IR Sensor Calibration and PI Controller Design

Autonomous Robot Navigation (E160) Lab 1 Report

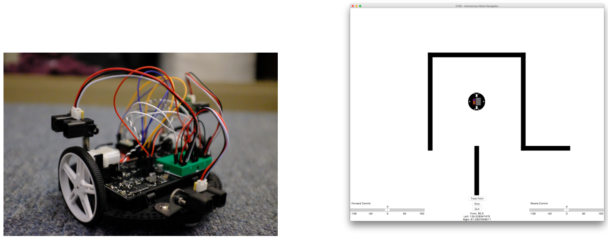
Eyassu Shimelis
Harvey Mudd College
eshimelis@hmc.edu

***Abstract*—This report documents the calibration of the infrared distance sensors; and the design of a closed loop controller for the E160 robot. The robot is a two-pound, differential-drive vehicle with a total of three infrared distance sensors attached to the front and sides, respectively. Within the range of operation ($20~\mathrm{cm}$ - $150~\mathrm{cm}$), each infrared sensor outputs a voltage inversely proportional to the measured distance. The robot's front sensor was calibrated from $20~\mathrm{cm}$ to $100~\mathrm{cm}$ (at $5~\mathrm{cm}$ increments). The results were analyzed in MATLAB, and a nonlinear inverse function was fit to the data. The resulting coefficient of determination for this fit, within the calibrated range, is $R^2_{inverse} = 0.99848$. This fit was compared to, and shown to be better than, two other fit types: a logarithmic fit and a second-order polynomial fit. After calibrating the sensors, a closed-loop, proportional-integral controller was developed to maintain a distance of $30~\mathrm{cm}$ between the robot and a wall. The robot can successfully reverse from or drive towards a moving wall in order to maintain the desired distance.**

I. INTRODUCTION

A. Robot Chassis and Sensors

The base of the E160 Bot is a differential-drive robot kit from Pololu (Romi). The kit includes a round, plastic chassis (Pololu #3510); a motor driver and power distribution board (Pololu #3543); two gear motors; and two Hall effect wheel encoders. The completed robot is shown in Figure 1(a) below.



(a)

(b)

Fig. 1: (a) Assembled E160 robot. (b) The E160 graphical user interface (GUI)

B. Wireless Communication

All of the state estimation and control calculations are computed on a separate computer. The robot is equipped with an short-range, XBee wireless module. These modules allow us to connect a computer to the E160 robot. The computer wirelessly transmits simple motor commands to the onboard Arduino Mini; which, in turn, is programmed to send the corresponding motor signals to the motor driver.

C. Software

On the computer-side of the system, the E160 robot is controlled by a custom GUI, written in Python. The provided architecture utilizes an object-oriented approach with the following main classes: *environment* and *robot*. This architecture is useful because it allows us to separately simulate either the environment or robot, making it easier to test.

In addition to these two main classes, there is additional code that handles the graphical user interface (GUI). The GUI, shown in Figure 1(b), visualizes both the *environment* and *robot*, also allowing us to use sliders to manually control the robot.

II. SENSOR CALIBRATION

The Sharp GP2Y0A02YK0F infrared distance sensor has an operational range between $20~\mathrm{cm}$ and $150~\mathrm{cm}$. Within this range, the sensor outputs a voltage inversely proportional to the measured distance, as shown in Figure 2 below.

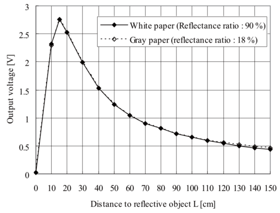


Fig. 2: Sharp GP2Y0A02YK0F infrared distance sensor, voltage calibration curve provided in product data sheet.

A. Setup

To calibrate the front infrared sensor, the robot was placed between $20~\mathrm{cm}$ to $100~\mathrm{cm}$ away from a wall, at $5 \pm 0.3~\mathrm{cm}$ increments. In total, distance measurements were collected at 17 points. At each point, between 117 and 194 measurements were collected and averaged.

B. Results

The collected data were processed in MATLAB to extract the mean and standard deviation of the measurements. After processing, Equations 1-3 were fit to the data and their resulting coefficients of determination are summarized in Table I.

Thanks to proctors John Lee and Zayra Lobo for their guidance during construction.

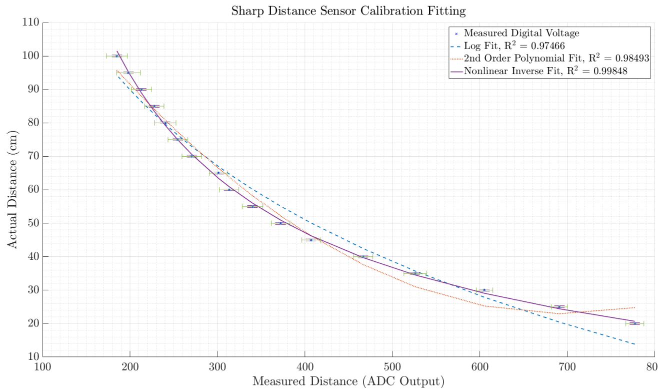


Fig. 3: Comparing measured voltage and actual distance, we see that the relationship is nonlinear. Three functions were fit to the data and plotted above: a logarithmic fit (Eq. 1), a second order polynomial fit (Eq. 2), and an inverse fit (Eq. 3).

$$y = c_0 + c_1 \log(x) \tag{1}$$

$$y = c_0 + c_1 x + c_2 x^2 \tag{2}$$

$$y = c_0 + c_1 x^{-c_2} \tag{3}$$

TABLE I: Coefficient of Determination Comparison

Fit Type	R^2 **Value**	Fit Coefficients $\{c_0, ..., c_n\}$
Logarithmic	0.97466	$\{386.8, -56.03\}$
2nd Order Polynomial	0.98493	$\{157.972, -0.388, 2.786e^{-4}\}$
Nonlinear Inverse Fit	0.99848	$\{-17.582, 7402.737, 0.791\}$

III. DESIGN OF CLOSED LOOP CONTROLLER

A. *Problem Definition*

After the front infrared sensor was calibrated, a closed loop controller was designed and implemented to actively maintain a desired distance between the robot and a wall. The goal of a closed loop controller is to eliminate the error between the actual and desired state of a system, see Eq. 4. The error, e, is defined as the difference between the current state, x, and the desired state, x_{des}.

$$e = x - x_{des} \tag{4}$$

The first controller was a simple proportional controller. This type of controller applies a control effort proportional to the error, see Eq. 5.

$$u_P = K_p e \tag{5}$$

Unfortunately, as the error approaches zero, the control effort weakens, therefore producing a steady state error. Figure 4, shows two experiments conducted with a proportional control gain of $K_p = 2$. The top two plots show a step response, both away from and towards the wall. Both effectively move the robot towards the desired distance; however, there is a significant steady state error of approximately 4 cm. To eliminate this steady state error, we can include an integral control gain, which applies control effort proportional to the sum of previous error, see Eq. 6. This type of closed loop controller effectively eliminates any steady state error. For the E160 robot, the integrated error is simply the past ten measurement errors, sampled at an approximate of rate of 10Hz. The results of this controller are illustrated in Figure 5 for two integral gains. The higher integral gain experiment results in significant oscillation, an undesirable effect.

$$u_{PI} = K_p e + K_I \int e \; dt \tag{6}$$

B. *Results*

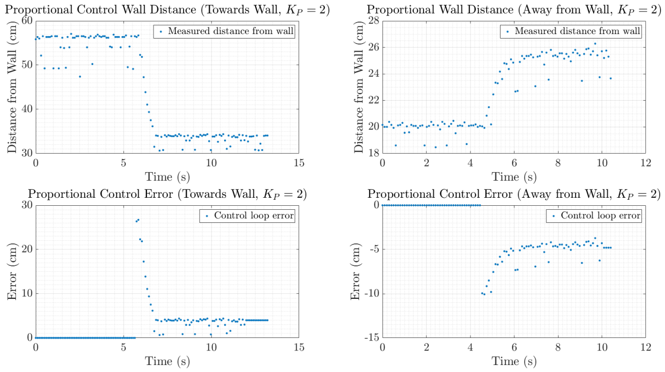


Fig. 4: P Control Results- TL: Step response towards wall. BL: Corresponding control loop error of plot above. TR: Step response away from wall. BR: Corresponding control loop error of plot above.

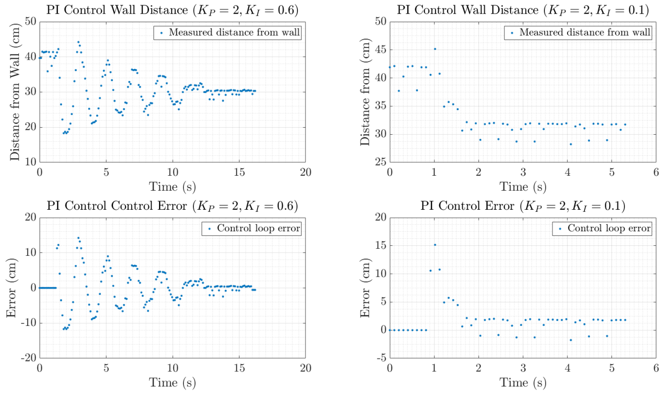


Fig. 5: PI Control Results - TL: Step response with high integral gain, significant oscillation. BL: Corresponding error to plot above. TR: Step response with reduced integral gain. BR: Corresponding error to plot above.

IV. CONCLUSION

Although the calibration model fits very well, it does not capture the nonlinearities for sensor measurements between $0 - 15$ cm. Additionally, the closed loop controller may perform well with averaged data to smooth out extraneous measurements. Although the robot exhibits nonlinear behavior, the PI controller does an excellent job compensating for these effects.

REFERENCES

[1] Sharp GP2Y0A02YK0F Analog Distance Sensor $20 - 150$ cm