Balance and Control of a Humanoid Robot on an Electric Scooter

Guilherme Christmann - 60775056H
Advisor: Prof. Jacky Baltes

Outline

- **Introduction** - Motivation
- **Introduction** - Aims and Objectives
- **Lit. Review** - Two-Wheeled Vehicles
- **Lit. Review** - Reinforcement Learning and PPO
- **Methodology** - Scooter and Robot
- **Methodology** - PID Controller
- **Methodology** - RL Agent - PPO
- **Results and Discussion** - PID Controller - Balance Control
- **Results and Discussion** - RL Agent - Balance Control
- **Results and Discussion** - PID Controller - Trajectory Controller
- **Results and Discussion** - RL Agent - Trajectory Controller
- **Conclusion and Future Work**

Introduction

Introduction - Motivation

- Autonomous driving research has taken off in recent years.
  - Most of the interest is in four-wheeled vehicles (FWV).
  - Two-wheeled vehicles (TWV) research is not as popular.
  - TWV are not self-stable, and harder to control.
- (Humanoid) Robotics is another area that has seen great development.
  - Push for more general robots, humanoid shape.
  - General robots imply interacting with world in the same way as humans.
  - (TWV) Vehicles pose an interesting challenge.

Introduction - Objectives

- Design a physics-based simulation environment, including 3D CAD model of the robot and scooter.
- Develop a steering-based PID controller to balance the robot-scooter system.
- Develop a steering-based RL agent to perform balance control.
- Develop steering-based PID controller to perform trajectory tracking.
- Develop a steering-based RL agent to perform trajectory tracking.
- Evaluate, compare and discuss the classical vs learned control.

Literature Review

Lit. Review - Two-Wheeled Vehicles

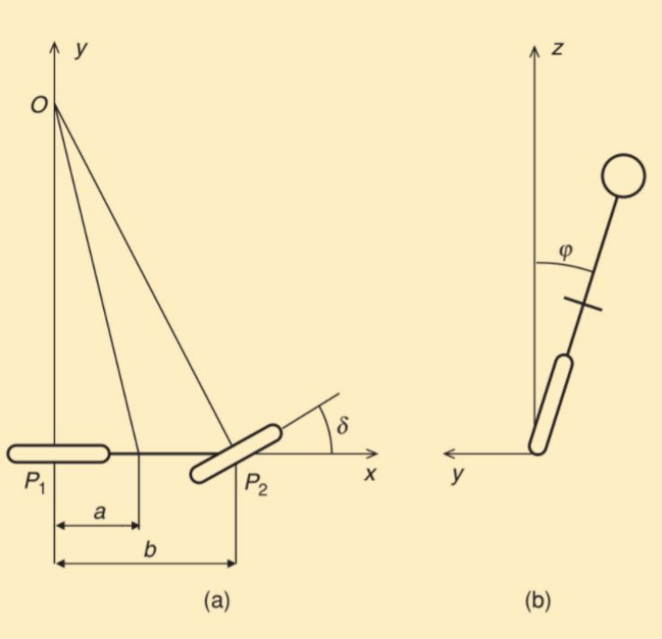
- Different from FWV, are generally not self-stable.
- When moving slowly TWV require continuous and quick input to maintain stability [1].
- Three main types of balance controllers:
  - Manipulating CoM with a rod of mass balancer;
  - Flywheel systems;
  - **Steering-based;**
- This work focuses solely on steering-based control.
- Aims to achieve stability by manipulating the steering angle.

Lit. Review - Two-Wheeled Vehicles

- In the simplest model we assume that the rider (robot) does not move;
- The only actuation happens directly in the steering joint.
- In the simplest model [2] we can apply the control law:

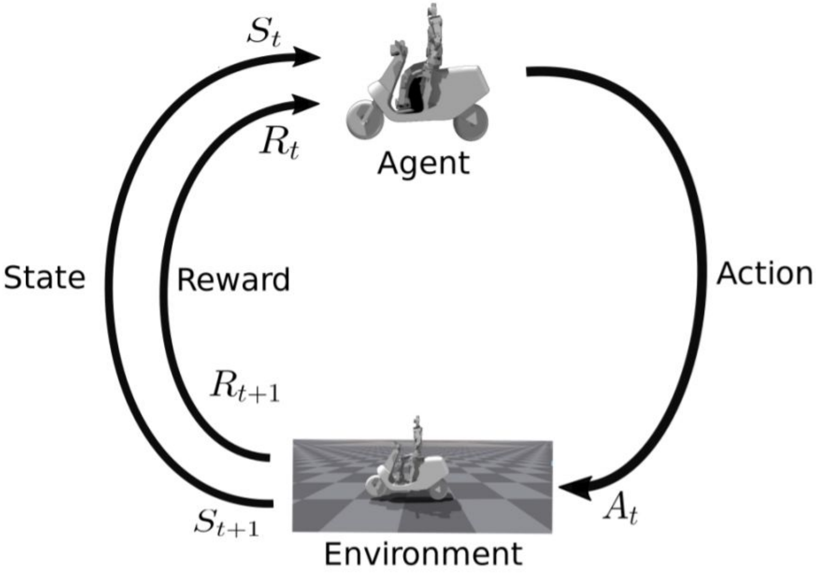
$$\delta = -k_1\phi$$

- Intuitively, turn in same direction of the fall.



Lit. Review - Reinforcement Learning and PPO

- RL is a family of algorithms inside ML.
- Interacts with data in an online manner, different from conventional Neural Nets.
- Essentially generates its own data by interacting with the environment in a loop.
- Learns by having a scalar value (reward) attributed to each action taken.



Lit. Review - Reinforcement Learning and PPO

- The elements in RL algorithms are [3]:
  - A policy - what generates the behavior of an agent for any given state.
  - A reward signal - determines the “goodness” of actions. Agent goal is to maximize it.
  - A value function - measures the expected future reward for any given state.
- Deep Q-Learning (DQN) [4] paper was responsible for the rise in popularity of Deep RL.
  - The name Deep RL comes from the usage of (deep) neural networks within the RL framework.
- Other methods were developed for continuous control: DDPG, and trust regions methods (TRPO and PPO).

Lit. Review - Reinforcement Learning and PPO

- Main problems of RL are data efficiency and robustness (progress loss).
- PPO is a SoTA RL method that tackles both problems.
  - Update is based on a clipped surrogate loss:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t]$$

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

$$A(s,a) = Q(s,a) - V(s)$$

- Avoids performing large updates, improves training stability.

Methodology

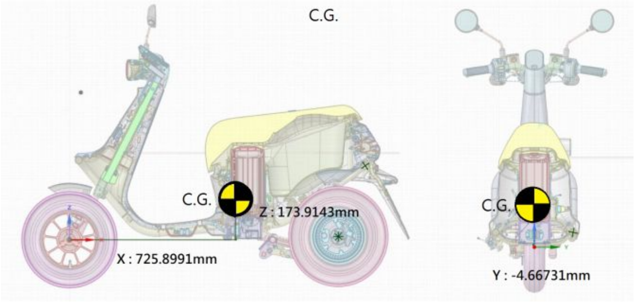
Methodology - Scooter and Robot

- Large-Sized Humanoid Robot THORMANG3

Degrees of Freedom	29
Actuator	200W x 10 / 100W x 11 / 20W x 8
Computer	Intel NUC (i5 Processor) – 8GB RAM DDR4 x 2
Wireless Router	DLink DIR-806A
Camera	Logitech C920 HD Camera
LiDAR	Hokuyo UTM-30LX-EW
F/T Sensor	ATi Mini58-SI-2800-120 x 2
IMU	MicroStrain 3DM-GX4-25
Battery	22V, 22000mA – 18.5V, 11000 mA
Height	137.5 cm
Weight	42 kg

Methodology - Scooter and Robot

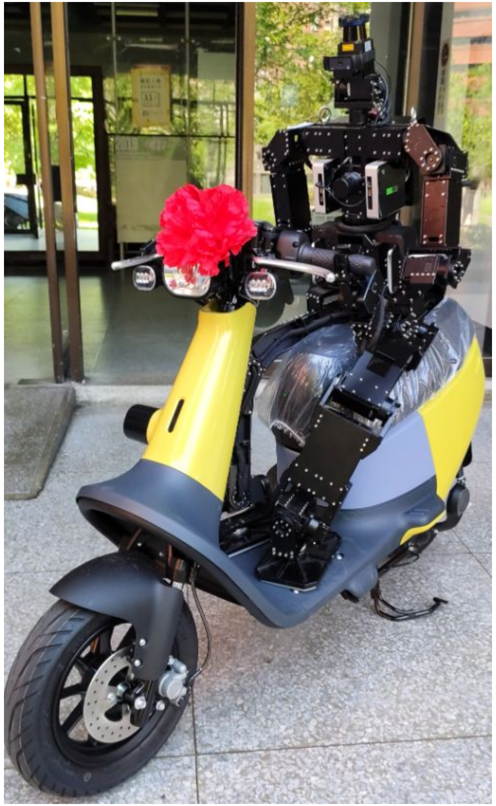
- Used a commercially available scooter - Gogoro Viva Plus:



$$\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} 4.40515 & 0.117584 & -1.66168 \\ -0.117584 & 18.0137 & 0.098875 \\ 1.66168 & -0.098875 & 14.8791 \end{bmatrix}$$

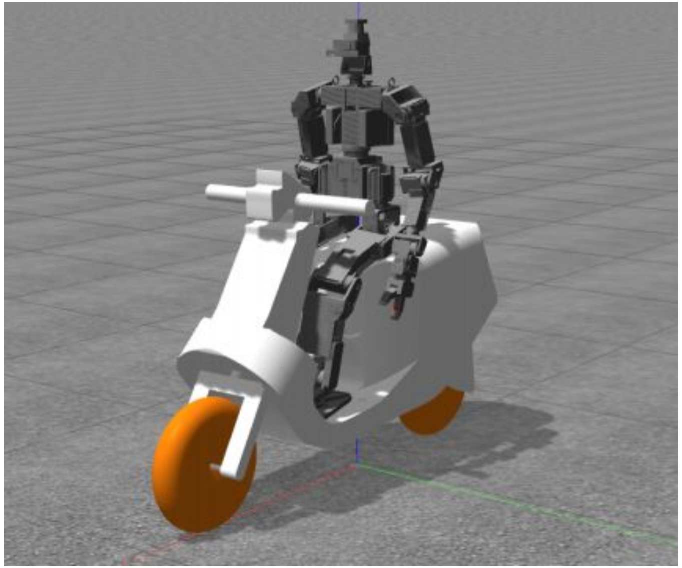
Wheel base	1164 mm
Seat height	740mm
Weight (with batteries)	79.2 kg
Max power output	3kW @ 500 rpm
Max horsepower	4.02 hp @ 500 rpm
Max torque (Motor / Wheel)	115 Nm @ 200 rpm
Hill climb ability	20% (11°): 25km/h 10% (6°): 40 km/h
Max lean angle	Left: 36° Right: 41°
Max riding range p/ battery	Approx. 85 km

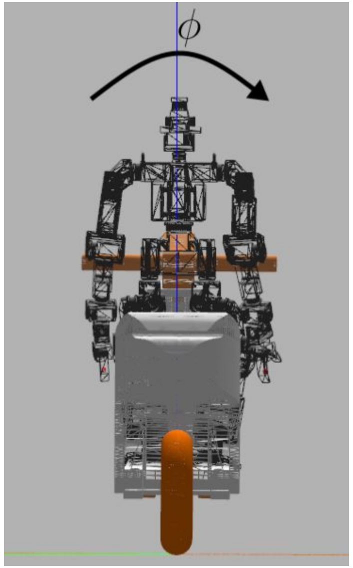
Methodology - Scooter and Robot

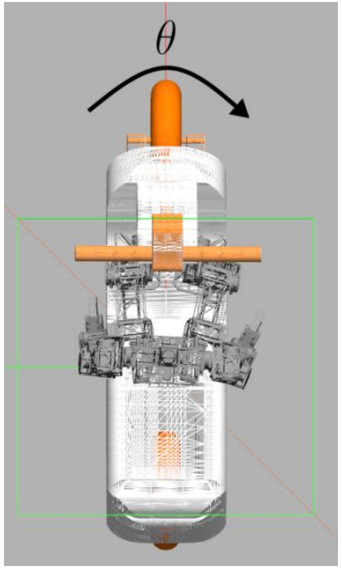


Methodology - Scooter and Robot

- 3D model of scooter designed with Autodesk Fusion 360:







Methodology - Balance Control - PID

- Based on the control law, a PID controller was designed:

$$\delta = -k_1\phi$$

- Error is measured based on a reference tilt angle.

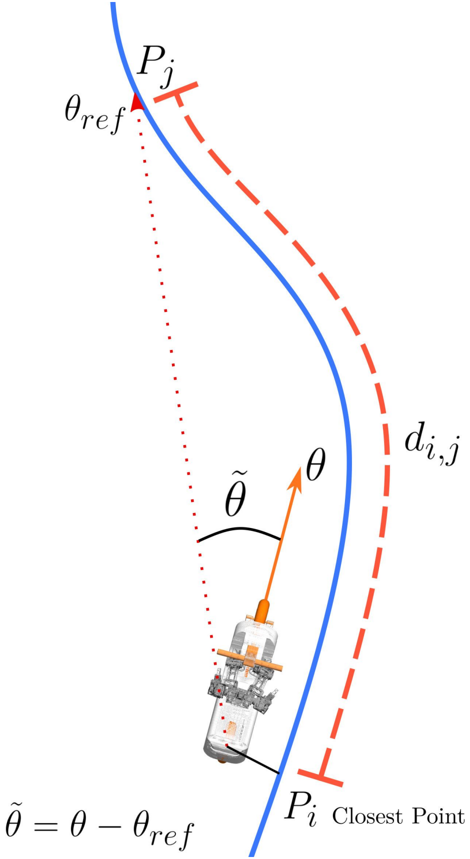
$$\text{Error} = \phi - \phi_{\text{ref}}$$

- Three experiments were realized for three velocities:
  - Balance with ϕref = 0, no disturbances;
  - Balance while turning (ϕref ≠ 0), no disturbances;
  - Balance with ϕref = 0 under disturbances.

System Velocity	P	I	D
10 rad/s (2.5 km/h)	8.0	0.04	4.0
20 rad/s (5 km/h)	2.0	0.04	4.0
40 rad/s (10 km/h)	1.0	0.04	2.0

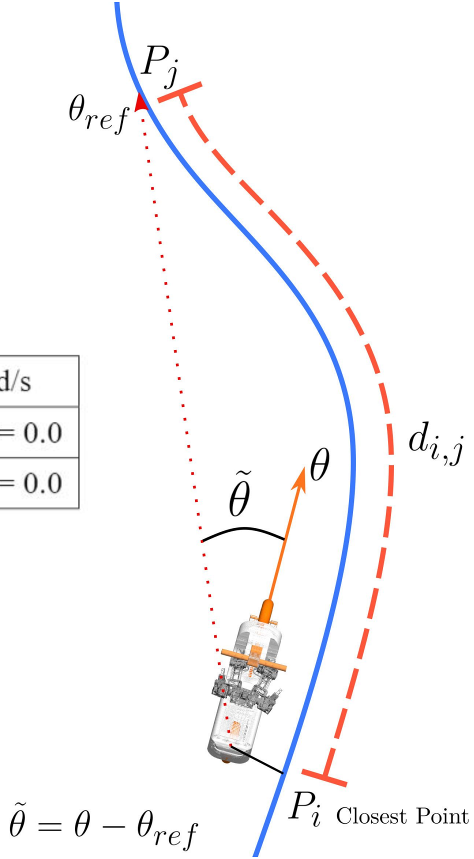
Methodology - Trajectory Tracking - PID

- Scooter can be turned by manipulating the tilt reference.
- A second PID controller was designed for trajectory tracking.
  - Measures current orientation error to the trajectory.
  - Sets the tilt reference value according to the error.
- Two experiments were realized:
  - Tracking trajectories: straight line path and sinusoidal path.
  - Tracking straight line under disturbances.



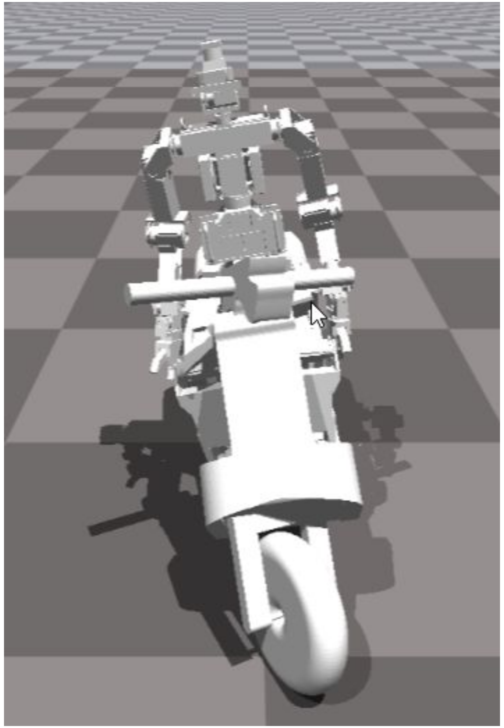
Methodology - Trajectory Tracking - PID

Mode	Velocity = 10 rad/s	Velocity = 20 rad/s	Velocity = 40 rad/s
Balance	P = 8.0, D = 4.00, I = 0.040	P = 2.0, D = 4.00, I = 0.040	P = 0.6, D = 0.8, I = 0.0
Tilt Reference	P = 0.1, D = 0.05, I = 0.001	P = 0.2, D = 0.25, I = 0.001	P = 0.8, D = 0.2, I = 0.0



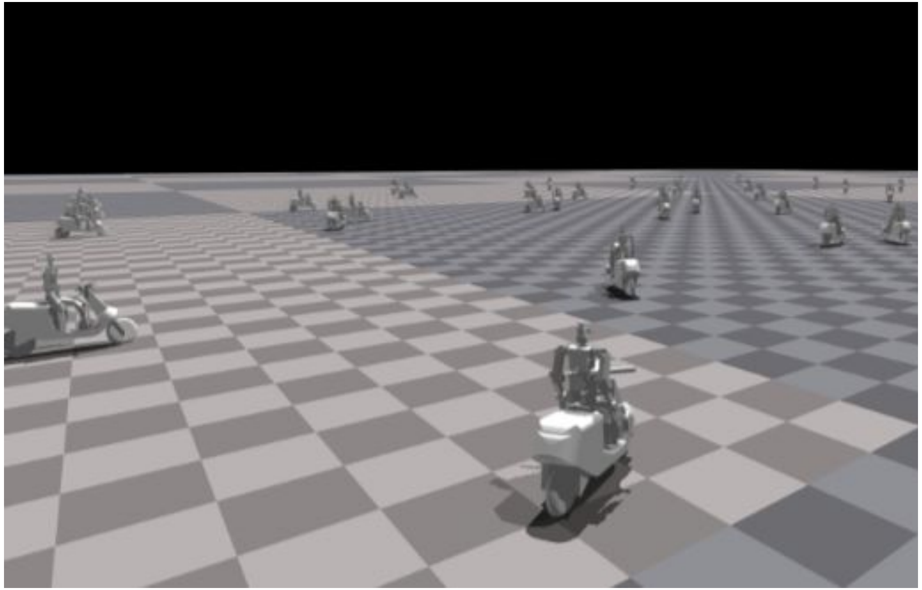
Methodology - RL Agent

- Two separate agents were trained:
  - Balance control, i.e. keep $\phi = 0$;
  - Commanded control, receive turning commands: $\dot{\theta}$
- A recent simulator NVIDIA Isaac Gym was used:
  - Allows direct access to GPU tensors with a PyTorch wrapper.
  - 64 environments were trained in parallel.
- Three-layer MLP network:
  - 256 -> 128 -> 64 Neurons, SELUs activation.
- Computer specifications:
  - GPU: 1x RTX 2080 Super (8GB VRAM)
  - CPU: 8-core AMD Ryzen 7 3700X
  - RAM: 32 GB



Methodology - RL Agent

Clip Range (ϵ)	0.2
Sampling steps (T)	10
Optimization Steps	10
Minibatches (per environment)	2
Optimizer	Adam
Learning Rate	3.0e-5
Gamma (γ)	0.99



Methodology - RL Agent

- Both agents were trained for velocities in range [10, 40] rad/s.
- Both agents only action was to change of steering angle (δ)
- Balance control agent:
  - 8 observations:
    - 1x Current steering angle;
    - 1x Rear wheel velocity;
    - 3x Orientation (roll, pitch, yaw);
    - 3x Angular velocities (Δroll, Δpitch, Δyaw);
  - Positive rewarded for keeping tilt angle close to 0: $R = 1.0 - \phi^2$
  - -2 reward for falling over.
  - Trained for 4 million timesteps (~2 hours of real-time).
- Two experiments:
  - Balance under no disturbances;
  - Balance under disturbances;

Methodology - RL Agent

- Commanded control agent:
  - 12 observations (1 being a command):
    - 1x Current steering angle;
    - 1x Rear wheel velocity;
    - 3x Orientation (roll, pitch, yaw);
    - 3x Angular velocities;
    - 3x Linear velocities;
    - 1x **Target ΔYaw (command)**
  - Positive rewarded for keeping ΔYaw close to target: $R = 1.0 - |\dot{\theta} - \dot{\theta}^{\text{targ}}|^{\frac{1}{5}}$
  - -2 reward for falling over.
  - Trained for 6.1 million timesteps (~3 hours of real-time).
- Every 2.5 s, ΔYaw is offset by a random value in the range -5.7° to 5.7°.
- Trajectory tracking is done in same way as PID controller, but only with P=1 term to set the target ΔYaw.

Methodology - RL Agent

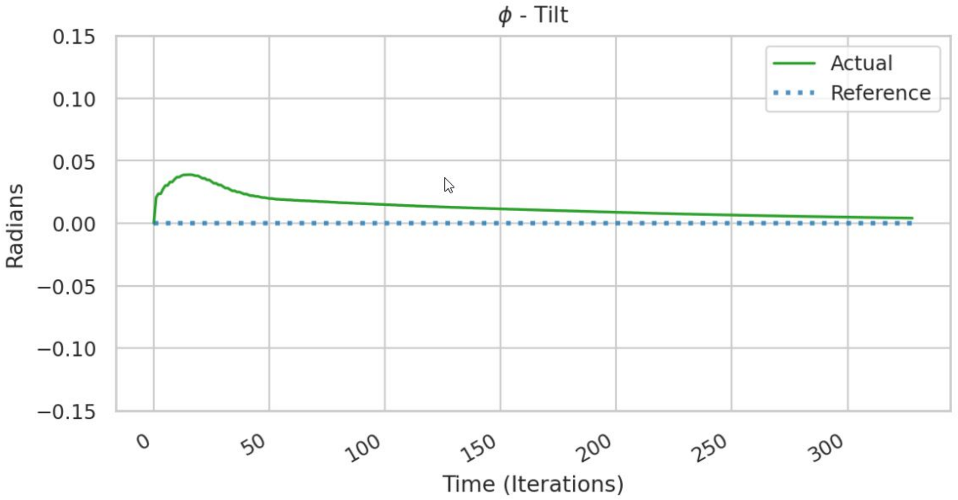
- Two experiments were performed for trajectory tracking:
  - Tracking a straight line path and sinusoidal path under no disturbances;
  - Tracking a straight line path under increasing disturbances;

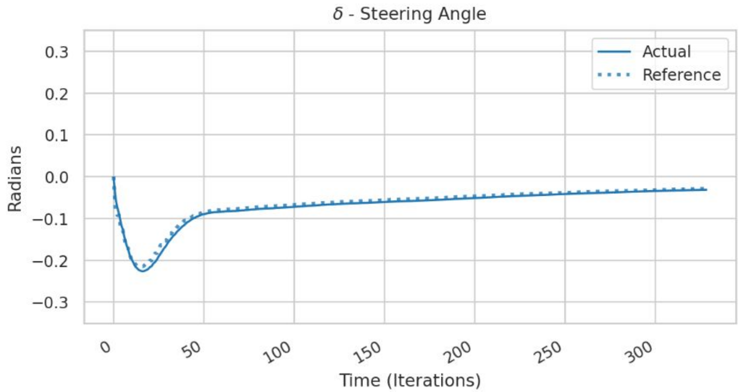
- The disturbance force was generated by hurling a box object towards the center of mass of the robot-scooter system.
  - After contact the difference in velocity is used to compute the resulting force:
  - $F = m \cdot \Delta V$
  - Larger forces are are generated by increasing the velocity.

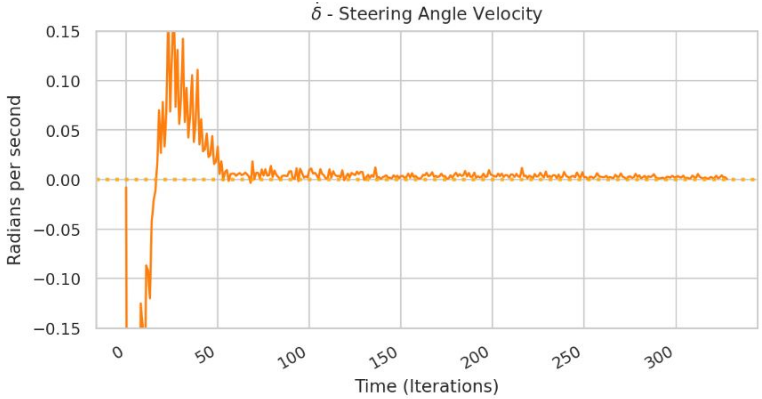
Results and Discussion

Results - PID Controller - Balance Control - Experiment 1

- Experiment 1 - Upright Control
- 10 rad/s -- 2.5 km/h

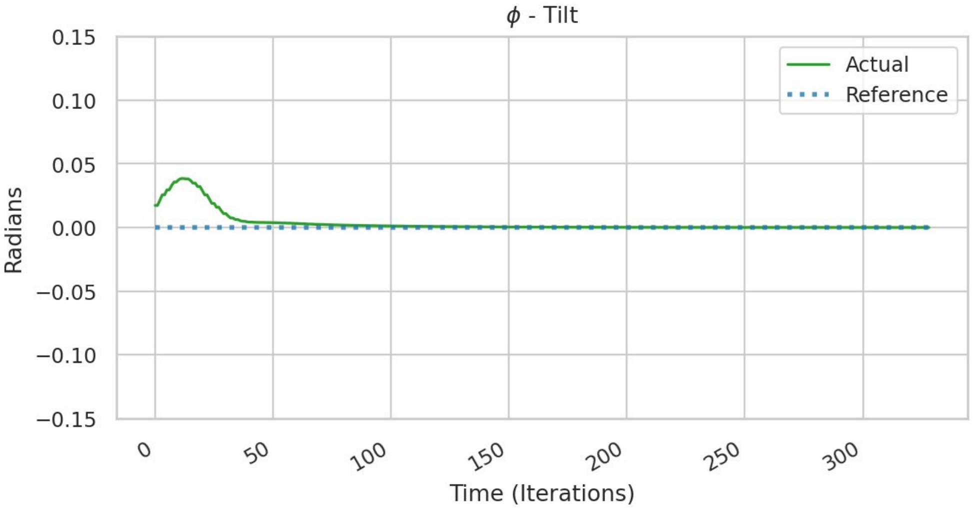


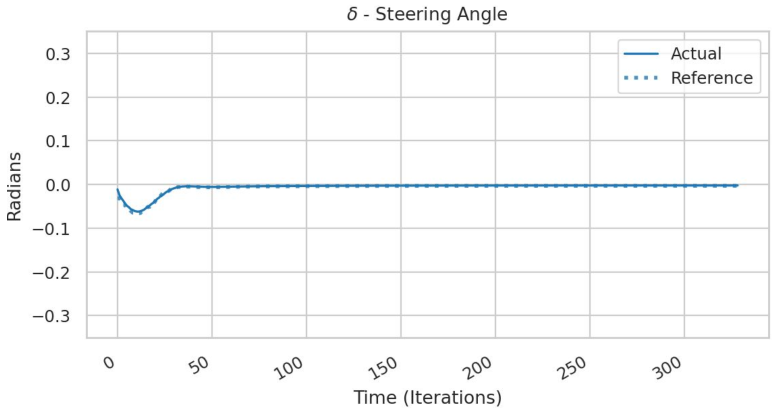


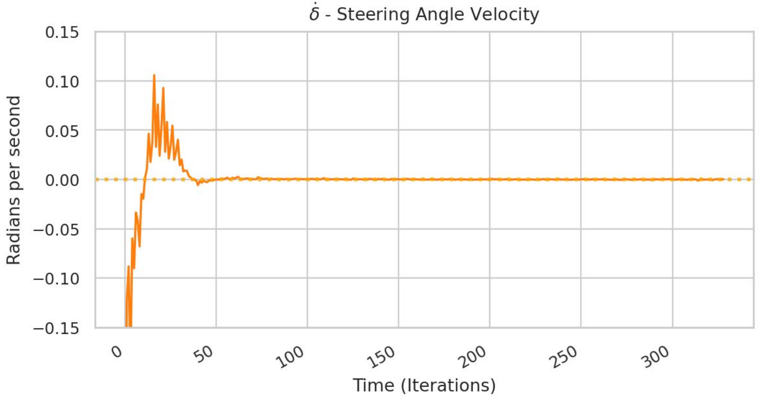


Results - PID Controller - Balance Control - Experiment 1

- Experiment 1 - Upright Control
- 20 rad/s -- 5 km/h

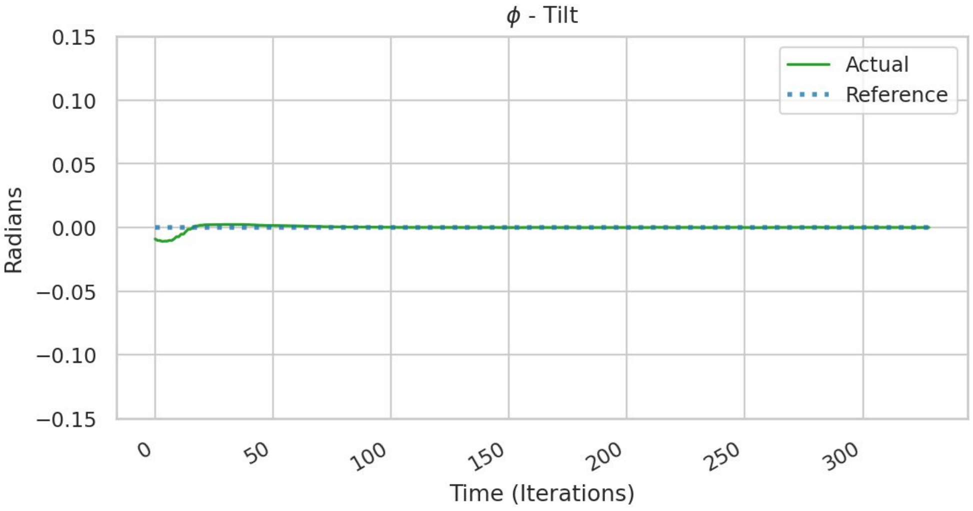


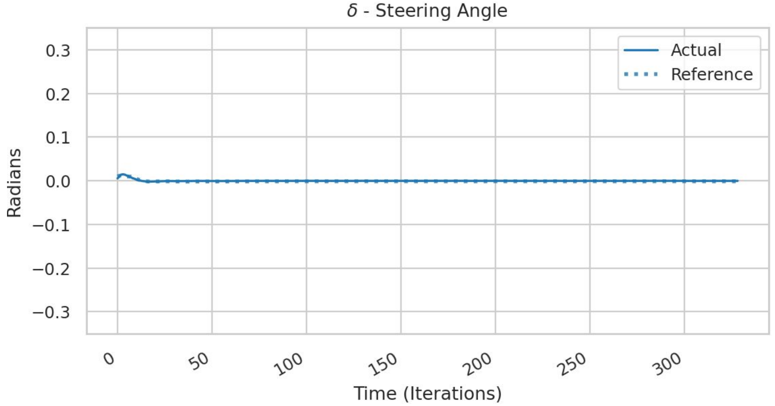


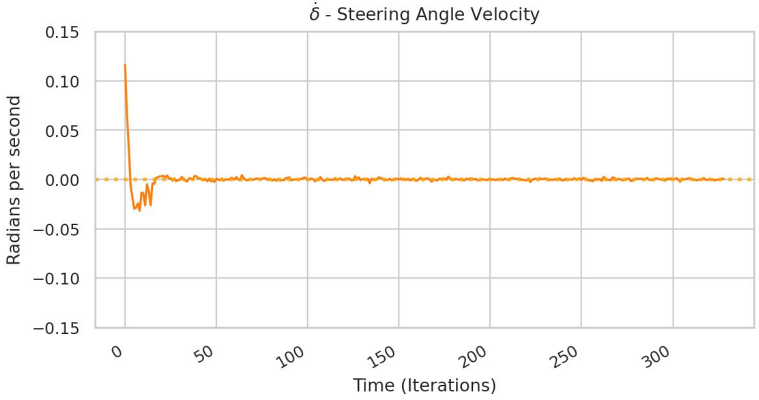


Results - PID Controller - Balance Control - Experiment 1

- Experiment 1 - Upright Control
- 40 rad/s -- 10 km/h

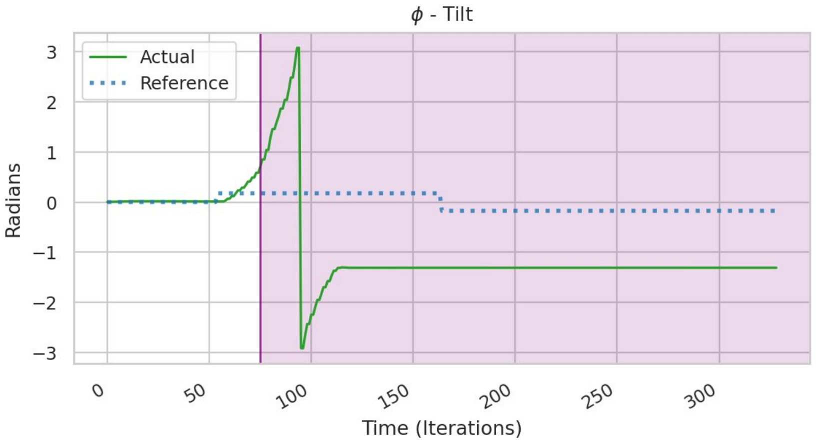


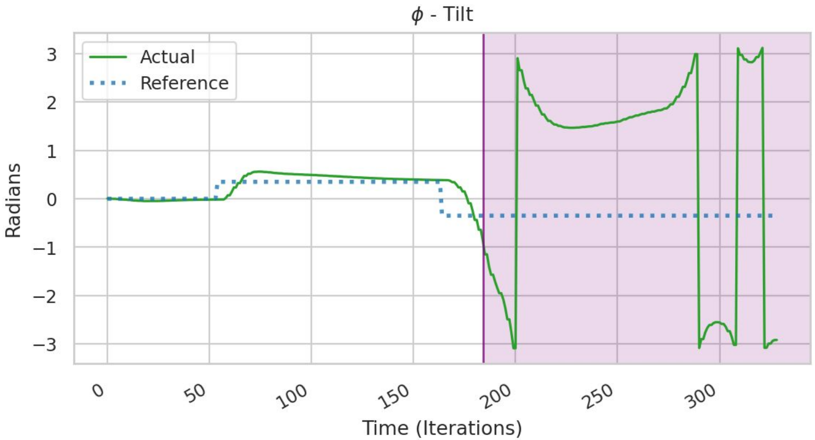




Results - PID Controller - Balance Control - Experiment 2

- Experiment 2 - Balance while turning (20 degrees of tilt)
- 10 rad/s -- 2.5 km/h
- 20 rad/s -- 5 km/h

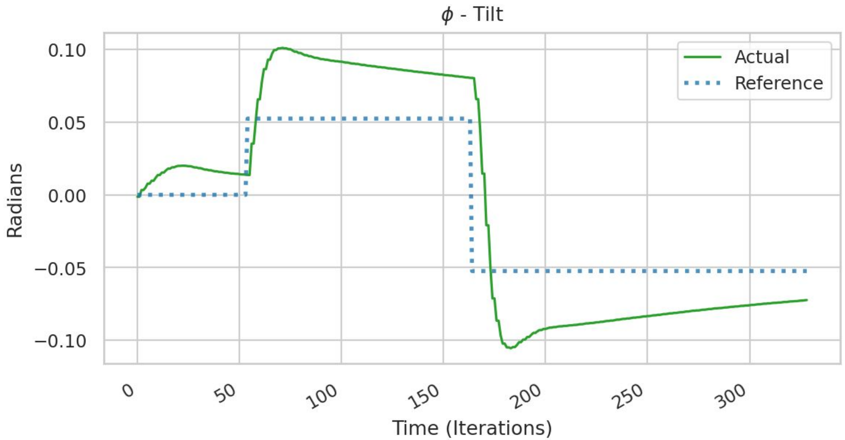




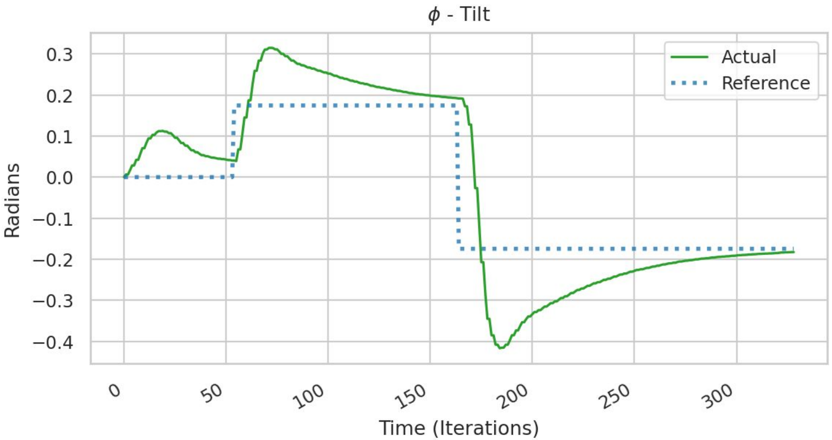
- Lower velocities fail to balance while turning.

Results - PID Controller - Balance Control - Experiment 2

- Experiment 2 - Balance while turning
- 10 rad/s -- 2.5 km/h (3 degrees)



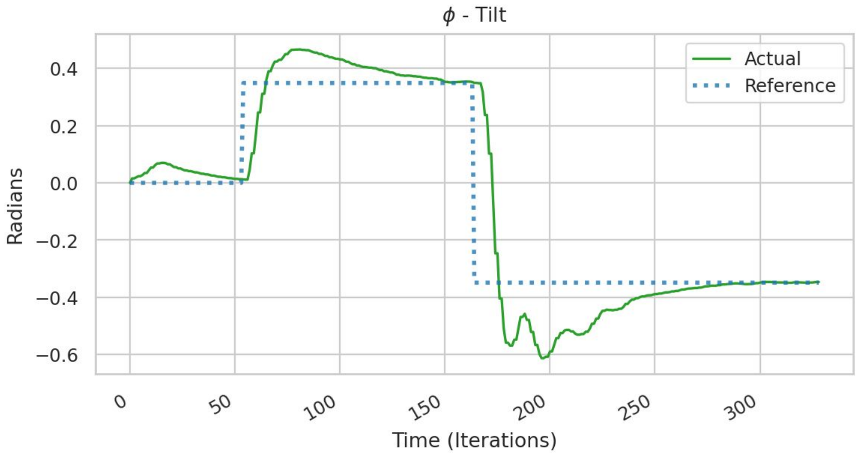
- 20 rad/s -- 5 km/h (10 degrees)



- Lower velocities are stable at reduced tilt angle changes (turning less)

Results - PID Controller - Balance Control - Experiment 2

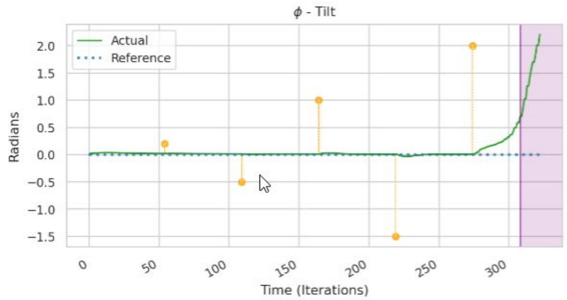
- Experiment 2 - Balance while turning
- 40 rad/s -- 10 km/h (20 degrees)



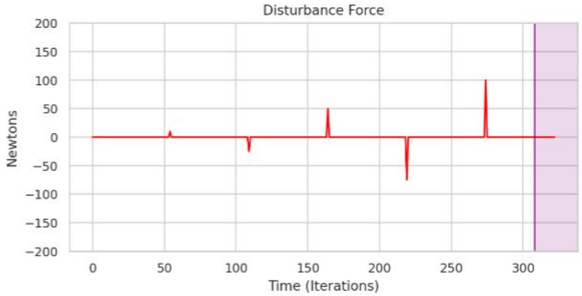
- At maximum velocity controller can handle large turns.

Results - PID Controller - Balance Control - Experiment 3

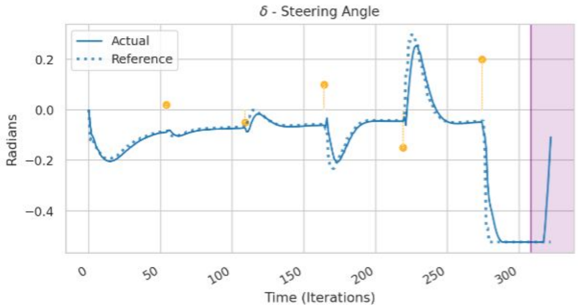
- Experiment 3 - Balance under disturbances - 10 rad/s (2.5 km/h)



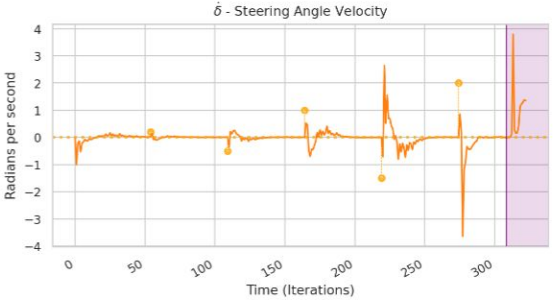
(a) Reference and actual tilt angle.



(b) Lateral disturbance force.



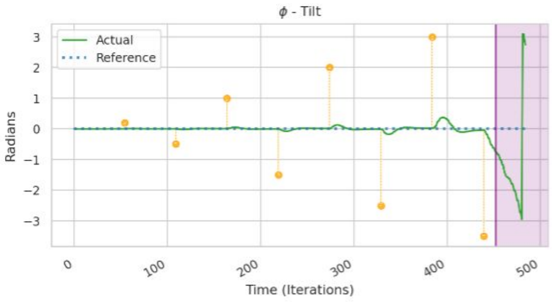
(c) Reference and actual steering angle.



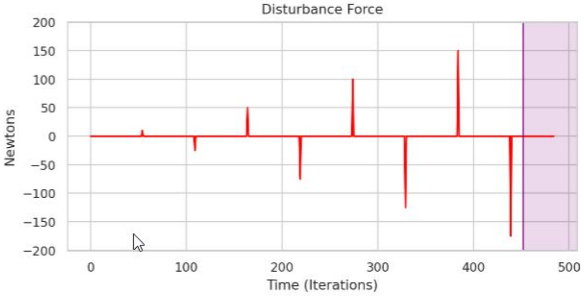
(d) Rate of change of the steering angle.

Results - PID Controller - Balance Control - Experiment 3

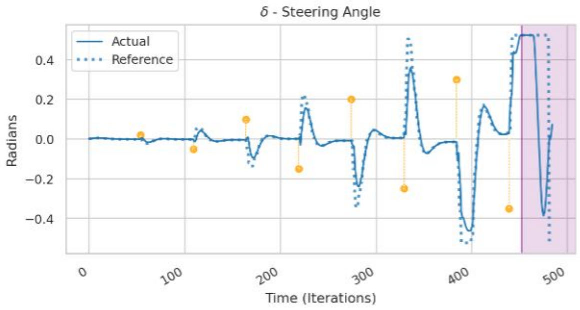
- Experiment 3 - Balance under disturbances - 20 rad/s (5 km/h)



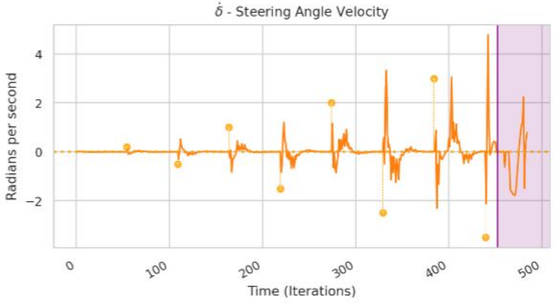
(a) Reference and actual tilt angle.



(b) Lateral disturbance force.



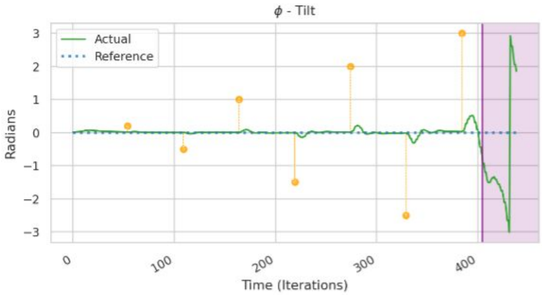
(c) Reference and actual steering angle.



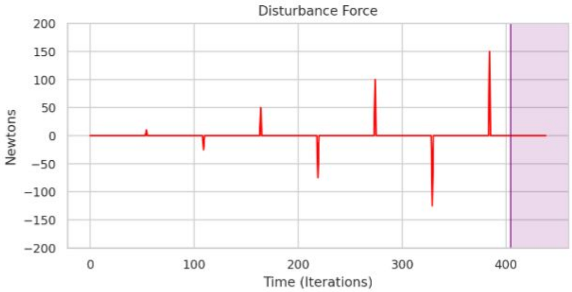
(d) Rate of change of the steering angle.

Results - PID Controller - Balance Control - Experiment 3

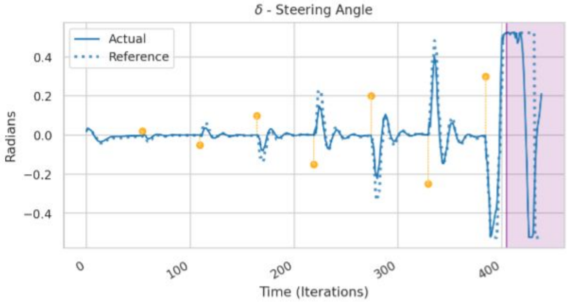
- Experiment 3 - Balance under disturbances - 40 rad/s (10 km/h)



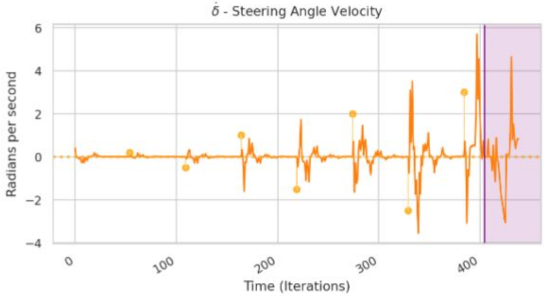
(a) Reference and actual tilt angle.



(b) Lateral disturbance force.

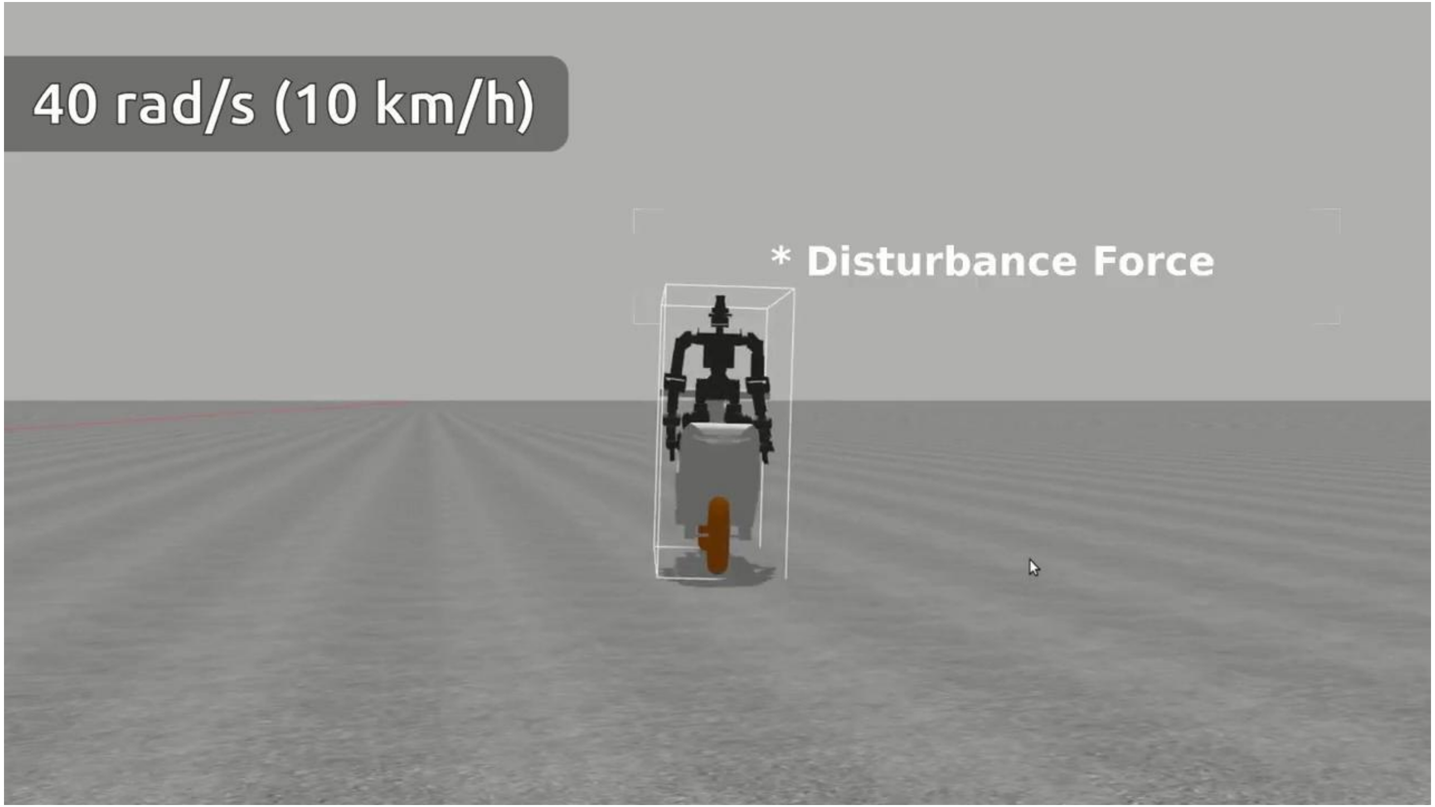


(c) Reference and actual steering angle.

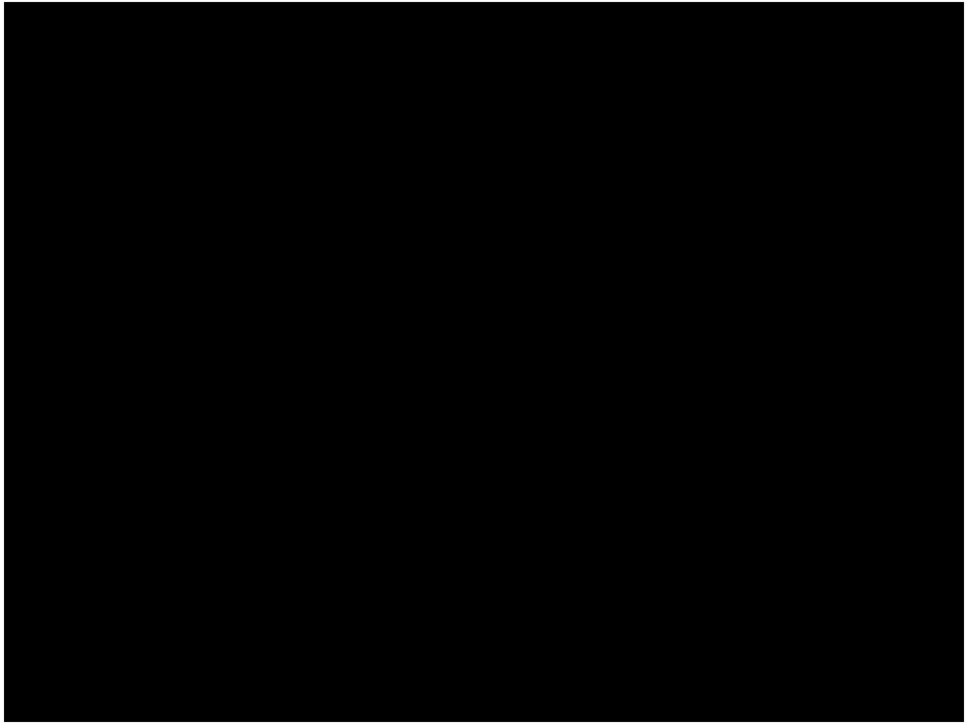


(d) Rate of change of the steering angle.

Results - PID Controller - Balance Control - Video

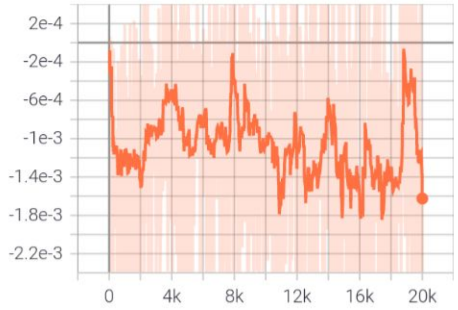


Results - PID Controller - Balance Control - Video

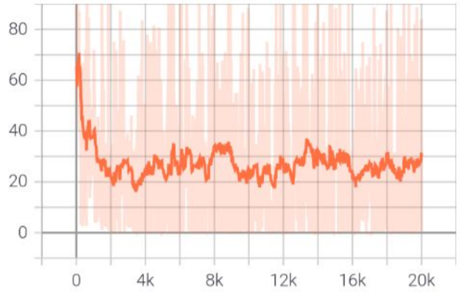


Results - RL Agent - Balance Control

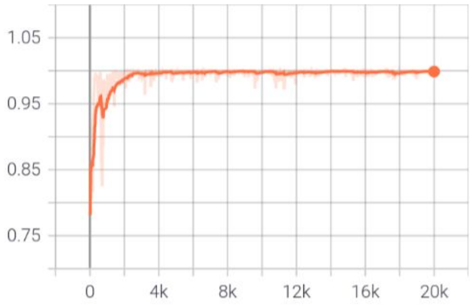
- Training graphs show steady progress:
    - Mean reward -> 1.0
    - Mean episode length -> 2000 (max)



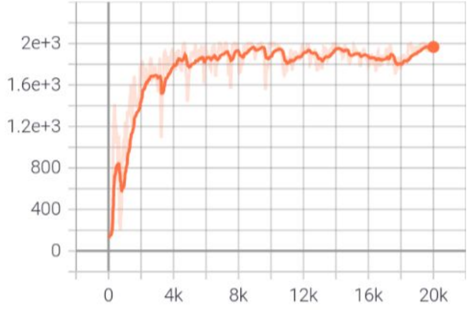
(a) PPO's surrogate loss over 20k training iterations.



(b) The value function loss over 20k training iterations.



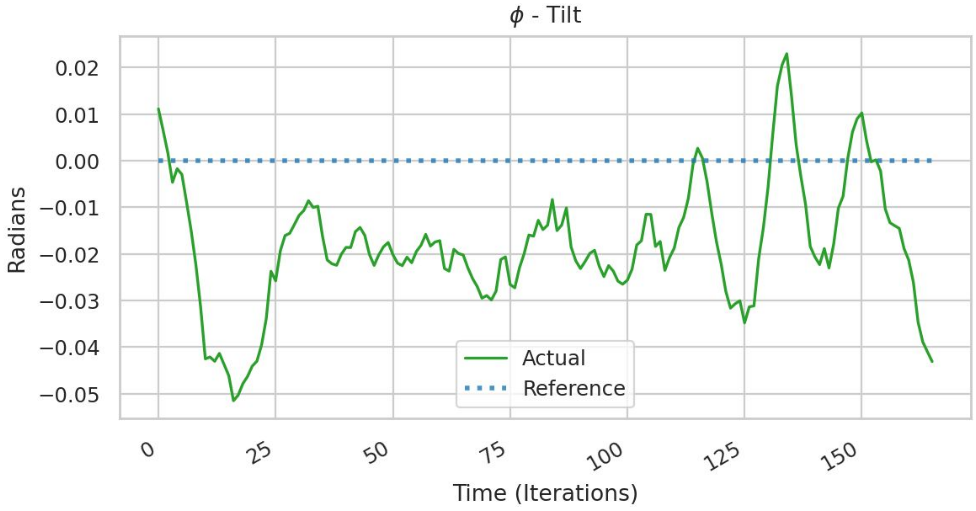
(c) The mean reward per step increases and converges close to the maximum reward value (1.0) as training progresses.

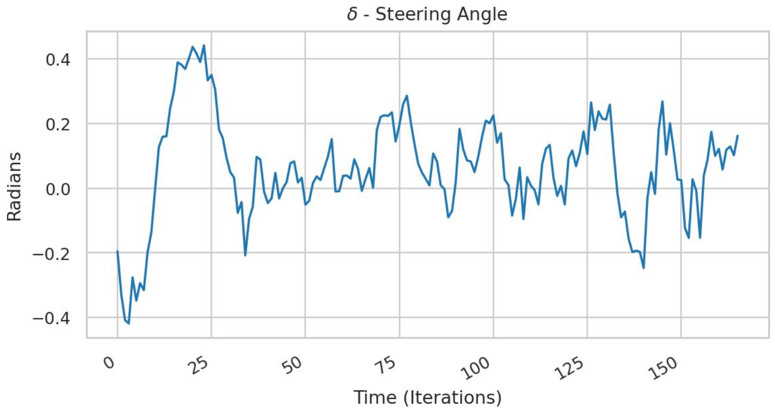


(d) The mean episode length increases and converges to the maximum episode length (2000) as training progress.

Results - RL Agent - Balance Control - Experiment 1

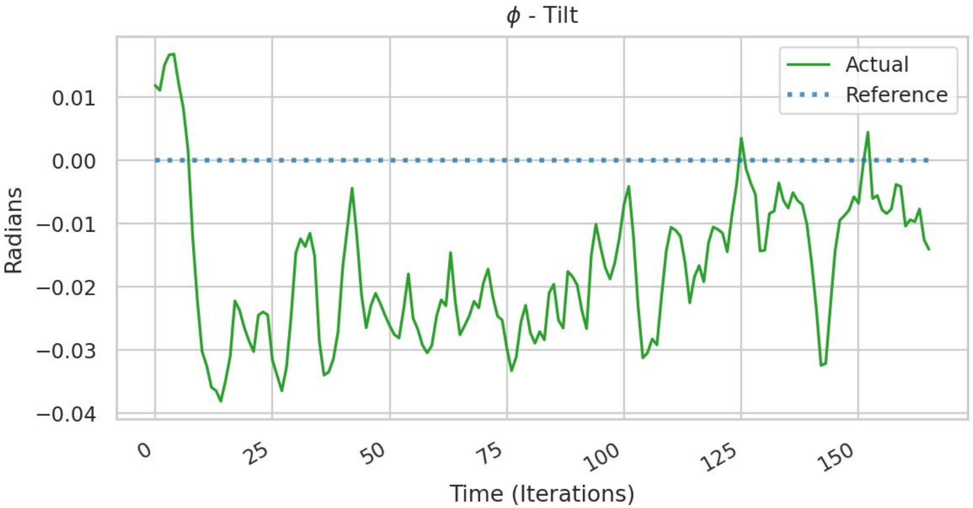
- Experiment 1 - Upright Control
- 10 rad/s -- 2.5 km/h

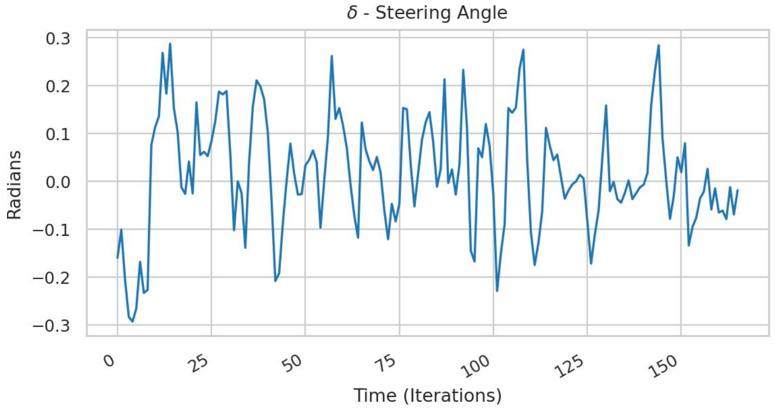


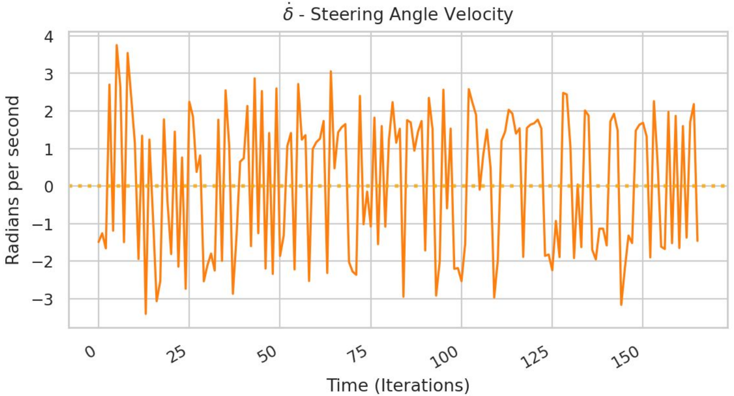


Results - RL Agent - Balance Control - Experiment 1

- Experiment 1 - Upright Control
- 20 rad/s -- 5 km/h

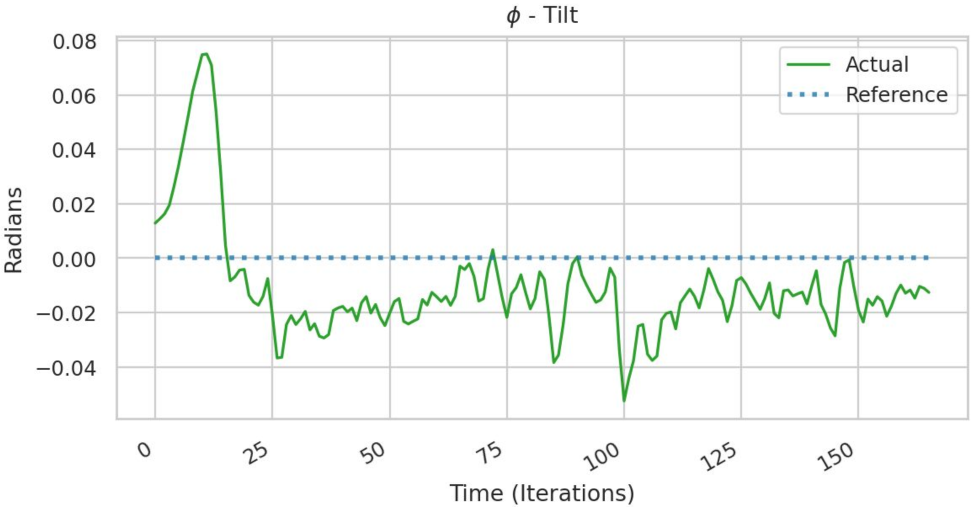


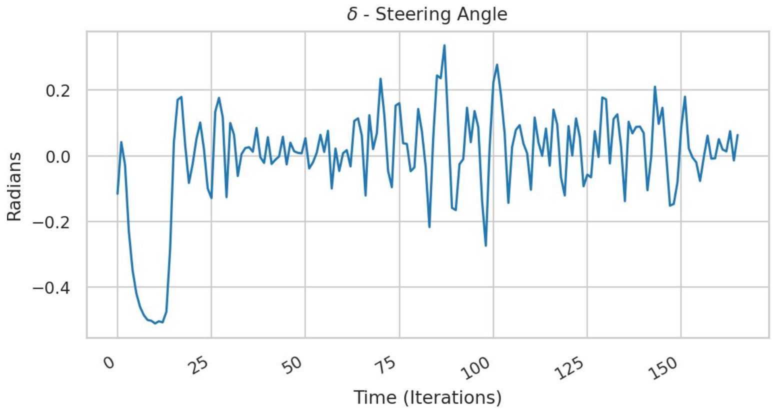




Results - RL Agent - Balance Control - Experiment 1

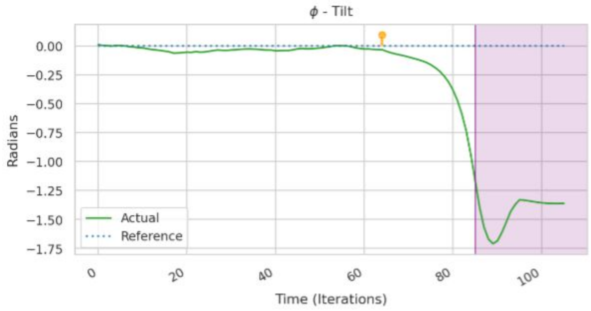
- Experiment 1 - Upright Control
- 40 rad/s -- 10 km/h



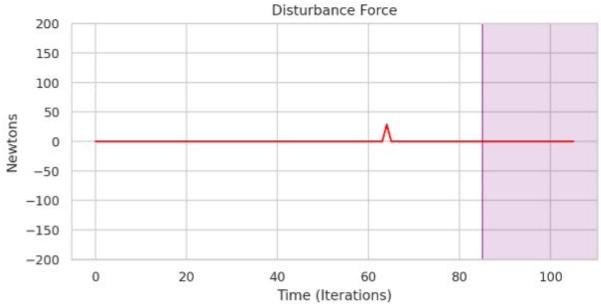


Results - RL Agent - Balance Control - Experiment 2

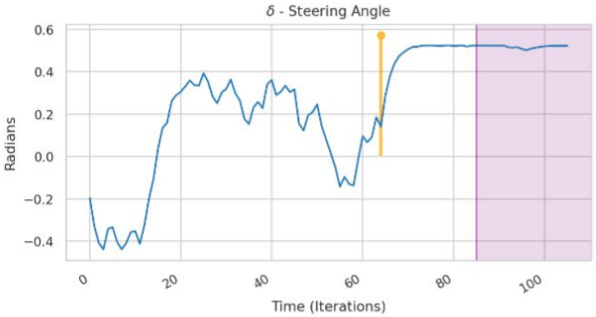
- Experiment 2 - Balance under disturbances - 10 rad/s (2.5 km/h)
  - Failed for a force of just 50 Newtons



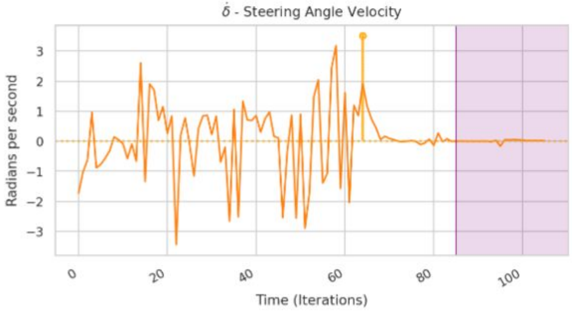
(a) Reference and actual tilt angle.



(b) Lateral disturbance force.



(c) Reference and actual steering angle.

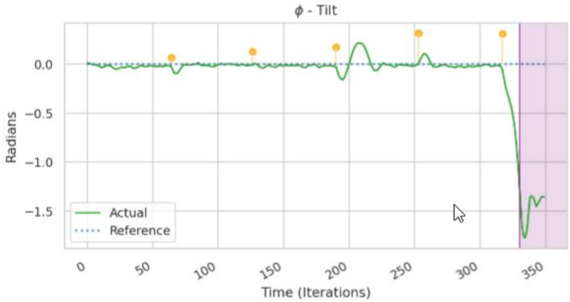


(d) Rate of change of the steering angle.

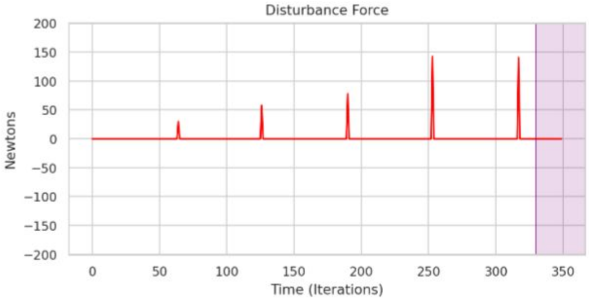
Results - RL Agent - Balance Control - Experiment 2

- Experiment 2 - Balance under disturbances - 20 rad/s (5 km/h)

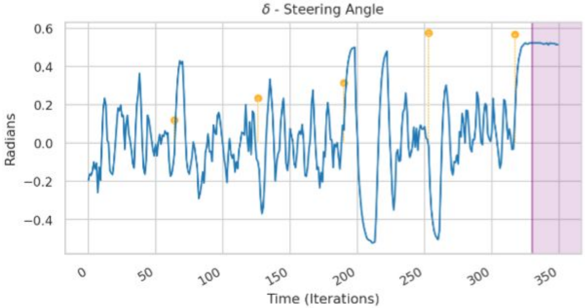
- Failed for a force of approximately 150 Newtons



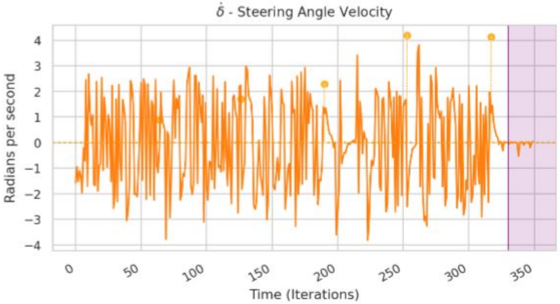
(a) Reference and actual tilt angle.



(b) Lateral disturbance force.



(c) Reference and actual steering angle.

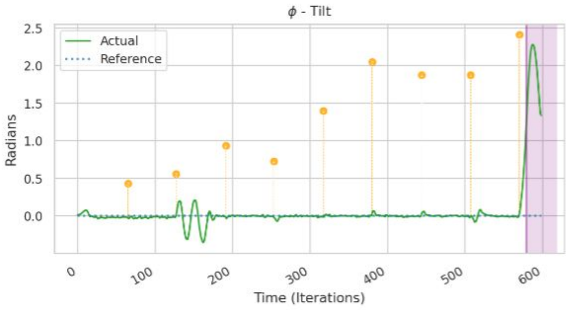


(d) Rate of change of the steering angle.

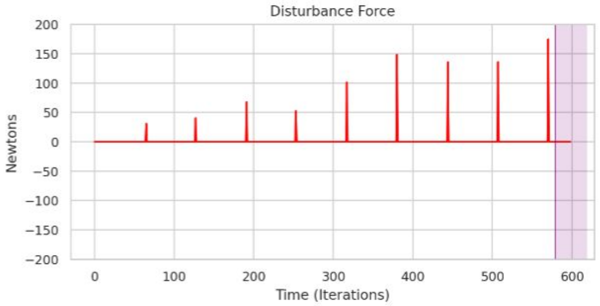
Results - RL Agent - Balance Control - Experiment 2

- Experiment 2 - Balance under disturbances - 40 rad/s (10 km/h)

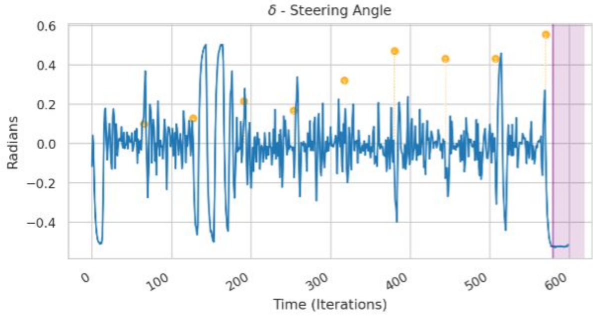
- Failed for a force of approximately 165 Newtons



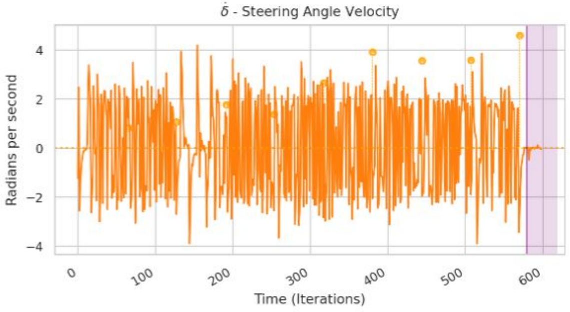
(a) Reference and actual tilt angle.



(b) Lateral disturbance force.

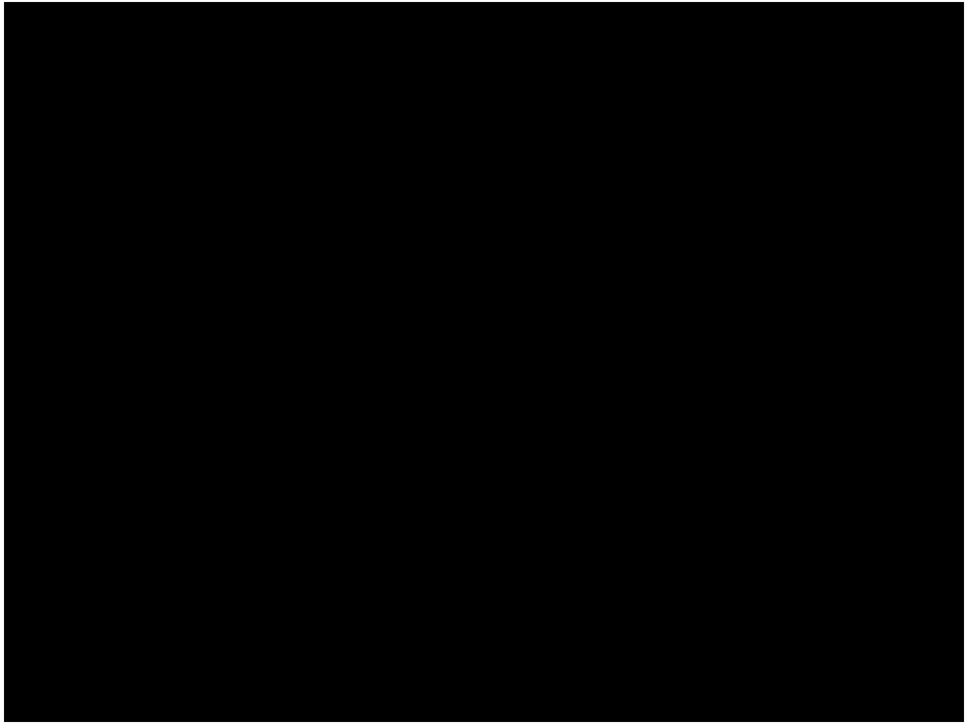


(c) Reference and actual steering angle.



(d) Rate of change of the steering angle.

Results - RL Agent - Balance Control - Video

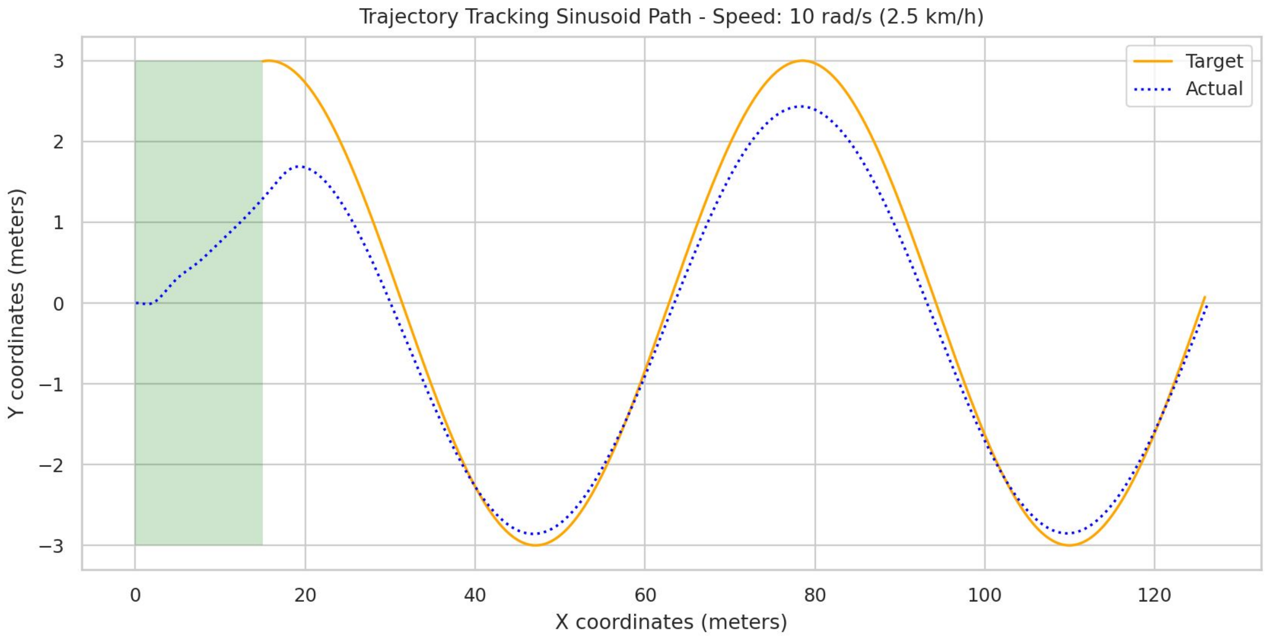


Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 10 rad/s (2.5 km/h)

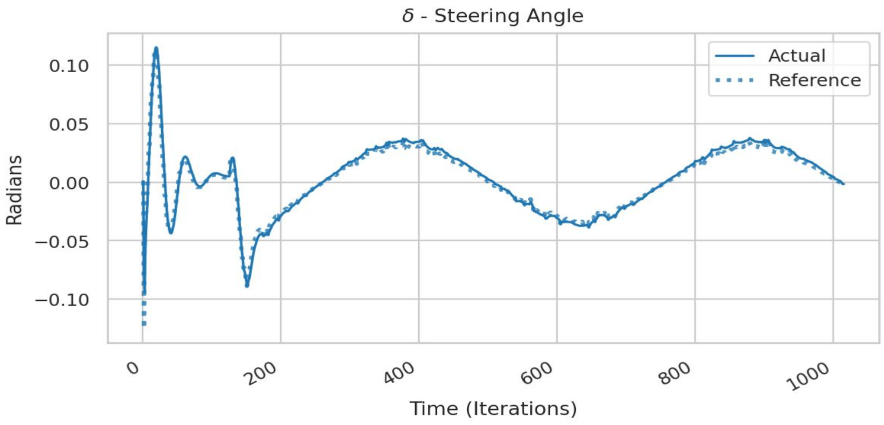
- Avg. Error:

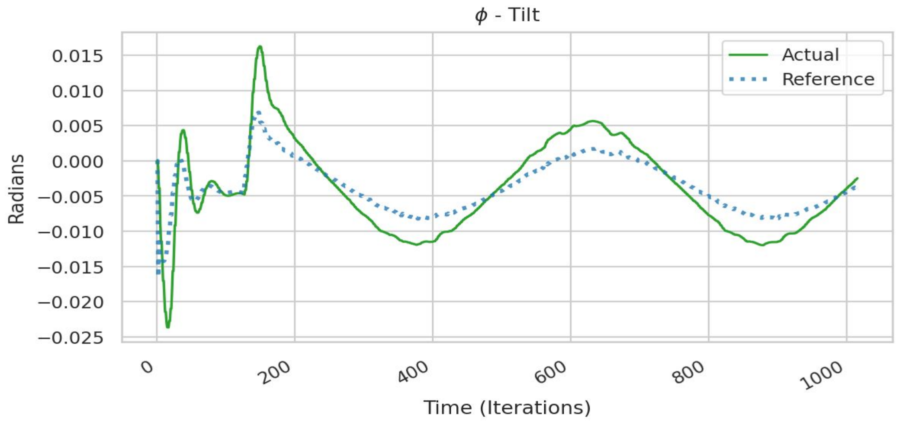
  0.3186 meters



Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 10 rad/s (2.5 km/h)



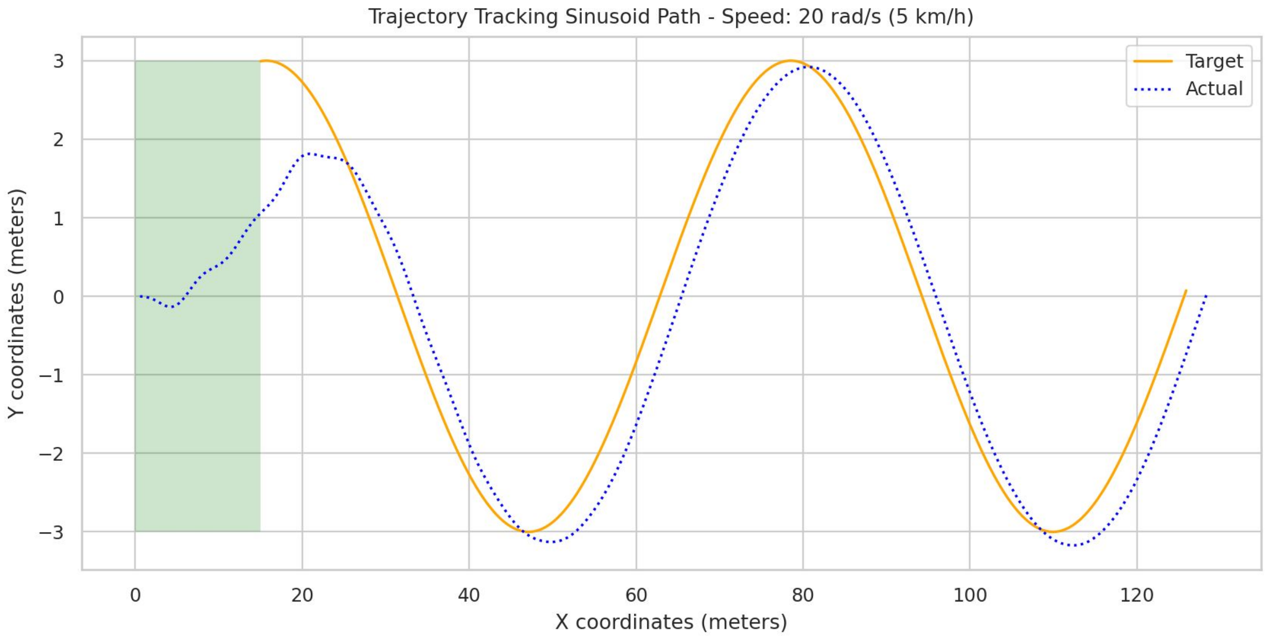


Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 20 rad/s (5 km/h)

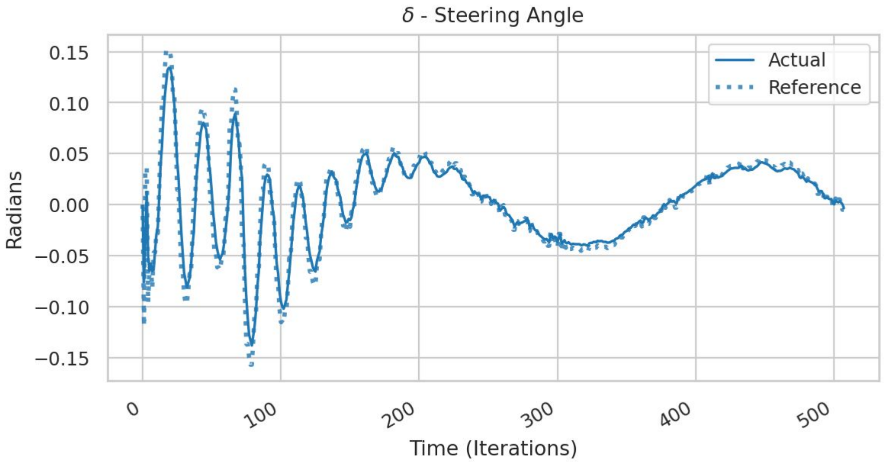
- Avg. Error:

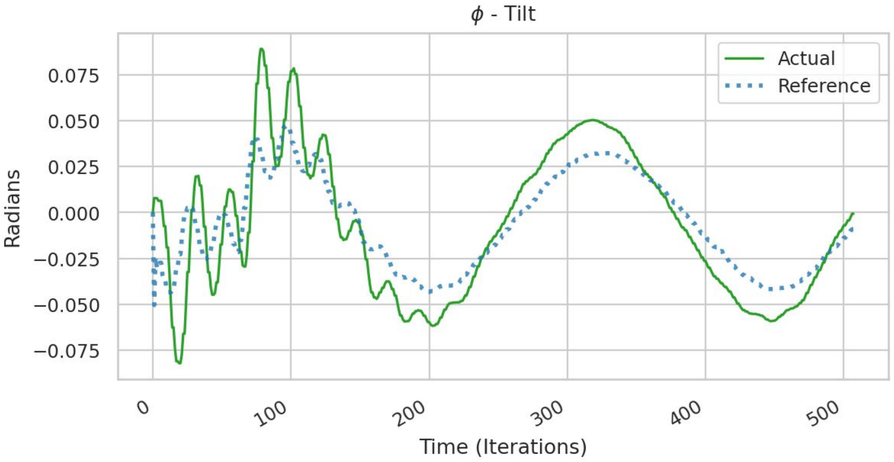
  0.4853 meters



Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 20 rad/s (5 km/h)



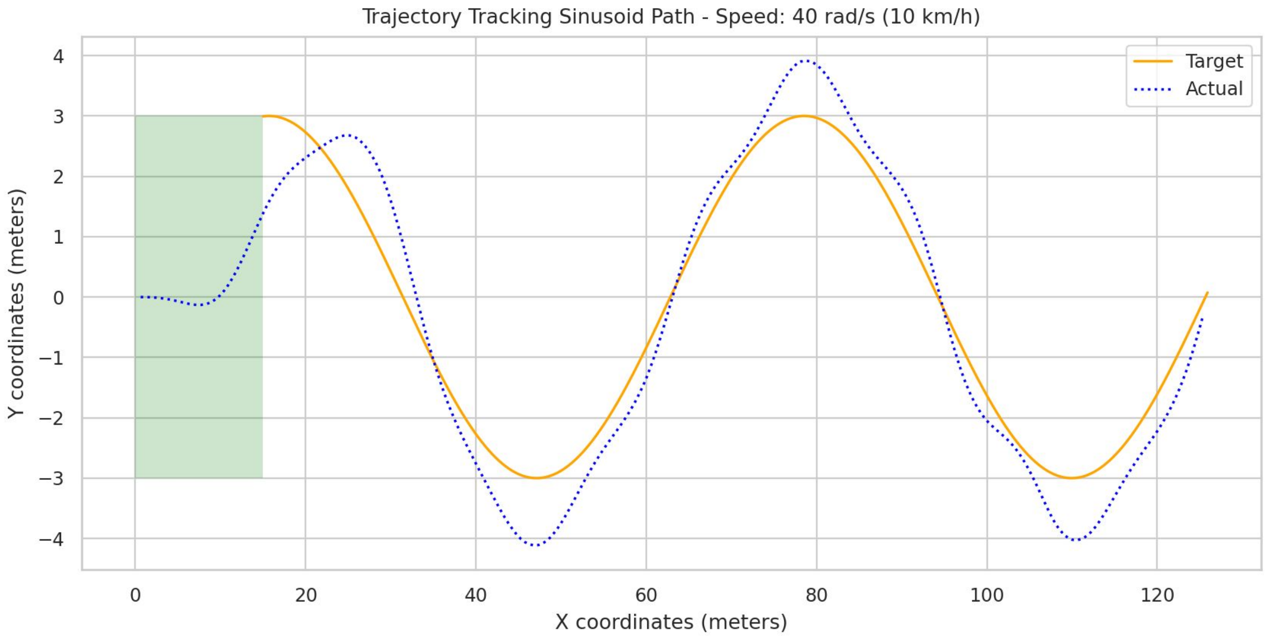


Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 40 rad/s (10 km/h)

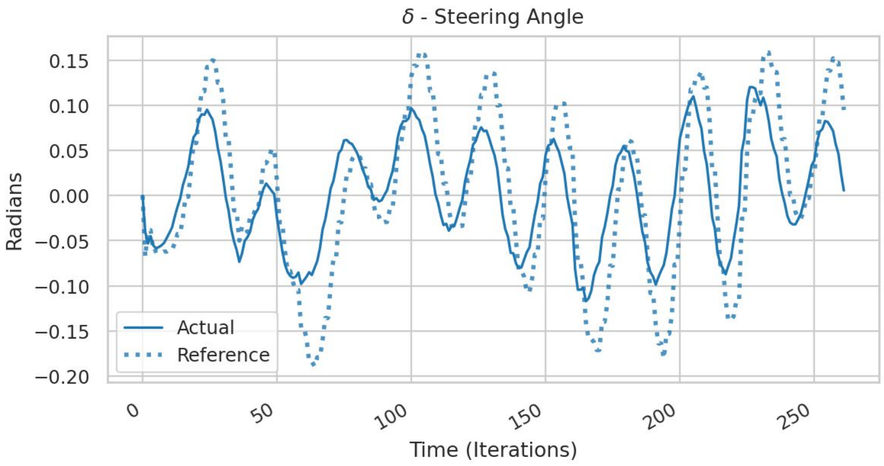
- Avg. Error:

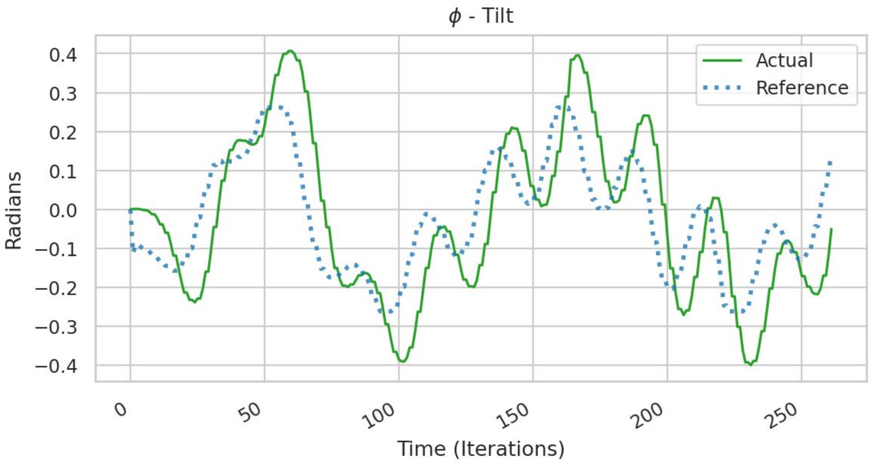
  0.5679 meters



Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 40 rad/s (10 km/h)



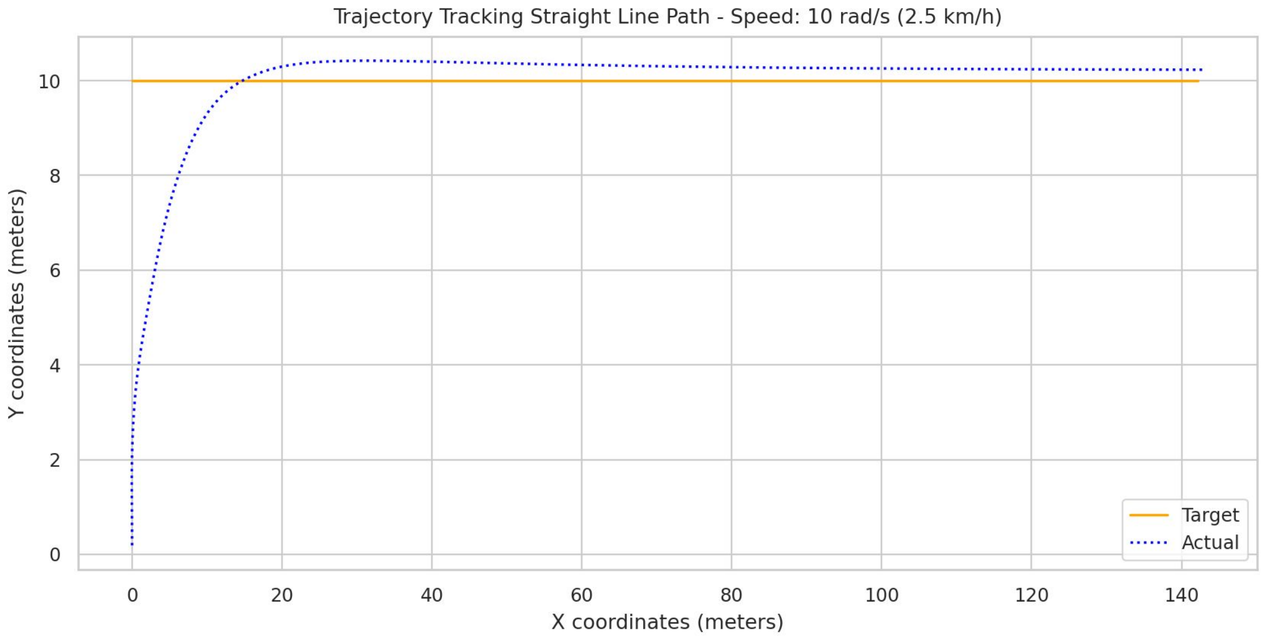


Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Straight Line Path - 10 rad/s (2.5 km/h)

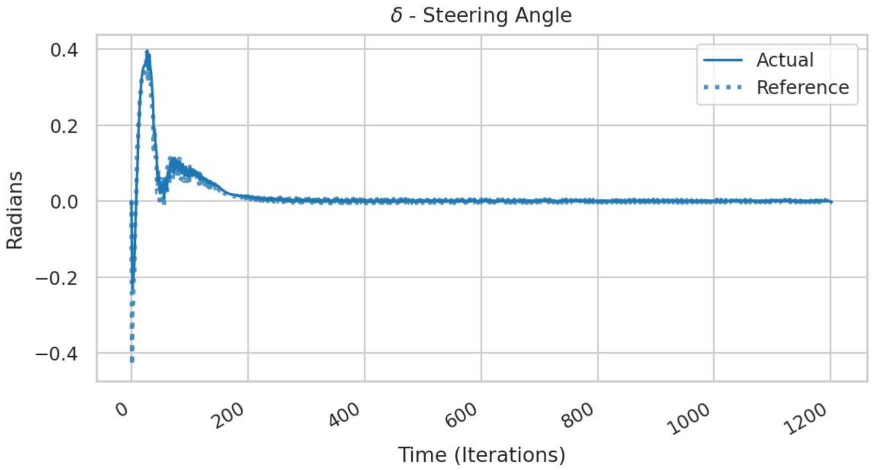
- Avg. Error:

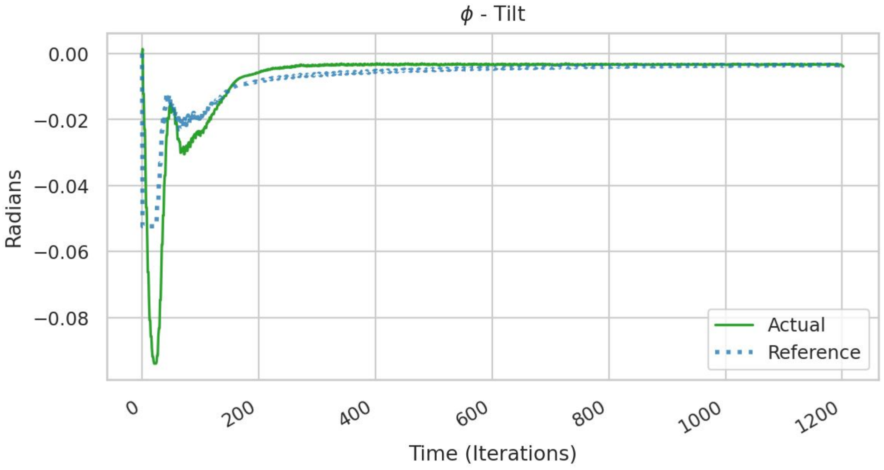
  0.7536 meters



Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Straight Line Path - 10 rad/s (2.5 km/h)



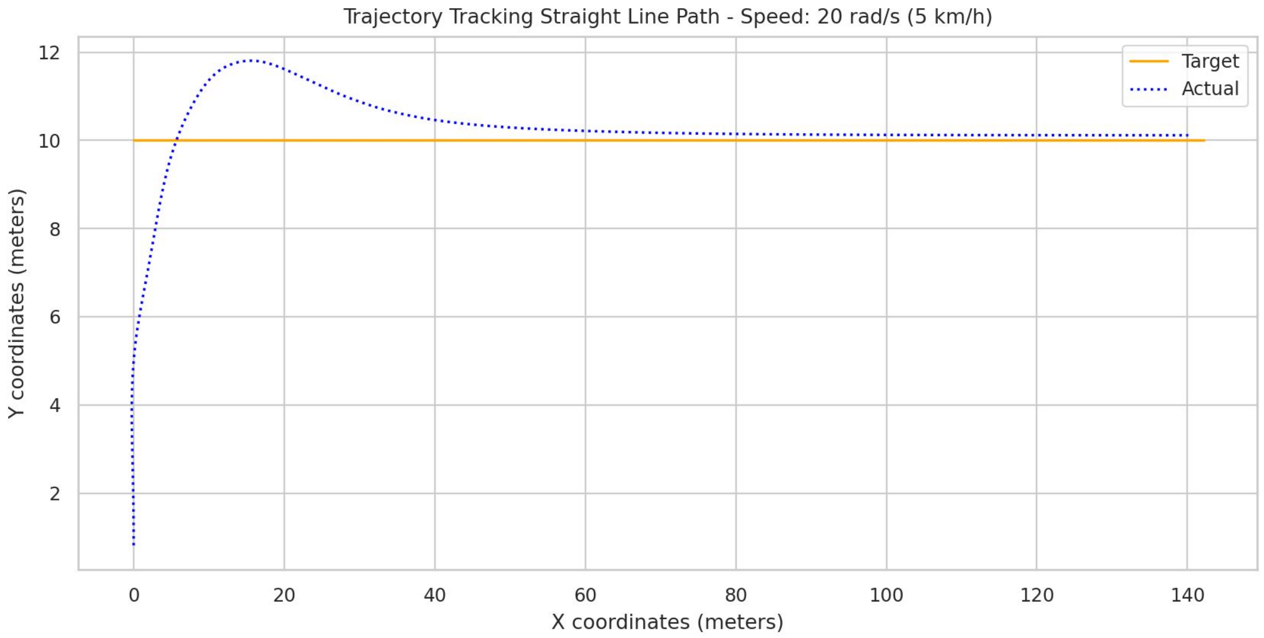


Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Straight Line Path - 20 rad/s (5 km/h)

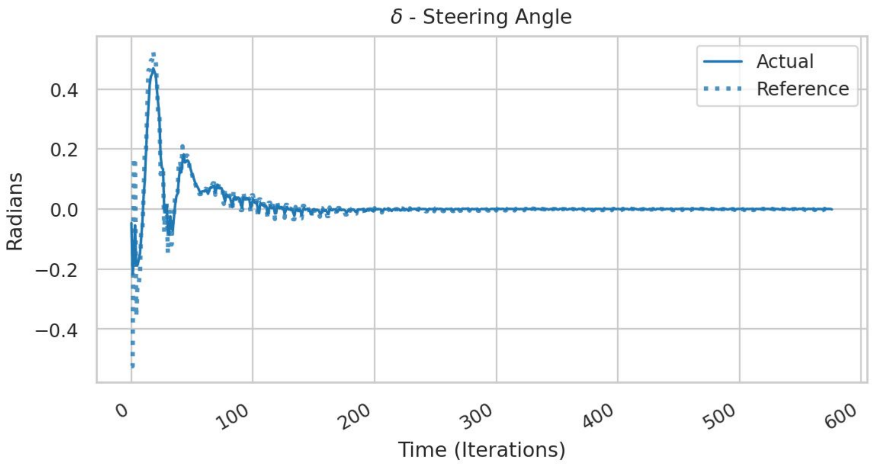
- Avg. Error:

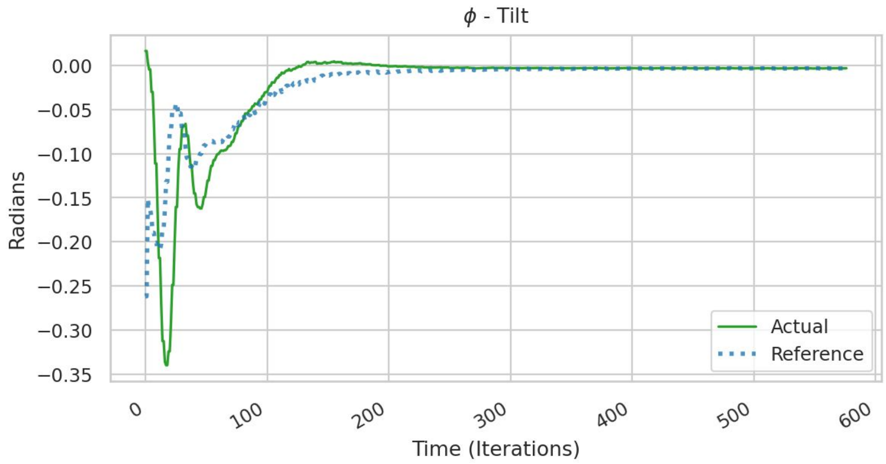
  0.8138 meters



Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Straight Line Path - 20 rad/s (5 km/h)



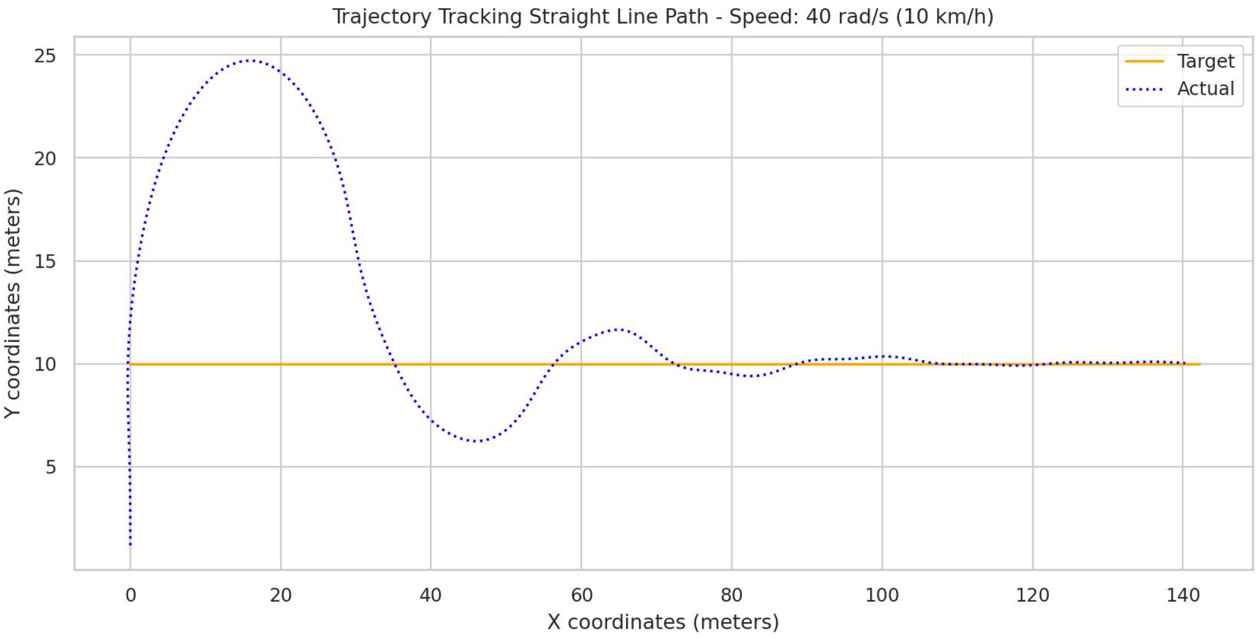


Results - PID Controller - Trajectory Control - Experiment 1

- Experiment 1 - Straight Line Path - 40 rad/s (10 km/h)

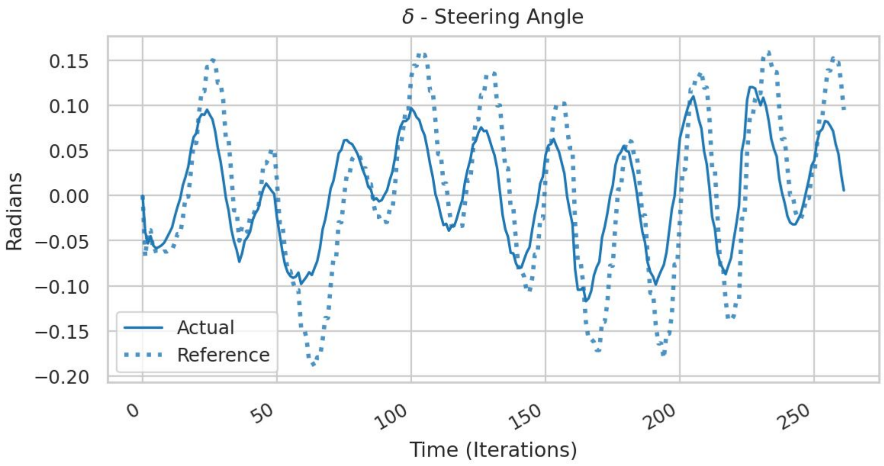
- Avg. Error:

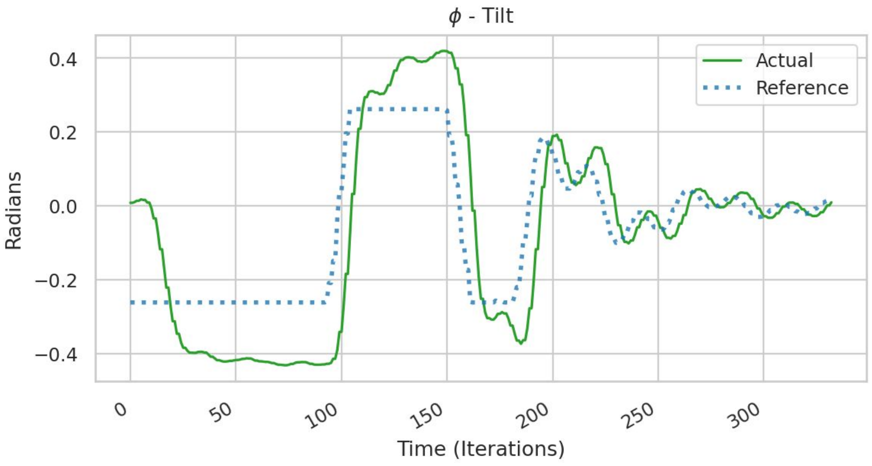
  3.6039 meters



Results - PID Controller - Trajectory Control - Experiment 1

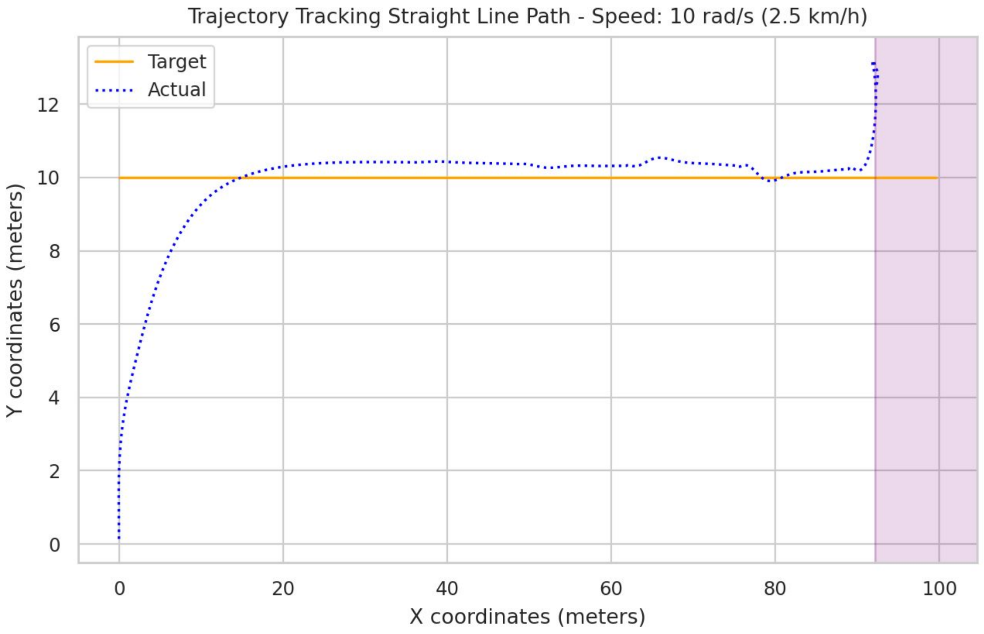
- Experiment 1 - Straight Line Path - 40 rad/s (10 km/h)

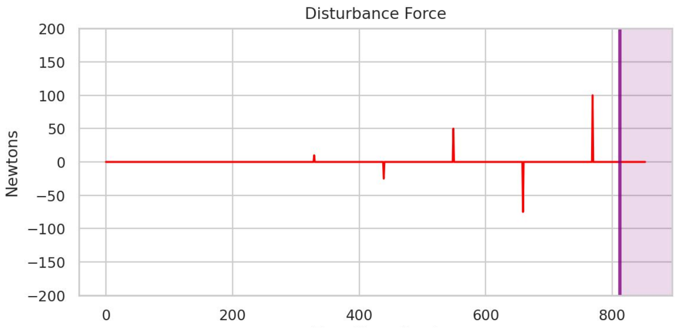


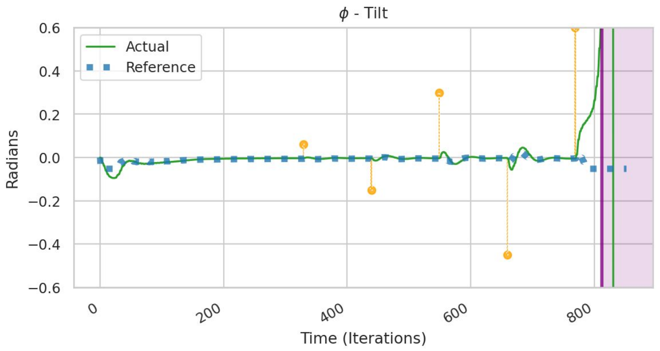


Results - PID Controller - Trajectory Control - Experiment 2

- Experiment 2 - Straight Line Path under Disturbances - 10 rad/s (2.5 km/h)

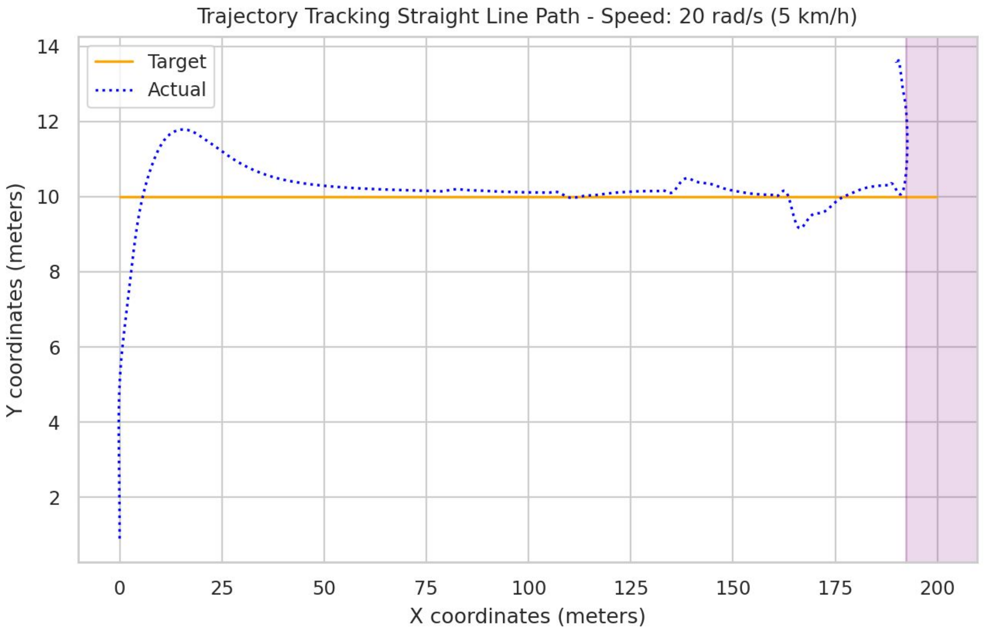


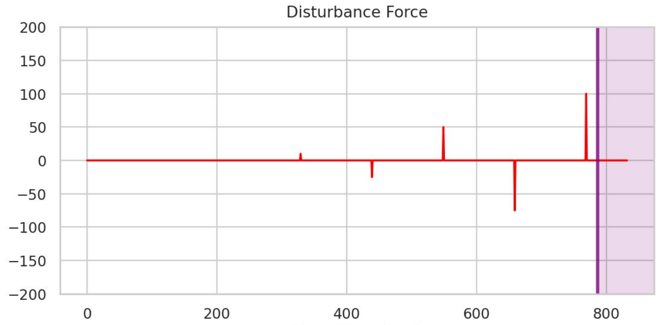


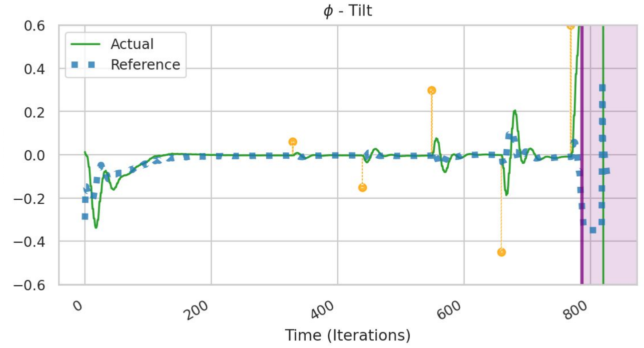


Results - PID Controller - Trajectory Control - Experiment 2

- Experiment 2 - Straight Line Path under Disturbances - 20 rad/s (5 km/h)

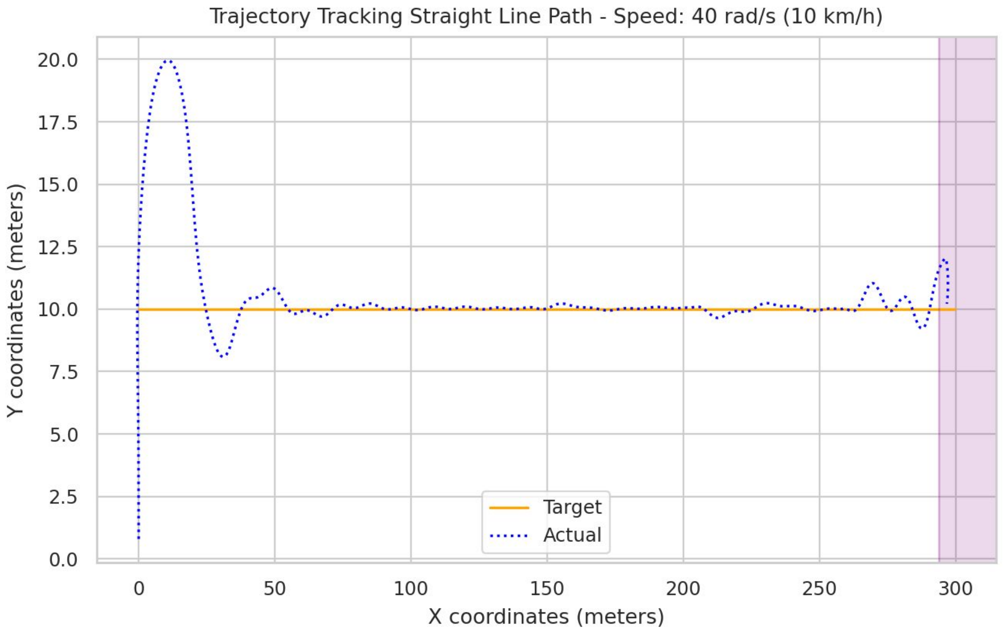


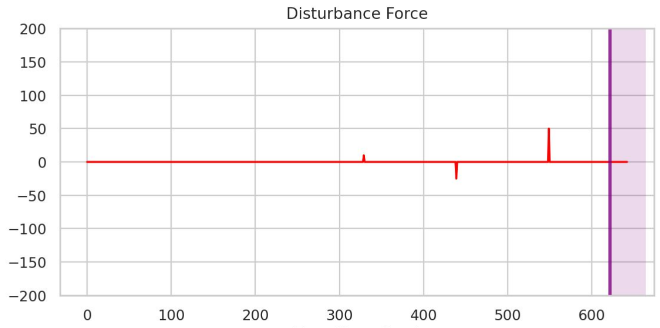


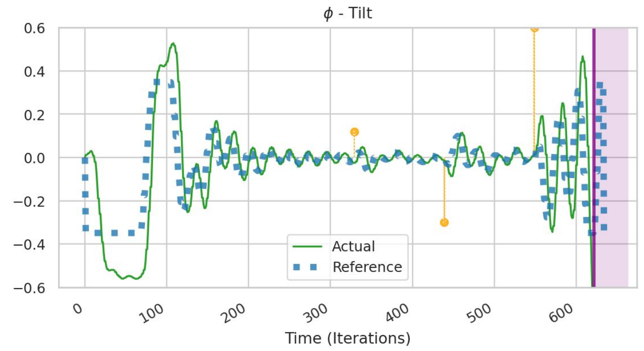


Results - PID Controller - Trajectory Control - Experiment 2

- Experiment 2 - Straight Line Path under Disturbances - 40 rad/s (10 km/h)

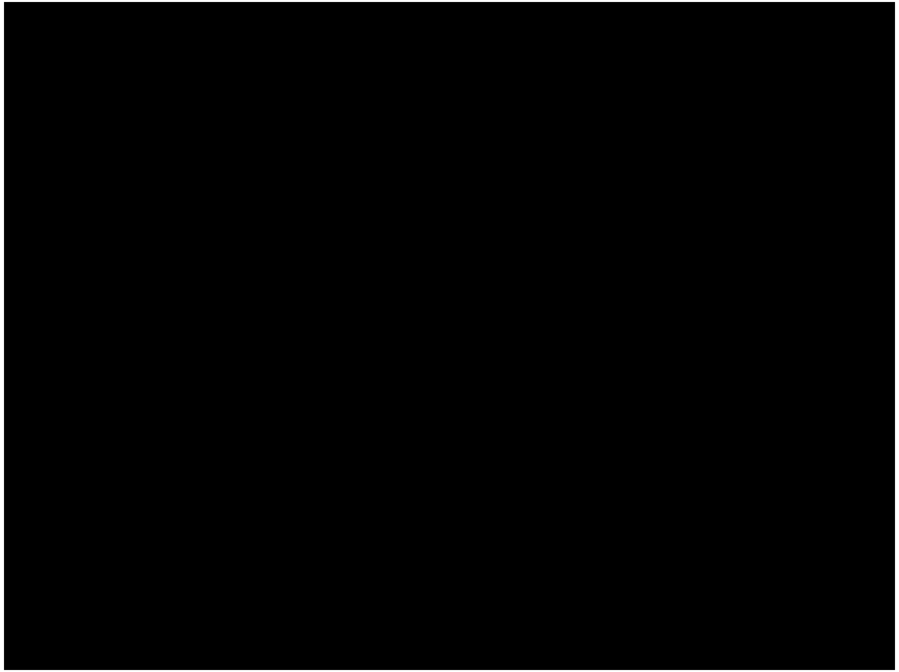






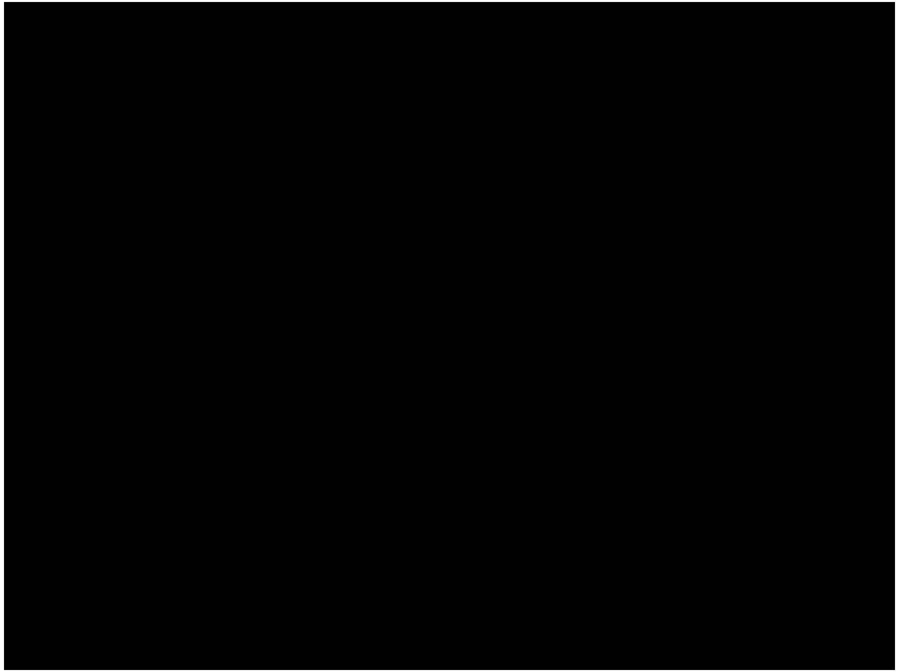
Results - PID Controller - Trajectory Control - Video

- Tracking the straight line path.



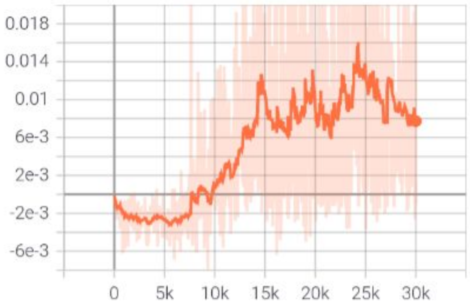
Results - PID Controller - Trajectory Control - Video

- Tracking the sinusoidal path.

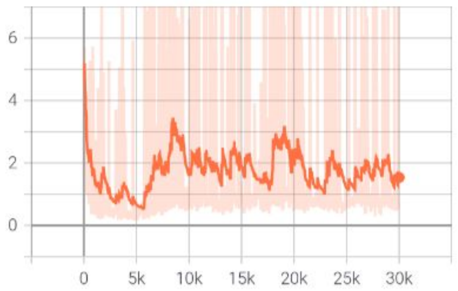


Results - RL Agent - Trajectory Control - Graphs

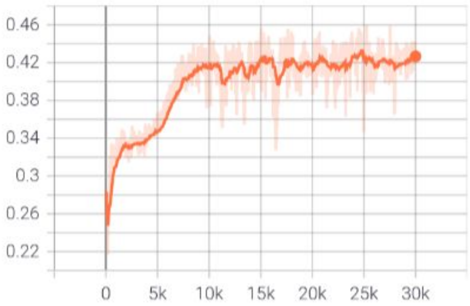
- Training graphs shows progress
  - Mean reward improves.
  - Mean episode length increases.
  - Metrics don't approach maximum value but a good policy is still learned.



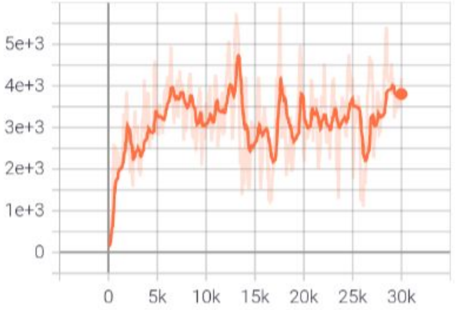
(a) PPO's surrogate loss over 30k training iterations.



(b) The value function loss over 30k training iterations.



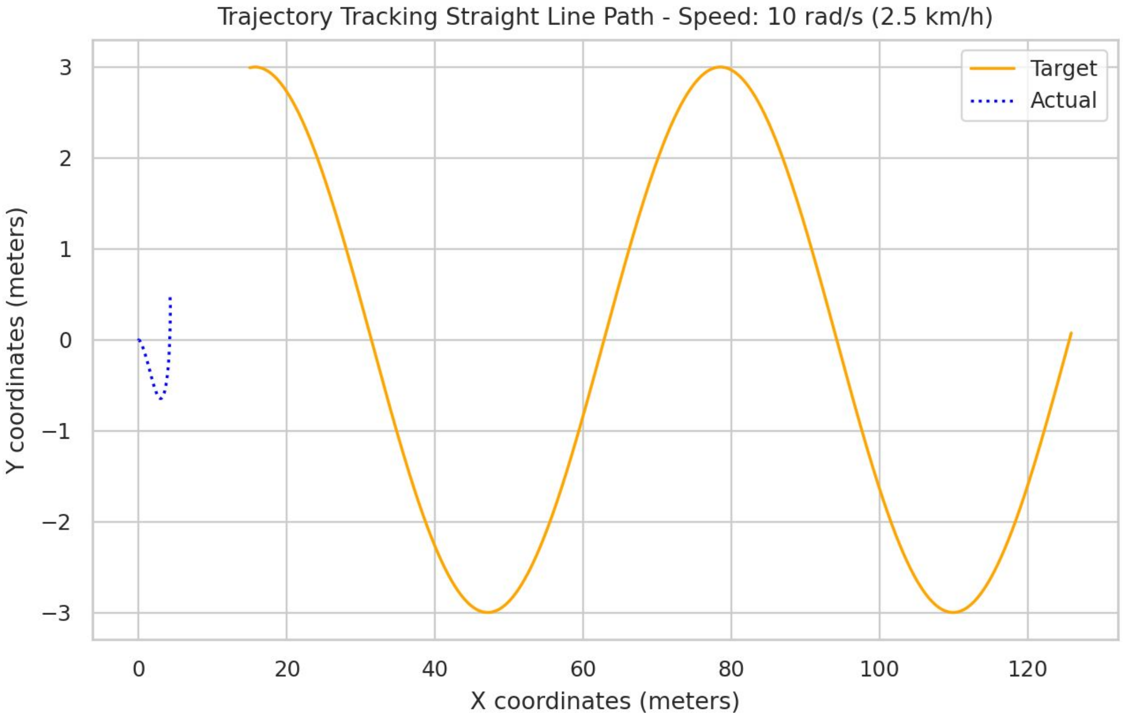
(c) The mean reward per step increases as training progresses.



(d) The mean episode length increases as training progress.

Results - RL Agent - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 10 rad/s (2.5 km/h)
- At the lowest velocity the agent fails

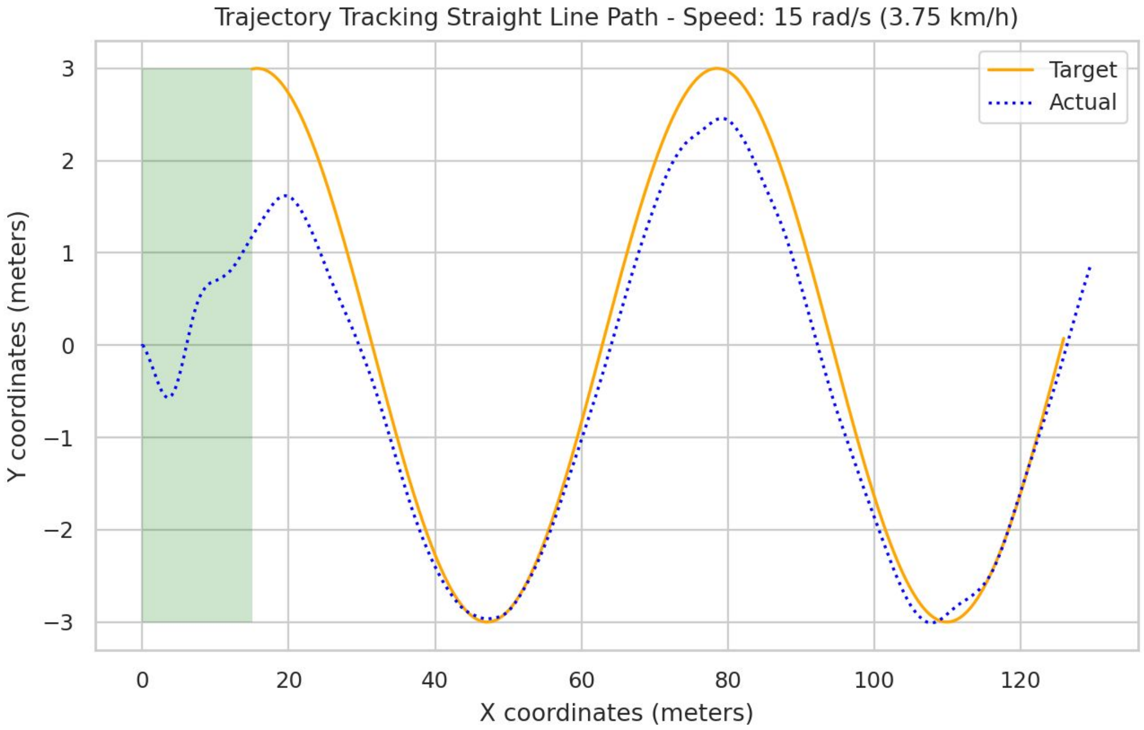


Results - RL Agent - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 15 rad/s (3.75 km/h)

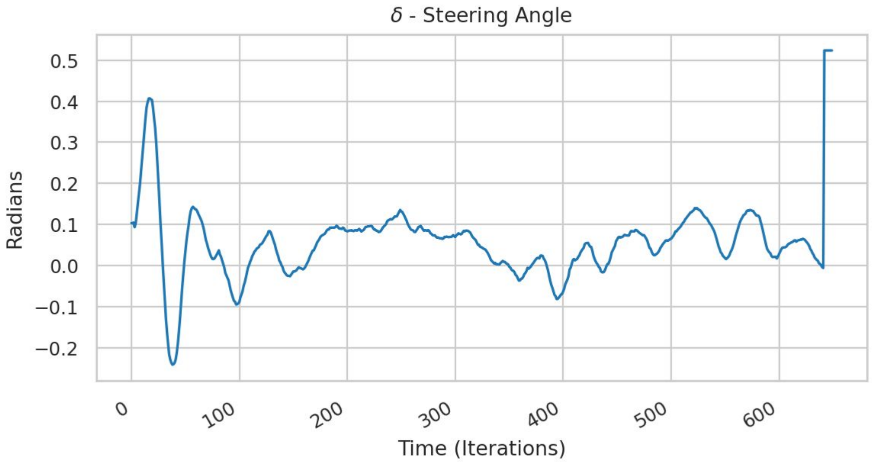
- Increasing velocity to 15 rad/s the RL agent can succeed

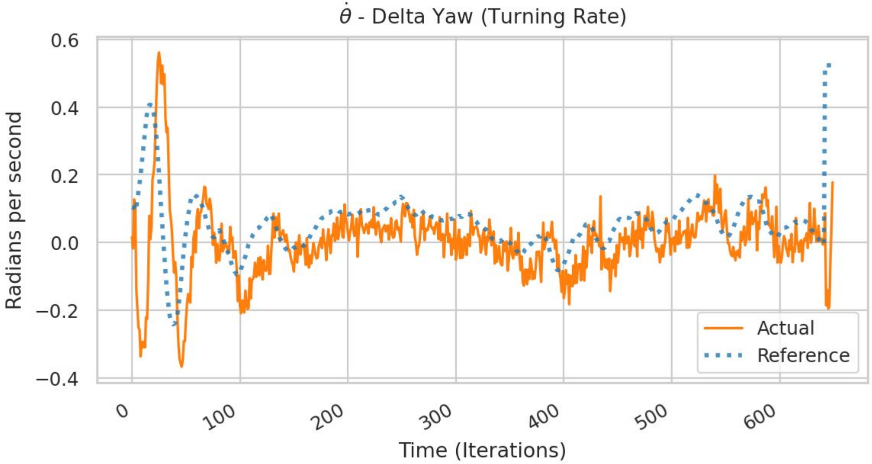
- Avg. Error: 0.3730 meters



Results - RL Agent - Trajectory Control - Experiment 1

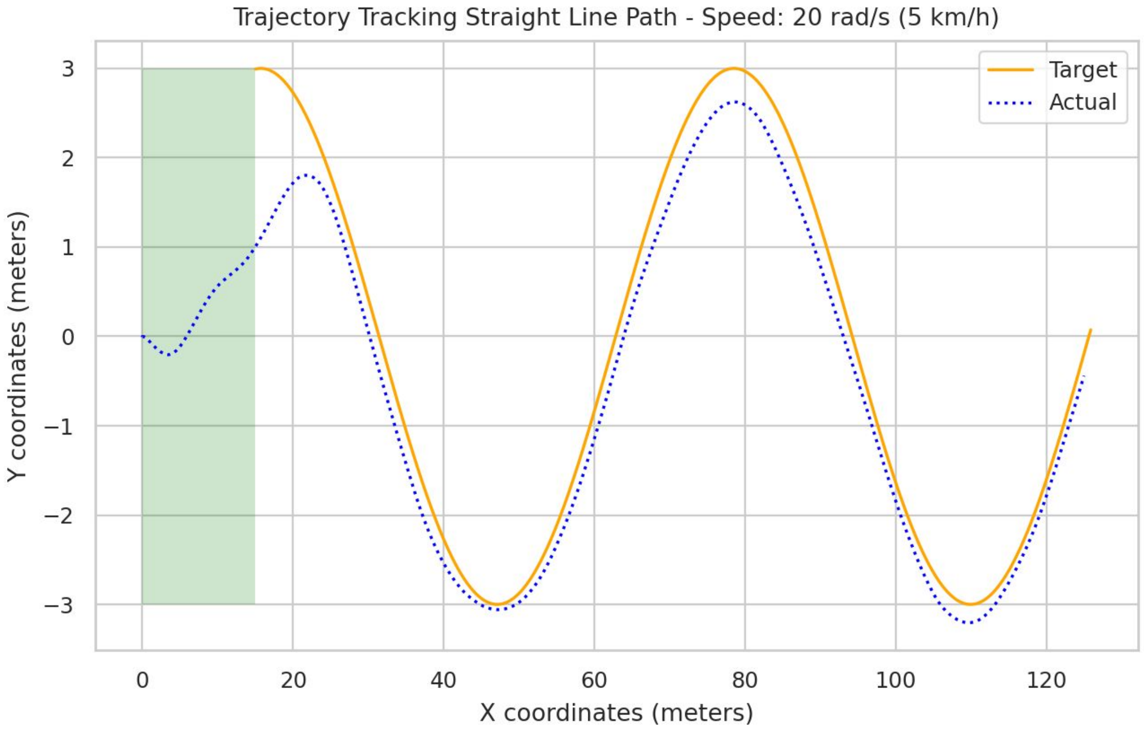
- Experiment 1 - Sinusoidal Path - 15 rad/s (3.75 km/h)





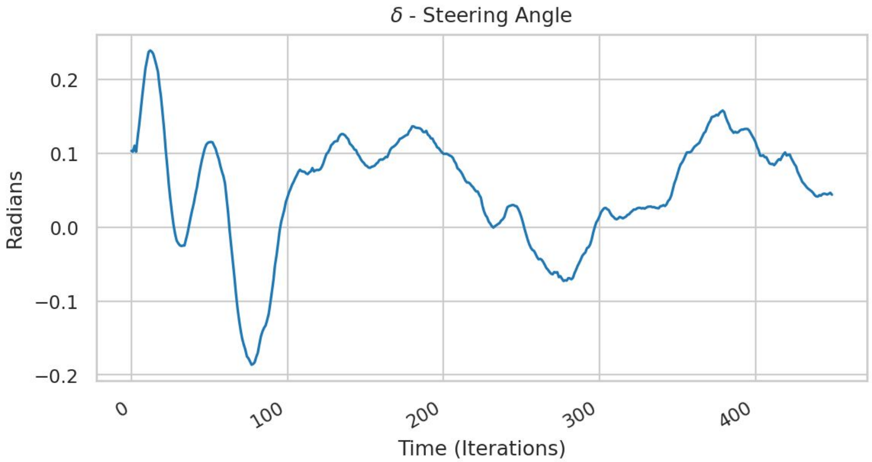
Results - RL Agent - Trajectory Control - Experiment 1

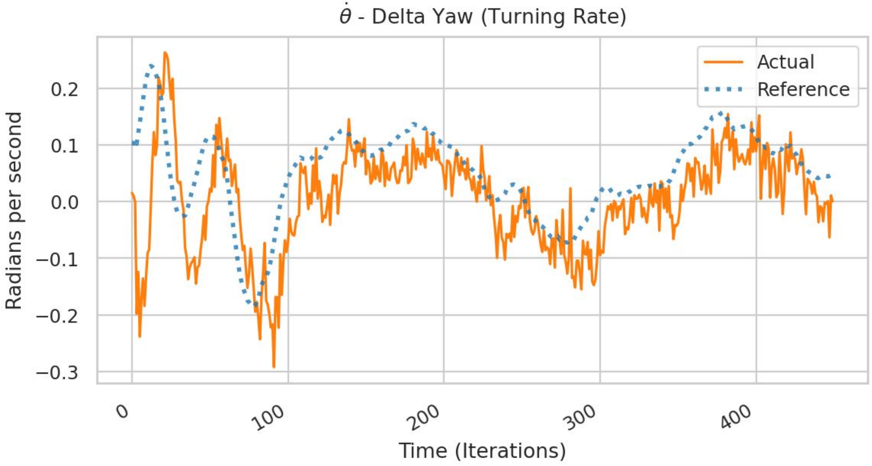
- Experiment 1 - Sinusoidal Path - 20 rad/s (5 km/h)
- Avg. Error: 0.3492 meters



Results - RL Agent - Trajectory Control - Experiment 1

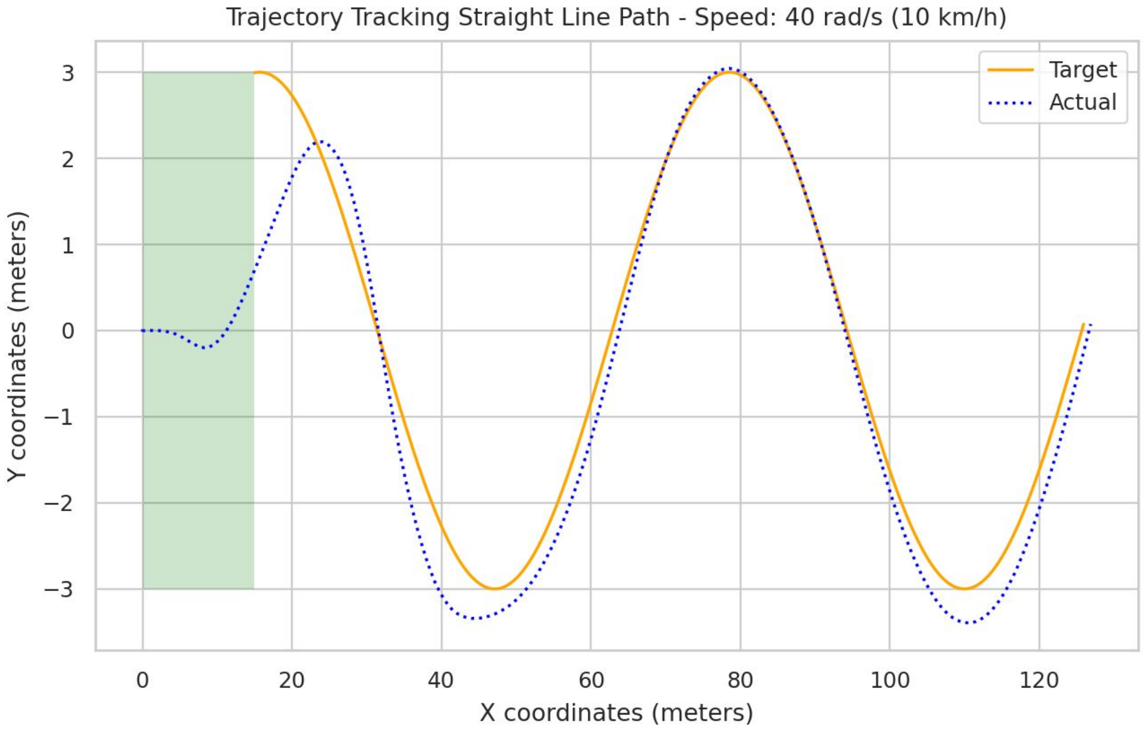
- Experiment 1 - Sinusoidal Path - 20 rad/s (5 km/h)





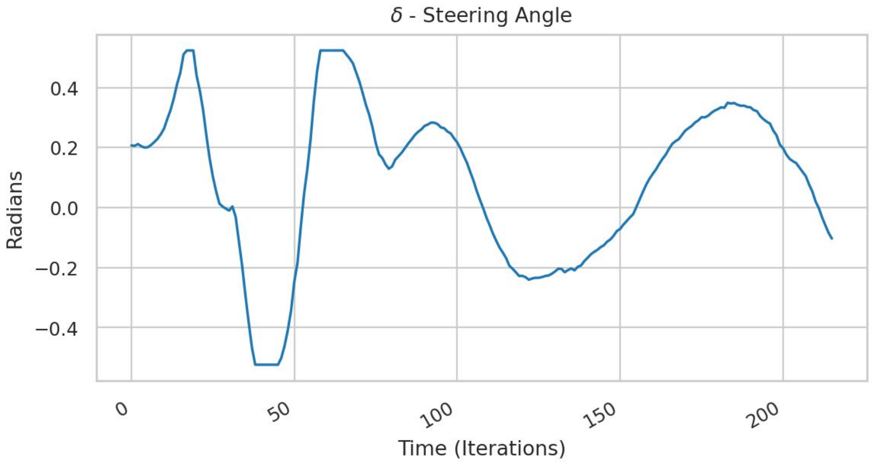
Results - RL Agent - Trajectory Control - Experiment 1

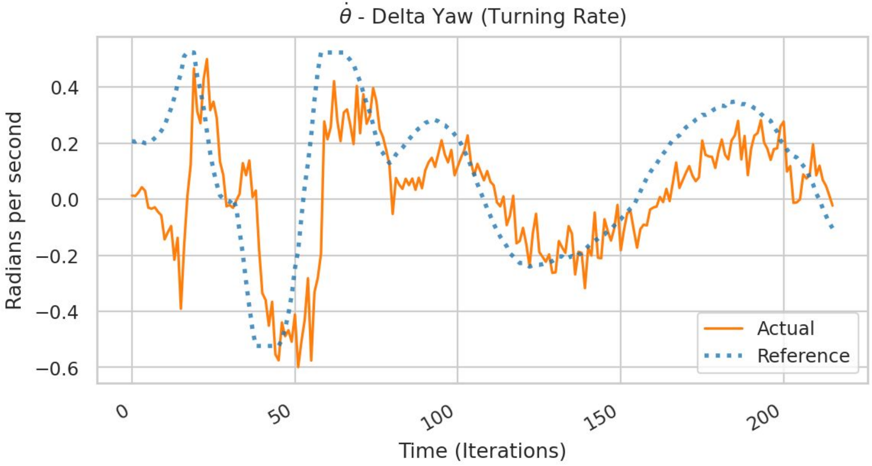
- Experiment 1 - Sinusoidal Path - 40 rad/s (10 km/h)
- Avg. Error: 0.3594 meters



Results - RL Agent - Trajectory Control - Experiment 1

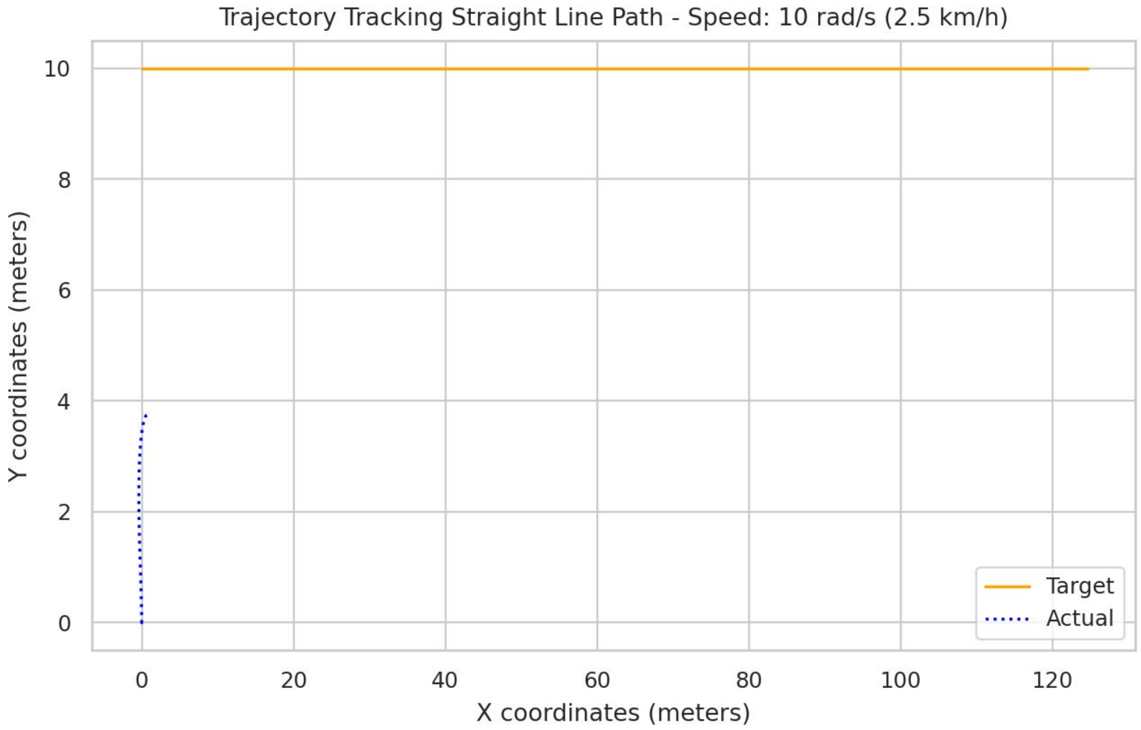
- Experiment 1 - Sinusoidal Path - 40 rad/s (10 km/h)





Results - RL Agent - Trajectory Control - Experiment 1

- Experiment 1 - Straight Line Path - 10 rad/s (2.5 km/h)
- At the lowest velocity the agent fails

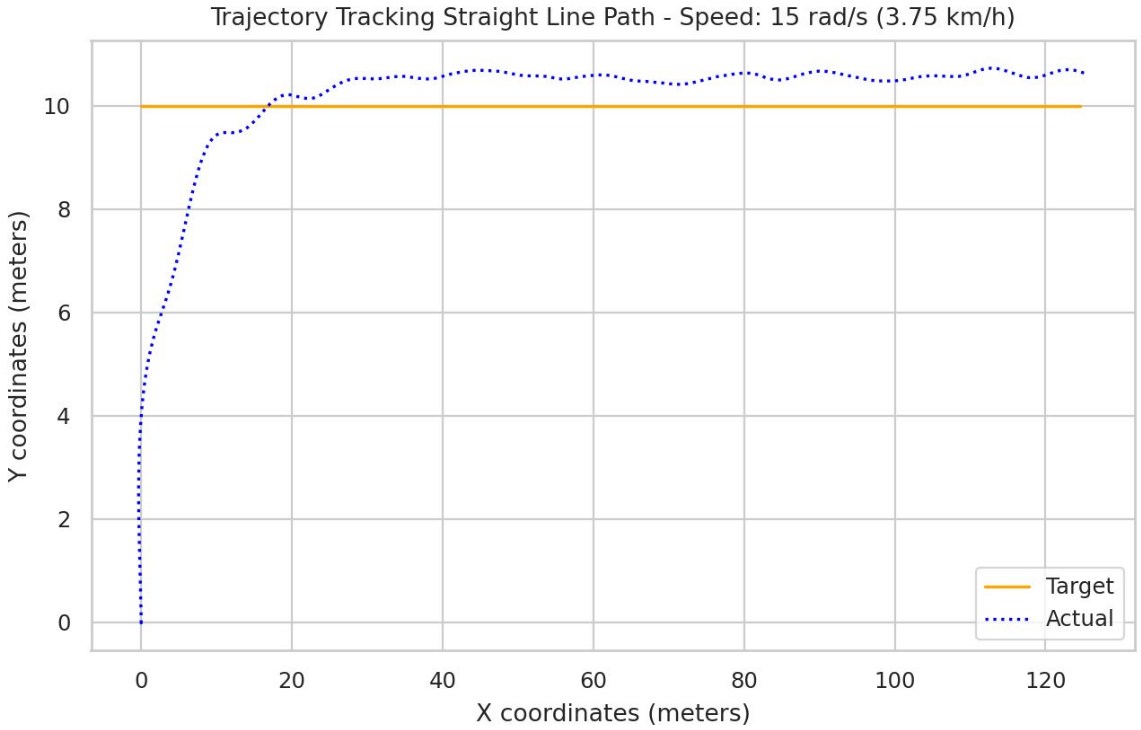


Results - RL Agent - Trajectory Control - Experiment 1

- Experiment 1 - Straight Line Path - 15 rad/s (3.75 km/h)

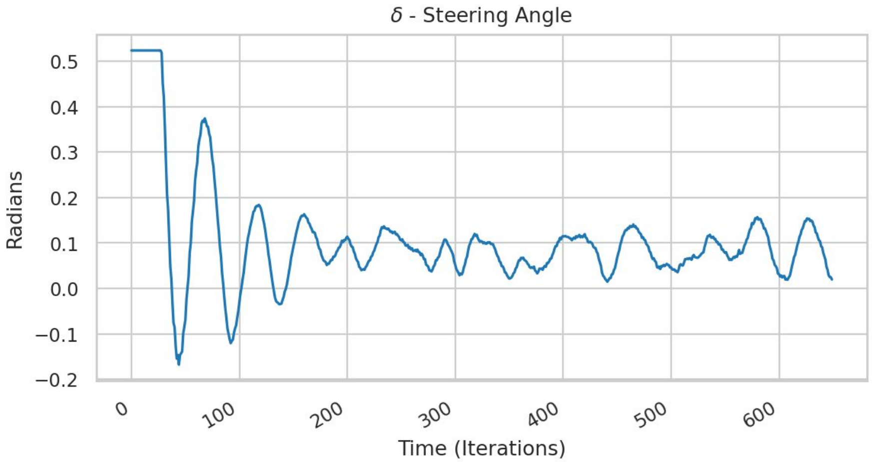
- Increasing velocity to 15 rad/s the RL agent can succeed

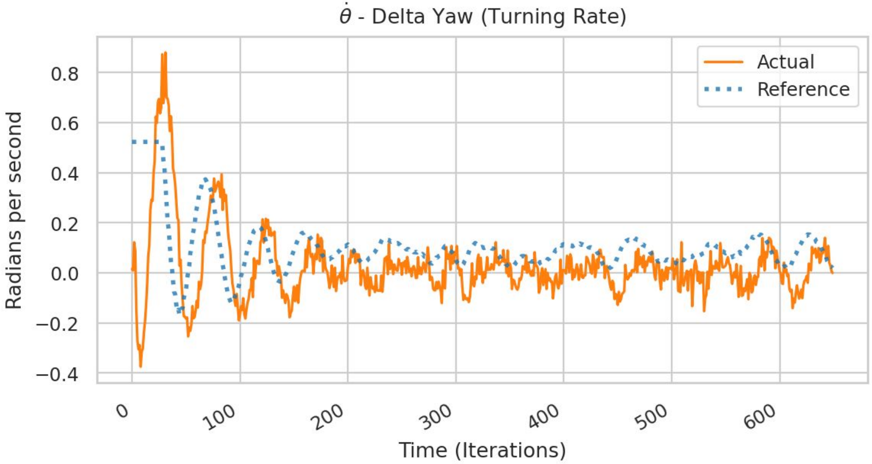
- Avg. Error: 1.0950 meters



Results - RL Agent - Trajectory Control - Experiment 1

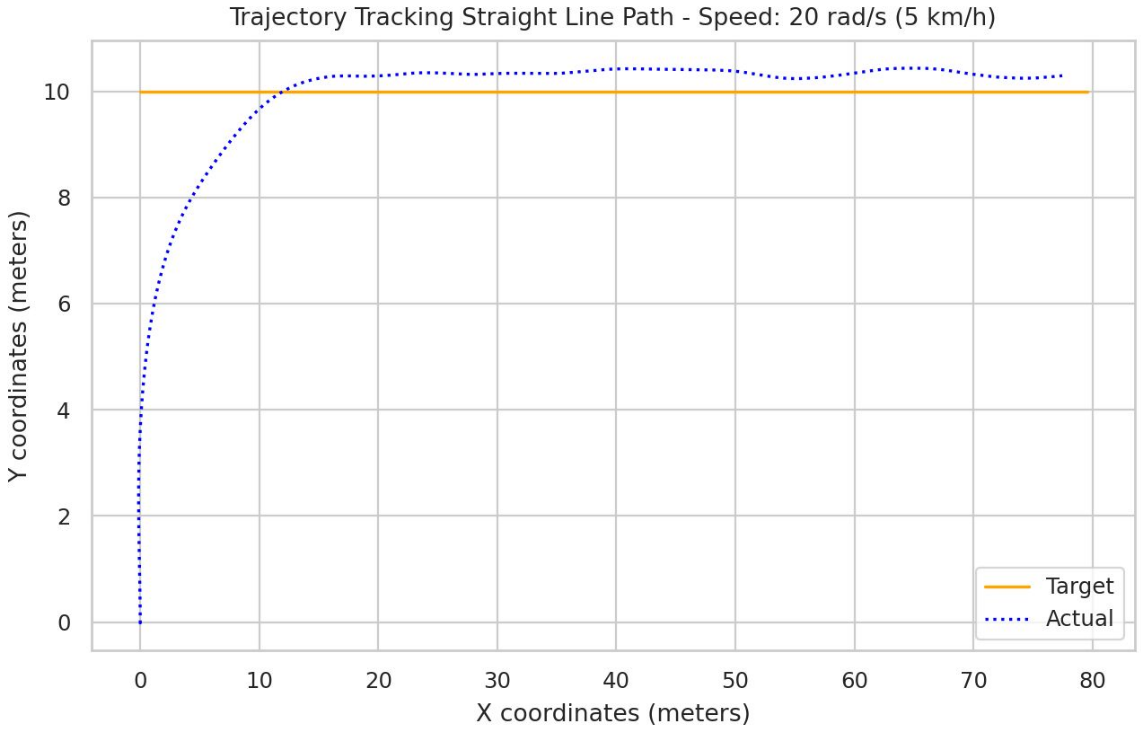
- Experiment 1 - Sinusoidal Path - 15 rad/s (3.75 km/h)





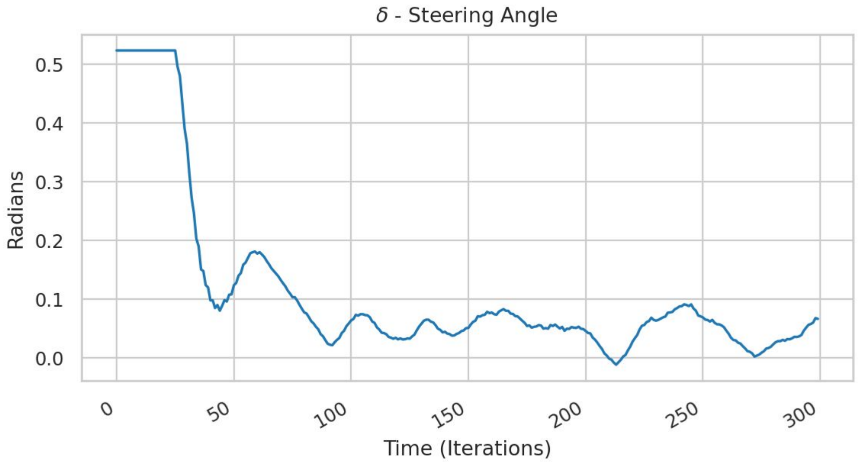
Results - RL Agent - Trajectory Control - Experiment 1

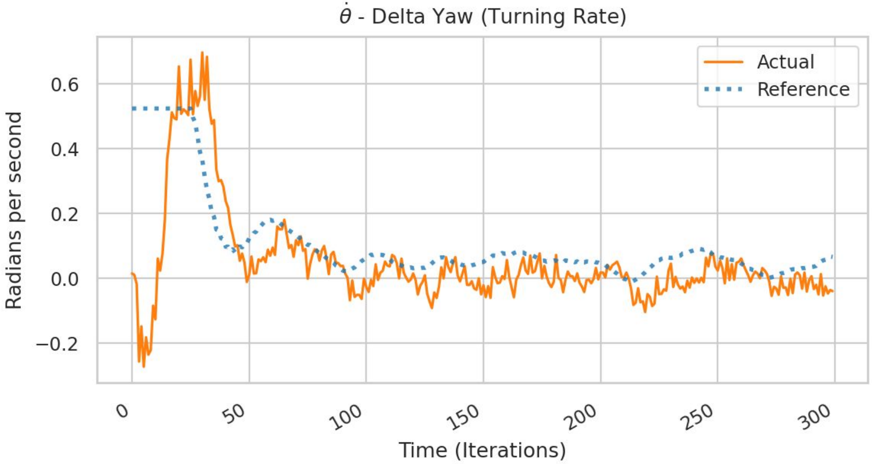
- Experiment 1 - Straight Line Path - 20 rad/s (5 km/h)
- Avg. Error: 1.2225 meters



Results - RL Agent - Trajectory Control - Experiment 1

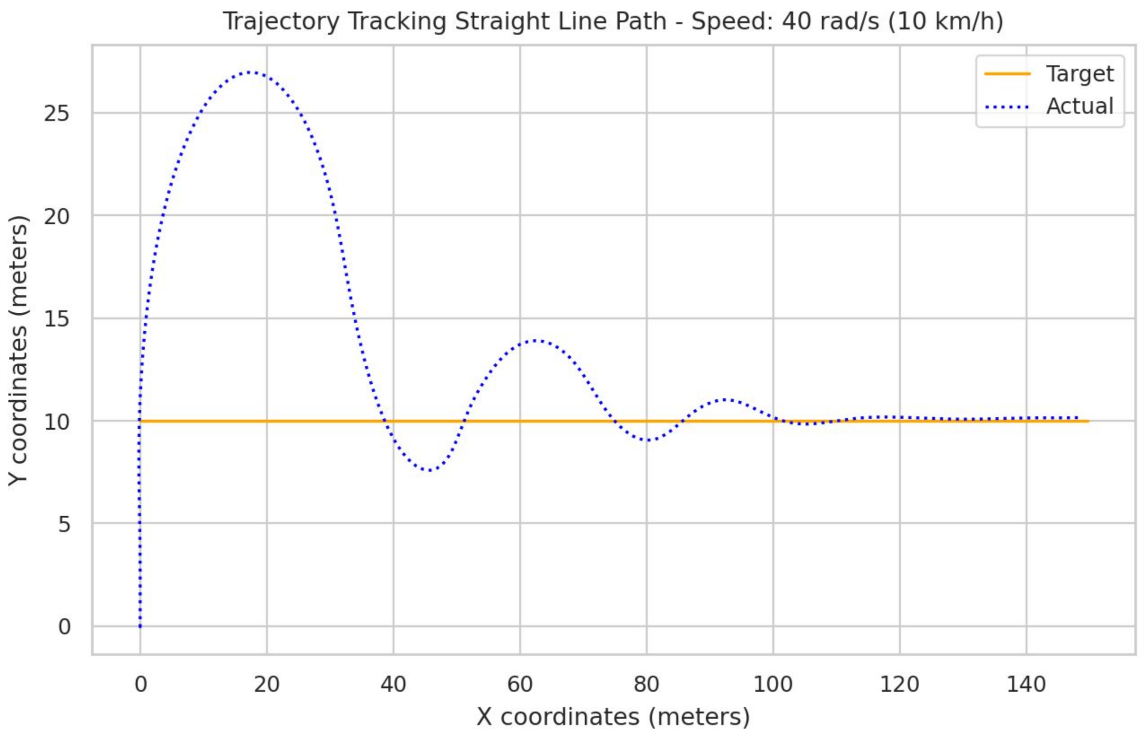
- Experiment 1 - Sinusoidal Path - 20 rad/s (5 km/h)





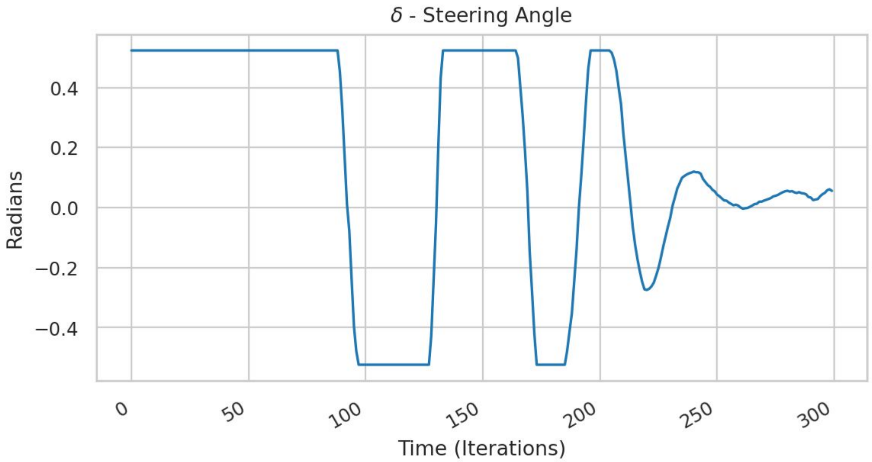
Results - RL Agent - Trajectory Control - Experiment 1

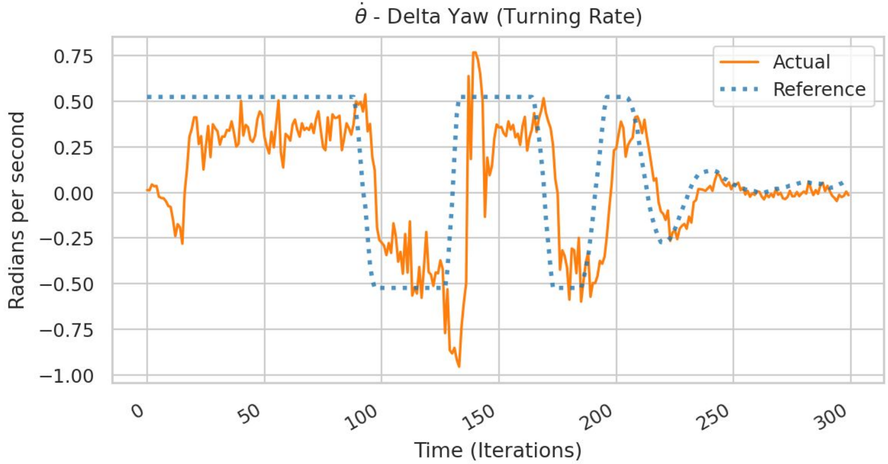
- Experiment 1 - Sinusoidal Path - 40 rad/s (10 km/h)
- Avg. Error: 4.2875 meters



Results - RL Agent - Trajectory Control - Experiment 1

- Experiment 1 - Sinusoidal Path - 40 rad/s (10 km/h)

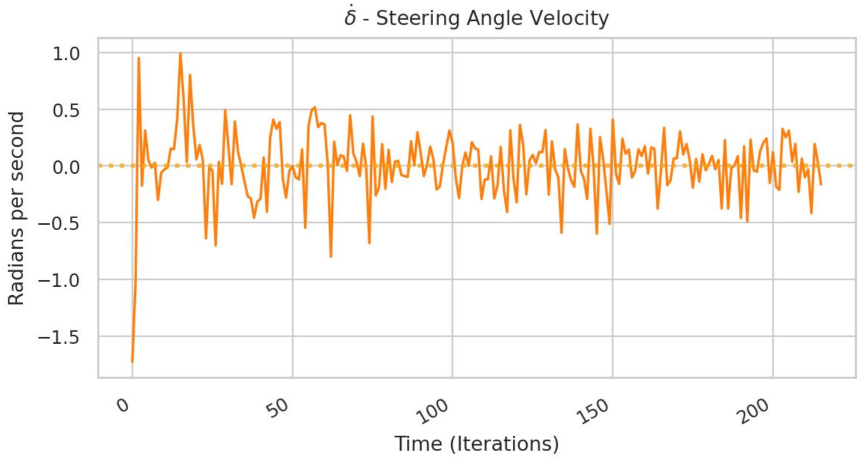




Results - RL Agent - Trajectory Control - Experiment 1

- Velocity of Steering Angle much less compared to Balance Control agent.

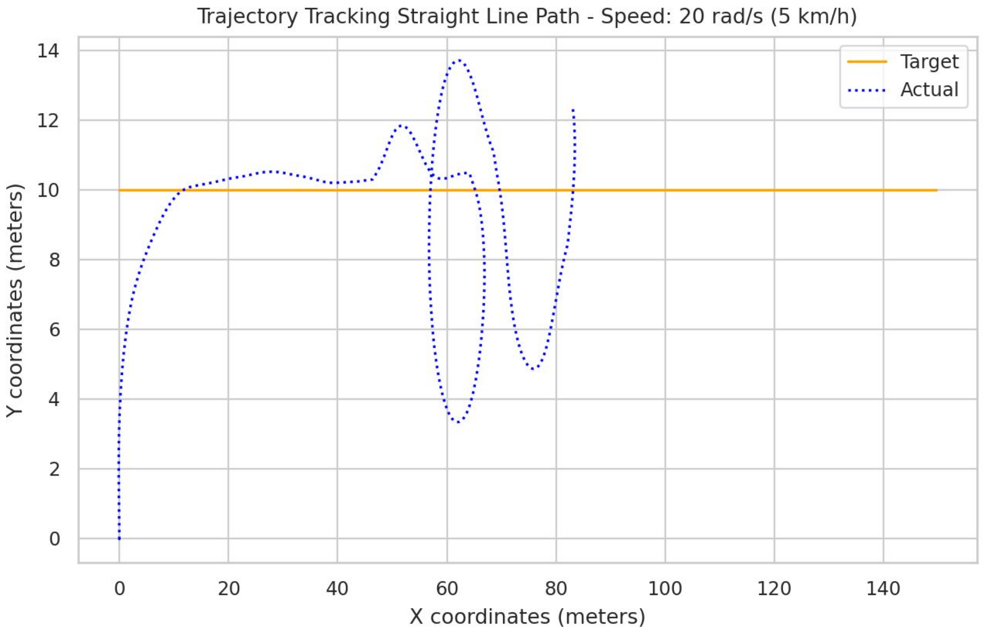
ΔYaw Agent

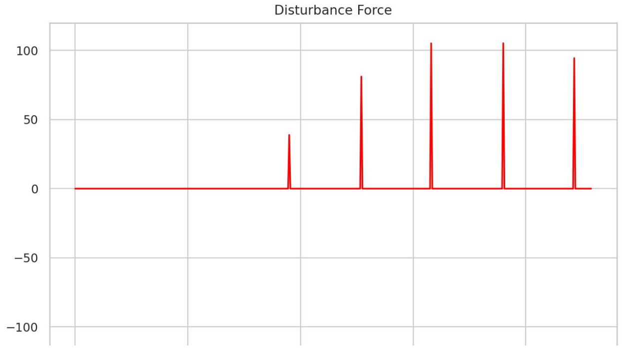


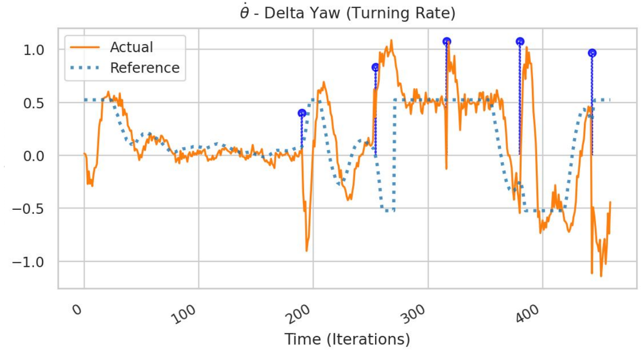
Balance Agent

Results - RL Agent - Trajectory Control - Experiment 2

- Experiment 2 - Straight Line Path under Disturbances - 20 rad/s (5 km/h)

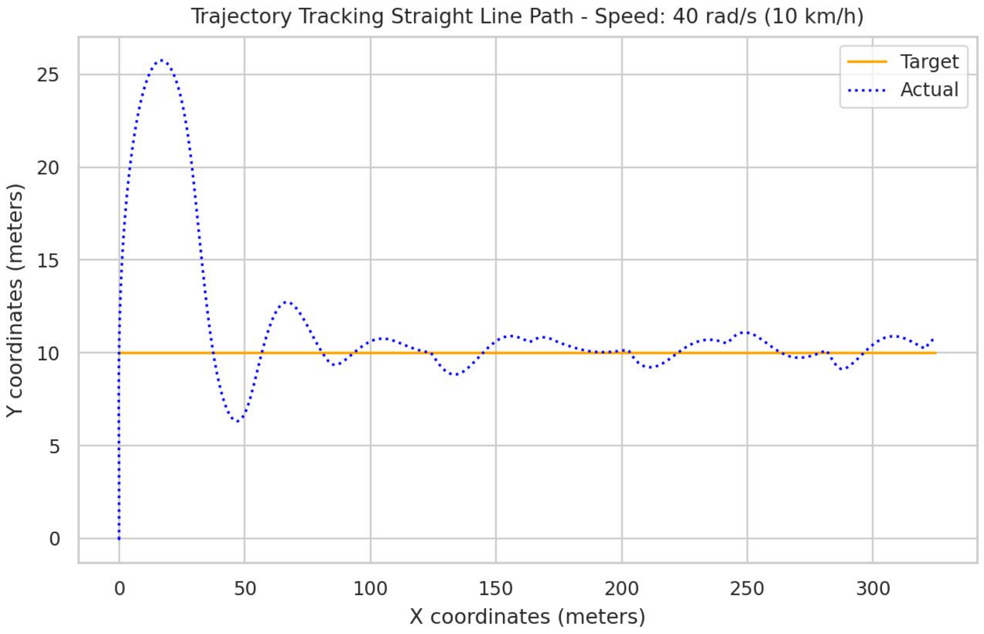


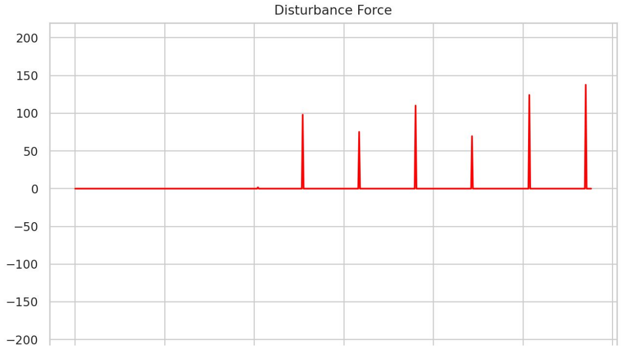


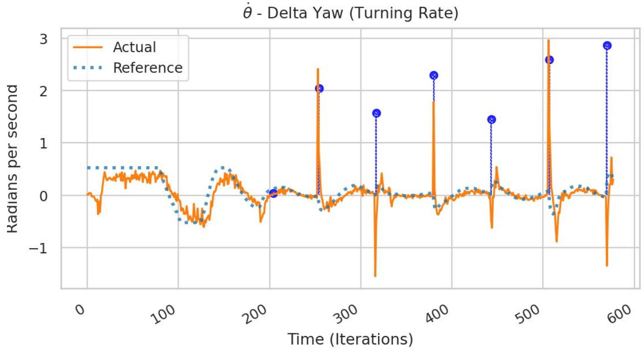


Results - RL Agent - Trajectory Control - Experiment 2

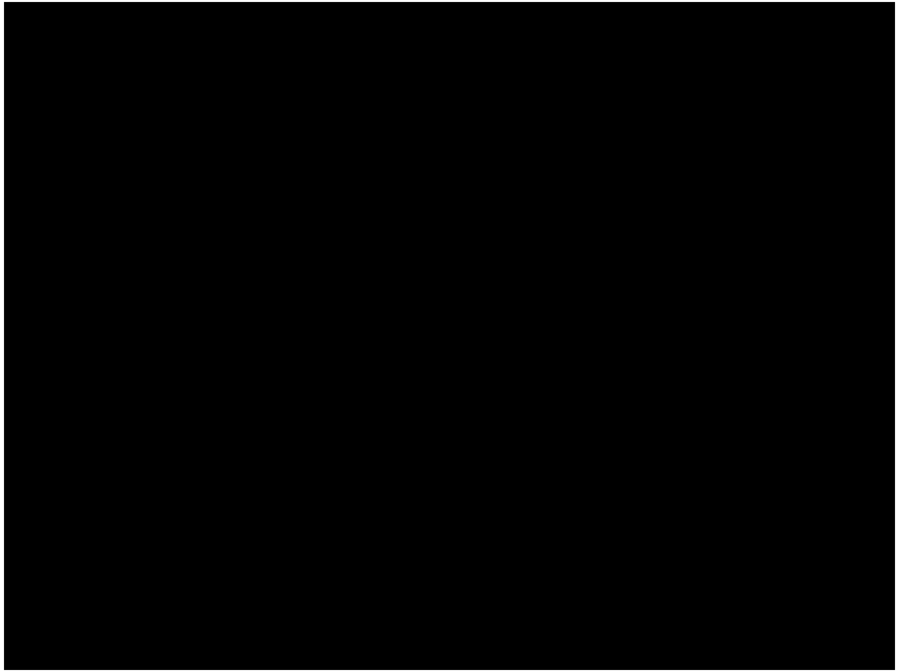
- Experiment 2 - Straight Line Path under Disturbances - 40 rad/s (10 km/h)



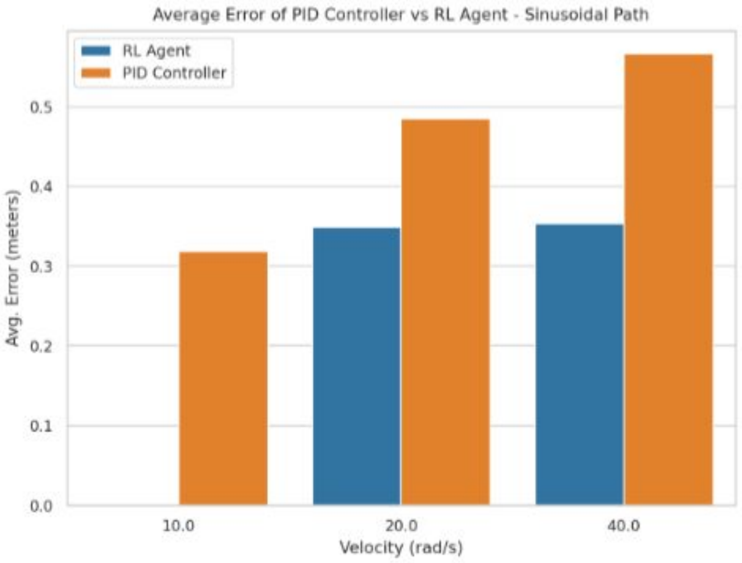




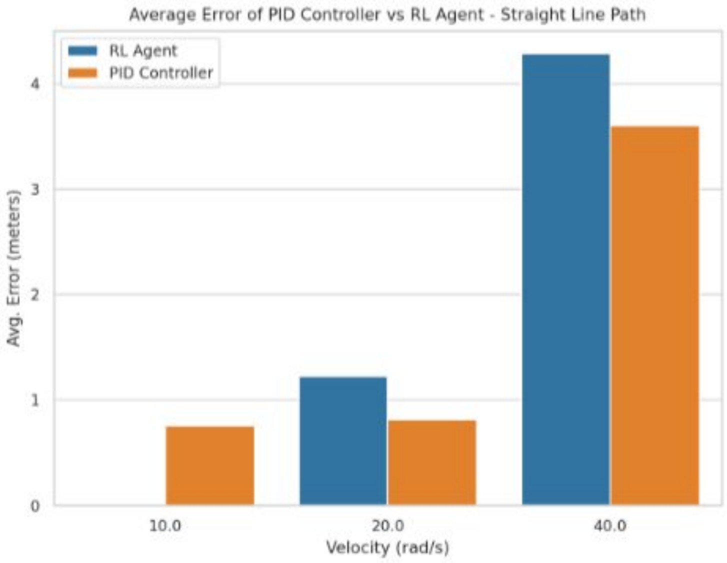
Results - RL Agent - Trajectory Control - Video



Results - PID Controller vs RL Agent - Trajectory Error



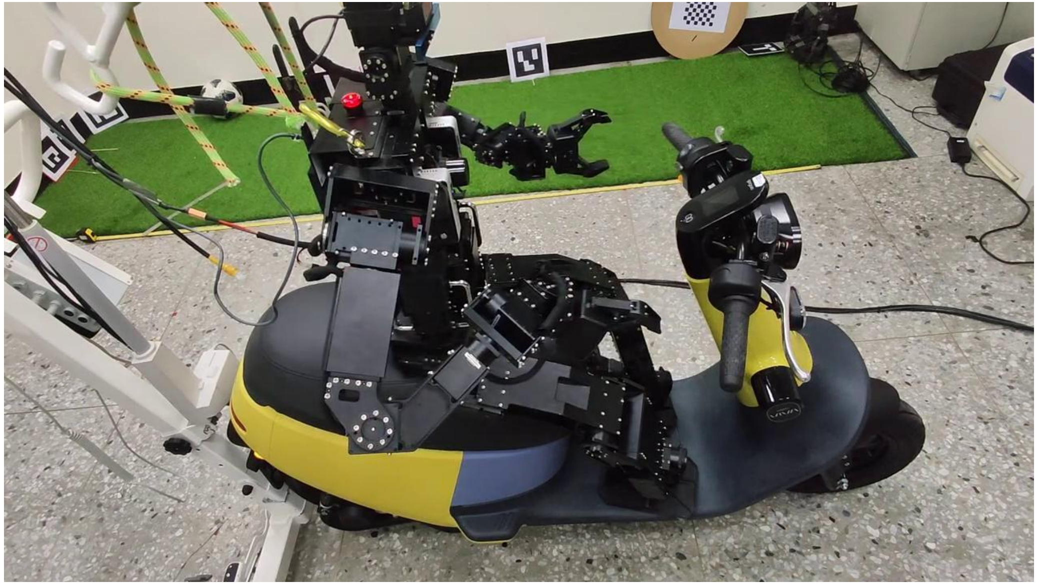
(a) Sinusoidal path.



(b) Straight line path.

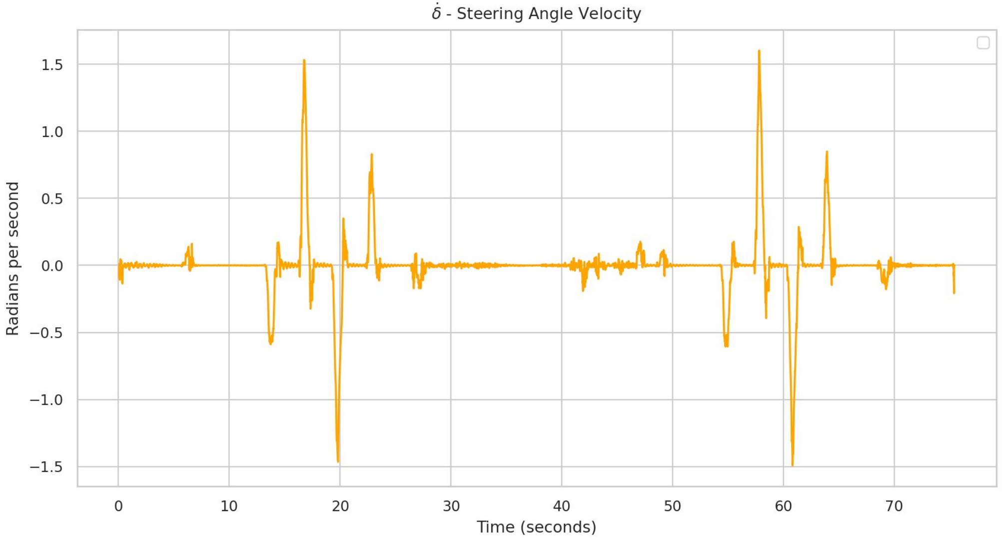
Results - Real Robot Steering Velocity Test

- Using IK an experiment was realized to test the real robot's maximum turning velocity.
- Max velocity measured was approximately 1.5 rad/s (~85 degrees)

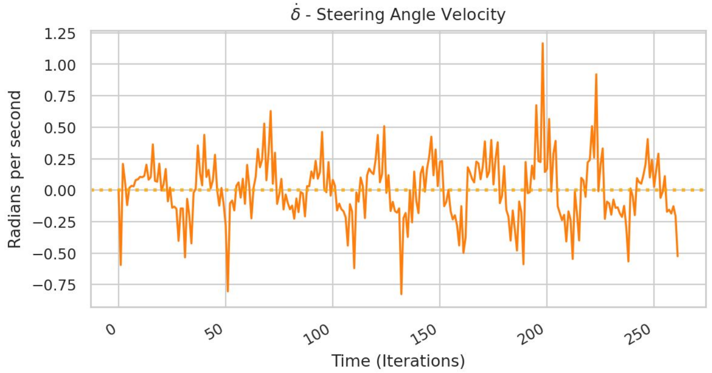


Results - Real Robot Steering Velocity Test

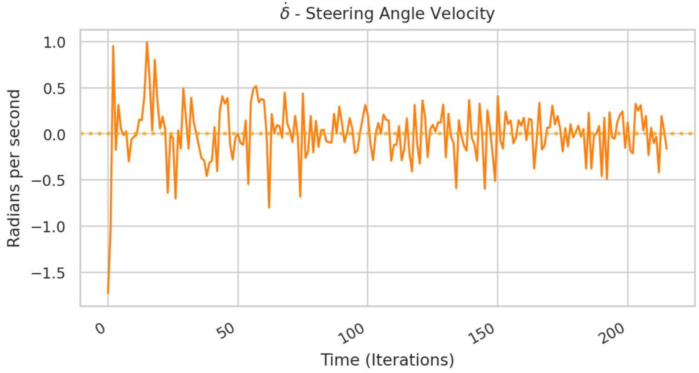
- Max velocity measured was approximately 1.5 rad/s (~85 degrees):



PID Controller (40 rad/s) < 1.25 rad/s



RL Agent (40 rad/s) < 1 rad/s



Conclusion

Conclusion and Future Work

- Two-wheeled vehicles pose an interesting control challenge for humanoid robots.
- PID-based controllers and RL-based controllers were designed to perform balance control and trajectory tracking by manipulating the steering joint.
- No controller was clearly better, and both performed well in different situations.
- Both controllers have some limitations regarding real application.
- Future work include:
    - Run experiments using same simulator, repeat tests and provide statistics.
    - Design controllers to directly control the joints of the robot, instead of steering angle.
    - Investigate controllers that take into account physical limitations of the real robot.
    - Investigate RL methods to directly approach a trajectory point.

References

[1] S. Singhania, I. Kageyama, and V. M. Karanam. Study on low speed stability of a motorcycle. Applied Sciences, 9(11):2278, 2019.

[2] K. J. Astrom, R. E. Klein, and A. Lennartsson. Bicycle dynamics and control: adapted bicycles for education and research. IEEE Control Systems Magazine, 25(4):26–47, 2005.

[3] R. S. Sutton and A. G. Barto.Reinforcement learning:An introduction. MIT press, 2018.

[4] V.Mnih, K.Kavukcuoglu, D.Silver, A.Graves, I.Antonoglou, D.Wierstra, andM.Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

Thank you for your attention