UNSW COMP3431: Robot Architectures S2 2010

Assignment #2

Localisation and Kicking

Due: Start of Lab, Week 13 (Noon, 22 October 2010)

1 Overview

This goal of this assignment is to give you broad familiarity with localisation and manipulation in robots. In particular, in this assignment you will write behaviours to move to a particular location on the field, and to move a robocup ball around the field.

Students will perform this assignment in small groups of up to 4. You are requested to work closely together on the project, developing the code together (rather than using the "divide and conquer" technique of splitting the project into parts, developing the parts separately, and then recombining). Please work in different groups to the groups from Assignment 1. If you have a problem forming groups, please talk to the lecturer in charge as soon as possible.

1.1 Use of others' software

As in assignment 1, you already have access to the source code of a complete soccer playing robot, including behaviours to complete most of this assignment. For this assignment you are free to use any base-station code, any C++ code on the robot (accessed through the VisionLink module) and the basic python behaviour files on the robot listed here:

```
PyLib/*
PyCode/Action.py
PyCode/Behaviour.py
PyCode/Constant.py
PyCode/Debug.py
PyCode/Global.py
PyCode/hFrameReset.py
PyCode/hMath.py
PyCode/hPauseMotion.py
PyCode/hPWalk.py
PyCode/hWalk.py
PyCode/hPSensor.py
PyCode/Indicator.py
PyCode/Packet.py
```

```
PyCode/pDoNothing.py
PyCode/pInitial.py
PyCode/pReady.py
PyCode/pSet.py
```

Note that parts of other files may be called by those files, and in particular Global.frameReset() calls HelpTrack.determineBallSource(). You should not add further references to other files, and you should look at the above code path to check that it does what you want if you use the global ball location variables.

You are *encouraged* to read the other python files, but please, no direct copying. You are also encouraged to build upon your code from assignment 1. You are encouraged to look at other RoboCup teams' approaches (please acknowledge any borrowed ideas in your report), although again, no direct copying of code.

1.2 Other Requirements

All behaviours should work both with and without other robots on the field, and with the robot wearing either colour uniform.

1.3 Deliverables

Part 2 of this assignment requires you to develop behaviours on the robot. The compiled code should have been tested on the robot before the class starts.

During the lab each team will make memory sticks from their working copies. The solutions will be compared and graded. Each of the behaviours should have its own python module usable by the `spip` script. The names of those python modules should be included in the report.

The report in part 3 should be an electronic file emailed to the lecturer in charge. It should be in an easily readable format, *e.g.* plain text, html, PDF or postscript.

2 Time Trial (10 marks - graded in lab)

In this part of the assignment, the ball will be placed down the yellow end of the field and the robot must get the ball to the blue side of the centre line as fast as possible.

Initially the robot will start at the opposite end of the field from the ball. The timer will start when the robot crosses the centre-line, and stop when the ball crosses the centre-line. If the ball is kicked out then it will be returned to the initial position.

3 Report (5 marks)

Describe the approach used in each of the previous section. Describe the strengths and weaknesses of that approach. The more scientific you are here, the better. *e.g.* it

is nice to say that technique A works better than technique B, but it is better to say that technique A averages 10 seconds, and technique B averages 15 seconds and so technique A is better than technique B. It would be even better still to say that technique A has a mean of 10 seconds with a standard deviation of 2 seconds, and technique B has a mean of 11 seconds with a standard deviation of 0.5 seconds.

Moreover, you could look at different parts of different solutions separately. e.g. If you choose to 'grab' the ball under the robot and then kick that ball, you could analyse the choice of grab separately from the choice of kick.

I don't want you to waste your time 'scientifically' checking things just for the report, but if you try something 'for real', then test it and record some results. How many times you test it will probably depend upon how well it is working: if the idea seems hopeless on the first few trials, then don't bother testing it a whole lot. You should still report the results you have, even if the data is "The first run was over 30 seconds, and I stopped it", and so the mean is "awful".