# nnodely: A Framework for Structured Neural Networks in Robotics

Gastone Pietro Rosati Papini, Alice Plebe, Mojtaba Sharifzadeh, Sebastiano Taddei, Mattia Piazza, Giovanni Scialla, Francesco Baroni, Giovanni Maria Francesco La Scala, Davide De Martini, Gioele Defrancesco, Filippo Faccini

***Abstract*— Robotic systems increasingly rely on learned models for perception, estimation, and control, yet neural networks are often developed as opaque artifacts with limited architectural structure and weak integration into robotic software pipelines, hindering modularity, validation, and reuse in safety-critical and resource-constrained applications. We advocate treating model-structured neural networks (MSNNs)—neural architectures that explicitly embed physical relations, parametric models, or control principles—as first-class software components within robotic systems. We present `nnodely`, an open-source framework supporting the structured design, training, composition, and deployment of such models through an intermediate architectural abstraction and interoperable export mechanisms. We outline the software engineering principles underlying `nnodely` and discuss open challenges toward systematic lifecycle management of structured neural components in robotics. `nnodely` is available at https://github.com/tonegas/nnodely.**

## I. INTRODUCTION

Neural networks are now pervasive in robotic systems, supporting perception, estimation, planning, and control. However, their development typically follows data-centric workflows and results in components that are weakly aligned with established robotics software engineering practices.

General-purpose frameworks such as PyTorch and TensorFlow provide flexible tensor abstractions, while specialized tools—including PINN libraries (e.g., DeepXDE), equation discovery frameworks (e.g., PySINDy)—support physics-aware learning or structured optimization. Conversely, robotic middleware and modeling environments (e.g., ROS 2, CasADi) focus on integration and deployment but treat learned models as opaque modules. Consequently, limited support exists for the systematic lifecycle management of structured neural components within robotic software systems.

MSNNs, which embed physical relations and control principles into neural architectures, bridge model-based and data-driven approaches [1], [2]. However, their development remains largely manual and fragmented. We introduce a framework enabling their structured design and interoperable integration within robotic systems.

The authors are with the Department of Industrial Engineering, University of Trento, Via Sommarive 9, 38123 Povo (TN), Italy (email: gastone.rosatipapini@unitn.it)

This work was supported under the FIS 2 Call, Grant Assignment Decree No. 1236 adopted on 01/08/2023 by the Italian Ministry of University and Research (MUR), for the project FIS-2023-03684 'Structured neural network framework for modeling and control of autonomous systems – Neu4mes', CUP E53C24003800001.

## II. NNODELY FRAMEWORK

`nnodely` is designed to support the engineering of MSNNs within robotic systems. The framework addresses the lack of structured development workflows for neural components.

At its core, `nnodely` provides an intermediate architectural representation that decouples high-level model specification from backend implementation, enabling portability and reproducibility. The framework structures the full development lifecycle—from model definition and dataset construction to training, validation, composition, and export—within a unified pipeline.

A key feature is the explicit management of interconnections between neural components. The framework distinguishes feedforward connections from closed-loop updates, enabling disciplined modeling of controllers and interconnected subsystems. In multi-model configurations, individual components can be selectively trained or exported, supporting modular development and incremental refinement.

Finally, `nnodely` ensures interoperability by exporting models to standard formats for integration with robotic middleware, simulation environments, and control toolchains. This positions structured neural networks as modular software elements rather than isolated training artifacts.

## III. CONCLUSIONS AND FUTURE DIRECTIONS

`nnodely` is under active development and aims to become a flexible framework for integrating structured neural components into robotic control pipelines. Key challenges include extending the development workflow to support seamless integration of neural modules from external researchers, strengthening interoperability with robotic control platforms and real-time stacks, and enabling efficient export and execution on embedded systems.

Overall, `nnodely` represents a step toward engineering MSNN-based components as reusable building blocks for robotics. Advancing these directions will strengthen its role as a framework for disciplined integration and deployment of structured neural models in robotic control architectures.

## REFERENCES

[1] 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)," *IEEE Access*, vol. 8, pp. 192 041–192 064, 2020.

[2] D. B. Mauro Da Lio and G. P. R. Papini, "Modelling longitudinal vehicle dynamics with neural networks," *Vehicle System Dynamics*, vol. 58, no. 11, pp. 1675–1693, 2020. [Online]. Available: https://doi.org/10.1080/00423114.2019.1638947