ELEN90095 — AI For Robotics — Final Project Report

Tianchen Lou – 1078368 – tianchenl@student.unimelb.edu.au
Arya Araban – 1439683 – aaraban@student.unimelb.edu.au

A – Road Simulation:

The training road in Figure 1 presents a challenging course designed to assess driver skills in various scenarios. Sharp turns, measuring 60 degrees, test the ability to handle sudden changes in direction, simulating real-life situations requiring quick reactions and precise steering control. Multiple 40 degree turns evaluate maneuvering through moderate curves commonly encountered on regular roads.

In addition, a wider 145-degree turn is included to assess the driver's ability to maintain speed while executing a smooth and controlled turn. This tests their proficiency in navigating longer, continuous curves. By incorporating these different types of turns, the training road provides a comprehensive assessment of both the car's handling capabilities and the driver's skills in responding to various turn scenarios.

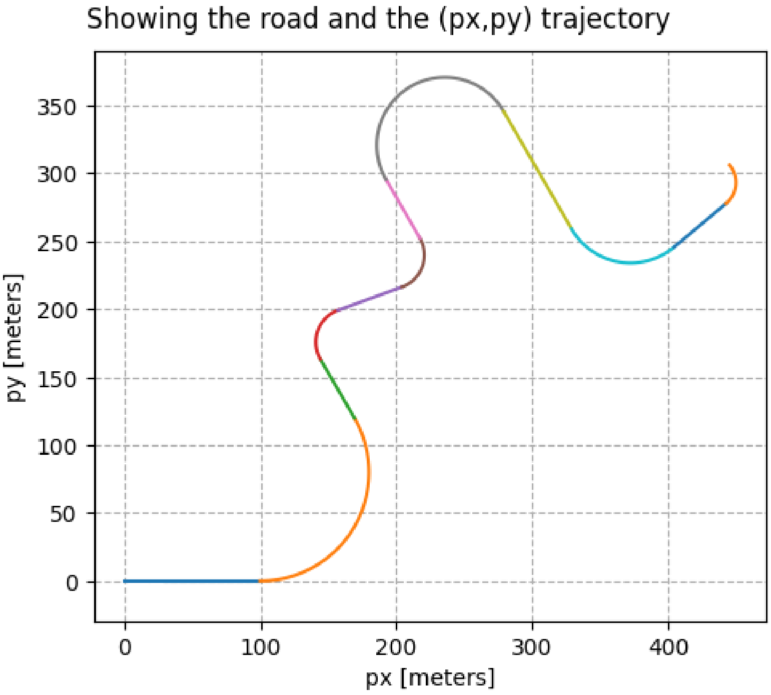

Figure 1

The evaluation road as seen in Figure 2 is designed to replicate common road conditions by including a sharp turn and a wider turn. These elements assess the policy’s ability to handle sudden changes in direction and execute longer, continuous curves. By simulating real-world scenarios, the evaluation road provides an accurate evaluation of the car's agility, the driver's skill, and the vehicle's stability and traction during such maneuvers.

Compared to the training road, which may have exaggerated elements, the design of the evaluation road is designed to mirror the conditions encountered in typical daily commutes. This alignment with real-world scenarios enhances the evaluation process by providing a practical and representative assessment of the car's handling capabilities and the policy’s proficiency in responding to typical turn scenarios commonly found on regular roads.

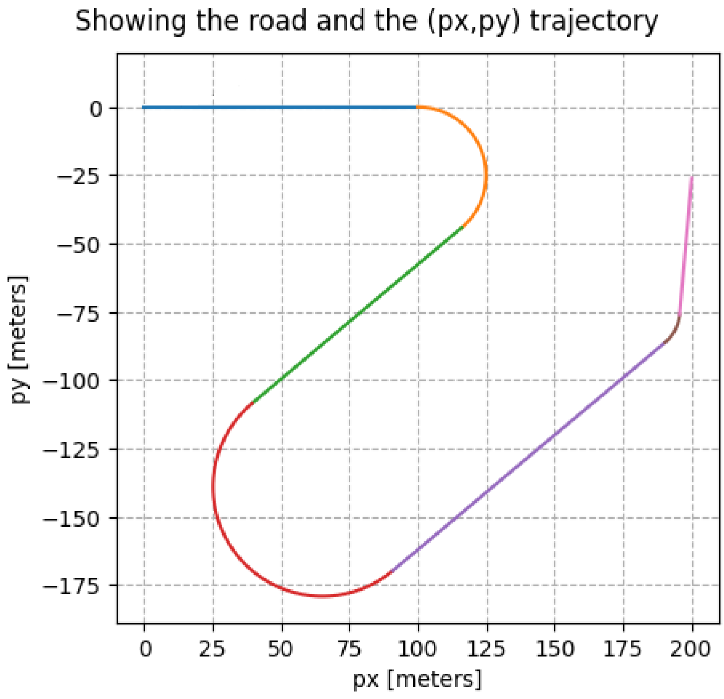

Figure 2

B – Performance Metrics:

3 sets of performance metrics were implemented to assess the quality of the policy in adhering to legal driving requirements, the comfort for the passenger, and the timeliness of the algorithm. For legality, a speed limit of 80km/h was placed on the car, as well as a penalty of straying more than 0.83 meters from the center of the road. This number was calculated given the average width of highway roads in Australia being 3.6m, and the average width of a car being 1.94m in Equation 1 (Craft, 2023) (Bicycle NSW, 2020).

$$Equation\ 1: Maximum\ deviation = (3.6 \div 2) - (1.94 \div 2) = 0.83m$$

For passenger comfort, the acceleration and angular acceleration of the car was considered. A maximum acceleration of 1.5m/s^2 and angular acceleration of 0.1 rad/s^2 were chosen with respect to the research article by de Winkle et al (2023), as motion comfort is an extremely important metric for autonomous vehicles. Finally, the total number of timesteps the car takes to finish the course is inversed and recorded as its reward for timeliness.

C – Policy Design and Synthesis:

1 - PPO:

1.1 - Formulation:

The Proximal Policy Optimization (PPO) algorithm was utilized as the initial approach. PPO is a reinforcement learning algorithm based on policy gradients, which aims to iteratively optimize a stochastic policy in order to maximize the cumulative reward. It builds upon standard policy gradient methods by incorporating techniques that facilitate stable learning and prevent the loss of previously learned good policies, known as catastrophic forgetting (Yu et al, 2022).

The key ideas behind PPO are:

- Using a surrogate objective function that clips the policy update to avoid excessive changes to the policy at each iteration. This improves training stability and avoids catastrophic forgetting of good policies.
- Utilizing an advantage estimator that reduces the variance of policy gradient estimates. This leads to more consistent policy updates.
- Updating the policy multiple times at each iteration using mini-batch stochastic gradient ascent. This further improves sample efficiency.

In our implementation, we used the PPO algorithm from the Stable Baselines library. This implements the clipped surrogate objective function and advantage estimator from the original PPO paper. The policy network is represented as a multilayer perceptron with parameters θ. At each PPO update, the policy loss is computed using the clipped surrogate objective in Equation 2:

$$Equation\ 2:$$

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

where $\hat{E}_t$ is the empirical average over a batch of transitions, $r_t(\theta)$ is the probability ratio between new and old policies, A_t is the estimated advantage, and ε is a hyperparameter that clips the objective. This loss function bounds the policy update at each step, improving stability. The policy parameters θ are then updated via stochastic gradient ascent on L^{CLIP}(θ).

1.2 - Observations:

The policy receives a set of observations related to the state of the environment and the vehicle. These observations are used to calculate the reward signals that provide training feedback to the policy. Several reward functions were designed to encourage behaviors that lead to safe, comfortable, and efficient driving:

Distance to Center Reward: This reward penalizes the distance of the vehicle from the center of the driving lane, as shown in Figure 3. It provides a linearly decreasing reward as the distance increases, with a maximum penalty when the distance exceeds 2 meters. Keeping the vehicle centered in its lane is important for safety and meeting the performance objectives.

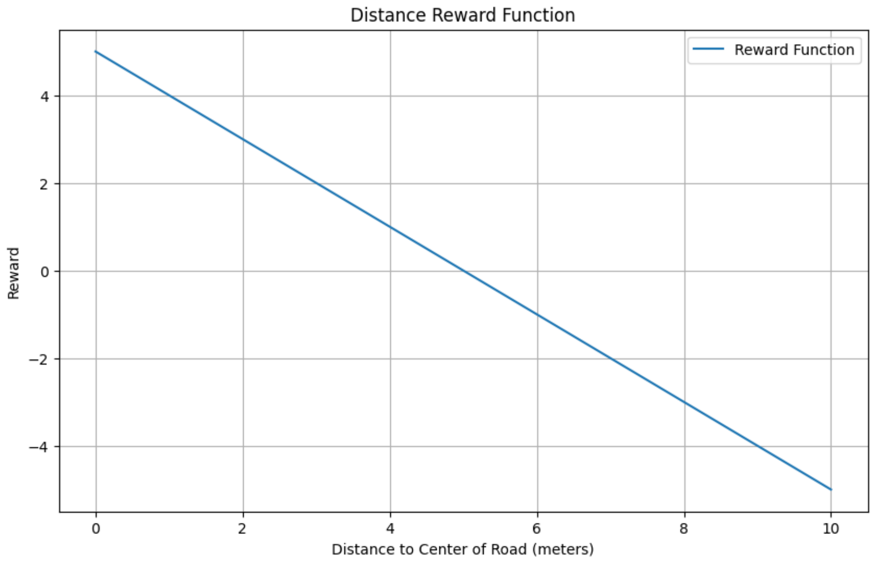

Figure 3

Speed Reward: The speed reward, shown in Figure 4, encourages the vehicle to maintain an appropriate speed within the speed limit. It provides an increasing reward from 0-25 km/h, a constant reward from 25-80 km/h, and a decreasing reward above 80 km/h. This incentive structure aims to achieve posted speed limits when conditions allow while discouraging excessive speeding.

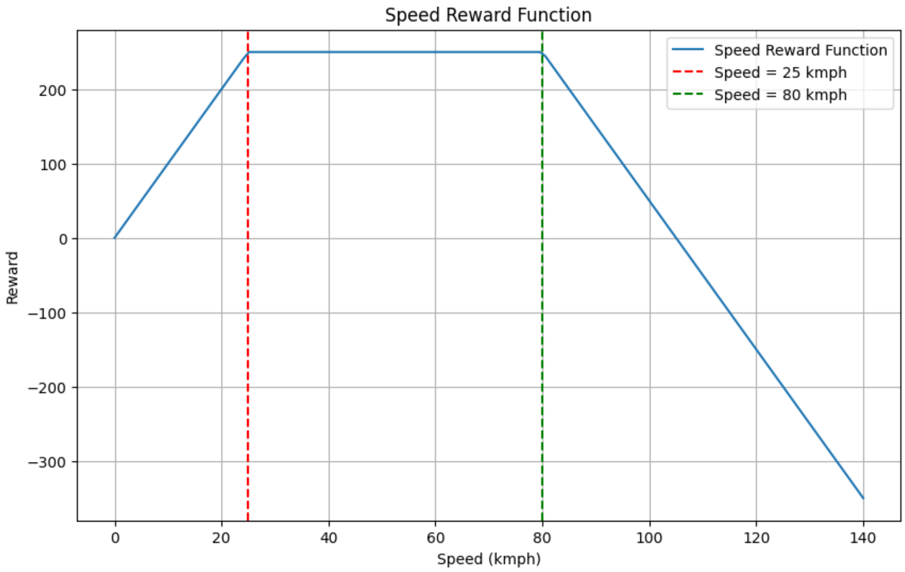

Figure 4

Smooth Steering Reward: This reward penalizes large changes in steering angle between successive timesteps, as shown in Figure 5. By discouraging unnecessary jerky motions, it helps achieve smooth and comfortable steering behavior.

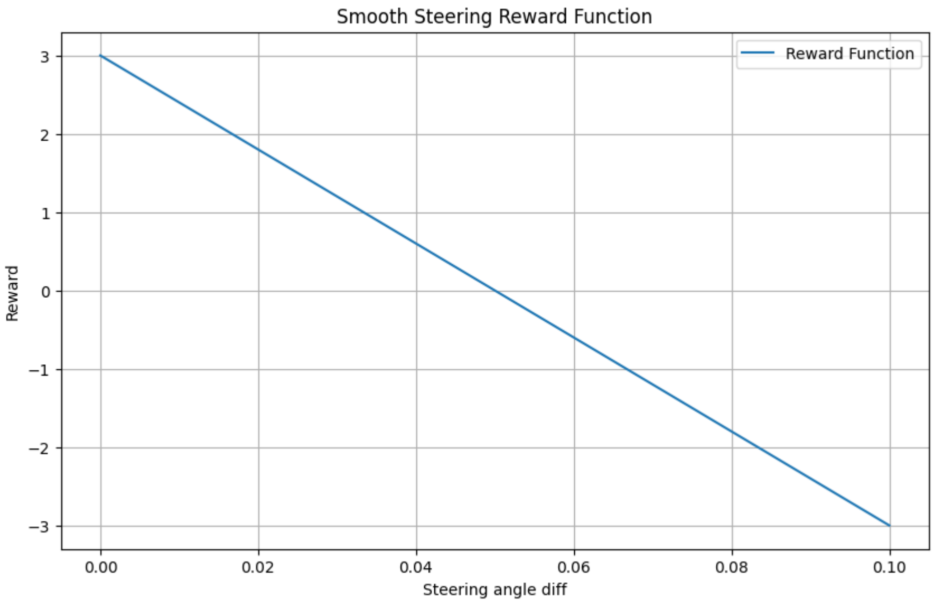

Figure 5

Road Angle Difference Reward: The road angle difference reward considers the difference between the vehicle's heading angle and the road angle at the closest point. As shown in Figure 6, it provides a decreasing linear reward as this difference increases. Matching the road angle is important for stable trajectory tracking.

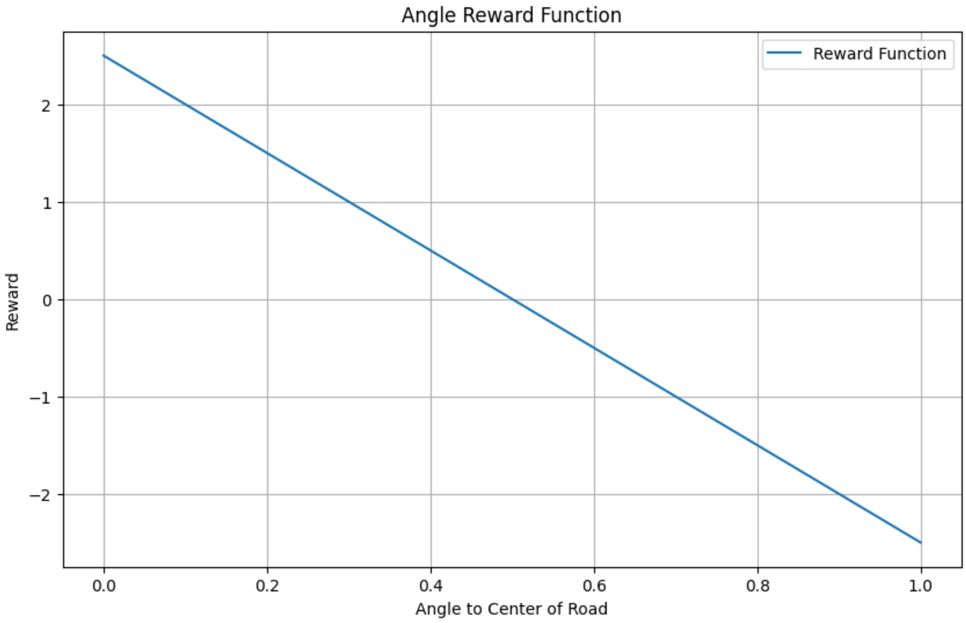

Figure 6

Progress Reward: A positive reward is provided at each timestep that the vehicle makes progress along the road. This encourages completing the course in an efficient and timely manner.

By combining these distinct reward signals, the policy is shaped towards driving behaviors that meet key objectives related to safety, comfort, efficiency, and legal compliance. The relative weightings of each reward component can be tuned as needed to adjust the priorities.

1.3 – Hyperparameter Tuning:

A grid search was performed to tune the hyperparameters of the PPO algorithm. As can be seen in table 1, The hyperparameters under consideration included the learning rate, batch size, clip range, and reward coefficients (which determine the weight of each reward signal in the total return).

Hyperparameter	Value range
Learning rate	[0.0003, 0.003, 0.03]
Batch size	[32,64,128]
Clip Range	[0.1, 0.25, 0.4]
Reward Coefficients	[[1.1, 1, 1, 1.15, 1.25], [1.2, 0.9, 1, 1.1, 1.4]]

Table 1

The grid search yielded 54 possible combinations for the hyperparameters. To manage the computational load of testing these combinations, each was trained for just one epoch with 5000 timesteps. For evaluation, a composite score was utilized, which is the sum of all performance metrics introduced in Section B. The performance results across various hyperparameters are presented in Figure 7. Meanwhile, Table 2 displays the top 5 performing combinations of the hyperparameters.

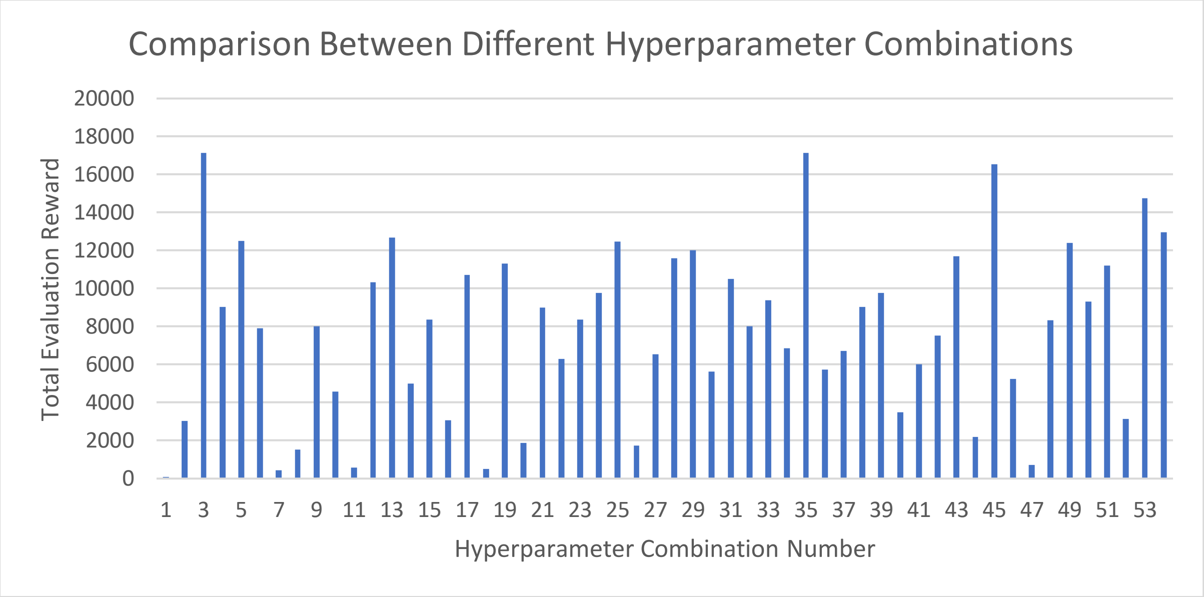

Figure 7

Combination number	Learning rate	Batch size	Clip Range	Reward Coefficients	Total evaluation performance
35	0.003	64	0.25	[1.1, 1, 1, 1.15, 1.25]	17154
3	0.03	32	0.25	[1.2, 0.9, 1, 1.1, 1.4]	17123
45	0.03	64	0.4	[1.2, 0.9, 1, 1.1, 1.4]	16539
53	0.0003	128	0.1	[1.1, 1, 1, 1.15, 1.25]	14747
54	0.003	32	0.1	[1.1, 1, 1, 1.15, 1.25]	12953

Table 2

Variations observed in the previous plot could potentially be attributed to the car deviating from the road at different timesteps, leading to termination. Regardless, the final model was trained using the combination of hyperparameters that yielded the best performance.

1.4 – Final Result:

The final policy was trained for 25,000 timesteps over 5 epochs. The training progress showed steady improvement in the cumulative reward over time.

The policy was then evaluated on the training and validation driving environments. As shown in Figure 8, the agent was able to closely follow the centerline of the training course, successfully navigating the various sharp and moderate turns.

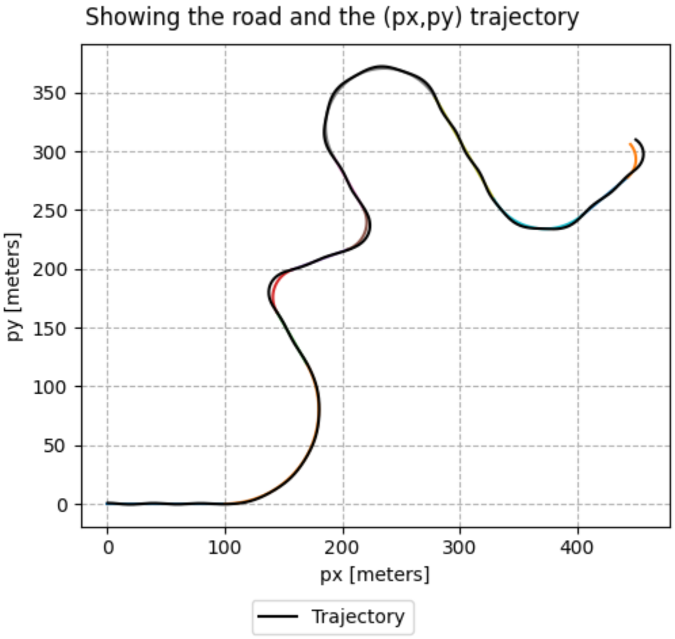

Figure 8

On the validation road, the policy also exhibited generally high trajectory tracking ability, as seen in Figure 9. It was able to handle the sharp and wider turns present on this course.

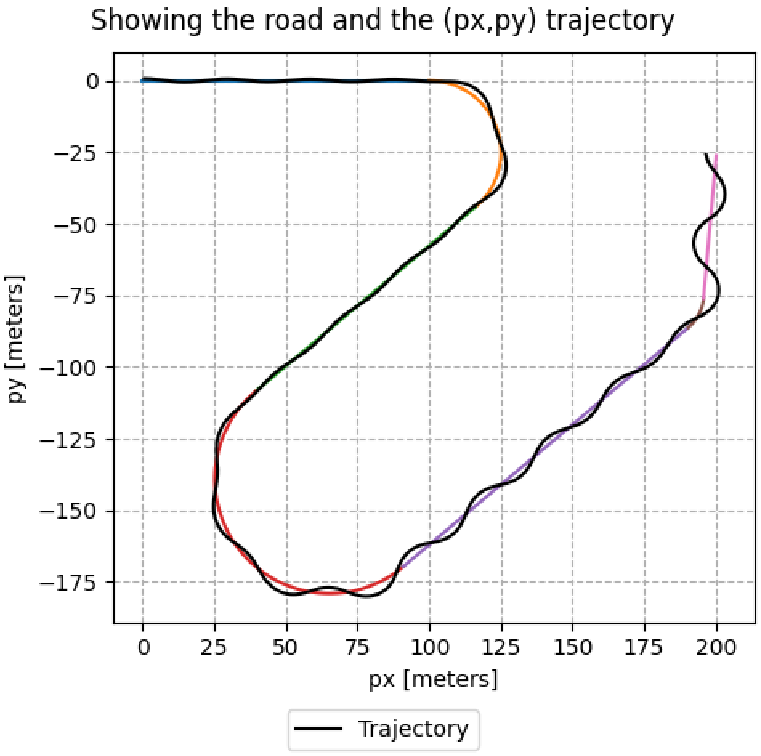

Figure 9

However, one limitation that was observed is that the towards the end of the road, the agent struggled to stay centered in the lane, visibly "swerving" from side to side even on straight road sections. This behavior suggests the policy has not fully learned to stably track the exact centerline. Some potential reasons for this include:

- The relative weightings of the reward components may need further tuning to place greater priority on lane centering.
- Exploration noise during training allowed succeeding while deviating slightly from the center, reducing incentive to follow it precisely.
- The policy network and training method have limited capacity to completely master the complex dynamics involved in perfect centerline tracing.

Overall, the PPO policy was largely successful at navigating the driving courses but displayed some difficulty with finely precise lane centering. Further tuning of the training process and reward functions could potentially improve this limitation.

2 - TRPO:

2.1 – Formulation:

A Trust Region Policy Optimization (TRPO) algorithm was selected as the team’s second reinforcement learning approach to the project. It is designed to enhance the stability of policy optimization methods in complex environments to a higher degree than the clipping methods in PPO using a trust region (Lapan, 2018). This constraint ensures that future policy updates will not significantly change from the current policy, providing stability during training. TRPO's objective function is formulated to maximize expected advantages under the new policy, subject to a constraint on the Kullback-Leibler divergence between the old and new policies.

Relative to the PPO method, TRPO is more computationally complex as it solves second order equations rather than first order. While studies show that the trust region can improve stability of training, it is still a significant tradeoff compared to the PPO's speed and efficiency (Engstrom et al, 2020). The method follows the loss function in Equation 3 below.

$$Equation\ 3$$

$$L_{\theta_{\text{old}}}^{IS}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t \right]$$

The definitions for the symbols are the same as in PPO, where $\hat{E}_t$ is the empirical average and A_t is the advantage. Instead of a clipping function however, the function divides the new and old policy results together to ensure the change between policies will not be too great to ensure training stability.

2.2 – Observations:

A number of observations were made available to the TRPO algorithm including constraints on acceleration. Relative to the observations in PPO, the functions implemented here use quadratics to create steeper drop offs in rewards to aim for a stricter performance. The sharper gradients were implemented as the training region aspect of TRPO was expected to still be able to stabilize training even with the higher punishments for going off course.

Across Figures 10 and 11, quadratic and cubic drop offs were used for the distance from the center of the road and the angle reward function. The speed function in Figure 12 still follows a flat and linear structure, but has shifted to prioritize higher speed within legal limits to finish the course faster. As a result of this shift, a new acceleration metric was also implemented in Figure 13 to ensure the car does not accelerate too fast in straight road sections to achieve these higher speed targets.

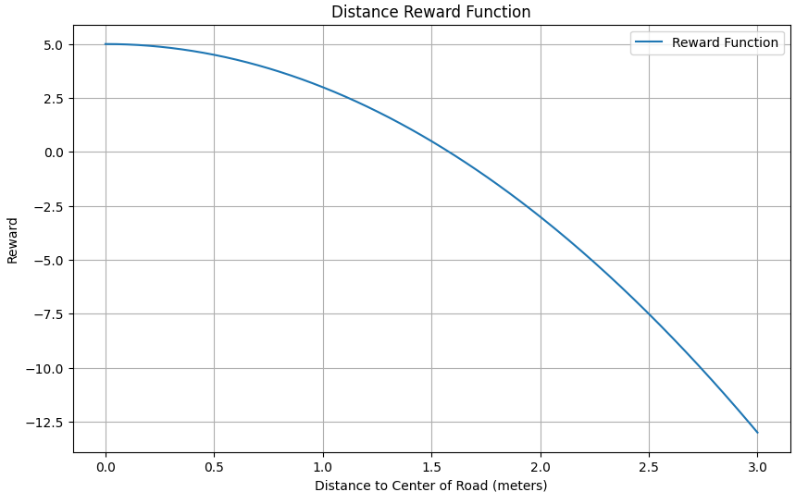

Figure 10

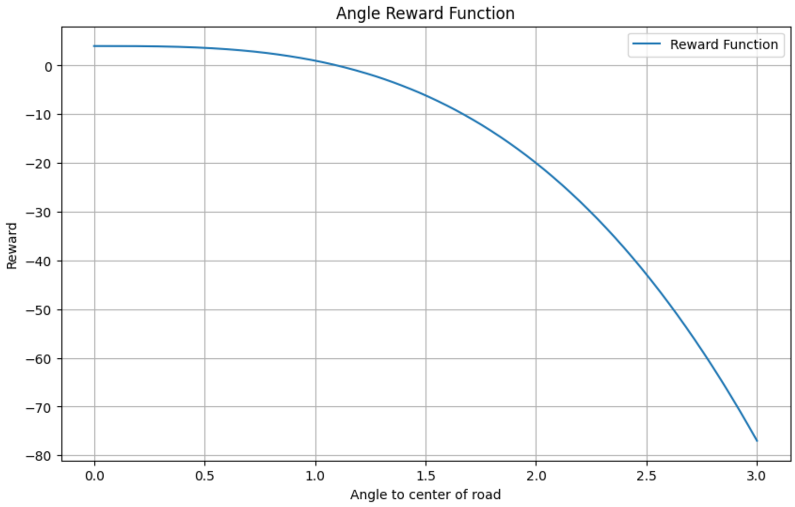

Figure 11

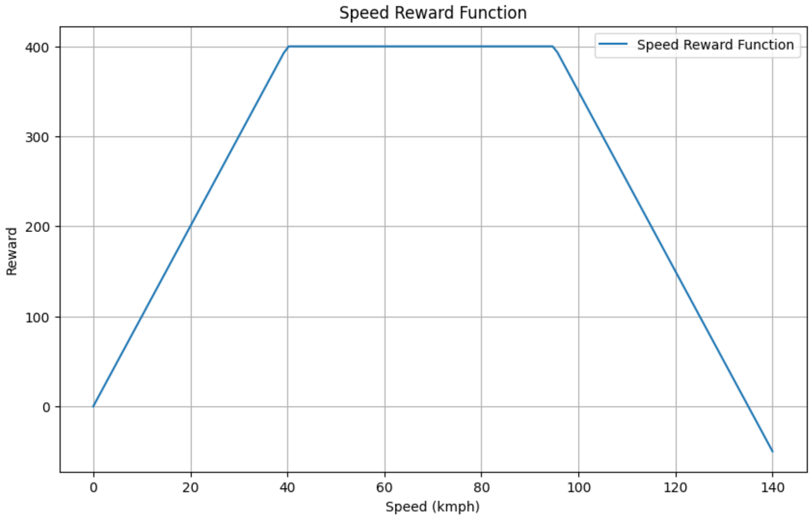

Figure 12

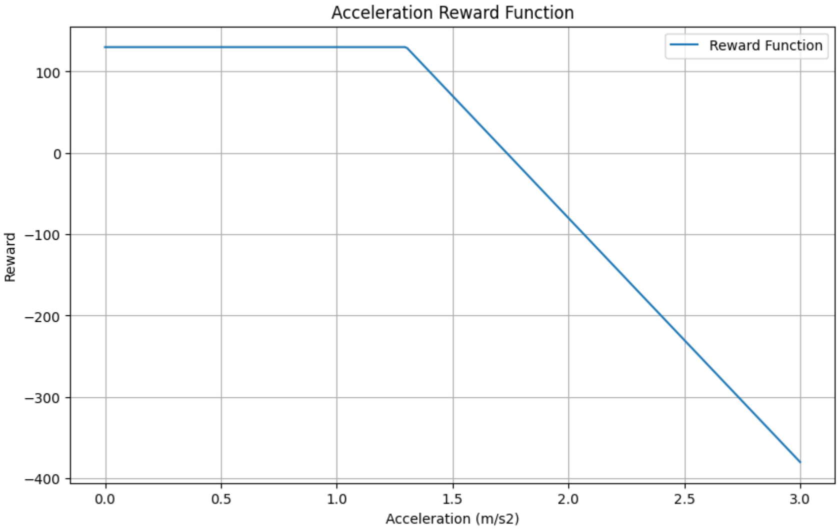

Figure 13

2.3 – Hyperparameter tuning:

For the TRPO policy, the learning rate, batch size, and discount factor (gamma) were selected to be tuned to achieve best performance. Unlike the PPO method above, TRPO does not have a clipping range variable, so instead, gamma was used as the third tunable parameter. Table 3 indicates the range and types of hyperparameters selected.

Hyperparameter	Value range
Learning rate	[0.0005, 0.005, 0.05]
Batch size	[32, 64, 128]
Gamma	[0.6, 0.8, 0.99]

Table 3

Using the grid search method 27 combinations of the above hyperparameters were tested based on their performance in evaluation after training as displayed in Figure 14. The top 5 performing combinations are listed in table 4. Like the TRPO graph, the total evaluation reward was the sum of all time steps where the car was successful in keeping to the metrics defined in Task B. The significant difference in evaluation values was due to trajectories of certain simulations terminating far earlier than others, thus achieving a lower result.

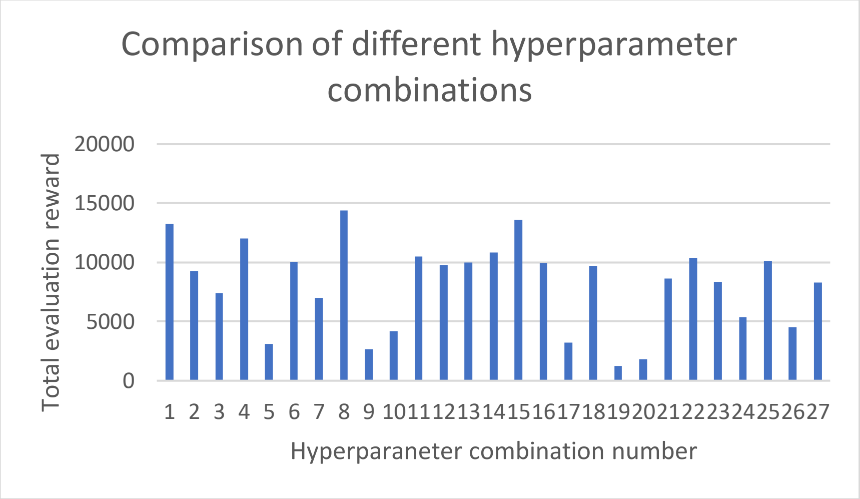

Figure 14

Combination number	Learning rate	Batch size	Gamma	Total evaluation performance
8	0.0005	64	0.99	14380
15	0.0005	128	0.99	13605
1	0.005	64	0.99	13291
4	0.005	128	0.8	12030
22	0.0005	32	0.99	10363

Table 4

Given the significant improvements in training observed with a lower learning rate and high gamma, the hyperparameter combination number 8 was selected to train the final model before evaluation.

2.4 – Final result:

The output of the TRPO algorithm on the training road is shown in Figure 15. While the algorithm closely followed the road at the first curve, its movement becomes more jagged and unsteady after the sharp curves despite further tuning.

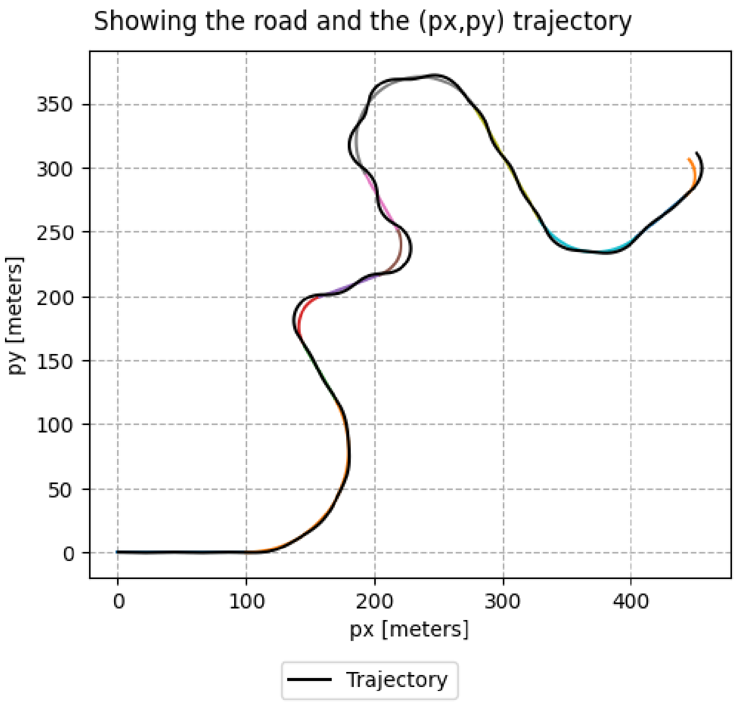

Figure 15

Applied to the evaluation road in Figure 16, the performance was similar to the training road. Overshooting on corners was still a present issue, but the car was able to follow the road relatively well until the ending section, where it struggled to stay on course. The overshooting can be attributed to the increased speed margins placed on the car, as well as less weight to follow the road angle precisely.

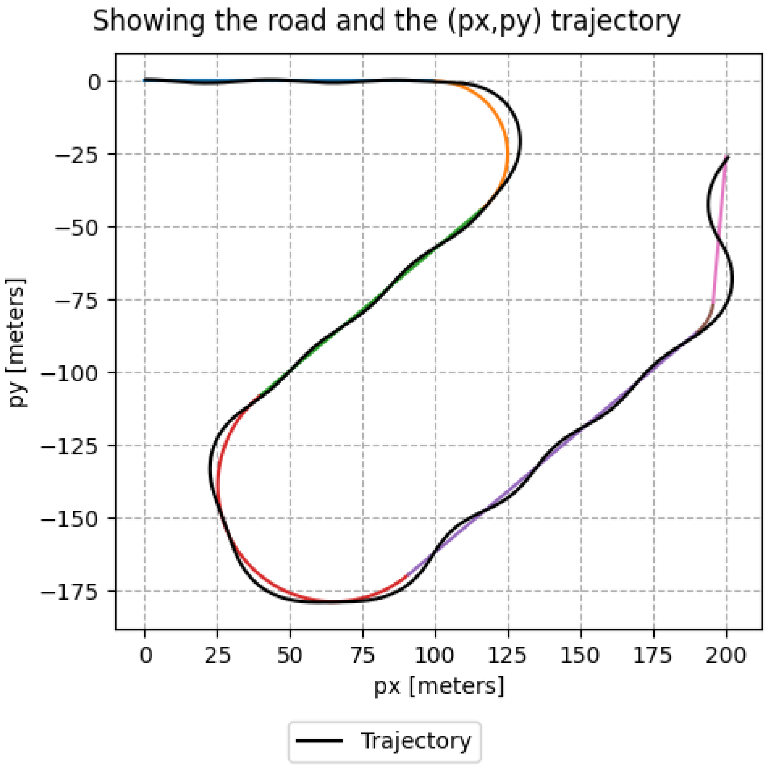

Figure 16

D – Comparative Policy Analysis:

For both evaluation roads, an additional function was added that measured the total performance rewards based on the values from Part B. These results are displayed in Table 5 below. All metrics except for time are expressed as a percentage of timesteps that the policy was successful divided by the total timesteps in the simulation.

	PPO	TRPO
Total timesteps	1735	1613
Acceleration	100%	98%
Angular acceleration	100%	67%
Distance to road center	70%	78%
Speed limit	100%	100%

Table 5

Considering the results above with respect to the trajectories displayed in Figure 9 and 16, it is clear that both algorithms achieved their goals given certain tradeoffs. The TRPO algorithm was faster than the PPO by roughly 100 timesteps, equivalent to 5 seconds. This was likely due to the stricter reward function it had on maintaining a high speed. The acceleration training function also worked well to prevent excessively uncomfortable high acceleration.

The PPO policy on the other hand took longer to complete the course but maintained 100% success in controlling acceleration and angular acceleration for a perfectly comfortable driving experience. As evidenced in its trajectory, the policy also prevented excessively high overshoots present in the TRPO algorithm, likely due to over speeding before turns. However, stronger oscillations led to a worse ability to keep to the center of the road overall.

Having analysed the results, the team agrees that Pareto optimality has not yet been achieved. Fusing the steering angle training function from the PPO to the TRPO policy will likely improve its 67% success in maintaining an acceptable magnitude of steering. However, other aspects such as the overshooting when turning corners relative to the time of completion is likely a Pareto-optimal situation, as reducing the speed training function to lower values will sacrifice its ability to be faster for the rest of the course.

E – Reflections from Industry Lectures:

1 – Arya (1439683):

1.1 – Addressing Domain Coverage:

The guest lecture by Peter Cudmore on addressing domain coverage highlighted strategies like reducing model complexity, data augmentation, and adversarial techniques for improving machine learning model performance. While his talk focused on computer vision applications, these methods are also relevant for control tasks like the model predictive control and reinforcement learning policies explored in this subject.

For example, in the line following robot project, we could have employed data augmentation by collecting data with added noise or variations to improve the policy's robustness. This may have helped the sim-to-real transfer by making the policy less sensitive to real world variability unseen in simulation. Reducing model complexity could also have been beneficial; using a simpler vehicle dynamics model may have enabled faster and more stable policy learning compared to a complex model.

These strategies relate strongly to the model predictive control task as well. A lower complexity linear time-invariant model likely would have allowed longer control horizons and faster optimization. Data augmentation methods like domain randomization could potentially improve an MPC's performance when transferring a policy from simulation to the physical system. In future MPC projects, explicitly addressing domain coverage as described in this lecture could be highly impactful.

2.1 – Defining an Autonomous System:

Michael Crump's overview of how BAE Systems approaches developing autonomous systems aligned closely with the methodologies used in projects throughout this subject. He emphasized holistic system engineering involving thorough analysis of use cases to derive requirements, architecting appropriate components, and design iteration.

In the prosthetics control presentation I delivered, a similar process was followed to review use cases for amputees, identify key functional requirements, explore EMG and computer vision components for capturing user intent, and research neural network architectures capable of decoding movement intents.

The modular, end-to-end viewpoint was also critical for the model predictive control line following robot project. We began by defining the use case of having an RC car autonomously track a line marked on the ground. Based on this use case, we determined the need for a downward-facing camera to provide line position data. We then explored different control policies, including PID, state feedback, and MPC controllers, to fulfill the line tracking requirement. At each stage, we drew upon the physical dynamics model, control algorithms, and line detection software to link together an integrated solution.

By consistently tying together the motivating use case, sensing capabilities, model architectures, performance metrics, and real-world constraints, we were able to incrementally construct systems that addressed the applications. This systems-level, use-case-driven approach provides a template I can apply during future autonomous systems development in my career.

2 – Tian (1078368):

2.1 – What is autonomy:

In the BAE guest lecture, a particular slide that stood out to me was the one that sought to define what "autonomous" really meant in regard to robotics. "20 years ago, a robot that was considered autonomous then would be considered remote controlled now". In the field of fast moving technology it is no surprise that such definitions and expectations would change after 2 decades, but this was interesting to me because it made me consider what we would consider autonomy 20 years from now. Perhaps by then, the control systems and reinforcement learning we were studying may not even be considered autonomous in some way.

This became more interesting when I began researching for my oral presentation on the topic of the Boeing 737-MAX MCAS. Even then, there were many articles that stated different definitions for the MCAS, as either an autonomous system, or as a pilot assist tool. Having listened to other presentations in my group about incidents of AI driving cars crashing and causing fatalities, it made me further reflect on the importance of definitions in this fast-moving space of technology.

Framing something as remote controlled or autonomous, or as an AI or a pilot assist tool can have significant impacts on how these technologies are treated legally and in how they are perceived by the public. Given that a failure of an autonomous system would be placed at engineering company which created it, it may be possible that truly autonomous products may still be defined as a tool or assistant to avoid these negative connotations should the system fail. To conclude my reflection, I believe that these definitions will probably cause major headaches for my classmates in law in the near future.

2.2 – Future of AI and robotics in Australia:

In the Boeing guest lecture, researcher Yan Yang discussed the issues of bias and overtraining in AI training around her work at Boeing working on the Loyal Wingman. This was interesting to me as I had learnt about bias in training in a previous AI related subject, and its potentially disastrous effects in the real world when applied to critical industries.

Although bias was not directly touched upon in this subject, the lecture still made me reflect on the potential risks of biased training data on the performance of these vehicles. Perhaps the pavement type the vehicles are trained on are only specific to some countries, and elsewhere it may not perform as expected due to the lack of training data. Additionally, certain situations like Melbourne's hook turns and trams may also be underrepresented in typical datasets, leading to a higher likelihood of performance issues should self driving cars be implemented here. Taking this further, more chaotic driving conditions that I have seen in parts of China and South East Asia may also not be adequately represented, as drivers can behave differently particularly around large intersections. Having reflected on the road conditions that I have seen, it is clear to me now that the herculean task of implementing self driving is even harder than I imagined when accounting for the edge cases which most datasets do not cover.

References:

Bicycle NSW. (2020). BIG COUNTRY, WIDER CAR LANES?.

Craft, M. (2023). What are the average dimensions of a car?. www.carsguide.com.au/

de Winkel, K. N., Irmak, T., Happee, R., & Shyrokau, B. (2023). Standards for passenger comfort in automated vehicles: Acceleration and jerk. Applied Ergonomics, 106, 103881.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., & Madry, A. (2020). Implementation matters in deep policy gradients: A case study on ppo and trpo. arXiv preprint arXiv:2005.12729.

Lapan, M. (2018). Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more. Packt Publishing Ltd.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). The surprising effectiveness of ppo in cooperative multi-agent games. Advances in Neural Information Processing Systems, 35, 24611-24624.