Assignment #2

1. In the provided Matlab source "braitenberg_gradient.m" you are given a simulation of a Braitenberg vehicle (a blue circle) located in an arena with a single chemical source (a black square). The chemical source has a Gaussian strength gradient, i.e. the magnitude of chemical concentration decreases in a Gaussian manner as the distance from the source increases. The vehicle has two chemical sensors and two motors, each driving a wheel.

   Implement Braitenberg vehicle sensorimotor couplings that allow the vehicle to reach the chemical source. You may use any vehicle suited for this task but try to implement both *Love* and *Aggression* vehicles for comparison. Use perceptron models with sigmoid activation functions to convert left and right sensor inputs (*left_sensor_signal(ts)* and *right_sensor_signal(ts)*, where *ts* = timestep) into motor velocity outputs *v_l* and *v_r*. You may use the extra empty plots to display inputs and motor velocity outputs. You will need to define a few parameters for the sigmoid activation function, namely the y-axis range, the x-axis shift and the slope. Please think carefully about how these parameters affect output y. Use the following formula for the sigmoid activation functions for the left and right sensorimotor couplings –

$$y = \frac{C}{1 + A(e^{-Bx})}$$

   Where,

   A = horizontal shift
   B = slope
   C = upper limit of y-axis range (0 to C)
   x = sensor input
   y = motor velocity

   Add your code between the following two lines in the code:

```
%%%%%%%%%%%%%%%%%%%%%%%%% Add your code here %%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

   Explore the behaviour of the vehicle in the following manner –

   (a) Set the parameter *add_noise* to 0, which disables sensor noise. Find good values for A, B and C which allow you to get as straight trajectory as possible. Which parameter (A, B or C) do you expect to affect the straightness of the trajectory? Why? Confirm your hypothesis by varying A, B and C and observing the change (in any) in the trajectories.

(b) Set the parameter *add_noise* to 1, which enables sensor noise (default signal-to-noise or SNR value = 20 dB). Observe the trajectory for 5 separate trials using same values for A, B and C as in (a). What do you expect to happen in a noisy environment?

(c) Vary the parameter *snr_db* from 30 dB to 15 dB in steps of 3 dB. What do you expect to happen to the trajectories if the SNR increases or decreases?

(d) Implement a simple bit-bang controller of the form –

IF *left_sensor_signal(ts)* > *right_sensor_signal(ts)* THEN *v_l* = V1 and *v_r* = V2 (where V1<V2)
IF *left_sensor_signal(ts)* < *right_sensor_signal(ts)* THEN *v_l* = V1 and *v_r* = V2 (where V1>V2)
IF *left_sensor_signal(ts)* = *right_sensor_signal(ts)* THEN *v_l* = *v_r* = V

Choose hardcoded values for V, V1 and V2 from 0 to 3 m/s. Compare the observed robot trajectories to those you obtain when using Braitenberg controllers.

(e) Increase the distance between the left and right sensors (given by *sensor_width*, default value = 16 cm) and re-run your Braitenberg as well as bit-bang controllers <u>without</u> changing any controller parameters. What do you expect to happen to the trajectories? What do you observe in the trajectories?

2. Download the cockroach simulator from Itslearning or from Github at the following location - https://github.com/alpeq/SourceTaxisSimulator. Investigate how moving the sensors affects the trajectory of the simulated cockroach. To do this, you have three parameters for the antennae which you can vary –

Frequency: This determines how fast the antennae oscillate
Mid-point: This determines the mid-point (in radians) around which the antennae oscillate
Amplitude: This determines the how wide towards the sides the antennae oscillate

Change the above parameters (one at a time) and observe the robot's behaviour for each of the three different Braitenberg models implemented in the simulator –

(a) "Braitenberg" (does not use stimulus dynamics)
(b) "Crossed dynamic" (contralateral couplings with stimulus dynamics)
(c) "Dynamic" (ipsilateral couplings with stimulus dynamics)