Modelação e Simulação Project Proposal

David Scarin 202108314 Francisco Silva 202109000
Tomás Rodrigues 202107937

October 25, 2024

Contents

1 Project Description

The objective of this project is to simulate and evaluate reinforcement learning algorithms that enable autonomous vehicles to perform complex driving tasks, such as merging, lane changing, and overtaking, while ensuring safety and optimizing traffic flow.

Specifically, the project will focus on simulating scenarios at road intersections where multiple autonomous vehicles must determine the order of crossing and appropriate speeds, aiming to maximize traffic efficiency and minimize collision risks.

This will be achieved using an agent-based simulation framework, where a group of four agents, each trained with different Deep Reinforcement Learning (DRL) policies, will face dynamic environmental changes. The agents will be tested under various combinations of environmental variables (e.g., traffic density, intersection layout, vehicle speed) to evaluate their behavior and decision-making processes in real-time. This experimental setup will create a Multi-Agent System (MAS) within a dynamic environment, concentrating on decision-making strategies for autonomous driving.

Through this project, we aim to address critical questions regarding the scalability and robustness of DRL models in autonomous driving. In particular, the research will explore how well these algorithms generalize to unseen environments—a key challenge for deep learning-based approaches. The insights gained can inform the real-world applicability of DRL for autonomous vehicles, especially in unpredictable or changing environments.

2 Goals of the Simulation Project

This project will consist of three key stages, each with specific goals that contribute to the overall aim of developing a robust and adaptable Multi-Agent System (MAS) for managing autonomous vehicles at road intersections. The stages are as follows:

1. **Successfully train the agents**

   In this phase, we will apply selected Deep Reinforcement Learning (DRL) algorithms to train autonomous agents in navigating a road intersection.

   The agents will learn to make decisions related to intersection crossing order and speed control, with the goal of optimizing traffic flow and avoiding collisions.

   Goal: By the end of this stage, the agents will be capable of autonomously navigating the intersection under standard conditions. This will serve as the baseline scenario, providing a reference for the system's performance before any environmental variables (such as traffic flow or lane configurations) are altered.

2. **Introducing System Perturbations**

In this stage, we will systematically vary two key factors—traffic flow and lane configuration at the intersection. This will involve changing the number of vehicles approaching the intersection and altering the layout of the lanes (e.g., adding turn lanes or modifying lane assignments). The aim is to evaluate how these changes affect the behavior of the MAS and the agents' decision-making processes.

Goal: The objective here is to assess how the MAS adapts to different intersection conditions, focusing on the system's ability to maintain traffic efficiency and safety under varying traffic densities and lane configurations. We will measure the agents' performance in terms of traffic flow optimization, collision avoidance, and decision-making effectiveness in response to these changes.

3. **Evaluate system scalability and robustness**

   The final phase will focus on evaluating the scalability and robustness of the MAS under more challenging or extreme conditions. This will involve increasing the complexity of traffic flow scenarios (e.g., sudden surges of vehicles) and testing different intersection configurations (e.g., multi-lane intersections or altered traffic signal patterns).

   Goal: The end goal is to assess the system's scalability—how well it handles higher traffic volumes and more complex lane setups—while ensuring that safety and traffic optimization are maintained. Additionally, this phase will evaluate the robustness of the agents' learned policies, particularly their ability to generalize to new and unforeseen intersection configurations or traffic patterns.

This approach ensures that each stage of the project builds toward understanding how well the MAS performs under varying traffic and lane conditions, with a focus on scalability, robustness, and real-world applicability at road intersections.

3 Models of Decision Support Considered

Assuming that [5]:

- Descriptive models help us understand the agents' current decision-making behavior.
- Normative models provide ideal benchmarks for comparison and optimization.
- Predictive models allow to anticipate how the system will behave under future conditions.
- Prescriptive models guide agents in making the best possible decisions under specific circumstances.

- Speculative models explore the system's performance in hypothetical or extreme future scenarios, ensuring adaptability and long-term robustness.

This simulation project can be classified under multiple decision support model categories, as it touches upon different aspects of decision-making processes. Here's how the project fits into each of the categories:

1. **Descriptive**

   The project has a strong descriptive component. During the initial stages, the agents are trained using DRL algorithms, and their behaviors are observed and recorded in a baseline scenario. This phase is focused on describing and understanding how the autonomous agents make decisions (e.g., crossing order, speed adjustment) based on the environment they are exposed to. The descriptive element helps us analyze how agents behave without interference and forms the basis for comparison with later stages.

2. **Normative**

   This project implicitly contains a normative aspect. The desired outcomes for autonomous vehicle behavior (e.g., maximizing traffic flow while minimizing collisions) are rooted in optimal decision-making criteria. Although the project does not directly implement normative models, the success criteria for the agents—safe and efficient decision-making—are benchmarks derived from an ideal (normative) vision of how the system should behave in various traffic conditions.

3. **Predictive**

   The project can be seen as predictive in its later stages. After training the agents, one of the goals is to predict how the multi-agent system (MAS) will perform under different traffic fluxes and lane configurations. The project aims to simulate a variety of scenarios (perturbations) and observe how the agents adapt. Through this, the system's future performance in unseen scenarios can be forecast, making predictive modeling a key aspect of the evaluation.

4. **Prescriptive**

   The project has a prescriptive element. The goal of DRL in this context is to prescribe optimal actions to agents in real-time traffic scenarios. By learning policies that guide the agents' decisions (e.g., what crossing order to take, when to accelerate or decelerate), the system effectively prescribes the best possible actions to optimize traffic flow and minimize collision risk. As the agents are trained to follow the best course of action in specific conditions, the project fits well within the prescriptive decision support model.

5. **Speculative**

The project could involve some speculative modeling, though this is not a primary focus. Testing agent performance in hypothetical situations would fall under this category, helping assess how flexible and adaptive the MAS is. However, speculative modeling is a minor aspect, as the primary focus is on concrete variations in traffic and lane configurations.

Based on these observations, the primary classification of the project would be **descriptive**, **predictive**, and **prescriptive**, as it involves understanding current decision-making, predicting agent behaviors under different conditions, and prescribing optimal actions for traffic optimization and safety.

Normative models are more implicit, serving as ideal benchmarks, while speculative elements might arise if the project explores hypothetical or extreme scenarios.

In addition to the decision support model categories, the simulation model can be classified as:

Dynamic: The model evolves over time as agents continuously interact with the environment, make decisions, and adapt to changing conditions at road intersections. The system state changes as traffic flows and lane configurations adjust.

Stochastic: The simulation includes elements of randomness, such as variations in traffic flow, vehicle arrival patterns, and potential uncertainties in agent behavior. This introduces probabilistic outcomes and variability across different simulation runs.

Discrete: The model operates in discrete time steps, where the agents' actions (e.g., crossing intersections, adjusting speed) are evaluated at specific intervals. Each decision is made at distinct time points, typical in agent-based simulations where agent behaviors are updated step-by-step.

These classifications emphasize the dynamic, probabilistic, and step-wise nature of this agent-based simulation, where individual agents make decisions over time.

3.1 Main Entities of the System

In the context of the simulation project focusing on autonomous vehicles navigating a road intersection, the main entities are the objects of interest that interact with each other and the environment dynamically. The primary entities of the system are:

1. **Autonomous Vehicles (Agents):**

   Each autonomous vehicle represents an agent in the simulation with decision-making capabilities. These vehicles are created, move around, change speed, and interact with other agents. They may enter and leave the system as they pass through the intersection.

   Attributes: Speed, position, direction, assigned policy (decision-making model), current lane.

   Resources they compete for: Road space, crossing priority, lanes.

2. **Intersection (Road Infrastructure)**:

   The road intersection is a static entity but a key part of the system. It defines where vehicles meet and interact. Different types of lane configurations or intersection designs can be applied.

   Attributes: Number of lanes, intersection layout.

   Resources they compete for: Lane capacity (the number of vehicles that can use a lane or section of road at a time).

3. **Traffic Flow:**

   Represents the overall movement of vehicles through the system. This is a dynamic entity in terms of the rate at which vehicles arrive at and depart from the intersection.

   Attributes: Arrival rate of vehicles, traffic density, vehicle types (e.g. vehicles, ego-vehicles).

   Resources they compete for: Access to the intersection, road segments.

4. **Agent Policies (Decision-Making Models):**

   Each autonomous vehicle (agent) operates based on a decision-making policy (learned behavior from DRL). These policies guide how each vehicle responds to other vehicles and environmental factors.

   Attributes: Learned policy, decision rules, reward function (for reinforcement learning).

   Resources they compete for: Computational resources for decision-making (though implicit in the model), control over vehicle behavior.

3.2 Variables of the System

Variables in the system represent pieces of information that reflect characteristics of the entire system, not of specific entities. These variables are either directly influenced by the system dynamics or serve as global parameters for the simulation.

1. **Traffic Density**: Reflects the overall number of vehicles within the system or passing through the intersection at any given time.

   Role: Affects how agents navigate and make decisions based on the number of other vehicles in the system.

2. **Traffic Flow Rate (Arrival Rate of Vehicles):** Indicates the rate at which vehicles enter the simulation, typically measured as vehicles per time unit (e.g., vehicles per minute).

   Role: It influences system congestion and vehicle interactions at the intersection.

3. **Intersection Configuration (Lane Layout):** Represents the structure of the intersection, such as the number of lanes or the presence of dedicated turn lanes.

   Role: Changes in this variable affect how vehicles maneuver and interact with each other.

4. **Average Wait Time:** This is a system-level variable that measures how long, on average, vehicles must wait to cross the intersection.

   Role: It reflects the system's efficiency and can be used to assess the performance of different policies.

5. **Collision Rate:** A key variable that reflects the number of collisions or near-collisions occurring in the system.

   Role: A measure of the system's safety, which influences the evaluation of agent decision-making policies.

6. **System Throughput:** The number of vehicles successfully passing through the intersection over a given period.

   Role: A variable that reflects the overall efficiency and capacity of the system, indicating how well it handles different traffic conditions.

State of the System:

The state of the system at any given time would include a collection of variables such as the number of vehicles in the system, traffic density, the intersection configuration, and the real-time position and speed of each vehicle.

The state contains all the necessary information to describe the system's current dynamics and predict future behaviors.

These elements collectively define how the simulation operates, how decisions are made, and how the performance of the system is evaluated.

4 Operation Policies to be Tested (scenarios)

The simulation project will involve three groups of four agents, each trained using Deep Reinforcement Learning (DRL) based on the DQN algorithm, with different policies (MlpPolicy, CnnPolicy, and Social Attention Mechanisms).

The DQN algorithm (Deep Q-Network) is a reinforcement learning technique that enables agents to learn optimal policies by estimating the value of action-state pairs. The algorithm utilizes neural networks to approximate Q-values, guiding agents in selecting actions that maximize long-term rewards.

We will compare agent's behavior based on three distinct policies:

1. **MlpPolicy (Multi-Layer Perceptron Policy)** The MlpPolicy utilizes a multi-layer perceptron neural network to process environmental observations and make decisions. This policy focuses on key input features, such as speed, distance to other vehicles, and intersection layout, allowing agents to act based on their immediate surroundings.

Scenario: Agents following the MlpPolicy will navigate the intersection by leveraging their observed features, applying learned behaviors to respond effectively to dynamic traffic conditions.

Objective: Assess the performance of MlpPolicy agents in terms of efficiency (e.g., throughput, wait times) and safety (e.g., collision rates) compared to agents using other policies. Analyze the influence of input features on their decision-making processes.

2. **CnnPolicy (Convolutional Neural Network Policy)** The CnnPolicy employs convolutional neural networks to process visual and spatial data, making it particularly effective in scenarios where visual inputs (like grid representations of the intersection) are crucial for decision-making.

   Scenario: Agents using the CnnPolicy will interpret complex visual representations of their environment, allowing them to make informed decisions regarding lane changes, merging, and crossing orders.

   Objective: Evaluate the performance of CnnPolicy agents in diverse traffic patterns, focusing on their ability to recognize spatial configurations and respond appropriately to traffic dynamics.

3. **Social Attention Mechanisms Policy** This policy integrates social attention mechanisms, allowing agents to observe and interpret the behaviors and intentions of nearby vehicles. This approach helps agents develop cooperative strategies and optimize their interactions with other vehicles. The attention architecture was introduced to enable neural networks to discover inter-dependencies within a variable number of inputs[3]

   Scenario: Agents utilizing this policy will actively monitor the behavior of surrounding vehicles, adapting their actions based on predicted movements and intentions of others.

   Objective: Investigate the effectiveness of social attention in enhancing cooperation among agents, potentially leading to improved traffic flow and reduced collision rates. Compare the performance of agents using Social Attention Mechanisms against those following other policies to assess their overall effectiveness in a multi-agent environment.

4.1 Testing Framework for Operation Policies:

1. **Experimental Scenarios:** Each policy will be tested under various conditions, such as:

   High Traffic Density: Increased number of vehicles entering the intersection simultaneously.

   Low Traffic Density: Sparse vehicle presence to evaluate agent behavior in less congested environments.

   Variable Lane Configurations: Different arrangements of lanes (e.g., dedicated turn lanes, straight lanes) to assess adaptability.

2. **Group Configuration**: Each group, consisting of four agents using the same policy , will be tested in each scenario.

   This allows for direct comparisons between the different policies under identical traffic conditions.

3. **Performance Metrics:**

   (a) **Efficiency**: Metrics such as average wait time, throughput (vehicles per time unit), and average speed of agents.

   (b) **Safety**: Collision rates, near-misses, and compliance with traffic rules.

   (c) **Adaptability**: How well agents generalize their learned behaviors to new traffic scenarios (e.g., changes in traffic density, lane configurations).

The idea is to be able to produce at the end of the project a table similar to the following one:

Scenario	**Policy**	**Efficiency (Wait Time, Throughput, Speed)**	**Safety (Collision Rates, Near-Misses)**	**Adaptability (Generalization to New Scenarios)**
Baseline	MlpPolicy	Avg. Wait: 12s, Throughput: 30 veh/min, Speed: 25 km/h	5 collisions,3 near-misses	Moderate
	CnnPolicy	Avg. Wait: 10s, Throughput: 32 veh/min, Speed: 28 km/h	3 collisions, 2 near-misses	High
	Social Attention Mechanisms	Avg. Wait: 8s, Throughput: 35 veh/min, Speed: 30 km/h	1 collision, 1 near-miss	Very High
High Traffic Density	MlpPolicy	Avg. Wait: 15s, Throughput: 28 veh/min, Speed: 20 km/h	4 collisions, 2 near-misses	Moderate
	CnnPolicy	Avg. Wait: 12s, Throughput: 30 veh/min, Speed: 22 km/h	2 collisions, 1 near-miss	High
	Social Attention Mechanisms	Avg. Wait: 10s, Throughput: 32 veh/min, Speed: 25 km/h	1 collision, 0 near-misses	Very High
Variable Lane Configurations	MlpPolicy	Avg. Wait: 18s, Throughput: 26 veh/min, Speed: 18 km/h	7 collisions, 4 near-misses	Low
	CnnPolicy	Avg. Wait: 14s, Throughput: 28 veh/min, Speed: 20 km/h	4 collisions, 2 near-misses	Moderate
	Social Attention Mechanisms	Avg. Wait: 10s, Throughput: 30 veh/min, Speed: 22 km/h	2 collisions, 1 near-miss	High

Table 1: Example of Simulation Scenario Framework and outcome in terms of results

5 Key Performance Indicators (KPI)

The following Key Performance Indicators (KPIs) and decision criteria will be used to effectively assess the performance and effectiveness of the operational policies:

1. **Traffic Flow Efficiency**

   Metric: Average vehicle throughput (vehicles per hour)

   This measures how well the intersection handles traffic. A higher throughput indicates that the system is efficiently managing vehicle movement.

2. **Safety Metrics**

   Metric: Collision rate (collisions per 1000 vehicle crossings)

   A crucial indicator of how safe the intersection is for autonomous vehicles. Reducing collisions is a primary goal for any traffic management system.

3. **Average Wait Times**

   Metric: Average wait time per vehicle (seconds)

   This provides insights into how quickly vehicles can navigate the intersection. Shorter wait times can lead to increased overall satisfaction for users and improved efficiency.

4. **Travel Time through the Intersection**

   Average travel time per vehicle (seconds)

   Evaluating the time it takes for vehicles to cross the intersection helps assess the effectiveness of different policies in facilitating movement.

5. **Agent Cooperation Rate**

   Percentage of successful cooperative maneuvers (e.g., yielding or merging)

   This metric will help evaluate how well agents interact and cooperate with each other, which is essential for improving traffic flow and safety in multi-agent scenarios.

6. **Adaptability to Varying Traffic Conditions**

   Performance variance (e.g., comparing metrics like wait times and throughput under different traffic densities)

   This evaluates how well agents can adjust their behavior to different scenarios, which is important for real-world applications.

5.1 Decision Criteria:

1. **Efficiency vs. Safety Trade-off**

   Criteria: Determine the balance between maximizing vehicle throughput and minimizing collisions.

   Finding an acceptable compromise between these two factors is essential for developing effective traffic management solutions.

2. **Robustness of Policies**

   Criteria: Assess the stability and consistency of performance across different operational policies.

   Policies that yield reliable performance across varying conditions can be considered more effective and trustworthy.

3. **Generalizability of Agent Behavior**

   Criteria: Evaluate how well agents trained under specific conditions can adapt to new, unseen scenarios (e.g., changes in traffic density or lane configurations).

   A system that generalizes well can be more easily implemented in real-world applications.

4. **Effectiveness of Cooperation Strategies**

   Criteria: Assess how cooperative policies improve overall system performance, such as traffic flow and safety.

   Effective cooperation among vehicles can significantly enhance the dynamics of traffic systems.

5. **Realism and Applicability Criteria**:

   Ensure that policies tested in simulations are feasible and realistic for implementation in actual autonomous vehicle systems.

   The ultimate goal is to develop solutions that can be applied in real-world settings, making this criterion essential.

These KPIs and decision criteria, will allow to assess the performance of different operational policies in the context of autonomous vehicles navigating a road intersection.

This structured evaluation will provide insights into which policies are most effective for enhancing traffic flow and safety.

6 Data Requirements for Highway Intersection Simulation

In our case the required data are the assumptions for the training/testing and simulation processes parameters:

1. **Intersection Configuration Data**

   Detailed layout of the highway intersection, including:

   - Number of lanes for each direction.

   - Lane markings (e.g., merging zones, turn lanes).

   - Traffic signal placements and timings.

2. **Traffic Flow Data**

   Information about typical traffic patterns and volumes approaching and crossing the intersection.

3. **Vehicle Characteristics**

   Specifications of the types of vehicles expected to navigate the intersection, including:

   - Vehicle types (ego-vehicles or "general" vehicles).

   - Speed and acceleration profiles.

   - Dimensions (length, width, height).

4. **Agent Behavior Profiles**

   Defined strategies for agent decision-making at the intersection, focusing on:

   - Merging behaviors when entering the intersection.

   - Lane-changing strategies based on surrounding traffic.

   - Decision-making in response to traffic signals.

5. **Traffic Regulation Data**

   Information about applicable traffic laws and signals specific to highway intersections, such as:

   - Speed limits in the intersection area.

   - Right-of-way rules (who yields to whom).

   - Signal timings (green, yellow, red durations).

6. **Performance Metrics Data**

   Metrics to assess the performance of the agents and overall system efficiency, including:

   - Collision rates (number of collisions per 1,000 vehicle crossings).

   - Average wait times for vehicles at the intersection.

   - Average throughput (vehicles passed per hour).

7. **Simulation Parameters**

   Settings governing the simulation environment, including:

   - Time intervals for updates (e.g., how often the simulation state is refreshed).

   - Total duration of the simulation run.

8. **Training Data for DRL Models**

   Data required for training the reinforcement learning agents, including:

   - Reward structures based on traffic efficiency (e.g., rewards for safe crossings).

   - Scenarios representing various intersection conditions for training.

7 Simulation tools, environments, languages

The simulation will be conducted within the Highway Env[1] environment, which provides both a physics engine for simulating vehicle dynamics and a rendering engine for visualizing the environment.

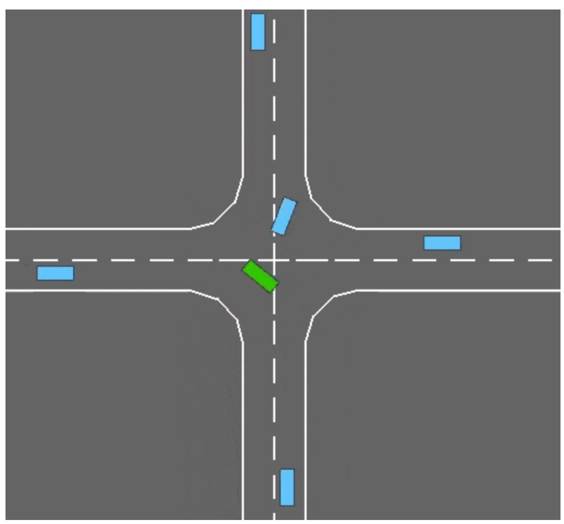

Figure 1: Intersection Environment [1]

Baseline agents will be trained using the Stable Baselines3 library [6] and PyTorch[4], leveraging its implementation of reinforcement learning algorithms optimized for complex environments[2].

All code will be developed in Python, ensuring compatibility with machine learning frameworks, simulation tools, and data analysis libraries essential for this project.

References

[1] Edouard Leurent. *An Environment for Autonomous Driving Decision-Making*. https://github.com/eleurent/highway-env. 2018.

[2] Edouard Leurent. *rl-agents: Implementations of Reinforcement Learning algorithms*. https://github.com/eleurent/rl-agents. 2018.

[3] Edouard Leurent and Jean Mercat. *Social Attention for Autonomous Decision-Making in Dense Traffic*. Presented at the Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Vancouver, Canada: NeurIPS, 2019.

[4] *PyTorch Module*. https://pytorch.org/. 2024. URL: https://pytorch.org/.

[5] Rosaldo Rossetti. *Systems Modelling and Simulation 1: Introduction to Simulation*. https://moodle2425.up.pt/pluginfile.php/149885/mod_resource/content/1/02b_MS_SystemsModelling.pdf. Lecture slides, FEUP – Faculty of Engineering, University of Porto. LIACC – Artificial Intelligence and Computer Science Lab, 2024.

[6] *Stable Baselines 3*. https://github.com/DLR-RM/stable-baselines3. 2024. URL: hhttps://github.com/DLR-RM/stable-baselines3.