

# Braitenberg vehicles

## Introduction

The title of Valentino Braitenberg's book "Vehicles: Experiments in Synthetic Psychology" betrays the nature of the subject material here; how simple robots can display behaviour which can be described in terms of human emotions such as *fear*, *aggression*, *love*. He formulated his vehicles as two-wheeled differential-drive robots with a small number of sensors. His vehicles 2a and 2b (the most often cited) move away and move towards light; light is the stimulus most commonly used. You can see the sensors and the motors in the diagram below. Also shown are the symbolic connexions between sensors and motors. Consider vehicle 2a; the left eye is connected to the left motor and the right eye to the right motor. The connexions are shown as + (excitatory) so when the left eye gets more light, then the left motor speeds up. How does 2a move? Well since the left eye is closer to the light then the right eye, the left eye will receive more light. Both motors are driven but the left motor gets more drive. So the robot will move forward but will veer to the right, avoiding the light.

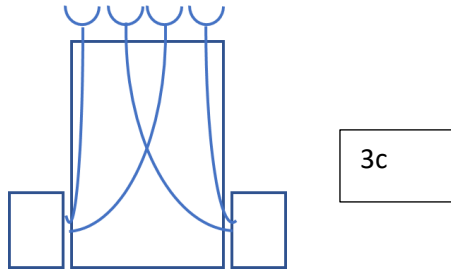
In psychological terms, 2a is a *coward* running away from the stimulus, while 2b is an *aggressor*, it runs towards the light and will collide with a high speed.



Vehicles 3a and 3b are shown below. One is attracted to the light, the other repels from the light; one is a *lover* and the other is an *explorer*. The negative signs show *inhibitory* connexions, a larger sensor signal will make the connected motor run slower. Of course, there must be some 'baseline' drive, else the robot would move backwards. Moving backwards was not mentioned by Braitenberg (according to my reading).



Braitenberg suggests a third variant of vehicle 3 which has multiple sensors, and these are of different types, e.g., sound or smell. One is shown in the diagram below on the left.

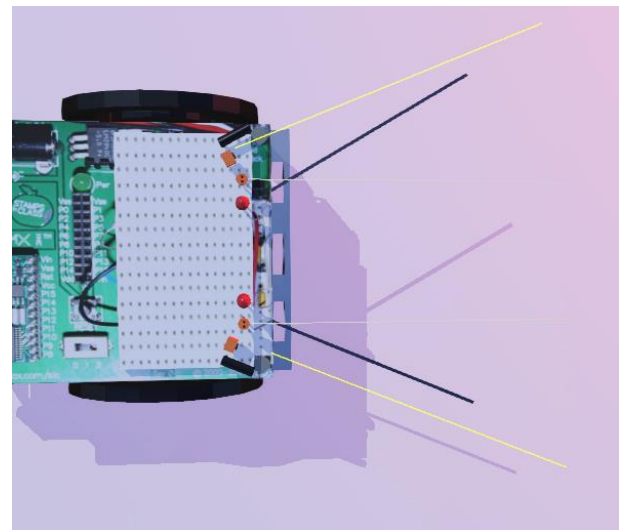
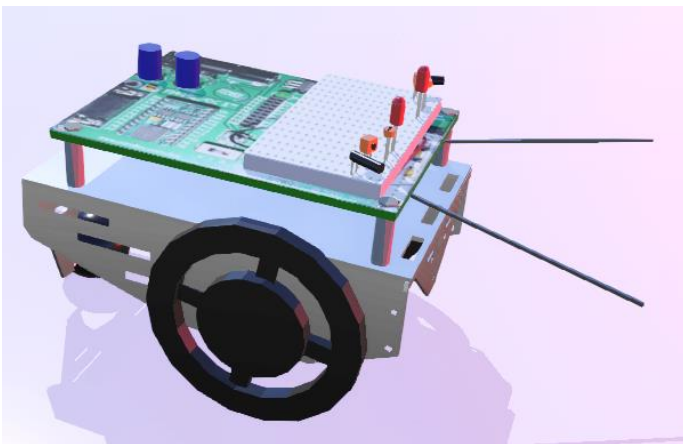


Here there are 4 sensors two connected to each motor; the connexion type (excitatory / inhibitory) is not specified here.

So far, the connexions have been linear; the motor speed is proportional to the sensor value passed to it, either increasing or decreasing. Braitenberg suggest other connexion functions which are *non-linear* such as the one shown above on the right. Here, for low values of input  $I$  the motor speed  $v$  increases, but when  $I$  gets larger than some threshold value, the motor speed decreases.

## Building a Braitenberg Vehicle

Here we shall use the BoeBot robot shown below on the left. You can see two 'whisker' collision sensors at the front. The diagram on the right shows two light sensors already added to the BoeBot ready for your use. We'll see how to add more sensors later.



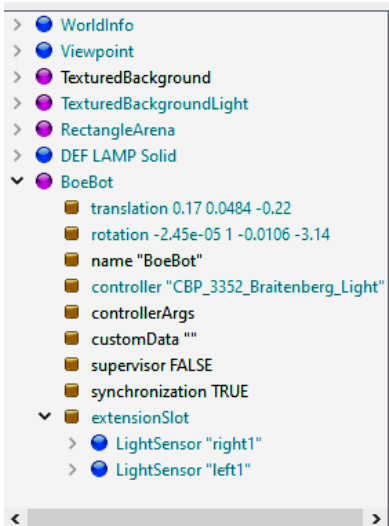
You can show the light sensors by selecting View > Optional Rendering > Show Light Sensor Rays in Webots. These sensors have names "left1" and "right1" which you use when you get the device in the code. The world you need to use is **CBP\_3352\_Braitenberg\_Light\_1.wbt**, and make sure the controller **CBP\_3352\_Braitenberg\_Light** is selected and also opened in the text editor.

Coding a vehicle is straightforward. The template code gives you the two eye intensity values, these are in the range 0 to 4095

```
leftIntens = invertLightSensorVals(leftEyeVal);
rightIntens = invertLightSensorVals(rightEyeVal);
```

All you need to do is to decide how to use **leftIntens** and **rightIntens** to provide values of **driveL** and **driveR** to drive the motors

```
omegaL = driveR*MAX_SPEED;
omegaR = driveL*MAX_SPEED;
```

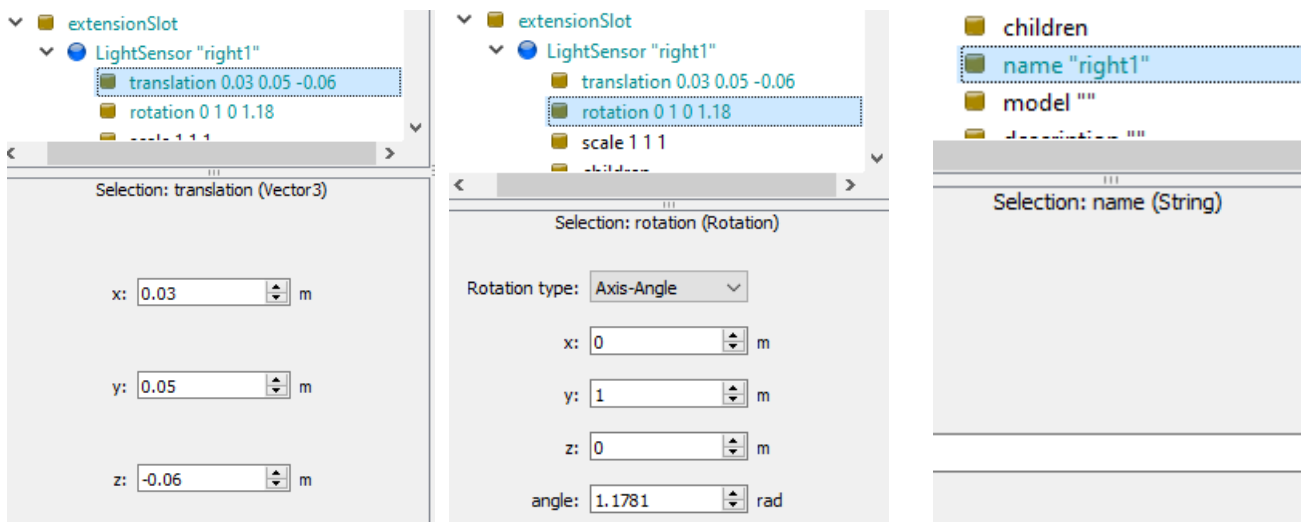


Here is the scene tree for this world where the BoeBot node is expanded to show you the two LightSensors in the extension slot.

You can also see the lamp node, DEF LAMP Solid.

It is easy to place additional lamps into the world, left click on the lamp node to select it, then right click and copy and paste. Then move the lamp in the world using its gizmos.

Adding another sensor is similar, just copy and paste an existing LightSensor into the extensionSlot. It will appear at the centre of the robot. So you must change its location and angle as follows. Also you will have to assign a name. Here's the values for the existing right1 sensor. Change x and z but not y (the sensor height) using the up-down arrows and the same for the angle. Make sure the sensor ray is being rendered.



You'll have to add code for the additional light sensors. It's fairly obvious where you need to copy and paste from the original code.

## Investigation Suggestions

- 1) Compare and contrast vehicles 2a and 2b with 3a and 3b.
- 2) Add a Lamp to the BoeBot, and place 2 in the world, perhaps removing the fixed Lamp. Investigate how the robots behave when simulating vehicles 2a, 2b, 3a, 3b.
- 3) Add an increasing number of these Boebots with lamps and look for any emerging behaviour
- 4) For a single BoeBot and Lamp, code some non-linear relationship between intensity and motor drive.
- 5) Add two more sensors. Make two of the four respond to red light and the other pair to cyan light. Put red and cyan lights into the world. Try to get the BoeBot moving to a red light while avoiding cyan lights.