MANCHESTER 1824
The University of Manchester

Robot programming lab

PART 1: Choregraphe

Implement the following exercises using Choregraphe, Pepper's graphical development tool and simulator. With the assistance of a TA, you can run your program on one of the real robots.

For the list of the available functionalities of Choregraphe, visit:
http://doc.aldebaran.com/2-5/software/choregraphe/index.html

1) Simple shake

   Using the timeline functionality, make Pepper shake its head once. Move the head to its left, then to its right and finally return it in a central position. Each step must last 2 seconds, for a total animation duration of about 6 seconds.

2) Shake and nod

   Implement two different timeline boxes: one will nod the head up and down and the other one will shake it left and right. Both timelines will have a duration of about 6 seconds each (2 seconds for each direction and 2 seconds to return neutral). Connect these boxes in any desired order and execute them twice in a loop (hint: check the Counter box).

3) Walk back and forth

   Design a behavior that will make Pepper walk forward for 50 cm, stop and then walk backwards for 50 cm to the starting position.

4) Walk and turn back

   Design a behavior that will make Pepper walk forward for 50 cm, stop, rotate 180° in place and walk forward for 50 cm back to the starting position.

PART 2: Python

These exercises involve the use of Python programming language (we use version 2.7 for compatibility with the robot software suite). With the assistance of a TA, you can run your code on one of the real robots.

You can find the API documentation at http://doc.aldebaran.com/2-5/naoqi/index.html.

1) Walk back and forth

   Design a behavior that will make Pepper walk forward for 50 cm, stop and then walk backwards for 50 cm to the starting position.

2) Walk and turn back

   Design a behavior that will make Pepper walk forward for 50 cm, stop, rotate 180° in place and walk forward for 50 cm back to the starting position.

3) 2D navigation

Given the following discrete grid map, where each cell has a length of $l = 0.5\ m$:

(0,3)	(1,3)	(2,3)	(3,3)
(0,2)	(1,2)	(2,2)	(3,2)
(0,1)	(1,1)	(2,1)	(3,1)
(0,0)	(1,0)	(2,0)	(3,0)

Consider the robot to be placed on the center of one of these cells, facing north (upwards) and able to walk freely in every direction. Write a Python script to make Pepper navigate from one specified cell to another. The function will accept as inputs the coordinates of the starting $\{S_x, S_y\}$ and ending $\{E_x, E_y\}$ positions, will calculate a trajectory (x, y, θ) for the robot and execute the motion. Verify the results in the simulator.

Hint: have a look at the atan2 function.