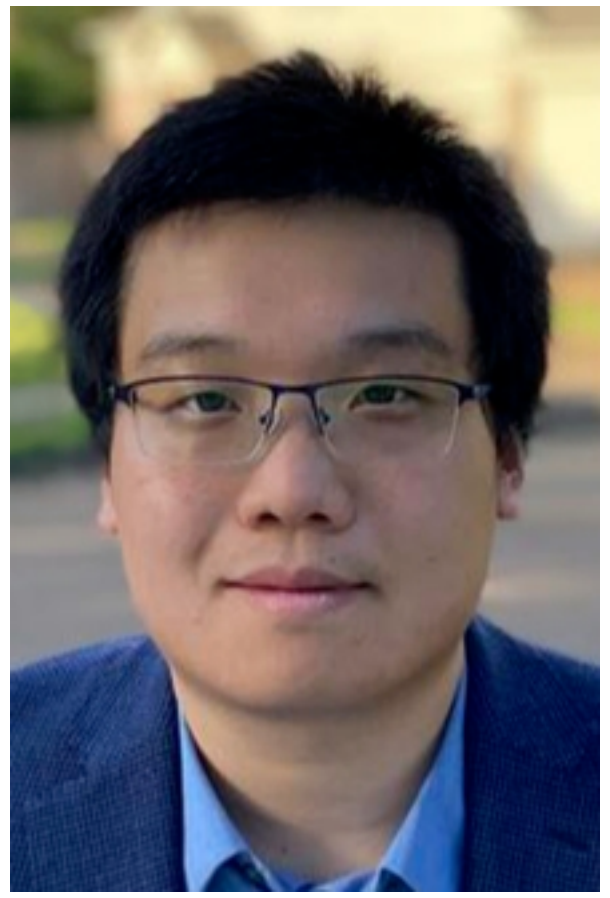
Learning to Control DC Motor for Micromobility in Real Time with Reinforcement Learning

Bibek Poudel, Thomas Watson and Weizi Li



IEEE ITSC 2022

THE UNIVERSITY OF MEMPHIS

Socio-Technical Autonomous Resilient Systems

Outline

- Introduction
- Methodology
- Experiments & Results
- Conclusion & Future Work

Introduction

Introduction

- Autonomous micromobility is on the rise

Image: Micromobility products from Segway, Nuro

Introduction

- Autonomous micromobility is on the rise
- DC motor is a key component

Image: Micromobility products from Segway, Nuro

Introduction

- Control (position, velocity) of a DC motor is non-trivial

Introduction

- Control (position, velocity) of a DC motor is non-trivial
- Non-linearities and uncertainties
    - Temperature-resistance relationship
    - Road friction, load

Introduction

- Control (position, velocity) of a DC motor is non-trivial
- Non-linearities and uncertainties
    - Temperature-resistance relationship
    - Road friction, load
- Need for a control strategy

Introduction

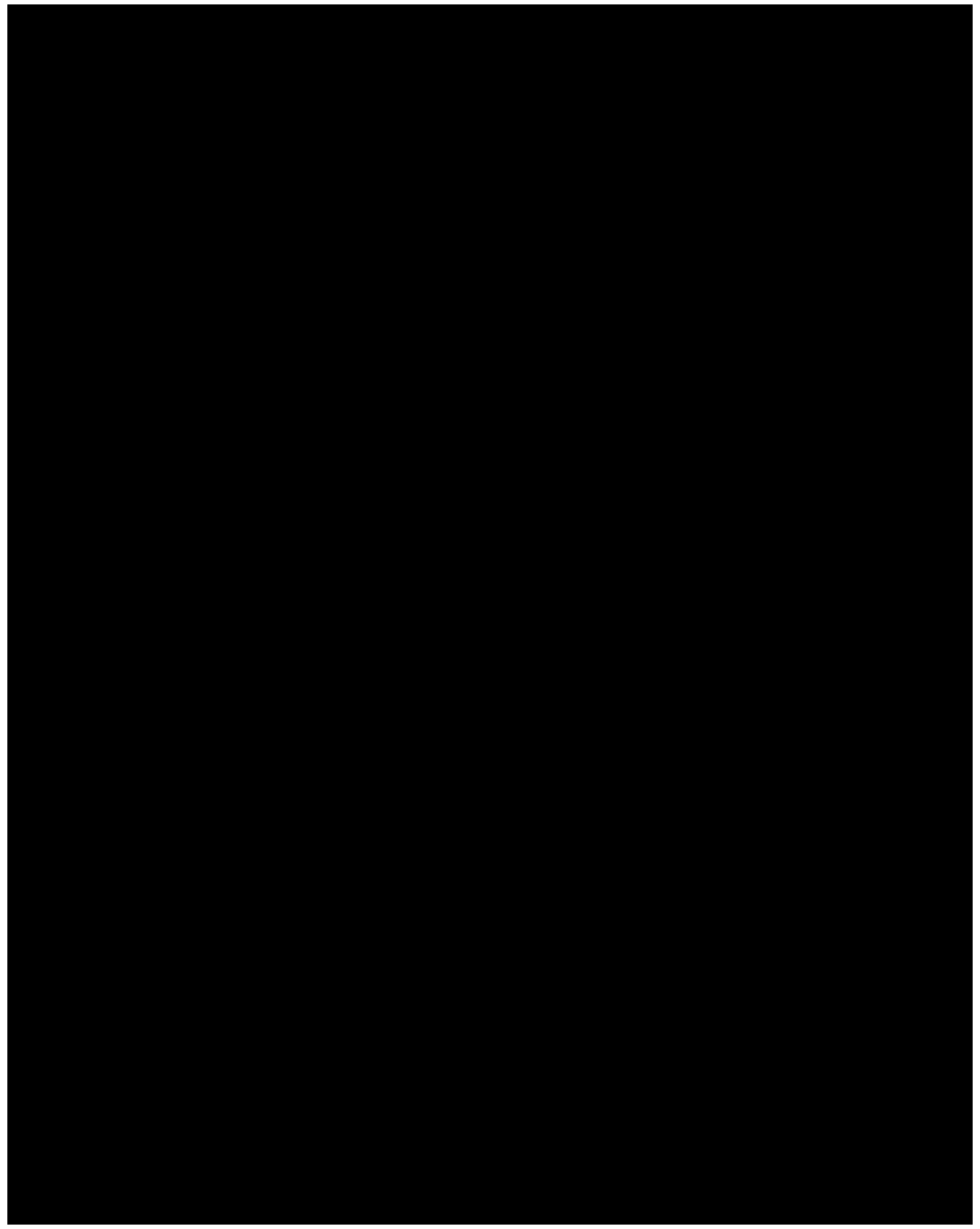
Image: Golf cart physical system

Introduction

Image: Golf cart physical system

Introduction

Image: Golf cart physical system



Video: Steering wheel rotation

Control Problem

- Task
  - Arbitrary initialization → go to center
  - Minimum Time Control Problem

Control Problem

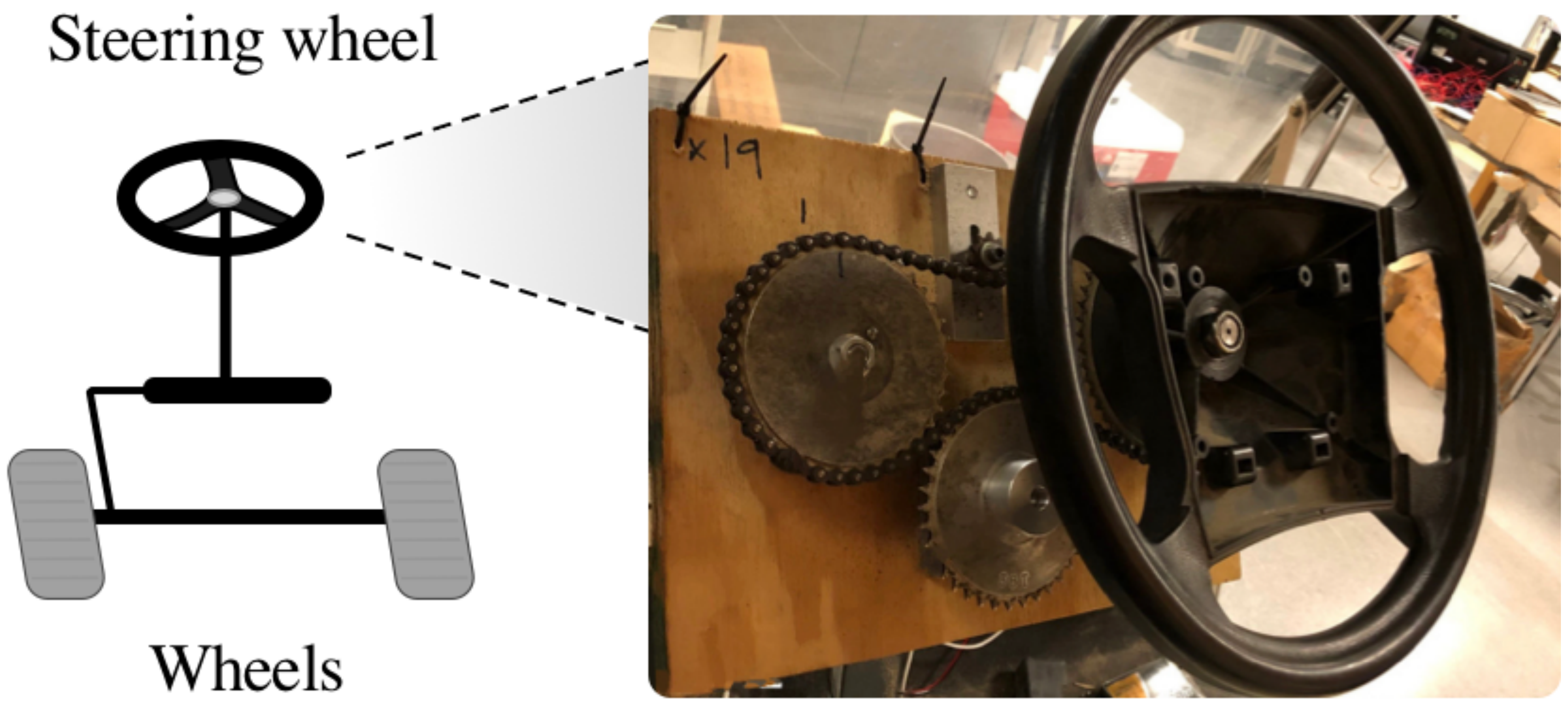
- Traditional techniques
    - Bang-bang
    - Proportional Integral Derivative (PID)
    - Model Predictive Control (MPC)

Control Problem

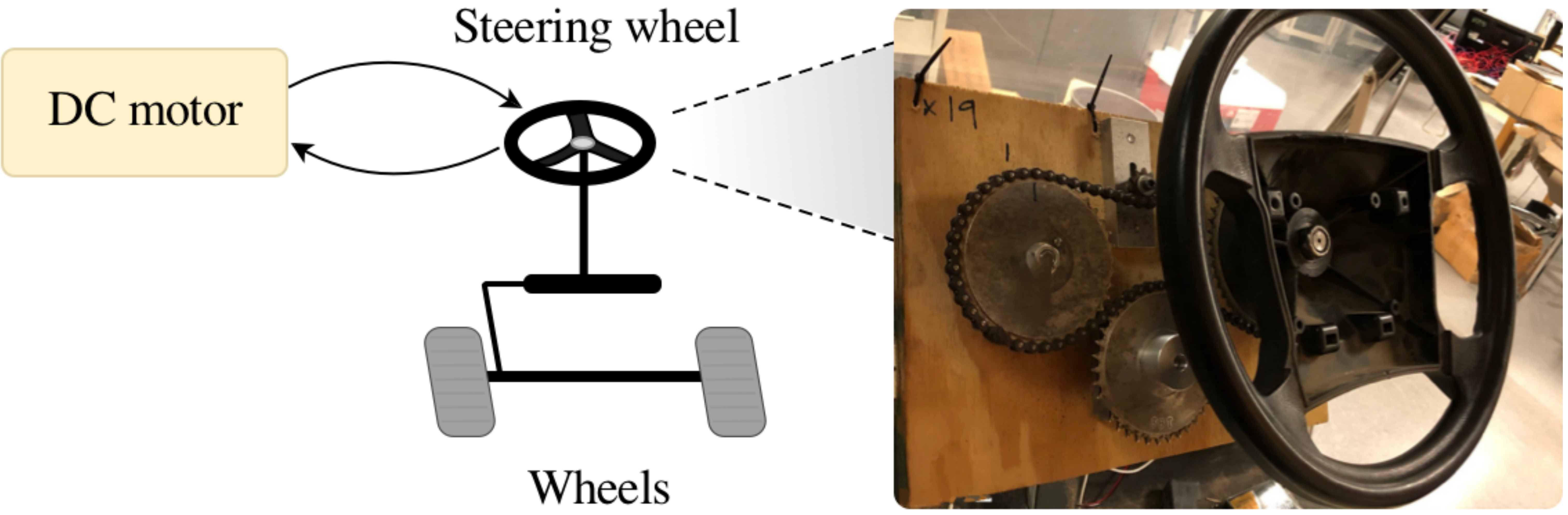
- Traditional techniques
  - Bang-bang
  - Proportional Integral Derivative (PID)
  - Model Predictive Control (MPC)
- Our approach
  - Model-free Reinforcement Learning
  - Neural Fitted Q (NFQ) Algorithm

Methodology

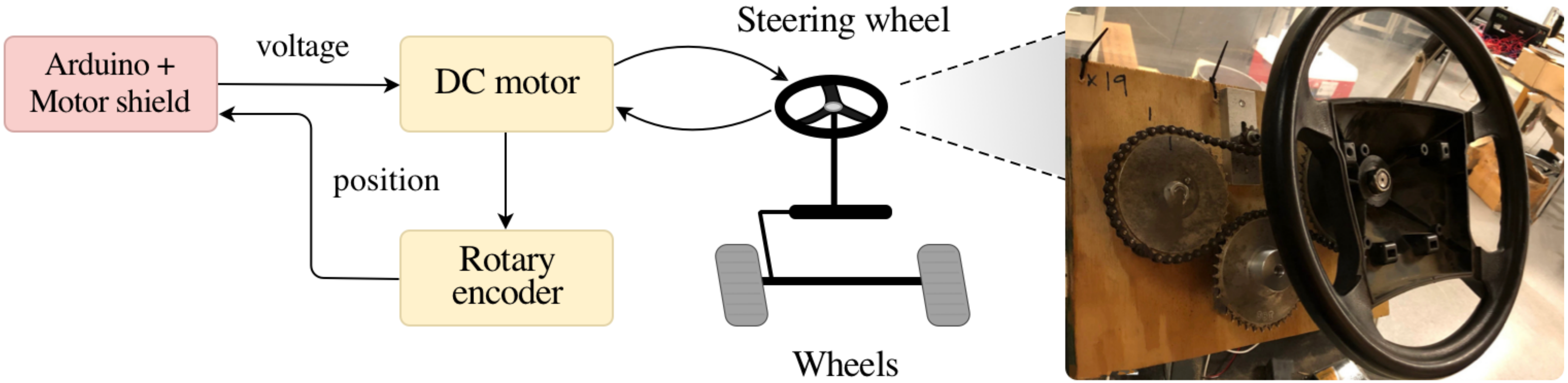
Systematic Diagram



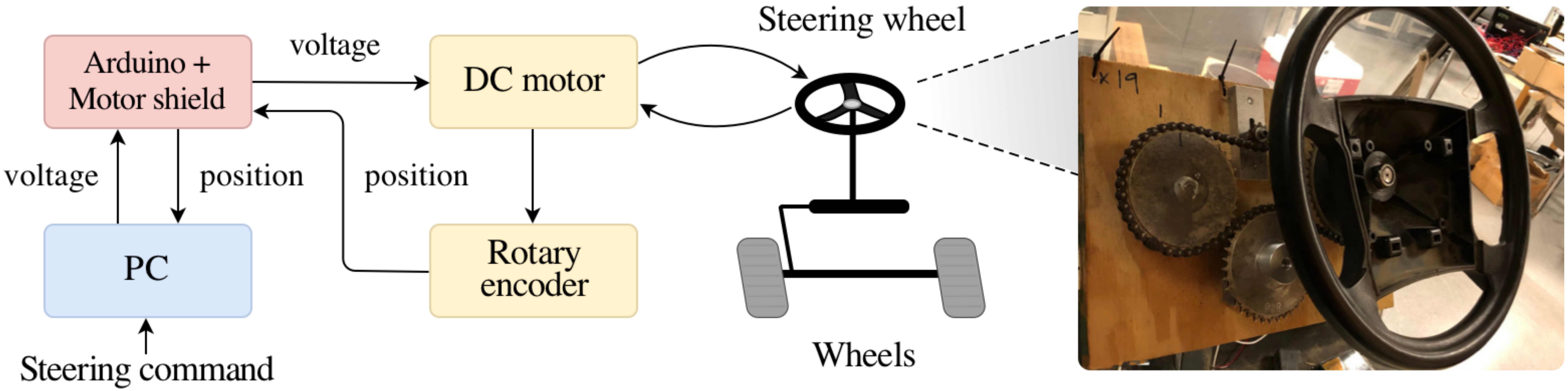
Systematic Diagram



Systematic Diagram



Systematic Diagram



Methodology

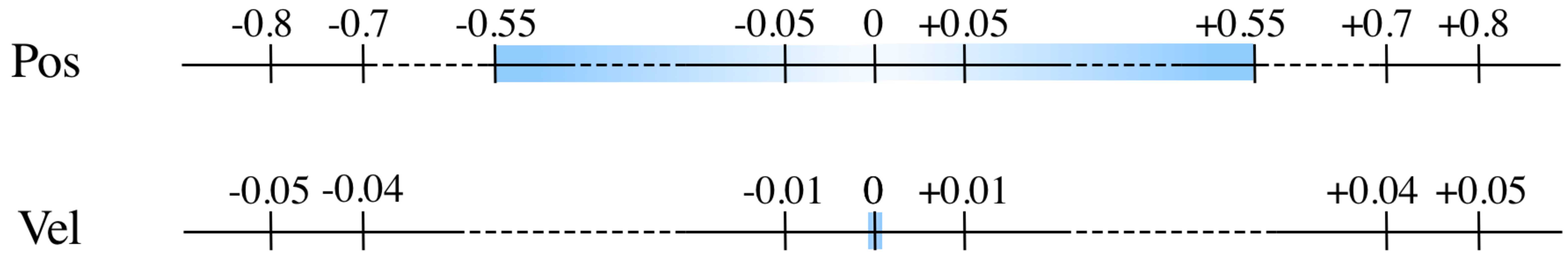
- Markov Decision Process (MDP)
  - State
  - Action
  - Transition function
  - Reward function

Methodology

- Markov Decision Process (MDP)
  - State
    - Motor position, velocity and applied voltage

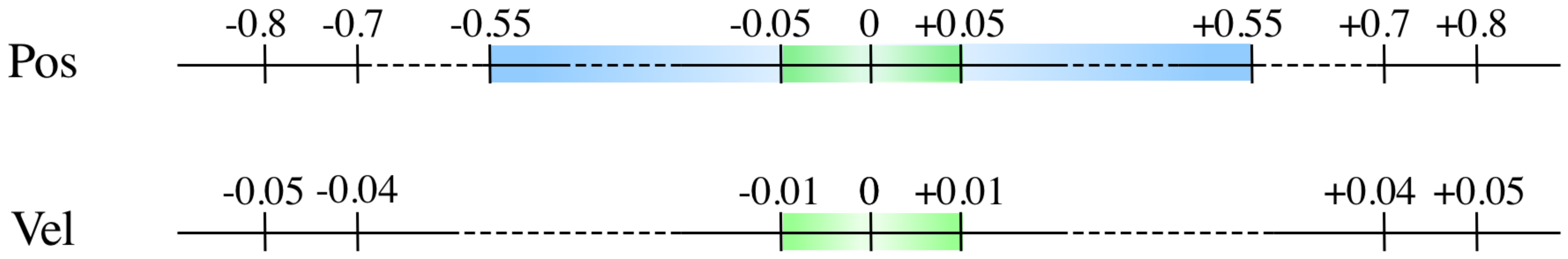
Methodology

- Markov Decision Process (MDP)
  - State
    - Motor position, velocity and applied voltage
  - Initial states



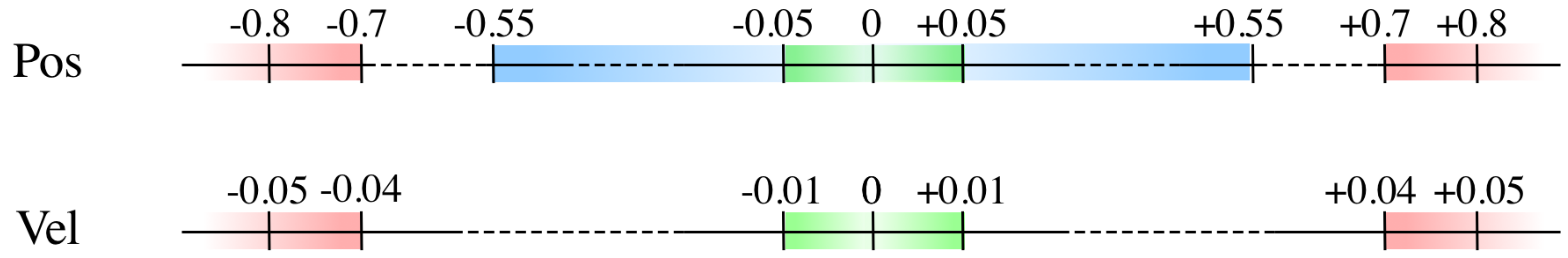
Methodology

- Markov Decision Process (MDP)
  - State
    - Motor position, velocity and applied voltage
  - Initial states
  - Goal states (G)



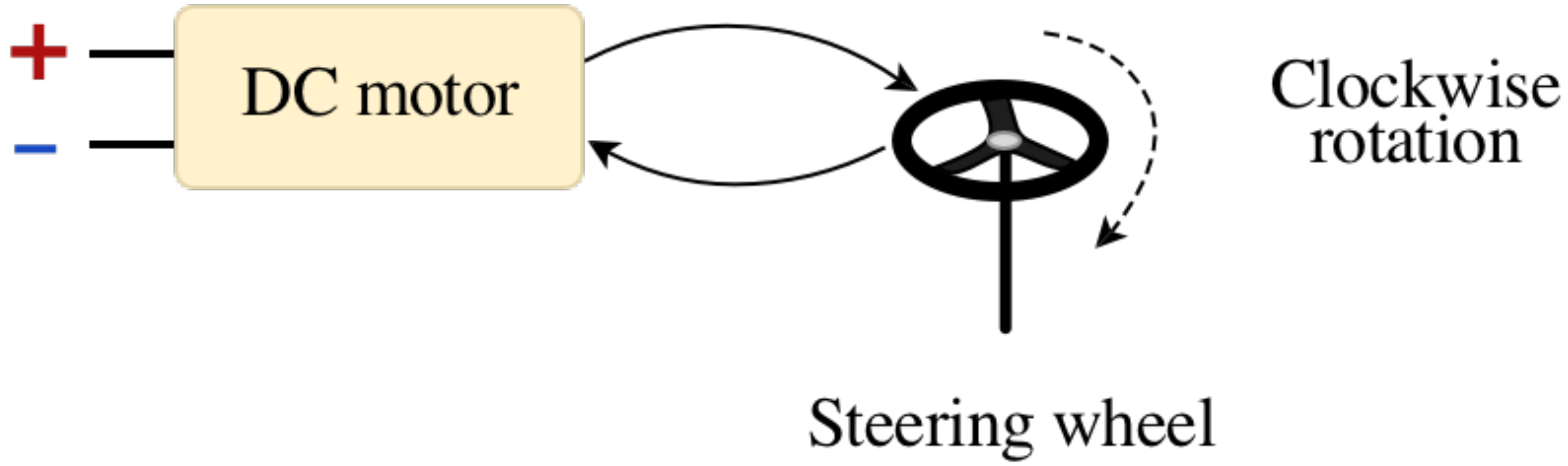
Methodology

- Markov Decision Process (MDP)
  - State
    - Motor position, velocity and applied voltage
  - Initial states
  - Goal states (G)
  - Forbidden states (F)



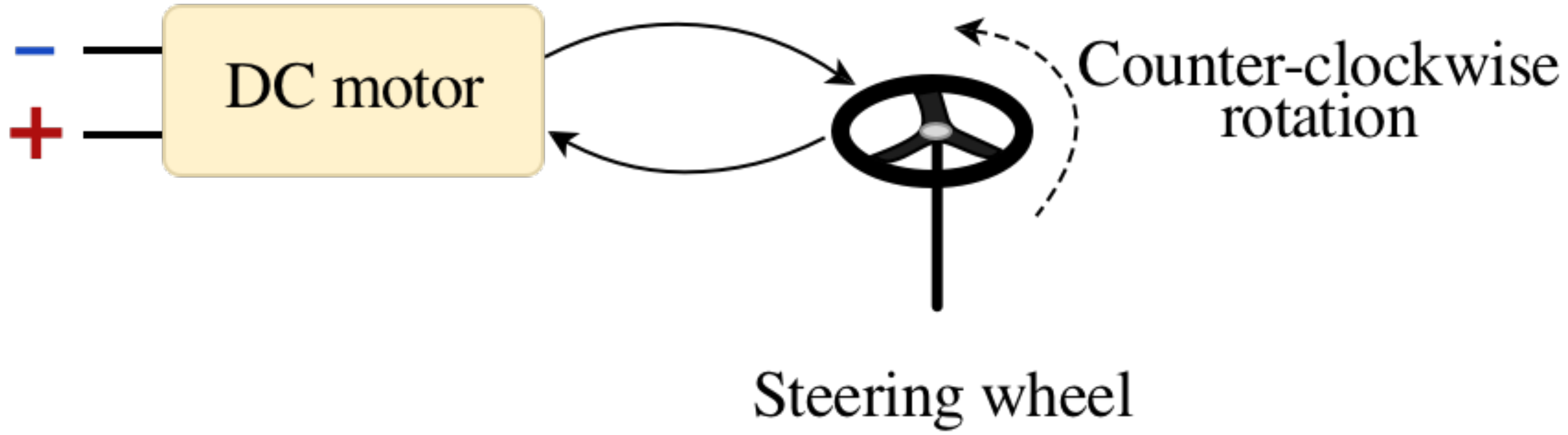
Methodology

- Markov Decision Process (MDP)
  - Action
    - Apply voltage (+, -)



Methodology

- Markov Decision Process (MDP)
  - Action
    - Apply voltage (+, -)
    - Apply voltage (-, +)



Methodology

- Markov Decision Process (MDP)
    - Transition function
        - Physical system (hardware)

Methodology

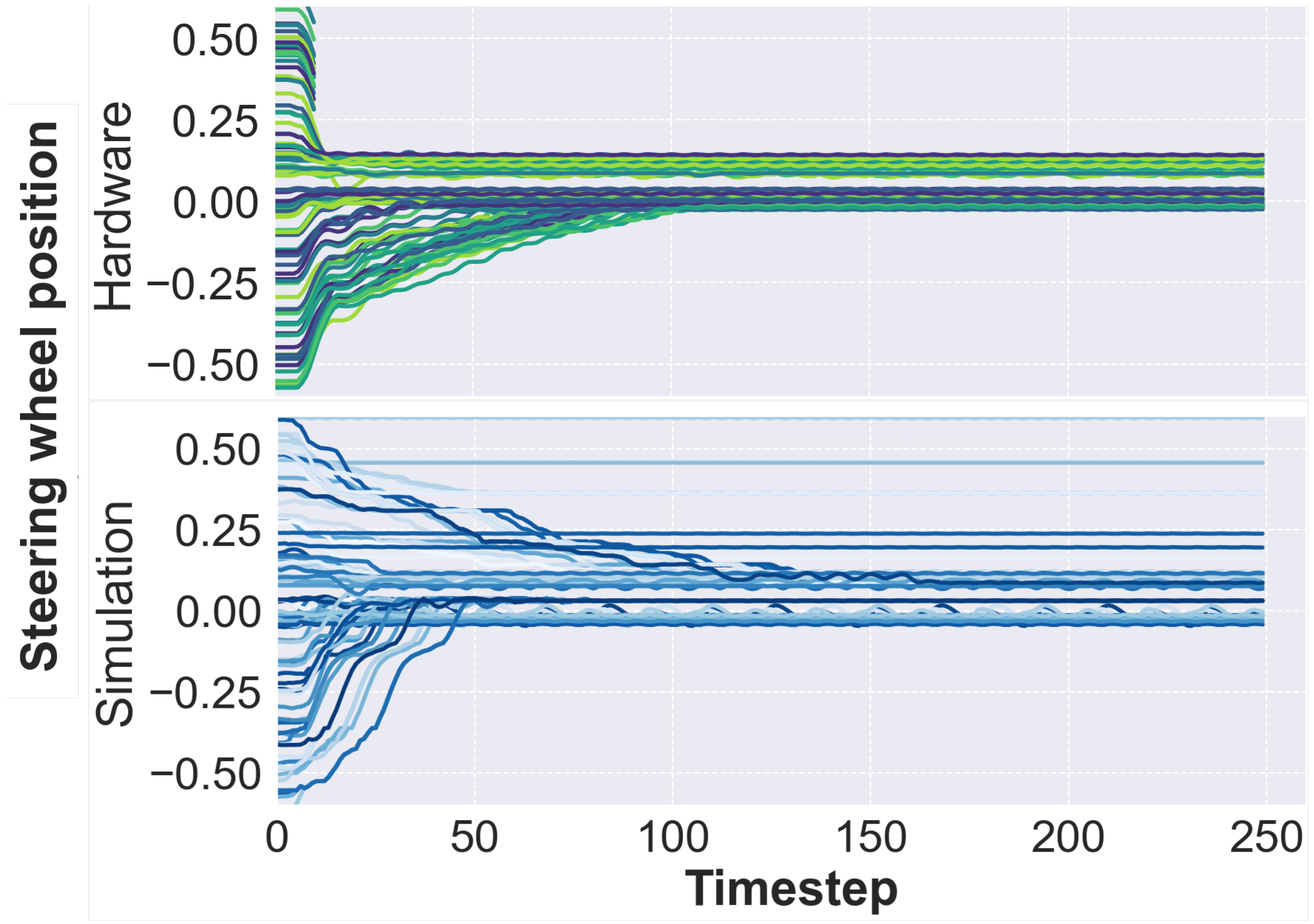
- Markov Decision Process (MDP)
  - Transition function
    - Simulation

Methodology

- Markov Decision Process (MDP)
    - Transition function
        - Simulation
        - **121,511** transitions from hardware
        - K-Nearest Neighbor

Methodology

- Markov Decision Process (MDP)
  - Transition function
    - Simulation
    - **121,511** transitions from hardware
    - K-Nearest Neighbor



Methodology

- Markov Decision Process (MDP)
    - Reward function

Methodology

- Markov Decision Process (MDP)
    - Reward function

$$\text{Cost} = \begin{cases} \mathbf{0} & \text{if state in Goal states (G)} \\ \mathbf{1} & \text{if state in Forbidden states (F)} \end{cases}$$

Methodology

- Markov Decision Process (MDP)
  - Reward function

$$\text{Cost} = \begin{cases} \mathbf{0} & \text{if state in Goal states (G)} \\ \mathbf{1} & \text{if state in Forbidden states (F)} \\ \mathbf{0.001} \times \text{ y} & \text{if state not in G and F} \end{cases}$$

Methodology

- Neural Fitted Q (NFQ) algorithm

$Q_0 \leftarrow$ Initialize MLPs
for iteration k $\in$ [1, N] **do**
    Generate pattern set,
    $\mathbf{P} = \{(input^l, target^l), l = 1, 2, \ldots D\}$, where:
        $input^l = s^l, a^l$
        $target^l = c(s^l, a^l) + \gamma \min_b Q_k(s'^l, b)$
    $Q_{k+1} \leftarrow$ Train on $\mathbf{P}$ // *via supervised learning*
    $k \leftarrow k + 1$
end for

Methodology

- Neural Network

	Simulation	Physical system
Activation	Sigmoid	Sigmoid
MLP Architecture	**4, 8, 5, 1**	**4, 5, 5, 1**
Parameters	**91**	**61**

Methodology

- Training setup

	Simulation	Physical system
Episodes	300	150
Timesteps	250	250
Optimizer	Rprop	Rprop
PC	Macbook Air	Macbook Pro

Experiments & Results

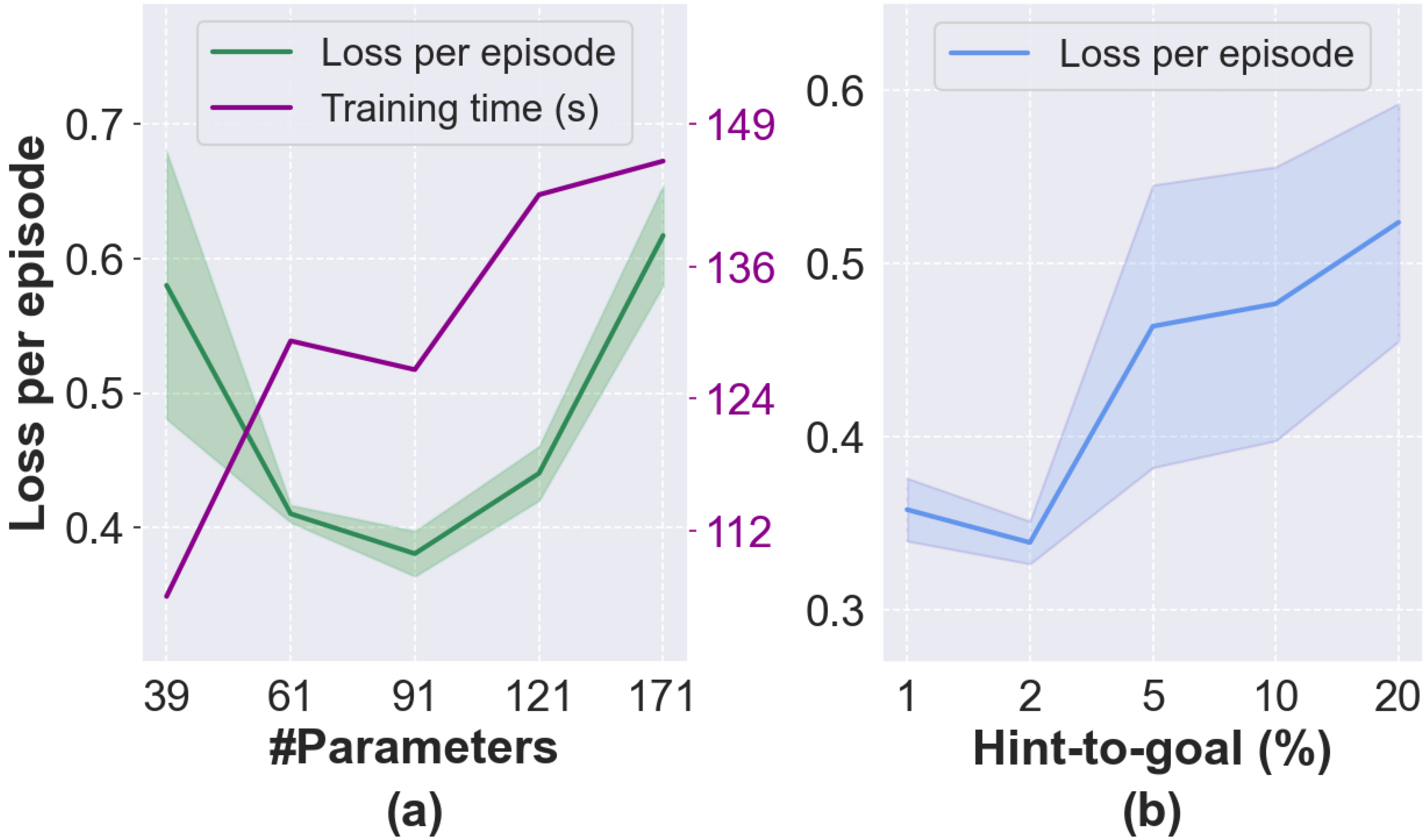
Experiments & Results

- Experiments in simulation

Experiments & Results

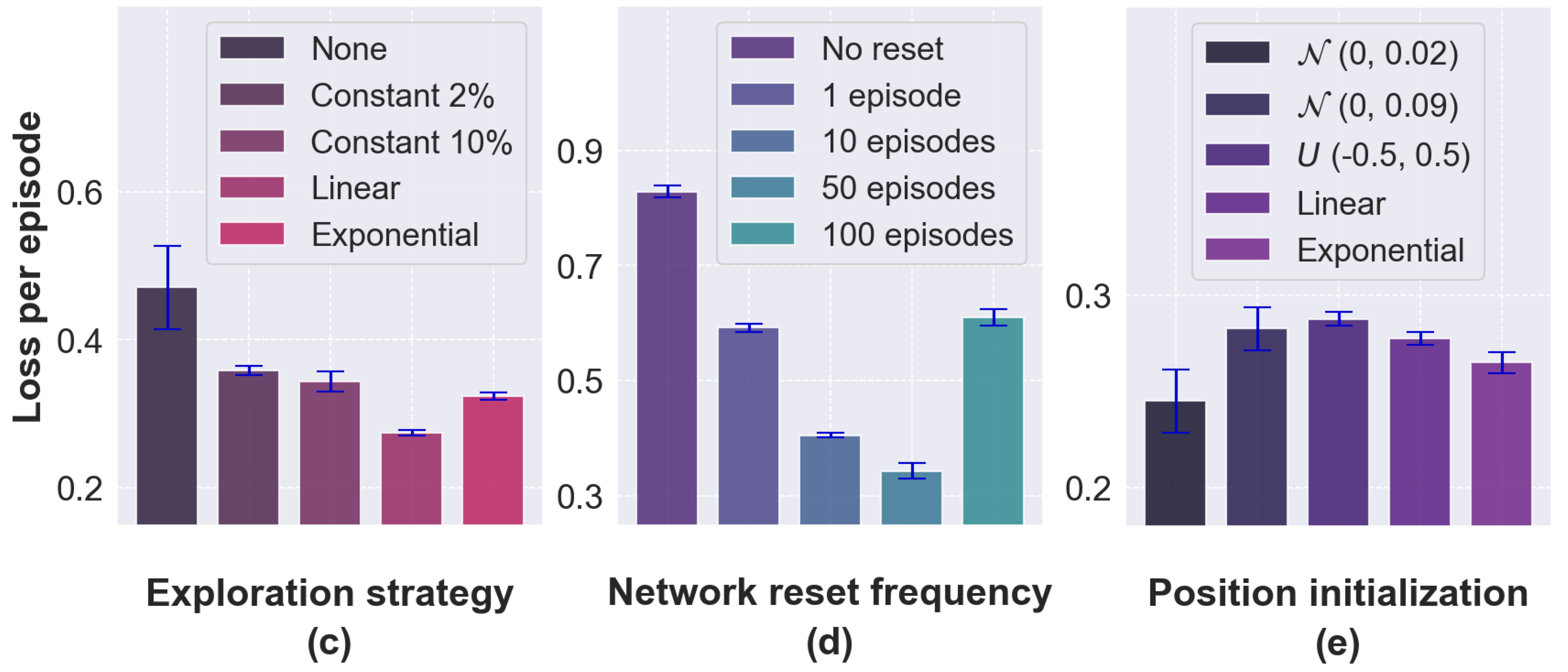
- Experiments in simulation
    - Number of parameters in Neural Network
    - Size of Hint-to-goal transitions
    - Exploration strategy
    - Network reset frequency
    - Position initialization

Experiments & Results



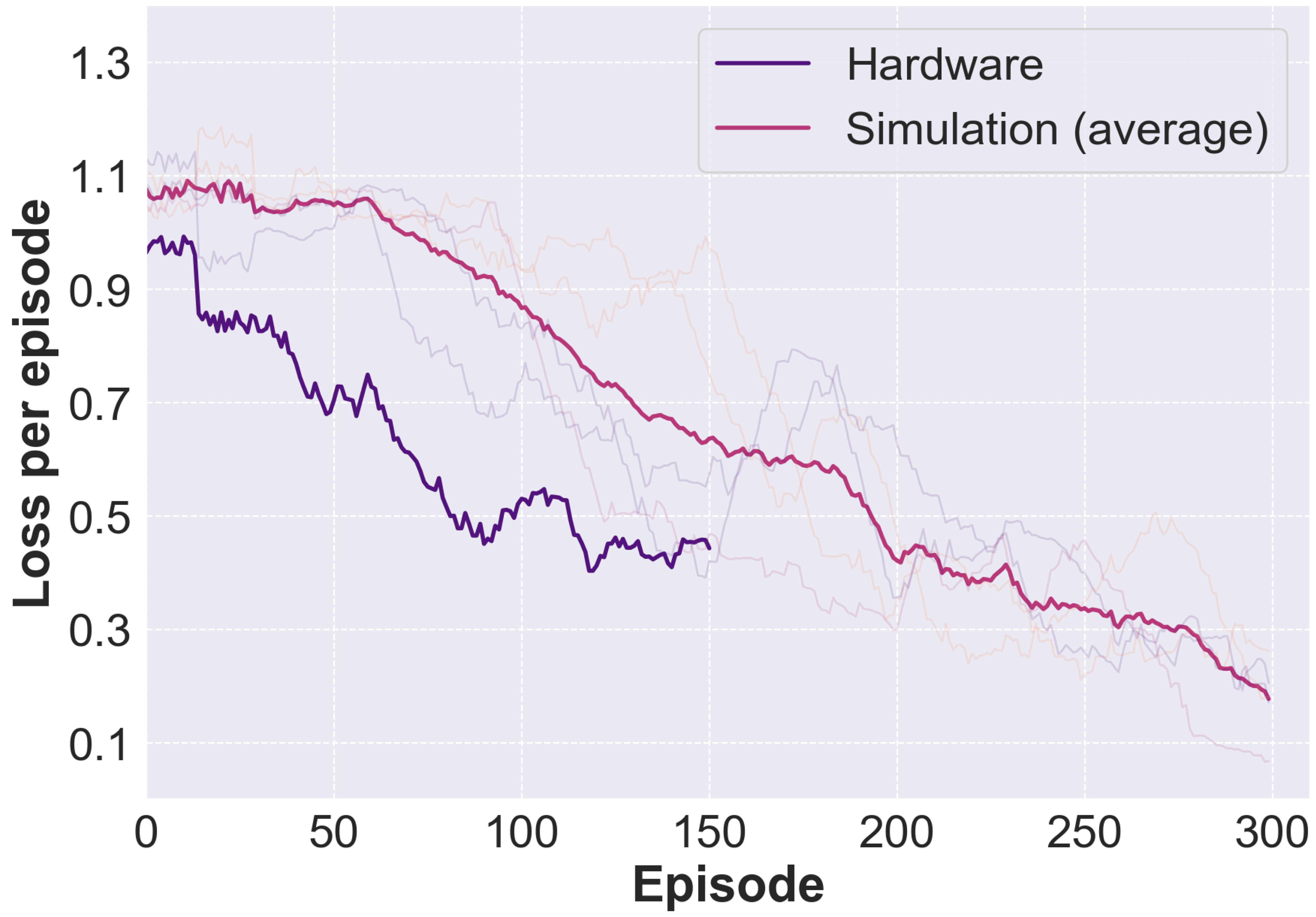
Experiments on simulation

Experiments & Results



Experiments on simulation

Experiments & Results



Results on physical system (hardware) and simulation

Conclusion and Future Work

Conclusion & Future Work

- Conclusion
  - Real-time control
  - Effective to search parameters and learning strategies in simulation

Conclusion & Future Work

- Conclusion
  - Real-time control
  - Effective to search parameters and learning strategies in simulation
- Future Work
  - Test the limit of algorithm
  - Replace the simulator with a parameterized function
  - Stability, sensitivity analysis

Thank You!

Bibek Poudel
bpoudel@memphis.edu

IEEE ITSC 2022

THE UNIVERSITY OF MEMPHIS

Socio-Technical Autonomous Resilient Systems