INVESTIGATION INTO SPEED VS ACCURACY FOR AN AUTOMATED VEHICLE CHARGING SYSTEM

A Dissertation
Presented to
The Academic Faculty

by

Bryan Cochran

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Mechanical Engineering at the
Woodruff School of Mechanical Engineering at Georgia Institute of Technology

Georgia Institute of Technology
December 2020

INVESTIGATION INTO SPEED VS ACCURACY FOR AN AUTOMATED VEHICLE CHARGING SYSTEM

Approved by:

Dr. Bert Bras, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Jun Ueda
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Anirban Mazumdar
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: August 26th, 2020

[To the students of the Georgia Institute of Technology]

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Bert Bras in giving me this opportunity to work on such an amazing project. It has been a long time coming and there were many hurdles that seemed insurmountable at the time but without his guidance, much of this would not have been possible.

I would also like to thank my extensive team that has rotated through over the semesters. We had some great times and it was a pleasure to watch you all grow as the semester went on. I would also like to thank Zulfiqar Zaidi and Yosuke Yajima for the work we were able to put together in the final semesters. I am still happily surprised about all the challenges that we were able to overcome and the product that we ultimately put out. Lastly, I would like to thank other lab mates in the Sustainable Design and Manufacturing Lab for their advice and feedback.

I would also like to thank Dr. Jun Ueda and Dr. Anirban Mazumdar for serving as research committee members.

Finally, I would like to thank my wife and daughter for all their support as I have pursued this advanced degree. I know that I could not have done any of this without the both of you. I am blessed that the two of you are as nerdy as I am, and I cannot wait to share more of these experiences with the both of you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

BB	Bounding Box
DIY	Do It Yourself
EV	Electric Vehicle
FOV	Field of View
IOT	Internet of Things
PHEV	Plug-in Hybrid Electric Vehicle
PWM	Pulse Width Modulation
RFID	Radio Frequency Identification
ROS	Robot Operating System
SCARA	Selective Compliance Articulated Robot Arm
SLS	Selective Laser Sintering
SPI	Serial Peripheral Interface

SUMMARY

Recent advances in energy storage technology have finally allowed Electric Vehicles to enter the mainstream market. The suite of electronics used on these vehicles for power management and driving assistance opens the possibility of these vehicles operating autonomously. An autonomous vehicle must be recharged for it to drive to a destination beyond the range of battery system or for it to operate continuously. To extend autonomous operation, an autonomous charging system was developed. A design requirement was that the system be made using consumer grade components that are common to the DIY IOT movement to decrease system cost. The design and manufacture of the autonomous charging system will be briefly discussed but is not the focus of this thesis.

The focus of this thesis is the investigation into the relationship between the operating speed and the accuracy of the automation algorithm. Initial development focused on delivering the best performance, but the run time of the automation algorithm was more than ten minutes, which was too lengthy. The only portions of the code that could be improved were the hunt cycles for the port cover and the port detent. During the hunt cycles, the algorithm uses closed loop feedback between a vision system and the kinematics of the robot. The feedback loop compares the BB centroid to the center of the camera's FOV. The hunt is completed when the comparison drops below a defined threshold. For the hunts, the accuracy was decreased by increasing the threshold. Three thresholds were chosen for the Port hunt and the Detent hunt and those thresholds represented high, medium, and low accuracy. An experiment was conducted using different combinations of

accuracy for each hunt. The hypothesis was that it was possible for the cycle time to be reduced by decreasing accuracy without sacrificing system performance.

Test results validated the hypothesis and the cycle time was reduced by 16% without impacting system performance. This was done by using the lowest accuracy parameter for the charging Port hunt and using the medium accuracy for the Detent hunt. During the process of conducting the DOE, additional areas of improvement were identified for both the software and the mechanical systems. The proposed improvements were developed and implemented prior to outdoor, full-cycle testing. Outdoor tests were then completed and verified that the implemented improvements along with the accuracy parameters that were the outputs from the test results decreased the full cycle time by 16%.

CHAPTER 1. INTRODUCTION

1.1 Thesis Motivation

Electric vehicles have been around since the middle of the 1800's, but the energy density of their batteries was no comparison to that of steam or gasoline. Recent advances in electric energy storage and battery management systems have finally allowed electric vehicles to enter the mainstream market almost a century and half after their first creation. The first mainstream rechargeable battery composition was a sealed lead acid battery and it was used for the first electric vehicles back in the early 1900s (U.S. Department of Energy 2019). The sealed lead-acid battery was not good as a primary power system for the early electric vehicles, but the advent of the lithium polymer battery in the 1990s gave electric vehicles new life. Lithium polymer batteries have two to three times the energy density and are much lighter than a sealed lead-acid battery but chemically more unstable (Albright, Edie, and Al-Hallaj 2012). This means that for a lithium polymer battery to operate sustainably for an extended period, its charging and discharging cycles must be meticulously maintained by a dedicated management system. Figure 1 below depicts an example layout of passive and active safety mechanisms that are used in most Lithium Ion batteries.

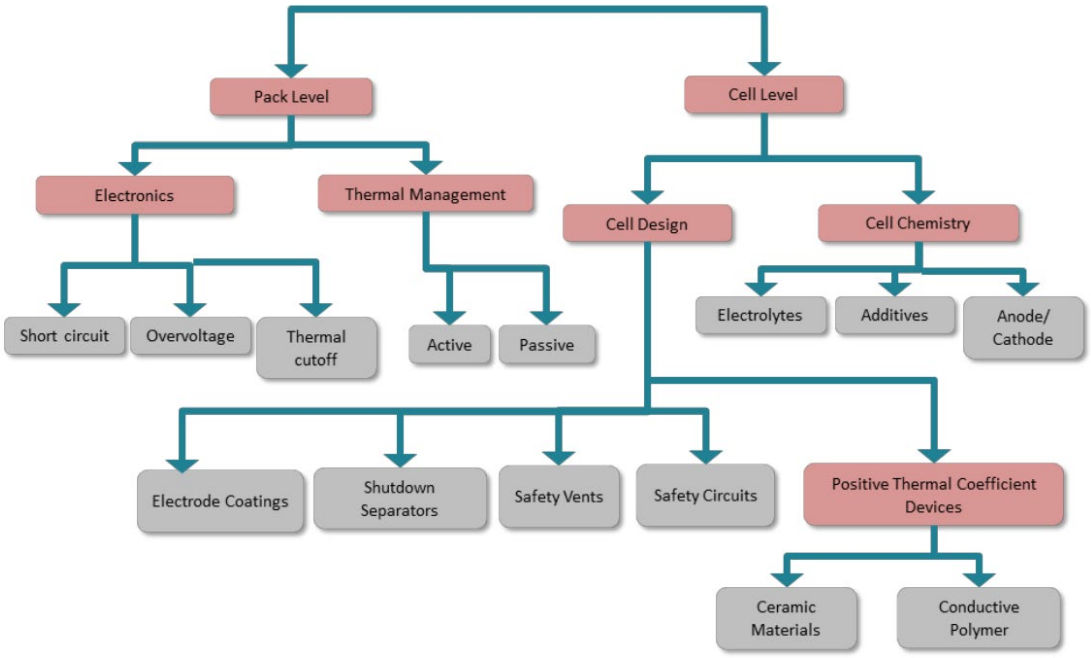

Figure 1: Lithium Ion Safety Mechanisms (Albright et al. 2012)

Recent developments in microcontroller technology have decreased the size and cost of these management systems. The pairing of lithium polymer battery technology with these new microcontrollers has led to the growth in EV production and sales over the last decade. Additionally, the decrease in the cost of these microcontrollers has made them easily accessible to consumers and spawned the Internet of Things (IOT) movement.

1.1.1 Electric Vehicles

There are two main types of electric vehicles: All-electric Vehicles (AEV or EV) and Plug-in Hybrid Electric Vehicles (PHEV). An EV runs solely on electricity and has no petroleum powered systems whereas a PHEV has a drive system that utilizes both electricity and petroleum power systems. The average range of most EVs on the market is between 80 and 100 miles, but some models can be driven more than 200 miles (U.S.

Department of Energy 2017). On the other hand, a PHEV can be driven indefinitely using its petroleum power system. By charging and engaging the electrical system, the operating costs are decreased by increasing the fuel efficiency of the vehicle. Both the EV and the PHEV can be connected to the grid so they can recharge. This is either done at home with a charger that connects to the home power grid or at a public charging station.

1.1.2 Autonomous Vehicles

The growth in microcontroller technology that helped make EVs possible is also already being used in the automotive industry to make cars safer. Modern cars use microcontrollers to monitor and manage braking systems, steering systems, throttle control, air conditioning, fuel injection, and roughly every individual system that is onboard. Because of the level of integration that these systems have, some companies have begun networking these systems to create fully automated vehicles for various economic motivations. In 2019 the National Safety Council estimated that approximately 38,800 people would die from traffic collisions in the United States and that 4.423 million would suffer nonfatal medically consulted injuries from collisions. The costs from these collisions is estimated at $400.4 billion for 2019 and reduction in these expenses is a major economic incentive (National Safety Council 2020). Widespread use of autonomous vehicles would substantially reduce these collisions by removing the root cause of most of these collisions, human operators.

There are numerous ethical considerations that are currently being debated that need to be considered regarding the safe, autonomous transportation of a human being whereas transport of simple freight contents circumvents these ethical considerations.

Because of this, the first application of vehicle automation that is being investigated is delivery of goods. Currently, if a person wants to drive to a destination with an EV that is outside of its battery range, he or she must plan a stop in transit where the vehicle can be recharged. This poses a problem unique to an automated vehicle. If an automated EV needs to extend its range by recharging itself, how does it do so? Current EV charging stations are no more than a weatherproof plug that is inserted into the EV by the human operator. The operator simply inserts the plug into the corresponding receptacle however the EV would either need to recruit a human to help or just sit indefinitely.

1.1.3 Robotic Charging Systems

A few companies and research institutions have worked at tackling this issue by developing a system that helps the EV by charging it automatically. Beside the benefit to automated EV systems, human consumers also find the ease-of-use of an automated system attractive. These systems can recognize when an EV or PHEV approaches and is needing a charge. The system then opens the charging receptacle if the vehicle has not done so already and inserts the charger to begin charging the battery. Most of the systems that have been developed thus far are very expensive though. Many use small industrial robotic arms and controllers that make the low-end cost of most systems more than $50k. These systems will be discussed in depth in the Literature Review in Chapter 2. This price tag puts it out of reach for most consumers and a difficult consideration for most businesses. Because of this, a lower cost option needed to be investigated, which leads to the objective of this thesis.

1.2 Thesis Objective

1.2.1 Project Overview

As an extension of a Georgia Tech Mechanical Engineering Capstone project that was sponsored by an external research sponsor, we developed an Automated EV Charging robot with the purpose of having the system cost being under $1,500 so it would be commercially viable for sale to individual consumers. To decrease cost, the control system was built using hobby grade microprocessors and the mechanical structure was modularly designed for fabrication instead of using a prefabricated robot. The system was designed to operate on a PHEV that was provided by the project sponsor for use in the system's development. The system also had to operate solely on its own with no interaction with the vehicle. This meant that the vehicle could not be modified to assist in system operation and there could be no communication between the vehicle and the charging system. The production vehicle needed to simply drive up and the system would actuate to charge the vehicle.

1.2.2 Thesis Overview

The purpose of this thesis is to outline the testing intended to reduce the overall operating time of the automated EV charging system's automation software. After this introduction, the literature review will discuss current automated charging systems and identify the areas of opportunity that motivated the creation of this system. Next, the details of the robotic system will be discussed in the Robot System Overview. The details of the design decisions that led to the current mechanical and software configurations will be omitted from this thesis, but the mechanics, electrical layout, and software of this robot will be discussed. The investigation into the effects that the accuracy criterion within the

software has on the system performance is the topic of the investigation in this thesis. The purpose and procedure of this test will be discussed along with the data from the test and the analysis of that data. Lastly, the implementation of the findings from the test will be reviewed prior to the conclusion of this thesis.

CHAPTER 2. LITERATURE REVIEW

This section of the thesis summarizes current industry solutions related to automated charging or refueling systems. Recent growth of the EV market and the expectation of the implementation of self-driving vehicles has led to a push in the development of automated charging systems. The robot mechanics and control software of the systems that are being developed use commercially available robotic components and controllers. Because of this there is very little academic literature related to the development of automated charging or fueling systems. The mechanical design of automated charging systems will be discussed followed by the study of the control system architecture. Finally, the methods of tuning parallel arm robotics systems will be discussed.

2.1 Mechanical Design of Automated Charging Systems

Multiple industrial and academic institutions are developing automated recharging systems across the world. With the increased use of unmanned aerial vehicles and the anticipation of the implementation of automated ground vehicles, these systems would be useful as support systems for autonomous craft. While some of the systems are mobile and are intended to deploy themselves to the target vehicle or provide support off the grid, most have been developed to be stationary. They are employed like current fueling systems but do not require human interaction to complete refueling.

The mobile systems are designed for deployment in remote areas or to help stranded vehicles. One such system is discussed in United States Patent US009873408B2 (Capizzo 2018). This patent discusses the embodiment of a vehicle that is referred to as an Automatic

Service Station Facility, or ASSF. This system is an automated service rover that will carry a variety of liquid fuels or electric storage devices and solar panels depending on which embodiment is chosen. The intended purpose of this system is to provide remote fueling or recharging of other unmanned systems be it for military, government, or commercial purposes. Though it could be used to support standard sized automobiles, the ASSF was designed to support medium sized ground based or aerial based drones. Vehicles needing to refuel, or recharge would need to drive onto or land on the ASSF for refueling.

Figure 2: FreeWire Mobi EV Charger

Other mobile systems have been developed specifically for automobile EVs. One of these systems is developed by a company called FreeWire. Their current flagship system, the Mobi EV Charger, is little more than a large format EV battery (FreeWire Technologies 2020) and can be seen above in Figure 2. This battery system is installed on

a wheeled cart or commercial sized gasoline vehicle and is deployed to recharge stranded electric vehicles in the same way that a stranded motorist gets a gallon of gasoline to drive to the nearest gas station. FreeWire is also developing another system, as seen in United States Patent US009592742B1(FreeWire Technologies 2017). In this system, the cart for the Mobi EV Charger is motorized and equipped with a robotic system that allows for autonomous interaction with the EV. The specifics of the interaction mechanism are not discussed in the patent.

Figure 3: Volkswagen e-smartConnect Automated EV Charging System

The final mobile system to be discussed has been developed by Volkswagen with the collaboration of Kuka robotics. Named the VW e-smartConnect, this system is shown above in Figure 3 and is designed to interface with the Volkswagen e-Golf. It is also compatible with other vehicles that use standard EV charging connectors. The system consists of a Kuka LBR iiwa serial robotic arm mounted to a mobile base and DC EV plug integrated into the base. The EV plug is the same that is currently used by consumers and the end effector of the Kuka arm is modified to grasp the charging plug. The client vehicle

transmits its profile and location to the charging system. The system is transported to the target location using a conveyor system. The system the engages with the vehicle using a vision system mounted on the end effector of the robotic arm. There are additional sensors that are used in this system to promote the accuracy and speed of its operation, but those systems have not been publicly disclosed (Volkswagen 2015).

As previously stated, most of the automated charging systems have stationary bases. They operate based off the same mentality behind current automobile fueling stations. The target vehicle drives within proximity to the fueling station where the operator, or the automated system, completes the fueling process on the vehicle. One of the earliest automated fueling systems that was patented was specifically designed for refueling petroleum vehicles. This system was patented in 2001 by William Pong and Edward Fredkin under the United States Patent US006237647B1 (Pong and Fredkin 2001). This system is designed for fueling a vehicle, not recharging an EV, but the system operations are nearly identical. It was intended to be implemented as a retrofit on standard gasoline fueling systems. The automated system would grasp the fuel nozzle from the station, open the fuel port on the vehicle, and complete the fueling process in the same manner as a human operator would. This system would use any serial robotic manipulator and any sensors that would be necessary for normal operation (Pong and Fredkin 2001). No information was found regarding whether or not this system was ever implemented but the operations of the system are very similar to most EV charging systems and as such, this patent was a reference to most of the patents that were reviewed in the creation of this literature review.

The automated charger that may be the most well-known is the charging “tentacle” that was developed by Tesla (Bishop 2015). The benefit of this design is the compact nature of the system. However, at this time the system has only been designed to only operate on Tesla vehicles. The specifics of the control system or the sensor suite have not been made public. It is unknown how the system operates to find the position of the charging port on the Tesla.

Figure 4: Kuka Carla_connect Automated EV Charging System

After the collaboration with Volkswagen to produce the e-smartConnect, Kuka developed their own automated charging system, called the Carla_connect (Kuka AG 2020) and is seen above in Figure 4. This system is specifically designed for indoor use and mounting to a vertical surface, such as a wall. The system uses a SCARA serial arm robotic arm along with a vision system incorporated into the end effector of the arm. The Carla_connect system is very similar to a system developed by a company called Power

Hydrant. The Power Hydrant is also a SCARA serial arm robotic system (PowerHydrant LLC 2020) and an image of this system can be seen below in . The main difference between the two systems is that the Power Hydrant is designed to interface with the front of the vehicle. This means that vehicles that do not have front mounted receptacles need to be modified to mate with the system. Based off the patent held by Power Hydrant, other interfacing options that they could investigate are magnetically actuated couplers or powered screws to mechanically interface the charger and the vehicle (Leary and PowerHydrant LLC 2016).

Figure 5: PowerHydrant Automated EV Charging System

2.2 Control System and Sensor Utilization of Automated Charging Systems

During the investigation of potential control systems for an automated charging or fueling system, three design approaches emerged. The first approach was that the system would be compatible with one vehicle make or model. This kind of system usually was developed by an organization that collaborated with a vehicle manufacturer. This is the case for the Volkswagen and Kuka charging systems; the e-smartConnect (Volkswagen

2015) and the Carla_conect (Kuka AG 2020). The e-smartConnect was developed to work specifically on the Volkswagen e-Golf and the system controller directly communicates with the vehicle. This vehicle-to-charger communication provides closed-loop feedback that helps the vehicle park closer to the charger and helps the charger now when the charging cycle is complete. Additionally, given that only one vehicle is can mate with the charger, the system can be trained to know where the charging port is in 3D space.

The second design approach is that the system is compatible with variety of vehicles, but the vehicles must be modified to facilitate communication with the system beforehand. Compared to the system designed for use on one manufacturer's vehicle, this system requires additional sensors to detect the charging port and it also requires modifications be made to the client vehicle to make it compatible. Modifications to client vehicles were either made to facilitate physical connection to the EV charging system, as seen in United States Patents US20130338820A1 and US009493087B2, or to facilitate communication with the EV charging system, as seen in US006237647B1, US20140203077A1 or US8853999B2.

The charging interface modifications recommended in patents US20130338820A1 and US006237647B1 were made to simplify the automated system requirements by making the interface easier to find or engage with. In the case of US20130338820A1, the EV charging interface would be relocated to the front of the vehicle. A modified license plate holder would be installed that would contain two parallel conductive surfaces. These surfaces would interface with a coupler on the end effector of the EV that would resemble ping pong paddles (Corbett and Maniaci 2013). The PowerHydrant system discussed in patent US009493087B2 would use a standardized charging socket, but the socket would

have to be relocated to the front of the vehicle. This would not require any modification to be made to the EV if it already had a front mounted charging socket but EVs with sockets on the side would either by incompatible with the system or would require modifications (Leary and PowerHydrant LLC 2016).

One such proposed system requiring modifications to facilitate communication is seen in United States Patent US006237647B1. This patent discusses the potential sensor suites and control methods of the automated system using those sensors. This system has a detector system that can identify the vehicle and subsequently access a database to retrieve information about the identified vehicle. Access to the database is triggered by either a RFID system or a transponder, which may need to be installed in the vehicle. The database stores vehicle geometry so the fuel filler can be easily located along with customer information. A vision system can also be used to detect the arrival of a vehicle and the position of the fuel filler. The robotic arm can also employ a camera to be used for control feedback during the insertion procedure. (Pong and Fredkin 2001).

The PowerHydrant system also requires the creation of a user profile to operate. This profile is used for storing the relative position of the charging port and client's information for billing. For the system to operate, the vehicle will create a wireless connection with the charger station. The signal proximity and magnitude triggers the operation of the system and shares the profile data with the system (Leary and PowerHydrant LLC 2016). A system designed by a team at University of California at Oakland, US20140203077A1, and one designed by Joseph Haddad, US8853999B2, both required the use of imbedded RFID tags to operate. The UC Oakland system uses the RFID for access and authentication to the network and scheduling (GADH et al. 2013) whereas

the Joseph Haddad system uses RFID to detect and identify a vehicle that needs to be charged. The system has a database of vehicle information of which it uses identify the vehicle based off its image and determine the location of the charging port based of information triggered by the RFID tag. The robotic arm is configured to move the charging connector based off information retrieved from the vehicle database to engage the port and commence vehicle charging (Haddad and Lysak 2018).

The final design approach is a system that operates with any compatible vehicle and without any communication with the vehicle or modifications to the vehicle. During research for this literature review, no such system was found. All systems found in this literature review detailed a communication interface between the charger and the vehicle. This is significant because the design requirement was that our system must be able to complete a charging cycle with no communication with the target vehicle and the target vehicle could not be modified to facilitate finding the location of the charging port on the vehicle.

2.3 Accuracy Tuning a Robotic Arm with Vision Based Closed-loop Feedback

The robotic charging system that is the focus of this thesis uses a camera mounted on the end effector for closed-loop feedback to the control system. All reviewed autonomous charging systems had a vision system and most had a camera mounted on the end effector as well. As stated in section 2.2, the reviewed systems all detailed a communication interface between the charger and the vehicle. The information shared between the two systems helps the robotic arm move very close to the charging port on the EV where the camera is used to verify the location prior to insertion. The system this thesis

focuses on relies solely on the camera and the vision system to find the location of the charging port. This manner of operation is like the implementation of vision systems for material handling and camera-based robotic calibration systems.

A system developed by T. Tsay et al. for use in semiconductor facilities operates using a similar vision system application. This robotic system was designed for autonomous deployment to different manufacturing lines. The system involves a serial robotic arm with a gripper end effector fitted with a monocular camera with zoom capabilities. The system is mounted to a mobile base that can drive autonomously. The camera is used initially for obstacle avoidance during transit, but it then transitions to being used for positing when it is within proximity of its desired location. The target location is identified by a positioning mark and once the system detects the mark, it evaluates whether the lateral position of the system is within the defined accuracy range. If the system has not achieved desired accuracy, then it repeats the procedure an additional time. To further enhance the accuracy of the system, the optical lens of the camera can be zoomed in to fill the positioning mark within the field of view and adjust the system's position if necessary (Tsay, Lai, and Hsiao 2010).

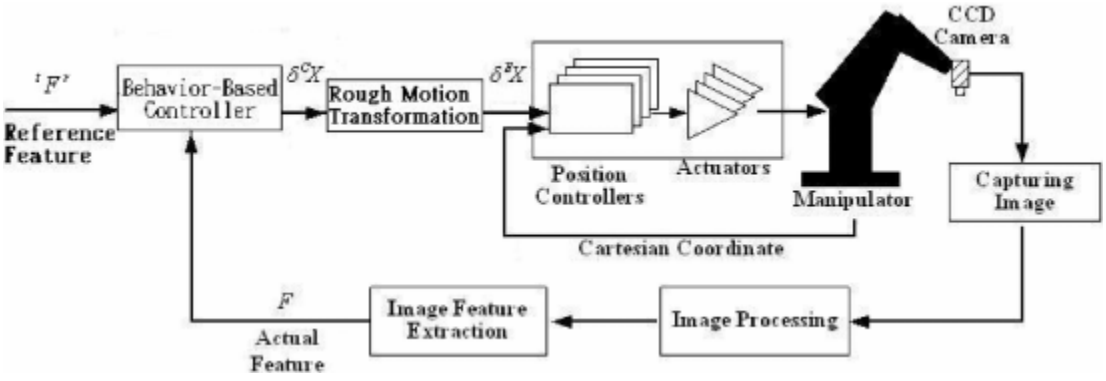

Figure 6: Behavior-based look-and-move control structure (Tsay et al. 2010)

The logic of the system developed by T. Tsay et al. is also used for picking operations of the system after the relocation process has been completed. The algorithm is operates using a behavior-based look-and-move control structure (Tsay and Chang 2004), which is seen above in Figure 2. The end effector camera captures of the workpiece, which is quadrangle, and analyzes the image to extract the coordinate of the center of geometry, the angle of the quadrangle within the field of view, and the ratio of the negative space on the two sides of the quadrangle. The system then feeds this data into six neural fuzzy controllers to convert the image feature errors into motion commands. To handle inaccuracies between the camera and the end effector, the neural fuzzy controllers use a back-propagation algorithm to tune the consequents of the fuzzy rules. This allows the system to respond accurately without the need of calibration of the camera and the end effector (Tsay et al. 2010).

This calibration of a robotic system with integrated video feedback is discussed extensively in the book published by H. Zhuang et al., Camera-Aided Robot Calibration. Though multiple calibration techniques are discussed in this book, the calibration technique of interest is the Robot Accuracy Compensation discussed in Chapter 11. The accuracy compensation technique discussed in this section is broken into three steps.

1. Repeatability assessment of the robotic system
2. Determination of the positioning errors of the robot system
3. Compensation of for the robot positioning errors

The first step requires that the robot be repeatedly moved to the same position and the end effector positions are recorded. The process is repeated in multiple positions within

the robot's workspace and statistical data is computed. This statistical data defines the upper bounds of the system's performance. The second step is like the first, but this time the robot's workplace is defined by a calibration board with dot patterns. The robot is moved to various points on the calibration board and the end effector position at these points is recorded. Statistical data is computed based off the differences between the desired and end effector positions. From this positioning errors at each point can be defined for each point. By knowing the positioning errors at each point, the system then compensates for those errors by using bilinear interpolation between adjacent points on the calibration grid (Zhuang and Roth 2018).

2.4 Comparison between Parallel and Serial Robotic Arms

The robotic systems that have been discussed thus far in this literature review have all utilized serial manipulator but the robotic system that is the focus of this thesis is a parallel manipulator. As such it is necessary to discuss the advantages and disadvantages that these different configurations possess and ultimately why it was decided for our automated charging system to utilize a parallel configuration. Figure 7 shown below outlines characteristics of parallel and serial robotic manipulators.

Feature	Serial robot	Parallel robot
Workspace	Large	Small and complex
Solving forward kinematics	Easy	Very difficult
Solving inverse kinematics	Difficult	Easy
Position error	Accumulates	Averages
Force error	Averages	Accumulates
Maximum force	Limited by minimum actuator force	Summation of all actuator forces
Stiffness	Low	High
Dynamics characteristics	Poor, especially with increasing the size	Very high
Modelling and solving dynamics	Relatively simple	Very complex
Inertia	Large	Small
Areas of application	A great number in different areas, especially in industry	Currently limited, especially in industry
Payload/weight ratio	Low	High
Speed and acceleration	Low	High
Accuracy	Low	High
Uniformity of components	Low	High
Calibration	Relatively simple	Complicated
Workspace/robot size ratio	High	Low

Figure 7: Characteristics of Serial and Parallel Robots (Pandilov and Dukovski 2014)

Because of cost considerations, our development team had to build our robot from the ground up instead of purchasing a pre-built robotic system. Serial arms were initially discussed due to the extensive knowledge base regarding serial arm applications along with the workspace that a serial system could operate in coupled with its relatively small size. Serial robotic systems are used commonly in robotic assisted assembly applications, welding, and machine loading and unloading due to ability to perform repetitive tasks in confined spaces (Pandilov and Dukovski 2014).

Looking into other configurations of robotic systems we started looking into parallel manipulators and the low inertia, high accuracy, and high stiffness that is inherent in parallel systems was very attractive in the initial design stages. These specific characteristics are major contributors to why parallel systems are used for motion platforms, fine positioning, or pick-and-place applications in industry (Pandilov and Dukovski 2014). The workspace limitations of a parallel configurations are one of the major contributors to its smaller areas of application when compared to the serial configuration.

2.5 Literature Review Summary

There are various embodiments of automated charging systems that are being developed currently. All systems identified during this literature review are either made to work autonomously on a specific vehicle make or model through collaboration with the vehicle manufacturer or require that the vehicle's geometry be pre-loaded on a database for operation. Additionally, all identified systems require some form of communication between the automated charging system and the target vehicle. The systems are also made using a serial robot configuration and no parallel robotic systems were found. In the following section, the robotic system that is the focus of this thesis will be discussed. This system utilizes a parallel configuration and operates with no interaction from the target vehicle.

CHAPTER 3. ROBOT SYSTEM OVERVIEW

3.1 Customer Specifications

The robotic system that is the focus of this thesis was created through the support by an external research sponsor. The design of the robot and the operational specifications are the result of a Capstone design project from Fall of 2017. The robot needs to consistently insert a Level 2 SAE J1772 EV charger into a corresponding SAE J1772 female socket on the fender of a PHEV that was loaned from our external research sponsor (SAE 2017). The robot must open the protective cover on the socket and then insert the male charging adapter into the female socket. The operational envelope was also specified by the sponsor. The operational envelope is the area where the system was required to consistently insert the charger given the charging port cover of the parked car is within the window. The mechanical design and the motor sizing were dictated by the specifics of operational envelope shown below in Figure 1.

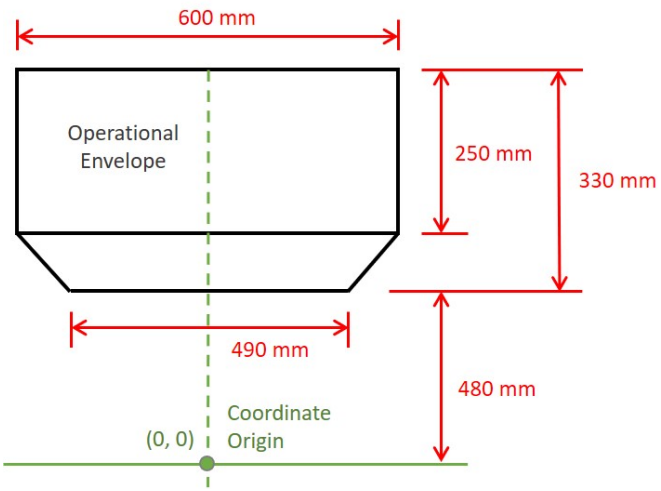

Figure 8: Robot Operational Envelope

3.2 Off-the-shelf Components

The other requirement from the sponsor was that the system needed to be made as economical as possible using off-the-shelf electronics that have recently become mainstream with the DIY IOT movement which meant that we could not use an industrial PLC or pre-built robotic system. The logic of the system was executed using four Arduino Uno microcontrollers, a Nvidia Jetson TX2, and a Raspberry Pi 3B+. The outline of this system is seen below in Figure 2. The Jetson was also used to process the real-time video streams from two Logitech C920 webcams and publish the processed video data for the Raspberry Pi. Three of the Arduinos were used as SPI communication interfaces with the three rotary induction encoders used on the robot. The fourth Arduino was used as a sensor interface to communicate with the Raspberry Pi. The sensor Arduino interfaced directly with a force sensitive resistor that acted as a touch probe and consolidates data from three SPI Arduinos.

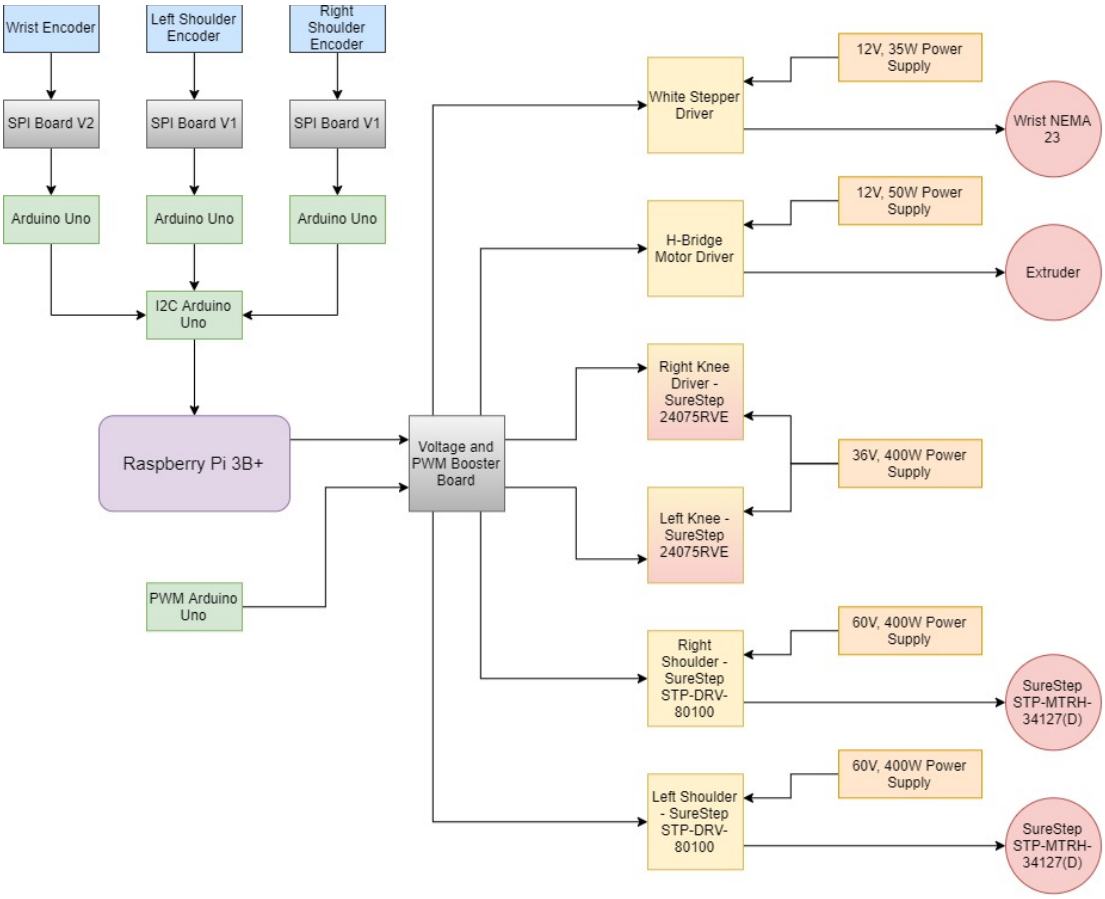

Figure 9: Diagram of Electrical Layout

3.3 Engineered Components

The sensor Arduino, the Jetson, and the Raspberry Pi all connected on the same network. The network between these three devices allows use of the Robot Operating System (ROS) for communication. The operating logic of the robot was executed by the Raspberry Pi and the Raspberry Pi outputs the control signals to the motor drivers through an intermediary board that increased the signal voltage and amperage so it properly interfaces with the motor drivers. The board boosted the 3.3V signals from the Raspberry Pi to a 5V signal which was required by all the motor drivers. The board also supplied a PWM waveform that sets the speed of the motors during operation so the 5V signals

include that modulated pulse. The schematic of the signal booster board is shown below in Figure 3.

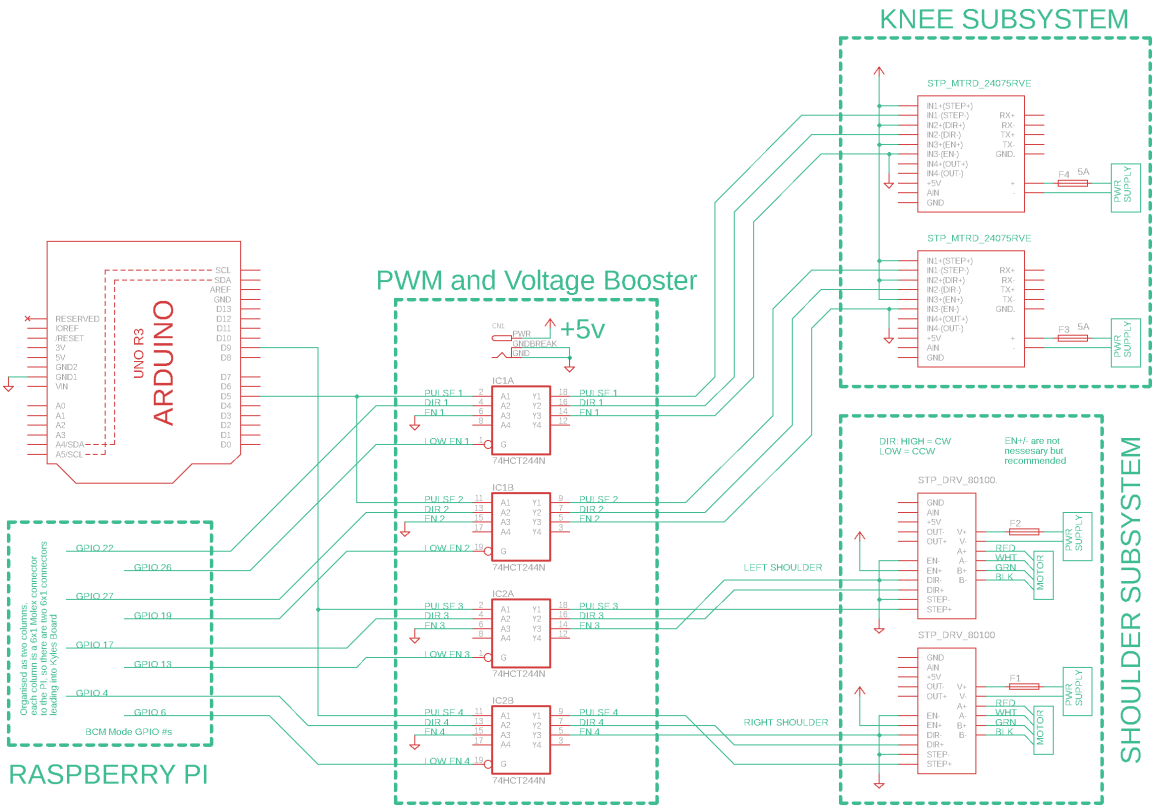

Figure 10: Schematic of PWM and Voltage Booster Board

3.4 Mechanical System Design

The robot was structurally designed as a 3 DOF Parallel, SCARA with an additional DOF achieved through a rotation joint at the end effector. Figure 4 shown below illustrates the robot during an insertion operation with the vehicle that was used during the development. Using the following coordinate system to describe the robot; the Z-axis was the vertical axis parallel to the tower of the robot, the Y-axis came straight out from the robot, and the X-axis moved laterally. The prismatic joint translated along the Z-axis and was mounted to the tower of the robot. Seen below in Figure 5, the NEMA 34's that drive

the arm are mounted to a mild steel plate. The plate is mounted to a linear bearing and two linear actuators that were driven by two NEMA 24 stepper servos. The two rotating joints that control motion on the XY plane were driven by two NEMA 34 motors that are each mounted to a 5:1 planetary gearbox which increased the output torque of the system to allow for reliable opening and insertion operations.

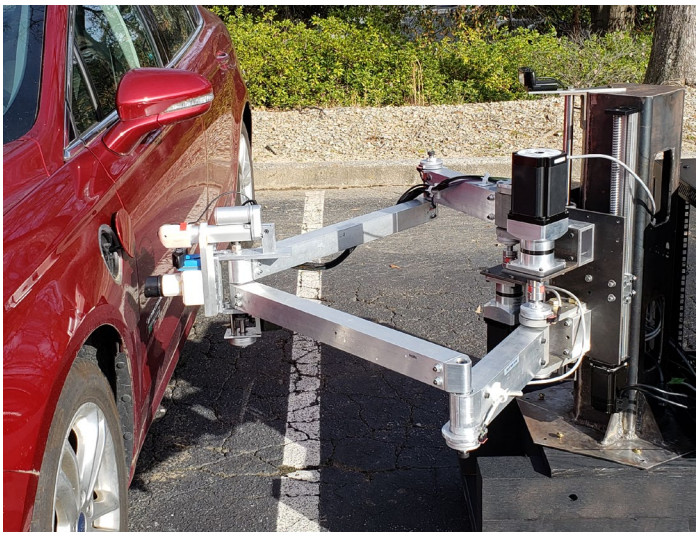

Figure 11: Parallel Arm SCARA Robot for autonomous EV charging

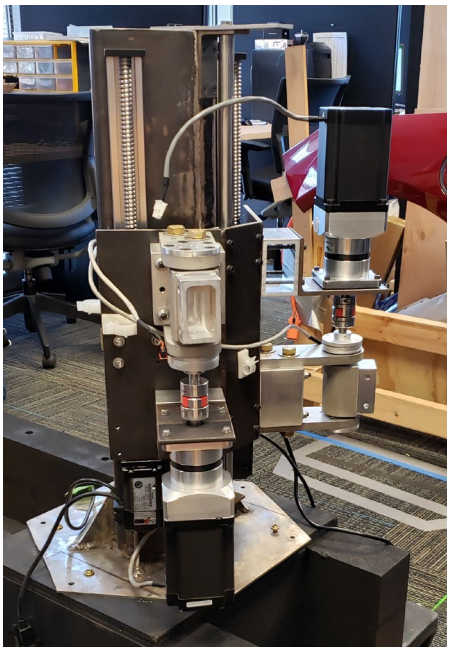

Figure 12: Close view of Parallel Arm SCARA Robot NEMA 34 Direct drive stepper motors with 5:1 planetary gearsets.

As specified by the manufacturer, the backlash of this gearbox coupled with the NEMA 34 is less than 25 arcminutes, or .417 degrees (SureGear 2020). The output of the gearbox was coupled to the input to the robot's arms using a jaw spider coupling with a shore 92 urethane spider. Lastly, the rotational joint of the End Effector was powered by a NEMA 17 stepper motor. The NEMA 17 mounted perpendicular to the axis of rotation, so motion was transferred to the end effector by way of a 2:1 straight miter gear combination. The backlash of this gearset was around 2-3 degrees. The end effector also used a linear actuator for the cover opening operation and distance measurement used a force sensitive resistor mounted to its tip.

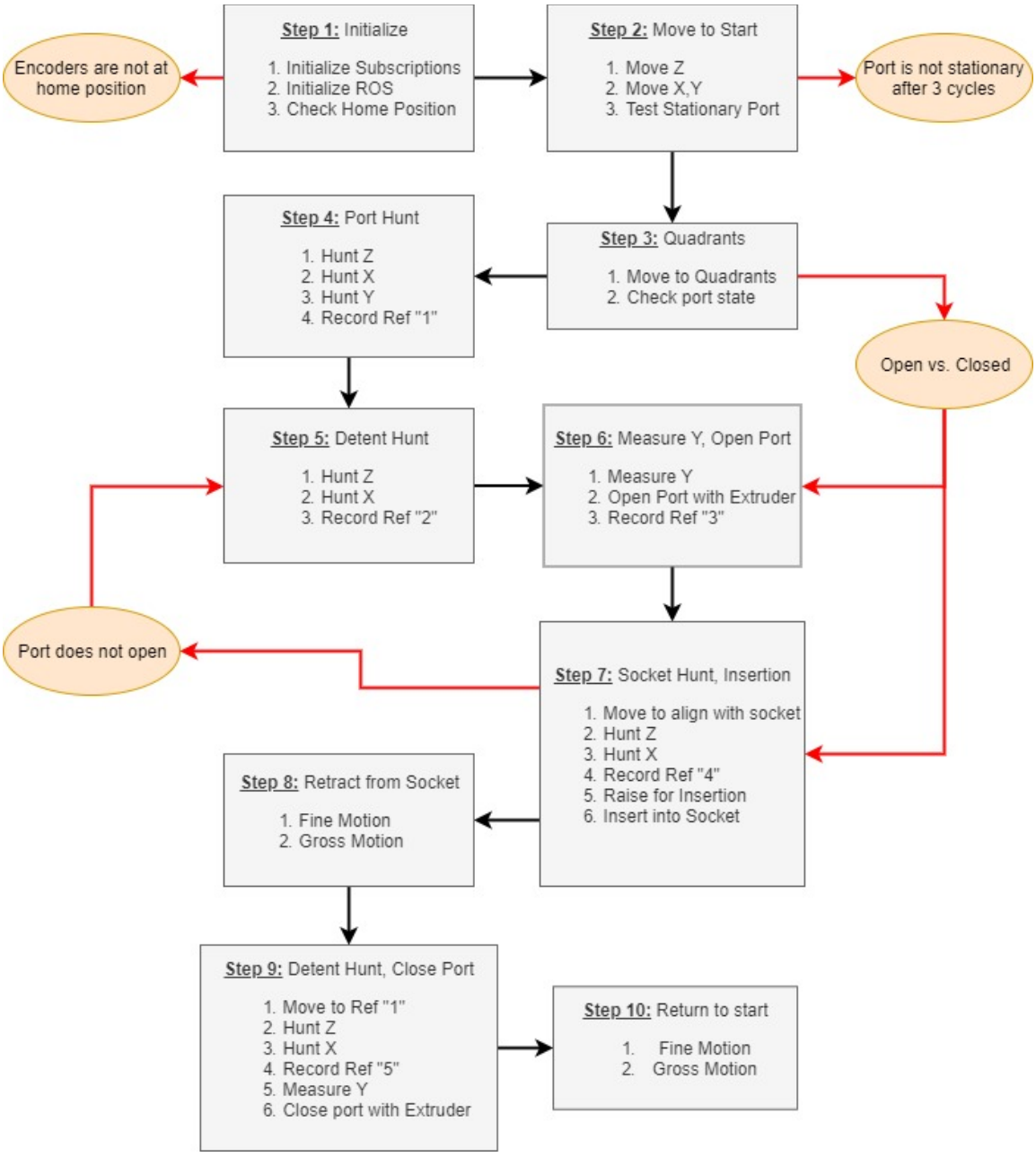

Figure 13: Full Cycle Automation Flowchart

3.5 Automation Algorithm Overview

The logic of the robot was written modularly so each separate action was a unique function. The automation routine of the system was composed of ten sequential steps that call upon individual functions when their operation is needed. Figure 4 shows a visual

layout of the automation logic, but each step will be explicitly detailed in the sections below. Prior to the operation of the automation routine, the network connection for ROS needed to be initialized on the Raspberry Pi and after the network was initialized, the vision code was started on the Jetson. The vision system was trained to separately identify

- the charging port cover,
- the Detent used to open the charging port, and
- the socket in which the plug would be inserted.

For example, if the charging port cover were identified within the field of view of the camera then the vision system would place a bounding box around the perimeter of the cover. The vision system then would communicate the dimensions of the bounding box and the coordinate of the center of the bounding box to any device on the network. The vision system was also trained to run the identification algorithm of all three features at the same time and publish the bounding box data of all the matches. An example of the output from the vision system is shown below in Figure 5. Within this figure, the bounding box labeled “fuel door release” refers to the Detent and the bounding box labeled “open_circular” refers to the charging port cover. The vision system can identify when the charging port is closed or opened. If the charging port was closed, the “open_circular” tag will change to “closed_circular”.

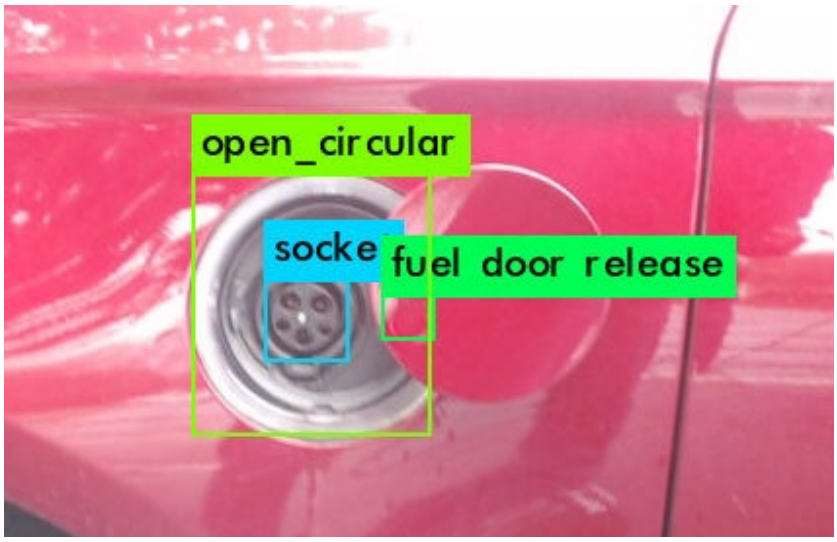

Figure 14: Example of Vision System match

The first step of the automation routine confirms that the vision system is publishing bounding box data. The specific operations of the vision system are discussed in Yosuke Yajima's Master's Thesis (Yajima 2020). Additionally, it ensures that the robot is in the home position based off the encoder readings. Figure 6 below depicts the geometry of the robot along with the coordinate system that is used for our algorithms. The measurements are in millimeters and this figure depicts the configuration for the robot when it is in the home position. If there are any issues with confirming vision system publication or that the robot is at its home position the algorithm will result in an error state and the robot will not move further through the automation routine.

The second step moves the robot to its initial starting position. This motion is executed using the encoders for closed loop feedback but all movement after this step uses the vision system for closed loop feedback to decrease the computational load required by the encoder feedback.

The third step uses the camera that is mounted to the support tower of the robot to determine the initial location of the fender within the operational window. The robot is then moved to one of five hunting locations based off the feedback from the tower camera. This ensures that the charging port is within the Field of View (FOV) of the end effector camera when the hunting sequence starts in step four.

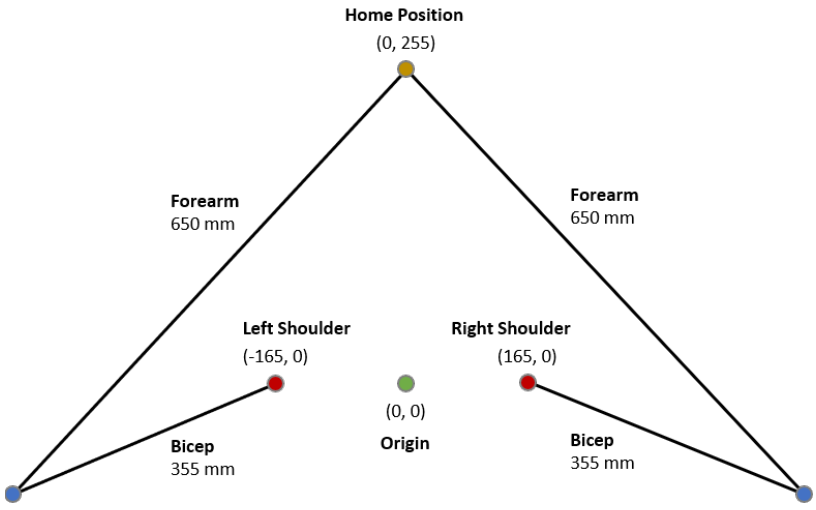

Figure 15: Geometry and Coordinate System for Robot

Referring to the vison system operation, the camera feeds are downscaled to a resolution window of 640 pixels by 480 pixels. Looking at this window, the vision algorithms use the upper left-hand corner as the origin of its coordinate system with the positive X direction moving to the right and the positive Y direction moving down to the bottom of the window. The end effector camera is mounted in the center of the end effector and during a hunting cycle, the robot incrementally moves so that the center of a bounding box, referred to as the BB Centroid from here on, from the vision system is within a defined tolerance of the center of the window (320, 240), which will be referred to as the FOV Centroid from here on.

Step four conducts a hunting cycle based off the bounding box of the charging port cover. The hunting cycle centers the bounding box for the robot's Z-axis and then the X axis independently. After the BB Centroid is aligned with the FOV Centroid, the system then incrementally moves forward until the base and height of the bounding box of the charging port cover reach a defined size. At this point, the end effector will be between 250 and 300 mm from the charging port cover. At this distance, the vision system will consistently detect the Detent on the charging port cover.

In step five the algorithm will repeat the hunting cycle on the Z and X axes based exclusively off the bounding box for the Detent. The only other difference between step four and step five is that the algorithm does not do a vision-based depth measurement.

For step six, the system measures the exact Y distance to the center of the charging port Detent. The robot initially drops 51 millimeters, which is the distance from the center of the end effector camera lens and the center of the linear actuator on the end effector. The linear actuator is then fully extended, and the system moves forward on the Y axis one millimeter at time until it registers contact on the touch sensor. The total distance moved during this step is then stored as a reference coordinate and the actuator is then retracted. The system then moves forward an additional 35 millimeters and fully extends the actuator to engage the locking mechanism on the charging port cover. When the actuator is retracted, the spring mechanism disengages, and the cover opens.

For the seventh step, the system moves to align with the charging socket and performs another hunting cycle to center itself. This hunting cycle is different than the previous two hunting cycles. Whereas the previous hunting cycles used the neural network

to locate the charging port cover and the Detent, this hunting cycle uses both a neural network that is trained on the socket as well as a Hough Circle Transformation. The neural network alone does not provide a match that is accurate enough for the system to consistently insert the plug into the socket. On the other hand, the Hough Circle Transformation provides the accuracy that our system needs to operate, but the geometry of the socket has far too many circles and circular features that are all detected by the algorithm. Thus, the neural network is used as a filter the Hough Circle Transformation. Our algorithm uses a circle that has a centroid that is ±4 pixels from the centroid of the neural network fit (Yajima 2020). The system translates +7.5 centimeters on the Z-axis after the hunting cycle is complete, which is the distance between the camera lens and the charging plug.

The eighth, ninth, and tenth steps involve reversing the plug from the socket, closing the charging port cover, and retracting the system back to its home position. The reversing is straightforward, with the system reversing 90 millimeters to disengage the plug from the socket and an additional 100 millimeters so the Detent of the charging port cover is within the field of view of the camera. The closing of the charging port cover is executed in the same manner as the opening. The Detent hunt is repeated to record the position again. The reason that this step is repeated instead of using the recorded position from step six is due to the charging port cover hinge axis not being parallel to the Y-axis of the robot. The Detent translates in the positive Z, the negative Y, and the positive X directions during opening which necessitates a separate hunting cycle to close properly. The ninth step is complete after the charging port cover is confirmed closed, after which the arms retract and

lower to their home position. At this point the automation has completed and the systems has reset itself for operation for the next vehicle in need of charging.

3.6 Flexible Coupler

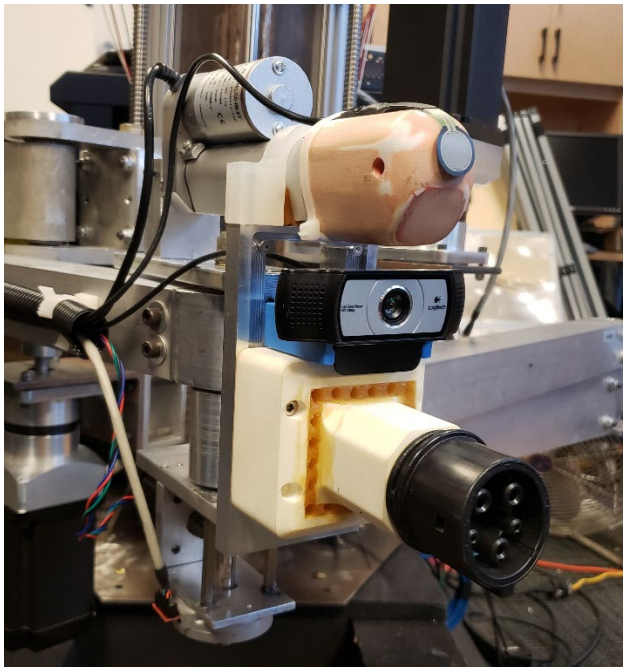

Figure 16: View of the front of the End Effector

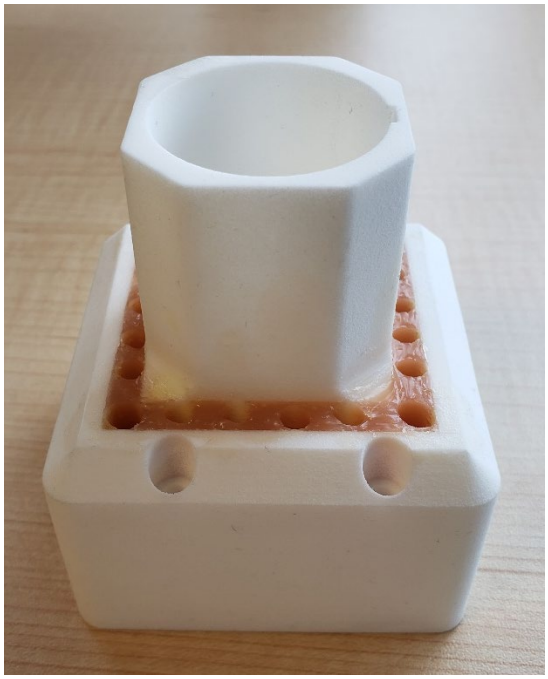

Figure 17: 3D Printed Flexible Coupler

The SAE J1772 (SAE 2017) plug is located on the bottom of the end effector as shown in Figure 7 above and the plug housing is shown in Figure 8. The plug is mounted in a three-piece, 3D printed and cast housing that gives the plug both compliance and flexibility during the insertion process. This compliance reduces the required system accuracy. The interior and exterior housing are SLS 3D printed using a specialized nylon powder made for SLS. The housings are spaced by 2.5 millimeters in all directions and are held in place. Within this void, a Shore 15 urethane is cast to allow for deflection in all directions. During the seventh step of the automation routine when insertion occurs, the system increments forward 90 millimeters with step of one millimeter with a delay of .25 seconds between steps. This repetition of motion and delay effectively vibrates the

interface between the plug and the socket, which eases the insertion by overcoming the friction and any possible binding within the interface due to misalignment.

3.7 Code Exceptions

During initial software development, three error conditions were identified as potential failure modes that needed resolution. The three error conditions that were identified early on are

- the “coordinate limit”,
- the “home position”, and
- the “port initial x position” conditions.

The “coordinate limit” condition was one of the first conditions that was implemented. This was set as a limit in the software that was based off the geometry of the robot. The “home position” error state was implemented primarily due to issues with the first firmware that was written for our encoders. Occasionally during startup, the firmware would misinterpret the encoder values and would offset the encoders by approximately 60 degrees. When the automation algorithm was started, the robot moved from home position to a starting coordinate. With the erroneous offset, the robot would think that the end effector was positioned in front of the starting coordinate instead of behind it and would actuate the motors in reverse, resulting in the motors grounding the arms into the frame. To resolve this issue, the encoder values of the system at home was recorded and are referenced during initialization. If the initial encoder values are not within 4 degrees of the reference values, the automation cycle is stopped, and the condition is flagged.

The completed system functioned consistently with the exceptions implemented during indoor verification tests, but errors started occurring with increasing frequency when performance tests started on a full-sized vehicle outdoors. The transition from the laboratory to the intended operating environment brought additional variables into the system that had not been accounted for initially. During testing, multiple error conditions were identified that would have substantial impact on the robustness of the system that needed to be resolved. These error conditions were resolved by writing code exceptions to allow for proper functionality in the operating environment. These exceptions would only be triggered if the identified error state were met and needed resolution.

The first of the error conditions are driven by operating environment negatively interacting with the system. Those failure modes and their respective error conditions are “port is not stationary”, “initial port state”, and “vision match lost during hunt”. The “port is not stationary” condition is driven by a failure mode of the automated charging cycle being started before the user has parked the car. This condition is resolved by the automation cycle comparing the centroid of charging port match with another centroid position taken 4 seconds later. If ΔX between those centroids is less than 7 pixels, then the centroid is stationary, and the automation can proceed. If not, another comparison is made with the same exit criteria. If the third comparison fails, then the automation algorithm is stopped, and an error message is generated that states that the user must make the vehicle stationary before an automated cycle can proceed.

The “initial port state” accounts for a user that opens the charging port before starting the automated cycle. Using the neural network training, the vision system can identify the charging port when it is opened and closed and can differentiate between the

two. The initial charging port state is recorded, and the automation cycle proceeds normally until it records the distance to the fender. If the charging port was initially closed, then the automation proceeds with no change. If the charging port was initially open, then the cycle skips the charging port opening step and it adds an offset of +25 mm during the insertion operation to account for the Y translation of the charging port when it opens. The "vision match lost during hunt" condition operates simply by pausing all motion of the robot if the vision match is lost prior to the insertion step. The motion resumes when the match is reacquired. The last condition that accounts for external interaction is the "angle extreme" error condition. For the robot to work properly, the plane of the charging port must be within 10 degrees of parallel to the X-axis of the robot. The vision system performs a comparison between the height and width of the bounding box to determine a rough approximation of the angle. When viewing the charging port while perfectly perpendicular, the bounding box is very close to being square. When viewing from perpendicular on the Z axis but at an angle on the X axis, the width of the bounding box will decrease but the height will stay the same. By comparing the discrepancy between the width and the height, an approximation of the viewing angle can be calculated.

Though the mechanical system was made as precisely as possible and provides a great deal of consistent and reliable performance, design oversights early on created errors that cannot be resolved without substantial redesign or rework. These errors primarily result in miniscule amounts of backlash and design oversights making it impractical to mechanically compensate for the backlash. Along with motion scaling adjustments made during the hunt cycles, this backlash is a direct cause of the two remaining failure modes. The first of these failure modes is an infinite loop in the hunt loops. To reiterate the function

of the hunting procedures, the robot incrementally moves during a hunting cycle so that the BB centroid from the vision system is within a defined tolerance of the FOV centroid. When the delta between the BB Centroid and the FOV Centroid gets below 15 pixels or so, a condition would occasionally arise where the motion of the robot during the hunt would overshoot the center of the field of view and the hunt would jump from one side of the FOV Centroid to the other with no convergence. This was initially resolved by implementing a counter that logged the number of hunt cycles that were executed and would halve the incremental distance if the counter exceeded 15.

Additionally, the wrist of the robot had to be disabled once the counter threshold was reached. During X and Y motion, the angle of the wrist is adjusted so the final position of the end effector remains perpendicular to the XZ-plane of the robot. During the very minute motions that occur when the increments are reduced, the wrist would need to move a fraction of a step to maintain perpendicular, but the stepper driver would default to a single step. This resulted in the wrist continuously drifting and the hunt never converging. As a result, the “infinite hunt loops” condition operates by halving the step size and disabling the wrist once the cycle counter reaches 15. Though it is implemented in both the X and Z hunts; the failure mode was only seen in the X-axis hunt.

The last error condition is the “port is not opened” after the opening cycle was completed. This is implemented by using the vision system to confirm that the charging port was opened after Step 6. If the charging port is opened, then the cycle continues uninterrupted. If the charging port was not opened, then the robot returns to the reference coordinate that was recorded prior to the Detent hunt in step 5 and then steps 5 and 6 are repeated. The robot can fail the confirmation three times before the automation cycle is

stopped and the robot returns to the home position. The user is then told that the automation was unable to be completed and that the cycle needs to be restarted.

CHAPTER 4. EXPERIMENT METHODS AND TEST RESULTS

4.1 Purpose of Experiment

At this stage in our development, the structural and electro-mechanical issues of the robot had been resolved and the software was functioning reliably. However, the software had not yet been tuned. The customer consistently voiced his concerns about the run time of one charging cycle and was insistent that the cycle time needed to be reduced. The code was audited to determine where operating time could be reduced and about a minute of the operating time was eliminated by removing numerous vestigial delay statements, but the whole cycle was still taking around 10 minutes to run.

Referencing Figure 4, the run time of Steps 1, 3, 8, and 10 could not be reduced any further but there was opportunity in reducing run time of the Port, Detent, and Socket hunt cycles. Each hunt cycle operates by calculating a ΔX or a ΔY between the vision match centroid and the center point of the field of view. The cycle will continue until the ΔX or ΔY drops below a predefined threshold. When the code was originally written, we defined the threshold to be as accurate as possible to give the system the best chances for a successful insertion of the plug and that threshold was set at 2 pixels for all hunts. With the increase in reliability that came with the robot and the software improvements implemented since our initial testing, we were confident that we could trade accuracy for a decrease in operating time with minimal impact to system performance.

4.2 Experiment Hypothesis

There were three hypotheses for this experiment that discuss the expected results of the experiment and where those results are expected to occur. The hypotheses are listed below:

1. The operating time of the automation algorithm could be reduced without a detrimental impact to the system performance.
2. The reduction in operating time will come from the accuracy improvements of the X and the Z hunts.
3. The highest accuracy criteria will still be necessary for repeat functionality during the Detent hunt, but low accuracy of the Port hunt will have no effect on the success of the hunting cycles.

The second hypothesis is inspired from testing the robot during development, the changes in operating time are expected to be seen from the hunt cycles that are being directly manipulated in the experiment. There is the possibility that changes will also be seen in the Y hunt due to the repeated X hunt at the end of the cycle but that is not expected to be any greater than the primary X hunt. The third hypothesis is also inspired from testing during development. During development, we needed the system to be as accurate as possible in order to find the detent, so it was expected that the highest accuracy was needed for repeat functionality.

4.3 Experiment Procedure

The charging port hunt and the Detent hunt were the subjects of this experiment and were recorded to accept tunable parameters. The socket hunt and the Y-hunts were not subjected for testing in this experiment for the following reasons. The socket hunt has

consistently been one the most significant failure modes of the system operation due to its sensitivity to proper placement at the highest accuracy threshold so decreasing its accuracy would have a direct and negative impact to system performance. The Y-hunt operates differently than the X-hunt and Z-hunt because it compares the size of the bounding box to a threshold instead of the centroid (Yajima 2020). This comparison is by nature more inaccurate given that the bounding box is dynamically placed on each frame of the video stream with a significantly higher standard deviation than that of the centroid placement. Any adjustment to the current threshold would also have a detrimental impact to the system performance.

The X-axis and Z-axis thresholds were paired for each hunt during this test, so within a hunting cycle the thresholds for each axis were identical. For the charging port hunt, the algorithm exit thresholds are shown below in Table 1. It is important to note that these values are in terms of pixels used for comparison to the vision feed, and these values are not equal to a length in millimeters. Because of this, a scaling factor is implemented in the algorithm for each hunt to allow for convergence on the centroid. For the experiment, every combinations of the accuracy criterion in Table 1 was run a total of ten times. During this experiment, each run was conducted based off an accuracy test code where the first digit represented the Port hunt accuracy and the second digit represented the Detent hunt accuracy. For example, the first accuracy criterion that was tested used a test code of “11” which corresponded to a Port threshold of two pixels and a Detent threshold of five pixels.

Table 1: Accuracy Thresholds used for DOE in term of video pixels

Accuracy Threshold (pixels)		
	Port	Detent
1	2	5
2	6	12
3	10	20

Because the socket hunt cycle was omitted, the experiment executed only steps 1 through 5 as well as step 10 of the automation algorithm that was shown in Figure 4. The modified routine that was used for the experiment is shown below in Figure 9. A Blackbox routine was written within the automation algorithm that collected data from each step of the algorithm. The Blackbox recorded runtimes of each step and sub-step in the automation algorithm, reference coordinates and their corresponding encoder angle, cycle counts for the hunting cycles, and error counts for some of the exception cases. All the data used for the analysis in this experiment came from the Blackbox files that were written at the completion of a test cycle.

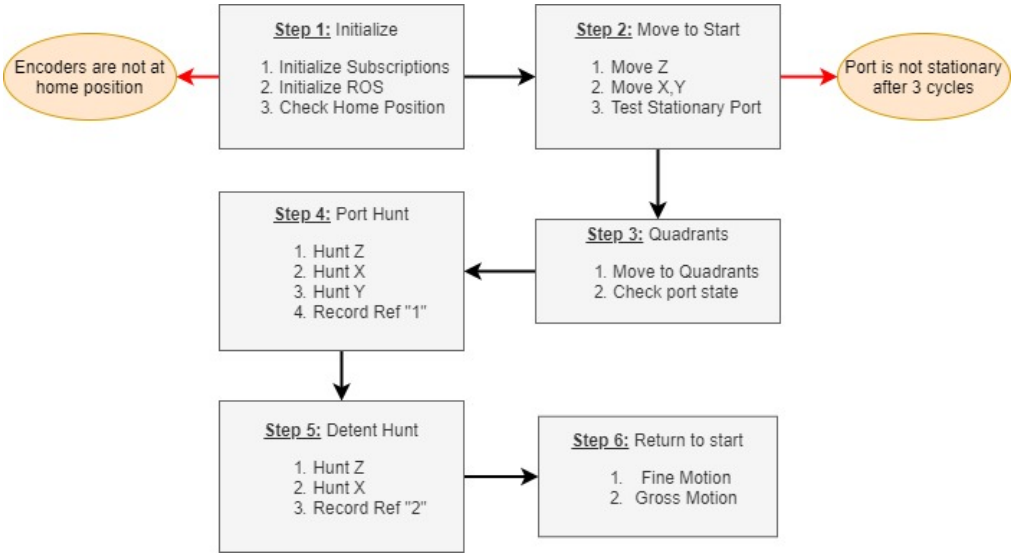

Figure 18: Shortened Automation routine used for Experiment

Additionally, each test was manually marked as “Pass” or “Fail” depending on whether the center of the touch probe was able to accurately make contact within the boundaries of the Detent. Figure 10 shows the alignment of the extruder with the Detent prior to actuation and what was considered successful contact with the Detent that resulted in a Pass. Throughout our testing for the last year, we determined that if the touch probe contacted the Detent during the measurement stage, it was 99% successful in opening the charging port. The rare failures resulted from the wrist incorrectly rotating the end effector within 5 degrees of perpendicular. In these instances, the touch probe would slip out of the Detent without opening the Port. The experiment involved 10 test runs of each pair of accuracies for the X hunt and the Detent hunt. There were three accuracies for each hunt which meant that the full experiment required a minimum of 90 test cycles to complete. Ideally a higher number of runs would be completed to obtain a more statistically relevant sample size, but the sample set was reduced due to time constraints.

Figure 19: Alignment and “Passing” contact between Extruder and Detent

For the development purposes of the robot, the project sponsor provided us with three different fenders; all of them were the front driver quarter panel and they were red, blue, and champagne. They were to be used for training the vision system as well as full

system testing. For our test we used the red fender and it was mounted on a wooden fixture that held it at the same orientation and height that it would be on a real vehicle. This allowed us to preform our tests indoors in a controlled testing environment.

4.4 Experiment Set-up

For this experiment, the base of the red test fender was placed parallel to the front of the robot. Figure 11 depicts the layout that was used for this experiment and all the coordinates are based off the standard coordinate system shown in Figure 6. The X and Y axes of the robot define this plane and the Z-axis is controlled by the linear actuators that translate the entire parallel mechanism up and down. The fender was positioned where the Detent was at a coordinate of (240, 495) millimeters. This position was chosen to require significant utilization of the hunting cycles, which is the subject of this test. Additionally, the diameter of the Detent was measured at 25 millimeters.

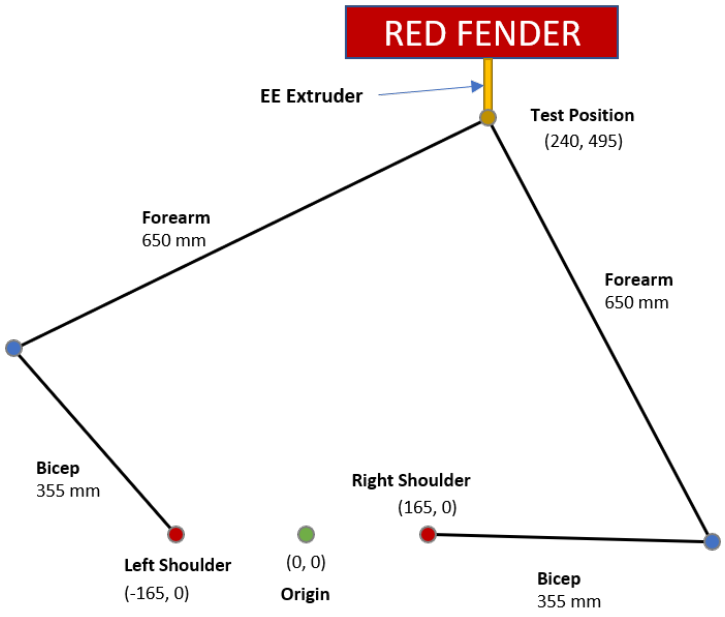

Figure 20: Experiment Set-up with Robot Geometry

4.5 Experiment Data and Analysis

4.5.1 Data Discussion

The system accuracy was analyzed based off the encoder angles instead of the positions recorded in the Blackbox. The firmware that was written for the encoder's samples were too low of a frequency to be used for continuous feedback, so it was not able to be implemented for real-time feedback for the entire operation. The encoders are used only during the initial operations of the automation cycle before the vision system starts providing positional feedback. Because of this we implemented two separate movement functions in the code; one that uses the encoders and the other that uses the vision system for positional feedback and accuracy.

The encoders that are used on the robot are true absolute, rotary induction encoders. The model that was selected has a is a 12-bit resolution and has 4,096 steps per a full revolution (Zettlex 2020:26). That means that the encoders can read motions greater than .0879˚. Also, the backlash of the gearbox coupled with the NEMA 34 is specified by the manufacturer at less than .417˚, but each unit was never tested for specific backlash (SureGear 2020). This robot was never designed with backlash compensation but was designed and manufactured to minimize backlash as much as possible. The first two encoder recordings were taken during the home position initialization and the initial Z-axis traversal. These two steps in the operation use the encoders for feedback and are the most accurate motions during the entire process. Therefore, the encoder values from these motions are used as the baseline in this analysis. Additionally, these motions occur at the initialization of the system and were not affected by the tuning parameters from the

experiment, so the data from these motions is taken from 90 runs and is shown below in Table 2. The purpose of this baseline data was to validate the system accuracy and consistency.

Table 2: Encoder Positions for Baseline Analysis (Values are in degrees)

		Left	Right	Wrist
Home Position (0, 255, 0) In millimeters	Avg.	288.219	-80.173	134.672
	St. Dev.	0.506	0.550	3.908
	Max	288.558	-79.589	146.942
	Min	286.976	-81.083	122.684
Initial Z-traversal (0, 255, 100) In millimeters	Avg.	287.259	-79.548	135.490
	St. Dev.	0.346	0.429	2.654
	Max	287.620	-78.271	141.727
	Min	286.214	-80.292	131.473

The home position is at a (X, Y, Z) coordinate of (0, 255, 0) and the coordinate after the initial Z-traversal is (0, 255, 100). The encoder position at home is also taken right after the system powers up and no motion has occurred, so the data at this position is better suited for determining how well the system was reset and not how accurate the motion system is. The encoder data after arriving at (0, 255, 100) provides a clear example of the best accuracy that this system can offer. With a minimum population standard deviation of 0.346˚ at start, which results in a translational standard deviation of 0.47 millimeters. Table 3 shows the translational accuracy based off the minimum standard deviation of 0.346˚.

Table 3: Translational Accuracy

	Y (mm)	X (mm)
X drift with .346˚ offset	255	0.473
	500	2.220
	812	3.503

After the robot moves to the starting hunt position, the encoders are no longer used for feedback, with exception to the wrist encoder to maintain perpendicularity to the target. Therefore, the encoder data from the hunt cycles is going to be used to evaluate the consistency of each hunt and is not valuable for determining the positional accuracy of the algorithm. This is primarily due to the backlash of the system along with the backlash of the end effector drive. The miter gearset that is used for motion transfer in the end effector has around 2˚ to 3˚ of backlash. Given that the position of the end effector determines the projected position of the vision system's field of view, 2˚ to 3˚ of backlash can equate to 10 to 20 mm of translational displacement depending on how far the camera is from the target. Therefore, the analysis of the performance of the system during the hunting cycles is going to be conducted based off the hunt counts, the operating times for the cycles, and the percentage of successful runs.

4.5.2 Data Analysis

The mean and standard deviation of the hunt counts are shown below in Table 4 and Table 5, respectively. The column directly to the right of the tuning parameter code is the success percentage of the ten runs at that parameter set. For clarification, the Port hunts have two additional steps that are not in the Detent hunt: Port YY and Port YX. The Port YY counts are from the Y-axis hunt cycle which is based off the scaling of the bounding

box within the field of view. The Port YX counts are from the X-axis hunting cycle that happens after the Y-axis hunt is complete and makes sure that the target is still centered in the field of view.

Table 4: Experiment Data – Mean of Hunting Cycle Counts

			Hunt Counts - Mean					
			Port Z	Port X	Port YY	Port YX	Det Z	Det X
Tuning Parameters	11	90%	3.70	9.20	13.30	14.40	7.90	7.30
	12	100%	4.10	9.40	12.40	13.50	4.10	5.20
	13	70%	4.10	8.60	18.50	16.20	1.40	4.80
	21	80%	3.50	8.20	4.90	14.20	8.40	7.10
	22	30%	4.64	8.45	6.73	16.64	3.82	4.73
	23	40%	3.30	8.30	6.80	17.60	2.20	4.20
	31	80%	3.00	7.00	3.36	15.64	8.64	6.45
	32	90%	3.00	7.09	2.18	10.73	3.00	5.09
	33	50%	3.00	7.20	1.40	8.30	1.20	4.60

Table 5: Experiment Data– Standard Deviation of Hunting Cycle Counts

			Hunt Counts - Standard Deviation					
			Port Z	Port X	Port YY	Port YX	Det Z	Det X
Tuning Parameters	11	90%	0.48	0.79	8.97	15.67	0.88	0.82
	12	100%	0.57	0.84	8.50	6.52	1.37	0.42
	13	70%	0.32	2.01	3.60	7.02	0.84	0.42
	21	80%	0.53	1.62	3.48	7.50	1.51	1.10
	22	30%	3.23	0.69	4.88	11.90	1.60	0.79
	23	40%	0.48	0.67	4.57	8.90	1.03	0.79
	31	80%	0.00	0.89	3.47	8.31	1.75	0.82
	32	90%	0.00	0.83	1.66	5.59	1.67	0.70
	33	50%	0.00	0.79	1.26	3.86	0.63	0.70

The first set of runs to discuss are with the hunts for the Port Accuracy 1 criteria. This was the hunt exit threshold that the robot has been using for the past year and is essentially the baseline for the hunt portion of the experiment. The Z-axis hunt took close to four hunt cycles to complete and had a standard deviation of roughly half a cycle. The X-axis hunt took around nine cycles to complete and had a standard deviation close to one cycle, with the data from the 13 runs being an outlier. One of the ten runs finished in four cycles, which decreased the mean and increased the standard deviation. With this run eliminated, the data from run 13 aligns with runs 11 and 12.

The data from the Port Accuracy 3 criteria shows a surprising improvement. In general, the variance in the number of Port hunt cycles is caused by system backlash, primarily in the end effector, and the fluctuations in the vision match on the Port. With the Port Accuracy 3 criteria, the symmetric allowance of ten pixels (see Table 1) counters the system backlash entirely for the Z-axis. This is seen with the mean of three hunt counts paired with the standard deviation of zero for all thirty Z-axis hunts with the accuracy 3 criteria, meaning that every Port Z hunt took exactly three hunt cycles to complete. The lower accuracy also decreased the hunt cycles for the X-axis. The average hunt cycles were reduced by almost two cycles. As seen below in Table 6, the reduction of two hunt cycles equated to a cycle time reduction of a little over 2 seconds.

Table 6: Experiment Data – Average Cycle Times

			Time Stamps (seconds) - Mean						
			Port Z	Port X	Port Y	Detent Z	Detent X	Detent Y	Full Cycle
Tuning Parameters	11	0.9	17.50	10.78	29.75	20.01	17.50	47.25	257.52
	12	1.0	16.19	10.87	27.69	9.44	12.01	49.77	237.52
	13	0.7	15.98	10.00	37.48	1.27	11.00	48.95	235.31
	21	0.8	15.51	9.59	20.17	21.55	16.98	48.85	245.55
	22	0.3	19.91	10.18	24.88	8.65	10.76	48.01	236.34
	23	0.4	15.91	10.04	26.07	3.80	9.39	48.82	226.39
	31	0.8	15.35	8.40	20.02	22.20	15.25	47.66	242.82
	32	0.9	15.11	8.05	13.21	6.02	11.72	50.40	216.38
	33	0.5	15.54	8.42	9.54	0.63	10.41	50.90	207.12

The accuracy 2 criteria for the Port hunts did not have favorable results and identified an error state that was fixed after the conclusion of this experiment. Most of the accuracy 2 failures occurred due Z-axis inaccuracies instead of X-axis inaccuracies which were the primary failure modes during all other hunts. Looking at the 23 data for the Port Z hunt in Table 4, the standard deviation is three and a quarter cycle. This is over 6 times higher than the standard deviation of all other Port Z hunts, excluding the zero values standard deviations from the accuracy 3 data. The failure from the accuracy 2 criteria was due to an adverse interaction between the scaling factor and the exit threshold for the Z-axis hunts.

Another interesting phenomenon occurred during the testing of the Port hunt cycles that proved to be advantageous. In the initial hypothesis we did not believe that the Y-axis hunt would be affected during the Port hunts. This was because the exit threshold for the

Z-axis and X-axis hunts are based on the centroid of the vision match, but the exit threshold for the Y-axis hunt is based on the size of the bounding box within the field of view. There should not be a relationship between the centroid of the bounding box and its size, but the hunt counts in Table 4 say otherwise. By looking deeper into the workings of the vison match algorithm the cause of this trend was found (Yajima 2020).

As previously stated, the Y-hunt functions by incrementing the robot forward along the Y-axis until the size of the bounding box exceeds a size threshold. The bounding boxes are dynamically placed on Port and Detent if they are within the field of view of the camera and if they match the criteria that the algorithm was trained to recognize. The algorithm can recognize the Detent and Port from different angles and on different colored fenders, but it cannot make a match on a partial target. The Port Y-hunt algorithm is designed to increment forward until the bounding box reaches a certain size, but the algorithm will also increment forward until it loses its vision match on the Port. The second exit state was not anticipated and is an error. It was never detected because once the vision match with the Port was lost, a stable match with the Detent was acquired and the algorithm was able to move onto the Detent hunt uninterrupted.

With the high accuracy criteria of the Port hunt, the centroid of the bounding box is within two pixels of the center of the field of view. Thus, the Y-hunt operates as intended and does not stop until the size of the bounding box reaches the threshold. This happened after roughly thirteen cycles but with a large standard deviation of around eight cycles. With the decrease in accuracy, the mean number of Y-hunt cycles also decreased. At the lowest accuracy, the average number of hunt cycles dropped to roughly two cycles with a standard deviation around one and three quarters of a cycle. The loss of the vision match

during the Y-hunt did not have any impact on the overall performance of the cycle but showed that the exit threshold for the Y-hunt may have been overkill. Within the Port hunt portion of the automation algorithm, the decrease in Y-hunt cycles had the single greatest decrease on overall cycle time which disproves the second part of the hypothesis, as stated in section 4.2.

There were only two individual hunt cycles for the Detent hunt: Z-axis and X-axis. Looking back to Table 4, the Detent Accuracy 3 criteria will not be considered for implementation because the best accuracy 3 performance was 70% success, which is unacceptable. For the X-axis hunts, the accuracy 1 criteria took an average of one and a half more hunt cycles to complete but for the Z-axis hunt, the accuracy 1 criteria took almost five more cycles to complete when compared to the accuracy 2 criteria. Though the success rates were similar, these additional cycles added almost fifteen seconds to the entire automation routine as seen in Table 6.

4.5.3 Experiment Conclusion

After analyzing the results of the experiment, my hypothesis was confirmed that it is possible to reduce the operating cycle time without an effect on system performance. Figure 12 and Table 7 below show the hunt times and standard deviations for the four data sets that had the highest success percentage. Additionally, Figure 13 shows the full cycle times for all nine data sets. The error bars on these figures represent a $\pm 1\sigma$ for the population of the data set.

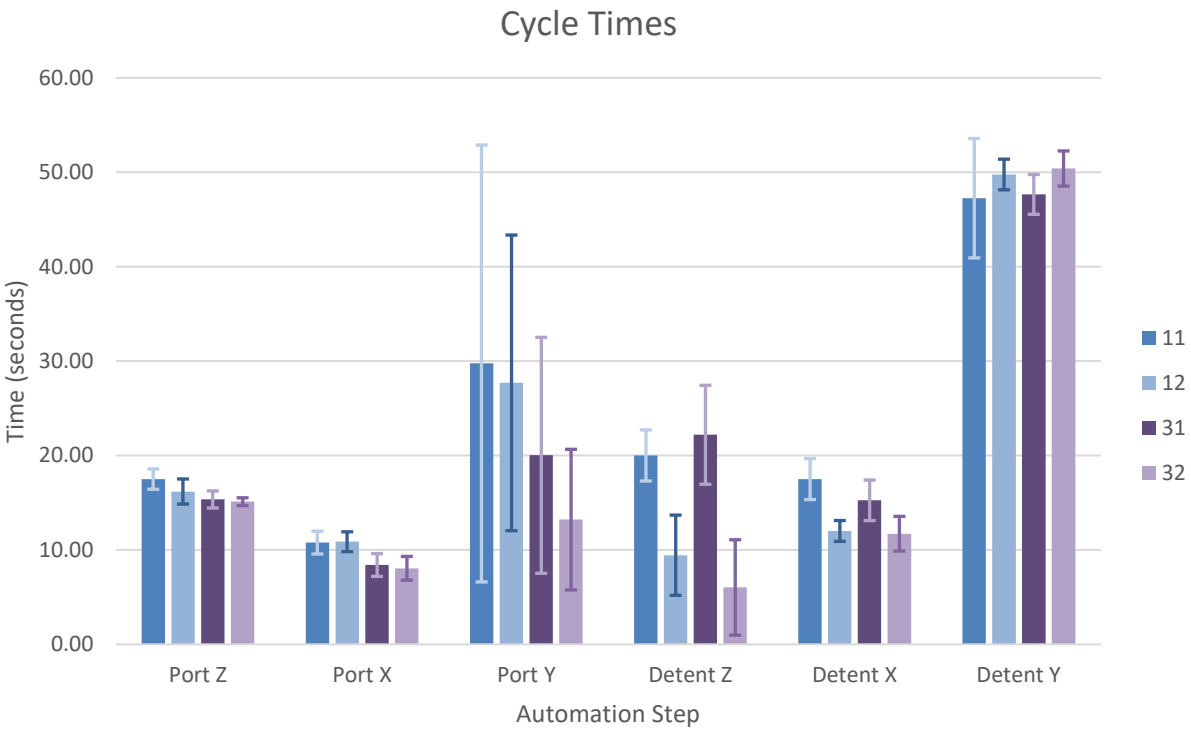

Figure 21: Automation Step Cycle times with ±1σ Error Bars

Table 7: Automation Step Cycle Times (used for Figure 12)

			Time Stamps (seconds) - Mean						
			Port Z	Port X	Port Y	Detent Z	Detent X	Detent Y	Full Cycle
Parameters	11	0.9	17.50	10.78	29.75	20.01	17.50	47.25	257.52
	12	1.0	16.19	10.87	27.69	9.44	12.01	49.77	237.52
	31	0.8	15.35	8.40	20.02	22.20	15.25	47.66	242.82
	32	0.9	15.11	8.05	13.21	6.02	11.72	50.40	216.38

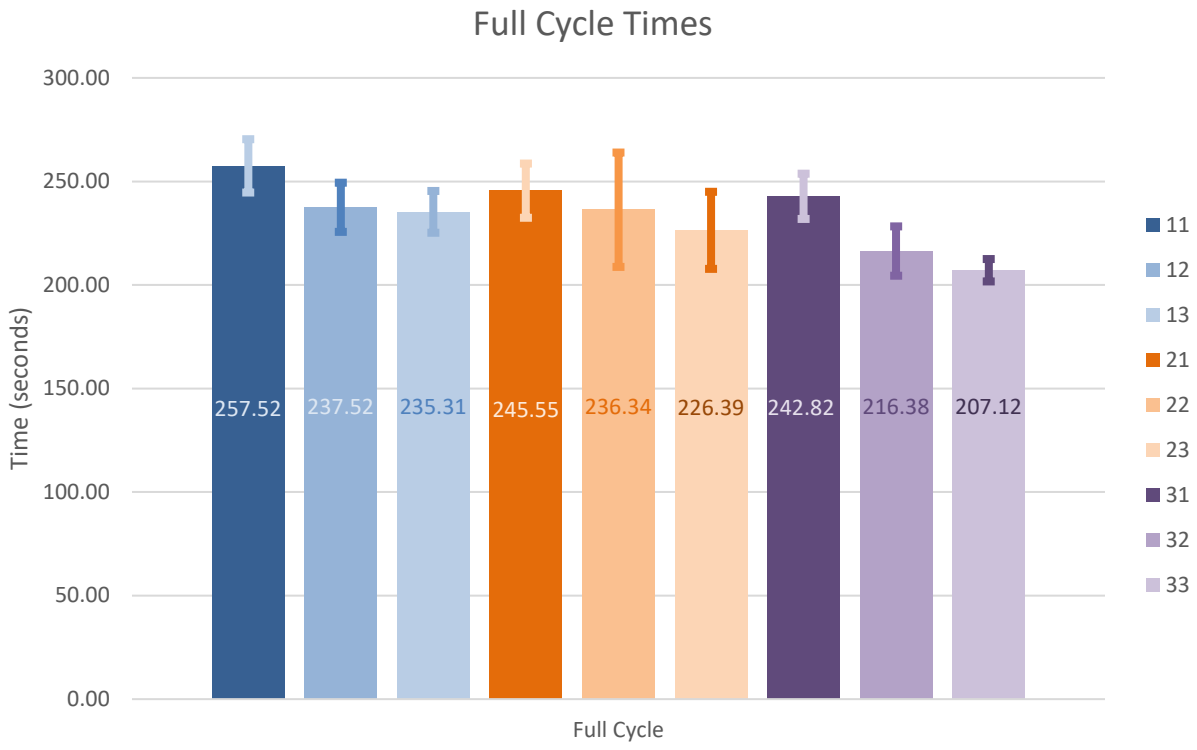

Figure 22: Average Full Cycle Times for all 9 Experimental test cycles with ±1σ Error Bars

Though the 33-accuracy criterion had the shortest operating time, the success rate was too low for implementation. The 32-accuracy criterion has a success rate of 90% and represents a reduction of 41.13 seconds in the operating time of the test cycle. This equates to a 15.97% reduction in operating time. Therefore, the accuracy criteria of the full automation cycle were adjusted to match these parameters to reduce operating time while maintaining performance. Additionally, the assumptions made in section 4.2 were verified regarding the areas of opportunity for this experiment. Referring to Table 8 below, the four sections identified in purple also coincide with Step 1, 2, 3, and 10 from Figure 13. They were initially omitted from this experiment due to minimal opportunities of improvement for operating time reduction. The mean and standard deviations of those sections are shown

in the bottom two lines of the table. The standard deviation of all 90 test runs for the three steps are all below a quarter of a second. The data from Step 10 has a higher standard deviation but this is expected given that the movement is starting from the final coordinate from the previous step.

Table 8: Validation of Initial Experiment Assumptions regarding omission of Automation Steps 1, 2, 3, and 10 from experiment

		Automation Step Runtimes (in seconds)										
					Port Hunt			Detent Hunt				
		Init	Start	Quad	Z	X	Y	Z	X	Y	Start	Full
11	0.9	2.860	40.573	6.148	17.503	10.776	29.751	20.007	17.504	47.253	42.070	257.519
12	1.0	2.834	40.508	6.070	16.188	10.870	27.694	9.438	12.008	49.770	39.703	237.521
13	0.7	2.885	40.486	5.996	15.984	10.001	37.479	1.267	11.002	48.951	39.953	235.312
21	0.8	2.811	40.538	6.001	15.506	9.592	20.165	21.548	16.981	48.852	40.250	245.550
22	0.3	2.872	40.605	6.705	19.915	10.175	24.884	8.653	10.762	48.009	41.402	236.344
23	0.4	2.758	40.600	6.453	15.914	10.043	26.068	3.802	9.392	48.819	40.825	226.390
31	0.8	2.804	40.611	6.431	15.345	8.402	20.021	22.197	15.255	47.659	40.808	242.820
32	0.9	2.828	40.331	5.955	15.107	8.053	13.210	6.023	11.716	50.399	40.766	216.385
33	0.5	2.758	40.488	6.100	15.542	8.425	9.543	0.634	10.415	50.901	41.003	207.118
	μ	2.823	40.527	6.206	16.334	9.593	23.202	10.396	12.782	48.957	40.753	233.884
	σ	0.043	0.084	0.246	1.426	0.994	8.053	8.173	2.831	1.156	0.685	14.433

4.6 Hypothesis Testing

The hypotheses for this experiment were stated in section 4.2 but they will be restated below.

1. The operating time of the automation algorithm could be reduced without a detrimental impact to the system performance.

2. The reduction in operating time will come from the accuracy improvements of the X and the Z hunts.
3. The highest accuracy criteria will still be necessary for repeat functionality during the Detent hunt, but low accuracy of the Port hunt will have no effect on the success of the hunting cycles.

The second and third hypotheses were disproven due to the trends identified during the analysis of the experiment data. The second hypothesis was disproven because the primary contributor to operating time reduction was from the Y hunt and not from the X or Z hunt, as was initially expected. The third hypothesis was also disproven because the medium accuracy condition for the Detent hunt proved to be successful, so the highest accuracy criterion for the Detent was not necessary.

The first hypothesis was proven by the trends of the experimental data, but the hypothesis was also proven through the use of a t-Test. To simplify verification testing, a Two-Sample t-Test was conducted based off of the full cycle times of the four successful tests parameters that were identified in Table 7. The 11-criterion was as the benchmark for the t-Test and the 12, 31, and 32-criterion were compared to it. The Two-Sample t-Test was chosen over the z-Test and the One-Sample t-Test because the sample size of each sampled data set was below 30 and the variance of the sample sets are not equal. The results of the t-Test are shown below in Table 8 and the calculations were executed using the Data Analysis Tools in Microsoft Excel.

Table 9: Two-Sample t-Test results for Successful Test Criterion

	11	12	31	32
Mean	257.52	237.52	242.81	217.47
Variance	418.19	303.15	132.78	143.25
Observations	10	10	10	10
Hypothesized ΔMean	---	0	0	0
df	---	18	14	15
t Stat	---	2.3546	1.9815	5.3455
P(T<=t) two-tail	---	0.0301	0.0675	0.0001
t Critical two-tail	---	2.1009	2.1448	2.1314

The t-Test was set up with the hypothesized difference between the means of the baseline and the test case equal to zero. This was chosen to prove that the test case was truly an improvement on the baseline case. The P value of the three test cases were all well below 5%, so all three test cases show a reduction in operating time that could not have occurred by chance. This therefore proves and validates the first hypothesis.

4.7 Experiment Verification Test

After the experiment was completed, the accuracy parameters were implemented into the production automation algorithm and the experiment was rerun for ten cycles with the updated parameters. The production algorithm was modified to operate only the same cycles that were used for the experiment instead of the full automation routine. The time stamps from the verification test along with the other successful test cycles are shown below in Table 8. Compared to the data from the 32-accuracy criterion from the experiment, the average full cycle time from the verification test is less than half of a second than the same cycle time from the 32-accuracy criterion in the experiment.

Table 10: Average Times for hunting cycles for Successful runs and Verification test

			Time Stamps (seconds) - Mean						
			Port Z	Port X	Port Y	Detent Z	Detent X	Detent Y	Full Cycle
Parameters	11	0.9	17.50	10.78	29.75	20.01	17.50	47.25	257.52
	12	1.0	16.19	10.87	27.69	9.44	12.01	49.77	237.52
	31	0.8	15.35	8.40	20.02	22.20	15.25	47.66	242.82
	32	0.9	15.11	8.05	13.21	6.02	11.72	50.40	216.38
	VER	0.9	14.73	9.04	14.71	9.37	11.99	43.87	215.94

4.8 Post-experiment Updates

During the experiment, observations were made that showed areas of potential performance improvement for both the software and the mechanical systems of the robot that were outside of the scope of the experiment. This experiment represented the highest volume of repetitive tests that the robot had ever undergone, so these observations were anticipated. There were three areas of improvement for both the software and the mechanical/electrical systems.

The three identified areas of improvements for the software were the infinite loop exception criteria, the firmware update for the SPI interface for the encoders, and the perpendicular adjustment code for the wrist. One of the primary failure modes and the largest contributor to the X-axis variance during the hunt cycles was due to the wrist overcompensating for perpendicular adjustment when the step size was reduced if the infinite loop exception criteria was triggered. When the infinite loop exception criteria were triggered, minute motions would be commanded by the algorithm. Due to the differences in both micro stepping and backlash between the wrist drive and the shoulder drives, the shoulder drives would accurately execute the desired motions, but the wrist could not and

would move too far. This failure mode was replicated in follow-up tests where the wrist would continue to drift. To fix this, the exception criteria was updated so that when the step size was reduced once the exception was triggered, the motion of the wrist was disabled.

As coincidence would have it, a week after the experiment was completed the GWW Electronics Shop indicated that they had updated the Arduino SPI interface board and the firmware. The specifics of the update are not known, but the changes that were made resulted in enabling the interface to communicate at 20 Hz instead of the earlier 4 Hz. This meant that real-time sampling of encoder angles was now feasible. Given the time in which this update was made available and how close it was to customer delivery, it was not feasible to implement this across the entire robot. Instead, this update was implemented for the interface to the wrist encoder. This update also made the next software update feasible, which was the perpendicular adjustment algorithm.

Prior to the implementation of this algorithm, the wrist would adjust perpendicularity at the initiation of a motion on the X and Y-axis. At the initiation of a motion, encoder readings would be taken for both shoulders and the wrist. From these encoder readings the motion of each shoulder and the wrist would be calculated and then executed. The issue that was uncovered was that sampling prior to motion would not account for inaccuracies during the motion. With the faster sampling rate that resulted from the firmware update to the SPI, we were able to read the wrist encoders after motions completed. The perpendicular adjustment algorithm was written to take advantage of these updates and worked by performing an isolated wrist encoder measurement and motion based off the current X and Y-coordinate. This algorithm was written as a single function,

so it could be used whenever we wanted additional confidence that the end effector was perpendicular to the fender.

The update to the SPI interface board also accounted for one of the three mechanical/electrical updates to the system after the experiment. The other two updates were upgrading the NEMA 17 End Effector drive motor to a NEMA 23 drive motor and changing the 3:1 miter gearset to a 2:1 miter gearset. The NEMA 23 motor supplied 125 oz-in of torque compared to the 44 oz-in supplied by the NEMA 17. The additional torque provided by the NEMA 23 meant that using a lower gear ratio would still supply higher output torque to the end effector. Due to the larger output shaft on the NEMA 23, the miter gear that was used on the NEMA 17 could not mount on the NEMA 23, so a new gearset needed to be selected. Additionally, the previous 3:1 gearset interfaced to the end effector shaft using a single rectangular key, but the keyway for the miter gear was too wide, which contributed to the backlash of the system. The opportunity was taken to remake the end effector drive shaft to minimize interface backlash. The new shaft was machined as a twin D shaft to interface with a twin set screws on the miter's hub. The upgraded End Effector drive system resulted in reducing the backlash and increasing the output torque compared to the previous system.

4.9 Full Cycle Outdoor Testing

After the implementation of the accuracy parameters from the experiment results along with the updates that were made after the experiment, the entire robot system was tested outdoors on the test vehicle that was supplied by the customer. The system was tested with a full automation cycle instead of the truncated experiment test cycle. Only five full

cycles were run for the final outdoor test. Due to time constraints, more extensive testing was not available as the customer needed for the robot to be delivered. These cycles were compared to three cycles that were run prior to the experiment. Unfortunately, due to rapidly changing updates to the system architecture, only two full cycle tests were documented that ran off the same software and hardware architecture as the experiment.

Table 11: Pre-test and Post-test Cycle Times

	Time after socket hunt (seconds)		
Date	Run #	Run Time	Avg
12-Dec-19	1	283	287
	2	291	
27-Jan-20	1	266	234
	2	211	
	3	231	
	4	239	
	5	222	

The baseline tests were run one week before the experiment, on December 19th of 2019 while the full cycle verification tests were run on January 27th, 2020. The relevant data from those tests is shown above in Table 9. The runtimes used for the comparison of these cycles was taken from the completion of the socket hunt because the full cycle data was not available for the two December tests. As seen in the table, the average run time of the January post-test runs was 53 seconds faster than the average runtime of the pre-test test runs. This represents an 18.57% decrease in runtime, which is very close to the 15.97% operating time reduction shown from the experiment. The additional improvements are due

to the improvements that were made to the end effector drive and the additional software improvements that were made prior to the January 27th tests.

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

In conclusion, the customer needs of producing a low-cost automated charging system using mostly off-the-shelf components was met. Though not as precise as current automated charging solutions that utilize industrial-grade robotic components, this system is able to successfully recharge a vehicle autonomously at a fraction of the cost. The advent of high volume, hobby-grade microelectronics allows for development of systems that can mimic the capabilities of industrial systems. This thesis exhibits the work that was taken to improve this system by reducing the overall operating time by 16% while maintaining system performance. In the process of conducting the improvement experiment, numerous shortcomings in the software were identified that further improved the system performance. Additionally, a few mechanical updates were implemented that also improved the system performance.

5.2 Future Work

Though many upgrades were done to the charging robot after the completion of the experiment, there are still many updates that can be implemented. There is the opportunity for additional improvements on the software that can be identified with other experiments. Another experiment that should be conducted would be a duplicate of the test outlined in this thesis, but the focus will be on improving the Y-axis hunt. The primary purpose of the new experiment will not be to improve the current Y-axis hunt but to better understand the phenomena that led to the cycle time reduction. By understanding this phenomena, further

improvements may be made. Also, a DOE should be conducted to better understand and model the backlash for all the axes and the end effector. Software based backlash compensation can potentially be implemented if we have a better understanding the degree of backlash in the system. Additionally, by better understanding the backlash in the system we can determine areas where mechanical backlash compensation would be beneficial if implemented.

Another improvement that can be made in the future is moving from a microcontroller and graphics processor control system to using a purpose-built Linux computer. Though this computer would be more expensive than the current microcontroller system, it would be more robust. We would be able to use any graphics system we need to allow for additional development of the vision system and with the ability to expand the system memory, we would be able to execute more processes at once with little concern about consuming too much of the system resources. Though additional testing would be necessary to determine if there are any risks to moving to this new system, there are numerous potential benefits.

The current system is one of the only automated charging systems that uses a parallel arm configuration. All the competitive systems that were identified in the literature review use serial arms, so there is probably value in using that system. Either a physical DOE or a simulation could be conducted to determine if there is any value in reconfiguring the system as a serial arm or continuing the use of the parallel arm system.

The last identified opportunity for future work is the implementation of a tactile feedback system for the compliant coupler. Right now, the compliant coupler allows the

coupler to flex within the end effector housing to prevent it from breaking if the plug and the socket are misaligned. If the compliant coupler was instrumented to allow feedback based on the degree of flex during socket and plug engagement, then the coupler could assist the entire robot in compensating for misalignment. Additionally, if this system were running continuously, it could identify if the vehicle tries to move while the plug is inserted. There is a great deal of value for the implementation of a system like this, but it would be one of the most difficult future work items to develop from the list that has been identified.

REFERENCES

Albright, Greg, Jake Edie, and Said Al-Hallaj. 2012. “A Comparison of Lead Acid to Lithium-Ion in Stationary Storage Applications Contributors :” *AllCell Technologies LLC* (March):14.

Bishop, Rollin. 2015. “A Robotic Snake Arm Is Here to Charge Your Tesla.” *Popular Mechanics*. Retrieved June 24, 2020 (https://www.popularmechanics.com/cars/hybrid-electric/a16773/tesla-robotic-charging-arm/).

Capizzo, Peter D. 2018. “US009873408B2 - Device for Refueling, Exchanging, and Charging Power Sources on Remote Controlled Vehicles, UAVs, Drones, or Any Type of Robotic Vehicle or Machine with Mobility.”

Corbett, Adrian, and Robert Maniaci. 2013. “US20130338820A1 - Automated Electric Vehicle Charging Station.”

FreeWire Technologies. 2017. “US009592742B1 - Systems, Apparatus, and Methods of Charging Electric Vehicles.”

FreeWire Technologies. 2020. “Mobi EV Charger.” Retrieved June 24, 2020 (https://freewiretech.com/products/mobi-ev/).

GADH, Rajit, Arunabh CHATTOPHADHYAY, Ching-Yen CHUNG, Peter CHU, Brahmavar PRABHU, Omar SHEIKH, and Joshua Chynoweth. 2013. “US20140203077A1 - Intelligent Electric Vehicle Charging System.”

Haddad, Joseph C., and Daniel B. Lysak. 2018. “US010106048B2 - Automated Electric Vehicle Charging System and Method.”

Kuka AG. 2020. “Charging Assistant Carla_connect Charges Electric Cars.” Retrieved June 23, 2020 (https://www.kuka.com/en-us/products/robotics-systems/kuka-ladeassistent-carla_connect).

Leary, Kevin Walter, and PowerHydrant LLC. 2016. “US009493087B2 - Method and System for Automatic Charging of Electric Vehicles.”

National Safety Council. 2020. “Preliminary Semiannual Estimates - Injury Facts.” Retrieved June 1, 2020 (https://injuryfacts.nsc.org/motor-vehicle/overview/preliminary-estimates/).

Pandilov, Zoran, and Vladimir Dukovski. 2014. “Comparison of the Characteristics between Serial and Parallel Robots.” *Acta Technica Corviniensis-Bulletin of Engineering* 7(1):143.

Pong, William, and Edward Fredkin. 2001. “US006237647B1 - Automatic Refueling Station.”

PowerHydrant LLC. 2020. “PowerHydrant - Autonomous, Conductive, Fast Charging for Electric Vehicles.” Retrieved June 24, 2020 (https://powerhydrant.com/).

SAE. 2017. *SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler J1772_201710*.

SureGear. 2020. “SureGear ® Planetary Gear Reducers for NEMA Motors-Overview.” 3.

Retrieved June 1, 2020 (https://cdn.automationdirect.com/static/specs/suregeargb.pdf).

Tsay, T. I. Jame., and C. J. Chang. 2004. “Pose Control of Mobile Manipulators with an Uncalibrated Eye-in-Hand Vision System.” Pp. 3033–38 in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 3.

Tsay, T. I. Jame., Y. F. Lai, and Y. L. Hsiao. 2010. “Material Handling of a Mobile Manipulator Using an Eye-in-Hand Vision System.” Pp. 4743–48 in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*.

U.S. Department of Energy. 2017. “Electric Vehicle Benefits | Department of Energy.” Retrieved June 1, 2020 (https://www.energy.gov/eere/electricvehicles/electric-vehicle-basics).

U.S. Department of Energy. 2019. “Timeline: History of the Electric Car | Department of Energy.”

Volkswagen. 2015. “E-SmartConnect: Volkswagen Is Conducting Research on an Automated Quick-Charging System for the next Generation of Electric Vehicles | Volkswagen Newsroom.” Retrieved June 23, 2020 (https://www.volkswagen-newsroom.com/en/press-releases/e-smartconnect-volkswagen-is-conducting-research-on-an-automated-quick-charging-system-for-the-next-generation-of-electric-vehicles-1512).

Yajima, Yosuke. 2020. “Vision Sytems for Mobility Applications.” Georgia Institute of

Technology.

Zettlex. 2020. “IncOder Product Guide Mini Range 37-58mm Inductive Angle Encoders.” 47. Retrieved (https://www.celeramotion.com/resources/product-documentation/).

Zhuang, Hanqi, and Zvi S. Roth. 2018. *Camera-Aided Robot Calibration*.