Dyson School of Design Engineering, Imperial College London

DE2 Electronics 2

Open-ended Challenges

Peter Cheung, 8 March 2024

Introduction

Labs 1 to 6 were designed to teaching you something. In these Labs, you were more or less told what to do, and often how to do them. Now you should be in a position to approach solving practical problems in signals, systems and control without handholding.

This document is NOT a lab, but it contains various suggestions on possible challenges that you might take on yourself. They are arranged in order of increasing levels of difficulties (Level 1 to 4) and progress from one to the next. Since each pair of students share one set of equipment, the Lab Oral in the final week **will be done in pairs**, but you also be asked to declare your personal contributions to achieving the challenges.

You are of course also encouraged to create your own challenge or modify any of the suggestions shown here. Remember, you have other deadlines. Please do not get sucked into this piece of work at the expense of your health or your achievements in other subjects.

Finally, you must return your Lab-in-a-Bag kit including the Segway, the Pybench board, the USB cable and the neopixel strip before you leave for your Easter break. I will only return your marks for the final week Lab Oral after you have returned your kit.

Challenge 1: Dancing LED lights (Level 1)

This is the easiest to achieve. In Lab 6, you already worked on the skeleton beat detection using Pybench and blinking the blue LED. You can combine beat detection with loudness to light up more LEDs, synchronised to the beat, when the volume of music is louder. You can also improve the beat detection algorithm as you see fit. This challenge was already specified in Lab 6 instruction sheet as "option" for Lab 6!

Challenge 2: Control speed and direction of motors using IMU (Level 2)

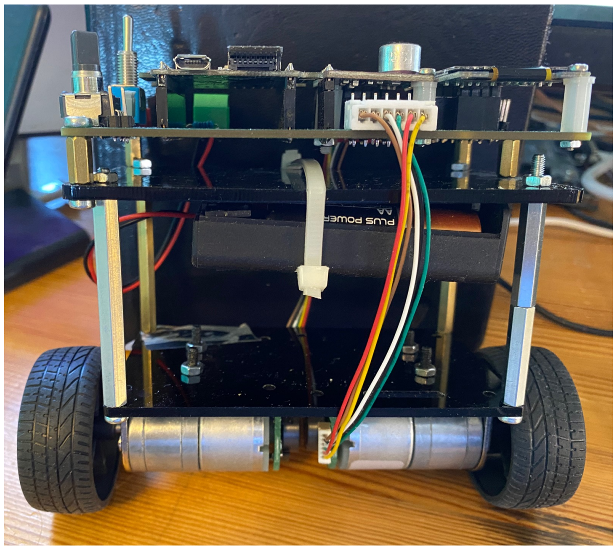
In Lab 5, you used the potentiometer to control the speed and direction of both motors, while in Lab 4, you learned to how to derive the pitch and roll angles from IMU measurements. This task is to combine what you have learned in these two labs so that you can control the speed and direction by tilting the PyBench board, one motor controlled with pitch angle and the other with roll angle. You should display the speed of each motor (revolution per second) on the OLED display alongside the angles.

Challenge 3: PID controller to control speed of the motors (Level 3)

This challenge is an extension to the previous challenge. In this challenge, you should implement the PID control method to make sure that the speed of both motors are the same, and they are determined by the pitch angle of the Pybench. You should demonstrate that the PID control algorithm is working by attaching the wheels and apply pressure to increase friction. The PID controller should maintain the speed with different friction.

allenge 4: Segway dancing to music WITH stabilizers (Level 3)

For this Challenge, you need to do a number of preparatory work.



Step 1: Assemble the Segway by attaching motor chassis to the Pybench using the standoffs and the base plastic plate. (See photo.) For this, you would need to use screw drivers to access various holes. You will also need to rotate one of the motor so that the sockets are on opposite sides of the Segway to maintain symmetry. Make sure that the motor cable reaches both sockets.

Step 2: Attach the 9V batteries to the Segway. You will need to use the Segway untethered to your laptop, with your user's program stored on the SD card. The battery box can be attached either to the underside of Pybench or above the motor base plastic plate. If you do not intend to do the self-balancing challenge, your best choice is to mount the battery as low as possible. However, if you are to attempt self-balancing, it is easier to balance if the centre of gravity is higher up. For this reason, you may want to attach the battery as high as possible.

Step 3: Assemble two stabilizers during Lab 7. These are to be attached to the edge of the base plastic plate so that the Segway will not fall over. A sample is available for you to copy how the stablizers are attached to the Segway.

Once you have the above preparatory work done, you are in a position to solve this challenge. You should create a set of dance moves for your chosen music, and encode the dance routine as a set of ASCII characters and stored in a text file on your computer. For example, FFBB could mean move forward two steps on two beats, followed by backward two steps on the next two beats. LRLRLRLR could mean step left than right and repeat again three more times (like a crab crawl). You can then transfer the dance move text file to the SD card. In your mPy program, you can open and read from this text file into an array of character, and then do the dance synchronised to the beat of the music. The Segway will not fall over because of the stabilizers.

Challenge 5: Self-balancing the Segway (Level 4)

This challenge requires you to implement a PID algorithm using the pitch angle as the set-pont variable and control the motors to maintain the pitch angle to be zero (verticle). You should attempt this challenge initially with stablizers (much like when you learned to ride a bike), and remove them after self-balancing is achieved. Download the Lab 7 instruction, which contains quite a few tips and pseudo-code segment to help you towards the goal.