



UNIVERSITÀ DEGLI STUDI
DI CAGLIARI



Certification of Autoencoder-based Models for Dynamical Systems

Marco Ledda^{*}, Diego Deplano^{*}, Alessandro Giua^{*}, and Mauro Franceschelli^{*}

^{*}Department of Electrical and Electronic Engineering, University of Cagliari, Italy

IEEE 64th Conference on Decision and Control 2025 – 10–12 December – Rio de Janeiro, Brazil

Outline

- ① Introduction
- ② Problem Statement
- ③ Proposed Method
- ④ Numerical Simulations
- ⑤ Conclusion

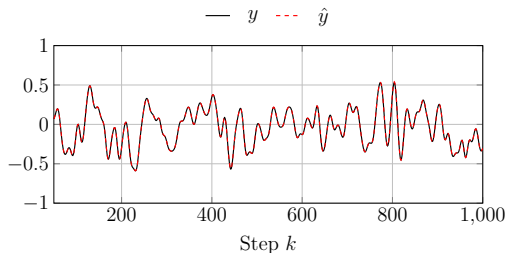
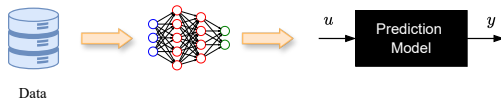
Outline

- ① Introduction
- ② Problem Statement
- ③ Proposed Method
- ④ Numerical Simulations
- ⑤ Conclusion

Introduction and Motivation

The need

- ✓ Abundance of data and hardware capabilities has led to the use of deep learning models [1, 2]
- ✓ Flexibility of DL models often comes at the price of **losing formal guarantees** on prediction errors.
- ✓ We need methods that **combine the expressive power of deep models with rigorous, worst-case error certification**.



[1] G. Pillonetto, A. Aravkin, D. Gedon, L. Ljung, A. H. Ribeiro, and T.B. Schon "Deep networks for system identification: A survey", Automatica.

[2] G. I. Beintema, M. Schoukens, and R. Toth, "Deep subspace encoders for nonlinear system identification", Automatica vol. 129, p. 109 666, 2021.

Preliminaries and Notation

Definition 1: Dynamical System

We consider discrete-time dynamical systems

$$\begin{cases} x_{k+1} = f(x_k, u_k) \\ y_k = g(x_k) \end{cases}$$

where $x_k \in \mathbb{R}^{n_x}$ is the state, $u_k \in \mathbb{R}^{n_u}$ the exogenous input, and $y_k \in \mathbb{R}^{n_y}$ the output, with $n_x, n_u, n_y \in \mathbb{N}$.

Definition 2: Artificial Neural Network

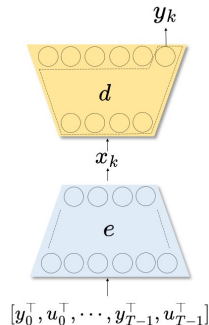
An ANN predicts y_k from a finite history of input-output pairs. Let $f_{\text{ann}}: \mathbb{R}^{T(n_y+n_u)} \rightarrow \mathbb{R}^{n_y}$. Given measurements $\{y_{k-T}, u_{k-T}, \dots, y_{k-1}, u_{k-1}\}$ of length T ,

$$\hat{y}_k = f_{\text{ann}}(y_{k-T}, u_{k-T}, \dots, y_{k-1}, u_{k-1}).$$

[3] D. Masti and A. Bemporad, "Learning nonlinear state-space models using autoencoders", *Automatica*, vol. 129, p. 109 666, 2021.

In this work, the ANN is an **autoencoder** [3] composed of:

- **Encoder** $e: \mathbb{R}^{T(n_y+n_u)} \rightarrow \mathbb{R}^{n_x}$: maps the history of measurements to a latent state representation.
- **Decoder** $d: \mathbb{R}^{n_x+1} \rightarrow \mathbb{R}^{n_y}$: reconstructs/predicts y_k from the latent state.



Outline

- ① Introduction
- ② Problem Statement
- ③ Proposed Method
- ④ Numerical Simulations
- ⑤ Conclusion

Problem Statement

- Given a **trained autoencoder-based model** used to predict the output of a dynamical system:

$$\hat{y}_k = f_{\text{ann}}(y_{k-T}, u_{k-T}, \dots, y_{k-1}, u_{k-1}).$$

- Dynamics of the system evolves according to $x_{k+1} = f(x_k, u_k)$, $y_k = g(x_k)$, with **admissible** inputs $u_k \in \mathcal{U}$, states $x_k \in \mathcal{X}$.
- At each step, the model incurs a **prediction error** $\varepsilon_k := \|\hat{y}_k - y_k\|_2^2$.

Certification Problem

- Goal:** quantify the worst-case one-step ahead prediction error

$$\varepsilon^* := \max_{(y_{0:T-1}, u_{0:T-1})} \|\hat{y}_T - y_T\|_2^2.$$

- In words: *What is the largest error*

$$\varepsilon := \|\hat{y}_k - y_k\|_2^2$$

that the given ANN can make when predicting the one-step ahead output of the dynamical system?

Why this matters

- ε^* provides a **formal error bound** for using the autoencoder inside control or decision-making loops.
- This bound can be interpreted as a **safety margin** on model predictions in safety-critical applications.

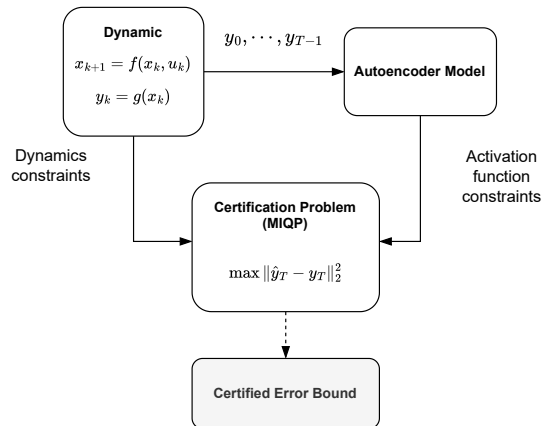
Related Certification Approaches

Method Class	Pros	Limitations
Convex-relaxation (Salman et al., 2019; Singh et al., 2019)	• Scalable to medium/large NNs • LP-based; efficient solvers available • Provides certified bounds on the output	• Relaxations of ReLU are conservative • Bounds may be loose for deep networks • Often unsuitable for dynamical systems where error accumulates
Lipschitz-based (Tsuzuku et al., 2018; Fazlyab et al., 2019)	• Very fast to compute • Global guarantees over large input sets • Works well with monotone or stable architectures	• Global Lipschitz bounds are overly conservative • Poorly captures local behavior or PWA structure • Hard to apply to closed-loop or multi-step predictions

Contribution

Main Contribution

- **Optimization-based certification framework** for autoencoder-based models used in dynamical systems.
- The certification problem is formulated as a **Mixed-Integer Quadratic Program (MIQP)** that searches for the **worst-case input sequence**.
- The input sequence found is applied to
 - the **system dynamics**, and
 - the **neural model**
- **Goal:** certify if the autoencoder output prediction is consistent with those generated by the known physical system.
- We are not merely finding a dangerous input sequence — we are identifying the worst one.



Outline

- ① Introduction
- ② Problem Statement
- ③ Proposed Method**
- ④ Numerical Simulations
- ⑤ Conclusion

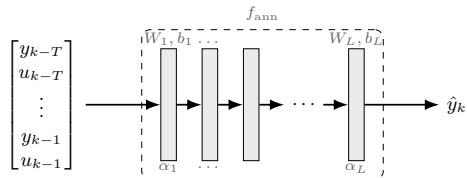
Assumptions

1: Piecewise Linear Dynamics

State space partitioned into convex polyhedra

$\mathcal{X}_i = \{x \in \mathbb{R}^n : \Pi_i x \leq \pi_i\}$. The dynamics is:

$$x_{k+1} = \begin{cases} A_1 x_k + B_1 u_k & \text{if } x_k \in \mathcal{X}_1 \\ \vdots & \\ A_m x_k + B_m u_k & \text{if } x_k \in \mathcal{X}_m \end{cases} \quad y_k = C x_k.$$



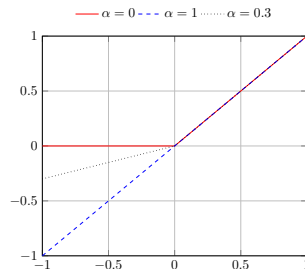
2: ANN to be certified

Fully-connected feedforward NN with L layers and n_ℓ neurons for $\ell \in [1, L]$ such that

$$f_{\text{ann}} = f_{L+1} \circ \dots \circ f_\ell \circ \dots \circ f_1,$$

$f_\ell : \mathbb{R}^{n_{\ell-1}} \rightarrow \mathbb{R}^{n_\ell}$ are *Leaky ReLU* layers:

$$f_\ell(h) = \text{LReLU}_\alpha(W^\ell h + b^\ell).$$



Formulation of the MIQP problem

- Find an input/output sequence that leads to the largest prediction error between the true system's output y_k and the ANN's prediction $\hat{y}_k$.
- Letting $k = T$ the prediction's time
 - past inputs $\mathcal{U}_p = \{u_0, \dots, u_{T-1}\}$ becomes **free** decision variables
 - past output $\mathcal{Y}_p = \{y_0, \dots, y_{T-1}\}$ becomes **constrained** decision variables

An answer to the problem can be found by solving this optimization problem:

$$\max_{\mathcal{U}_p, \mathcal{Y}_p, \mathcal{X}_T, y_T, \hat{y}_T} \|\hat{y}_T - y_T\|_2^2 \quad (1a)$$

$$\text{s.t.} \quad x_0 = \mathbb{0}, \quad (1b)$$

$$\forall k \in \{0, \dots, T-1\} : y_k = g(x_k), \quad x_{k+1} = f(x_k, u_k), \quad (1c)$$

$$y_T = g(x_T) \quad (1d)$$

$$\hat{y}_T = f_{\text{ann}}(y_0, u_0, \dots, y_{T-1}, u_{T-1}) \quad (1e)$$

Proposition 1 – Encoding PWA dynamics as MILP constraints

Original constraints in the certification problem

$$\begin{cases} y_k = g(x_k), \\ x_{k+1} = f(x_k, u_k), \\ y_T = g(x_T) \end{cases} \quad \forall k \in [1, T-1]$$



Replace by equivalent MILP constraints.

Mixed-integer linear reformulation

Region selection:

$$\forall k, j : \Pi_j x_k - \pi_j \leq U'(1 - \delta_{k,j})\mathbb{1}, \quad \forall k : \sum_{j=1}^m \delta_{k,j} = 1, \delta_{k,j} \in \{0, 1\}$$

where

$$U' := \max_{j=1, \dots, m} \left\{ \max \left\{ \sup_{x \in \mathcal{X}} \{ \Pi_j x - \pi_j \} \right\} \right\},$$

Local affine dynamics: introduce auxiliary variables $z_{k,j}$:

$$\begin{aligned} \forall k, j : \quad & z_{k,j} \leq U'' \delta_{k,j} \mathbb{1}, \quad z_{k,j} \geq L'' \delta_{k,j} \mathbb{1}, \\ & z_{k,j} \leq A_j x_k + B_j u_k - L''(1 - \delta_{k,j})\mathbb{1}, \\ & z_{k,j} \geq A_j x_k + B_j u_k - U''(1 - \delta_{k,j})\mathbb{1}, \end{aligned}$$

with

$$\begin{aligned} U'' &:= \max_{j=1, \dots, m} \left\{ \max \left\{ \sup_{x \in \mathcal{X}, u \in \mathcal{U}} \{ A_j x - B_j u \} \right\} \right\}, \\ L'' &:= \min_{j=1, \dots, m} \left\{ \min \left\{ \inf_{x \in \mathcal{X}, u \in \mathcal{U}} \{ A_j x - B_j u \} \right\} \right\}. \end{aligned}$$

State update:

$$\forall k : \quad x_{k+1} = \sum_{j=1}^m z_{k,j}.$$

Proposition 2 – Encoding ANN as MILP constraints

Original constraint

$$\hat{y}_T = f_{\text{ann}}(y_0, u_0, \dots, y_{T-1}, u_{T-1}).$$



Replace with an equivalent set of MILP constraints.

[4] J. Katz et al., “The Integration of Explicit MPC and ReLU-based Neural Networks,” IFAC-PapersOnLine, 2020.

Mixed-integer linear reformulation

Layer input and output:

$$h^0 = [y_0^\top \quad u_0^\top \quad \dots \quad y_{T-1}^\top \quad u_{T-1}^\top]^\top, \quad \hat{y}_T = f_{L+1}(h^L).$$

Affine pre-activation (layer ℓ):

$$z^\ell = W^\ell h^{\ell-1} + b^\ell, \quad \ell = 1, \dots, L.$$

LReLU activation — lower bounds:

$$h^\ell \geq z^\ell, \quad h^\ell \geq \alpha z^\ell.$$

LReLU activation — upper bounds (big- M):

$$h^\ell \leq z^\ell + M(1 - \delta_\ell), \quad h^\ell \leq \alpha z^\ell + M\delta_\ell,$$

where $\delta_\ell \in \{0, 1\}^{n_\ell}$ selects the active slope.

Big- M constant:

$$M := \max_{\ell=1, \dots, L} \left\{ \max \left\{ \sup_{x \in \mathcal{X}} \{ \alpha(W^\ell x + b^\ell), W^\ell x + b^\ell \} \right\} \right\}.$$

Implementation and Complexity

- Formulation requires determination of bounds U' , U'' , L'' , and M . They can be computed from the training dataset.
- Number of variables scales linearly with:
 - length T of the input/output sequence;
 - The dimension n_u of the dynamical system's output;
 - The dimension n_y of the dynamical system's input;
 - The dimension n_x of the dynamical system's state;
 - The dimension m of the dynamical system's linearity regions;
 - Total number $n_L = \sum_{\ell=1}^L n_\ell$ of neurons in all layers.
- **Boolean variables:** scales with $\mathcal{O}(T \cdot m + n_\ell)$
- **Number of real variables and number of constraints:** scales with $\mathcal{O}(T(m \cdot n_x + n_y + n_u) + n_\ell)$.

Outline

- ① Introduction
- ② Problem Statement
- ③ Proposed Method
- ④ Numerical Simulations**
- ⑤ Conclusion

Numerical Simulations Setup

Case Studies

- **Example 1: LTI system**

$$\Sigma_{\text{LTI}} := \begin{cases} x_{k+1} = \begin{bmatrix} 0.96 & -0.1 & 0 \\ 0.1 & 0.95 & 0 \\ 0 & 0 & 0.9 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix} u_k \\ y_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_k \end{cases}$$

- **Example 2: PWA System - Gear System**

$$\Sigma_{\text{PWA}} := \begin{cases} x_{k+1} = a_{q_k} x_k + b u_k - d \\ y_k = x_k \end{cases}$$

a_{q_k} is a gear-dependent coefficient.

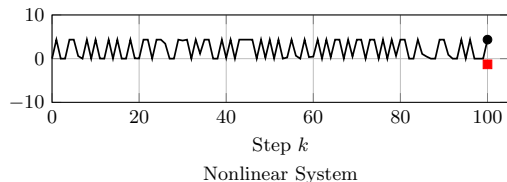
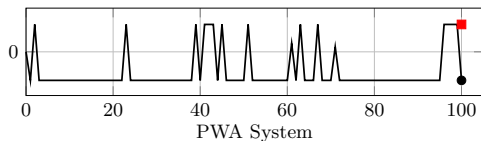
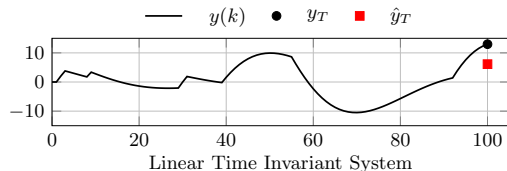
- **Example 3: Cascaded Tanks Nonlinear System**

$$\Sigma_T := \begin{cases} x_{k+1,1} = x_{k,1} - k_1 \sqrt{x_{k,1}} + k_2 u_k \\ x_{k+2,2} = x_{k,2} + k_3 \sqrt{x_{k,1}} - k_4 \sqrt{x_{k,2}} \\ y_k = x_{k,2} \end{cases}$$

Autoencoder and dataset

- Encoder/decoder: 3 hidden layers with ReLU (LReLU with $\alpha = 0$), final linear layer.
- Latent dimension: $n_{\Lambda+1} = 2n_x$.
- Observation window: $T = 100$ (pairs (y_k, u_k) from $k = 0$ to $k = 99$).
- Training set: 20 000 samples, validation set: 2 000 samples.
- Excitation: step sequences with random amplitudes (Gaussian, zero mean, unit variance).
- Added noise: Gaussian white noise with $\sigma = 0.02$.

Discussion of the results



- Black curve represents the observed outputs from time $k = 0$ to $k = 99$. Prediction occurs at time step $k = 100$.
- At prediction time $k = 100$ is shown the model's output (black marker) and the ANN prediction (red marker)
- Clear discrepancy in all the cases.
- **Nonlinear case:** nonconvex nature of the problem brings us to say that we have found a lower bound for the worst-case scenario.

Objective function values and solving time

System	Objective value	Solving time
Example 1	56.40	0.26 s
Example 2	58.98	0.90 s
Example 3	31.92	4.14 s

Outline

- ① Introduction
- ② Problem Statement
- ③ Proposed Method
- ④ Numerical Simulations
- ⑤ Conclusion

Conclusion and Future works

- We proposed an **MIQP-based certification method** for autoencoder-based models of dynamical systems.
- The certification problem is formulated as a **worst-case one-step prediction error maximization** over admissible input–output sequences.
- PWL dynamics and LReLU neural networks are **encoded exactly** via mixed-integer linear constraints (Propositions 1 and 2).
- Numerical tests on LTI, PWA gear, and nonlinear two-tank systems show non-trivial worst-case discrepancies between the AE and the true system.

Future Works

- Integrate certified error bounds into MPC schemes
- Interpret worst-case input–output sequences as **adversarial trajectories**, formalizing attack scenarios

Equivalence between Recurrent Neural Networks with ReLU Activations and Continuous Piecewise Affine Dynamical Systems^{*}

Marco Ledda^{*} Diego Deplano^{*} Alessandro Giua^{*}
Mauro Franceschelli^{*}

^{*} DIEE, University of Cagliari, 09123 Cagliari, Italy.

Under Review

Adversarial Learning Attacks on Autoencoder-based models for Nonlinear Dynamical Systems

Marco Ledda, Student, IEEE, Alessandro Giua, Fellow, IEEE, Mauro Franceschelli, Senior, IEEE

Under Review



UNIVERSITÀ DEGLI STUDI
DI CAGLIARI



Certification of Autoencoder-based Models for Dynamical Systems

Thank you!

Marco Ledda

`marco.ledda2@unica.it`

`www.marcoledda.it`



FULL PAPER HERE!