modely

A Framework for Designing Model-Structured Neural Networks for Modeling, Control, and Estimation of Physical Systems

October 3^{rd}, 2025

Prof. Gastone Pietro Rosati Papini

University of Trento

Overview

The Neu4mes Project

The Neu4mes Project

Pricipal Investigator Gastone Pietro Rosati Papini

Founded by the **Italian Ministry of University and Research** (MUR) under the *Science Italian Found 2023* program, for **1.2 Million Euros**.

Aim Develop a new approach to model and control physical system using **Model-Structured Neural Networks**

Outcome the nnodely framework, a **wiki website** for collecting MSNN models.

Case Study Several autonomous systems, including: **Drones**, **Quadrupeds**, **Vehicles**.

More details at https://tonegas.it/

Ministero dell'Università e della Ricerca

FIS

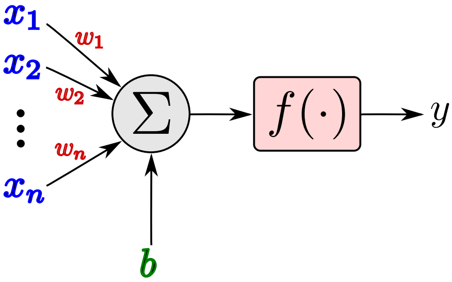
UNIVERSITY OF TRENTO

Motivation

Motivation

Beyond Black-Box Neural Networks

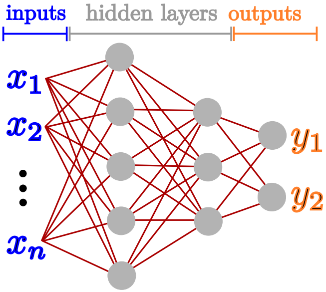
- A **neural network** (NN) is a mathematical model inspired by the structure of the human brain
- The basic component of a **neural network** (NN) is the **Perceptron**[1]



$$y = f\left(\sum_{k=1}^{n} x_k \cdot w_k + b\right)$$

[1] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain.". In: *Psychological review* (1958).

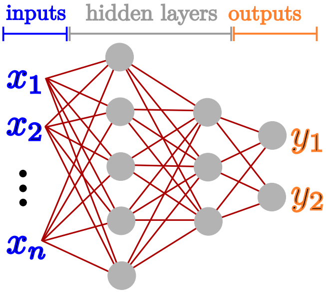
Multi-layer Perceptron (MLP)



- **Universal approximation theorem**[2]: MLP with 1 hidden layer approximates any continuous function
- … But is an MLP really the best choice for **every task**?

[2] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Math. Control Signal Systems* (1989).

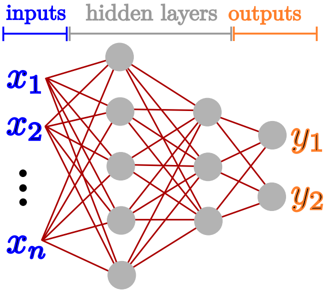
Multi-layer Perceptron (MLP)



- **Universal approximation theorem**[2]: MLP with 1 hidden layer approximates any continuous function $\longrightarrow$ in theory
- … But is an MLP really the best choice for **every task**?
  - **Internal architecture** of a NN matters $\longrightarrow$ Tensorflow Playground

[2] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Math. Control Signal Systems* (1989).

Multi-layer Perceptron (MLP)



- **Universal approximation theorem**[2]: MLP with 1 hidden layer approximates any continuous function ⟶ in theory
- … But is an MLP really the best choice for **every task**?
  - **Internal architecture** of a NN matters ⟶ Tensorflow Playground
  - Moreover big NN has common issues: data-hungry, black-box

[2]George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Math. Control Signal Systems* (1989).

The Internal Architecture Matters

Literature Examples

- *Computer vision*: **convolutional neural networks**, embedding the concept of spatial invariance[3]
- *Speech recognition*: **long short-term memory networks**, embedding the concept of short- and long-term memory[4]
- *Natural language processing*: **transformers** embed the concept of self-attention[5] (example: chatGPT)

[3] Yann LeCun, Koray Kavukcuoglu, and Clement Farabet. "Convolutional networks and applications in vision". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. 2010.

[4] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* (1997).

[5] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

Literature Examples

- *Computer vision*: **convolutional neural networks**, embedding the concept of spatial invariance[3]
- *Speech recognition*: **long short-term memory networks**, embedding the concept of short- and long-term memory[4]
- *Natural language processing*: **transformers** embed the concept of self-attention[5] (example: chatGPT)

- Can we devise **NN architectures** to model/control **physical systems**?
  - Improving: → **explainability, reliability and data-efficiency**

[3] Yann LeCun, Koray Kavukcuoglu, and Clement Farabet. “Convolutional networks and applications in vision”. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. 2010.

[4] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* (1997).

[5] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

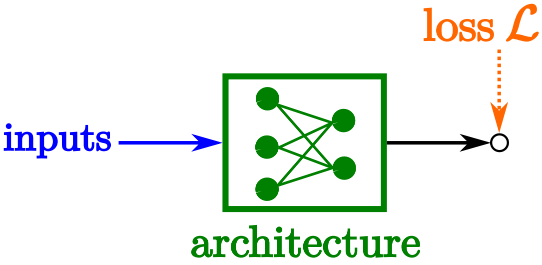
Motivation

Physics-Guided Learning

Physics-Guided NN Approaches

How to **embed physics**- or model-based knowledge in NNs?

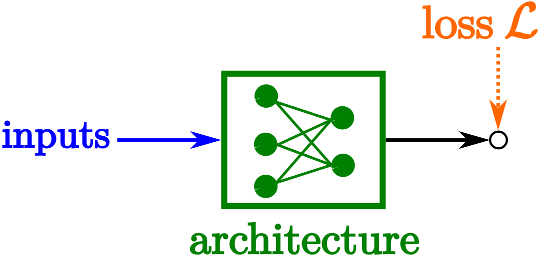
- Curate the training data or **NN inputs**
- Design a **loss function** $\mathscr{L}$
- Design the **internal architecture**



Physics-Guided NN Approaches

How to **embed physics**- or model-based knowledge in NNs?

- Curate the training data or **NN inputs**
- Design a **loss function** $\mathscr{L}$ $\rightarrow$ PINNs[6]
- Design the **internal architecture**

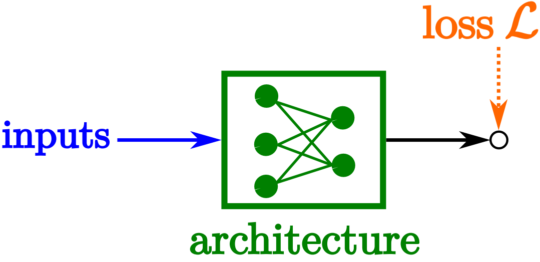


[6] M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* (2019).

Physics-Guided NN Approaches

How to **embed physics**- or model-based knowledge in NNs?

- Curate the training data or **NN inputs**
- Design a **loss function** $\mathscr{L}$ $\rightarrow$ PINNs[6]
- Design the **internal architecture** $\rightarrow$ **Model-Structured NNs** (MS-NNs)



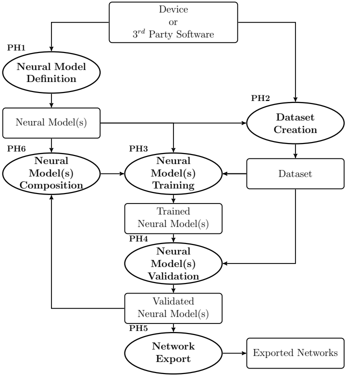
[6] M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* (2019).

Physics-Guided NN Approaches

How to **embed physics**- or model-based knowledge in NNs?

- Curate the training data or **NN inputs**
- Design a **loss function** $\mathscr{L}$
- Design the **internal architecture**

modely



NNodely

NNodely

Framework

modely
neuralize your model

The main concepts:

- Linear **superposition of forces**
- Discrete-time system dynamics modeled with finite/infinite impulse response layers (**FIR** or **IIR**)
- Unknown nonlinearities addressed with neuro-fuzzy **local models**
- Capturing the residuals of analytical models (such as friction, delays, and backlash) through **ad-hoc neuralization** with **custom funct.**
- Embedding the concept of **time** inside the NN
- Flexible managing of custom **recursive networks**

modely
neuralize your model

The aims:

- **Speed-up** the development of MS-NNs w.r.t. standard deep-learning frameworks (e.g., pure PyTorch or TensorFlow)
- **Support professionals** with classical modeling backgrounds, such as physicists and engineers, in developing **physics-guided ML** solutions
- A **repository** of incremental **know-how** to collect examples of MS-NNs for various applications

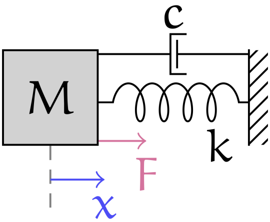
Use Cases

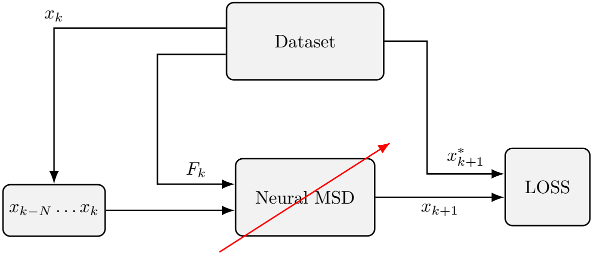
Use Cases

Modeling and Control of a Mass-Spring-Damper System

Mass-Spring-Damper: Modeling

Learn the **dynamics** of a Mass-Spring-Damper system (Neural MSD)

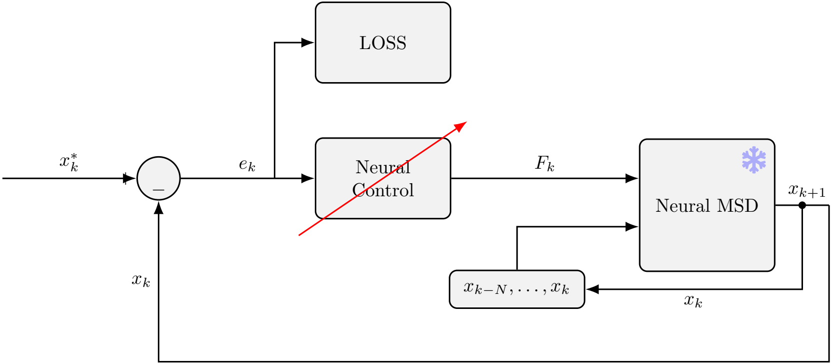




- Continuous dynamics: $\ddot{x} = (-kx - c\dot{x} + F)\,/M$
- Discretized dynamics: $x_{k+1} = \sum_{k=0}^{N_x-1} x[-k] \cdot h_x[(N_x - 1) - k] + F[0] \cdot h_F$
- MSNN dynamics: $x_{k+1} = \mathrm{FIR}(\bar{x}) + \mathrm{FIR}(F)$

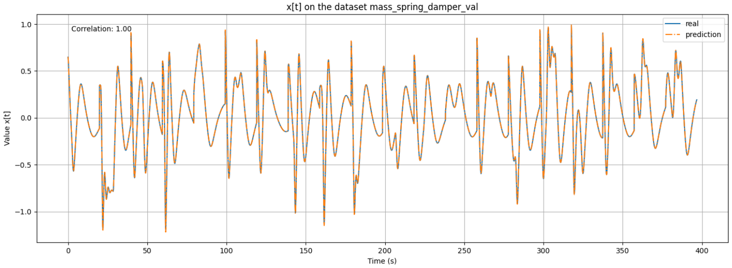
Mass-Spring-Damper: Control

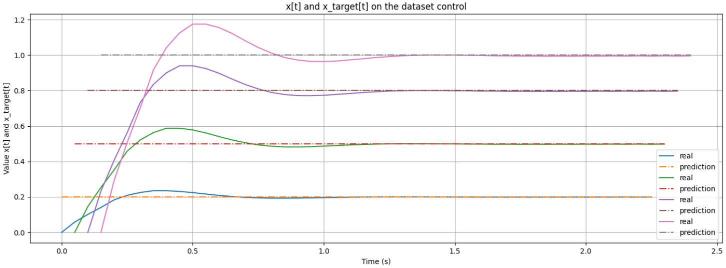
- Learn a **Neural PID** to control a mass-spring-damper system
- Controller gradients are propagated through the learned dynamic model (neural MSD) → **differentiable end-to-end** controller training



$$\text{LOSS} = \sum_{\text{examples}} e_k^2 = \sum (x_k - x_k^*)^2$$

Mass-Spring-Damper: Results for Modelling and Control

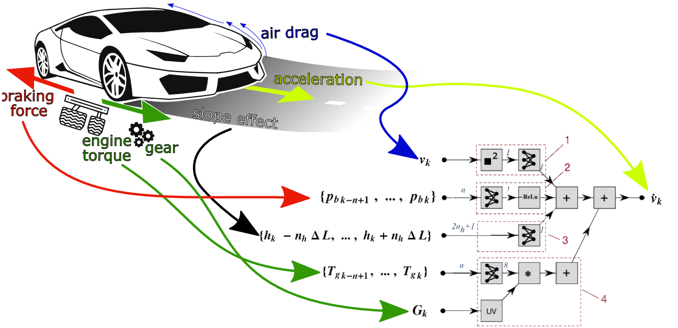




Use Cases

Modeling and Control of the Vehicle Dynamics

Modelling of Longitudinal Vehicle Dynamics[6]

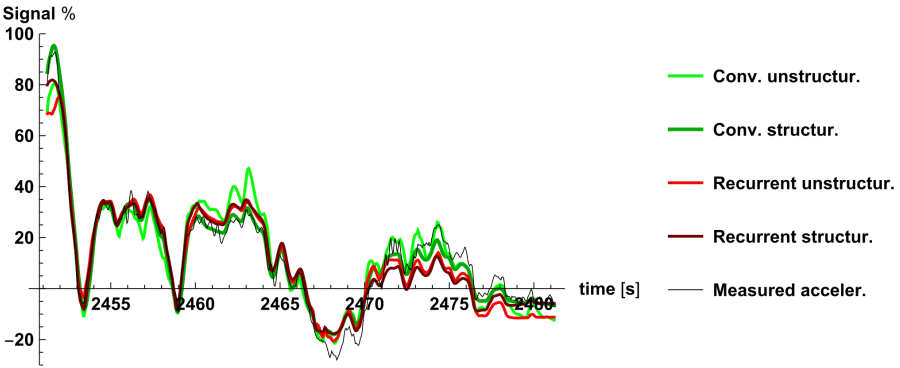


$$m\,\dot{v}_x = F_{x_f} + F_{x_r} - k_d v_x^2 - m\big(g c_r \cos(\theta) + g \sin(\theta)\big)$$

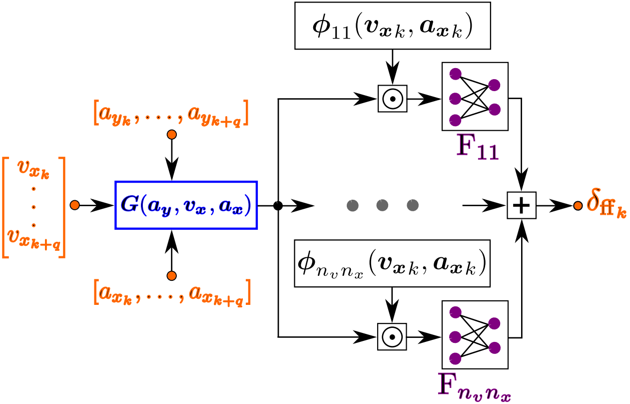
$$\dot{v}_x = \overbrace{\sum_i^{\text{gear}} g_i \mathrm{FIR}_i(\bar{T_g}) - \mathrm{Relu}(\mathrm{FIR}(\bar{p_b}))}^{(F_{x_f}+F_{x_r})/m} + \overbrace{\mathrm{FIR}(v_x^2)}^{k_d v_x^2/m} - \overbrace{\mathrm{Linear}(\bar{h_k})}^{g c_r \cos(\theta)+g\sin(\theta)}$$

[6] Mauro Da Lio, Daniele Bortoluzzi, and Gastone Pietro Rosati Papini. "Modelling longitudinal vehicle dynamics with neural networks". In: *Vehicle System Dynamics* (2020).

Results for modelling of longitudinal vehicle dynamics



Neural control of a lateral vehicle's dynamics for racing scenarios, experimental test.

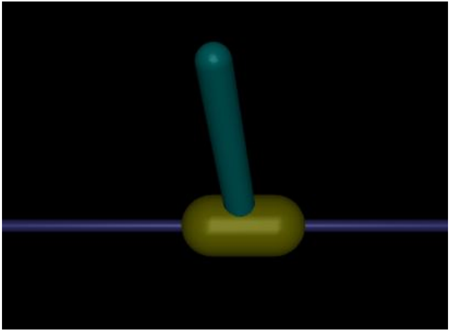


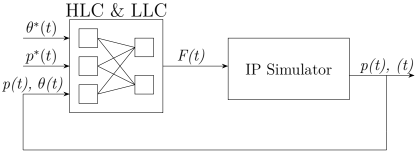
Show video of the F1Tenth car

Use Cases

Other Examples of MS-NNs in Different Domains

Control of an Inverted Pendulum

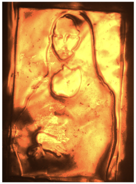
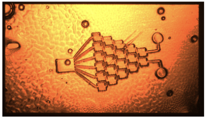
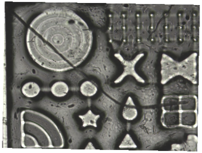




Show video

Controlling a 3D Printing Process

MS-NN to learn a 3D printing process parameters

$$\phi(t) = 1 - \exp^{-t^3 k}$$

Models nucleation and growth in crosslinking during polymerization.

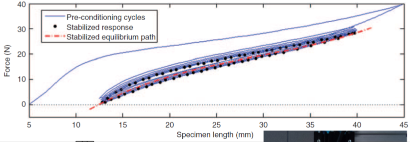
$$d_c = D_p \log\left(\frac{E}{E_C}\right)$$

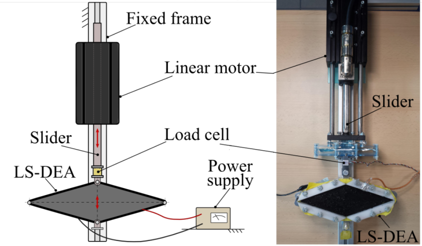
Describes how deep the resin will cure, depending on $\frac{E}{E_C}$.

- Degree of polymerization $\phi(t)$ (unitless)
- Reaction rate constant k (s^{-n})
- Cure depth d_c (μm)
- Penetration depth D_p (μm)
- Delivered energy dose E (mJ/cm^2)
- Critical energy E_C (mJ/cm^2)

MS-NN for physical modeling of a viscoelastic material

$$f_{\mathrm{tot}} = f_{\mathrm{e}}(\lambda) + f_{\mathrm{d}}\left([\lambda_t, \dots, \lambda], [\dot{\lambda}_t, \dots, \dot{\lambda}]\right) + f_{\mathrm{v}}(V, \lambda)$$



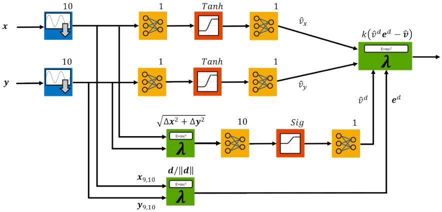


- total force realized: f_{tot}
- stretch: λ
- voltage: V
- elastic force: f_{e}
- viscoelastic and hysteretic force: f_{d}
- electric force: f_{v}

Learning a Pedestrian's Dynamics

MS-NN based on social force model[7]

$$\mathbf{f}(t) = m\frac{v^d(t)\mathbf{e}^d(t) - \mathbf{v}(t)}{\tau} + \sum_w \mathbf{f}_w^W(t)$$

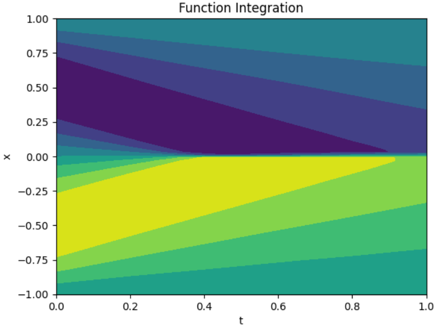


- forces on pedestrian: $\mathbf{f}$
- pedestrian mass: m
- pedestrian speed: $\mathbf{v}$
- desired velocity: v^d
- waypoint direction: $\mathbf{e}^d$
- velocity change rate: τ
- obstacle forces: $\mathbf{f}_w^W$

[7] Alessandro Antonucci et al. "Efficient Prediction of Human Motion for Real-Time Robotics Applications With Physics-Inspired Neural Networks". In: *IEEE Access* (2022).

Support to Physics-Informed Neural Networks (PINNs)

Integrate the Burger's equation[8] in the range $x \in [-1, 1]$, $t \in [0, 1]$.



$$u = NN(t, x),$$
$$\frac{u}{dt} + u\frac{u}{dx} - \left(\frac{0.01}{\pi}\right)\frac{u}{dx^2} = 0,$$
$$u(0, x) = -\sin(\pi x),$$
$$u(t, -1) = u(t, 1) = 0.$$

[8] M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* (2019).

NNodely-applications on GitHub

Our applications are available at
https://github.com/tonegas/nnodely-applications

Some of our Papers about MS-NNs

1. M. Piccinini et al., "A Physics-Driven Artificial Agent for Online Time-Optimal Vehicle Motion Planning and Control," in IEEE Access, vol. 11, pp. 46344-46372, 2023, doi: 10.1109/ACCESS.2023.3274836.
2. M. Da Lio, M. Piccinini and F. Biral, "Robust and Sample-Efficient Estimation of Vehicle Lateral Velocity Using Neural Networks With Explainable Structure Informed by Kinematic Principles," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 12, pp. 13670-13684, 2023, doi: 10.1109/TITS.2023.3303776.
3. E. Pagot, M. Piccinini, E. Bertolazzi and F. Biral, "Fast Planning and Tracking of Complex Autonomous Parking Maneuvers With Optimal Control and Pseudo-Neural Networks," in IEEE Access, vol. 11, pp. 124163-124180, 2023, doi: 10.1109/ACCESS.2023.3330431.
4. M. Piccinini, M. Zumerle, J. Betz and G. Pietro Rosati Papini, "A Road Friction-Aware Anti-Lock Braking System Based on Model-Structured Neural Networks," in IEEE Open Journal of Intelligent Transportation Systems, vol. 6, pp. 522-536, 2025, doi: 10.1109/OJITS.2025.3563347.
5. M. Da Lio, R. Donà, G. P. R. Papini, F. Biral and H. Svensson, "A Mental Simulation Approach for Learning Neural-Network Predictive Control (in Self-Driving Cars)," in IEEE Access, vol. 8, pp. 192041-192064, 2020, doi: 10.1109/ACCESS.2020.3032780.
6. M. Da Lio, D. Bortoluzzi and G. P. R. Papini, "Modelling longitudinal vehicle dynamics with neural networks", in Vehicle System Dynamics, vol. 58, no. 11, pp. 1675-1693, 2020, doi:1080/00423114.2019.1638947.
7. M. Piccinini, A. Mungiello, G. Jank, G. P. R. Papini, F. Biral, J. Betz, "Model-Structured Neural Networks to Control the Steering Dynamics of Autonomous Race Cars," in IEEE International Conference on Intelligent Transportation Systems (ITSC), 2025, accepted.
8. A. Antonucci, G. P. R. Papini, P. Bevilacqua, L. Palopoli and D. Fontanelli, "Efficient Prediction of Human Motion for Real-Time Robotics Applications With Physics-Inspired Neural Networks," in IEEE Access, vol. 10, pp. 144-157, 2022, doi: 10.1109/ACCESS.2021.3138614.
9. A. Mungiello, F. Jahncke, G. P. R. Papini, S. Santini, J. Betz, M. Piccinini, "Model-Structured Neural Networks to Learn the Dynamics of Robotic Vehicles," in Vehicle System Dynamics, 2025, under review.

modely

neuralize your model

Let's apply the framework on **your task**

and

Star it if you like!

Ministero dell'Università e della Ricerca

FIS

UNIVERSITY OF TRENTO

Gastone Pietro Rosati Papini
gastone.rosatipapini@unitn.it
tonegas.it