IC382

Final report

Cheng Yat Hin

20106076D

N02

Contents

I. Introduction

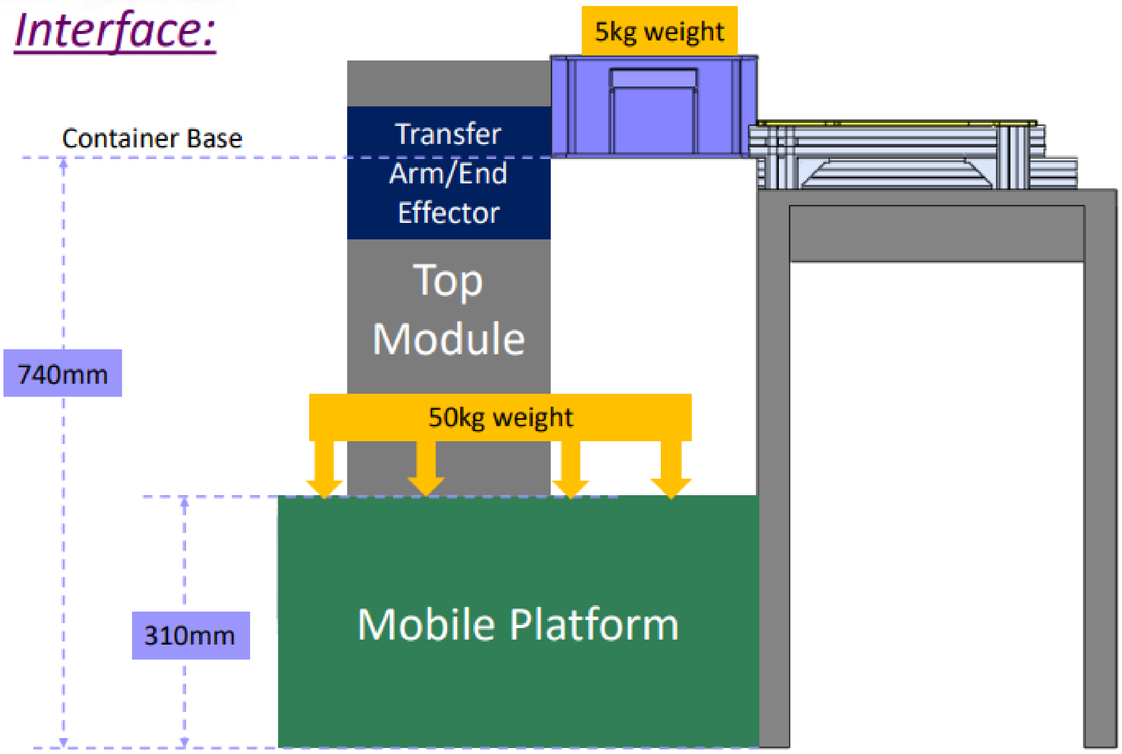
In this IC382 multi-disciplinary project, me and my groupmate are tasked to create a moving platform.

Our goal is to design and build a platform so it could perform two major tasks: First, the platform should be able to move a meter forward, turning 90 degree clockwise or counterclockwise, then moving one meter forward again without human controlling, meaning it could automate its pathing movement. Secondly, the top module needs to be able to load and unload a 5kg load.

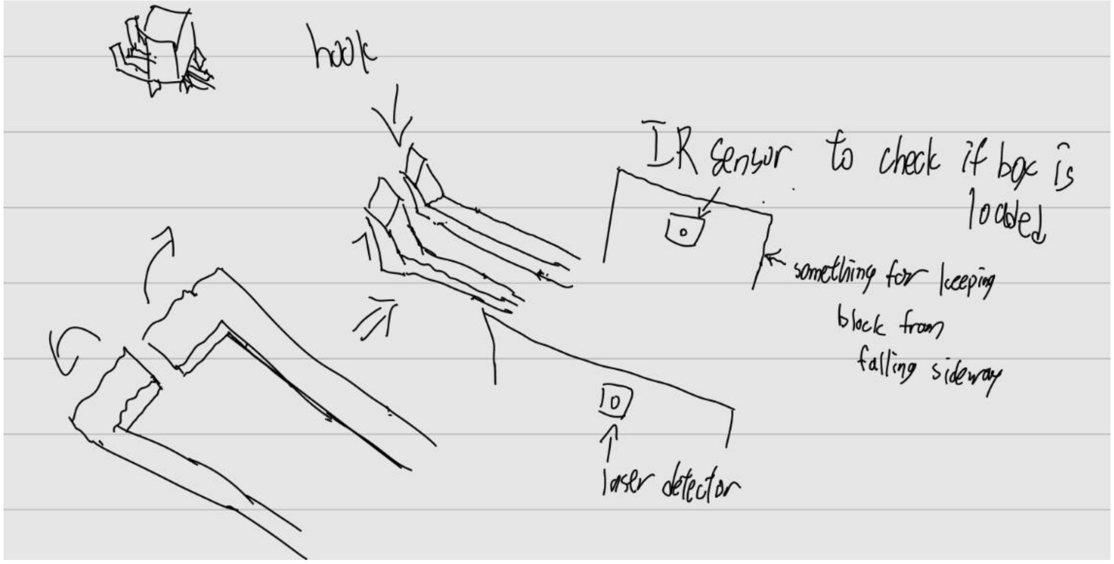
As an EIE student, my work is mainly focused on electronic-related design and implementations. This individual report will focus on my work during the second semester, therefore there will only be EIE related work and what I had done.

II. Background

To build this platform, the top module container base has to be 740mm high and the mobile itself should be able to withstand 50kg weight.



For the bottom module, we have to program the raspberry pi microcontroller to communicate with the DM422 Step motor drive, embedded computer, and encoder. And for the top part, we need to program the Arduino uno to communicate with a linear actuator, two light sensors and two servo motor. Since the bottom module's assembly is already finished. Only the assembly and the wiring of the top module will be needed. For the top module, the linear actuator will move the container base below the box. The servo motor will act as a claw or gate to trap the box and bring it onto the platform. Finally, the light sensor to determine when the linear actuator should stop when going forward/backward. For the bottom module, the encoder can be used in the wheel to track its movement and calculate the distance traveled.



III. Methodology

Bottom module

In this semester, the bottom module part mainly consists of the programming only, namely the encoder_to_odom file which controls the movement of the platform with the help of encoder's callback to determine the odometry of the platform and mission control part where it automates the process what the platform would perform according to the reading.

For the encoder_to_odom.cpp file, I helped with the debugging of the platform, me and my groupmate recalculated the dimensions and the value of each rotation for the encoder. Then tested the performance of the file. Initially the encoder value was not zero at the beginning, I suggested fixing it by adding an if-statement on the encodercallback() function as the value of the variable does not correspond to the encoder value initially, which caused the problem. After that was fixed, we have some more tests on the encodercallback() function such as changing the equation's method of resetting the encoder reading or the comparison statement into different values and that worked out.

For the mission control, I performed the simulation on the gazebo environment with the controller. Allowing the car to move forward, turn left and right, going backward, finally going to a coordinate automatically. We then copied the code to our platform's controller.py and tested it with putty by opening both controller.py and encoder-to-odometry.cpp. But during the testing in real life environment, we encountered multiple errors, in which it involved inertia, direction of castor wheel and the communication rate between two files. After I debugged the problem of communication rate problem and problem of the inertia with a dynamic speed setting. The platform could move as intended.

Top module

As for the top module, my work is mainly involved in the wiring of the top module and some part of the programming.

For wiring, I was tasked to gather the necessary wires for Arduino board's connection involving the servo motor, linear actuator, and light sensors. Some wire had a short length, so I needed to extend the wire using a hot air gun and plastic tube.

For programming, I helped my groupmate Donald and Jacob to debug the code on the top module Arduino Code. After a bug was discovered, the controller.py loaded the x axis value as 0 for a few second after the platform moves. I suggested increasing the publishing rate and the subscriber's rate so both encoder to odom.cpp and controller.py could communicate in sync. Then I suggested a multi-stage approach on turning 90 degree and moving one meter forward which solves the problem of castor wheel in wrong direction, causing misdirection of the platform.

Teamwork

Our team worked phenomenal throughout the project. Even though there are challenges such as the MCU board pin malfunctioning during demonstration. Our ME groupmate were very supportive and tried whatever they could to assist the EIE groupmate and we tried our best to let them understand what happened and what we could do to compensate with this situation. We also tried our best to communicate using English as ME groupmate were mostly unable to communicate with Cantonese, we still tried our best to communicate using English as much as we could.

In the context of EIE part, we had a clear division of labor throughout the semester, and everyone was doing what they could to finish this project. Everyone was utilizing what they specialized in and contributed to the project.