Design of Deep Neural Networks as Add-on Blocks for Improving Impromptu Trajectory Tracking

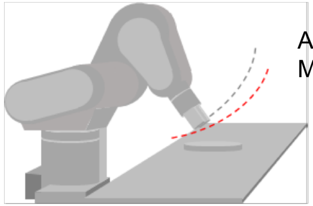
Conference on Decision and Control (CDC) 2017

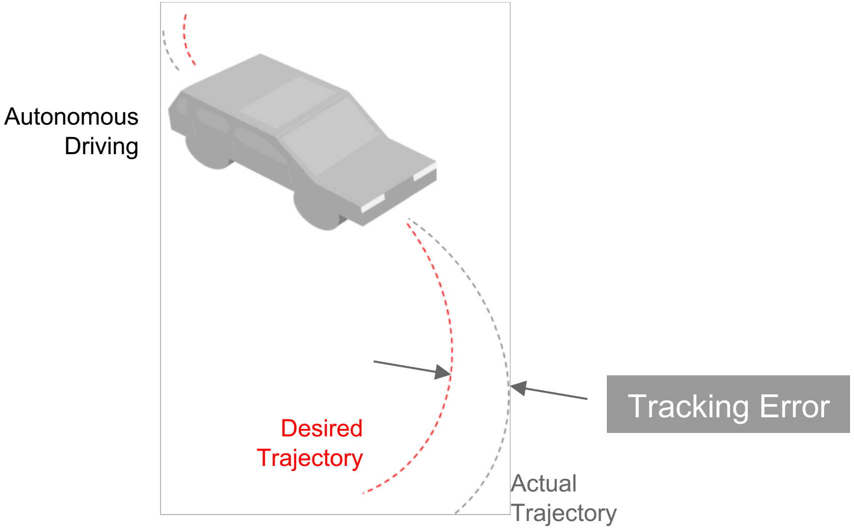
SiQi Zhou, Mohamed K. Helwa, and Angela P. Schoellig
Dynamic Systems Lab | University of Toronto Institute for Aerospace Studies

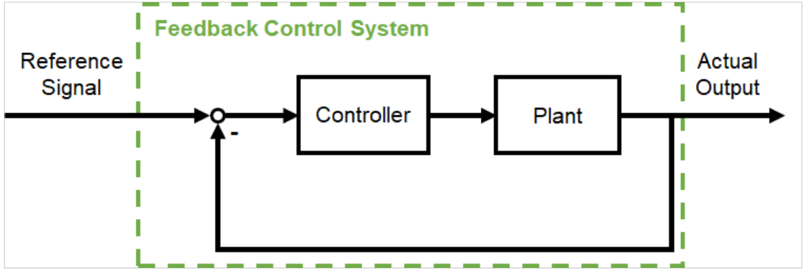
Institute for Aerospace Studies
UNIVERSITY OF TORONTO

DYNAMIC
SYSTEMS LAB

Designing control systems for high-accuracy tracking is challenging

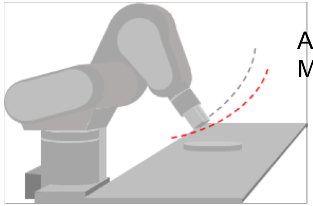






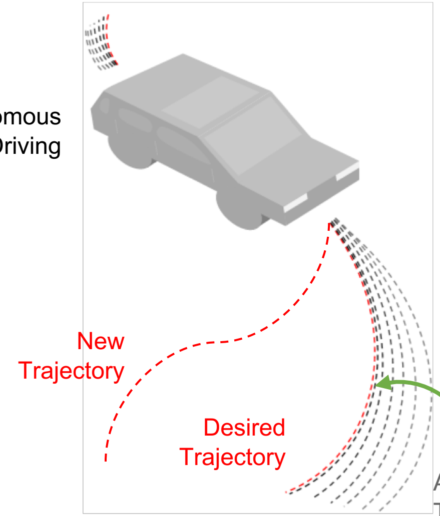
Perfect tracking cannot be achieved for arbitrary trajectories.

Designing control systems for high-accuracy tracking is challenging



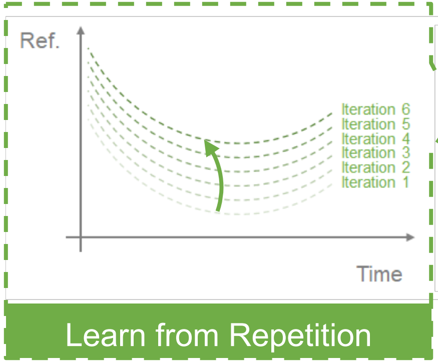
Automated Manufacturing

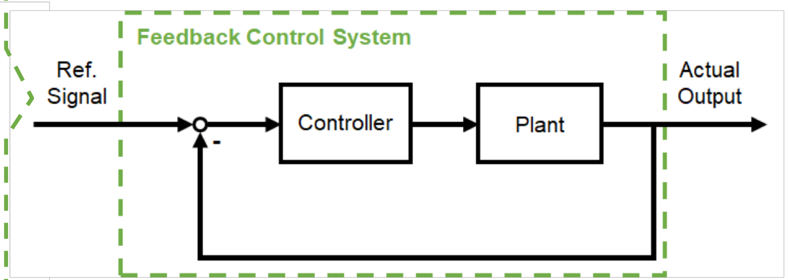
Autonomous Driving



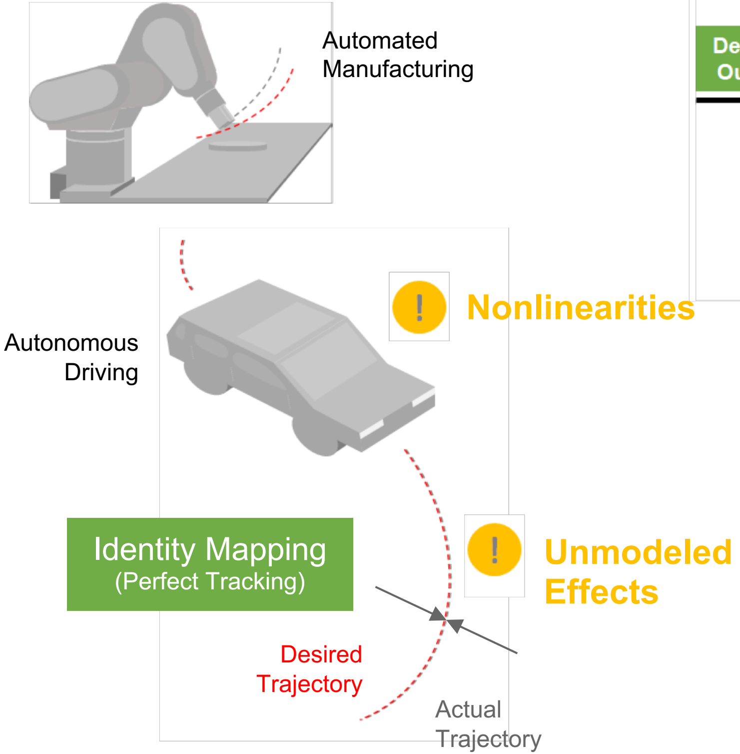
Actual Trajectory

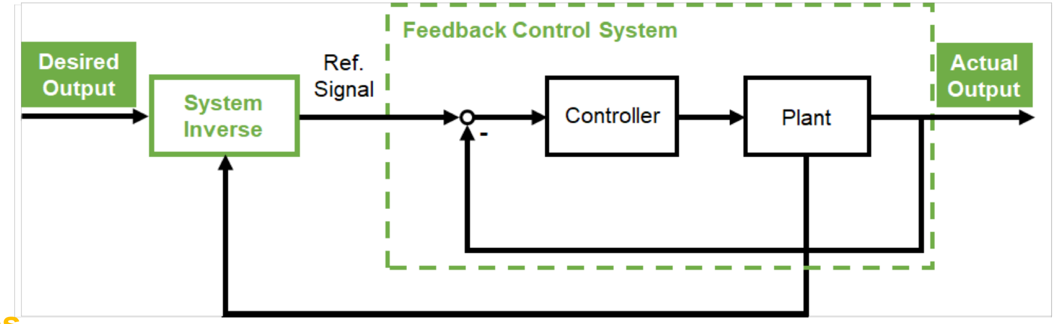
Improve Iteratively





Designing control systems for high-accuracy tracking is challenging



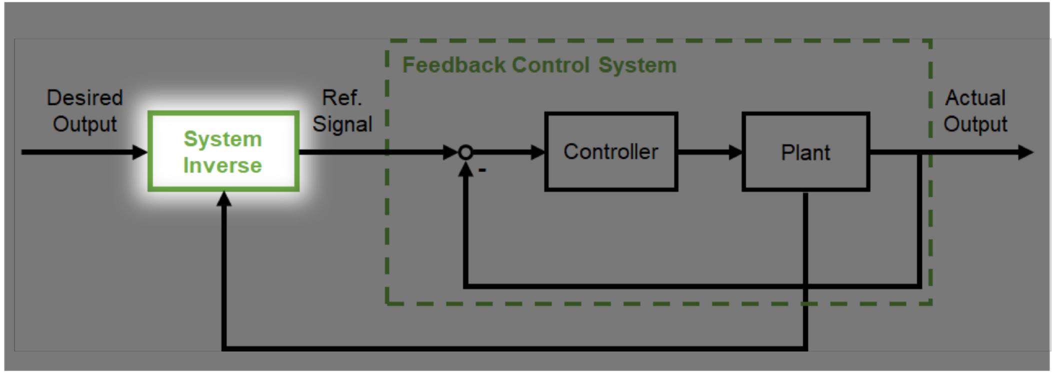


- Obtaining a sufficiently accurate inverse model is difficult in practice.
- Applying to non-minimum phase systems (i.e., systems with unstable inverse dynamics) is not trivial.

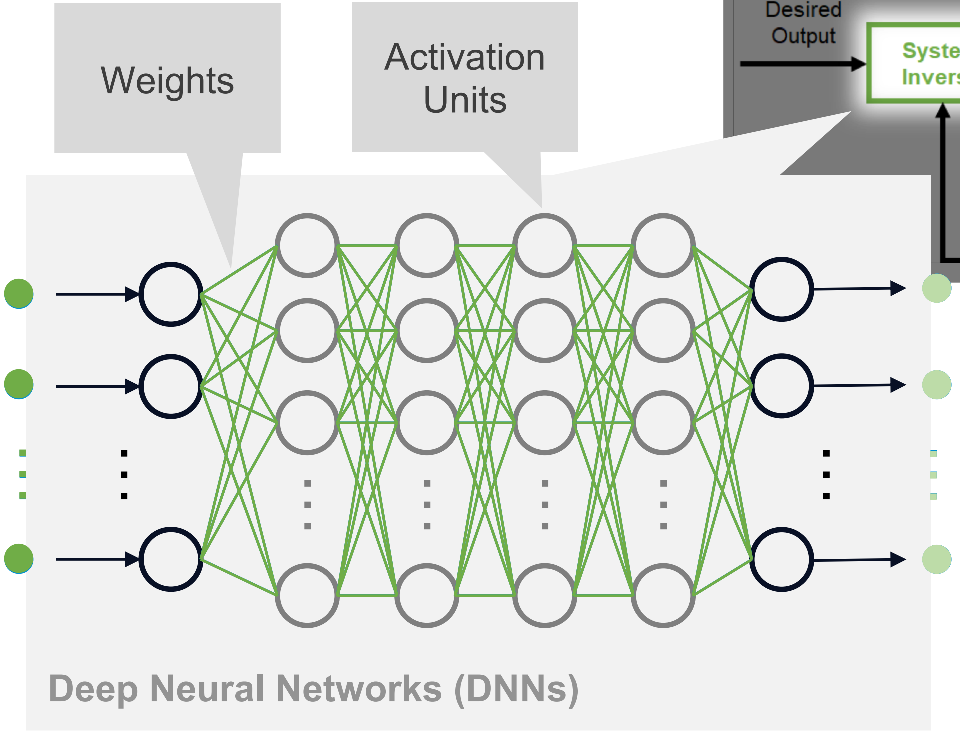
Learning add-on blocks to enhance ‘black-box’ control systems

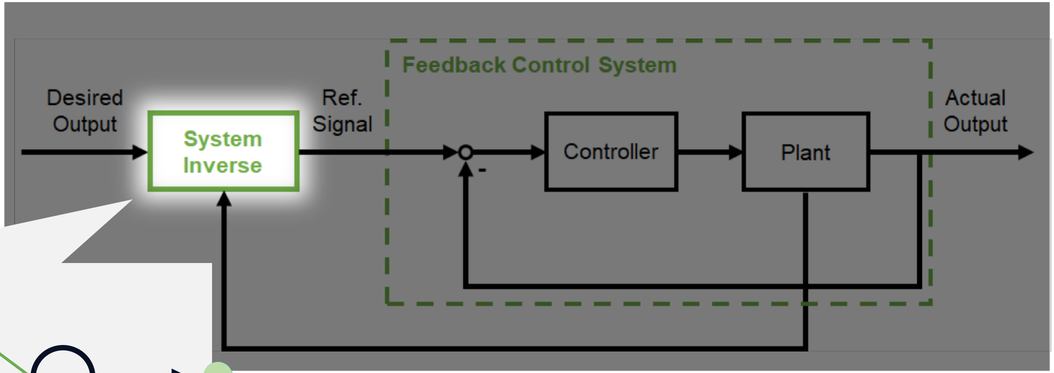
DYNAMIC
SYSTEMS LAB

Learn inverse of closed-loop systems from input-output data to achieve high-accuracy impromptu tracking (i.e., tracking arbitrary trajectories in one shot)



Deep Neural Networks (DNNs) as the learning technique





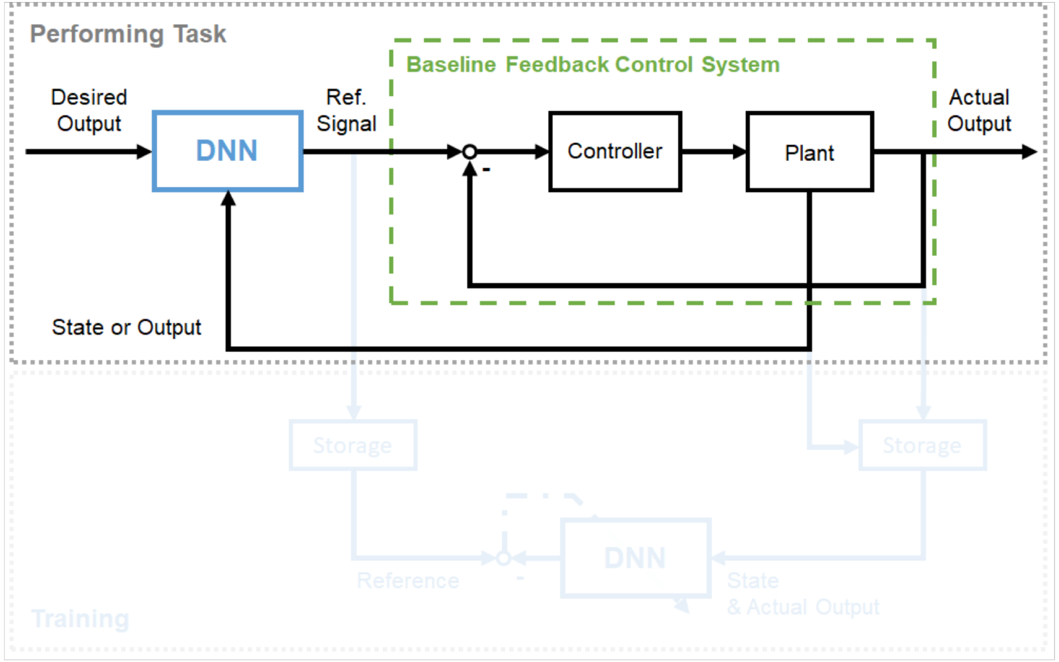
- Single hidden layer networks are universal function approximators.

- Representativeness of network grows as the number of layers grows deeper.

DNN as add-on blocks to enhance 'black-box' control systems

DYNAMIC
SYSTEMS LAB

Learn inverse of closed-loop systems from input-output data to achieve high-accuracy impromptu tracking (i.e., tracking arbitrary trajectories in one shot)



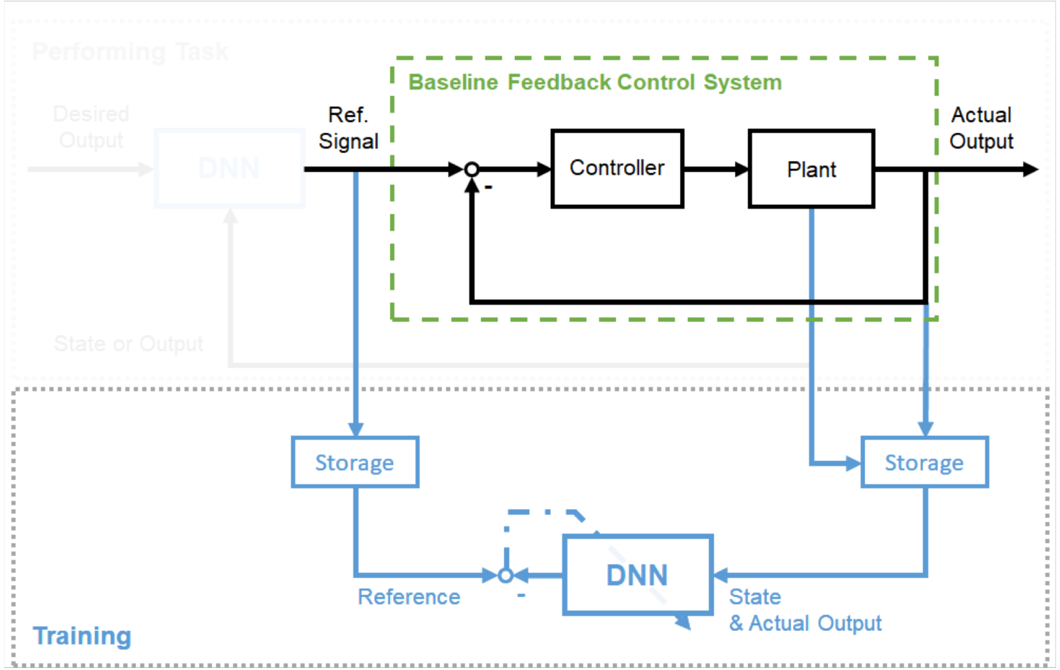
DNN as add-on blocks to enhance 'black-box' control systems

DYNAMIC SYSTEMS LAB

Learn inverse of closed-loop systems from input-output data to achieve high-accuracy impromptu tracking (i.e., tracking arbitrary trajectories in one shot)

Overview

- Training: a DNN module is trained with **reversed input-output data** of the baseline system.

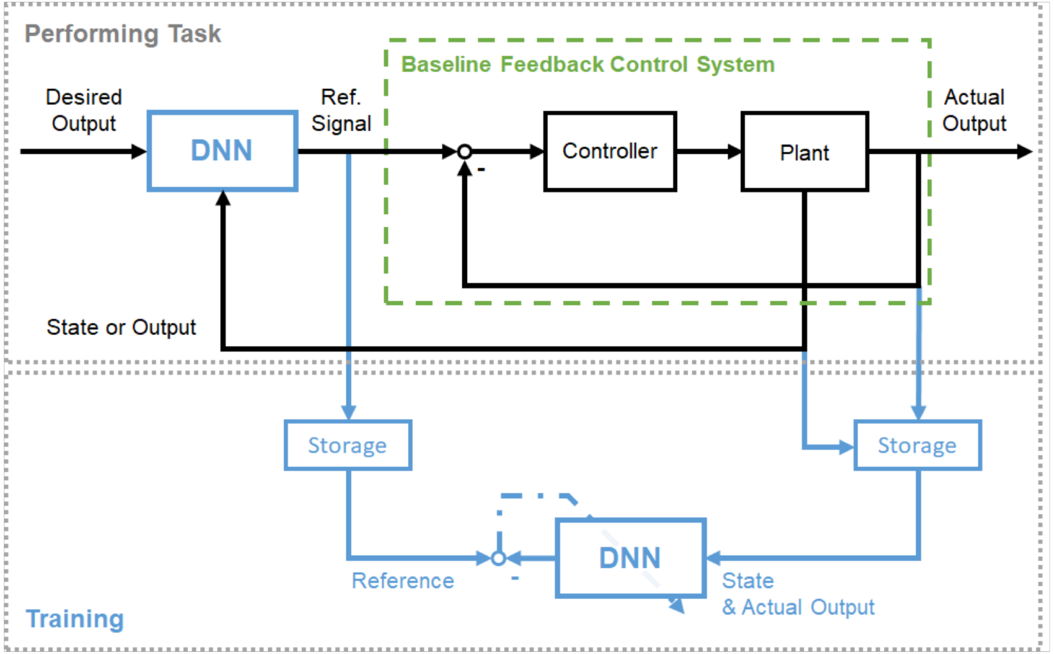


DNN as add-on blocks to enhance 'black-box' control systems

Learn inverse of closed-loop systems from input-output data to achieve high-accuracy impromptu tracking (i.e., tracking arbitrary trajectories in one shot)

Overview

- Training: a DNN module is trained with **reversed input-output data** of the baseline system.
- Performing task: the DNN add-on module **adjusts the reference** signal sent to the baseline system.



The DNN add-on module reduces tracking error by 40%-50%

Objective

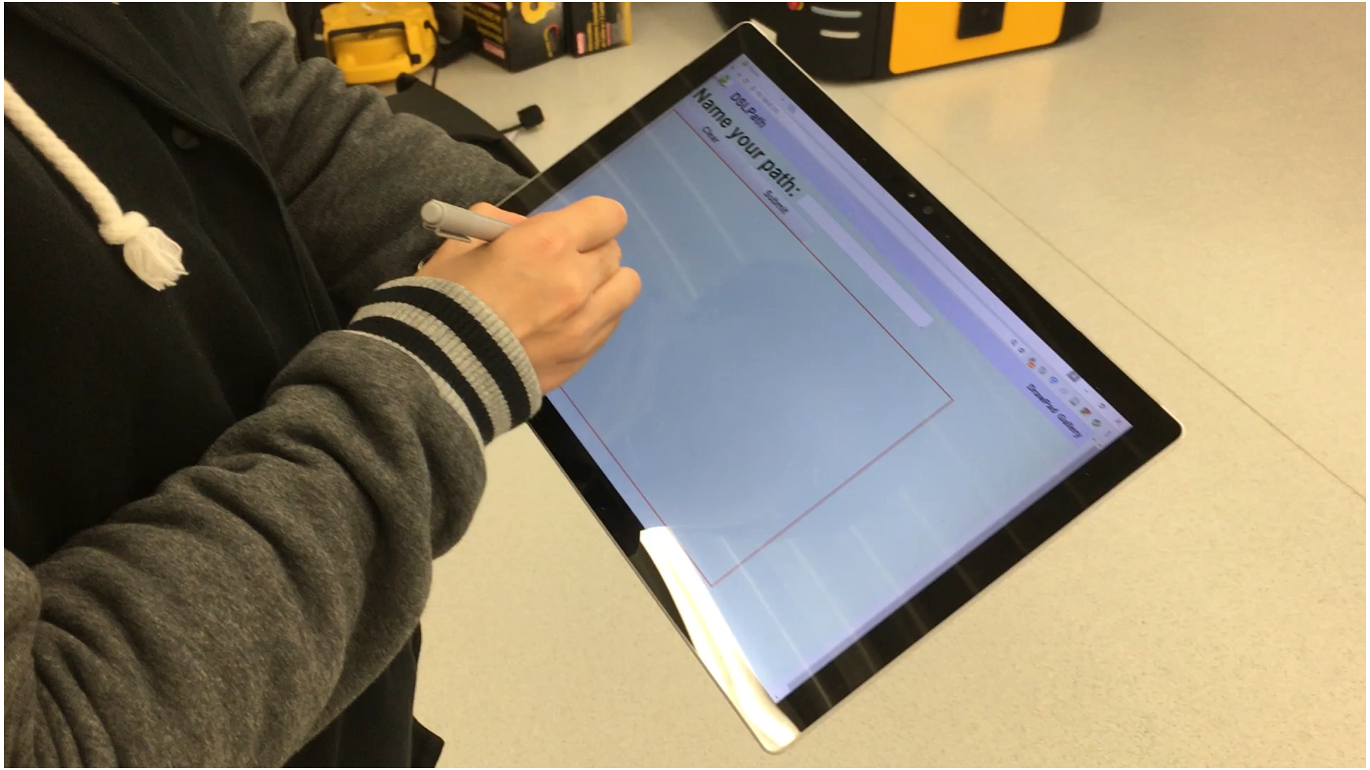
To track arbitrary hand-drawn trajectory with high-accuracy impromptu

Procedure

1. Collect data
2. Train network
3. Track hand-drawn trajectories

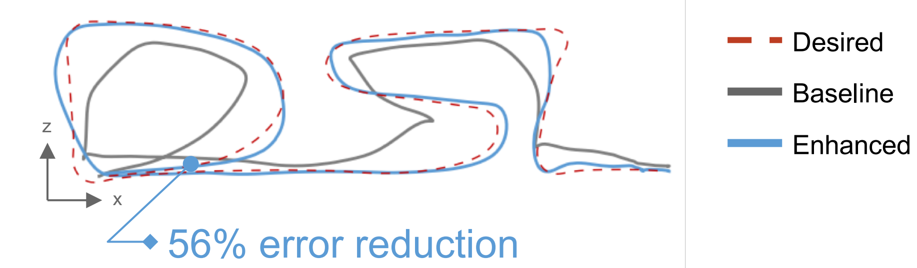
Fly-as-You-Draw Project

Q. Li, J. Qian, Z. Zhu, X. Bao,
M. K. Helwa, and A. P. Schoellig
"*Deep Neural Networks for Improved, Impromptu Trajectory Tracking of Quadrotor*"
(ICRA 2017)



The DNN add-on module reduces tracking error by 40%-50%

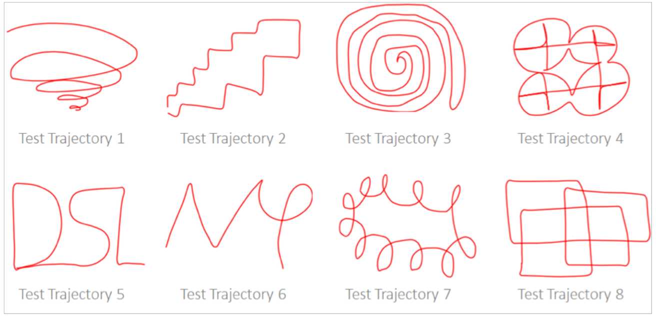
Quadrotor Path in x-z Plane



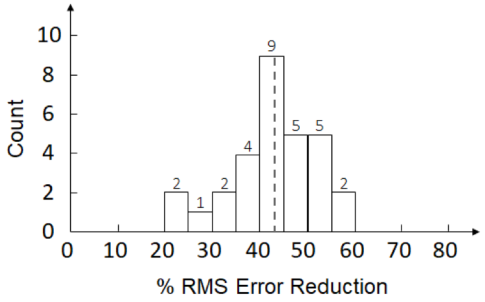
- 56% error reduction was achieved with only 20 min of training on pure sinusoidal trajectories.
- On average of 30 hand-drawn trajectories, 43% error reduction was achieved.
- The dependent inputs of the DNN module were determined through experimental trial-and-error.

From **ICRA 2017**

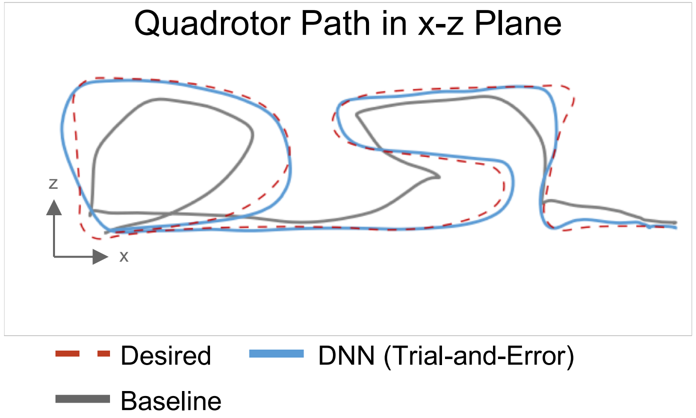
Examples of Untrained Test Trajectories

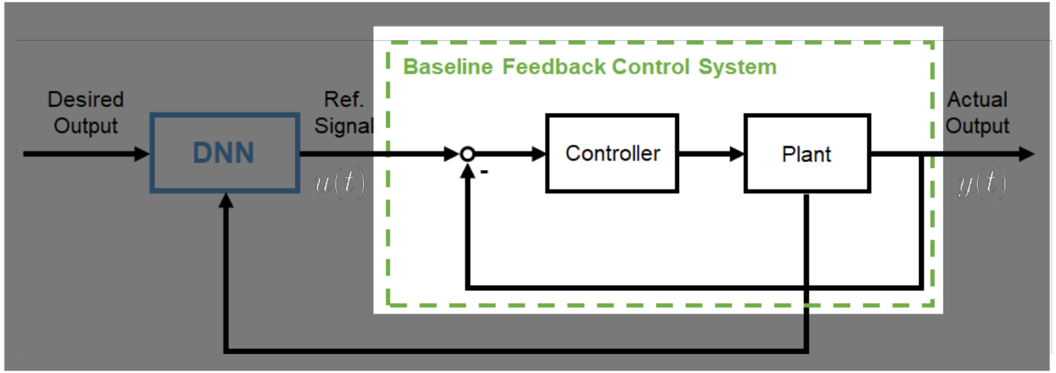


% Error Reduction Distribution



Control theory guides us towards more efficient training





Baseline System Dynamics

Linear

$$x(t+1) = Ax(t) + bu(t)$$
$$y(t) = cx(t)$$

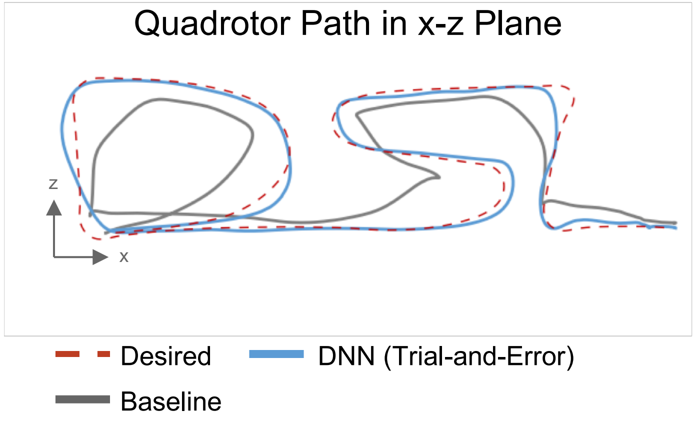
Nonlinear

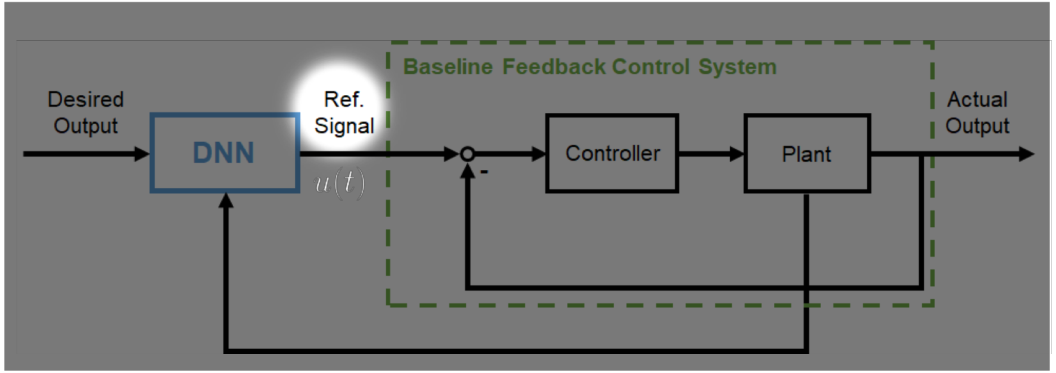
$$x(t+1) = f\left(x(t)\right) + g\left(x(t)\right)u(t)$$
$$y(t) = h\left(x(t)\right)$$

Platform-Independent Formulation

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)**

Control theory guides us towards more efficient training





Baseline System Dynamics

Linear

$$x(t+1) = Ax(t) + bu(t)$$
$$y(t) = cx(t)$$

Nonlinear

$$x(t+1) = f\left(x(t)\right) + g\left(x(t)\right) u(t)$$
$$y(t) = h\left(x(t)\right)$$

Ideal Control Law

Output Equation of the System's Inverse Dynamics

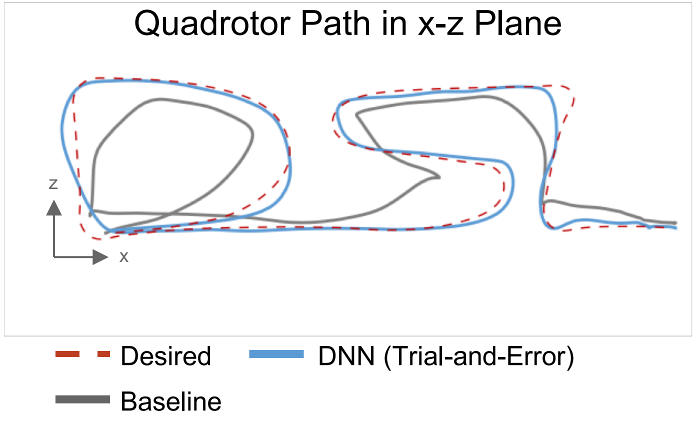
$$u(t) = \frac{1}{cA^{r-1}b}\left(-cA^r x(t) + y_d(t+r)\right)$$

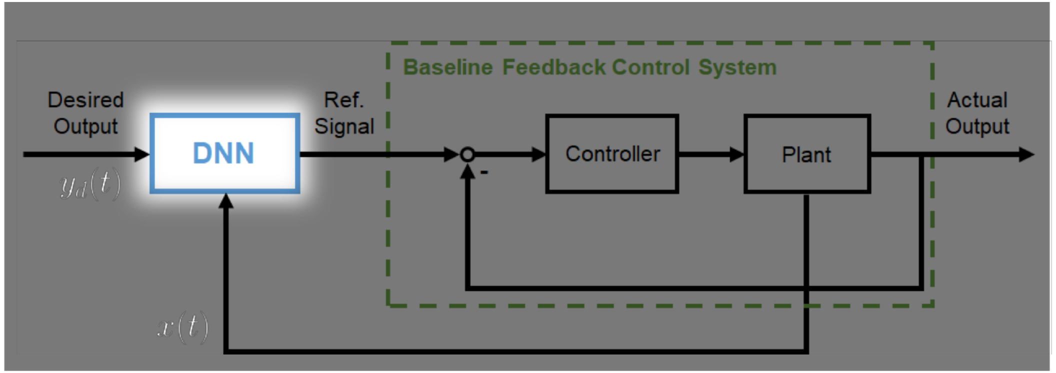
$$u(t) = F\left(x(t), y_d(t+r)\right)$$

Platform-Independent Formulation

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)**

Control theory guides us towards more efficient training





Necessary Inputs

$$\{x(t), y_d(t+r)\}$$

Platform-Independent Formulation

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)**

Baseline System Dynamics

Linear

$$x(t+1) = Ax(t) + bu(t)$$
$$y(t) = cx(t)$$

Nonlinear

$$x(t+1) = f(x(t)) + g(x(t))\,u(t)$$
$$y(t) = h(x(t))$$

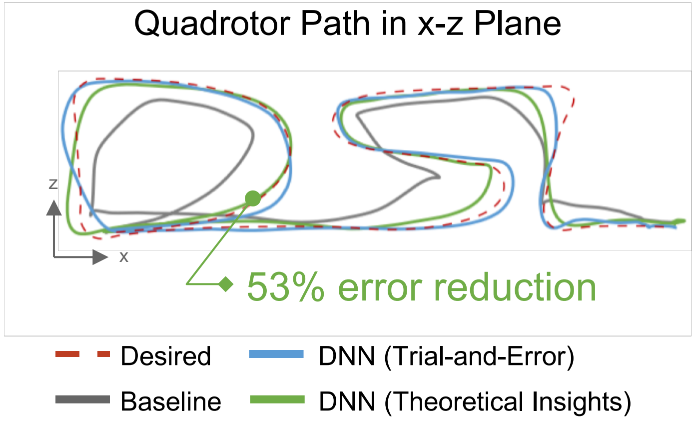
Ideal Control Law

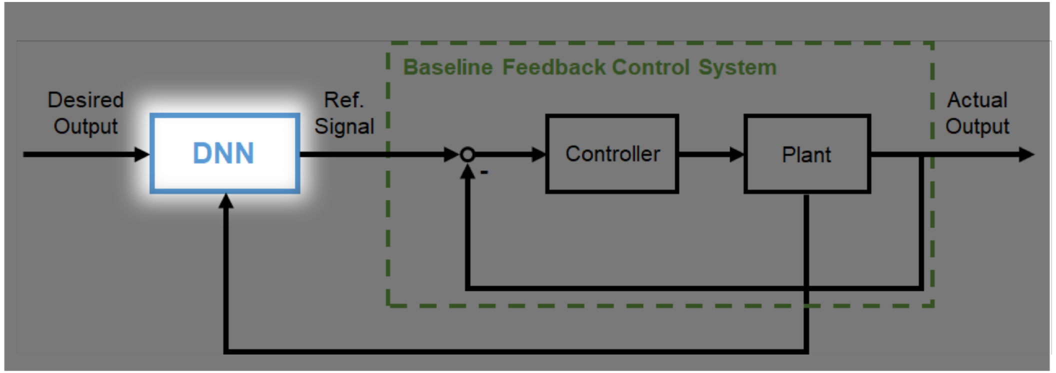
Output Equation of the System's Inverse Dynamics

$$u(t) = \frac{1}{cA^{r-1}b}\left(-cA^{r}x(t) + y_d(t+r)\right)$$

$$u(t) = F\left(x(t), y_d(t+r)\right)$$

Control theory guides us towards more efficient training





Necessary Inputs

$\{x(t), y_d(t+r)\}$

Relative Degree r

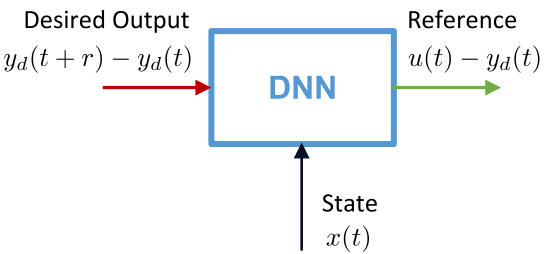
- Inherent delay of the baseline system, or the number of time steps between applying reference input and first seeing effects in output
- Can be experimentally identified through simple step responses

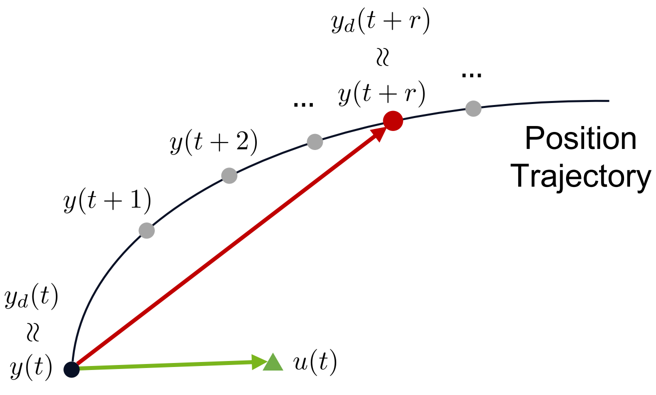
Platform-Independent Formulation

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)**

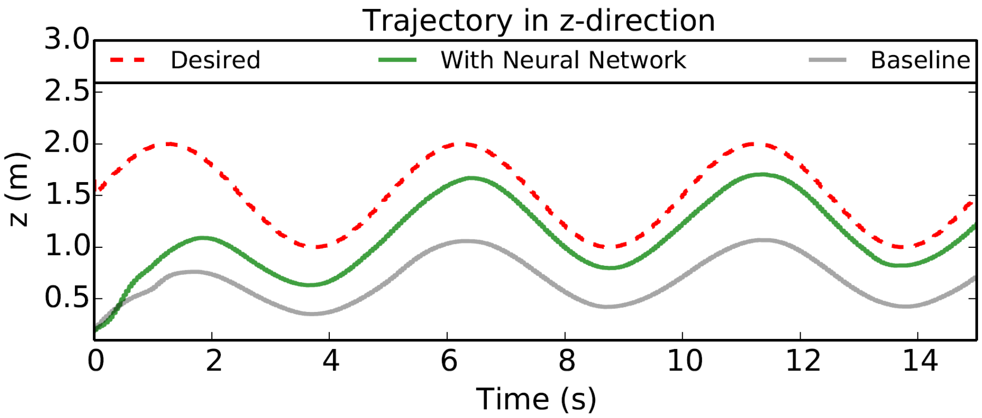
Similar performance (53% tracking error reduction) with DNN input dimension reduced by 2/3

Condition for more data-efficient training

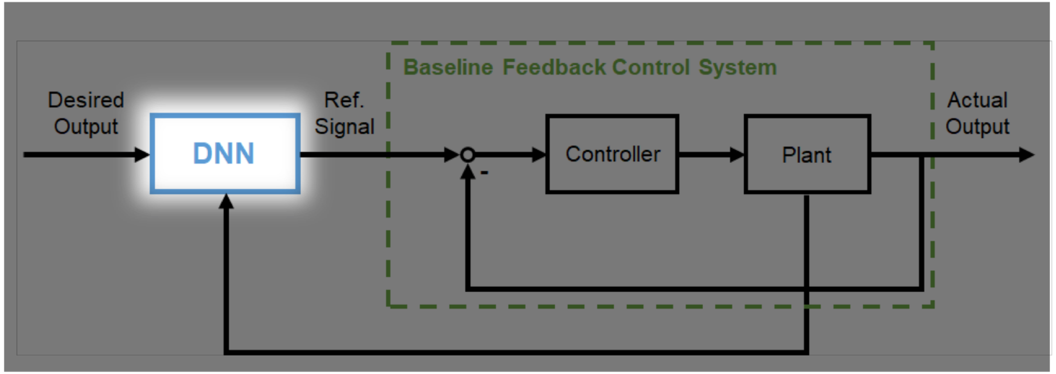




- Difference learning scheme: In previous work, for the quadrotor tracking problem, relative positions w.r.t. the desired trajectory are used to simplify the DNN training.
- Condition: the baseline black-box system achieves zero steady state error for step inputs.
- If not achieved, the underlying function becomes one-to-many, which cannot be learned by the DNN.



Summary of insights



Insight 1:

a. In order to achieve unity mapping from the desired to the actual output, the DNN module can be formularized as the output equation of the baseline system's inverse dynamics.

b. Due to the association with the inverse dynamics, the efficacy of the proposed approach relies on two necessary conditions (1) the system has a well-defined relative degree and (2) the system has stable zero dynamics.

Summary of insights

Insight 2: In order to achieve unity mapping from desired output to actual output,

a. based on the state-space formulation, the input features should be selected as

$$\{x(t), y_d(t+r)\}$$

can be determined from simple step-response experiments

b. based on the transfer-function formulation (for linear systems), the input features can be alternatively selected as

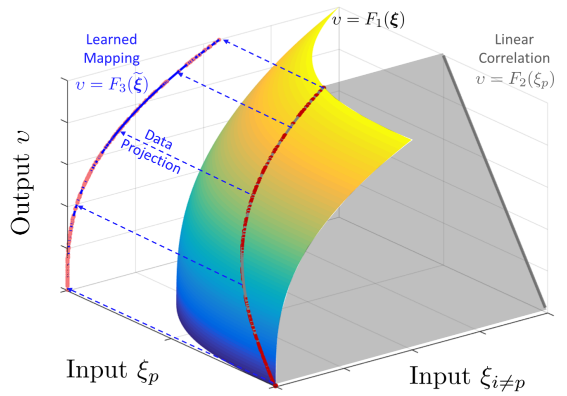
$$\{y_d(t-n+r), \ldots, y_d(t+r-1), y_d(t+r), u(t-n+r), \ldots, u(t-2), u(t-1)\}$$

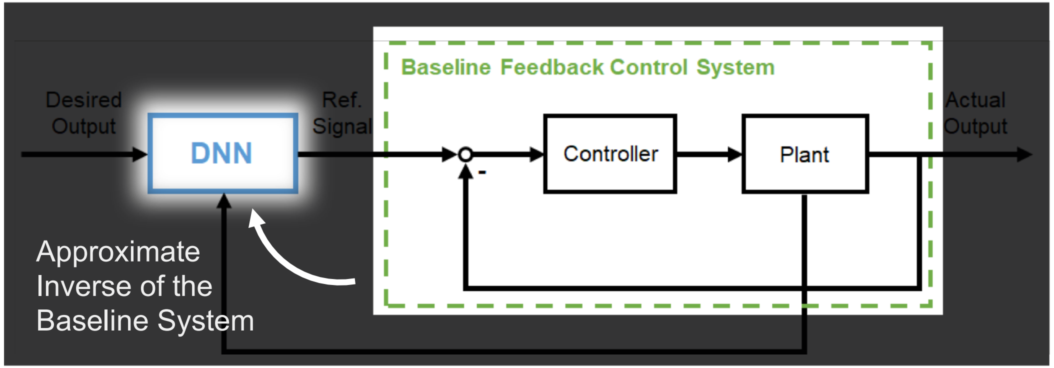
independent of state

Insight 3: The applicability of the data-efficient difference learning scheme relies on the condition that the baseline system achieves zero steady state error for step inputs.

Direct application to non-minimum phase systems is not safe

DYNAMIC SYSTEMS LAB





- Straightforward application does not work for non-minimum phase systems (i.e., systems with unstable inverse dynamics)
- Learning stable inverse approximations through removing inputs from the DNN module
- Compromise exactness for stability

Adaptation to Non-Minimum Phase Systems

S. Zhou, M. K. Helwa, and A. P. Schoellig
"*An Inversion-Based Learning Approach for Improving Impromptu Trajectory Tracking of Robots with Non-Minimum Phase Dynamics*"
(Submitted to RA-L and ICRA 2018)

Direct application to non-minimum phase systems is not safe

DYNAMIC
SYSTEMS LAB

Inverted Pendulum Experiment

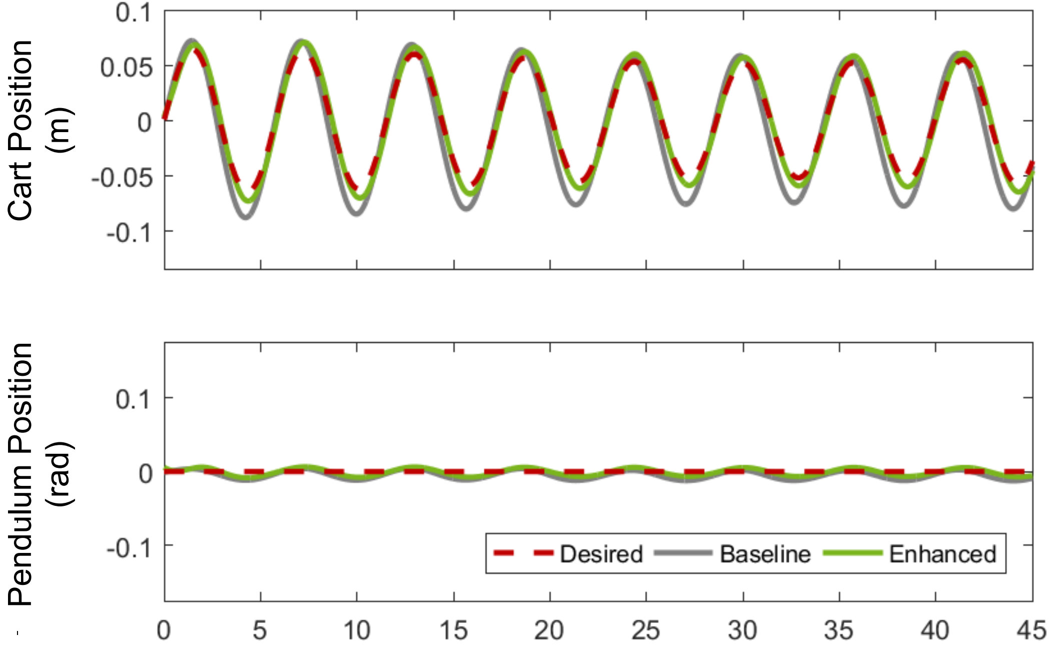
(Image from Quanser)

Adaptation to Non-Minimum Phase Systems

S. Zhou, M. K. Helwa, and A. P. Schoellig

"*An Inversion-Based Learning Approach for Improving Impromptu Trajectory Tracking of Robots with Non-Minimum Phase Dynamics*"

(Submitted to RA-L and ICRA 2018)



Less information leads to better performance

Summary of Contributions

First Practical Implementation

Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig. ICRA 2017

- Proposed **DNN as add-on block approach** for enhancing black-box tracking control systems
- Successfully tested on quadrotor vehicles for **tracking arbitrary hand-drawn trajectories**

Current Work

S. Zhou, M. K. Helwa, and A. P. Schoellig. CDC 2017

- Provided **platform-independent formulation** of the proposed DNN-enhanced control architecture
- Proposed **efficient input selection** of the DNN add-on module for enhancing black-box systems
- Identified **necessary conditions** for the proposed approach to be effective

Follow-up Work

S. Zhou, M. K. Helwa, and A. P. Schoellig. Submitted to RA-L and ICRA 2018

- Proposed an **approximate inverse learning** approach to extend the DNN-enhanced architecture to non-minimum phase systems

Neural networks are effective for improving tracking performance of black-box control systems; control insights are important for safe and efficient network design.

DYNAMIC SYSTEMS LAB

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

DYNAMIC
SYSTEMS LAB

Thank you!

Drone Light Writing
generated with fly-as-you-draw app

Institute for Aerospace Studies
UNIVERSITY OF TORONTO