

ARTIFICIAL NEURAL NETWORK-BASED ROBOTICS

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Justin Ng

June 2018

© 2018
Justin Ng
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Artificial Neural Network-Based Robotics

AUTHOR: Justin Ng

DATE SUBMITTED: June 2018

COMMITTEE CHAIR: Andrew Danowitz, Ph.D.
Assistant Professor of Electrical Engineering

COMMITTEE MEMBER: Xiao-Hua Yu, Ph.D.
Professor of Electrical Engineering

COMMITTEE MEMBER: Fred W. DePiero, Ph.D.
Professor of Electrical Engineering

ABSTRACT

Artificial Neural Network-Based Robotics

Justin Ng

Artificial neural networks (ANNs) are highly-capable alternatives to traditional problem solving schemes due to their ability to solve non-linear systems with a non-algorithmic approach. The applications of ANNs range from process control to pattern recognition and, with increasing importance, robotics. This paper demonstrates continuous control of a robot using an actor-critic algorithm based on deep deterministic policy gradients (DDPG) originally conceived by Google DeepMind. The robot performs tasks such as locomotion within an enclosed area and object transportation. The paper also details the robot design process and explores the challenges of implementation in a real-time system.

ACKNOWLEDGMENTS

Thanks to:

- Everyone for everything.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
1 Electrical Design	1
1.1 Introduction	1
1.2 Power	1
1.3 Sensors	2
1.3.1 Adafruit 9-DOF IMU	2
1.3.2 VL53L0X Rangefinders	5
1.4 Motor Drivers	9
1.5 Servo	10
1.6 Microcontroller	12
1.7 Interconnect PCB	13
1.7.1 Schematic Capture	13
1.7.2 Board Layout	13
1.7.3 Assembly	13
1.7.4 Reworks	13
BIBLIOGRAPHY	14
APPENDICES	

LIST OF TABLES

Table	Page
1.1 Motor Control Truth Table	10
1.2 Servo Required Pulse Widths	11

LIST OF FIGURES

Figure	Page
1.1 DC-DC Buck Regulator [3]	2
1.2 Adafruit 9-DOF IMU Mounted	3
1.3 FreeIMU GUI [6]	4
1.4 VL53L0X Rangefinder Mounted	6
1.5 VL53L0X Measured vs. Actual Distance	7
1.6 VL53L0X Squared Error for Uncalibrated vs. Calibrated	8
1.7 VL53L0X Standard Deviation	9
1.8 L298N Motor Driver Wiring Diagram [9]	10
1.9 Servo PWM Control Scheme [8]	11
1.10 STM32 Nucleo-64 Development Board [14]	12

Chapter 1

ELECTRICAL DESIGN

1.1 Introduction

The electronics of the robot use a combination of off-the-shelf parts and custom designed circuits.

1.2 Power

A four cell, 1800 mAH lithium polymer (LiPo) battery powers the entire system. The battery is connected using polarized XT60 connectors to prevent reverse connection. The battery voltage varies between 16.8 V when fully charged and 14.8 V when depleted so two off-the-shelf DC-DC switching regulators, shown in Figure 1.1, buck battery voltage down to 12 V and 7 V supplies. The switching regulators accept a 7 – 40 V supply and can output 1.2 – 35 V at 8 A each. The 12 V bus powers the three motor drivers boards while the 7 V bus powers the STM32 Nucleo-64 development board and two AZ1085CD low-dropout linear regulators (LDO). One LDO produces a 5 V bus while the other provides 3.3 V, each at 3 A.

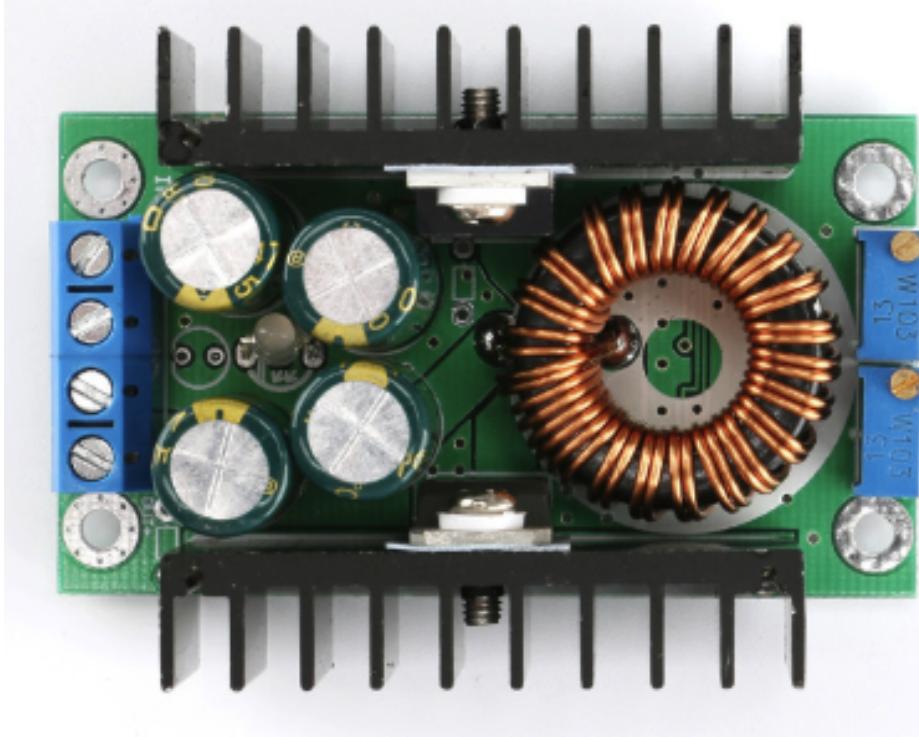


Figure 1.1: DC-DC Buck Regulator [3]

1.3 Sensors

The robot uses five off-the-shelf sensors for determining its position: four VL53L0X 1-D LIDAR rangefinders and one Adafruit 9-DOF inertial measurement unit (IMU).

1.3.1 Adafruit 9-DOF IMU

The Adafruit 9-DOF IMU incorporates the L3DG20H gyroscope and LSM303DLHC accelerometer/compass combo on a single carrier board to allow full inertial measurement in a convenient form factor [2]. Figure 1.2 shows the IMU mounted in the 3D printed bracket. The robot only utilizes the accelerometer and magnetic compass to realize a tilt-compensated compass. The LSM303DLHC can measure both acceleration and magnetic fields in three dimensions with configurable bandwidth and

full-scale ranges. It uses 400 kHz I²C for control and data transfer and draws power from the 3.3 V supply. Raw measurements from the IMU possess significant offset and scaling error so a calibration routine is required.

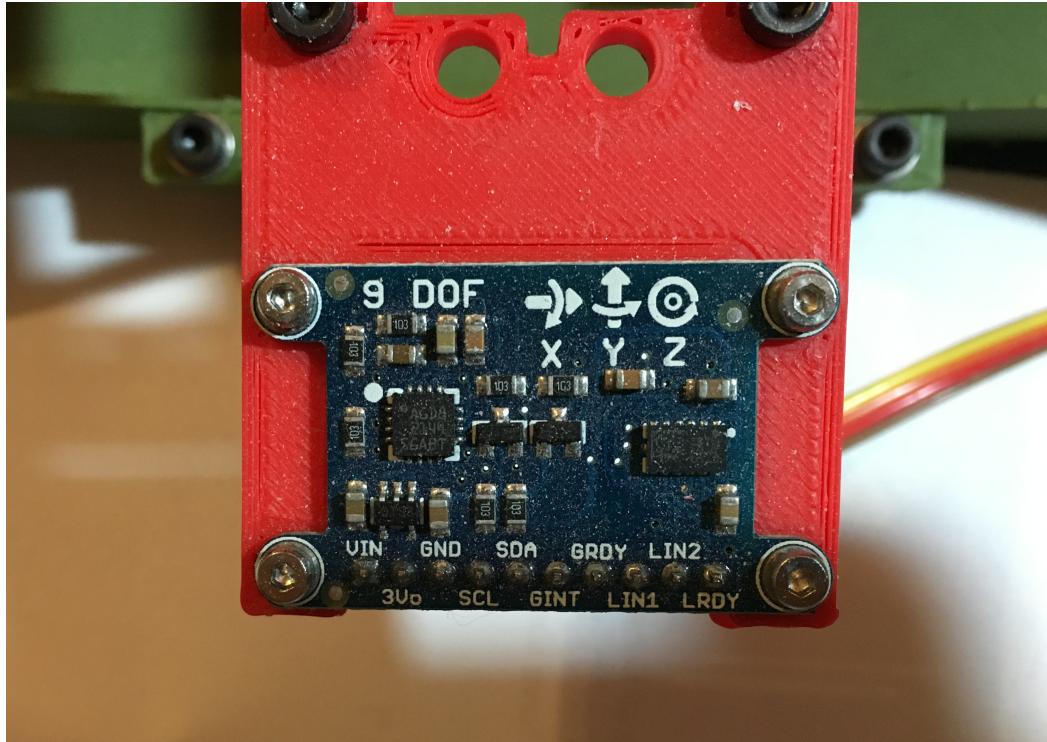
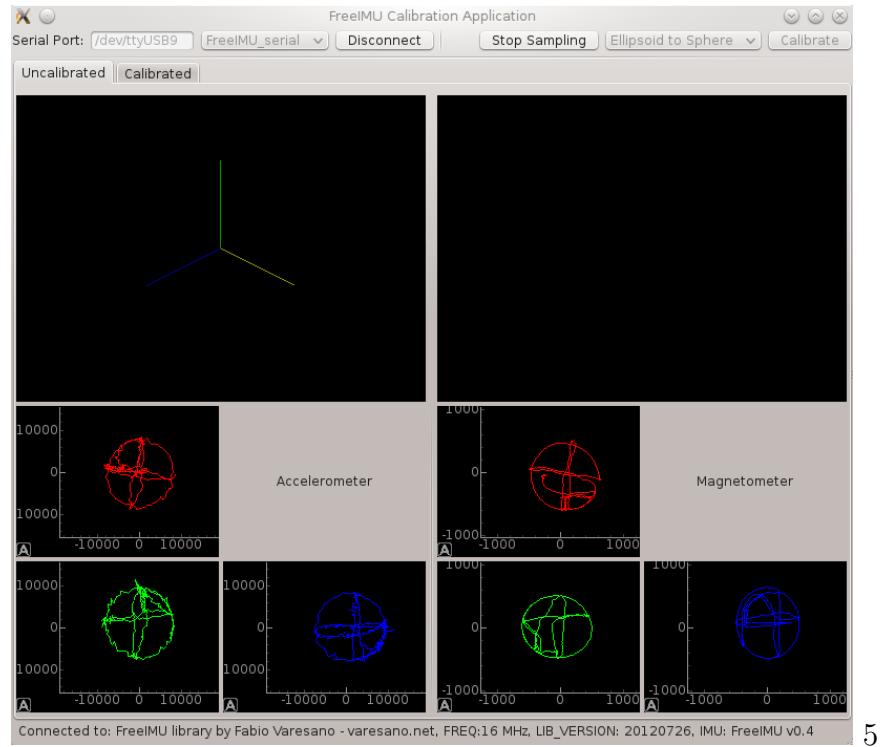


Figure 1.2: Adafruit 9-DOF IMU Mounted

The accelerometer and magnetometer calibration process utilizes the FreeIMU program written by Fabio Varesano in Python [6]. The program features a graphical user interface (GUI), shown in Figure 1.3, to display accelerometer and magnetometer measurements in real time and plots them in 3D space. The calibration program was originally designed to calibrate the open-source FreeIMU IMU when connected to an Arduino with the FreeIMU calibration firmware so the program's device communication back-end was modified to accept the LSM303DLHC IMU connected to an STM32 microcontroller with custom firmware. The calibration algorithm assumes that the sensor's measurements are linearly distorted and therefore produces a linear scaling factor and offset for each of six measurements (accelerometer X, Y, Z

and magnetometer X, Y, Z). The correction algorithm is shown below where val can represent any of the six measurements.

$$val_{calibrated} = m_{scale}val_{measurement} + b_{offset} \quad (1.1)$$



5

Figure 1.3: FreeIMU GUI [6]

The calibration procedure is as follows:

1. IMU should be mounted on fully assembled robot and connected to microcontroller.
2. Connect microcontroller serial port to computer through Serial-to-USB converter.
3. Click "Begin Sampling" to start recording magnetometer and accelerometer measurements.
4. Point the IMU x-axis at the ground and rotate 360° around that axis. Repeat for y-axis and z-axis.

5. Point the IMU x-axis at the sky and rotate 360° around that axis. Repeat for y-axis and z-axis.
6. Repeat Steps 3 and 4 at least twice to increase data size.
7. Click "Stop Sampling".
8. Click "Calibrate" to calculate scaling and offset constants.

The procedure ensures that ends of each axis eventually receive the maximum acceleration and magnetic field. To ensure hard-iron errors (such as from motors and permanent magnets) as well as soft iron errors (from local ferromagnetic materials like steel) are compensated for in the calibration, the process should be performed with the fully assembled robot and redone each time the robot is modified [11]. Since the expected fields are known (acceleration due to gravity, strength of Earth's magnetic field), the required linear transformation can be calculated. See [6] for details of the exact algorithm.

1.3.2 VL53L0X Rangefinders

The rangefinders, marketed by STMicroelectronics as the "world's smallest Time-of-Flight ranging sensor", are capable of measuring between 30 and 2000 mm with a 30 Hz sample rate [16]. It operates by firing pulsed light from a vertical cavity surface emitting laser (VCSEL), measuring time taken for the laser pulse to reflect back to the sensor, and calculating the distance based on the known speed of light. The robot uses cheap \$8 VL53L0X breakout boards in lieu of designing and assembling custom carriers; the sensor itself comes in a 4.4 x 2.4 mm lead-less package making hand soldering prohibitively difficult. Figure 1.4 shows the sensor board mounted in the 3D printed bracket. The sensor breakout boards include the VL53L0X module, decoupling capacitors, an LDO for the sensor's 2.8 V power supply, and level shifters for the I²C lines. Control and data transfer both occur over 400 kHz I²C so the breakout only requires four wires: 3.3 V, ground, and the I²C data and clock lines.

To characterize the accuracy and precision of the sensor, distance measurements were taken by targeting the sensor at a sheet of standard white printer paper. The data is compared with measurements from a tape measure; results are shown in Figure 1.5.

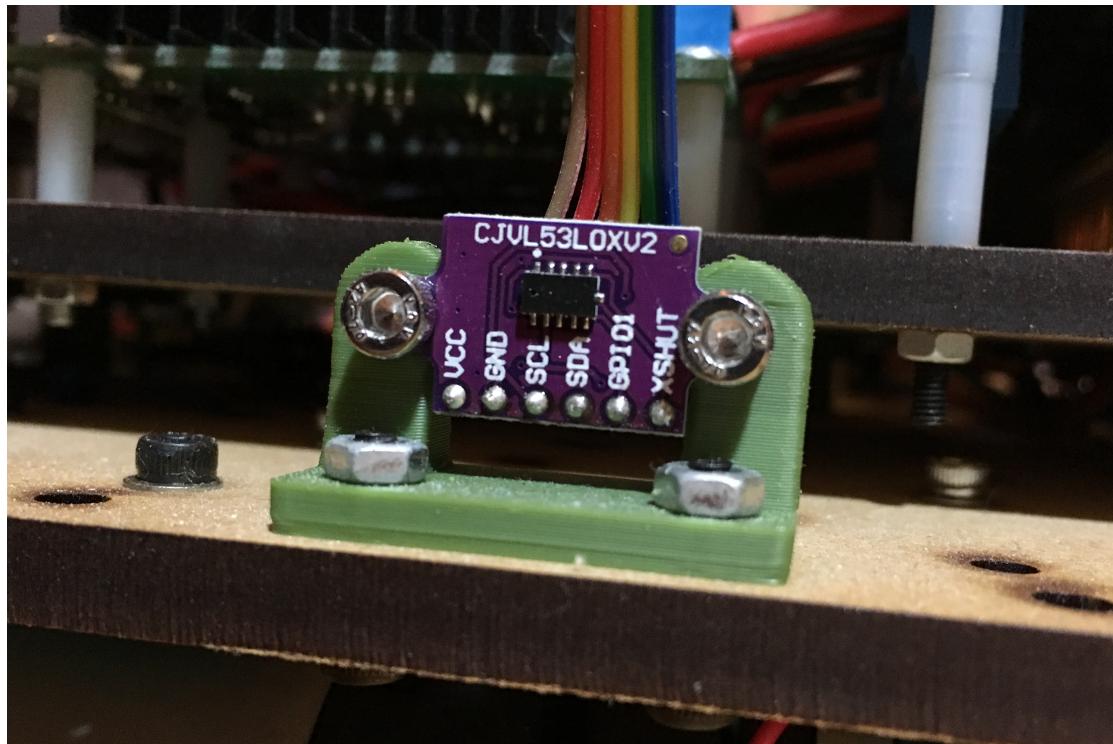


Figure 1.4: VL53L0X Rangefinder Mounted

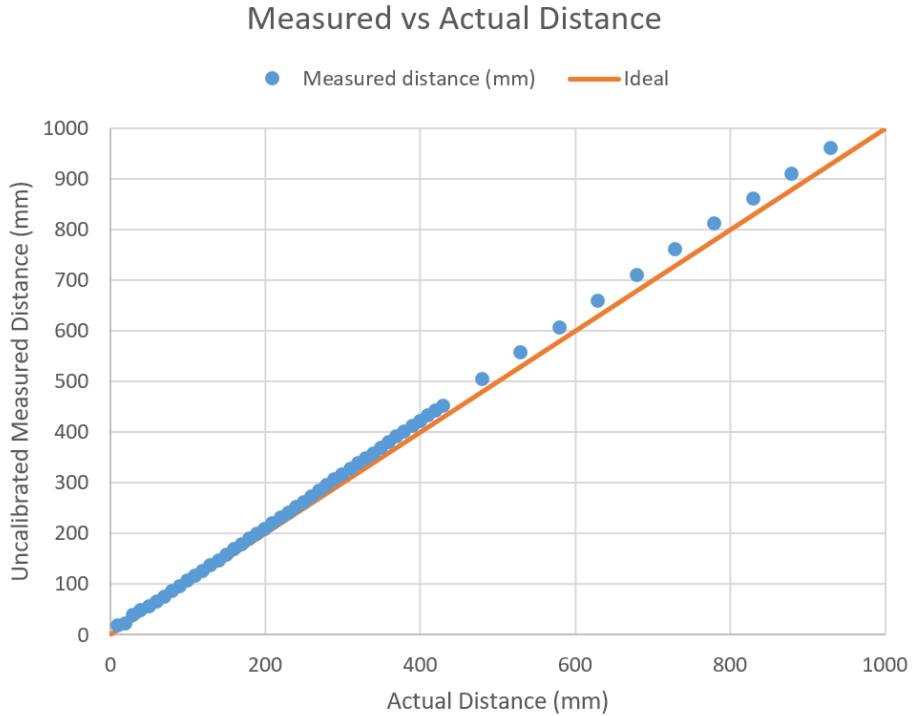


Figure 1.5: VL53L0X Measured vs. Actual Distance

Since the measurements are linear, only a linear correction is required as follows:

$$d_{calibrated} = m_{scale}d_{measurement} + b_{offset} \quad (1.2)$$

where $d_{measurement}$ is distance as returned by the sensor, $m_{scale} = 0.96502507$, $b_{offset} = -3.8534743$, and $d_{calibrated}$ is the calibrated distance. The squared error for each data point before and after calibration is shown in Figure 1.6. The mean squared error for the uncalibrated data points is 342.3 mm and 15.0 mm after calibration, indicating the use of an appropriate model and constants.

Squared Error for Uncalibrated vs. Calibrated

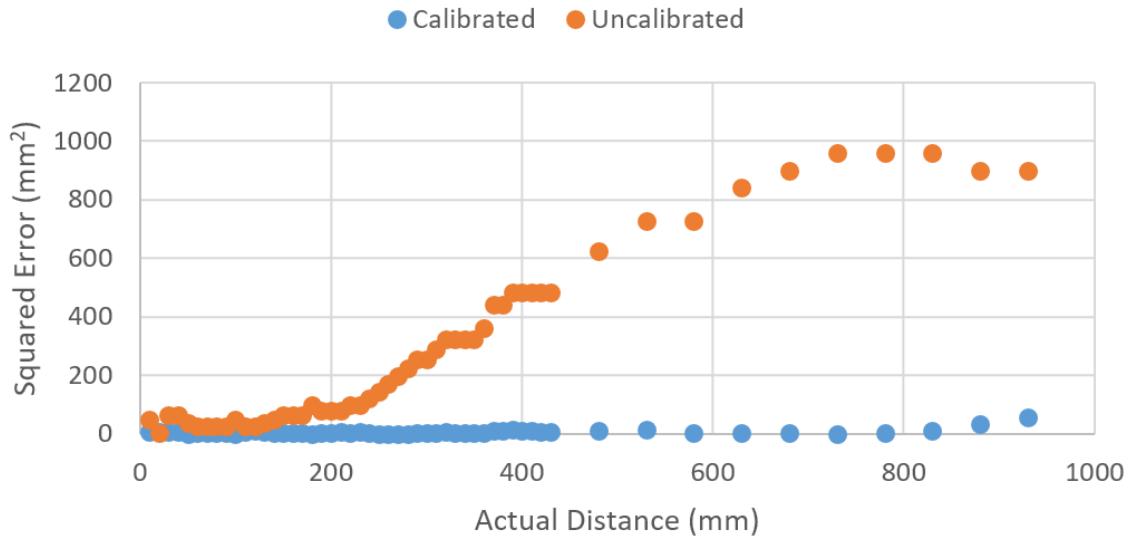


Figure 1.6: VL53L0X Squared Error for Uncalibrated vs. Calibrated

Additionally, 100 measurements were taken at each distance and the variance computed to characterize the spread. The results are shown in 1.7. The rangefinders are very accurate; at 900 mm with calibration, the error is only 7 mm or 0.8% error. Of course, this error is with respect to the average measurement at 900 mm. At 900 mm, the standard deviation is 14 mm so individual measurements can vary, especially with non-ideal surfaces. The closer the target, the more accurate and less varied the measurement. No experiments were carried out to determine the measurement characteristics on various colored, textured, transparent, or angled surfaces as the expected target surface is white painted wood.

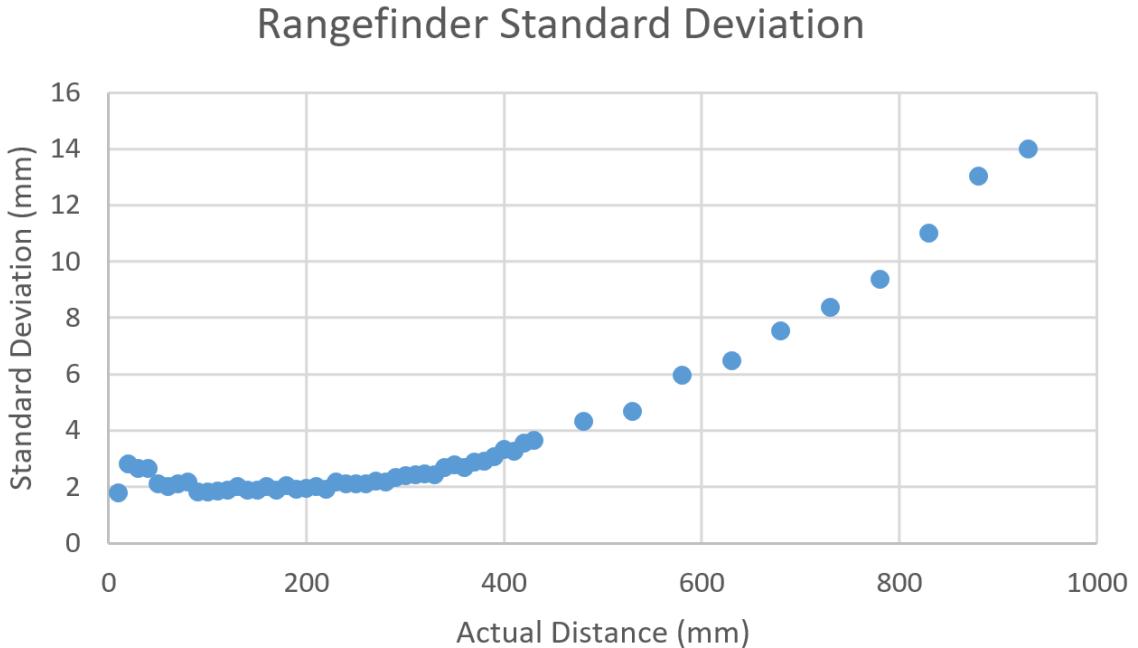


Figure 1.7: VL53L0X Standard Deviation

1.4 Motor Drivers

The system uses three off-the-shelf L298N motor driver boards since they are easily obtainable for less than \$6 each and incorporate features such as heat-sinking, flyback voltage protection, supply filtering, and screw terminal connections. Implementing comparable motor drivers with a similar feature set would undoubtedly cost more. Each L298N is a dual H-bridge driver with 2 A maximum output per bridge using a 5 – 35 V supply. Two motor drivers handle the four robot drive motors while the third powers the blower fan and shooting mechanism motors.

Figure 1.8 shows a wiring diagram for each motor driver. The board uses four digital control inputs, each controlling the state of one half-bridge. Each motor uses a pair of inputs: IN1 and IN2 control one motor while IN3 and IN4 control the other. To achieve direction and speed control, IN1 and IN3 are pulse width modulated (PWM) while IN2 and IN4 are digitally set. Table 1.1 is a truth table of the motor state

versus inputs.

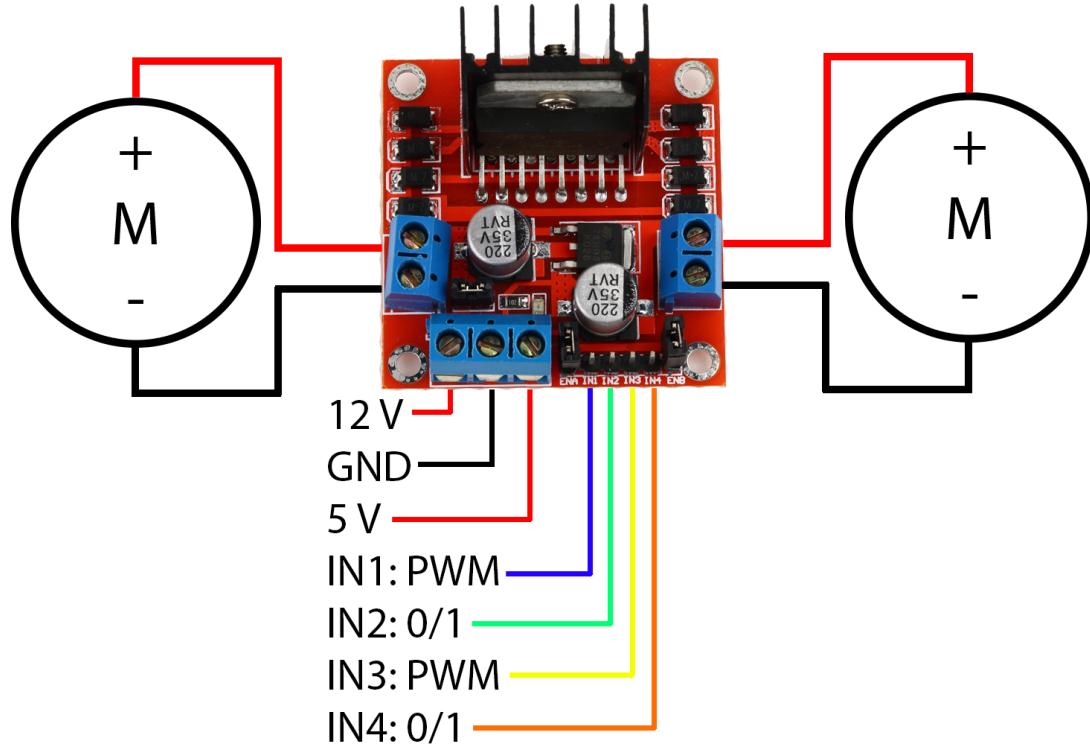


Figure 1.8: L298N Motor Driver Wiring Diagram [9]

Table 1.1: Motor Control Truth Table

IN1/IN3 Duty Cycle	IN2/IN4 State	Motor State
0%	0	Stopped
>0%	0	Forward, speed increases with duty cycle
<100%	1	Reverse, speed decreases with duty cycle
100%	1	Stopped

1.5 Servo

A GWS S03N standard servo powered from the 5 V bus actuates the gating mechanism in the ball hopper. Most servos are controlled by driving the control wire with a pulse

width modulated (PWM) signal with a period of 15 – 25 ms and a pulse width between 0.5 ms and 2.5 ms where the pulse width determines the position of the servo as shown in Figure 1.9.

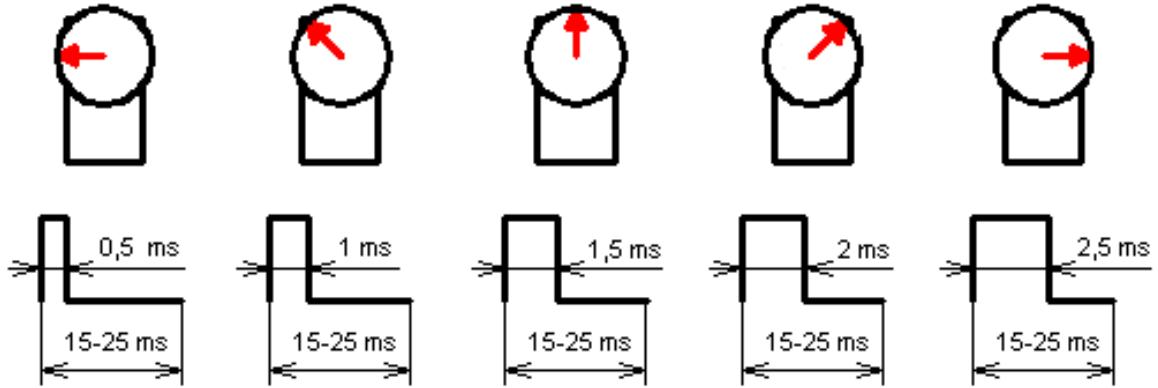


Figure 1.9: Servo PWM Control Scheme [8]

Due to cheap manufacturing and loose tolerances, the actual required pulse widths can vary from servo to servo, requiring calibration to obtain accurate positional control. The process for calibration is simple: apply various pulse widths and record the resulting servo positions. The calibrated pulse widths and servo angles are recorded in Table 1.2.

Table 1.2: Servo Required Pulse Widths

Servo Position	Pulse Width (ms)
0°	0.674
45°	1.082
90°	1.490
135°	1.898
180°	2.306

1.6 Microcontroller

An STMicroelectronics STM32F446RE microcontroller (MCU) serves as the bridge between the robot's low-level electronics and the high-level control system running on a desktop computer. Specifically, the MCU collects data from sensors over I²C and general purpose input/output (GPIO), generates control signals for the motor drivers and servo, and services commands from UART (universal asynchronous receiver-transmitter). The MCU resides on an STMicroelectronics Nucleo-64 development board, shown in Figure 1.10, which conveniently integrates an ST-LINK V2 debugger, programmer, and USB-to-UART interface.

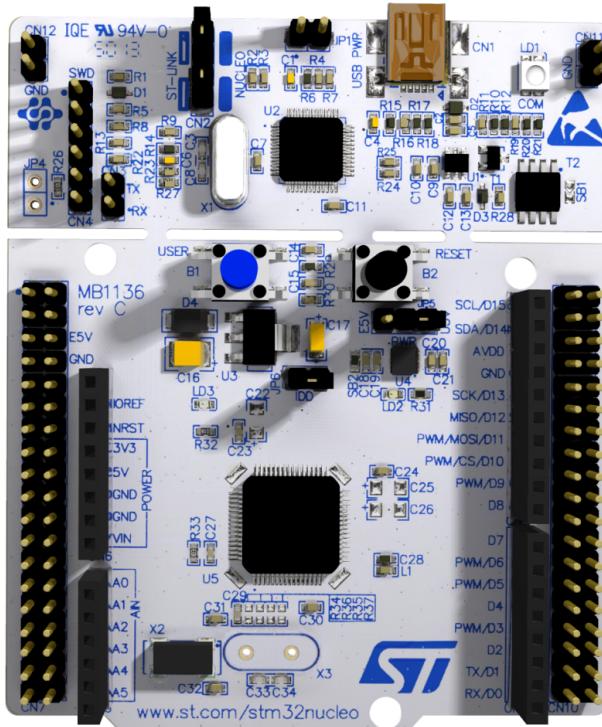
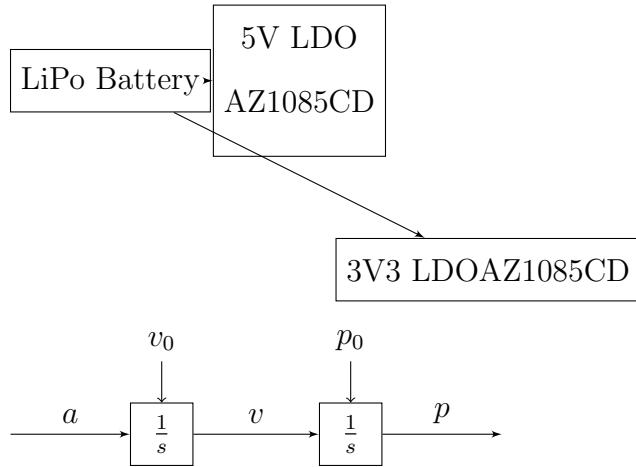


Figure 1.10: STM32 Nucleo-64 Development Board [14]

1.7 Interconnect PCB

The interconnect PCB is a custom designed board with three goals: regulate voltage, route power, and connect peripherals to the development board. The PCB routes input battery power to two external buck regulators, producing 12 V and 7 V. The 7 V bus is further regulated to 5 V and 3.3 V logic rails using two AZ1085CD LDOs and then routed to the development board, sensors, etc. Most importantly, the interconnect board provides dedicated 2.0 mm JST connectors for all sensors, motor drivers, and the servo to simplify and robustify wiring across the robot. In addition, hand-crimped cables made with JST connectors and 28 gauge ribbon cable prevent rats nest wiring. A block diagram is shown in Figure



1.7.1 Schematic Capture

1.7.2 Board Layout

1.7.3 Assembly

1.7.4 Reworks

BIBLIOGRAPHY

- [1] . . . , 2018. [Online; accessed May 11, 2018].
- [2] Adafruit. Adafruit 9-DOF IMU Breakout.
<https://www.adafruit.com/product/1714>, 2014. [Online; accessed May 15, 2018].
- [3] AliExpress. DC-DC CC CV Buck Converter.
https://www.aliexpress.com/item/DC-DC-CC-CV-Buck-Converter-Volt-Step-Down-12V-19V-24V-Car-Laptop-Power-Supply/32822603345.html?spm=2114.search0104.3.135.191f73dfqonWDU&ws_ab_test=searchweb0_0,searchweb201602_2_10152_10151_10065_10344_10130_10068_10324_10547_10342_10325_10546_10343_10340_10548_10341_10545_10696_10084_10083_10618_10307_10059_308_100031_10103_10624_10623_10622_10621_10620,searchweb201603_32,ppcSwitch_5&algo_expid=30a33cc9-3acf-4021-b514-6ba550085dd9-19&algo_pvid=30a33cc9-3acf-4021-b514-6ba550085dd9&priceBeautifyAB=0, 2018. [Online; accessed May 11, 2018].
- [4] Cal Poly. Roborodentia - Cal Poly, San Luis Obispo. <http://www.myurl.com>, 2018. [Online; accessed May 11, 2018].
- [5] J. Doe. *The Book without Title*. Dummy Publisher, 2100.
- [6] Fabio Varesano. FreeIMU Magnetometer and Accelerometer Calibration GUI.
<http://www.varesano.net/blog/fabio/freeimu-magnetometer-and-accelerometer-calibration-gui-alpha-version-out>, 2012. [Online; accessed May 11, 2018].
- [7] R. R. Labbe. Kalman and Bayesian Filters in Python, Sep 2017.

- [8] Leopoldo Armesto. How to generate a PPM signal with Arduino to control a servo? <http://robotica.webs.upv.es/en/how-to-generate-a-ppm-signal-with-arduino-to-control-a-servo/>, 2015. [Online; accessed May 11, 2018].
- [9] MechaMan. Working with L298N DC Motor Driver. <http://fritzing.org/projects/working-with-l298n-dc-motor-driver>, 2018. [Online; accessed May 13, 2018].
- [10] R. Negenborn. *Robot Localization and Kalman Filters*. PhD thesis, Sep 2003.
- [11] Nevada Mark. FAQ: Hard & Soft Iron Correction for Magnetometer Measurements. <https://ez.analog.com/docs/DOC-2544>, 2014. [Online; accessed May 11, 2018].
- [12] D. H. Nguyen and B. Widrow. Neural Networks for Self-Learning Control Systems. *IEEE Control Systems Magazine*, Apr 1990.
- [13] STMicroelectronics. LSM303DLHC Datasheet. <http://www.st.com/resource/en/datasheet/DM00027543.pdf>, 2013. [Online; accessed May 15, 2018].
- [14] STMicroelectronics. STM32 Nucleo-64 User Manual. http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf, 2017. [Online; accessed May 11, 2018].
- [15] STMicroelectronics. VL53L0X Datasheet. <http://www.st.com/content/ccc/resource/technical/document/datasheet/group3/b2/1e/33/77/c6/92/47/6b/DM00279086/files/>

DM00279086.pdf/jcr:content/translations/en.DM00279086.pdf, 2018.

[Online; accessed May 12, 2018].

[16] STMicroelectronics. VL53L0X Datasheet.

<http://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html>,

2018. [Online; accessed May 15, 2018].