2022; Wei et al., 2022; Tsukamoto et al., 2021; Janny et al., 2021; Sun et al., 2021; Dawson et al., 2023). However, our work differs in two main aspects. (i) In the proposed approach, we take advantage of results coming from output regulation theory. The regulator equations are solved thanks to an end-to-end weakly supervised DNN that directly estimates steady-state variables from the reference. We show that our model performs well even on challenging references. (ii) In the former results, the stabilizer is usually implemented by means of Control Contraction Metrics (Manchester and Slotine, 2017). Existing methods require solving an online optimization problem in each point to compute the geodesic (i.e. the shortest path between the system and the trajectory to be tracked). This is usually a computationally demanding task, whose exact solution is not always guaranteed to be found since the optimization task is performed only in compact sets. Conversely, in our design, the structure of the control action is derived analytically, and only its implementation is obtained through the solution of an *offline* optimization problem. Experiments show that such approximation does not hinder the performances of our method.

*Notation:* $|\cdot|$ denotes the Euclidean norm. $\mathcal{B}_\varepsilon \subset \mathbb{R}^n$ denotes the closed ball of radius $\varepsilon > 0$, i.e. $\mathcal{B}_\varepsilon := \{x \in \mathbb{R}^n : |x| \leq \varepsilon\}$. We say that $\zeta : [0, +\infty) \to \mathbb{R}$ is a class-$\mathcal{K}$ function if $\zeta(0) = 0$ and $\zeta$ is strictly increasing. Given a $C^1$ vector field $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ and a $C^1$ 2-tensor $P : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{n\times n}$, the Lie derivative of the 2-tensor $P$ along the vector field $f$, denoted as $L_f P(x,t)$, is defined as $L_f P(x,t) := \dot{P}(x,t) + P(x,t)\frac{\partial f}{\partial x}(x,t) + \frac{\partial f}{\partial x}^\top(x,t)P(x,t)$ with elements $(L_f P(x,t))_{i,j} = \sum_k \big[2P_{ik}\frac{\partial f_k}{\partial x_j}(x,t) + \frac{\partial P_{ij}}{\partial x_k}(x,t)f_k(x,t) + \frac{\partial P_{ij}}{\partial t}(x,t)\big]$. We say that a $C^1$ vector field $g : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$ (resp., a $C^1$ matrix function $g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{n\times m}$) is a Killing vector field (or that it satisfies the Killing vector property) with respect to $P$ if $L_g P(x,t) = 0$ (resp. $L_{g_i} P(x,t) = 0$ for all $i = 1, \ldots, m$, with $g_i$ denoting the $i$-th column of $g$) $\forall (x,t)$. We say that a function $\omega \in \mathcal{L}_2$ if it is measurable and $\int_0^{+\infty} |w(s)|^2 ds < +\infty$.

## 2. PRELIMINARIES

### 2.1 Problem statement

In this paper we consider a system of the form

$$\dot{x} = f(x) + g(x)u \tag{1a}$$
$$e = h(x) - r(t) \tag{1b}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ is a control action, $f, g, h$ are sufficiently smooth functions and $e \in \mathbb{R}^{n_e}$ is the error between an output $y = h(x)$ and a known smooth time-varying reference $r(t)$ taking values on a compact set $\mathcal{R} \subset \mathbb{R}^{n_e}$. In order to simplify the analysis, we assume forward completeness of the trajectories for all times $t \geq t_0, t_0 \in \mathbb{R}$ inside a forward invariant compact set $\mathcal{F} \subset \mathbb{R}^{n_x}$ and we define $\overline{g} := \sup_{x\in\mathcal{F}} |g(x)|$. Our goal is to design a feedback control action $u$ such that the error $e$ asymptotically goes to zero. We formalize our problem as follows. Let $c \geq 0$. Assume to know a smooth function $\gamma : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_u}$ such that the system (1) in closed-loop with the feedback control $u = \gamma(x,t)$ has bounded trajectories and such that $\lim_{t\to+\infty} |e(t)| \leq c$.

Then:

- if $c = 0$, we say that the *asymptotic* output tracking control problem is solved;
- if $c > 0$, we say that the *approximate* output tracking control problem is solved.

### 2.2 Proposed approach

From a regulation theory viewpoint, asymptotic output tracking can be achieved if and only if there exist two sufficiently smooth mappings $\pi : \mathbb{R}^{n_e} \to \mathbb{R}^{n_x}$ and $\psi : \mathbb{R}^{n_e} \to \mathbb{R}^{n_u}$ solution of the so-called *regulator equations* (see for instance (Isidori and Byrnes, 1990; Byrnes and Isidori, 2003))

$$\begin{aligned} \dot{\pi}(r(t)) &= f(\pi(r(t))) + g(\pi(r(t)))\psi(r(t)) \\ 0 &= h(\pi(r(t))) - r(t)\,. \end{aligned} \tag{2}$$

Here, the mapping $\pi$ represents the steady-state manifold on which the tracking error is zero. Hence, the state $x$ must converge asymptotically to it. Similarly, the mapping $\psi$ represents the steady-state control action making such a manifold forward invariant along the trajectories of the system. In order to have a well-posed problem, we assume that if $r(t) \in \mathcal{R}$ for all $t \geq t_0$, then $\pi(t)$ solving (2) is bounded and satisfies $\pi(t) \in \mathcal{F}$ for all $t \geq t_0$. Under perfect knowledge of the model and of the solution of the regulator equations, we look for a controller solving the asymptotic regulation problem and taking the form

$$u = \gamma(x,t) = \psi(t) + \alpha(x, \pi(t), t), \tag{3}$$

where $\psi$ solves (2) and $\alpha$ is any function that forces the dynamics of $x$ to converge to the steady-state $\pi(t)$ and that is asymptotically vanishing. In other words, $\alpha$ is any function such that the trajectories of the closed-loop (1), (3) satisfy $\lim_{t\to+\infty} |x(t) - \pi(t)| = 0$ and such that $\alpha(x, x, t) = 0$ for all $(x,t) \in \mathbb{R}^{n_x} \times \mathbb{R}$. Following these lines, we design $\alpha$ according to a leader-follower multi-agent synchronization strategy. As such, we consider a trivial directed graph whose network is composed of only two agents. In particular, the leader is the (steady-state) dynamics in (2) and the follower is the closed-loop system

(1), (3). With this approach, the problem of designing the function $\alpha$ in (3) can be seen as designing a control action achieving synchronization between two nonlinear systems whose open-loop dynamics is defined by

$$\dot{x} = \varphi(x,t) := f(x) + g(x)\psi(t)\,. \tag{4}$$

For the design, we take inspiration from (Giaccagli et al., 2022b). There, the synchronization problem is cast into the contraction framework. The authors provide a constructive design achieving multi-agent synchronization for a network of input-affine time-varying nonlinear systems.

**Contribution** – The general control structure (3) is inspired by the results in (Pavlov et al., 2006, Section 5.4). Recall in this sense the strong link between contractive and convergent systems. However, we highlight four main differences in our approach. (i) In (Pavlov et al., 2006, Section 5.4) the authors propose the design $\alpha = K(x-\pi)$, with $K$ being a constant matrix. In our design, through the notion of "Killing vector field", we provide a more general structure for the controller $\alpha$. (ii) We show that approximate, rather than asymptotic, tracking can be achieved under a non-perfect knowledge of the mappings $(\pi, \psi)$. (iii) We provide a DNNs-based algorithm for the estimation of $(\pi, \psi)$ (and of the control action $\alpha$) that generalizes over references. (iv) We link the performances of the DNNs to the tracking error.

## 3. MAIN RESULTS

In order to solve the output tracking problem, we propose a controller of the form (3) where $(\pi(t), \psi(t))$ solve (2) and the function $\alpha$ is computed using results of contraction theory applied to a multiagent framework. However, for systems presenting significant nonlinearities, it is often hard to analytically solve the regulator equations and to compute contractive stabilizers. Hence, our approach relies on expressive function approximators, i.e., DNNs. In what follows, we present the practical algorithm to implement the proposed method alongside theoretical results guaranteeing stability of the solution.

### *3.1 Approximate output tracking: the analytic solution*

In order to justify the use of function approximators, we first highlight the robustness properties of the closed-loop system under the controller (3). The goal is to show that (i) under perfect knowledge of the system, of the regulation equations and of the control structure, the asymptotic output tracking problem is solved; (ii) it is still possible to achieve approximate output tracking by means of a approached solution. Then, we show that the tracking error can be straightforwardly linked to the approximation errors of the estimated quantities, and we provide bounds guaranteeing approximate tracking up to arbitrary precision.

We start by assuming the following.

*Assumption 1. Consider system* (1), (2) *and let* $\varphi(x,t) = f(x) + g(x)\psi(t)$. *There exist a* $C^1$ *matrix function* $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ *taking symmetric positive definite values, a* $C^2$ *function* $\beta : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_u}$, *positive real numbers* $\underline{p}, \bar{p}, \varepsilon, \rho > 0$ *such that, for all* $(x,t) \in \mathbb{R}^{n_x} \times \mathbb{R}$, *the following holds:*

*(i) The matrix function* $P$ *satisfies*

$$\begin{aligned} &L_\varphi P(x,t) - \rho P(x,t) g(x) g^\top(x) P(x,t) \preceq -\varepsilon P(x,t)\,, \\ &\underline{p} I \preceq P(x,t) \preceq \bar{p} I\,. \end{aligned} \tag{5}$$

*(ii) The function* $\beta$ *satisfies the integrability condition*

$$\frac{\partial \beta^\top}{\partial x}(x,t) = P(x,t) g(x)\,. \tag{6}$$

*(iii) The Killing vector property*[1] *holds, namely*

$$L_g P(x,t) = 0. \tag{7}$$

Let us briefly give some insight on Assumption 1. Equation (5) can be seen as a nonlinear Riccati-like inequality, where the matrix function $P$ represents a (Riemannian) metric, see (Lohmiller and Slotine, 1998). The uniform bounds on $P$ are required in order to ensure that $x^\top P(x,t)x$ can be taken as radially unbounded Lyapunov function and to show that $|x(t) - \pi(t)| \to 0$ as $t \to +\infty$. Condition (6) is related to the contraction analysis of the closed-loop. Indeed, the evolution of the distance between $x$ and $\pi$ is studied by looking at the Jacobian of the system. Equality (7) characterizes the domain of the synchronizing control law $\alpha$. In particular, it ensures the regulation manifold is globally exponentially stable. Under these conditions, $g$ is said to be a "Killing vector" field for $P$ (or to satisfy the Killing vector property), see e.g. (Manchester and Slotine, 2017, Section III.A) or (Giaccagli et al., 2022a, Section II.B). This implies distances between trajectories of the system are invariant with respect to signals entering along the directions of $g$. For a detailed explanation, see Giaccagli et al. (2022b).

*Remark 1.* For linear systems $\dot{x} = Ax + Bu$, the condition (5) rewrites as the Riccati equation $PA + A^\top P - \rho PBB^\top P \leq -\varepsilon P$ where $P$ is a constant positive definite matrix. In other words, the metric is Euclidean. The integrability constraint (6) is always satisfied by linearity and (7) always holds. In such a case, the conditions of Assumption 1 boil down only to a stabilizability assumption on $(A, B)$.

We are now ready to present our first result.

*Proposition 1. Consider system* (1), (2) *and let* $\varphi(x,t) = f(x) + g(x)\psi(t)$. *Let Assumption 1 hold and let* $\omega : \mathbb{R} \to \mathbb{R}^{n_u}$ *be in* $\mathcal{L}_2$. *Then, for any* $\kappa > \frac{\rho}{2}$, *the trajectories of system* (1) *in closed-loop with*

$$u = \psi(t) + \alpha(x, \pi(t), t) + \omega(t) \tag{8a}$$

*where*

$$\alpha(x, \pi(t), t) = -\kappa(\beta(x,t) - \beta(\pi, t)) \tag{8b}$$

*satisfy*

$$\begin{aligned} &|\mathcal{X}(x_0, t_0, t) - \Pi(\pi_0, t, t_0)| \\ &\qquad \leq k|x_0 - \pi_0| \exp(-\lambda(t - t_0)) + \zeta(|\omega(t)|) \end{aligned} \tag{9}$$

*for all* $(x_0, \pi_0, t, t_0) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times [t_0, \infty) \times \mathbb{R}$, *for some* $k, \lambda > 0$ *and for some class-*$\mathcal{K}$ *function* $\zeta$, *with* $\mathcal{X}(\cdot)$ *being the trajectory of* (1) *in closed-loop and* $\Pi(\cdot)$ *the trajectory of* (2).

*Proof:* See Appendix A.1. □

[1] See Notation section.

*Remark 2.* For systems with significant nonlinearities, the Killing vector condition (7) may be hard to verify. However, it can be relaxed by losing the global characterization of the results in Proposition 1. Indeed, by following the same lines, it is possible to show that for any fixed $\kappa > \frac{\rho}{2}$ and any compact set $\mathcal{E} \subset \mathbb{R}^{n_x}$, there exists $\epsilon_{\mathcal{E}} > 0$ such that if $|L_g P(x,t)| \leq \epsilon_{\mathcal{E}}, \forall \mathcal{E} \subset \mathbb{R}^{n_x}$, then the results in Proposition 1 still hold for $|\omega(t)|$ sufficiently small and for all time-existence of solutions $(\mathcal{X}(\cdot), \Pi(\cdot))$ in $\mathcal{E}^2$ (see Giaccagli et al. (2022b) for more details).

The result of Proposition 1 shows that the control law (8) guarantees that trajectories of (1) in closed-loop remain close to the solution of (2). In particular, the error between the two depends on the component $\omega(t)$ in (8). Our objective is to approximate the control action with DNNs. Then, in our case $\omega(t)$ represents an approximation error. Without full knowledge of the model and of $(\pi(t), \psi(t))$ solution of (2), we end up using a control law of the form

$$u = \hat{\psi}(t) - \kappa(\hat{\beta}(x,t) - \hat{\beta}(\hat{\pi}, t)), \tag{10}$$

where $\hat{\psi}, \hat{\pi}, \hat{\beta}$ represent suitable approximations of the functions $\psi, \pi, \beta$ in (8). In what follows, we link the error in the control action to the approximation capabilities of our DNN structure. More specifically, we show that if the functions $\hat{\psi}, \hat{\pi}, \hat{\beta}$ are sufficiently close to $\psi, \pi, \beta$, then approximate regulation is still achieved. This lays strong foundations for the following section, as we exploit DNNs to learn an approximate version of the exact functions, which are not explicitly computable in general. Hence, via Proposition 1 and the following result, we highlight the link between the approximation and the tracking error.

*Proposition 2. Consider system* (1) *in closed-loop with the control law* (10). *Let* $(\pi, \psi)$ *be a solution of* (2) *and let* $\varphi(x,t) := f(x) + g(x)\psi(t)$. *Let* $(\kappa, \beta)$ *be chosen as in Proposition 1. Then, for any compact sets* $\mathcal{W}_{\tilde{x}} \subset \mathbb{R}^{n_x}$, $\mathcal{R} \subset \mathbb{R}^{n_e}$ *such that* $r(t) \in \mathcal{R}$ *for all* $t \geq t_0$ *and for any* $\delta \geq 0$, *there exist a compact set* $\mathcal{W}_x$ *and a scalar* $\mu_\delta \geq 0$ *such that, if the following holds for all* $(x,t) \in \mathcal{W}_x \times [t_0, \infty)$

$$\begin{aligned} |\hat{\beta}(x,t) - \beta(x,t)| &\leq \mu_\delta, \\ |\hat{\psi}(t) - \psi(t)| &\leq \mu_\delta, \\ |\hat{\pi}(t) - \pi(t)| &\leq \mu_\delta, \end{aligned} \tag{11}$$

*then*

$$\lim_{t \to +\infty} |\mathcal{X}(x_0, t, t_0) - \Pi(\pi_0, t, t_0)| < \delta.$$

*for any* $(x_0, \pi_0)$ *satisfying* $(x_0 - \pi_0) \in \mathcal{W}_{\tilde{x}}$.

*Proof:* See Appendix A.2. □

### *3.2 DNN-based output tracking controller*

Our *ansatz* is to rely on DNNs to find approximate solutions for the output tracking problem. We split our controller into two parts: a steady-state component from solving the regulator equations and a stabilizing part based on Proposition 1. DNNs are typically continuous functions by construction. Hence, if the approximation error over the training dataset is bounded, the error over a compact set including the dataset is also bounded by continuity. This allows the application of Proposition 2. Instead of solving a time-consuming online optimization problem, the steady-state trajectory $\pi(t)$ is generated on the fly by our neural state reference generator. Such a generator is trained offline to solve the regulator equations. The stabilizer is derived from property (6) by modeling $\beta(x,t)$ as a DNN.

In what follows, since time-dependency of $\varphi$ is due only to the reference $r(t)$ (because of the tracking task), we consider $t_0 = 0$. Thus, we discretize the problem using Euler scheme with a small timestep $\tau$, yielding $x(k\tau) = x_k$. We refer to $\mathfrak{f}(\cdot)$ as multi-layer perceptron (MLP) and $\upsilon_{k+1} = \mathfrak{g}(\cdot, \upsilon_k)$ is a gated recurrent unit (GRU) with hidden state $\upsilon$ (Cho et al., 2014) where we omitted gating functions from the notation [2].

**The state reference generator** – estimates state and command trajectories $(\pi, \psi)$ verifying the dynamics (4) such that the observed part $h(\pi)$ remains as close as possible to the reference $r$. Coherence with the dynamics can be ensured by forecasting solely the initial state $\pi(t{=}0) = \pi_0$ and the set of steady-state inputs $\{\psi_k\}_{k \in [0,K]}$. The entire state trajectory can then be inferred readily from the dynamical model. Since the plant may not be fully observable from a single point, estimating the initial state can take advantage of longer reference signal $r{=}\{r_k\}_{k \in [0,K]}$. We thus propose the following structure:

$$\begin{cases} q_{k+1} = \mathfrak{g}_1(r_k, q_k) \\ \pi_0 \quad = \mathfrak{f}_1(q_K) \end{cases}, \begin{cases} z_{k+1} = \mathfrak{g}_2(\pi_k, r_{k+1}, z_k) \\ \psi_k \quad = \mathfrak{f}_2(z_k) \end{cases} \tag{12}$$

We have two components: one dedicated to the estimation of the initial state $\pi_0$ and another to the estimation of a one-step input. In the first one, the recurrent unit $\mathfrak{g}_1$ gathers temporal information from the reference into a single vector $q_K$, which is then decoded to the initial state $\pi_0$ through $\mathfrak{f}_1$. In the second component, the control signal $\psi_k$ is inferred from the current state of the agent $\pi_k$, the target reference observation $r_{k+1}$ and a latent memory vector $z_k$. The two parts combine as in the yellow box in Figure 1. The first component estimates $\pi_0$ given the current reference. Then, the second component is invoked recursively to estimate a sequence of inputs $\{\psi_k\}_{k \in [0,K]}$. For each loop, successive states $\pi_k$ are computed via the system dynamics $\varphi$ with the estimated input $\psi_k$. As such, the first component is used only once in the interval $[0, K]$. Note that the reference signal may change during the interval $[0, K]$. In that case, a new estimate of $\pi_0$ is obtained by running through the state reference generator again. The model is trained to minimize the following objective:

$$\min \sum_{k=0}^{K} |r_k - h(\pi_k)|^2 \qquad \text{s.t.} \quad \dot{\pi} = \varphi(\pi, \psi). \tag{13}$$

*Remark 3.* The state reference generator is trained in an unsupervised manner, in the sense that no ground truth states and controls are needed for training. The references can be chosen arbitrarily insofar as these remain admissible by the dynamics. Nevertheless, the training requires prior knowledge of a model of the system $(f, g, h)$. Since in many practical cases a good model may not be available, we robustify the state reference generator by training it with uniform noise on the dynamics. This exploits the generalization capabilities of DNNs by enriching the training set. Then, even if the model is faulty, the DNNs have

[2] Our code can be found at: `https://github.com/SteevenJanny/OutputTracking_contraction.git`

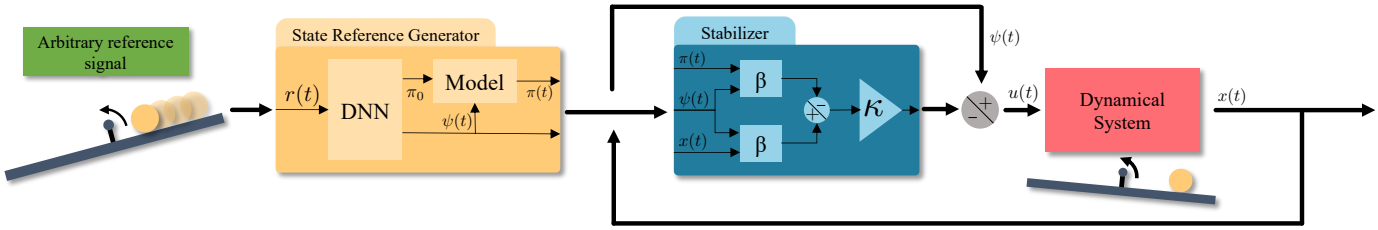



Fig. 1. We address the output tracking problem with a twofold approach: the *state reference generator* approximates the solution of the regulator equations and computes states and commands given an arbitrary reference signal. The *stabilizer* leverages a learned contractive function to force the dynamical system to track the reference.

more chances of producing suitable trajectories. Note that the bounds on the noise can be directly related to the confidence in the model.

**Stabilizer** – relies on the function $\hat{\beta}(\cdot, t)$ modeled as an MLP. To include time dependence, the network inputs are both the system state and the steady-state input $\psi(t)$, namely $\hat{\beta}(x,t) = \mathfrak{f}_3(x, \psi(t))$. Inspired by (Giaccagli et al., 2022b), this module is trained by following a two-step procedure. First, we look for a symmetric metric $P \succ 0$ satisfying the synchronisation constraint (5) and the Killing vector property (7). This metric is also modeled as an MLP $P(x,t) \approx \mathfrak{f}_4(x, \psi(t))$ with parameters $\theta_4$. We enforce symmetry by solely learning the upper triangular part of $P(x,t)$. Synchronization, Killing vector and positive definiteness hard constraints are relaxed as training objectives by minimizing the following loss:

$$J_{P,1}(x, \psi, \boldsymbol{p}) = \sum_{i=1}^{4} w_i J_i(x, \psi, \boldsymbol{p}),$$

where $\boldsymbol{p} = (\rho, \varepsilon, \epsilon, \underline{p})$ is a set of learned parameters controlling constraints in equations (14), $w_{P,1} := (w_1, \dots, w_4)$ is a vector of (positive) scalar weights and

$$J_i(x, \psi, \boldsymbol{p}) = \ln\left(\max\left(\Re\left\{\lambda_M(M_i)\right\}, 0\right) + 1\right),$$

with $i = 1, \dots, 4$, $\lambda_M$ being the maximum eigenvalue, $\Re\{\lambda\}$ the real part of $\lambda$ and $M_i$ defined as

$$\begin{aligned} M_1 &= L_f \mathfrak{f}_4(x,\psi) - \rho \mathfrak{f}_4(x,\psi) g(x) g^\top(x) \mathfrak{f}_4(x,\psi) + \varepsilon I, \\ M_2 &= L_g \mathfrak{f}_4(x,\psi) - \epsilon I, \\ M_3 &= -L_g \mathfrak{f}_4(x,\psi) - \epsilon I, \\ M_4 &= -\mathfrak{f}_4(x,\psi) + \underline{p} I. \end{aligned} \tag{14}$$

For a discussion about this objective, see (Giaccagli et al., 2022b). An important property of this loss is that terms vanish when the respective constraints are satisfied.

The parameter vector $\boldsymbol{p}$ controls the synchronization constraint (through $(\rho, \varepsilon)$), the Killing vector approximation ($\epsilon$) and the positive definiteness margin ($\underline{p}$). We propose a modified objective for training the metric function $\mathfrak{f}_4$ and estimating the vector of positive scalar parameters $\boldsymbol{p}$. This objective is modelled as a switching loss function, composed by two interacting elements

$$J_P(x, \psi, \boldsymbol{p}) = J_{P,1}(x, \psi, \boldsymbol{p}) + \sigma J_{P,2}(\boldsymbol{p}), \tag{15}$$

with switch variable $\sigma = 0$ if $J_{P,1}(x, \psi, \boldsymbol{p}) > 0$ and $\sigma = 1$ otherwise. The second component activates once a suitable metric is found (i.e., once $J_{P,1}$=0). Its aim is to improve the estimation of $\boldsymbol{p}$, while looking for a better metric. Formally, it is defined as

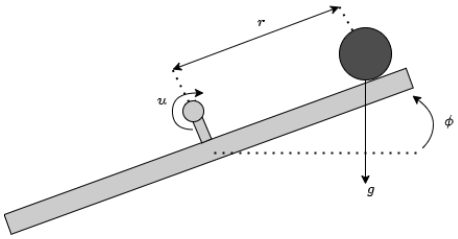



Fig. 2. Block-scheme of the ball and beam system

$$\begin{aligned} J_{P,2}(\boldsymbol{p}) = w_5 \ln(\epsilon^2 + 1) + w_6 \ln(\rho^2 + 1) \\ - w_7 \ln(\varepsilon^2 + 1) - w_8 \ln(\underline{p}^2 + 1), \end{aligned} \tag{16}$$

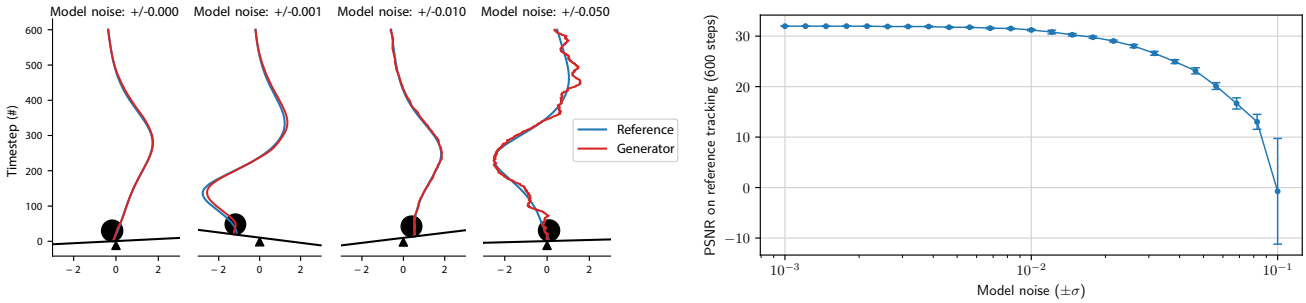
being $w_{P,2}$:=$(w_5, \dots, w_8)$ a vector of (positive) scalar weights. The sub-objective (16) aims at minimizing $\epsilon$ and $\rho$, i.e., at getting close to perfect Killing conditions and at reducing the controller dependence of the Riccati-like inequality (5). At the same time, it aims at maximizing $\varepsilon$ and $\underline{p}$, i.e., at increasing the contraction rate and the positivity of $P$. The composite objective (15) switches between metric search and contraction parameters optimization. First, it looks for a suitable metric along with set of parameters $\boldsymbol{p}$. Then, it freezes the metric estimator parameters $\theta_4$ and tries improving the contraction parameters $\boldsymbol{p}$. If the metric still satisfies $J_{P,1}(x, \psi, \boldsymbol{p})$=0, another step is taken in the direction of $\boldsymbol{p}$ improvement. If not, it unfreezes $\theta_4$ and the loop starts again. Note that, by using $J_{P,1}$ as a discriminant, we can set the trained network to be the last one verifying the contraction condition $J_{P,1}(x, \psi, \boldsymbol{p})$=0.

There are multiple advantages in using the switching objective (15). First, it achieves better estimation of parameters $\boldsymbol{p}$. Second, it improves controller robustness, e.g., smaller $\varepsilon$ implies faster contraction, that is, better stability margins (Sontag, 2010). Third, it weakens the dependence of $\boldsymbol{p}$ from the initial condition. As a matter of fact, $\boldsymbol{p}$ can be initialized to looser bounds, which ease training. The second objective will then try to tighten the conditions (14) progressively. Finally, it can escape from local minima as the shape of the loss function drastically changes on switches.

Once a suitable $P$ metric is found, $\hat{\beta}$ can be learned. That is, once $\mathfrak{f}_4$ has been trained, $\mathfrak{f}_3$ is optimized relatively to the following cost:

$$J_\beta(x, \psi) = \left| \frac{\partial \mathfrak{f}_3}{\partial x}(x, \psi) - g(x)^\top \mathfrak{f}_4(x, \psi) \right|^2.$$

Each model is trained with Adam optimizer until convergence on a training set composed of states and commands $(x, \psi)$ from pre-trained state reference generator. Inter-




(a) (b)

Fig. 3. **State reference generator** (a) We show four estimations from the state reference generator in different regimes where uniform noise is added to the model. (b) We measure peak signal to noise ratio (PSNR, dB) between the reference and the output for different noise range. We show that our approach is robust up to a sensible amount of noise.

mediate derivatives in (14) are obtained via automatic differentiation.

*Remark 4.* The state reference generator and the stabilizer can be trained separately, as long as the training samples $\psi_i$ for the stabilizer come from a similar distribution to the one of the output $\psi(t)$ of the state reference generator. Training $\mathfrak{f}_3, \mathfrak{f}_4$ on the outputs of the state reference generator is a way to ensure this.

## 4. SIMULATIONS

We test our solution on the well-known ball and beam setup, represented in Fig. 2 . The plant can be described by a system of the form (1) (Hauser et al., 1992) where $x \in \mathbb{R}^4$, $u \in \mathbb{R}$ and

$$f(x)=\begin{pmatrix} x_2 \\ B(x_1 x_4^2 - g_a \sin(x_3)) \\ x_4 \\ 0 \end{pmatrix}, \; g(x)=\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \; h(x)=x_1 \, .$$

From a physics viewpoint, $x=(r, \dot{r}, \phi, \dot{\phi})^\top$ with $r, \phi$ the ball position and beam angle respectively, $B$ a constant depending on system parameters and $g_a$ the gravitational acceleration. The interest of this setup lies in the fact that the relative degree [3] is not well-defined when the beam angular velocity and the ball position are zero. Therefore, input-output linearization and normal forms-based approaches fail to give a suitable controller. To make the problem harder, we sample the reference signal using the trajectory of the first component $z_1$ of a Lorenz oscillator whose dynamics is described by

$$\begin{cases} \dot{z}_1 = 10(z_2 - z_1) \\ \dot{z}_2 = z_1(28 - z_3) - z_2 \\ \dot{z}_3 = z_1 z_2 - \frac{8}{3} z_3 \, , \end{cases} \tag{17}$$

with random initial conditions. As (17) is a chaotic oscillator, it is exponentially sensitive to initial conditions, making it hard to find analytical solutions to the regulator equations. Then, approaches as in (Pavlov et al., 2006) become unfeasible in practice. Here, $\mathfrak{f}_i$ are four layers, 64 hidden units and tanh-activated MLPs. $\mathfrak{f}_1$ and $\mathfrak{f}_2$ use layer normalization on intermediate layers.

**State reference generator objective** – is to estimate a trajectory in the state space as well as the commands allowing to follow it from a reference signal, which can potentially concern solely a part of the state. The sought solution is thus not necessarily unique. Nevertheless, our approach can exploit the long-range information of a reference sequence to estimate an adequate initial condition. We show examples of predictions from the state reference generator in Figure 3a. In a second time, we predict an input leading the system to follow the reference. This is a challenging task, especially since it is learned without direct supervision, that is, without knowledge of the optimal command for data in the training set. Our approach performs well, even in a very noisy environment. Figure 3b shows the evolution of the error when the dynamical model of the state reference generator is disturbed by a uniform noise whose amplitude is varied. The results are obtained on a set of new references absent from the training dataset. The standard deviation $\sigma$ (represented by vertical bars) is obtained by averaging the results over five iterations. We find that the state reference generator is consistently robust to model errors up to a significant intensity.

**Behavior of the stabilizer** – is illustrated in Figure 4. For a given new reference, the state reference generator estimates the solutions of the regulator equations $(\pi, \psi)$. We then simulate the stabilizer starting from six random initial conditions $x_0$. Experimentally, we find that the system quickly converges towards the trajectory of the state reference generator. This is in accordance with the previous theoretical results. At timestep $k$=300, we drastically change the reference signal. The generator reacts rapidly to such a change and estimates a new $(\pi(t), \psi(t))$. Thanks to the stabilizer, the system converges quickly to the new reference. Once finished, the trajectory remains close to the state reference without deviating from it. As mentioned above, the analytical solution to the output tracking problem is difficult to obtain with such a nonlinear system under chaotic references. Our approach, although it is an approximation of the analytical solution, experimentally demonstrates very satisfactory performances. We report quantitative results in Table 6, in particular in the context of a system perturbed by Gaussian noise modeling measurement errors. We observe that the learned stabilizer $\hat{\beta}$ is robust even in noisy scenario. We also evaluate the advantages given by our switching objective, which reaches more stringent parameters $\boldsymbol{p}$. Table 6 compares noise robustness of our approach to the one without the fine-tuning component. We observe experimentally that our improved loss function leads to more robust control

[3] see (Isidori, 1995, Chapter IV)

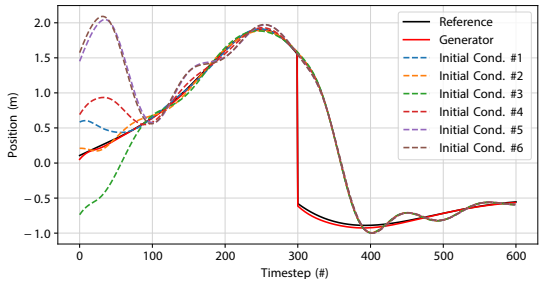



Fig. 4. **Stabilizer** rapidly converges to the state reference even when the initial state is far from the generator at timestep $k = 0$. At timestep $k$=300 timesteps, we abruptly change the reference signal. The state reference generator and the stabilizer react accordingly.

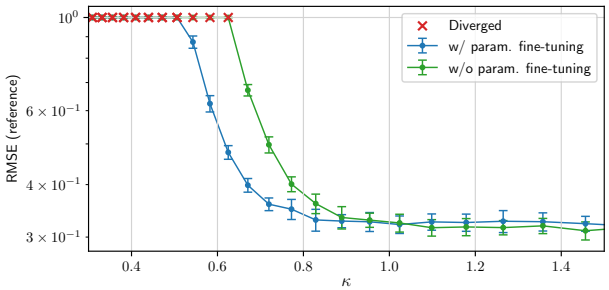



Fig. 5. **Control Gain** $\kappa$: Our fine-tuning step of the $P$ metric increases the range of values allowed for the stabilizer to track the reference. Red cross indicate that the system diverged from the reference signals.

| Parameter Finetuning | Noise StDev | | | |
|---|---|---|---|---|
| | 0 | 0.01 | 0.05 | 0.1 |
| With | **0.343** ± 0.015 | **0.323** ± 0.015 | **0.361** ± 0.011 | **0.385** ± 0.014 |
| Without | 0.385 ± 0.015 | 0.366 ± 0.016 | 0.408 ± 0.012 | 0.439 ± 0.016 |

Fig. 6. **Noise robustness** is improved when using $\hat{\beta}$ trained with the parameter fine-tuning step. We measure RMSE from reference for different gaussian noise StDev. on state measurements. Our model can still performs correctly with uncertain observations.

laws. Moreover, we also observe that fine-tuning allows for lower control gain $\kappa$, (Figure 5). This is linked to the size of the domain of attraction when the Killing vector property holds only approximately (Giaccagli et al., 2022b).

## 5. CONCLUSIONS

In this paper, we provided a deep-learning based solution to the output tracking problem for input-affine nonlinear systems. Through results coming from regulation theory, we relied on a state-feedback controller composed of two terms. The first one is the steady-state control action. The second term guarantees convergence to the reference trajectory on which the tracking task is achieved. This has been addressed through contraction tools, specialized to a multi-agent synchronization setting. To face the difficulties of providing the analytical form of our solution, we relied on DNNs to approximate the controller, and we linked the approximation precision to the tracking error. We split the learning task in two separate modules. We leveraged long-term memory and multilayer perceptron networks to approximately solve the regulator equations. Classical multilayer perceptrons were also used to approximate the synchronizing component. We show that the approach does not hinder tracking guarantees for sufficiently precise networks. We train our models in a weakly supervised setup, that does not require ground truth solutions to the regulator equations nor very accurate dynamical model. To validate the proposed result, we test our controller on the ball and beam example. Future studies will focus on the generalization of the proposed design to non-input affine systems and by taking into consideration possible saturations on the input.

## REFERENCES


Bertsekas, D. (2019). *Reinforcement learning and optimal control*. Athena Scientific.

Byrnes, C. and Isidori, A. (2003). Limit sets, zero dynamics, and internal models in the problem of nonlinear output regulation. *IEEE Trans. on Automatic Control*.

Chen, C., Xie, L., Xie, K., Lewis, F., and Xie, S. (2022). Adaptive optimal output tracking of continuous-time systems via output-feedback-based reinforcement learning. *Automatica*.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.

Dawson, C., Gao, S., and Fan, C. (2023). Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*.

Devasia, S., Chen, D., and Paden, B. (1996). Nonlinear inversion-based output tracking. *IEEE Trans. on Automatic Control*.

Francis, B. and Wonham, W. (1976). The internal model principle of control theory. *Automatica*.

Giaccagli, M., Astolfi, D., Andrieu, V., and Marconi, L. (2022a). Sufficient conditions for global integral action via incremental forwarding for input-affine nonlinear systems. *IEEE Transactions on Automatic Control*, 67(12), 6537–6551.

Giaccagli, M., Zoboli, S., Astolfi, D., Andrieu, V., and Casadei, G. (2022b). Synchronization in networks of nonlinear systems: Contraction metric analysis and deep-learning for feedback estimation. *Submitted to IEEE Trans. on Automatic Control*.

Hauser, J., Sastry, S., and Kokotovic, P. (1992). Nonlinear control via approximate input-output linearization: The ball and beam example. *IEEE Trans. on Automatic Control*.

Isidori, A. (1995). *Nonlinear Control Systems*. Springer.

Isidori, A. and Byrnes, C.I. (1990). Output regulation of nonlinear systems. *IEEE Trans. on Automatic Control*.

Janny, S., Andrieu, V., Nadri, M., and Wolf, C. (2021). Deep KKL: Data-driven output prediction for non-linear systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 4376–4381. IEEE.

Limon, D. and Alamo, T. (2021). Tracking model predictive control. In *Encyclopedia of Systems and Control*. Springer.

Lohmiller, W. and Slotine, J.E. (1998). On contraction analysis for non-linear systems. *Automatica*.
Manchester, I. and Slotine, J. (2017). Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Trans. on Automatic Control*.
Martin, P., Devasia, S., and Paden, B. (1996). A different look at output tracking: Control of a VTOL aircraft. *Automatica*.
Pavlov, A., Van De Wouw, N., and Nijmeijer, H. (2006). *Uniform output regulation of nonlinear systems: a convergent dynamics approach*. Springer.
Serrani, A., Isidori, A., and Marconi, L. (2001). Semi-global nonlinear output regulation with adaptive internal model. *IEEE Trans. on Automatic Control*.
Sontag, E.D. (2010). Contractive systems with inputs. In *Perspectives in mathematical system theory, control, and signal processing*. Springer.
Sontag, E. and Wang, Y. (1995). On characterizations of the input-to-state stability property.
Sun, D., Jha, S., and Fan, C. (2021). Learning certified control using contraction metric. In *Conference on Robot Learning*.
Tsukamoto, H. and Chung, S. (2020). Neural contraction metrics for robust estimation and control: A convex optimization approach. *IEEE Control Systems Letters*.
Tsukamoto, H., Chung, S., and Slotine, J. (2021). Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview. *Annual Reviews in Control*.
Wei, L., McCloy, R., and Bao, J. (2022). Discrete-time contraction-based control of nonlinear systems with parametric uncertainties using neural networks. *Computers & Chemical Engineering*.
Wondergem, M., Lefeber, E., Pettersen, K., and Nijmeijer, H. (2010). Output feedback tracking of ships. *IEEE Trans. on Control Systems Technology*.
Wu, Y. and Zou, Q. (2009). Robust inversion-based 2-DOF control design for output tracking: Piezoelectric-actuator example. *IEEE Trans. on Control Systems Technology*.
Xie, X. and Duan, N. (2010). Output tracking of high-order stochastic nonlinear systems with application to benchmark mechanical system. *IEEE Trans. on Automatic Control*.
Zhao, P., Lakshmanan, A., Ackerman, K., Gahlawat, A., Pavone, M., and Hovakimyan, N. (2022). Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics. *IEEE Robotics and Automation Letters*.

## Appendix A.

### *A.1 Proof of Proposition 1*

The proof follows the line of results in Giaccagli et al. (2022b) and combines them with ISS-like arguments. For space reasons, we only highlight the main parts. Define the state-error $\tilde{x} := x - \pi$. Its dynamics read as

$$\dot{\tilde{x}} = \varphi(\pi+\tilde{x},t) - \varphi(\pi,t) - \kappa g(\pi+\tilde{x}) \times (\beta(\pi+\tilde{x},t) - \beta(\pi,t)) + g(\pi+\tilde{x})\omega(t)\,. \quad \text{(A.1)}$$

Let $\tilde{\mathcal{X}}(\tilde{x}_0,t,t_0)$ be a solution defined for all $t \geq t_0$ and consider the function $\Gamma : [0,1]\times\mathbb{R}\times\mathbb{R}\to\mathbb{R}^{n_x}$ satisfying $\Gamma(1,t_0,t_0) = \tilde{\mathcal{X}}(\tilde{x}_0,t_0,t_0)$, $\Gamma(0,t_0,t_0) = 0$ and $\Gamma(s,t_0,t_0) = \gamma(s)$, where $\gamma:[0,1]\to\mathbb{R}^{n_x}$ is any $C^1$ curve and solution to

$$\frac{\partial\Gamma}{\partial t}(s,t,t_0) = \varphi(\Gamma+\Pi,t) - \varphi(\Pi,t) - \kappa g(\Gamma+\Pi)(\beta(\Gamma+\Pi,t) - \beta(\Pi,t)) + g(\Gamma+\Pi)w(s)$$

with $\Pi = \Pi(\pi_0,t,t_0)$ being the trajectory of (2) (arguments are dropped for space reasons) and $w(s) = s\omega$. Take the candidate Lyapunov function

$$V(t) = \int_0^1 \frac{\partial\Gamma^\top}{\partial s}(s,t,t_0)P(\Gamma+\Pi,t)\frac{\partial\Gamma}{\partial s}(s,t,t_0)\,ds \quad \text{(A.2)}$$

with $P$ solving (5). Taking its time-derivative and through the Killing vector assumption and the integrability condition (6), we get

$$\dot{V}(t) \leq \int_0^1 \frac{\partial\Gamma^\top}{\partial s}(s,t,t_0)[T_1(s,t_0,t) + T_2(s,t_0,t)]\frac{\partial\Gamma}{\partial s}(s,t,t_0) + T_3(s,t_0,t)\,ds\,,$$

with

$$\begin{aligned}
T_1(s,t_0,t) &= L_\varphi P(\Gamma+\Pi,t)\\
T_2(s,t_0,t) &= -2\kappa P(\Gamma+\Pi,t)g(\Gamma+\Pi)g^\top(\Gamma+\Pi)P(\Gamma+\Pi,t)\\
T_3(s,t_0,t) &= \frac{\partial\Gamma^\top}{\partial s}(s,t,t_0)P(\Gamma+\Pi,t)g(\Gamma+\Pi,t)\omega(t) + \omega^\top(t)g^\top(\Gamma+\Pi,t)P(\Gamma+\Pi,t)\frac{\partial\Gamma^\top}{\partial s}(s,t,t_0)\,.
\end{aligned}$$

From the (generalized) inequality of Young [4] with $a = \frac{\partial\Gamma^\top}{\partial s}(s,t,t_0)\sqrt{P(\Gamma+\Pi,t)}$, $b = \sqrt{P(\Gamma+\Pi,t)}g(\Gamma+\Pi)\omega(t)$ and $c = \frac{\lambda}{2}$ it follows that

$$T_3(s,t,t_0) \leq \frac{\lambda}{2}\frac{\partial\Gamma^\top}{\partial s}(s,t,t_0)P(\Gamma+\Pi,t)\frac{\partial\Gamma}{\partial s}(s,t,t_0) + \frac{2}{\lambda}\omega^\top(t)g^\top(\Gamma+\Pi)P(\Gamma+\Pi,t)g(\Gamma+\Pi)\omega(t)$$

Taking $\kappa \geq \frac{\rho}{2}$ and employing (5) we get

$$\dot{V}(t) \leq -\tfrac{\lambda}{2}V(t) + \tfrac{2}{\lambda}\overline{p}\,\overline{g}^2|\omega(t)|^2\,.$$

From (5) and since $\tilde{\mathcal{X}}(\tilde{x},t,t) = \tilde{x}(t)\,\forall\,t$, it follows that, for any $t \geq t_0$

$$\underline{p}\,|\tilde{x}(t)|^2 \leq V(t) \leq \overline{p}\,|\tilde{x}(t)|^2\,. \quad \text{(A.3)}$$

Hence, the proof concludes by Gronwall lemma and by following standard ISS-like arguments Sontag and Wang (1995).

### *A.2 Proof of Proposition 2*

By adding and subtracting $\psi(t), \beta(x,t), \beta(\pi(t),t)$ and $\beta(\hat{\pi}(t),t)$, we rewrite the control (10) as $u(t) = u^\star(t) + \tilde{u}(t)$ with $u^\star := \psi(t) - \kappa(\beta(x,t) - \beta(\pi(t),t))$ and $\tilde{u}$ defined as

$$\tilde{u}(t) := \hat{\psi}(t) - \psi(t) - \kappa[(\hat{\beta}(x,t) - \beta(x,t)) + (\beta(\hat{\pi}(t),t) - \hat{\beta}(\hat{\pi}(t),t)) + (\beta(\pi(t),t) - \beta(\hat{\pi}(t),t))\,. \quad \text{(A.4)}$$

Consider the Lyapunov function (A.2) with $\tilde{x} = x - \pi$. Following the same lines as in the proof of Proposition 1, it follows that

$$\dot{V}(t) \leq -\tfrac{\lambda}{2}V(t) + \tfrac{2}{\lambda}\overline{p}\,\overline{g}^2|\tilde{u}(t)|^2\,.$$

[4] Generalized inequality of Young: $2ab \leq ca^2 + \frac{b^2}{c}$ for any $c > 0$

Consider now the reference $r$. Since $r(t) \in \mathcal{R}$ for all $t \geq t_0$, there exist a compact set $\mathcal{W}_\pi$ such that $\pi(t) \in \mathcal{W}_\pi$ for all $t \geq t_0$. Now, define

$$\eta_1 := \sup_{x \in \mathcal{W}_\pi} |x| \,, \; \eta_2 := \sup_{x \in \mathcal{W}_{\tilde{x}}} |x| \,, \; \eta_3 := \max\{\eta_2, \delta\} \,.$$

Then, $\mathcal{W}_{\tilde{x}} \subseteq \mathcal{B}_{\eta_3}$. Define $\overline{\mathcal{V}} := \left\{\tilde{x}_0 \in \mathbb{R}^n : V(t_0) \leq \bar{p}\eta_3^2\right\}$, and note that $\tilde{x}_0 \in \mathcal{B}_{\eta_3}$ implies $\tilde{x}_0 \in \overline{\mathcal{V}}$ due to (A.3). Differently put, $\mathcal{B}_{\eta_3} \subseteq \overline{\mathcal{V}}$. Pick $\mathcal{W}_x = \mathcal{B}_{\eta_4}$, where

$$\eta_4 = \eta_1 + \sup_{x \in \overline{\mathcal{V}}} |x| \,.$$

Then, $\overline{\mathcal{V}} \subseteq \mathcal{W}_x$. Moreover, if $\tilde{x}_0 \in \overline{\mathcal{V}}$, then $x_0 \in \mathcal{W}_x$. Pick

$$\mu_\delta = \frac{\lambda \delta \underline{p}}{2\sqrt{2}\,\overline{p}\,\overline{g}(1 + 2\kappa + \kappa \overline{p}\,\overline{g})} \,.$$

From (11), (A.4) and the relation (6), it follows that, for all times $t \geq t_0$ such that $x(t) \in \mathcal{W}_x$

$$\begin{aligned} |\tilde{u}(t)| &\leq \mu_\delta + 2\kappa\mu_\delta + \kappa \overline{p}\,\overline{g}\,|\pi(t) - \hat{\pi}(t)| \\ &\leq \mu_\delta(1 + 2\kappa + \kappa \overline{p}\,\overline{g}) \leq \tfrac{\lambda \delta \underline{p}}{2\sqrt{2}\,\overline{p}\,\overline{g}} \,. \end{aligned}$$

Consider now the set $\underline{\mathcal{V}} := \{\tilde{x}_0 \in \mathbb{R}^{n_x} : V(t_0) < \underline{p}\delta^2\}$ and suppose $\tilde{x}_0 \in \underline{\mathcal{V}}$. By (A.3), if $\underline{p}\,|\tilde{x}_0|^2 < \underline{p}\delta^2$ then $|x_0 - \pi_0| < \delta$. Now, note that $\underline{\mathcal{V}} \subsetneq \mathcal{B}_{\eta_3} \subseteq \overline{\mathcal{V}}$ and suppose $\tilde{x}_0 \in \overline{\mathcal{V}} \backslash \underline{\mathcal{V}}$. This implies $|\tilde{x}_0|^2 \geq \frac{\underline{p}}{\overline{p}}\delta^2$. However, since $\tilde{x}_0 \in \overline{\mathcal{V}}$ implies $x_0 \in \mathcal{W}_x$, we have

$$\begin{aligned} \dot{V}(t_0) &\leq -\lambda V(t_0) + \tfrac{2}{\lambda}\overline{p}\,\overline{g}^2 |\tilde{u}(t_0)|^2 \\ &\leq \tfrac{\lambda}{2}\left(\tfrac{\underline{p}^2}{2\overline{p}}\delta^2 - V(t_0)\right) \leq \underline{p}\tfrac{\lambda}{2}\left(\tfrac{1}{2}\tfrac{\underline{p}}{\overline{p}}\delta^2 - |\tilde{x}_0|^2\right) < 0 \,. \end{aligned}$$

Hence, the level set $\overline{\mathcal{V}}$ is forward invariant. Moreover, in view of the above results, the set $\underline{\mathcal{V}}$ is attractive and forward invariant, with domain of attraction including $\overline{\mathcal{V}}$. Recall that $\mathcal{W}_{\tilde{x}} \subseteq \mathcal{B}_{\eta_3} \subseteq \overline{\mathcal{V}}$. Hence, for all $\tilde{x}_0 \in \mathcal{W}_{\tilde{x}}$, it holds $\lim_{t \to +\infty} |\mathcal{X}(x_0, t, t_0) - \Pi(\pi_0, t, t_0)| \leq \delta$.