Introduction to Mechatronics (ME/AE 6705)
Final Project Description

Design and Development of a Line Following Robot using a PID Controller

1 Objective

The purpose of the project assignment is to combine the concepts studied in the lab assignments into a system that solves a complex task. The main objective of the project is to deign and implement a feedback controller for a wheeled mobile robot that will follow a black line on a white background. Infrared (IR) sensors will be used to detect the position of a black line under the robot. Several MSP432 peripherals and functionalities such as Timers, PWM signals, and GPIOs will be used in the project.

Furthermore, a virtual project demonstration and racing competition will be organized for the line following robots built by students, at the time of the final exam scheduled for the course (Monday, May 3, 11:20am - 2:10pm).

Note 1: This project description contains some suggested approaches, but feel free (and it is recommended) to innovate in your algorithmic and design approaches.

2 Problem Statement

Build a line following robot that autonomously follows a black line on a white surface as quickly as possible using IR sensors. In addition to tracking accuracy and speed, your robot will be evaluated on the smoothness of operation.

2.1 Mandatory Requirements

- The robot should use a PID controller for tracking the line. A finite-state machine approach or an open-loop control design will not receive full credit.
- Using the line tracking error from the IR sensor array, design a PID controller that modifies the applied PWM signal to each individual motor and reduces the tracking error. A block diagram for such system is shown in Figure 1.
- The PID controller must have a control update rate of at least 50 Hz.

2.2 Optional Recommendations

- A cascaded PID controller can be used in the design, so that a second PID control is implemented to control the speed of the motors using the PWM duty cycles. This means that we can implement two PID controllers in series. A controller that calculates desired motors' speeds to reduce the tracking error, and a second controller to modify the applied PWM signals to achieve the desired speeds.

- The DC motors' encoders can be used to measure the actual speed of the robot wheels.

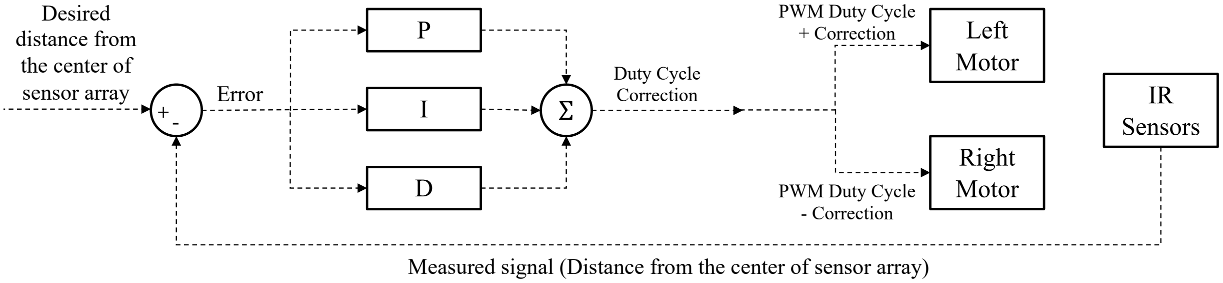
Desired distance from the center of sensor array
+
-
Error
P
I
D
Σ
Duty Cycle Correction
PWM Duty Cycle + Correction
PWM Duty Cycle - Correction
Left Motor
Right Motor
IR Sensors
Measured signal (Distance from the center of sensor array)

Figure 1: The system block diagram using a PID controller to track the line with the IR sensors.

3 Required Parts

The main components required for this lab are included in the Mechatronics kit provided for the course. These components are listed in Table 1.

Part	Quantity
MSP432 LaunchPad	1
Romi Chassis Base Plate	1
Romi Chassis Motor Clip	2
Pololu Wheel 70×8mm	2
Romi Chassis Ball Caster Kit	1
Romi Battery Contact Set	1
DC Motor 6V (w/ encoder)	2
8-Channel QTRX Sensor Array for Romi	1
Aluminum Standoff: 1-1/2" Length, 2-56 Thread, M-F	4
Machine Hex Nut: #2-56	4
Machine Screw: #2-56, 1/4" Length, Phillips	4
H-Bridge Motor Driver IC (SN754410)	1
Breadboard	1
Jumper Wire	–
Wire Kit	–
Diode 1N5819	8
10kΩ Resistor	4
Test Lead Gator to TIP Plug 6"	2
AA Alkaline Batteries	4
Multimeter	1

Table 1: Required parts for the Line Following robot.

In addition to the components listed in Table 1, a black electrical tape, and six letter size white papers (or a 17" × 33" foam board) are needed to make the track for testing the robot. A small piece of double-sided tape could be used to attach the IR sensor array to the underneath of the Romi chassis.

4 Hardware Setup

Information on how to assemble the Romi chassis with the DC motors and wheels can be found at: https://www.pololu.com/docs/0J68/4.

Note 2: We will only use 4 AA batteries to power the robot, which are installed in the main battery compartment of the chassis. Therefore, there is no need to install the battery contact terminals in the small battery compartment (there will be 4 battery contacts that won't be used).

Note 3: To facilitate connecting jumper wires to the motor terminals and encoder pins, it is recommended to create a small opening on the chassis under the encoder pin, so that the jumper wires could pass through the chassis and get attached to the pins. This can be done by using a knife or a screwdriver. The recommended position of the openings is highlighted in Figure 2 in Appendix A.

The MSP432 LaunchPad and the breadboard can be installed on top side of the chassis using the aluminium standoffs, machine screws and hex nuts included in the kit. As an alternative option, the breadboard can be secured to the chassis using its adhesive backing.

The power from batteries can be brought to the breadboard by connecting the test lead gator clips to the battery contact terminals, while connecting the male jumpers to the breadboard.

The QTRX IR sensor array can be attached under the chassis using a double sided tape, such that the male headers pass through the openings of the chassis and are accessible from top. The recommended position for the sensor array is shown in Figure 3 in Appendix B.

Note 4: When developing the program and driving the robot under the debug session, we can power up the MSP432 LaunchPad normally through the USB debug port (the motors would be powered through the batteries, similar to Lab 8). When testing the robot in the testing track, you will have one of two choices to power up the MSP432:

1. Use a USB extension cable to connect the MSP432 to your computer and power it up that way. Alternatively, you can follow the robot while holding your laptop near it.
2. Use a 5V voltage regulator to power up the MSP432 from the battery supply from your circuit (for example L7805: https://www.sparkfun.com/products/107). In that case, you should:
   a. On the LaunchPad, remove all the jumpers of the J101 header except GND, 3V3 and 5V.
   b. Attach the voltage regulator 5V output to a 5V pin on the LaunchPad (any 5V pin would work). Do not forget a common GND between the regulator and the LaunchPad.

Warning: If the LaunchPad is powered up with the batteries, the debug cable must not be connected to the board, or the LaunchPad may get damaged.

5 Software Overview

The main program of the robot must be implemented in C/C++ on the MSP432 LaunchPad. The DC motor speed control must be conducted using PWM signals and an H-bridge motor driver circuit implemented on the breadboard (similar to Lab 8).

The QTRX sensor array provides 8 outputs (one per sensor), which can be measured with digital GPIOs. More information about the sensor can be found at:
https://www.pololu.com/product/3672.

The simplified typical sequence for reading a sensor is:

– Set the I/O line to an output and drive it high.
– Wait for at least 10 μs for the sensor output to rise.
– Make the I/O line an input.
– Wait for exactly 1 ms (may need tuning).
– Check the sensor output, a LOW level indicates high reflectance, and a HIGH level indicates low reflectance on the sensor.

This process can be done in parallel for all the 8 sensors.

The exact timing requirements can be implemented using the MSP432 timers.

As mentioned in Section 2.1, a PID feedback controller must be implemented to control the robot according to the sensor outputs. A finite-state machine or an open-loop system will not receive full credit. The fixed control update rate can be achieved by generating periodic interrupts using the MSP432 timers.

6 Testing Track

To test the robot performance and participate in the robot competition, construct a robot testing track using black electrical tape and six letter size white papers as shown in Figure 4 in Appendix C.

The line width is 0.75” (width of a regular black electrical tape). The background color must be white. You can tape paper sheets together to create the testing track. Alternatively, you can obtain white foam boards from arts supply stores (e.g. Blick or Michaels) to have a strong and rigid surface.

Note 5: The robot must be able to traverse the track in *both* directions.

7 Final Demonstration and Competition

A virtual demonstration/competition will be held at the time of the final exam scheduled for the course on Bluejeans. Each student has 3 minutes to demonstrate the project by starting the robot from the start point. The robot performance will be evaluated on three criteria:

- Having a full PID controller implemented.
- The capability to accurately and smoothly track the black line and return to the start point (in both directions) without manual assistance.
- The total time it takes to traverse the track (average of both directions).

Each student is allowed to test her or his robot at most two times for each direction. Only the best performance will be considered.

Extra credit will be assigned to the top 3 winners.

8 Grading

The final code for the project must be uploaded as a zip folder to Canvas by the time of the final exam (Pass/Fail).

- Full credit (200 points) will be received if:
  - The robot makes it around the testing track in both directions without manual assistance
  - A full PID controller is implemented properly for the robot
- 50 points will be deducted if the robot works only in one direction.
- 100 points will be deducted in case of not implementing a PID controller.

In case of robot failure, or not implementing a full PID controller, partial credit will be given for any relevant, correct work.

Appendix A

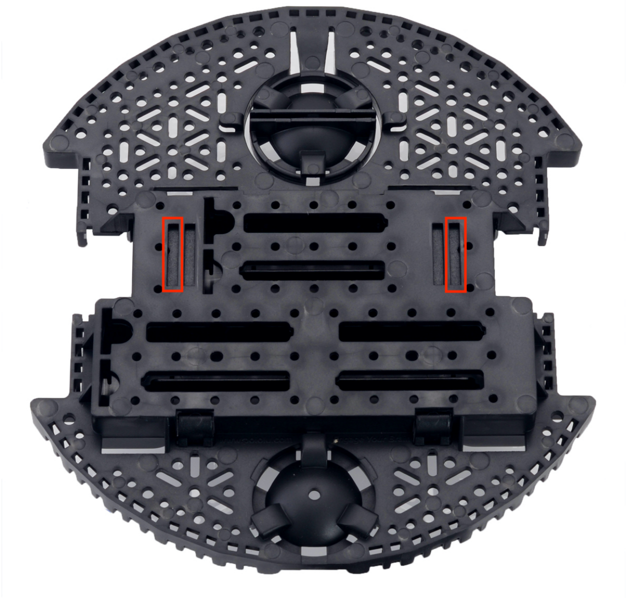

Figure 2: Recommended locations on the Romi chassis to create openings for passing the motors' jumper wires.

Appendix B

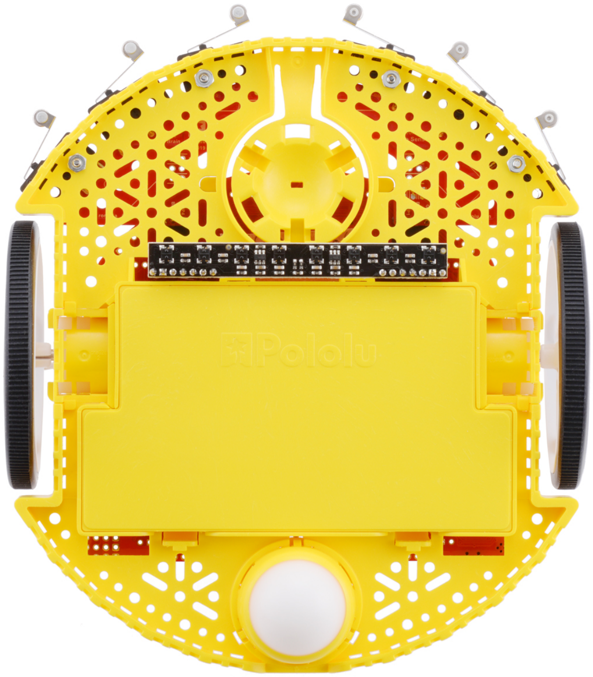

Figure 3: Recommended location to attach the QTRX sensor array.

Appendix C

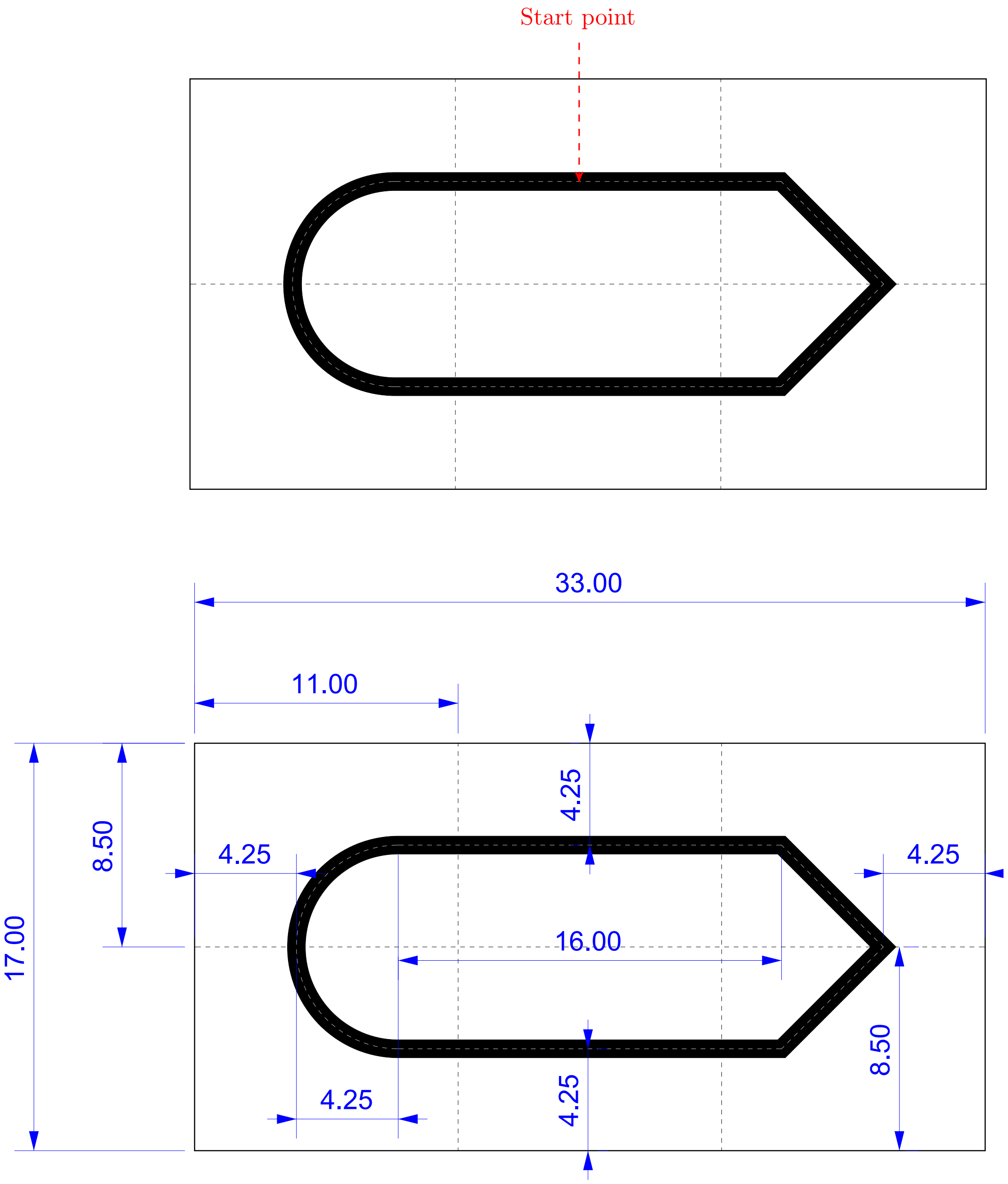

Figure 4: The testing track pattern and its dimensions for the line following robot. All dimensions are in inches. Six letter size papers can be used to create the track.