Workshop 3

Brayan Sierra Diaz
20212020036

Daniel Alejandro Presiga
20212020116

Professor Carlos A. Sierra
Systems Engineer
Universidad Distrital Francisco José de Caldas
Bogotá DC

June 13, 2025

Technical Implementation

Throughout the first two workshops, we progressively built the conceptual and structural foundations of an intelligent autonomous agent. In Workshop 1, we specified a modular system architecture that involved the integration of sensors and actuators to simulate a realistic vehicular agent. Workshop 2 added dynamical system thinking to the project, analyzing feedback loops, stability concerns, and the formulation of a mathematical model inspired by Bellman's equation.

Building upon those foundations, this third workshop focused on technically implementing a **Deep Q-Network (DQN)** agent capable of perceiving and adapting to its environment through learning. The agent was designed to navigate a simulated urban scenario while learning to make decisions that optimize travel time and legal behavior (e.g., obeying traffic lights and avoiding collisions).

We used the `Stable-Baselines3` library along with `Gymnasium` to create and train the agent in a custom environment. The agent's state was represented as a 4-dimensional vector: vehicle speed, proximity to the nearest object, status of the traffic light, and time. The action space included discrete choices: accelerate, maintain speed, or decelerate.

To structure learning, we designed a **reward function** that encouraged appropriate driving behavior. For instance:

- Negative rewards for crossing red lights or driving too close to another car.
- Positive rewards for respecting traffic laws and reaching the destination quickly.

The core learning algorithm minimized the following loss function:

$$L(\theta) = E_{(s,a,r,s')}\left[\left(Q(s,a;\theta)\left(r + \gamma \cdot \max_{a'} Q(s',a';\theta^{-})\right)\right)^2\right]$$

This formulation, adapted from the Bellman equation (as discussed in Workshop 2), enabled gradient descent to improve policy decisions over episodes. A **target network** was employed to stabilize learning.

Test Scenarios

We defined multiple test scenarios to evaluate our agent in controlled conditions. These scenarios were designed with the dynamic system perspective in mind, identifying how chaotic or deterministic certain traffic elements could be.

- **Baseline Scenario:** This served as our control case. The environment had moderate traffic and predictable patterns. It was used to validate the correct operation of the learning algorithm and reward system.
- **Heavy Traffic Simulation:** The agent was placed in high-density traffic conditions. The proximity sensor frequently detected nearby vehicles, forcing the agent to slow down, reroute, or adapt.

- **Complex Routes and Detours:** Roads with intersections, dead-ends, and multiple valid paths were added to the simulation. This scenario tested the agent's ability to explore and exploit routes efficiently.

- **Sensor Degradation:** Here, we intentionally added noise to the proximity and visibility readings. This tested the robustness of our cybernetic feedback loop in conditions similar to real-world sensor failures.

- **Dynamic Reward Shaping:** Midway through training, we altered the reward structure (e.g., stricter penalties for minor traffic violations). The agent's adaptability was evaluated by how quickly it re-optimized its behavior.

Each scenario was executed across multiple runs with fixed random seeds for reproducibility.

Performance Analysis

To analyze the effectiveness of our implementation, we gathered various metrics:

- **Total Episode Reward:** This gave an overall indicator of performance. We expected it to increase as the agent learned.

- **Average Completion Time:** This directly reflected the agent's navigation efficiency.

- **Navigation Errors:** We counted rule violations, unnecessary turns, or route loops.

- **Policy Stability:** We examined how volatile the policy remained across episodes, looking for convergence patterns in the learning curve.

- **Resilience and Adaptability:** Changes to sensor noise and reward dynamics allowed us to test the cybernetic principles of adaptability and feedback loop robustness.

As expected from our system design, the agent gradually exhibited **bounded behavior** (as proposed in Workshop 2), learning to obey traffic rules, avoid obstacles, and take efficient paths. However, we observed that in some cases the agent would overfit certain routes, leading to suboptimal generalization in unfamiliar scenarios. This behavior aligns with our earlier observations about chaotic attractors in urban traffic systems.

We identified the following avenues for future enhancement:

- Introducing prioritized experience replay to avoid overfitting.

- Implementing a dueling DQN architecture to separate state-value and advantage estimations.

- Extending the system to support multi-agent coordination or competition.

References

- Farama Foundation. (n.d.). *Minigrid*. Retrieved from `https://github.com/Farama-Foundation/minigrid`

- DigitalOcean. (n.d.). *Getting started with OpenAI Gym*. Retrieved from `https://www.digitalocean.com/community/tutorials/getting-started-with-openai-gym`

- FSNDZOMGA. (n.d.). *A beginner's guide to reinforcement learning using Stable Baselines 3*. Retrieved from `https://fsndzomga.medium.com/a-beginners-guide-to-reinforcement-`

- Amin, S. (n.d.). *Deep Q-Learning (DQN)*. Retrieved from `https://medium.com/@samina.amin/deep-q-learning-dqn-71c109586bae`