WORKSHOP #1 – SYSTEMS DESIGN

Johan Sebastián Gutiérrez Pérez - 20211020098

Francisco Jose de Caldas District University

System Sciences Foundations

Eng. Carlos Andrés Sierra

April 9th, 2025

Abstract

This project presents the preliminary design of an autonomous intelligent agent based on principles of system sciences, integrating cybernetics and AI through RL (reinforcement learning). The goal is developing an adaptative agent, in this case, a drone, with the ability to deliver packages in a simulated environment composed of streets and buildings. Cybernetic principles are through virtual sensors, like proximity sensors and feedback loops that allow the drone to detect obstacles, make decisions and autoregulate its behavior. The artificial intelligence is applied through a reward system that guides the learning of the agent: the correct decisions, as delivering a package without collisions generates positive rewards while the wrong actions like crashing or taking incorrect routes are punished. This way, the agent can improve its performance over the time. This design stablishes the foundations for future implementations. This proposal is supported by tools like Gymnasium and Stable-Baseline3, that are adapted for the design of autonomous agents.

Introduction

This project present as first approach to the design of an autonomous intelligent agent that uses cybernetics and RL. The objective is to create the foundations for developing an adaptative drone that is capable of deliver packages in a simulated environment composed of streets and buildings. This environment allows implementing virtual sensors and feedback loops, which enables the drone make decision and regulate its behavior when facing dynamic situations.

The project is focused on how an agent can learn from its mistakes and optimize its decisions through a reward system. Correct decisions are positively reinforced and errors are negatively reinforced. This logic is useful for the drone since it allows it to adapt to the simulation and improve its performance.

In the context of the course, this design integrates concepts as autoregulation, modular structure and feedback, supported by AI frameworks such as Gymnasium and Stable-Baseline3.

1. System requirements

1.1. Functional Specifications

The autonomous agent designed for this project is a delivery drone in a simulated city, composed of streets and buildings. It integrates sensors, actuators and a reward system that guides its learning process.

Sensors: The drone has virtual proximity sensors that are crucial for detecting nearby obstacles, in this case, the buildings. These sensors are essential to avoid collisions and navigate along the streets. The collected information is processed continuously and used as input for de decision-making algorithm.

Actuators: The drone has virtual actuators represented by directional propellers. These actuators perform several basic actions such as go forward, go backward, turn right and turn left. These actions are executed based on the taken decisions for the RL model and also are influenced by the feedback of the environment.

Reward system: The learning process is guided by a reward function that encourages the wished behaviors and punishes the errors. For example, a successful delivery and without collisions is encouraged by a positive score. Otherwise, crashes or unnecessary detours are punished by a negative score. This way, the agent can learn optimal routes and improve them.

The environment is designed using the library Gymnasium, which integrates all these components and has a suitable base for train the agents using RL, so, the agent can perceive its environment and act accordingly.

1.2. **Use cases**

Use case 1: Secure navigation through the environment:

- Goal: The drone must move through the streets avoiding crashes.

- Description: Based on the data collected by the proximity sensors, the drone detects obstacles and adjust its direction using the actuators. If the drone avoids correctly a building, the reward system reinforces the action.
- Expected result: The drone is able to navigate without crashing and improve its skill to detect and react to obstacles.

Use case 2: Successful delivery of a package

- Goal: Reach de destination an simulate the delivery of the package.
- Description: The drone follows the route learned and reaches the delivery point. If the drone arrives without collisions or unnecessary detours, it receives a positive score.
- Expected result: The agent reinforces the sequence and actions that led it to complete successfully the task and optimize future routes.

Use case 3: Penalty for collisions or navigation errors

- Goal: Punish unwanted behaviors, such as collisions or not optimal routes.
- Description: If the drone crashes or follows an unnecessarily long route, the reward system gives a negative score.
- Expected result: The agent avoids repeating these actions and look for more optimal and safe routes.

2. System Architecture

This system is based on a modular architecture, where each component or module has a specific function and communicates with the others using a structured flow of information.

Simulated environment:

It is a virtual representation of a city with streets and buildings, developed with the library Gymnasium. This environment defines the physical rules of the system, the spatial limits and the conditions of the interactions between the agent and its environment.

Agent(drone):

The main component that acts in the simulation. The drone perceives the environment through the virtual sensors and executes actions using simulated actors. It learns from its experience via an RL algorithm.

Sensors:

The proximity sensors detect how close are the obstacles. This information is transformed into data used by the agent as inputs to make decisions.

Actuators:

The drone has directional propellers as actuators, allowing it to perform movement actions. These actions are determined by the RL model and influenced by feedback from the environment.

Reward system:

The reward system module takes and processes the information from the sensors and the environment feedback to adjust its decisions and improve its performance over the time.

Data flow:

The flow starts with the sensors that collect the information from the environment. This information is processed by the RL algorithm, which decides the next action. Then the actuators execute that action and then the environment generates a new observation and reward. This loop repeats, shaping a feedback loop.

Feedback loop:

The system has a loop where the agent perceives, acts and learns. This loop allows the autoregulation of the drone since it adjusts its behavior, promoting the adaptation to the environment and a continuous improvement.

3. Preliminary Implementation Outline

3.1. Frameworks and tools

For the design of this project, two frameworks were selected: Gymnasium and Stable-Baselines3. The main reason is that both have useful functions for simulating and training the agent.

First, Gymnasium, this framework is modern and adaptative when creating and managing simulated environments. It was chosen because it allows to implement personalized 2D environments, which is ideal for modeling the virtual city composed of the buildings and streets without using complex 3D engines. It also offers standardized interfaces for the interactions between the agent and the city.

Then Stable-Baselines3 (SB3) is used for modeling the RL system because offers a wide range of algorithms such as PPO and DQN, which are suitable for training autonomous agents in simulated environments.

Other frameworks were considered, as RLLib (designed for training on a large scale), PettingZoo (focused on multi-agent systems) and unity ML-Agents (that requires a 3D engine). Nevertheless, these frameworks involved a higher complexity or were not as optimal as Gymnasium or SB3.

3.2. Development Timeline

The development of this project is separated in steps that goes form the simplest to the most complex, so that the agent progressively learns to move and improve.

- **Step one: Create the simulated environment**
  The virtual city is designed with streets and buildings using Gymnasium. Allowing the drone can start the learning process.
- **Step two: Define the drone and its sensors**

The drone is created with virtual sensors to detect obstacles and with the basic movement actions.

- **Step three: Training with Q-learning**
  A Q-learning algorithm is used to train the drone, saving the decisions and their values in a table. This step is used to verify that all components work as expected.
- **Step four: Use DQN for improving**
  The basic table is replaced with a neural network, a more advanced algorithm that allows the agent to learn in larger and more complex environments.
- **Step five: Adjustments and tests**
  Functionality and results are reviewed to adjust parameters and enhance the agent.

References

Farama Foundation. (2022). *Gymnasium: A standard API for reinforcement learning environments*. https://www.farama.org/Gymnasium/

Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., & Dormann, N. (2021). *Stable-Baselines3: Reliable reinforcement learning implementations*. GitHub. https://github.com/DLR-RM/stable-baselines3

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). The MIT Press. http://incompleteideas.net/book/the-book-2nd.html