A-NC: Adaptive Neural Control with implicit online inference of privileged parameters

Marcin Paluch* PALUCHM@INI.UZH.CH
Florian Bolli* BOLLIF@INI.UZH.CH
Pehuen Moure PEHUEN@INI.UZH.CH
Xiang Deng DXIANG@INI.UZH.CH
Tobi Delbruck TOBI@INI.UZH.CH
Institute of Neuroinformatics, ETHZ/UZH, Winterthurerstrasse 190, 8057 Zürich

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

Rapid changes in the environment and robot parameters pose significant challenges for control systems, particularly when key parameters are not directly measurable. In this paper, we introduce a novel approach using classical Recurrent Neural Network (**RNN**) controllers to dynamically adapt policies in response to these changes. We propose strategies for data collection and processing that enable the successful training of efficient Gated Recurrent Unit (**GRU**) nonlinear controllers capable of adapting to changing parameters.

We demonstrate this approach using a simulated and a physical cartpole robot. The RNNs are trained through supervised learning on data generated in simulation using Nonlinear Model Predictive Control (**NMPC**). We vary the cartpole's angle sensor offset or pole length jointly with pole mass, none of which are directly measurable by the robot. Our results show how the RNN controller adjusts its policy based on past trajectories, leading to control that mimics the NMPC, outperforming Domain Randomization (**DR**) technique applied to feedforward neural networks. Unlike NMPC, which relies on explicit knowledge of environment parameters, the RNN implicitly estimates these parameters from past trajectories, allowing it to adapt its control policy dynamically. It also outperforms NMPC control performance when the parameters relevant for NMPC are not known.

Keywords: Adaptive Control, Meta-Learning for Robotics, RNNs

1. Introduction

Unmeasurable parameters in dynamical systems continue to pose a major challenge for modern control engineering. While Nonlinear Model Predictive Control (NMPC) can adapt when values of system parameters are known, its reliance on direct measurements and high computational cost limit its applicability.

Robust or adaptive control algorithms address incomplete information about parameter values. Robust methods aim to ensure task completion despite system dynamics changes, often at the expense of efficiency. In contrast, adaptive controllers modify their policies in response to discrepancies between expected and observed behaviors, enabling better performance. They are rooted in the MIT rule (Whitaker, 1959) and have evolved since the 1950s (Annaswamy and Fradkov, 2021). Yet, the need for improved solutions persists, reflected in technological advances and novel reinterpretations of classical methods, e.g. of the L1-adaptive control method (Hanover et al., 2022).

* These authors contributed equally to this work.

In this work, we propose an "**Adaptive Neural Controller** (**A-NC**)". It adapts to changes in privileged parameters (**PP**) - not directly measurable variables affecting robot behavior - using GRU RNN (Cho et al., 2014). It is trained via supervised learning on a dataset generated by an NMPC controlling a cartpole system. The NMPC is fully informed of parameter variations, while the A-NC receives only the robot's current state, targets, and most recent control action. We show that the A-NC leverages its memory of past trajectories — encoding and storing information about changing parameters into its hidden states — to produce control signals closely matching the parameter-aware NMPC. Its control performance significantly surpasses that of experience-deprived networks trained on the same dataset, which can only approximate the average NMPC behavior across parameter variations. The A-NC achieves control performance comparable to networks explicitly trained for true parameter values and outperforms simple DR training. Additionally, it is orders of magnitude faster to compute than the NMPC.

We analyze the GRU's memory states to understand their capacity to encode PP variations. We demonstrate that PP changes can be directly decoded from these memory states using Principal Component Analysis (**PCA**) and a Multilayer Perceptron (**MLP**).

To facilitate reproducibility and encourage further research, we provide an open-source implementation of the proposed methods, including the dataset generation process, training framework, and evaluation tools. The codebase is available at `https://github.com/SensorsINI/CartPoleSimulation/tree/L4DC25-adaptive-neural-control`.

2. Related Work

Our work strongly relates to two research directions.

The first research direction involves inferring PPs from sensor data using neural networks. Some tasks require only information from instantaneous observations (Chen et al., 2019). Kaufmann et al. (2020) provides temporal input to a stateful network similar to ours but does not demonstrate whether memory is essential. In contrast, inferring PPs changes from past trajectories inherently depends on temporal information. Methods targeting this task (e.g., Kumar et al. (2021), Chiappa et al. (2022), Lee et al. (2020)) adopt a multi-network strategy: one network encodes PPs into a latent space, another maps this latent representation to control outputs, and a third predicts PPs from system dynamics. In contrast, our method shows that a single RNN suffices to adapt the control policy using only temporal information stored implicitly in the network's memory.

The second research direction pertains to deep meta-learning, broadly defined as pre-training a network to quickly adapt to any task within a targeted family of tasks. A common strategy is to optimize the network during pretraining, not for task-specific performance directly, but for adaptability—ensuring strong performance after additional learning steps before execution. These additional steps may involve gradient descent (Finn et al., 2017) or leveraging the memory capabilities of RNN (Hochreiter et al., 2001; Duan et al., 2016; Wang et al., 2016). In the latter approach, the network processes a sequence of inputs to identify the target task, adjusting its memory state to modify its output and optimize performance for that specific task. Our method adopts this principle by using past system trajectories to adapt the control policy to the changes of the underlying parameters and can thus be viewed as a form of meta-learning, where the task-specific configuration is stored in the RNN hidden states.

A closely related work is Subramoney et al. (2024), which applies a similar algorithm to control a simulated robotic arm using a recurrent Spiking Neural Networks (**SNN**). However, that work primarily focuses on meta-learning within the SNNs, and only briefly explores its potential applications to robotics. We extend this approach to the physical cartpole and provide a novel perspective: rather than framing the problem as "learning to learn" a policy with limited samples, we interpret it as implicitly inferring parameters from past trajectories to adjust the control policy, thereby achieving an adaptive control algorithm.

Adaptive control methods often use DR training, however, existing studies do not clearly distinguish between three potential outcomes of applying DR with RNNs: (1) a control policy averaged across parameters variation, (2) a policy optimized for the best average performance across parameters variation (achievable through reinforcement learning), and (3) dynamic adaptation of the policy to a parameter value, which is the focus of this work. The most closely related example can be found in a report of the robotic competition MyoChallenge23 (Caggiano et al., 2024), where the winning team implemented a controller similar to the method described in this paper. Nonetheless, the authors merely suggest that adaptive mechanisms of the kind explored here may exist and may have contributed to the success of their design.

3. Methods

We propose a method for developing an A-NC capable of dynamically adjusting to changes in system parameters and environmental conditions. This method involves generating diverse training data in simulation, leveraging an optimal controller to generate control signals, and training an RNN to imitate the controller while inferring PPs from recent dynamics. The following subsections detail the key components of this method.

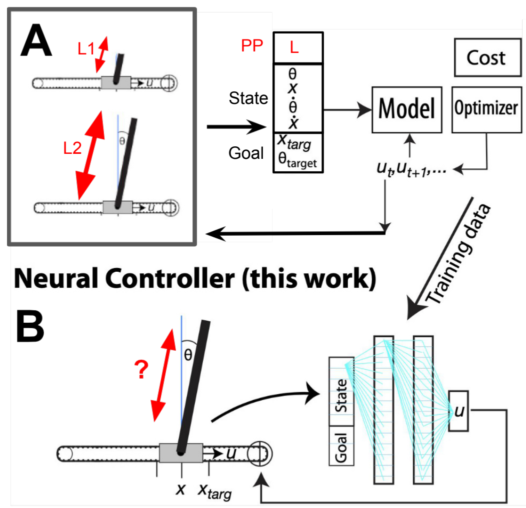

Figure 1: Overview of the proposed method.

3.1. Data Collection and Augmentation

We collect state-action pairs in simulation across a range of privileged parameters (e.g. pole length, sensor offset, etc.) using a Nonlinear Model Predictive Control (**NMPC**), as illustrated in Fig. 1A. It operates with explicit knowledge of the system parameters and assumes ideal measurement conditions. For this reason, it is referred to as the **informed** controller in this work.

Parameters representing robot and environmental dynamics are systematically varied during data collection. This enables the RNN to adapt across a wide range of system behaviors and uncertainties.

We found that it was not sufficient to train the A-NC on raw MPC data. We used this data, but augmented it with the following three methods.

STATE SPACE COVERAGE AND PP DISTURBANCES

We introduced suboptimal trajectories into the dataset by alternating between "uninformed" and "informed" NMPC during data collection. The NMPC operated with the correct PPs for 4.5 s before transitioning to random PPs for 1.5 s. This uninformed state causes the NMPC to deviate from the optimal trajectory. This alternation ensured that the A-NC was exposed to realistic scenarios where it initially deviates from the planned trajectory after a sudden change in the PPs. However, the system recovers after the NMPC returns to the informed state.

The control sequences of the uninformed NMPC segments are suboptimal and not what we want the A-NC to learn. Therefore we overwrote them in the dataset with new control generated by an informed controller, operating along the same suboptimal trajectory. This way, the suboptimal state trajectories are retained, but the A-NC can learn the optimal corrective actions for these PP disturbances. Only the control is overwritten, while the suboptimal state history is preserved. This decoupling of the state history from the control ensures that the dataset provides examples of disturbances together with optimal recovery strategies. We also added noise to the NMPC control to further improve the state exploration and the A-NC robustness.

ROBUSTNESS TO SENSOR NOISE

In some random short sequences within the training dataset, we replaced true state values with random values, while the PPs remained constant. We hypothesize that it helps the A-NC to hold a stable representation of the PPs. The angle data used for training was quantized to match the resolution of the 12-bit pole angle ADC. The derived state, $\dot{\theta}$ (angular velocity), was also affected by this quantization. Incorporating sensor quantization into the training data reduces the simulation-to-reality gap.

3.2. RNN Training Procedure

The RNN is trained to clone the behavior of the optimal controller by predicting control actions directly from the state and the goal. In contrast to the informed teacher controller, the RNN only has access to the observable states without explicit knowledge of the varying privileged parameters, as illustrated in Fig. 1 B. The input to the RNN includes the immediate physical states of the system as well as the objectives that the optimal controller is supposed to reach (e.g., target cart position, target pole equilibrium up or down).

The network is trained to minimize the discrepancy between the predicted control and the optimal one using a mean squared error loss. The RNN architecture leverages the temporal information to implicitly infer the PPs from the past trajectory.

3.3. Comparing Adaptive Control and Domain Randomization

One of the main challenges in studying the adaptive control presented in this paper is distinguishing its effect from that of DR, as commonly employed in reinforcement learning. DR trains a Neural Controller (**NC**) to find a single, generalized policy that works reasonably well across the entire range of varying PPs included in the training data. While this approach achieves robustness to varying conditions, it inherently leads to a suboptimal policy, as the optimal policy would change with the PPs.

In contrast, adaptive control relies on the ability of the controller to implicitly estimate the specific PPs based on past trajectories and dynamically adjust its control policy. This allows the controller to achieve performance that is optimal for the given environment or system parameters, rather than settling for a ”one-size-fits-all” solution.

To explore this distinction, we introduce the concept of an “Experience-Deprived Neural Controller (**ED-NC**)”. An ED-NC is trained on a dataset where the history of states and control signal — and the PPs implied by them — is deliberately made inconsistent with the NMPC policy the network is requested to imitate. The ED-NC was provided with same state history as the A-NC but was trained to return a control generated with randomly varying PPs. This approach removes correlation between the actual PPs of the informed NMPC controller and the state history, preventing the network from learning or inferring the system's parameters from sequential data. As a result, the ED-NC is forced to rely solely on the current state and target, mimicking the behavior of a DR-based approach.

By depriving ED-NC in this way, it serves as a baseline to compare against the A-NC. This approach removes the controller's ability to adapt based on past states. It ensures that any differences in control performance results from the presence or absence of memory in the A-NC, providing a clear comparison between adaptive control and DR.

Note: This behavior could also be achieved by training an MLP on the same dataset, since feedforward MLPs have no memory. However, for a fair comparison, we ensured that both the baseline (ED-NC) and the evaluated network (A-NC) share the exact same architecture and learning capacity.

4. Experiment Setup

4.1. Cartpole

Fig. 2 shows the cartpole system. It is a well-known underactuated nonlinear control problem where a belt-driven cart moves along a linear track with a pole attached via a pivot joint. The primary objectives are to swing the pole from a resting position to an upright configuration (swing-up) and to maintain it in the upright position (balance) at a target cart position along the track. The system measures the cart's position (x) using a motor encoder and the pole's angle (θ) using a potentiometer. The potentiometer includes a screw that sets the arbitrary angle offset Θ, which we typically set so that the jump from zero to maximum voltage occurs during a swing-up and not during balancing. An intermediate voltage represents the vertical pole angle. Any small error in this value makes it impossible to balance the pole because the controller targets an angle away from zero, where the pole is falling. We can change the pole length L and mass M by swapping the pole for shorter or longer poles and between aluminum and steel poles. This setup represents a classic example of an underactuated nonlinear system, making it ideal for testing nonlinear adaptive control algorithms in a tabletop setup.

4.2. Data Collection and Training

All data was collected in simulation using an **informed** NMPC with the Resampling Parallel Gradient Descent (**RPGD**) optimizer (Heetmeyer et al., 2023) running at control frequency of 50 Hz. Both the simulator and the NMPC used the cartpole dynamical equations from Green (2020). The informed NMPC generates optimal control signals by re-evaluating strategies at each timestep, accounting for dynamic environments.

The informed NMPC dynamics model had the true Θ, L, and M, and received Table 1 input features. The cost function penalized deviations of the pole angle from the target equilibrium (ξ), the cart position from the target position (χ), large control output (F), and high kinetic energy of the pole. Details of the cost function configuration are available here.[1].

1. https://github.com/SensorsINI/CartPoleSimulation/blob/L4DC25-adaptive-neural-control/Control_Toolkit_ASF/config_cost_function.yml

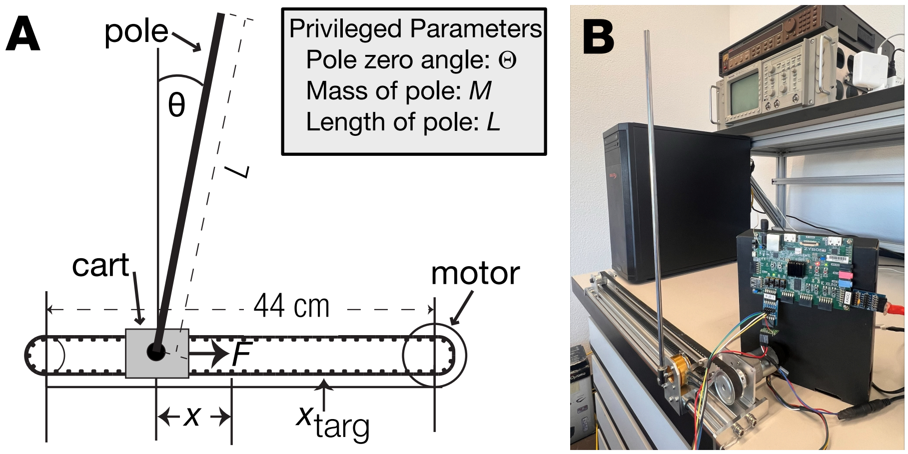

Figure 2: Cartpole system. **A**: key state variables (x, $\dot{x}$, θ, $\dot{\theta}$) and the control input (F). **B**: the physical cartpole with control board used for experiments.

Table 1 provides the network architecture and training properties, which control the model configuration, including the number of layers, activation functions, optimizer, and other training settings. The training dataset will be described individually in each experiment.

Table 1: Neural Network Architecture and Training Properties

Architecture	8IN-64GRU-64GRU-1OUT
Activation / Loss / Optimizer	tanh / mse / adam
Input Features	cart position, velocity, pole angle, angular velocity, target pole angle (up / down), target cart position
Output Features	Motor force F
Learning Rate	10^{-3} - 10^{-7}, reduced by 0.5 with patience of 2
Epochs / Batch Size / Sequence Length	60 / 256 / 65

5. Experiment Results

The training process aimed to enable the GRU to learn an implicit representation of the varying parameters by observing their effect on state evolution over time.

5.1. Adaptive Control under Changing Physical Parameters

In this experiment, we trained the A-NC to handle variations in pole mass and length. The training dataset includes 5M samples (∼28 h) with parameters uniformly distributed within [0.05, 0.8] m for pole length and [0.015, 0.15] kg for pole mass. Validation data consists of 180k samples (1 h). The network was trained as described in Table 1. The A-NC was tested in simulation with fixed but unknown physical parameters chosen randomly from the range of the training dataset. These parameters were not disclosed to the RNN-controllers. We evaluated the control performance of each controller, based on their ability to swing-up and follow a target position. For this simulation experiment, we set the pole length to 15 cm and the pole mass to 50 g.

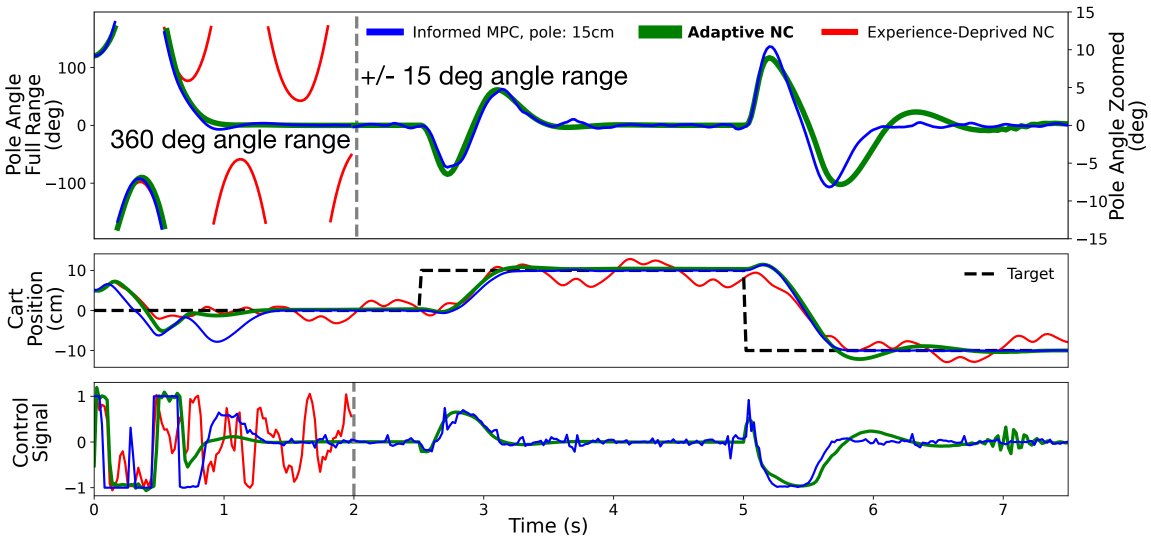

Figure 3: Comparison of control performance for the informed NMPC, A-NC, and ED-NC, showing the cart position (x), pole angle (θ), and control signal (u) over time.

Fig. 3 compares the control performance of the informed NMPC, adaptive A-NC, and experience-deprived ED-NC. As expected, the informed NMPC (blue) achieved precise control with minimal deviation from the desired trajectory. It successfully swung up the pole (as indicated by the zero angle on the angle plot in Fig. 3), balanced, and accurately tracked the target position. Similarly, the A-NC (green) closely followed the informed NMPC, demonstrating its ability to adapt to fixed but unknown physical parameters. In contrast, the ED-NC (red) only managed to roughly track the target cart position, but was unable to swing up and balance the pole.

For better insight, we evaluated the ability of the A-NC to mimic the control signals of the informed NMPC. In this simulation experiment we set the pole length to 5 cm and the pole mass to 15 g.

Fig. 4 shows the control signals produced during the experiment by the following controllers:

- **Informed MPC (blue):** The NMPC with full knowledge of the fixed pole parameters. As the NMPC served as teacher during GRU training, this is considered as optimal control strategy for this experiment.
- **Adaptive A-NC (green):** The A-NC recurrent neural network with internal memory, enabling it to implicitly adapt to the unknown parameters over time.
- **ED-NC (red):** The ED-NC recurrent neural network trained on inconsistent state histories generated using random PPs as described in Section 3.3. This DR-based approach forces the network to rely solely on the current state and target.

Fig. 4A shows that the uninformed A-NC control (green) closely matched the informed NMPC (blue), but the ED-NC (red) significantly deviated from it. Fig. 4B compares the NMPC, the ED-NC, and the “Average NMPC” (orange), which is the average behavior of the NMPC over PPs variation. It shows that ED-NC almost perfectly matches the Average NMPC. This observation highlights the limitations of DR techniques in handling dynamic PP variation.

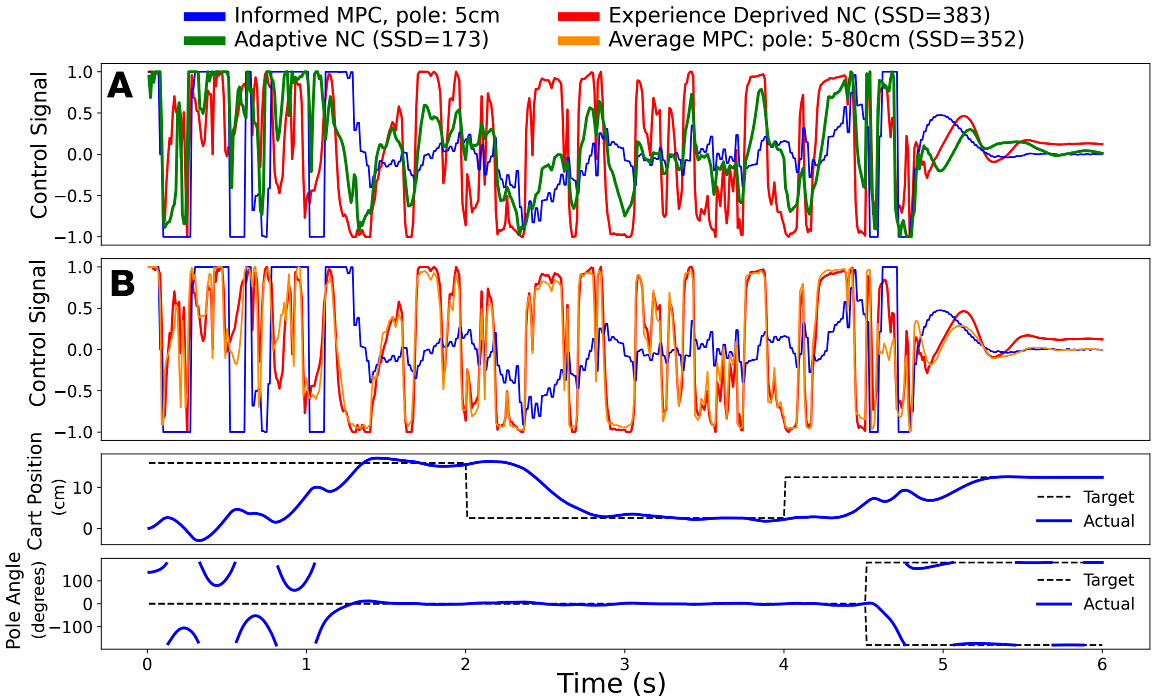

Figure 4: Control signals generated by the informed NMPC, adaptive GRU, and experience-deprived ED-NC during changes in pole length. The bottom two plots show the target cart position (χ), target pole equilibrium (ξ), and pole angle (θ). SSD is the sum of squared differences between the control and the NMPC control.

The experiment demonstrates the ability of the adaptive uninformed A-NC to approximate the control signals of the informed NMPC more closely than the experience-deprived ED-NC, highlighting the importance of memory in the A-NC.

5.2. Analysis of GRU Hidden States

To better understand the adaptive capabilities of the GRU, we analyzed its hidden states to investigate whether an explicit representation of the pole length can be retrieved. This analysis was conducted in two parts: applying PCA to the hidden states, and training an MLP to read out the parameters.

Principal Component Analysis of Hidden States

In the first part of the analysis, we performed PCA on the A-NC's hidden states to determine whether the primary dimensions of variation correlate with physical parameters. The first principal component (PCA1) was compared directly with the pole length to identify the relationships between the hidden state dynamics and the pole length.

Fig. 5A shows the A-NC PCA1 and the pole length over time. The strong alignment between these values suggests that the GRU implicitly encodes the pole length in its hidden states, even without explicit supervision. Fig. 5B shows that ED-NC PCA1 does not consistently encode pole length.

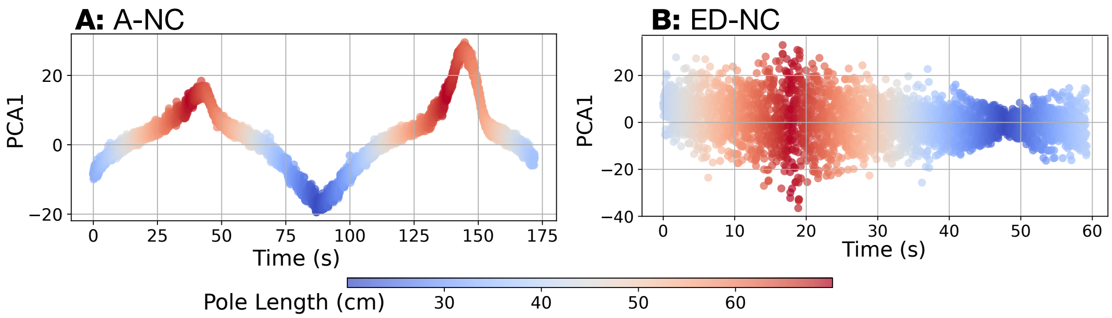

Figure 5: **A**: The first principal component (PCA1) of the A-NC's hidden states over time and varying pole length. **B**: The same PCA analysis applied to the ED-NC.

MLP READOUT OF PRIVILEGED PARAMETERS

In the second part of the analysis, we trained an MLP to retrieve the physical parameters from the adaptive GRU's hidden states. The MLP consisted of 2 layers of 128 units with $\tanh$ activation functions. It was trained on a dataset containing 1.4M samples, where the pole length was systematically varied while the pole mass was fixed at 50 g.

Fig. 6 plots retrieved pole length over time compared to the ground truth length. The network successfully retrieved the pole length, demonstrating that the GRU's hidden states encode it. Arrows indicate times where the cartpole swing-up provides dynamics to enable this retrieval.

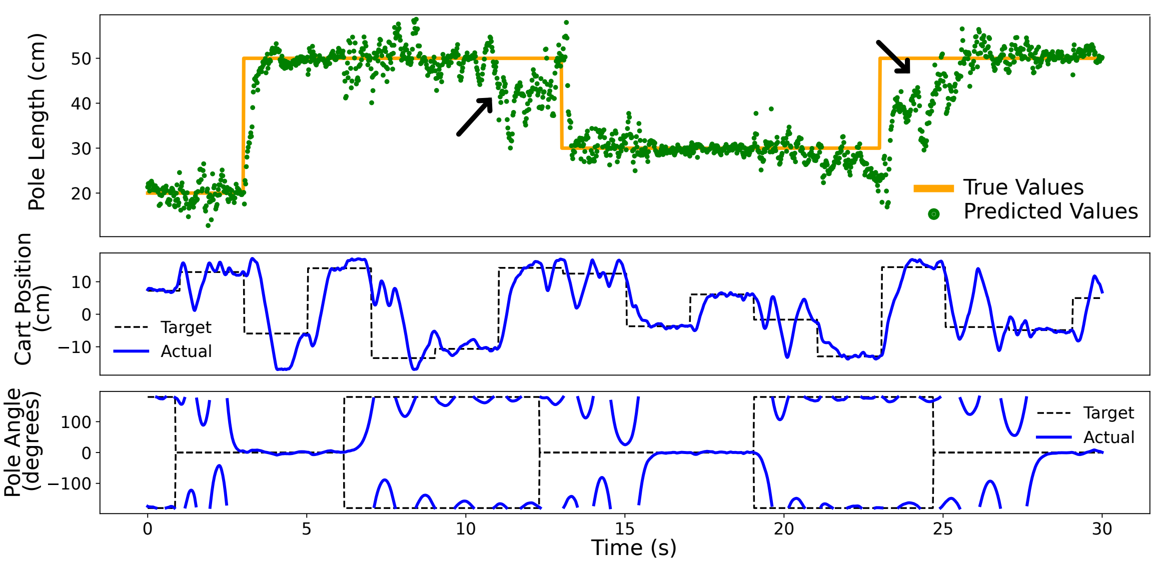

Figure 6: MLP retrieved pole length (green) from the GRU's hidden states (Section 5.2) compared to the ground truth value (yellow).

5.3. Pole Angle Offset Adaptation on Physical Cartpole

In this experiment, we trained the A-NC with the properties described in Table 1 across various angle sensor offsets. During the data collection, the offset is varied with a uniform distribution over the complete $360°$ range. The dataset consisted of 135k samples (2700 s) and the validation set contained 23k samples (450 s). The angle offset was randomly set after every sequence during training. This dataset allows the network to learn to generalize across a wide range of sensor offsets.

The video of this experiment[2] shows that even with a sensor offset of $> 90°$, the A-NC can still balance the cartpole.

6. Conclusion

This work demonstrates the effectiveness of adaptive RNN-based controllers for dynamically adjusting control policies in response to changes in system parameters and environmental conditions. By leveraging past trajectories, the proposed adaptive GRU implicitly estimates physical parameters and adapts its control policy without requiring direct measurement or explicit parameter estimation. This capability allows the GRU to achieve control performance comparable to fully informed NMPC while being computationally more efficient.

We proposed novel strategies for data collection and preprocessing for training a robust A-NC. These strategies include introducing parameter disturbances, incorporating sensor noise and quantization effects, and introducing noise to the history to ensure that the network develops persistent internal representations of system parameters.

Our experiments on the cartpole system, both in simulation and on physical hardware, validate the effectiveness of the A-NC. The controller consistently outperformed experience-deprived networks, demonstrating its ability to dynamically adapt to varying parameters such as pole mass, pole length, and angle sensor offsets. Furthermore, our analysis of the GRU's hidden states revealed that it encodes implicit representations of physical parameters, confirming its capacity to adaptively infer relevant dynamics.

This work demonstrates adaptive RNN controllers as a viable alternative to NMPC, especially in scenarios with limited parameter measurements or computational power. Future work could explore applications to more complex systems and alternative architectures for improved adaptability.

Acknowledgments

We thank Nikita Voinov for valuable discussions and contributions during the exploration of the topics discussed in this work. The authors thank the reviewers for their valuable comments. This work was supported by reserve funding, the Samsung Global Research Neuromorphic Processor Project (`https://sensors.ini.ch/research/areas/neural-hardware`), and by the Swiss National Science Foundation grant "VIPS: ultra-low power VIsual Perception System" (`https://data.snf.ch/grants/grant/181010`).

2. `https://drive.google.com/file/d/1bnbxJL_1MCLTRwKlTkHMJ8Ei0QMZr1wR/view?usp=sharing`

References

Anuradha M Annaswamy and Alexander L Fradkov. A historical perspective of adaptive control and learning. *Annu. Rev. Control*, 52:18–41, January 2021. ISSN 1367-5788,1872-9088. doi: 10.1016/j.arcontrol.2021.10.014. URL http://dx.doi.org/10.1016/j.arcontrol.2021.10.014.

Vittorio Caggiano, Guillaume Durandau, Cheryl Wang, Chun Kwang Tan, Pierre Schumacher, Huwawei Wang, Alberto Chiappa, Alessandro Marin Vargas, Alexander Mathis, Jungnam Park, Jungdam Won, Gunwoo Park, Beomsoo Shin, Minseung Kim, Seungbum Koo, Zhuo Yang, Wei Dang, Heng Cai, Jianfei Song, Seungmoon Song, Massimo Sartori, and Vikash Kumar. MyoChallenge 2023 towards human-level dexterity and agility. https://openreview.net/pdf?id=3A84lx1JFh, 2024. Accessed: 2024-11-27.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning (CoRL)*, 2019.

Alberto Silvio Chiappa, Alessandro Marin Vargas, and Alexander Mathis. DMAP: a Distributed Morphological Attention Policy for learning to locomote with a changing body. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 37214–37227. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/f0fae49cdfab57c41c30c9b0244093cb-Paper-Conference.pdf.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv [cs.AI]*, 8 November 2016. URL http://arxiv.org/abs/1611.02779.

Chelsea Finn, P Abbeel, and S Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *ICML*, volume 70, pages 1126–1135. PMLR, March 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

Colin Green. Equations of Motion for the Cart and Pole Control Task, January 2020. URL https://sharpneat.sourceforge.io/research/cart-pole/cart-pole-equations.html.

Drew Hanover, Philipp Foehn, Sihao Sun, Elia Kaufmann, and Davide Scaramuzza. Performance, precision, and payloads: Adaptive nonlinear MPC for quadrotors. *IEEE Robot. Autom. Lett.*, 7 (2):690–697, April 2022. ISSN 2377-3766,2377-3774. doi: 10.1109/lra.2021.3131690. URL http://arxiv.org/abs/2109.04210.

Frederik Heetmeyer, Marcin Paluch, Diego Bolliger, Florian Bolli, Xiang Deng, Ennio Filicicchia, and Tobi Delbruck. RPGD: A small-batch parallel gradient descent optimizer with explorative resampling for nonlinear model predictive control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 29 May 2023. doi: 10.1109/icra48891.2023.10161233. URL http://dx.doi.org/10.1109/icra48891.2023.10161233.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *Artificial Neural Networks — ICANN 2001*, Lecture notes in computer science, pages 87–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. doi: 10.1007/3-540-44668-0_13. URL http://dx.doi.org/10.1007/3-540-44668-0_13.

Elia Kaufmann, Antonio Loquercio, Rene Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Deep drone acrobatics. In *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, 12 July 2020. doi: 10.15607/rss.2020.xvi.040. URL http://dx.doi.org/10.15607/rss.2020.xvi.040.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and System XVII (RSS 2021)*. RSS Foundation, July 2021. doi: 10.15607/RSS.2021.XVII.011. URL https://www.roboticsproceedings.org/rss17/p011.html.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.*, 5(47):eabc5986, 21 October 2020. doi: 10.1126/scirobotics.abc5986. URL http://dx.doi.org/10.1126/scirobotics.abc5986.

Anand Subramoney, Guillaume Bellec, Franz Scherr, Robert Legenstein, and Wolfgang Maass. Fast learning without synaptic plasticity in spiking neural networks. *Sci. Rep.*, 14(1):8557, 12 April 2024. doi: 10.1038/s41598-024-55769-0. URL https://www.nature.com/articles/s41598-024-55769-0.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv [cs.LG]*, 17 November 2016. URL http://arxiv.org/abs/1611.05763.

Henry Philip Whitaker. An adaptive system for control of the dynamics performance of aircraft and spacecraft. *Inst. Aeronautical Sciences, Paper 59-100*, 1959.