Short-term trajectory planning using reinforcement learning within a neuromorphic control architecture

Florian Mirus, Benjamin Zorn, Jörg Conradt
florian.mirus@bmwgroup.com, benjamin.zorn@bmwgroup.com, conradt@tum.de

Introduction

In this paper, we propose a neuromorphic system for vehicle control. Our system is implemented entirely in a spiking neuron substrate using the Nengo simulator [2] and is designed to be both distributed and hierarchical. The proposed architecture for a learnable and energy-efficient vehicle control system is a holistic neuromorphic approach for determining steering, gas and brake pedal as well as gear signals. In a sample instantiation, we train a trajectory selection module using reinforcement learning to investigate the feasibility of a learnable, neuromorphic control system in an automotive context. We evaluate our approach in TORCS (The Open Racing Car Simulator), which allows us to generate training data in a safe and controlled simulation environment.

System architecture

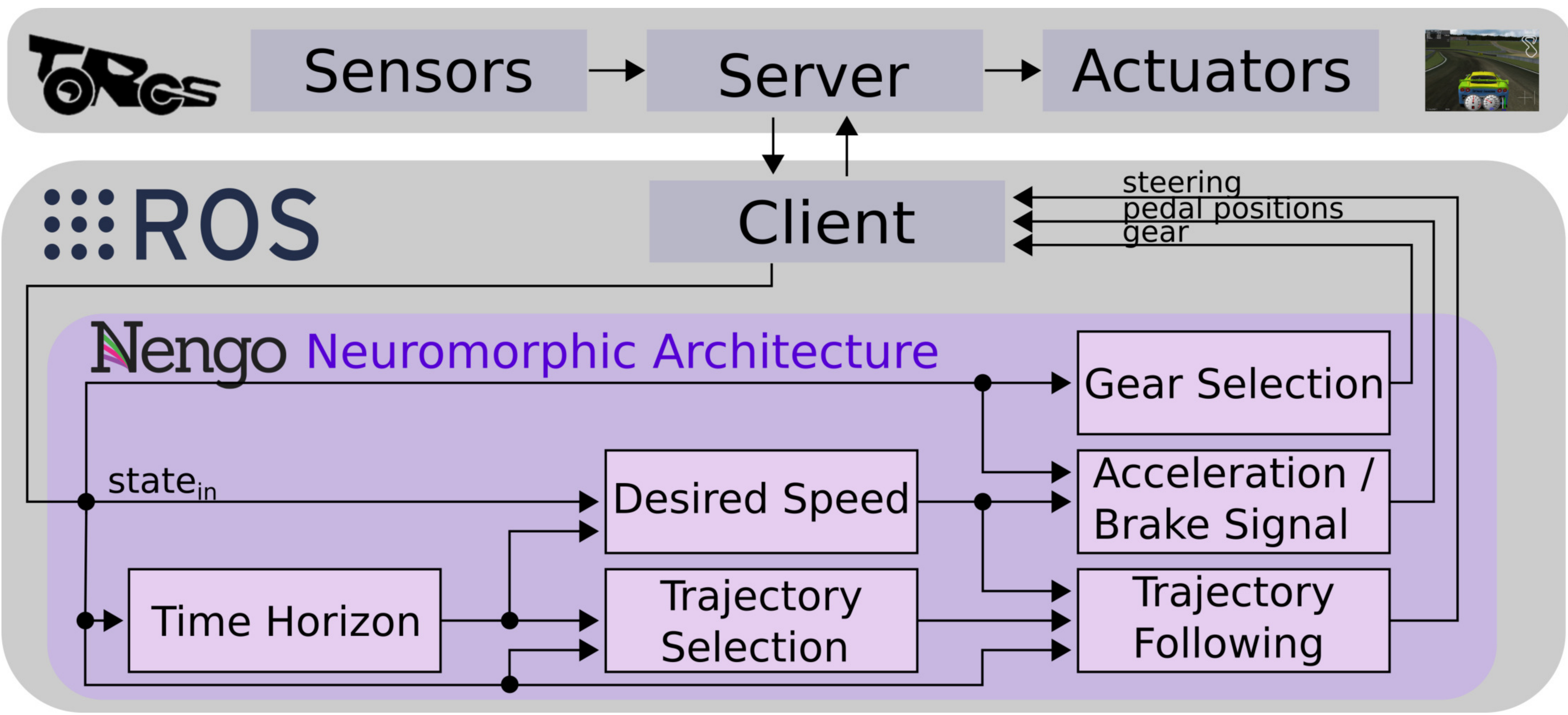

Figure 1: Proposed distributed neuromorphic architecture utilizing individual modules for separate control signal calculation.

Neuromorphic reinforcement learning

The core of this work is the learning Spiking Neural Network for *trajectory selection* consisting of two sub-networks (fig. 2). The yellow subnetwork A encodes the current state in a neural population. The cyan subnetwork B encodes the reward received from the environment:

$$r(t) = |p(t)|\,\beta_1 \cdot |\theta(t)|\,\beta_2 + |p(t)-p(t-1)|\,\beta_3 + |\theta(t)-\theta(t-1)|\,\beta_4, \quad (1)$$

The associative learning process is implemented using the PES (Prescribed Error Sensitivity) [1] learning rule, which modifies connection weights based on a (multi-dimensional) error signal $\mathbf{e}$:

$$e_j = \begin{cases} r(t) - Q(s(t), a_j) & \text{if } a_j \text{ is selected} \\ 0 & \text{else.} \end{cases} \quad (2)$$

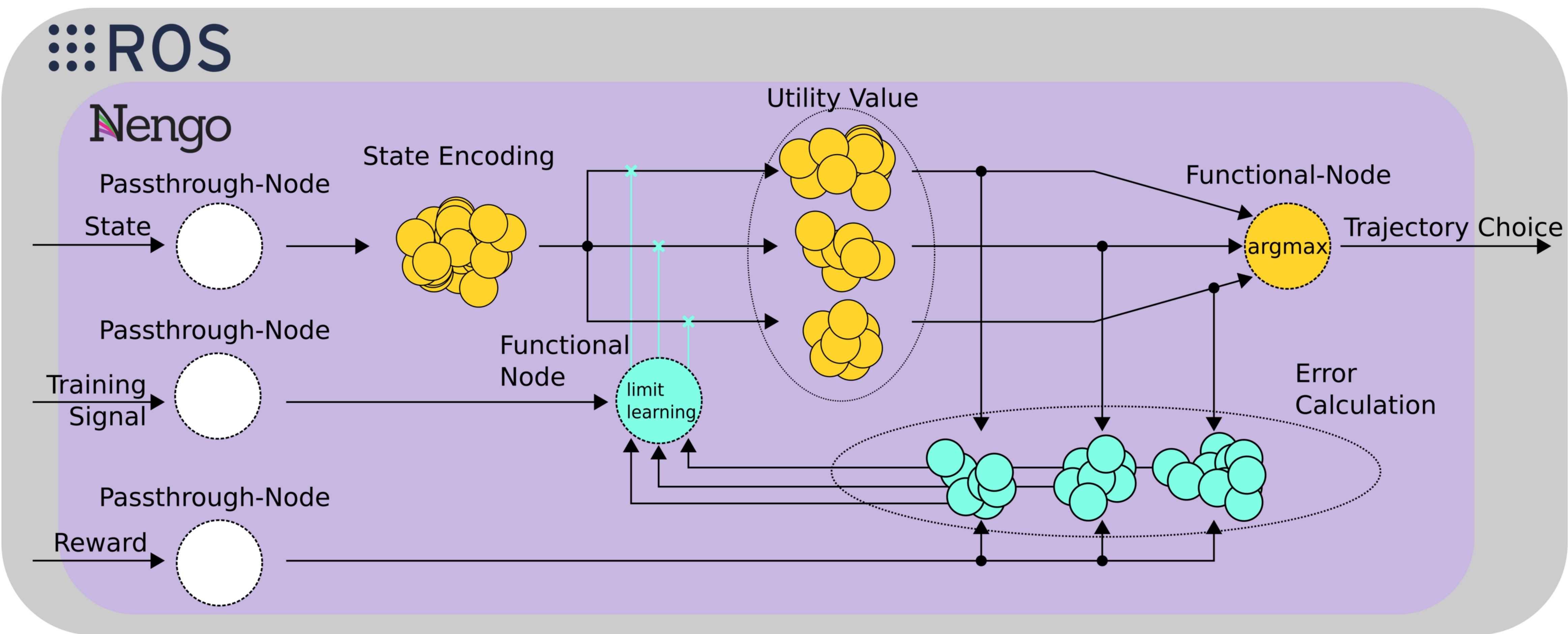

Figure 2: Simplified visualization of the SNN (Spiking Neural Network) for *trajectory selection*. The yellow (A) resp. cyan networks (B) implement action selection resp. weight optimization for three exemplary trajectory choices

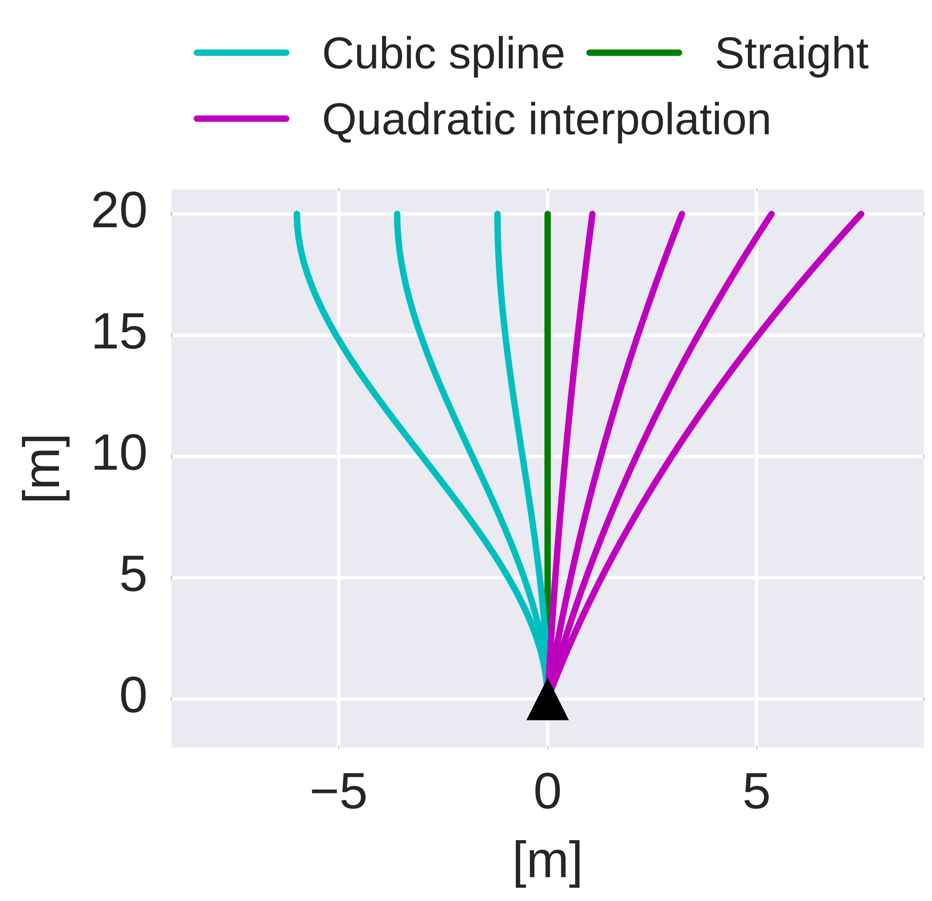

Figure 3: Visualization of a set of exemplary trajectories.

Experiments

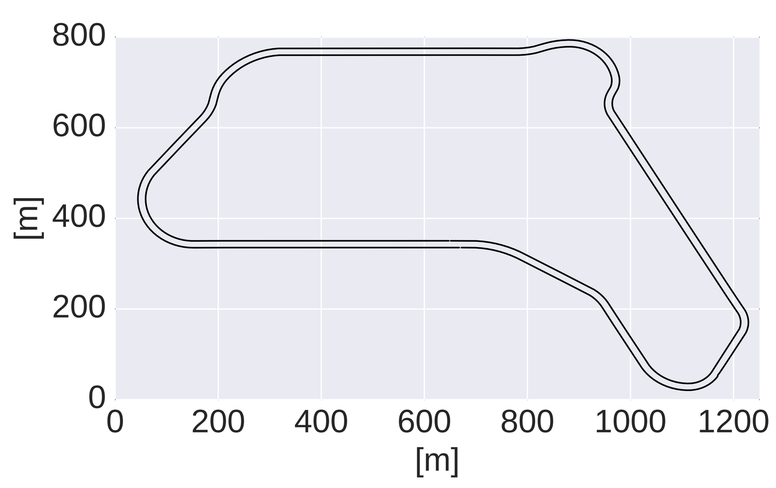

Figure 4: Training track

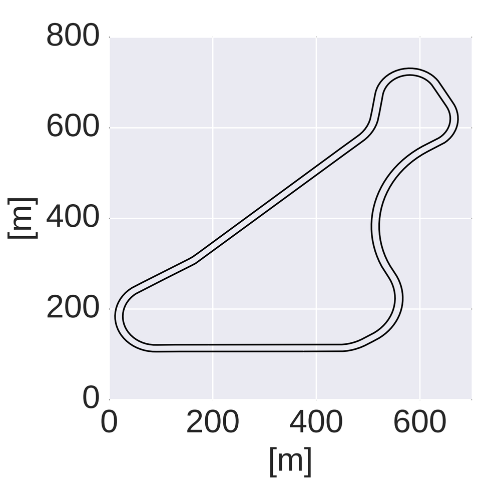

Figure 5: Validation track

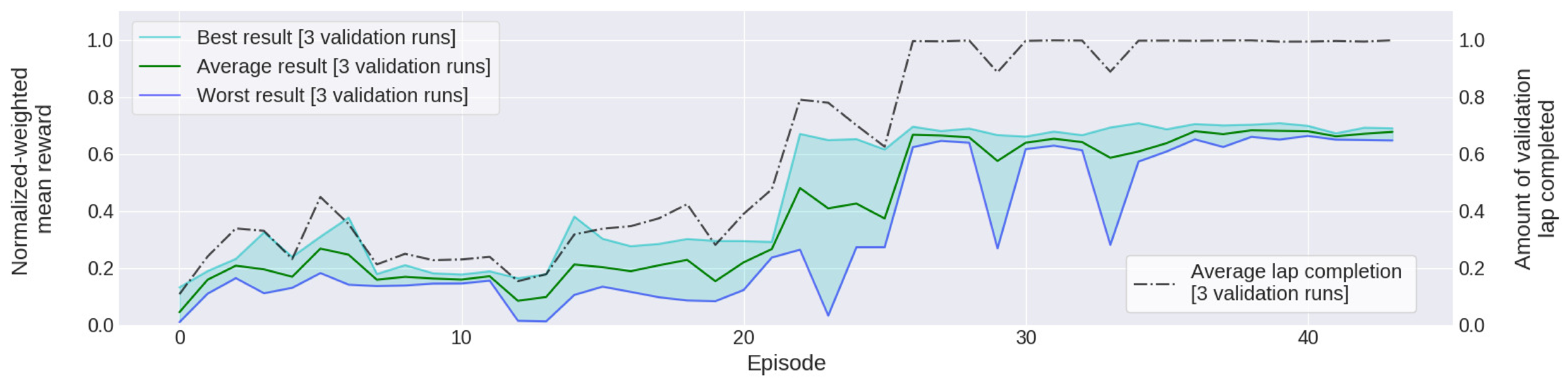

Figure 6: Mean reward normalized by amount of lap completion for each validation episode.

References

[1] T. Bekolay, C. Kolbeck, and C. Eliasmith. "Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks". In: *35th Annual Conference of the Cognitive Science Society*. 2013, pp. 169–174.

[2] T. Bekolay et al. "Nengo: a Python tool for building large-scale functional brain models". In: *Frontiers in Neuroinformatics* 7 (2014), p. 48. ISSN: 1662-5196.