REINFORCEMENT LEARNING FOR THE TRACKING OF UNMANNED AERIAL VEHICLES

Sun Zizhuo[1], Jaden Tjeng[2], Jerry Tamilchelvamani[3]
[1]Hwa Chong Institution (High School), 661 Bukit Timah Rd, Singapore 269734
[2]NUS High School of Mathematics and Science, 20 Clementi Avenue 1, Singapore 129957
[3]Defence Science and Technology Agency, 1 Depot Road Defence Technology Tower A, Singapore 109679

1. Abstract

The increasing prevalence of low-cost drones raises concerns about unauthorized operations and threats to airspace security due to their misuse. The accessibility of low-cost drones poses an asymmetric threat as current methods for tracking and neutralising these drones are costly. One solution to tackle this issue is to employ low-cost drones to track and home in on the drone target, which requires a computer vision model with a controller using visual inputs. Traditionally, Proportional-Integral-Derivative (PID) controllers are used for this purpose, but such a deterministic method may struggle to deal with dynamic environments due to its inability to make anticipatory moves. However, controllers trained with Proximal Policy Optimisation (PPO)-based reinforcement learning show promising potential in tackling these limitations. This study compares the performance of a PID controller and a PPO controller in tracking drone targets through simulations conducted in Unity. Results revealed that while the PID controller maintained high engagement success in simpler scenarios such as straight line and circular paths, engagement success dropped significantly when dealing with more challenging conditions such as sharp turns and continuous curvature. The PPO controller showed excellent reliability and successfully preserved the target within its field of view even in the challenging conditions that the PID controller struggled with.

2. Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have been employed in various fields such as disaster management, agriculture, and defence [1]. However, their increasing prevalence raises concerns about unauthorized operations and threats to airspace security due to their misuse [2]. The accessibility of low-cost drones poses an asymmetric threat as current methods for tracking and neutralising these drones are costly [3]. Hence, finding a cost-effective method to track, monitor and take down such drones is therefore crucial for mitigating these risks. One solution to tackle this issue is to employ low-cost drones to track and home in on the drone target.

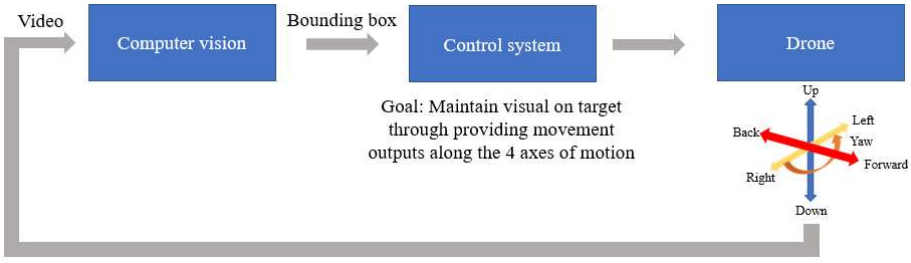


Figure 1: Drone tracking feedback loop

Currently, drone tracking methods involve feeding sensory input, such as from visual or infrared sensors, into a control system that maintains continuous monitoring of the drone. Vision-based methods use high-resolution cameras coupled with object detection algorithms (YOLO, R-CNNs) to detect and track drones in real-time [4]. The control system will then provide movement outputs along the four axes of movement, completing the drone tracking

feedback loop in Fig 1. Traditionally, Proportional-Integral-Derivative (PID) control is used as the control system for drones through correcting small tracking errors. While PID controllers are easy to design and implement, their deterministic nature prevents them from making anticipatory moves to better preserve the target within its field of view. All environmental factors such as wind were also encapsulated into the overall tracking error, and thus PID controllers have a less nuanced understanding of the different causes for the tracking errors and may struggle to maintain optimal performance without continuous adjustments [5].

Reinforcement learning offers a promising solution to these limitations by enabling drones to adapt to changes in the environment through continuous learning. Proximal Policy Optimization (PPO) has become particularly popular due to its balance between performance and computational efficiency [6]. Through feedback-based learning, PPO allows agents to autonomously optimize their behaviour in uncertain and erratic environments which may otherwise overwhelm a PID controller.

Due to the iterative process required to train the PPO controller and the energy intensity of training it in real-world scenarios, a Unity simulation environment was used in the training process before deployment in real-world scenarios.

3. Materials and Methods

3.1 Homogenous Projection

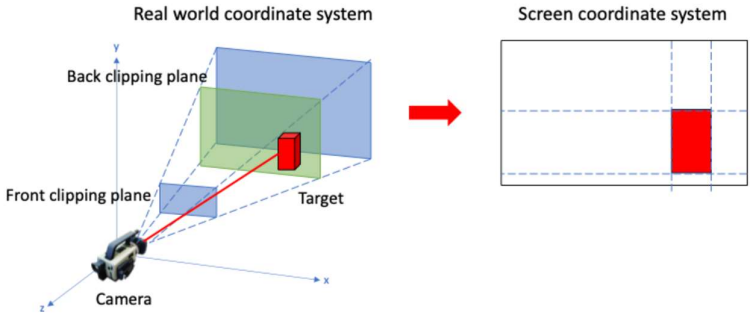


Figure 2: Conversion of 3D coordinates to screen space coordinates

In real-life, drones are given inputs from computer vision models that provide the pixel coordinates as well as the width and height of the target from drone camera feeds. To simulate these real-life inputs as closely as possible, the 2D projection of the target from the perspective of the drone was required as inputs for the model even though the absolute 3D coordinates of the target were known. 2D screen coordinates (x', y') as well as screen-space width and height (w', h') can be obtained from a 3D point, (x, y, z) through homogenous transformations, represented by a series of matrix multiplications:

$$v_{NDC} = \frac{P \cdot V \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}{w} \quad (1)$$

Where P is the camera's projection matrix, V the camera'sx view matrix and w the fourth coordinate after applying $P \cdot V$. This results in a point in normalized device coordinate space, v_{NDC}. It is then mapped to screen space through the following transformations:

$$(x', y') = \begin{cases} x' = \dfrac{NDC_x + 1}{2} \cdot w_s \\ y' = \dfrac{NDC_y + 1}{2} \cdot h_s \end{cases} \tag{2}$$

Where w_s and h_s is the width and height of the screen, respectively. w' and h' can be determined by taking the difference in the coordinates of the bounding box's min and max corners.

3.2 Proximal Policy Optimization

Reinforcement learning was conducted using Unity's ML-Agents with Proximal Policy Optimization (PPO) [6]. The agent was exposed to a limited number of inputs, including information regarding the drone entity it controls (position and direction in which it is facing) and screen-space coordinates of the target and obstacles. Through creating an appropriate reward function, the agent will converge to an optimal policy to track the target.

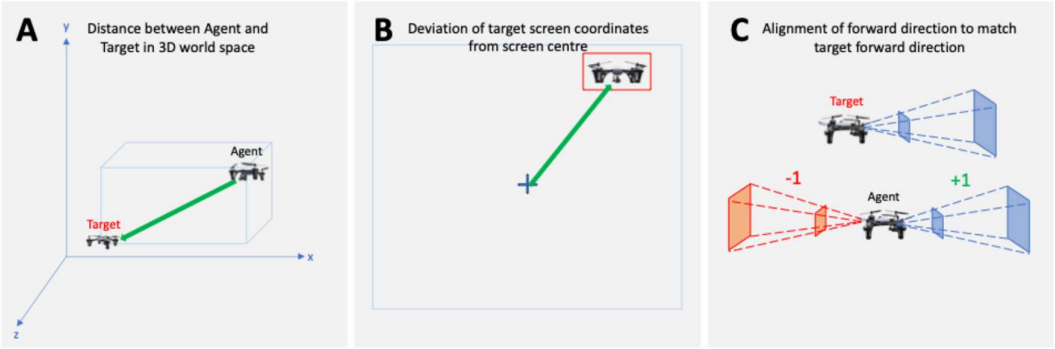


Figure 3: Considerations for reward function

One of the considerations in creating the reward function was the distance between the agent and the target, as shown in Fig 3A. Using the Euclidean distance, the reward was scaled appropriately based on the maximum depth of vision such that a distance closer to the target will be given a bigger reward.

$$R_{distance} = 1 - \frac{\sqrt{(x_a - x_t)^2 + (y_a - y_t)^2 + (z_a - z_t)^2}}{40} \tag{3}$$

Another consideration was the deviation of the target from the screen centre, as shown in Fig 3B. Using the Euclidean distance between the normalized screen coordinates of the centre of the target and the screen centre, the reward was scaled to shape the behaviour of keeping the target to the centre of the agent's field of view.

$$R_{screen} = 1 - \sqrt{(x_{normalized} - 0.5)^2 + (y_{normalized} - 0.5)^2} \tag{4}$$

The agent was also rewarded for aligning its forward direction with the target by taking the dot product of the forward direction unit vectors as shown in Fig 3C. Thus, the reward is scaled to the range of [-1,1] where an exact alignment would give a reward of 1, an exact opposite alignment would give a reward of -1, and a perpendicular alignment would give a reward of 0.

$$R_{alignment} = \hat{F_a} \cdot \hat{F_t} \tag{5}$$

The overall reward for each episode can then be expressed as:

$$R_{total} = R_{distance} + R_{screen} + R_{alignment} \quad (6)$$

An episode was terminated and a penalty of 50 was given whenever the agent was more than 60 metres away from the target or could not view the target for more than 5 seconds. Although 40 metres is the maximum depth of view for the agent, an extra leeway of 20 metres is provided before the episode is ended. This potentially allows for the agent to learn to regain sight of the target within its field of view. However, the agent would still face a smaller penalty as per equation (7) for this extra distance of 20 metres outside of its depth of view.

3.3 Curriculum Learning

Curriculum learning is a methodology that can be applied to the training process in reinforcement learning. Through ordering training examples in increasing difficulty, knowledge accumulated from previous training examples can be leveraged upon when learning a subsequent, more challenging task, thus leading to faster convergence to an optimal policy [7]. The curriculum that was devised gradually incremented the variation of the target's base speed and maximum speed. The agent was also subjected to increasingly difficult target movements, starting from a straight-line path, followed by a circle, then a completely random path.

Lesson	Base speed (m/s)	Maximum target speed variation (m/s)
1	2	0
2	5	1
3	10	2
4	15	5

Table 1: Base speed and maximum target speed variation for each lesson

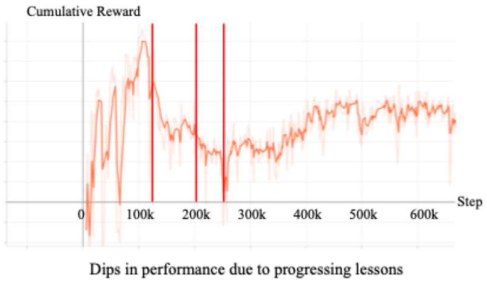


Figure 4: Graph of cumulative reward over episodes

As shown in Fig 4, dips in cumulative reward occurred whenever training parameters were adjusted during curriculum learning. Temporary performance decline, as indicated by dips in cumulative reward, reflected the agent's adaptation to increasing task complexity.

3.4 Proportional-Integral-Derivative (PID) Controller

The PID controller, used as a benchmark for comparison with the PPO controller, is described by the following discretized formula:

$$U[n] = U[n-1] + E[n]\left(K_P + K_I T + \frac{K_D}{T}\right) - E[n-1]\left(K_P + \frac{2K_D}{T}\right) + E[n-2]\frac{K_D}{T} \quad (7)$$

Where $U[n]$ is the control output and $E[n]$ the error at the n-th timestep. T represents the timestep while K_P, K_I and K_D are gain constants for the proportional, integral, and derivative terms of the controller. The following K_P, K_I and K_D values were tuned manually and used for this study.

	ψ	x	y	z
K_P	0.32	0.28	0.28	0.50
K_I	0.025	0.0384	0.0384	0.0384

Table 2: K_P, K_I and K_D values for different axis of motion

3.5 Evaluation Criterion

The performance of the PID controller and the PPO controller were evaluated based on a score *S,* which indicates the percentage of time each controller could keep the target within its field of view over a period *T*. This is expressed as

$$S = \frac{1}{T}\sum_{t=1}^{T} I_t \tag{8}$$

Where I_t is the indicator variable, such that

$$I_t = \begin{cases} 1 \; if \; target \; in \; FOV \\ 0 \; otherwise \end{cases} \tag{9}$$

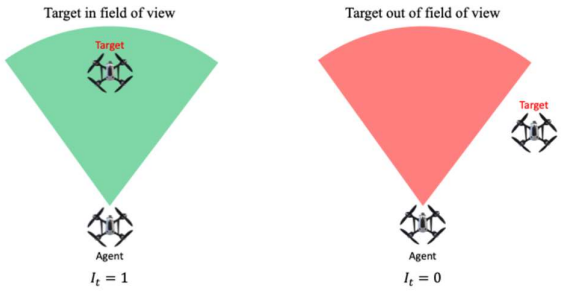


Figure 5: Diagram of evaluation criteria

A higher score indicates a better ability to keep the target in the agent's field of view.

3.6 Experiment Setup

The training process was run on Windows 11 operating system powered with an Intel Core i9-12900 CPU @2500Mhz and 64GB of RAM. The training model was developed with Unity's ML-Agents 0.29.0.

The limitations for flight states used for this study were as follows:

Parameter	Limit
Maximum horizontal speed	20 m/s
Field of View	80°
Maximum view range	40 m

Table 3: Limitations for flight states

The target was moved along different paths to assess the agent's performance under different unpredictable conditions.

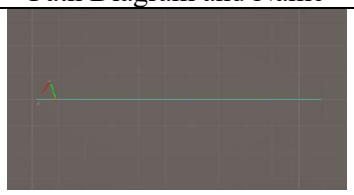
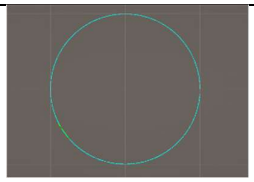
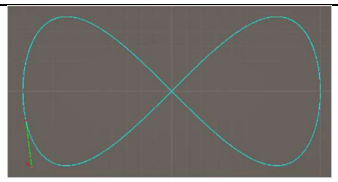
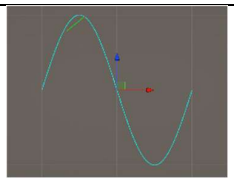
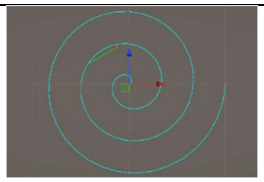
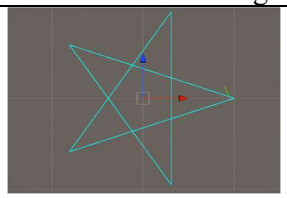
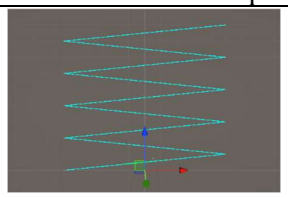
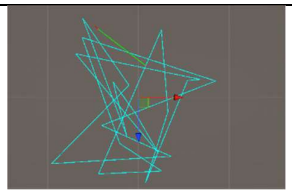
Path Type	Path Diagram and Name
Simple	 Line
Continuous Curvature	 Circle  Figure 8  S-shape  Spiral
Sharp Turns	 Star  Zigzag
Random	 Random

Table 4: Different types of paths used testing (top-down view)

For each path, three different bounding areas (50m x 50m, 100m x 100m, 200m x 200m) were evaluated to observe for any trends in engagement success due to the size of the path.

For each path they were subjected, the agents were initialized at the same random position within a 15-metre range and at least 5 metres away from the target. The target was initially at rest and accelerated uniformly to a speed of 15m/s. A run was considered to have ended when the target completed the path once.

4. Results

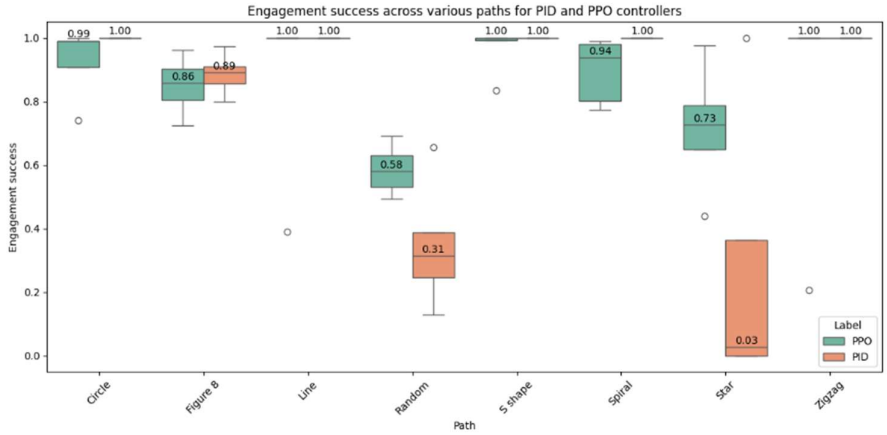


Figure 6: Boxplot of engagement success for PID and PPO controllers across various paths bounded by a 50m x 50m area

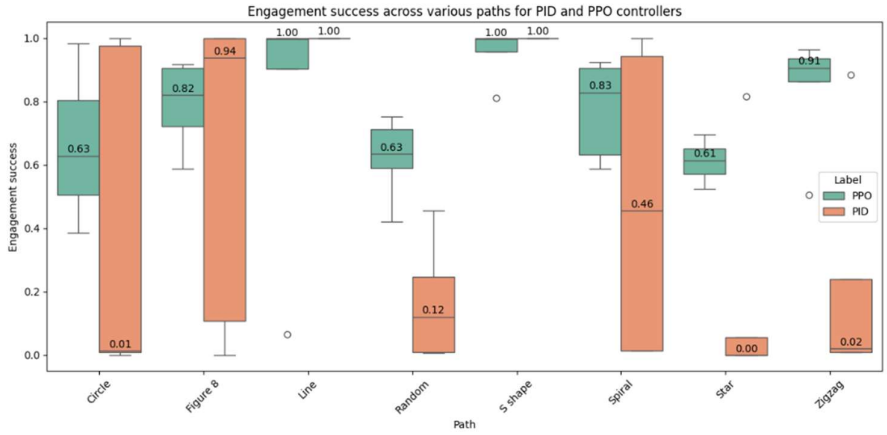


Figure 7: Boxplot of engagement success for PID and PPO controllers across various paths bounded by a 100m x 100m area

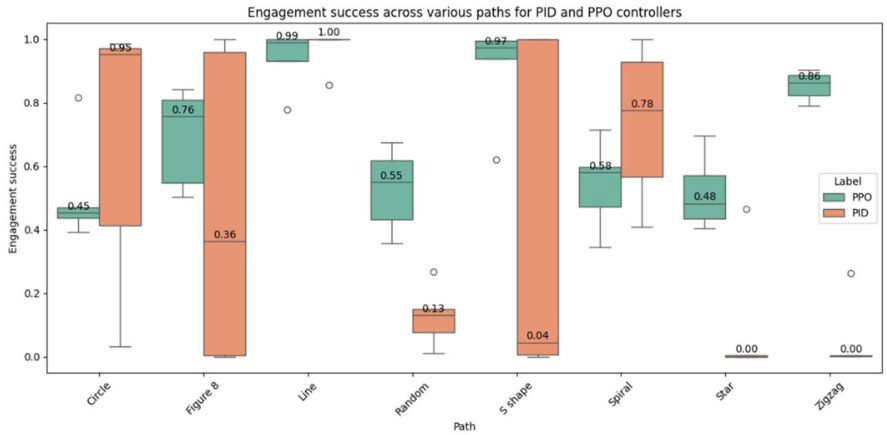


Figure 8: Boxplot of engagement success for PID and PPO controllers across various paths bounded by a 200m x 200m area

The results from the performance evaluation revealed that the PPO controller outperformed the PID controller in most cases. From Fig 6, the PPO controller significantly outperformed the PID controller in the Star and Random paths. From Fig 7 and Fig 8, a similar trend can be observed regardless of the bounding area, where the PPO controller significantly outperformed the PID controller in the Star, Random and even the Zigzag paths.

However, there were some cases where the PID controller exhibited better engagement success than the PPO controller. From Fig 6 and Fig 7, the PID controller had slightly higher engagement success for the Figure 8 path and from Fig 8, the PID controller had significantly higher engagement success for the Circle and Spiral paths.

Another observation from the results was that the PID controller had less reliable performance for larger bounding areas. This was evident from the significantly larger interquartile ranges for engagement success of the PID controller across most paths in Fig 7 and Fig 8 than that in Fig 6. On the other hand, the PPO controller had generally small interquartile ranges across all paths and all bounding areas, indicating that there was less variability in engagement success.

5. Findings

The observed trends could be attributed to the fundamental differences between the two controllers. The PID controller's reliance on predefined error minimization rules made it effective for simple paths but less capable of adapting to dynamic movements. This limitation became more evident in paths with sharp turns such as the Star, Zigzag and Random paths, thus explaining the consistent poorer engagement success for the PID controller in these paths. On the other hand, the PPO controller, trained using reinforcement learning, exhibited higher adaptability due to its ability to generalize across varying conditions and paths. Therefore, it could anticipate and react to the sharp changes in direction of the target, which the deterministic PID controller was unable to do.

Moreover, the greater reliability of the PPO controller observed from the results could be due to different levels of sensitivity to the initial conditions of the agents. During the experiment runs, it was observed that the PID controller was more sensitive to its starting position. If it was initialised closer to the target, it was more likely to lose track of the target as it was unable to quickly adjust to the target’s movements.

6. Conclusion

This study compared drone target tracking abilities of a PID controller and a PPO controller through simulations conducted in Unity. Results revealed that the PPO controller was generally better than the PID controller in maintaining the target within its field of view and tracking it, especially in unpredictable conditions and sharp turns. Furthermore, the PPO controller exhibited greater reliability, achieving less variability in engagement success over multiple simulation runs.

Future research could focus on exploring alternative training methodologies to allow the PPO controller to adapt to vastly different environments than its training environment including the addition of environmental factors such as wind and obstacles. One area that could be explored is adversarial reinforcement learning, where a destabilizing adversary is introduced, allowing the agent to be exposed to more active adversarial strategies that reflect real-life scenarios more accurately. Thus, this can allow the PPO controller to be more robust to differences in training and test conditions [8].

7. Code Availability

All data and code used for running simulations is available on a GitHub repository at https://github.com/zyrvad/rl-drone-tracker.

8. Acknowledgements

We would like to thank Jerry Tamilchelvamani (Head Engineering C3 Development Programme Centre, DSTA) for his guidance and support throughout the project work cycle. We would also like to thank Daryl for his help with the PID controller as well as the math in other aspects of the project.

9. References

[1] M. Ayamga, S. Akaba, and A. A. Nyaaba, "Multifaceted applicability of drones: A review," *Technol. Forecast. Soc. Change*, vol. 167, p. 120677, Feb. 2021. [Online]. Available: https://doi.org/10.1016/j.techfore.2021.120677

[2] J. Pyrgies, "The UAVs threat to airport security: risk analysis and mitigation," *J. Airline Airport Manage.*, vol. 9, no. 2, pp. 63–96, May 2019. [Online]. Available: https://doi.org/10.3926/jairm.127

[3] C. T. D. L. Van Bossuyt and B. Hale, "Reducing asymmetry in countering unmanned aerial systems," May 02, 2022. [Online]. Available: https://dair.nps.edu/handle/123456789/4561

[4] F. Svanström, F. Alonso-Fernandez, and C. Englund, "Drone Detection and Tracking in Real-Time by Fusion of Different Sensing Modalities," *Drones*, vol. 6, no. 11, Oct. 2022. [Online]. Available: https://doi.org/10.3390/drones6110317

[5] I. Lopez-Sanchez and J. Moreno-Valenzuela, "PID control of quadrotor UAVs: A survey," *Annu. Rev. Control*, vol. 56, p. 100900, Jul. 2023. [Online]. Available: https://doi.org/10.1016/j.arcontrol.2023.100900

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347* [cs.LG], Aug. 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[6] A. Juliani *et al*., "Unity: A General Platform for Intelligent Agents," *arXiv:1809.02627* [cs.LG], May 2020. [Online]. Available: https://arxiv.org/abs/1809.02627

[7] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey," *arXiv:2003.04960*, Mar. 2020. [Online]. Available: https://arxiv.org/abs/2003.04960

[8] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust Adversarial Reinforcement Learning," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 2817–2826. [Online]. Available: https://proceedings.mlr.press/v70/pinto17a.html

10. Appendix

Appendix A: Table of Results

	Minimum		1st Quartile		Median		3rd Quartile		Maximum	
	PPO	PID	PPO	PID	PPO	PID	PPO	PID	PPO	PID
Line	0.391	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Circle	0.741	1.00	0.907	1.00	0.991	1.00	0.991	1.00	1.00	1.00
Figure 8	0.724	0.798	0.803	0.855	0.858	0.891	0.901	0.909	0.961	0.973
S-shape	0.835	1.00	0.991	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Spiral	0.772	1.00	0.802	1.00	0.937	1.00	0.980	1.00	0.99	1.00
Star	0.440	0	0.649	0	0.726	0.027	0.787	0.364	0.974	1.00
Zigzag	0.206	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Random	0.492	0.130	0.532	0.246	0.581	0.314	0.629	0.387	0.692	0.656

Table 5: Five number summary of engagement success for PPO and PID controllers across various paths bounded by a 50m x 50m area

	Minimum		1st Quartile		Median		3rd Quartile		Maximum	
	PPO	PID	PPO	PID	PPO	PID	PPO	PID	PPO	PID
Line	0.066	1.00	0.904	1.00	0.997	1.00	1.00	1.00	1.00	1.00
Circle	0.385	0	0.504	0.009	0.628	0	0.803	0.974	0.982	1.00
Figure 8	0.587	0	0.723	0.107	0.819	0.9	0.906	1.00	0.916	1.00
S-shape	0.812	1.00	0.957	1.00	0.996	1.00	1.00	1.00	1.00	1.00
Spiral	0.589	0.013	0.633	0.013	0.828	0.5	0.904	0.942	0.925	1.00
Star	0.523	0	0.571	0	0.614	0	0.65	0.056	0.695	0.815
Zigzag	0.506	0.008	0.862	0.009	0.906	0	0.936	0.239	0.963	0.884
Random	0.421	0.007	0.590	0.010	0.635	0.100	0.712	0.246	0.752	0.456

Table 6: Five number summary of engagement success for PPO and PID controllers across various paths bounded by a 100m x 100m area

	Minimum		1st Quartile		Median		3rd Quartile		Maximum	
	PPO	PID	PPO	PID	PPO	PID	PPO	PID	PPO	PID
Line	0.778	0.856	0.930	1.00	0.989	1.00	1.00	1.00	1.00	1.00
Circle	0.391	0.032	0.437	0.413	0.453	0.952	0.471	0.971	0.815	0.986
Figure 8	0.503	0.00	0.546	0.005	0.756	0.363	0.809	0.959	0.841	1.00
S-shape	0.621	0.00	0.938	0.006	0.973	0.045	0.994	1.00	0.994	1.00
Spiral	0.346	0.408	0.472	0.565	0.580	0.775	0.598	0.929	0.715	1.00
Star	0.405	0.00	0.435	0.00	0.483	0.005	0.0571	0.005	0.697	0.465
Zigzag	0.790	0.003	0.822	0.003	0.862	0.003	0.887	0.004	0.903	0.264
Random	0.358	0.011	0.432	0.077	0.550	0.131	0.617	0.149	0.676	0.268

Table 7: Five number summary of engagement success for PPO and PID controllers across various paths bounded by a 200m x 200m area