7Z10SS Masters Project NPC ToR Coversheet

|  |  |
| --- | --- |
| Department of Computing and Mathematics  Computing and Digital Technology Postgraduate Programmes  Terms of Reference Coversheet | |
| Student name: | Jack Bernard Andrew Atherton |
| University I.D.: | 23720700 |
| Academic supervisor: | Dr Peng Wang |
| External collaborator (optional): |  |
| Project title: | Evolutionary Learning in Autonomous Vehicles: Towards Self-Driving Cars That Adapt and Improve Over Time |
| Degree title: | MSc Artificial Intelligence |
| Project unit code: | 6G7V0007\_2324\_9F |
| Credit rating: |  |
| Start date: | 01/06/24 |
| ToR date: | 05/06/24 |
| Intended submission date: | 27/09/24 |
| Signature and date student: | Jack Bernard Andrew Atherton (05/06/24) |
| Signature and date external collaborator (if involved): |  |

This sheet should be attached to the front of the completed ToR and uploaded with it to Moodle.

MMU 1 CMDT

**Manchester Metropolitan University**

**MSc Computing**

**2023/2024**

**Project Terms of Reference**

**Jack Atherton (23720700)**

**Project Protocol**

**1. Project Title:**

Evolutionary Learning in Autonomous Vehicles: Towards Self-Driving Cars That Adapt and Improve Over Time

**2. Student Information:**

Name: Jack Atherton   
I.D.: 23720700   
Degree Title: MSc Artificial Intelligence   
Supervisor: Peng Wang

**3. Background and Rationale:**

Autonomous vehicles have developed immensely based on the advancements of sensors, learning machines, and artificial intelligence to the extent that the goal of a driverless car can be envisaged shortly. However, the major remaining challenge is how to guarantee reliable and efficient operations in such highly complex and dynamic environments. Traditional programming techniques often fail to work under such uncertain and varied real-world driving conditions; for this, a solution is needed that is far more adaptable and robust.

This project will address this approach to solving problems in the natural world by using evolutionary algorithms, a way of working that mirrors the processes of natural selection. The algorithms improve solutions through successive generations of testing and refinement as species change and adapt over time to their surroundings. From there, the algorithms find optimal settings and policies that enable autonomous vehicles to cope through evolutionary algorithms by iterating neural network controllers with several driving conditions.

The methodology intends to structure an evolutionary process that entails controllers based on neural networks subjected to a diverse and challenging driving task in a closed environment. In successive iterations, the controllers are henceforward subjected to tests, evaluation, and fine-tuning. During the evolutionary process, the best controllers that drive most effectively are selected to become the groundwork for the next generation. It comprises the variations within its process that can result in still more improvements. This process, carried over generations, paves the way for controllers of optimum configuration, which will be able to deal with diverse, not entirely predictable situations on the road.

The process adopts another approach, that evolutionary algorithms are used in developing autonomous vehicles much safer and more competent in dealing with real-world driving complexities. This target is to adjust self-driving cars to new and unforeseen conditions, thus rendering them stable and effective under ultimately conditions. The process might well bring a great leap forward into the equivalent field of autonomous driving to develop vehicles capable of making safe and efficient journeys across the labyrinthine and dynamic landscape.

**4. Objectives:**

1. Design a realistic driving simulation environment.   
2. Implement a genetic algorithm framework to evolve neural network controllers.   
3. Develop fitness functions to evaluate performance.   
4. Conduct experiments comparing evolutionary methods to traditional approaches.   
5. Test the robustness and adaptability of the evolved controllers.   
6. Document and present findings.

**5. Approach:**

**Simulation Environment:**

Tools: Unity and Blender to create realistic driving scenarios.   
Setup: Define a track and various driving conditions (e.g., weather, traffic).

**Genetic Algorithm Framework:**

Implementation: Use DEAP framework to create and evolve neural network controllers.   
Process: Initialize a population of controllers, evaluate performance, select top performers, apply genetic operations (mutation and crossover), and repeat over multiple generations.

**Fitness Functions:**

Metrics: Completion time, collision rate, and path accuracy to score the performance of controllers.

**Comparative Experiments:**

Baseline Methods: Compare the evolved controllers against traditional rule-based and machine-learning approaches.   
Data Collection: Record performance metrics for analysis.

**Robustness and Sensitivity Testing:**

Different Scenarios: Test controllers in various driving conditions (like different weather or traffic) to see how well they adapt.   
Sensitivity Checks: Analyze how changes in settings affect car performance to ensure they are robust.

**Qualitative Feedback:**

Expert Opinions: Get feedback from experts in autonomous driving to gather insights and suggestions for improvement.

**Documentation and Reporting:**

Record Keeping: Keep detailed notes of all experiments and settings for transparency.   
Statistical Analysis: Use statistical methods to validate and interpret results.

**6. Resources Required:**

Simulation Platforms: Unity, Blender   
Programming Languages: Python, C++   
Deep Learning Frameworks: TensorFlow, PyTorch/Keras   
Evolutionary Algorithm Framework: DEAP   
Documentation Software: Microsoft Word   
Research Materials: Relevant literature and datasets

**7. Evaluation:**

Performance Analysis: Compare evolved controllers to traditional methods based on collision rates, completion times, and adaptability.   
Robustness Testing: Evaluate performance across diverse scenarios.   
Statistical Validation: Confirm the significance of results through statistical analysis.

**8. Timeline:**

Month 1-2: Literature review, initial setup of simulation environment.   
Month 3-4: Implement genetic algorithm framework and develop fitness functions.   
Month 5-6: Conduct initial experiments and refine algorithms.   
Month 7-8: Comparative analysis and robustness testing.   
Month 9: Gather qualitative feedback and perform sensitivity analysis.   
Month 10: Document findings and prepare the final report.

**9. Ethical Considerations:**

Ensure that all simulations and experiments adhere to ethical standards and do not harm individuals or real-world systems.