Certification of Learning-Based Feedback Linearization Under Measurement Perturbations

Marco Ledda, *Student, IEEE*, Diego Deplano, *Member, IEEE*,
Alessandro Giua, *Fellow, IEEE*, Mauro Franceschelli, *Senior, IEEE*

***Abstract*— In this paper, we study the finite-horizon discrepancy between an ideal feedback-linearized closed loop and a closed loop driven by an imperfect inversion law when the measured output is affected by bounded perturbations. We formulate the problem of determining the perturbation sequence that maximizes the mismatch between the actual plant output and the ideal linearized output over a prescribed horizon. We then address this problem for a learning-based feedback-linearization architecture in which the nominal inversion is computed from an approximate model and corrected by an additive feedforward neural network. Under a sampled-data implementation, the comparison is cast in discrete time as a finite-dimensional constrained optimization problem that explicitly captures the plant dynamics, the ideal linearized dynamics, the outer linear controller, the perturbation budget, and the neural network structure. Since the resulting formulation is generally nonconvex, the computed solution provides a finite-horizon lower bound on the worst-case discrepancy. Numerical results on a nonlinear system show that the proposed framework identifies perturbations that induce substantially larger deviations than non-optimized Gaussian disturbances.**

I. Introduction

Feedback linearization is a classical nonlinear control paradigm that transforms nonlinear input–output dynamics into an equivalent linear behavior through a suitable change of coordinates and nonlinear state feedback [1, 2]. By cancelling the plant nonlinearities, it reduces control design to the regulation or tracking of a linear reference model and has therefore been widely studied in both theory and engineering applications. However, exact feedback linearization is intrinsically model-dependent, as it requires accurate knowledge of the plant vector fields and decoupling structure. Modeling errors, unmodeled dynamics, parametric uncertainty, and sensing or actuation imperfections may compromise the cancellation mechanism. Sampled-data implementations introduce further difficulties, since continuous-time exact linearization properties are not automatically preserved after discretization [3–5]. These limitations have motivated approximate, adaptive, and robust variants of feedback linearization. To reduce reliance on exact analytical models, several works have investigated data-driven approximations of the linearizing law. Early contributions considered adaptive control and neural network approximations for uncertain linearizable systems [6–8], while more recent approaches rely on deep learning [9, 10], Gaussian processes [11], reinforcement learning [12], and data-driven formulations based on complete dictionaries or Koopman constructions [13, 14]. These methods provide promising alternatives when the exact inversion map is unavailable or only partially known.

Most existing works focus on controller synthesis and nominal closed-loop performance, while providing limited guarantees under perturbations. This is particularly relevant in safety-critical applications, where learned controllers may amplify measurement errors. Although noise-aware learning, uncertainty quantification, and robustness under noisy measurements have been investigated [11, 15–18], existing results do not characterize the worst-case finite-horizon mismatch between an ideal feedback-linearized loop and its learning-based approximation. Recent certification methods for learning-enabled dynamical and control systems further motivate tools that complement nominal performance with reliability guarantees [19, 20].

Main contribution – The main contribution of this paper is to address this challenge by proposing a method to certify the discrepancy between an ideal feedback-linearized loop and a learning-based approximation under bounded measurement perturbations. We propose an optimization-based certification framework to quantify how far a learned feedback-linearizing loop may depart from the ideal one under bounded measurement perturbations. Since the resulting problem is generally nonconvex, the computed objective value should be interpreted, in general, as a lower bound on the true worst-case discrepancy.

Structure of the paper – The remainder of the paper is organized as follows. Section II presents the notation used in the remainder of the paper. Section III introduces the problem statement. Section IV shows the mathematical formulation of the certification method, specifying the assumptions and constraints involved. Section V presents the numerical results. Section VI draws the conclusion and outlines future research directions.

The work of Diego Deplano was supported by the project e.INS-Ecosystem of Innovation for Next Generation Sardinia (cod. ECS 00000038) funded by the Italian Ministry for Research and Education (MUR) under the National Recovery and Resilience Plan (NRRP) - MISSION 4 COMPONENT 2, “From research to business” INVESTMENT 1.5, “Creation and strengthening of Ecosystems of innovation” and construction of “Territorial R& D Leaders”.

This work is partially supported by Fondazione di Sardegna with project “Artificial Intelligence for Smart Technologies (AISTech)”, call 2023, F23C25000310007 and by Fondazione di Sardegna with project ”Formal methods for Resiliency and Security in Networks of Systems (ReSNetS)” call 2025, F83C26000350007.

All authors are with the DIEE, University of Cagliari, 09123 Cagliari, Italy. Emails: marco.ledda2@unica.it, diego.deplano@unica.it, giua@unica.it, mauro.franceschelli@unica.it. Mauro Franceschelli is the corresponding author.

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

II. Notation and preliminaries

The sets of natural, integer, and real numbers are denoted by $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$, respectively. Matrices $M \in \mathbb{R}^{n\times n}$ are denoted by uppercase letters, vectors $\boldsymbol{v} \in \mathbb{R}^n$ by boldface letters, and scalars $s \in \mathbb{R}$ by lowercase letters, while sets and spaces are denoted by calligraphic letters such as $\mathcal{S}$. The vectors of zeros and ones of dimension n are denoted by $\mathbb{0}$ and $\mathbb{1}$, respectively. The Euclidean norm is denoted by $\|\cdot\|_2$. The symbol $\circ$ denotes function composition: for two functions f and g, the composition $f \circ g$ is defined as $(f \circ g)(x) = f(g(x))$. The symbol $\odot$ denotes the Hadamard product. The symbol † denotes the Moore–Penrose pseudoinverse of a matrix. The map $\mathrm{ReLU}(\cdot)$ denotes the *Rectified Linear Unit* (ReLU) activation function and acts componentwise as

$$(\mathrm{ReLU}(\boldsymbol{v}))_i := \max\{0, v_i\}, \qquad i = 1, \dots, r, \tag{1}$$

for any vector $\boldsymbol{v} \in \mathbb{R}^r$.

III. Problem Statement

We consider nonlinear control-affine systems in continuous time $t \in \mathbb{R}$,

$$\Sigma := \begin{cases} \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t)) + g(\boldsymbol{x}(t))\boldsymbol{u}(t), \\ \boldsymbol{y}(t) = h(\boldsymbol{x}(t)), \end{cases} \tag{2}$$

where $\boldsymbol{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state, $\boldsymbol{u}(t) \in \mathbb{R}^q$ is the input, and $\boldsymbol{y}(t) \in \mathbb{R}^m$ is the output. Input-output feedback-linearization aims to replace the original input $\boldsymbol{u}(t)$ with a new input $\boldsymbol{v}(t)$ through a state-dependent change of variables, chosen so that the input-output dynamics become linear. Assume that, for each output $y_i(t)$, there exists a relative degree $\gamma_i \in \mathbb{N}$ such that the input $\boldsymbol{u}(t)$ appears explicitly in the γ_i-th derivative $y_i^{(\gamma_i)}(t)$, for all $i = 1, \dots, m$. Although not every control-affine system admits well-defined relative degrees or exact feedback linearization, this class provides the standard setting in which, whenever such relative degrees exist, the differentiated input-output relation depends affinely on the control input, namely,

$$\left[y_1^{(\gamma_1)}(t), \cdots, y_m^{(\gamma_m)}(t)\right]^\top = \boldsymbol{b}_\Sigma(\boldsymbol{x}(t)) + A_\Sigma(\boldsymbol{x}(t))\boldsymbol{u}(t).$$

where the vector $\boldsymbol{b}_\Sigma(\boldsymbol{x}(t))$ collects the nonlinear drift terms and the matrix $A_\Sigma(\boldsymbol{x}(t))$ collects the decoupling terms, which both depend on the specific dynamics of system Σ described by f, g, h [1]. Then, under the assumption that $A_\Sigma(\boldsymbol{x}(t))$ has full row rank, the input transformation $\boldsymbol{u}(t) = \boldsymbol{u}_{\text{LIN}}(\boldsymbol{x}(t), \boldsymbol{v}(t))$ given by

$$\boldsymbol{u}_{\text{LIN}}(\boldsymbol{x}(t), \boldsymbol{v}(t)) := A_\Sigma^\dagger(\boldsymbol{x}(t))\big(-\boldsymbol{b}_\Sigma(\boldsymbol{x}(t)) + \boldsymbol{v}(t)\big), \tag{3}$$

yields $y_i^{(\gamma_i)} = v_i(t)$ for all $i = 1, \cdots, m$. In turn, the ideal linearized system with state variable $\boldsymbol{\xi}(t) = [\cdots, \xi_{i,j}(t), \cdots]^\top$ such that $\xi_{i,1}(t) = y_i(t)$ has the following dynamics

$$\forall i = 1, \cdots, m : \begin{cases} \dot{\xi}_{i,1}(t) = \xi_{i,2}(t) \\ \dot{\xi}_{i,2}(t) = \xi_{i,3}(t) \\ \qquad \vdots \\ \dot{\xi}_{i,\gamma_i}(t) = v_i(t) \end{cases}$$

which in compact form becomes $\dot{\boldsymbol{\xi}}(t) = A\boldsymbol{\xi}(t) + B\boldsymbol{v}(t)$ and $\boldsymbol{y}(t) = C\boldsymbol{\xi}(t)$, where $\boldsymbol{\xi}(t)$, $\boldsymbol{v}(t)$, and $\boldsymbol{y}(t)$ are the state, the input, and the output of the linearized system, respectively, and A, B, C are the standard Brunovský-form matrices. By construction, $\boldsymbol{\xi}(t) \in \mathbb{R}^\gamma$ with $\gamma := \sum_{i=1}^m \gamma_i$.

In practice, the input transformation in eq. (3) is often not known exactly, for instance when the dynamics of the system in eq. (2) is unknown and, consequently, the output of the ideal linearized model does not coincide with that of the actual linearized model. Let us denote by $\boldsymbol{y}_{\text{LIN}}(t)$ and $\boldsymbol{v}_{\text{LIN}}(t)$ the output and the input under the exact feedback linearization, i.e.,

$$\Sigma_{\text{LIN}} := \begin{cases} \dot{\boldsymbol{\xi}}(t) = A\boldsymbol{\xi}(t) + B\boldsymbol{v}_{\text{LIN}}(t), \\ \boldsymbol{y}_{\text{LIN}}(t) = C\boldsymbol{\xi}(t), \end{cases} \tag{4}$$
$$\boldsymbol{\xi}(0) := [\cdots, \xi_{i,j}(0), \cdots]^\top = [\cdots, y_i^{(j-1)}(0), \cdots]^\top$$

In other words, the two outputs coincide $\boldsymbol{y}_{\text{LIN}}(t) \equiv \boldsymbol{y}(t)$ under the exact input-output linearizing law (3) and consistent initialization, otherwise it is not guaranteed.

With this notation, we formalize the problem of interest.

Problem 1. *Consider a nonlinear control-affine system as in eq.* (2)*, driven by the imperfect feedback-linearizing input $\widetilde{\boldsymbol{u}}(\boldsymbol{x}(t), \boldsymbol{v}(t))$, and the corresponding ideal linearized system as in eq.* (4)*, driven by $\boldsymbol{u}_{\text{LIN}}(\boldsymbol{x}(t), \boldsymbol{v}(t))$. According to Figures 1–2, given an output reference $\boldsymbol{r}(t)$, let both systems be driven by the same proportional controller with gain matrix K. Determine the worst-case disturbance signal $\boldsymbol{\delta}(t)$ over a time horizon $t \in [0, T]$ with $T \in \mathbb{N}$, acting on the output and maximizing the discrepancy between the*

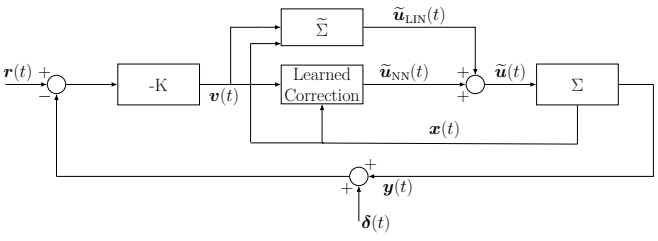


Fig. 1: Schematic representation of the closed-loop system linearized with an approximate model $\widetilde{\Sigma}$ corrected by a learning block.

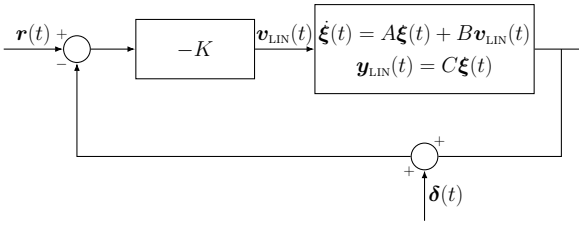


Fig. 2: Schematic representation of the closed-loop system perfectly linearized.

© 2026 IEEE. Personal use of this material is permitted.

actual output $\boldsymbol{y}(t)$ and the ideal output $\boldsymbol{y}_{\text{LIN}}(t)$, subject to the point-wise constraint $\|\boldsymbol{\delta}(t)\|_\infty \leq \varepsilon$ where $\varepsilon > 0$.

We address Problem 1 in the case in which the feedback-linearizing law is designed according to the method proposed by Westenbroek et al. in [12]. We assume that the system dynamics is not known exactly and, instead, the following approximated model is known

$$\widetilde{\Sigma} := \begin{cases} \dot{\widetilde{\boldsymbol{x}}}(t) = \widetilde{f}(\widetilde{\boldsymbol{x}}(t)) + \widetilde{g}(\widetilde{\boldsymbol{x}}(t))\boldsymbol{u}(t), \\ \widetilde{\boldsymbol{y}}(t) = \widetilde{h}(\widetilde{\boldsymbol{x}}(t)), \end{cases} \tag{5}$$

from which one can compute the ideal linearization feedback using the available information on the system dynamics $\widetilde{\Sigma}$ and the measured true state $\boldsymbol{x}(t)$ as follows

$$\widetilde{\boldsymbol{u}}_{\text{LIN}}(\boldsymbol{x}(t), \boldsymbol{v}(t)) := A^{\dagger}_{\widetilde{\Sigma}}(\boldsymbol{x}(t))\big(-\boldsymbol{b}_{\widetilde{\Sigma}}(\boldsymbol{x}(t)) + \boldsymbol{v}(t)\big).$$

To compensate the discrepancy between the dynamics of Σ and $\widetilde{\Sigma}$, an additive control input $\widetilde{\boldsymbol{u}}_{\text{NN}}(\boldsymbol{x}(t), \boldsymbol{v}(t))$ can be computed via a proper neural network, such that the overall input transformation is given by

$$\widetilde{\boldsymbol{u}}(\boldsymbol{x}(t), \boldsymbol{v}(t)) := \widetilde{\boldsymbol{u}}_{\text{LIN}}(\boldsymbol{x}(t), \boldsymbol{v}(t)) + \widetilde{\boldsymbol{u}}_{\text{NN}}(\boldsymbol{x}(t), \boldsymbol{v}(t)). \tag{6}$$

In this work, we consider fully-connected feedforward neural networks consisting of $L+1 \in \mathbb{N}$ layers, each with update rule $f_\ell : \mathbb{R}^{n_{\ell-1}} \to \mathbb{R}^{n_\ell}$ with $n_\ell \in \mathbb{N}$ denoting the number of neurons for $\ell \in \{1, \cdots, L+1\}$, such that $n_0 := n + m$ and $n_{L+1} := q$. Let $\boldsymbol{h}_\ell \in \mathbb{R}^{n_\ell}$ be the output vector of the ℓ-th layer for $\ell \in \{1, \cdots, L\}$, and $\boldsymbol{h}_0 := [\boldsymbol{x}(t)^\top \, \boldsymbol{v}(t)^\top]^\top$ be the input vector of the neural network. The output layer $\ell = L+1$ computes an affine combination of the output of the previous layer, while all hidden layers $\ell \neq L+1$ first compute an affine combination of the previous layer and then apply the *Rectified Linear Unit* (ReLU) activation function as in eq. (1), namely,

$$\begin{aligned} \boldsymbol{h}_{L+1} &:= f_{L+1}(\boldsymbol{h}_L) := W_{L+1}\boldsymbol{h}_L + \boldsymbol{b}_{L+1}, \\ \boldsymbol{h}_\ell &:= f_\ell(\boldsymbol{h}_{\ell-1}) := \text{ReLU}(W_\ell \boldsymbol{h}_{\ell-1} + \boldsymbol{b}_\ell), \ \forall \ell \in \{1, \cdots, L\}, \end{aligned}$$

where $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ and $\boldsymbol{b}_\ell \in \mathbb{R}^{n_\ell}$ are the weight matrix and bias vector. With this notation, the additive input transformation due to the neural network is given by

$$\widetilde{\boldsymbol{u}}_{\text{NN}}(\boldsymbol{x}(t), \boldsymbol{v}(t)) := f_{\text{NN}}(\boldsymbol{x}(t), \boldsymbol{v}(t)), \ \ f_{\text{NN}} := f_{L+1} \circ \cdots \circ f_1. \tag{7}$$

IV. Proposed method via discrete-time approximation

This section formulates the certification framework in discrete time by introducing the sampled models of the plant and ideal linearized system, the resulting finite-horizon optimization problem, the reformulation of the neural correction, and its computational complexity.

A. Discrete-time approximation

We consider a sampled-data implementation with sampling period $\Delta t > 0$ and sampling instants $t_k := k\Delta t$, $k \in \mathbb{N}$. Let $\boldsymbol{x}(\cdot)$ denote the state trajectory of the plant and define the sampled state as $\boldsymbol{x}_k := \boldsymbol{x}(t_k)$. Moreover, let $\boldsymbol{u}_k \in \mathbb{R}^q$ denote the control input applied on the interval $[t_k, t_{k+1})$ under a zero-order-hold policy, namely

$$\boldsymbol{u}(t) = \boldsymbol{u}_k, \qquad t \in [t_k, t_{k+1}).$$

Under this convention, the continuous-time plant in (2) induces the discrete-time dynamics

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} \big(f(\boldsymbol{x}(\tau)) + g(\boldsymbol{x}(\tau))\boldsymbol{u}_k\big)\, d\tau. \tag{8}$$

where the integral computes the sampled-data state increment associated with the sampling period Δt. In practice, the integral is approximated through standard numerical integration methods, such as Euler or Runge–Kutta schemes. This leads to a discrete-time map $f_d : \mathcal{X} \times \mathbb{R}^q \to \mathbb{R}^n$ representing the state increment, namely,

$$\boldsymbol{x}_{k+1} \approx \boldsymbol{x}_k + f_d(\boldsymbol{x}_k, \boldsymbol{u}_k).$$

Analogously, the ideal linearized model in (4) is discretized as

$$\Sigma_d := \begin{cases} \boldsymbol{\xi}_{k+1} = A_d \boldsymbol{\xi}_k + B_d \boldsymbol{v}_k, \\ \boldsymbol{y}_{\text{LIN},k} = C_d \boldsymbol{\xi}_k, \end{cases} \tag{9}$$

with

$$A_d := e^{A\Delta t}, \quad B_d := \int_0^{\Delta t} e^{A\tau} B d\tau, \quad C_d := C. \tag{10}$$

where A_d, B_d, and C_d are the discrete-time matrices obtained from the continuous-time chains of integrators.

With these definitions in place, the original continuous-time certification problem in 1 can be reformulated in discrete time as a finite-horizon optimization problem over the perturbation sequence $\{\boldsymbol{\delta}(t)\}_{t=0}^{N}$ with $N = T/\Delta t$, where T is the length of the time horizon, whose goal is to determine a perturbation sequence that maximizes the cumulative discrepancy between the ideal and the real output.

B. Bilinear formulation of the neural network dynamics

We first derive an exact representation of the neural correction $\widetilde{\boldsymbol{u}}_{\text{NN},k}$. Since the nominal term $\widetilde{\boldsymbol{u}}_{\text{LIN}}(\boldsymbol{x}_k, \boldsymbol{v}_k)$ is available in closed form, only the neural component requires an auxiliary reformulation. Let

$$\boldsymbol{h}_{0,k} := \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{v}_k \end{bmatrix} \in \mathbb{R}^{n_0}$$

be the input of the neural network. For each hidden layer $\ell \in \{1, \ldots, L\}$, it holds

$$\boldsymbol{h}_{\ell,k} = \text{ReLU}(W_\ell \boldsymbol{h}_{\ell-1,k} + \boldsymbol{b}_\ell), \tag{11}$$

whereas the output layer is linear, namely

$$\widetilde{\boldsymbol{u}}_{\text{NN},k} = W_{L+1}\boldsymbol{h}_{L,k} + \boldsymbol{b}_{L+1}. \tag{12}$$

To encode the ReLU activation exactly with bilinear constraints, we introduce an auxiliary vector $\boldsymbol{s}_{\ell,k} \in \mathbb{R}^{n_\ell}$ and impose the constraints

$$W_\ell \boldsymbol{h}_{\ell-1,k} + \boldsymbol{b}_\ell = \boldsymbol{h}_{\ell,k} - \boldsymbol{s}_{\ell,k}, \tag{13a}$$

$$\boldsymbol{h}_{\ell,k} \geq 0, \tag{13b}$$

$$\boldsymbol{s}_{\ell,k} \geq 0, \tag{13c}$$

$$\boldsymbol{h}_{\ell,k} \odot \boldsymbol{s}_{\ell,k} = \mathbb{0}. \tag{13d}$$

© 2026 IEEE. Personal use of this material is permitted.

These constraints decompose each pre-activation component into its positive and negative parts and enforce their complementarity. The following proposition formalizes the exactness of this representation.

Proposition 1. *Fix $\ell \in \{1,\dots,L\}$. A pair $(\boldsymbol{h}_{\ell,k}, \boldsymbol{s}_{\ell,k}) \in \mathbb{R}^{n_\ell} \times \mathbb{R}^{n_\ell}$ satisfies* (13) *if and only if*

$$\boldsymbol{h}_{\ell,k} = \mathrm{ReLU}(W_\ell \boldsymbol{h}_{\ell-1,k} + \boldsymbol{b}_\ell)$$
$$\boldsymbol{s}_{\ell,k} = \mathrm{ReLU}(-(W_\ell \boldsymbol{h}_{\ell-1,k} + \boldsymbol{b}_\ell)),$$

component-wise. Consequently, constraints (12)–(13) *provide an exact lifted representation of the neural correction $\widetilde{\boldsymbol{u}}_{\mathrm{NN},k}$.*

Proof: The proof is written in component-wise form. Fix $i \in \{1,\dots,n_\ell\}$ and let h_i, and s_i denote the i-th components of $\boldsymbol{h}_{\ell,k}$, and $\boldsymbol{s}_{\ell,k}$, respectively. Let $z_i := (W_\ell \boldsymbol{h}_{\ell-1,k} + \boldsymbol{b}_\ell)_i$ denote the i-th element of the resulting vector of the affine transformation. Then (13) is equivalent to

$$z_i = h_i - s_i, \qquad h_i \geq 0, \qquad s_i \geq 0, \qquad h_i s_i = 0.$$

If $z_i > 0$, complementarity and nonnegativity imply $s_i = 0$ and $h_i = z_i$. If $z_i < 0$, they imply $h_i = 0$ and $s_i = -z_i$. For $z_i = 0$, one necessarily has $h_i = s_i = 0$. Hence

$$h_i = \max\{0, z_i\} = \mathrm{ReLU}(z_i),$$
$$s_i = \max\{0, -z_i\} = \mathrm{ReLU}(-z_i).$$

Conversely, these two definitions satisfy nonnegativity and complementarity, and

$$h_i - s_i = \max\{0, z_i\} - \max\{0, -z_i\} = z_i.$$

Thus (13) holds component-wise. Combining it with (12) yields the exact lifted representation of $\widetilde{\boldsymbol{u}}_{\mathrm{NN},k}$. ∎

C. Proposed Optimization Problem

Therefore, the formulation of Problem 1 is given by the following optimization problem

$$\max_{\boldsymbol{\delta}} \quad \sum_{k=1}^{N} \left\| \boldsymbol{y}_{\mathrm{LIN},k} - \boldsymbol{y}_k \right\|_2^2 \tag{14a}$$
$$\text{s.t.} \quad \boldsymbol{\xi}_0, \boldsymbol{x}_0 \quad \text{given}, \tag{14b}$$
$$\forall k \in [0, N-1]: \quad \boldsymbol{\xi}_{k+1} = A_d \boldsymbol{\xi}_k + B_d \boldsymbol{v}_{\mathrm{LIN},k}, \tag{14c}$$
$$\boldsymbol{v}_{\mathrm{LIN},k} = -K\big(\boldsymbol{r}_k - (\boldsymbol{y}_{\mathrm{LIN},k} + \boldsymbol{\delta}_k)\big), \tag{14d}$$
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + f_d(\boldsymbol{x}_k, \widetilde{\boldsymbol{u}}_k), \tag{14e}$$
$$\boldsymbol{v}_k = -K\big(\boldsymbol{r}_k - (\boldsymbol{y}_k + \boldsymbol{\delta}_k)\big), \tag{14f}$$
$$\widetilde{\boldsymbol{u}}_k = \widetilde{\boldsymbol{u}}_{\mathrm{LIN}}(\boldsymbol{x}_k, \boldsymbol{v}_k) + \widetilde{\boldsymbol{u}}_{\mathrm{NN},k}, \tag{14g}$$
$$\|\boldsymbol{\delta}_k\|_\infty \leq \varepsilon, \tag{14h}$$
$$\forall k \in [0, N]: \quad \boldsymbol{y}_{\mathrm{LIN},k} = C_d \boldsymbol{\xi}_k, \tag{14i}$$
$$\boldsymbol{y}_k = h(\boldsymbol{x}_k), \tag{14j}$$
$$\forall \ell \in [1, L]: \quad W_\ell \boldsymbol{h}_{\ell-1,k} + \boldsymbol{b}_\ell = \boldsymbol{h}_{\ell,k} - \boldsymbol{s}_{\ell,k}, \tag{14k}$$
$$\boldsymbol{h}_{\ell,k} \geq 0, \tag{14l}$$
$$\boldsymbol{s}_{\ell,k} \geq 0, \tag{14m}$$
$$\boldsymbol{h}_{\ell,k} \odot \boldsymbol{s}_{\ell,k} = \mathbb{0}, \tag{14n}$$
$$\widetilde{\boldsymbol{u}}_{\mathrm{NN},k} = W_{L+1} \boldsymbol{h}_{L,k} + \boldsymbol{b}_{L+1}. \tag{14o}$$

Proposition 2 (Feasibility). *Assume that the maps f_d, h, $\widetilde{\boldsymbol{u}}_{\mathrm{LIN}}$, and f_{NN} are well defined for every state and input generated over the finite horizon. Then, for every $\varepsilon \geq 0$ and given initial conditions $\boldsymbol{x}_0$ and $\boldsymbol{\xi}_0$, the feasible set of Problem* (14) *is nonempty.*

Proof: Choose $\boldsymbol{\delta}_k = \mathbb{0}$ for all $k = 0, \dots, N-1$, which satisfies $\|\boldsymbol{\delta}_k\|_\infty \leq \varepsilon$. The ideal and actual closed-loop states and outputs can then be recursively computed from the given initial conditions. For each neural network layer, let

$$\boldsymbol{h}_{\ell,k} = \mathrm{ReLU}(W_\ell \boldsymbol{h}_{\ell-1,k} + \boldsymbol{b}_\ell),$$
$$\boldsymbol{s}_{\ell,k} = \mathrm{ReLU}(-(W_\ell \boldsymbol{h}_{\ell-1,k} + \boldsymbol{b}_\ell)).$$

By Proposition 1, these variables satisfy the neural network constraints. Hence, all constraints in Problem (14) admit at least one feasible solution. ∎

The resulting formulation is a generally nonconvex nonlinear program, so standard solvers are not guaranteed to recover the global optimum. The ReLU relations are encoded exactly through bilinear complementarity constraints, which integrate directly into the algebraic optimization model and avoid explicit nonsmooth $\max$ operators. Exact mixed-integer ReLU formulations are also available [21], but would significantly increase the problem complexity.

Remark 1. *Let $J_N^\star$ denote the global optimal value of the Problem 1. Suppose that the optimizer returns a primal bound J_N^{pb} and a dual bound J_N^{db} such that*

$$J_N^{\mathrm{pb}} \leq J_N^\star \leq J_N^{\mathrm{db}}.$$

Assume moreover that $J_N^{\mathrm{pb}} > 0$ and that the relative gap satisfies

$$\frac{\left| J_N^{\mathrm{db}} - J_N^{\mathrm{pb}} \right|}{\left| J_N^{\mathrm{pb}} \right|} \leq \eta_{\mathrm{gap}}$$

for some $\eta_{\mathrm{gap}} \geq 0$. Then

$$J_N^{\mathrm{pb}} \leq J_N^\star \leq J_N^{\mathrm{db}} \leq (1 + \eta_{\mathrm{gap}})\, J_N^{\mathrm{pb}}.$$

Therefore, J_N^{pb} and J_N^{db} provide, respectively, a lower bound and an upper bound on the true worst-case discrepancy $J_N^\star$. In particular, the primal bound is η_{gap}-optimal in relative terms.

The proposed certification problem is intended as an *offline* analysis tool, rather than as an online optimization layer to be solved during closed-loop operation. For a fixed learned inversion law and finite horizon, it evaluates how far the closed-loop response may depart from the ideal feedback-linearized behavior under bounded measurement perturbations. The solution of Problem (1) therefore provides an a posteriori robustness assessment of a given controller architecture and training outcome, supporting model validation, comparison, and sensitivity analysis with respect to the perturbation budget.

© 2026 IEEE. Personal use of this material is permitted.

D. Discussion on the complexity

We now briefly discuss how the certification problem scales with the horizon and neural network size. Let $n_h := \sum_{\ell=1}^{L} n_\ell$ denote the total number of hidden neurons and let $n_\xi := \sum_{i=1}^{m} \gamma_i$ be the dimension of the linearized state. Introducing the auxiliary variables and exact ReLU reformulation, the number of variables and constraints scales as

$$\mathcal{O}\big(N(n_\xi + n + m + q + n_h)\big).$$

Hence, the formulation grows linearly with both the horizon N and the network size, while its main computational burden stems from the nonconvexity of the resulting NLP.

V. Numerical Results

This section evaluates the proposed framework on a double pendulum, assessing its ability to identify bounded measurement perturbations that amplify the discrepancy between the ideal linearized behavior and the actual closed-loop response under the learned inversion law.

A. Double pendulum

We consider a fully actuated planar double pendulum, modeled as a 2-DOF robotic manipulator with generalized coordinates $\boldsymbol{q} = [\theta_1, \theta_2]^\top$, state $\boldsymbol{x} = [\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top \in \mathbb{R}^4$, input $\boldsymbol{u} = [u_1, u_2]^\top \in \mathbb{R}^2$, and regulated output $\boldsymbol{y} = \boldsymbol{q}$. Its continuous-time dynamics are

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}) = \boldsymbol{u}, \tag{15}$$

where $M(\boldsymbol{q})$ is the inertia matrix, while $C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ and $G(\boldsymbol{q})$ collect the Coriolis–centrifugal and gravitational terms, respectively. In first-order control-affine form,

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \tag{16}$$

with

$$f(\boldsymbol{x}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ -M(\boldsymbol{q})^{-1}(C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q})) \end{bmatrix},$$

$$g(\boldsymbol{x}) = \begin{bmatrix} \mathbb{0}_{2\times 2} \\ M(\boldsymbol{q})^{-1} \end{bmatrix}.$$

Despite its low dimension, the system exhibits strong nonlinear coupling and is therefore a suitable benchmark for evaluating discrepancies induced by imperfect learned inversion. A schematic representation is shown in Fig. 3.

B. Experimental setup

The plant is discretized according to the sampled-data setting introduced in Section IV, with sampling time $\Delta t = 0.1$ s. The certification problem is solved over a finite horizon of length $N = 14$. The learned correction term is implemented by a feedforward neural network with two hidden layers followed by a final linear output layer. The hidden layers use ReLU activation functions, while the output layer is left linear. Each hidden layer contains $n_\ell = 128$ neurons. The network is trained for 300 epochs on a dataset built from multiple rollout sequences generated under the ideal inversion policy. Each rollout has duration 0.75 s.

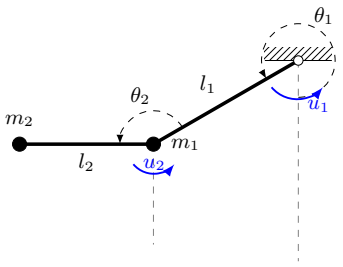


Fig. 3: Representation of the fully actuated double pendulum with state $\boldsymbol{x}$, joint torques as inputs, and joint angles as outputs.

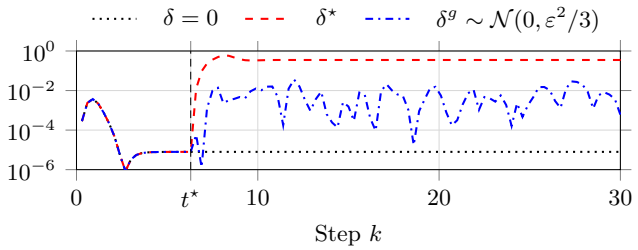


Fig. 4: Closed-loop error for $\varepsilon = 0.15$ under the optimized perturbation $\delta^\star$ and a Gaussian perturbation $\delta^g \sim \mathcal{N}(0, \varepsilon^2/3)$, activated at time $t^\star$.

C. Results

We first evaluate whether the proposed optimization procedure can identify harmful measurement perturbations. Since the certification problem is a nonconvex nonlinear program, the optimizer generally returns only a locally optimal feasible sequence $\boldsymbol{\delta}^\star$. Its objective value therefore provides a lower bound on the worst-case finite-horizon discrepancy.

In the first experiment, $\boldsymbol{\delta}^\star$ is applied to the measurement output with perturbation budget $\varepsilon = 0.15$. As a stochastic baseline, we consider

$$\delta_k^{\mathrm{g}} \sim \mathcal{N}\left(0, \frac{\varepsilon^2}{3}\right), \qquad k = 0, \ldots, N-1,$$

which has dispersion comparable to that of a uniform perturbation over $[-\varepsilon, \varepsilon]$, although it is not subject to the same pointwise bound. Figure 4 reports the resulting closed-loop error. Before the activation time $t^\star$, all curves coincide with

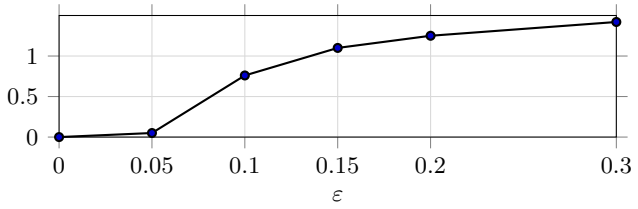


Fig. 5: Behavior of the cumulative discrepancy $\sum_{k=1}^{N} \|y_{\mathrm{LIN},k} - y_k\|_2^2$ evaluated for increasing values of ε.

© 2026 IEEE. Personal use of this material is permitted.

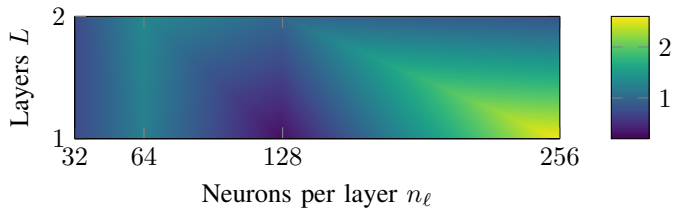


Fig. 6: Heatmap of the cumulative discrepancy $\sum_{k=1}^{N} \|\boldsymbol{y}_{\mathrm{LIN},k} - \boldsymbol{y}_k\|_2^2$ as a function of the number of hidden layers L and neurons per layer n_ℓ, with $\varepsilon = 0.15$.

the nominal response. After $t^\star$, the error induced by $\boldsymbol{\delta}^\star$ grows substantially faster than the produced by the Gaussian baseline, reaching values nearly two orders of magnitude larger. Hence, the optimization procedure identifies structured perturbations that are significantly more detrimental than random disturbances of comparable dispersion.

We also evaluate the cumulative discrepancy for increasing values of ε. In Figure 5 we observe a monotone trend which follows from the nested feasible sets: enlarging the perturbation budget cannot decrease the maximum achievable objective. Thus, larger measurement uncertainty allows progressively larger deviations from the ideal closed-loop behavior.

A second set of experiments is aimed at understanding how this discrepancy depends on the neural approximation itself. To this end, we retrain the learned correction for different network architectures, varying both the number of hidden layers $L = \{1, 2\}$ and the number of neurons per layer $n_\ell = \{32, 64, 128, 256\}$, while keeping the perturbation budget fixed at $\varepsilon = 0.15$. For each architecture, we evaluate the cumulative discrepancy and summarize the results in the heatmap of Fig. 6. The heatmap shows a clear dependence of the certified discrepancy on the network architecture. Intermediate or sufficiently expressive architectures attain the smallest cumulative errors, whereas shallow or poorly sized networks may produce substantially larger discrepancies. This indicates that the observed mismatch is influenced not only by the perturbation, but also by the approximation error of the learned correction. For a fixed perturbation budget, changes in the network architecture lead to significant variations in the discrepancy, suggesting that bounded perturbations amplify inaccuracies already present in the learned inversion map.

VI. Conclusions

This paper proposed an optimization-based framework to assess the discrepancy between an ideal feedback-linearized loop and a learning-based approximation under bounded measurement perturbations. Numerical results showed that the method identifies perturbations significantly more detrimental than Gaussian disturbances of comparable dispersion. They also revealed a strong dependence on the neural architecture, showing that model selection affects both nominal performance and sensitivity to perturbations. Future work will focus on less conservative and more computationally efficient certification methods.

References

[1] "Exact linearization methods," in *Nonlinear Control Systems: An Introduction*, A. Isidori, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 178–253.

[2] A. J. Krener, "Feedback linearization," in *Mathematical Control Theory*, J. Baillieul and J. C. Willems, Eds. New York, NY: Springer New York, 1999, pp. 66–98.

[3] J. Grizzle and P. Kokotovic, "Feedback linearization of sampled-data systems," *IEEE Transactions on Automatic Control*, vol. 33, no. 9, pp. 857–859, 1988.

[4] G. O. Guardabassi and S. M. Savaresi, "Approximate linearization via feedback — an overview," *Automatica*, vol. 37, no. 1, pp. 1–15, 2001.

[5] L. Menini and A. Tornambè, "Exact and approximate feedback linearization without the linear controllability assumption," *Automatica*, vol. 48, no. 9, pp. 2221–2228, 2012.

[6] S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Transactions on Automatic Control*, vol. 34, no. 11, pp. 1123–1131, 1989.

[7] F.-C. Chen and H. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Transactions on Automatic Control*, vol. 40, no. 5, pp. 791–801, 1995.

[8] A. Yeşildirek and F. Lewis, "Feedback linearization using neural networks," *Automatica*, vol. 31, no. 11, pp. 1659–1664, 1995.

[9] R. G. Goswami, P. Krishnamurthy, and F. Khorrami, "Data-driven deep learning based feedback linearization of systems with unknown dynamics," in *2023 American Control Conference (ACC)*, 2023, pp. 66–71.

[10] M. Ledda, D. Deplano, A. Giua, and M. Franceschelli, "On the computation of relu-based rnns equivalent to cpwa models of dynamical systems and vice versa," in *2026 IFAC World Congress*, 2026.

[11] J. Umlauft and S. Hirche, "Feedback linearization based on gaussian processes with event-triggered online learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4154–4169, 2020.

[12] T. Westenbroek, D. Fridovich-Keil, E. Mazumdar, S. Arora, V. Prabhu, S. S. Sastry, and C. J. Tomlin, "Feedback linearization for uncertain systems via reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1364–1371.

[13] C. De Persis, D. Gadginmath, F. Pasqualetti, and P. Tesi, "Data-driven feedback linearization with complete dictionaries," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 3037–3042.

[14] C. De Persis, D. Gadginmath, F. Pasqualetti, and P. Tesi, "Feedback linearization through the lens of data," *IEEE Transactions on Automatic Control*, vol. 71, no. 3, pp. 1630–1643, 2026.

[15] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using gaussian processes," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 5249–5255.

[16] S. B. da Cunha, "On the robustness of feedback linearization," *International Journal of Dynamics and Control*, vol. 12, pp. 3318–3331, 2024.

[17] M. Wei, L. Zheng, Y. Wu, H. Liu, and H. Cheng, "Safe learning-based control for multiple uavs under uncertain disturbances," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 7349–7362, 2024.

[18] M. Ledda, A. Giua, and M. Franceschelli, "Evaluating performance of neural network models for nonlinear dynamical systems via adversarial perturbations," in *2026 IEEE 22nd Conference on Automation Science and Engineering (CASE)*, 2026.

[19] M. Ledda, D. Deplano, A. Giua, and M. Franceschelli, "Certification of autoencoder-based models for dynamical systems," in *2025 IEEE 64th Conference on Decision and Control (CDC)*, 2025, pp. 401–406.

[20] P. Sosnin and C. Tsay, "Scaling mixed-integer programming for certification of neural network controllers using bounds tightening," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 1645–1650.

[21] J. Katz, I. Pappas, S. Avraamidou, and E. N. Pistikopoulos, "The integration of explicit mpc and relu based neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11 350–11 355, 2020.

© 2026 IEEE. Personal use of this material is permitted.