Hands on Introduction to Control

Introduction

In this set of exercises, we will be using the DETI robot to demonstrate some effects related to the control of a servo system using a PID controller.

The exercises are based on:

- the standard platform available for compiling and loading programs for the DETI robot;
- rmi-PID, a C module that implements the PID controller and a "hack" of the mr32 library, compatible with the PID module.
- gnuplot, for generating graphs.

A high level program, `rmi-PIDexample.c`, contains the code to run the demos. This program generates a square-wave set-point for the motors speed, with values +30 and -30. This will make the robot turn around is axis in alternate directions, allowing you to verify the system's step response. To avoid having the robots moving over the table, test the system with the wheels standing off the ground.

A `makefile` is supplied to compile and run the programs.

Getting the software

All the software is available as a BitBucket project. To get a copy of this project, issue the command:

```
git clone https://bitbucket.org/pnfonseca/rmi-pid-demo.git
```

You will see a folder rmi-pid-demo, containing all the code.

The compilation chain is the one you have used for Lab Project #1. You may need to install gnuplot, which usually is distributed as a package.

Compiling

To compile and load the program, run make as follows:

```
$ make -B KP=1 KD=0 KI=0 run
```

This will create a program with PID parameters tuned to K_p=1, K_d=0 and K_i=0. These are real-valued variables, so you can set K_d=0.1 with

```
$ make -B KP=1 KD=0.1 KI=0 run
```

The program will be loaded to the robot and will start execution on pressing the START button (black button). This will allow you to verify the robot system's step response

Creating graphs

After pressing the START button, the robot program will output to the pterm terminal the command values and the motors response in 4 columns: Right motor set-point, Right motor speed, Left motor set-point and Left motor speed.

```
# RMI PID example
#
# PID params: Kp=5.000000, Ki=0.000000,
Kd=0.100000
#
# Press start to continue
30    0     -30   0
30    13    -30   -2
30    7     -30   -3
```

Important note: if you notice that the motors are oscillating heavily, turn the robot immediately off! Sudden changes of direction in electric motors may be accompanied by high current peaks, which can damage both the motors and the controlling electronics.

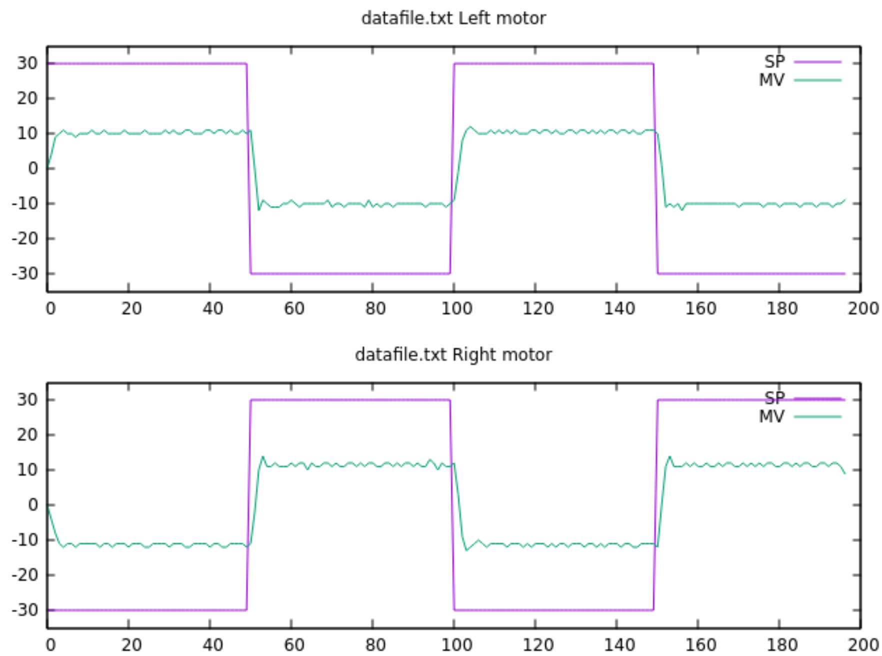
Allow the robot to perform 3 or 4 changes of direction and stop the program, turning the main switch off. (You can also exercise your programming skills and have the robot automatically stopping after a few changes of direction.)

The values written to the terminal (starting from the lines that begin with '#') should be stored in a text file, that will be used to generate the graphs with the system step response. The simplest way to do this is to select the values in the terminal window, copy (Shift-Ctrl-C), paste in a text editor and save the file.

You will find a script called `PIDgraphs.gp` to create the graphics in Gnuplot. You can run this script by issuing the following command (assuming that you have stored the robot program output in a file called `datafile.txt`)

```
$ gnuplot -e "filename='datafile.txt'" PIDgraphs.gp
```

This should generate a graph like the following, where "SP" corresponds to the set-point and "MV" to the motor speed value.



Using what you have learned about PID tuning, try to find the best set of values for the PID controller of the DETI robot.

The performance of a control system, and hence, of the controller, is highly dependent on the system dynamics. In this case, the load applied to the wheels is one of the main factors affecting the system's dynamics. After tuning the PID parameters with the wheels standing off the ground, repeat the experiments with the robot now standing on the ground (and with the wheels having now to turn the robot...)