Constructive Artificial Intelligence Practical Session 10

by Lola Cañamero and Matthew Lewis
m.lewis4@herts.ac.uk

This document gives hints about using Webots in response to questions asked by students. The suggestions here might be useful for coursework CW2 (depending on how you tackle the problems).

1: Pushing objects with the e-puck

To implement a garbage collection robot, you will need to make the e-puck robot push items around the environment.

An object that you might use is the drink can. This is available in Webots 7 under Proto (Webots) → objects → indoor → drinks.

If the can is too heavy, then the e-puck will not be able to push it. You can reduce the mass of the can by changing the object's properties in the scene view panel on the left of the Webots interface. For example, a can with mass 0.01kg can be pushed by the e-puck.

If you choose to use other objects for your "garbage" items, then you may need to add a **Physics** node in order to be able to set their mass. For example, the cardboard box item needs a Physics node added if it is going to be pushed (and you may want to reduce its size too). To add a Physics node to the cardboard box object, highlight the Physics slot of the box (in the scene tree in the left panel), click the add icon (plus sign), and choose the Physics node in the popup window.

If you want to make the robot centre itself to the can for a more effective push, you can think of it as similar to line following. In line following, the robot uses its ground sensors to detect a line, and steers towards the line if it is slightly to one side of the line. To centre the robot to a can, you can use the distance (proximity) sensors to detect the can, and steer the robot towards the can if it is to one side.

Detecting light in Webots

There are two types of light source available in Webots: DirectionalLight, and PointLight. The default Webots 7 environment starts with a directional light, but you can replace it with a point light.

Light can be detected with the e-puck's light sensors. These are the same sensor devices as the infra-red distance sensors, but used in a passive mode (so they are not sending out infra-red light,

just measuring the amount of infra-red light that arrives at the sensor). A value of 0 means “lots of light” and a value of >4000 means “almost no light”.

Note: You **can** use both the light and distance sensors on the e-puck at the same time.

To initialise the light sensors, get them with names “ls0”… “ls7”, and call the `enable()` function when you set them up (similar to the distance sensors – see the example code for object avoidance from practical 7, particularly the initialisation and `read_sensors()` functions).

The light sensors in Webots are very “directional”, that is, they detect direct light very well, but not light coming from the side, or ambient light (light in the background) or reflected light. For example, if the robot is directly *underneath* a point light, then the values returned by the sensors will be >4000 (“almost no light”) because the light sensors are directed out to the side of the robot, and not up.

If you move the point light down to the level of the robot, then you may find that it is difficult to get the sensors to detect how distant the light is, since the sensors will return 0 (“lots of light”) even when the point light is very far away. To change this (for example, if you want to know how far away the light is) you can change the **attenuation** of the light source. For a point light that is level with the robot, a value of (x,y,z) = (0,50,0) will mean that the value read by the sensors will go down slowly as the robot approaches the light. Unfortunately, with this setting it will be difficult to see what is happening in the simulator, because the view will be very dark – you may want to turn on wireframe rendering (under View in the menu) to see where the objects are.