Introduction to Vision and Robotics 2015/16

Second Assessed Practical: Robot Control[1]

1 Introduction

The aim of this practical is to build and to control a robot such that it interacts with its environment. The tasks are described in the next section below.

You will work in pairs of students and must submit one joint report, i.e. only one student of each pair needs to submit the report. The report must contain a brief explanation of how the work was shared and how you think the marks should be distributed (e.g. equally or in some other proportion if you and your partner did not contribute equally to the work). You can choose the partner yourself and you can choose a different partner than in the first IVR assignment. Please contact michael.herrmann@ed.ac.uk immediately after this assignment is given out, if you do not have a partner.

The assignment is estimated to take about 40 hours of work in total, i.e. your share will be about 20 hours. This time includes writing the report, but excludes revision of any material from the lectures or tutorials. Try to distribute the work evenly and efficiently between you and you partner and in such a way that you can contribute what you are good at.

The work will be done using the EV3 Lego Mindstorms robot. You will need to have access to such a robot if you want to work successfully on this assignment. If this is not the case, please contact your tutor.

2 The Tasks

There are two tasks which are to be solved using a robot with two independently controllable wheels. Keep the design of the robot as simple as possible in the beginning and try to improve it when necessary. You will be to present the tasks with the note "DEMO" in a life demonstration. In the report, all task are to be described.

A. PID control

i. Construct a two-wheeled balancing robot using the EV3 kit. You can follow the construction plan for the Gyro Boy (see http://robotsquare.com/2014/07/01/tutorial-ev3-self-balancing-robot/) or other existing plans, or mix existing designs with your own ideas. In any case, you may like to leave out unnecessary components, but first check below, what parts may be unnecessary. The mark for this task will also consider design decisions in other questions of this assignment.

ii. Set up a PID controller that keeps the robot upright, when started in an upright position. The controlled variable for the controller will be a weighted sum of wheel (or rather motor) position, wheel speed, angle and angular velocity. The information about the angle can be obtained from the gyro sensor which gives information about tilt of the robot. The controller would thus receive as an input

$$x = g_{motor} * (p - p_0) + g_{motor\_vel} * v_{motor} + g_{angle} * (a - a_0) + g_{angular\_vel} * v_{angle} \quad (1)$$

The controller then uses the value, the derivative and the integral of x in order to determine the control signal. As in question (i) you can use existing control

1 To be given out on 03/03/2016

programs and/or known values for the g parameters in (1) and for the three controller parameters k_p, k_i and k_d, if you cite your sources. In the report, explain in detail, how the controller works and how it reacts to disturbances and any other behaviour of the robot.

iii. Vary the parameters of the controller such that robot is barely able to stay upright, but moves back and forth instead.

iv. Set the gain for the integral term to zero, i.e. k_i=0. Possibly re-adapting other parameters and the design of the robot and the controller, is it still possible to keep the robot balancing?

v. Set the gain for the derivative term to zero, i.e. k_d=0. Possibly re-adapting other parameters and the design of the robot and the controller, is it still possible to keep the robot balancing?

B: Navigation

Choose **one** of the following tasks. In each case the robot will need to be able to move forward and to turn while keeping its balance. The robot may also need additional sensors or motors.

- Homing: After travelling along a rectangular path the robot returns to the starting position and stops there. Use external sensing in order to achieve a good precision in this task
- Line following: Add a down-facing colour sensor and show that your robot is able to robot to follow a line on the floor. Try to keep the movement of the robot smooth and fast.
- Obstacle avoidance: The robot is shown to be able to move away from a wall and to circumvent a non-moving obstacle (such as a box of a EV3 kit).

You will be asked to present the result at the demonstration, but also your report should include evidence at what level of precision your robot actually solved the task.

Hints

- Revise control theory! The main ideas of control theory and some reading material were given in the lecture.
- If you see several options in solving a task, congratulations! You have understood a very crucial point in robotics. For the assignment, you may not be able to realise all of the options, so try to choose the one that seems to be the most interesting among the realisable possibilities.
- Avoid perfectionism. If your demonstration does not work at first attempt, you will be given a second chance a few minutes later.
- The tasks can be carried out in Matlab or in a different programming language. Note that the communication between the robot and Matlab sometimes is a source of problems. It is preferred if the control programs runs on the EV3 brick.
- It may happen that some basic function of the robot is not working. The robot may not respond or not interact with Matlab etc. Please refer to the guidelines of the supervised practicals for help. Next check the course page for up-to-date information on the issue. If the problem is still not solved, then contact your tutor.
- In order to solve tasks (e.g. moving, orienting, obstacle avoidance) consider using use concurrent processes on the EV3. Include a graphical representation of the

program structure in your report.

- You should approach the tasks in several stages. Take a sensible approach to the system design: THINK about what components you need; PLAN the tasks to be done, how they will be done, by whom and in what order; IMPLEMENT, and TEST. Your final mark will be based on how well you explain your approach to the task and evaluate the capability of your program. If your control algorithm is not reliable, you won't fail the assignment if you can provide a sensible explanation of what you attempted and the limitations and problems in your report.
- If you have problems in question A, you can still solve question B by using a castor wheel which should make balancing easy.

3 Writing the report

The report should be a concise description of what you did, why, and what happened. The report should comprise between 1000 and 2000 words. These limits are not strict, but you should be able to stay in this range by moving details into an appendix. Appendices will not be marked, but will be considered for clarification. Program listings are not required since you are asked to submit your code separately. The report should contain the following sections:

a) Introduction: Include explanation of any concepts that were not covered in the lectures and an overview of your approach.

b) Methods: Describe how you built your robot and give a functional outline of your code. Explain how each part of it is meant to work. Where suitable, justify your decisions, e.g. why you used one method rather than another, what you tried that did not work as expected, etc.

c) Results: Provide some actual quantitative data, from repeated trials on how well your algorithm controls the robot. This can include data on tests that you have performed in preparation of the task, trajectories of the robot(s) in the actual task, comparison between different strategies. Represent the data graphically. Well documented failure will get more marks than unsupported claims of success (well-documented success will get even more marks!).

d) Discussion: Assess the success of your program, and explain any limitations, problems or improvements you would make if there was more time.

4 Live demonstration

You will have to demonstrate your program working in the EV3 robot. The demonstration session for this second practical will take place in the Forrest Hill Robot Lab on

Friday 25/03/2016 from 9:30 am onwards.

If you prefer, you can demonstrate your work already on Thursday 24/032016. In this case, contact the lecturer one week before.

5 Submission

Your report should be a single PDF file and our code a separate zipfile. Both should be submitted electronically by 4pm on Thursday 24/03/2016. The command to use for on-line submission from your DICE account is:

submit ivr 2 FILENAME CODE

where FILENAME is the name of your file, e.g. report_s1234567.pdf and CODE is the name of the archive containing your code, e.g. code_s1234567.tgz. The code is not subject to marking, but will be referred to in case of ambiguities in the report. The assignment will be marked as follows:

	Marks (to a maximum of 100, worth 20% of coursemark)
Demo: Construction and hardware design in parts A & B	10
Demo: PID control (Aii)	10
Demo: PID control (Aiii)	10
Demo: PID control (Aiv + Av)	10
Demo: Navigation (B)	20
Report: Problem analysis	10
Report: Program design	10
Report: Results and evaluation	10
Report: Clarity (including figures, diagrams etc.)	10

6 Demonstrators

Davide Modolo (d.modolo@sms.ed.ac.uk), IrisKyranou (iriskyr@gmail.com) and Calum Imrie (s1120916@sms.ed.ac.uk) are the demonstrators for IVR. They will be in the lab at the agreed demonstration sessions, see course page. You can also e-mail them when course equipment or software is not working, but not if your software has bugs (that's your problem!). For any other questions about this practical please contact Michael Herrmann (michael.herrmann@ed.ac.uk).

7 Plagiarism

This assignment is expected to be in your own words and code. Quotations with proper, explicit attribution are allowed., but it should also become clear what is your own work. Use proper citation style for all citations, whether traditional paper resources or web-based materials. Before submitting please acknowledge any additional sources of code that you use and ensure that your submission follows the school policy on plagiarism: http://www.inf.ed.ac.uk/teaching/plagiarism.html.