EE30241 Coursework – Activity 3a
Reactive behaviour and the Perceptron

Uriel Martinez-Hernandez

November 2021

Introduction

In this activity you will implement two different approaches for robot control using reactive behaviours and sensory-motor control based on the Braitenberg Vehicles approach, and robot control based on the perceptron model. For the implementation of these activities, you will use the Pioneer mobile robot in CoppeliaSim and Python language. To get started, download the zip file *'Files for CW3a-reactive_behaviour_and_perceptron'* from Moodle, where you will find the following files.

1. ***aggressionReactiveBehaviour.ttt***: A CoppeliaSim scene with the implementation of the aggression behaviour using the Braitenberg vehicles approach. This is an example that you can follow to implement the fear, love and explorer behaviours.

2. ***wallFollowingWithReactiveBehaviours.ttt:*** A CoppeliaSim scene with a prepared environment where you will implement the fear or explorer behaviour to allow the Pioneer robot to explore the environment based on the wall following exploration.

3. ***simplePerceptron_template.py:*** A template for the implementation of the perceptron model with 2 inputs and 1 output. This template includes an example of the initialisation of the weights, bias, input data and targets. You will use this file implement the training and testing of this your perceptron model.

4. ***blockColor_with_perceptron_template.py:*** A template for random generation of R,G,B data for recognition of red and green colours. You will use this file implement the training and testing of this your perceptron model.

5. ***blockColor_with_perceptron_with_CopperliaSim_template.py:*** A template for random generation of R,G,B data for recognition of red and green colours. This file also contains the lines of code for communication with CoppeliaSim. You might need to change the parameters such as IP or port number for the communication with your computer.

6. ***perceptronRobotControl.ttt:*** A CoppeliaSim scene to control the velocity of the Pioneer mobile robot when red and green colours are detected by the 3-input perceptron model.

Activity 1 – Fear, love and explorer robot

Save the *aggressionReactiveBehaviour.ttt* with the new names *fearRobotBehaviours.ttt, loveRobotBehaviours.ttt* and *explorerRobotBehaviours.ttt* to implement fear, love and explorer reactive behaviours based on the Braitenberg vehicles approach. You can have a look at the *aggressionReactiveBehaviour.ttt* scene to familiarise with the Lua syntax in CoppeliaSim.

Activity 2 – Wall following for environment exploration using reactive behaviour

Open the file *wallFollowingWithReactiveBehaviours.ttt* and copy the lines of code of the fear or explorer reactive behaviour to control the motors of the Pioneer mobile robot. If the implementation of the reactive behaviour is correct, then the robot will be able to follow the walls of the scene and explore the environment.

Activity 3 – Perceptron for recognition of AND, OR input patterns

Open the file *simplePerceptron_template.py* that has been provided to you in Moodle and implement a 2-input perceptron model to recognise the input pattern from 2-bits AND, OR, NOR logic gates. Implement the perceptron model using Python language.

Activity 4 – Perceptron for colour recognition and control of mobile robot

Open the file *blockColor_with_perceptron_template.py* and implement the perceptron model with 3 inputs (R,G,B data) for recognition of red and green colours. This file generates random R,G,B data that you can use to create the training and testing datasets. You need to train and test your perceptron model to recognise green and red colour objects.

Once your perceptron is trained do the following:

- Open the scene *perceptronRobotControl.ttt* in CoppeliaSim.
- Copy and paste the code for the training process from *blockColor_with_perceptron_template.py* into the template *blockColor_with_perceptron_with_CopperliaSim_template.py*, which contains the lines of code to communicate with CoppeliaSim and control the Pioneer mobile robot based on the recognition output from your perceptron model.
- Run the simulation in CoppeliSim, the robot should start moving.
- Run your file *blockColor_with_perceptron_with_CopperliaSim_template.py*.
- If everything is current, then the robot should move slowly when the green object is detected and fast when the red object is detected.

Files that you will need to submit from this activity:

- *fearRobotBehaviours.ttt*
- *loveRobotBehaviours.ttt*
- *explorerRobotBehaviours.ttt*
- *wallFollowingWithReactiveBehaviours.ttt*
- *simplePerceptron_template.py*
- *blockColor_with_perceptron_template.py*
- *blockColor_with_perceptron_with_CopperliaSim_template.py*

IMPORTANT
Add at the top of your files the author's name, clear comments to the code in your files and remember that it is important to indent your code to make it easier to read by others. All these aspects will be considered for your mark.

VM (or personal computer) configuration

Remote communication Python - CoppeliaSim
Make sure you have the following files in the same folder where you have your CoppeliaSim scenes:

- sim.py
- simConst.py
  The above files are in the following path:
  <Your-path-to-CoppeliaSim>/programming/remoteApiBindings/python

- remoteApi.so (linux), remoteApi.dll(windows) or remoteApi.dylib (macOS)
  The above file is in the following path:
  <Your-path-to-CoppeliaSim>/programming/remoteApiBindings/lib/lib

Conda environment
Depending on the configuration of your VM or personal computer you might need to create a conda environment with spyder and required libraries.

Create a conda environment with spyder

- `conda create -n spyder-env spyder=4`

Activate your conda environment

- `conda activate spyder-env`

You might need to install python libraries such as the following:

- `conda install numpy`
- `conda install matplotlib`
- `conda install -c conda-forge mlxtend`
- `conda install -c conda-forge tensorflow`
- `conda install seaborn`