Deep Neural Networks as Add-on Modules for High-Accuracy Impromptu Trajectory Tracking

SiQi Zhou, Mohamed K. Helwa and Angela P. Schoellig
With contributions from Qiyang Li, Jingxing Qian, Zining Zhu and Xuchan Bao

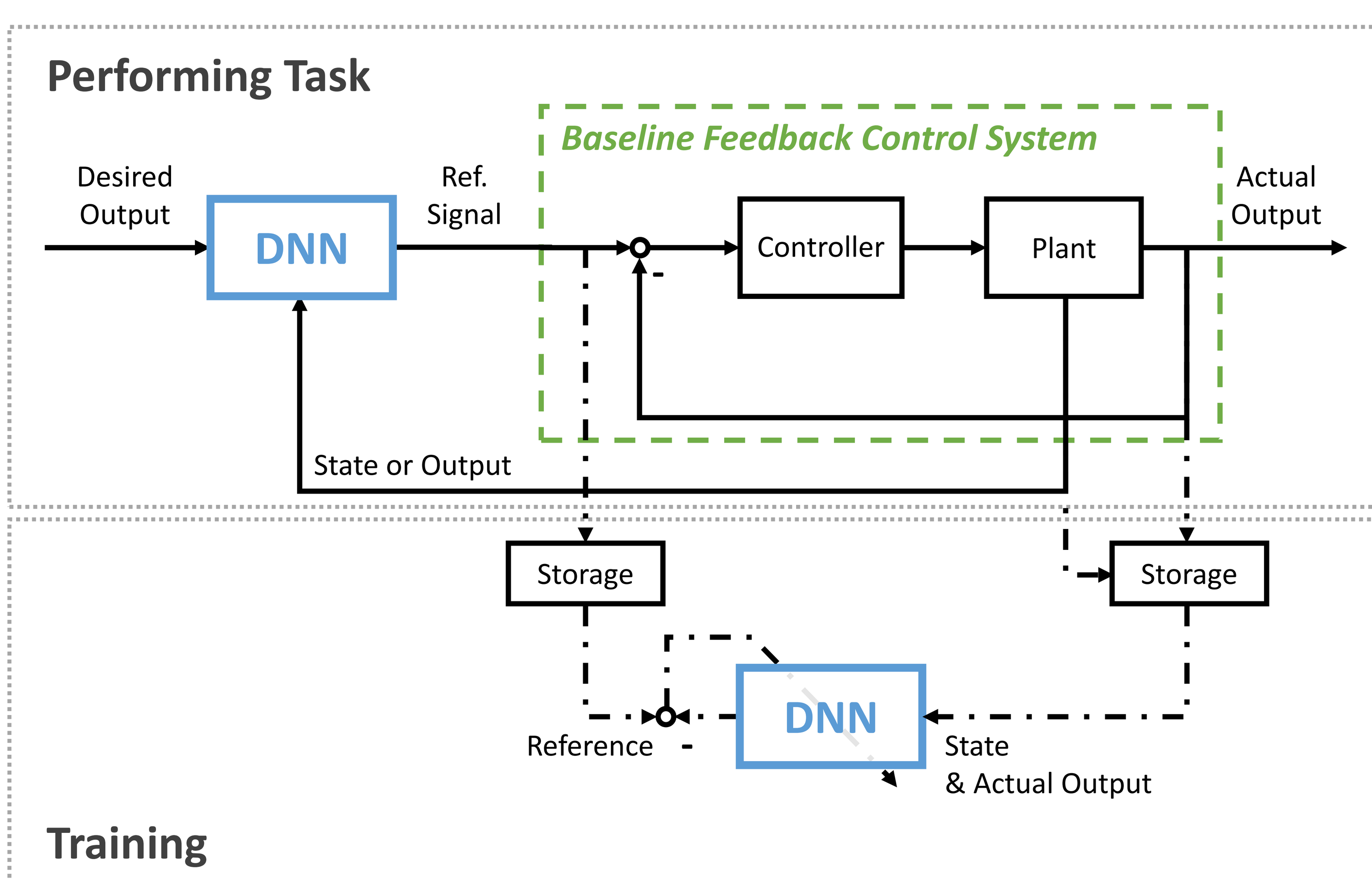
DYNAMIC SYSTEMS LAB

Background & Motivation

- Goal: to achieve **high-accuracy** tracking for **arbitrary, feasible trajectories** in **one shot** (i.e., impromptu tracking)
- Challenges:
  - Uncertainties due to **unmodeled dynamics** and **unknown effects** from the environment
  - Tracking **trajectories not known *a priori***
- Techniques from the literature and limitations:
  - **PID controller** for tracking – parameters are difficult to tune for arbitrary trajectories
  - **Adaptive control** for parameter uncertainties – past tracking experience is not efficiently utilized
  - **Iterative Learning Control** for high-accuracy tracking – experience does not generalize well to untrained tasks

Methodology

- Idea: **Learning of an add-on module** to enhance the impromptu tracking capabilities of a **black-box control system**
- Control architecture and training:



Theoretical Results

- The ideal control law that should be modeled by the **DNN module** is the **output equation of the inverse dynamics** of the **baseline feedback control system**.
- Generic **feedback control system**:

$$\underbrace{x(t+1)}_{\text{State}} = f(x(t)) + g(x(t))\,\underbrace{u(t)}_{\text{Reference Signal}}$$
$$\underbrace{y(t)}_{\text{Actual Output}} = h(x(t))$$

- Necessary inputs and output of the **DNN module**:

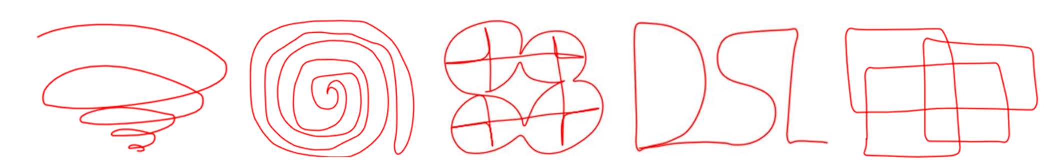
$$\mathcal{I} = \{x(t), \underbrace{y_d}_{\text{Desired Output}}(t+\underbrace{r}_{\text{Relative Degree}})\} \rightarrow \mathcal{O} = \{u(t)\}$$

- Necessary conditions for effectively applying the approach: The **baseline system** must have *(i)* a **well-defined relative degree** and *(ii)* **stable inverse dynamics**.
- Additional insight: The inverse learning of the **DNN module** can be made more efficient if the **baseline system** achieves **zero steady-state error for step inputs**.

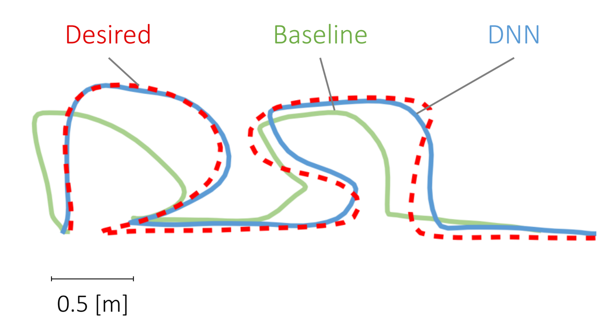
Experimental Results

- In [1], the proposed approach was tested on **quadrotor vehicles** using 30 hand-drawn trajectories. An average of **43% tracking error reduction** was achieved.

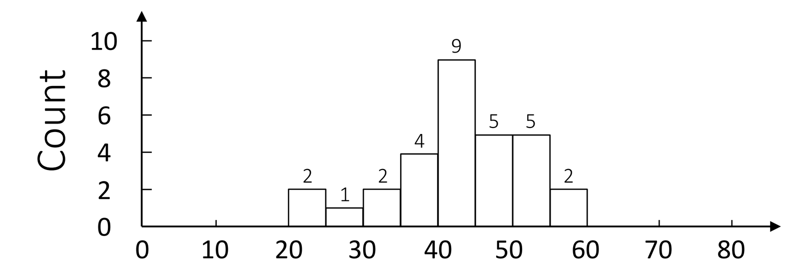
Samples of Hand-Drawn Trajectories



Result of One Test Trajectory
(Error Reduction 42%)



Error Reduction Distribution
(Mean 43%)



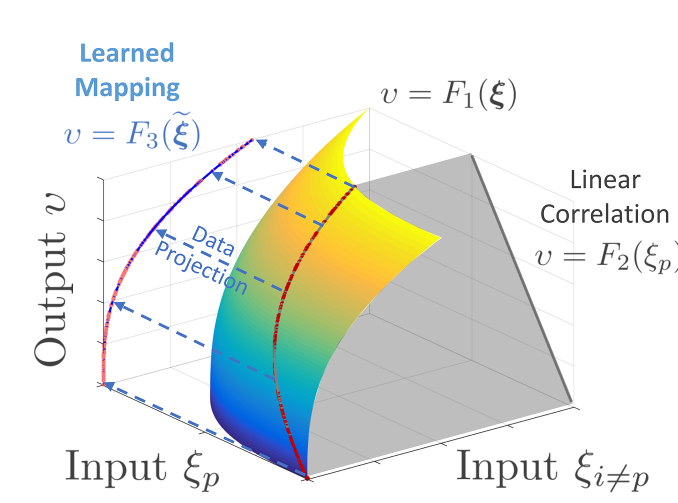
- In [1], inputs of the **DNN module** were determined through trial-and-error. In [2], training the **DNN** based on the theoretical results led to similar performance but with the **DNN input dimension reduced by 2/3**.

Error Reduction on 5 Trajectories

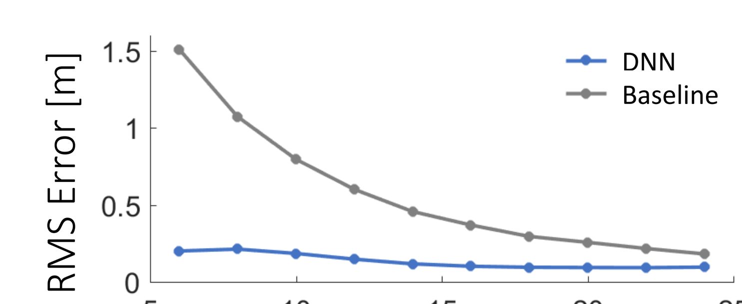
Trajectory Index	DNN with 36 inputs [1]	DNN with 12 inputs [2]
1	55.6%	52.4%
2	61.4%	48.4%
3	39.7%	42.0%
4	43.9%	46.6%
5	26.6%	36.9%
Average	45.5%	45.2%

Non-Minimum Phase Systems

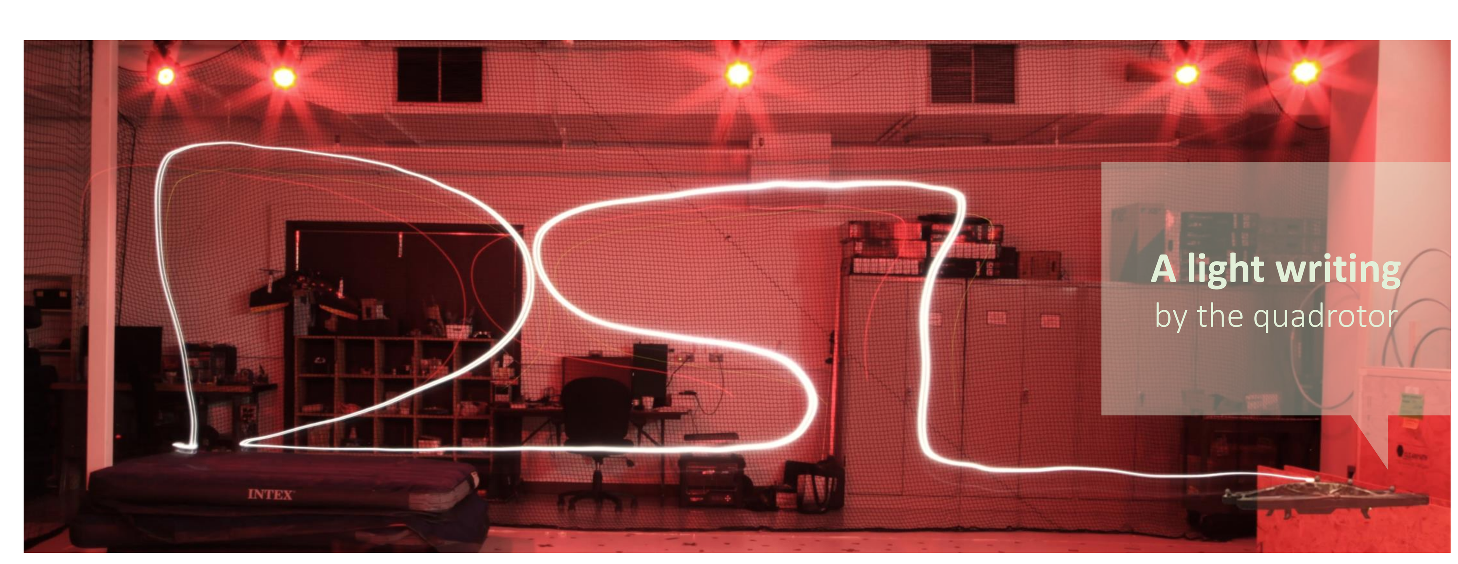
- This approach cannot be directly applied to **non-minimum phase systems** (i.e., systems with unstable inverse dynamics).
- Modification: **Learning an approximate inverse** of the **baseline system** by removing certain inputs from the **DNN module** — **less information leads to better performance** [3].



Tracking Error Comparison
(Inverted Pendulum on Cart Simulation)



Long Exposure



More Information

[1] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa and A. P. Schoellig, "Deep Neural Networks for Improved, Impromptu Trajectory Tracking of Quadrotors," **ICRA 2017**.
Implementation details of the quadrotor experiments including DNN structure and training.

[2] S. Zhou, M. K. Helwa and A. P. Schoellig, "Design of Deep Neural Networks as Add-on Blocks for Improving Impromptu Trajectory Tracking," **CDC 2017**.
Guidelines for designing the DNN module – a general framework derived from control theory.

[3] ———, "An Inversion-Based Learning Approach for Improving Impromptu Trajectory Tracking of Robots with Non-Minimum Phase Dynamics," Submitted to R-AL and ICRA 2018.
Application of the inversion-based learning approach to non-minimum phase systems.