

Technology for Designing the Steering Subsystem Component of an Autonomous Vehicle

William Shaler Brown

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Mechanical Engineering

Dr. Andrew J. Kurdila, Chairman
Dept. of Mechanical Engineering

Dr. Alfred L. Wicks, Co-Chairman
Dept. of Mechanical Engineering

Dr. Steve Southward
Dept. of Mechanical Engineering

Dr. John B. Ferris
Dept. of Mechanical Engineering

08/30/2007
Danville, VA

Keywords: Autonomous Vehicles, Steering Subsystem, Steering Characterization,
Control Simulation

Technology for Designing the Steering Subsystem Component of an Autonomous Vehicle

William Shaler Brown

ABSTRACT

Autonomous vehicles offer means to complete unsafe military operations without endangering the lives of soldiers. Such solutions have fueled many efforts towards designing autonomous, or unmanned, systems. Military and academic research efforts alike continue to focus on developing these systems. While many different autonomous vehicles have been introduced, however, such complex systems have limited drive-by-wire operability. The complete process to up-fit a vehicle to fully autonomous operation involves the design, up-fit, testing and verification of many different subsystems. The objective of this thesis is to design and model an autonomous steering system requiring little modifications to an existing steering system. It is desirable to still operate the vehicle manually as well as preserve the vehicle's visual appearance. Up-fit and implementation of the designed steering system and verification of its functionality has been documented as well. Utilization of the supplied controller and software has enabled the testing and characterization of the system. The proposed design offers a solution to a wide variety of wheeled vehicles steered via the traditional and common steering wheel method. In addition, modifications have been made to an existing simulation of an unmanned vehicle in a military testbed environment (Fort Benning). The simulation accounts for the control methodology as it has been designed and tested with, which offers the ability to analyze the dynamics of the unmanned system.

Table of Contents

Index of Figures	iv
Index of Tables	v
Chapter 1 – Literature Review	1
1.1 Introduction	1
1.2 Background	2
1.2.1 Multi-vehicle Operations	2
1.2.2 Drive-by-wire Conversion	6
1.3 Contributions	7
1.3.1 Steering Subsystem.....	7
1.3.2 Modified Simulation.....	8
Chapter 2 – Ground Vehicle Platform	10
2.1 IR 810 E Club Car	10
2.2 Sensor Suite	11
2.3 Other Features of Autonomous Operation	12
Chapter 3 – Mechanical Up-fit	13
3.1 Steering Subsystem	14
3.2 Steering Characterization	19
3.3 Conclusions	33
3.3.1 Discussion of Error Dynamics	35
3.4 Further Tests	35
Chapter 4 – Single Vehicle Operation: Control Simulation	36
4.1 Equations of Motion (EOM)	36
4.2 Simulation Introduction	48
4.3 Original Model	49
4.4 Proposed Modifications	59
4.5 Results	61
4.6 Notes on Simulation	64
Chapter 5 - Conclusion	67
Appendix A	69
Appendix B	72
References	79

Index of Figures

Figure 1 (a) Components of the REDCAR Program, (b) Components of the IFP Scenario developed by ROCC at SSC San Diego, (c) ROBOCUP team competition, (d) Mathematical abstraction of robotic soccer / steal the flag in academic research [16]. Image used by permission from original creator.....	3
Figure 2 (Left) MDARS system, (right) Software interface illustrating urban operation [16]. Images used by permission from original creator.	4
Figure 3 Ingersol Rand ClubCar XRT810. Photographed by author.....	7
Figure 4 Virginia Tech's DARPA Grand Challenge Vehicle Rocky's Joystick Steering. Photograph used by permission.	15
Figure 5 Designed Steering System. Image created by author.	17
Figure 6 Actual Steering System. Photographed by author.....	19
Figure 7 Software User Interface. Image created by author.	21
Figure 8 Test Series 1	24
Figure 9 Test Series 2	27
Figure 10 Test Series 3	31
Figure 11 Linear Relationship Between Actual Position and Target Position	33
Figure 12 Difference of Linear Relationship	34
Figure 13 Vehicle Diagram. Image created by author.....	38
Figure 14 Free Body Diagram (a) Isometric View (b) Top View (c) Front View. Images created by author.....	43
Figure 15 Simulation Block Diagram. Imaged created by author.	50
Figure 16 Command Block Diagram. Imaged created by author.	51
Figure 17 Controller Block. Imaged created by author.	52
Figure 18 PseudoControl Block. Imaged created by author.	53
Figure 19 Steering Control Block. Imaged created by author.	54
Figure 20 Desired Control Force (Fc) . Imaged created by author.	55
Figure 21 Driving Control Block. Imaged created by author.	56
Figure 22 The Actuator Block. Imaged created by author.....	57
Figure 23 Unmanned Ground Vehicle (UGV) Dynamics. Imaged created by author....	58
Figure 24 Modified Controller Block. Imaged created by author.	60
Figure 25 Modified Actuator Block. Imaged created by author.	60
Figure 26 Driving Block inside Actuator. Imaged created by author.....	61
Figure 27 Annotated Fort Benning in Simulation. Imaged created by author.....	62
Figure 28 Annotated Tracking Diagram of Vehicles Traversed Path Upon Reaching its Target Location. Imaged created by author.	63

Index of Tables

Table 1 Data Table of Test Series 2.....	28
Table 2 Data Table of Test Series 2.....	32

Chapter 1 – Literature Review

1.1 Introduction

Autonomous vehicles offer an alternative solution to accomplish unknown and potentially extremely dangerous missions in the military without putting lives at risk. Additionally, the diverse missions that autonomous vehicles can perform enable them to complete tasks that may otherwise be impossible for a person. These tasks include reconnaissance or transportation of goods, for example. Research in academia, government and private laboratories is booming in the field of autonomous systems. The pool of research literature treating unmanned systems is voluminous. It is certainly far too vast a body of literature to summarize in detail in this Masters Thesis. In fact, survey articles now give good overviews of the progress in many of the sub-fields comprising the research area known as “unmanned systems.” Progress in the development of unmanned ground vehicles can be found in [1] and [2] and [3]. Similar summaries outline significant developments and breakthroughs for autonomous air vehicles in [4] and [5]. Marine vehicles with autonomous capability are reviewed in [6] and [7].

Of all the sub-disciplines summarized above, the following two areas most closely relate to the program of research in this thesis as summarized in Section **1.2**:

- (1) Multi-vehicle operations
- (2) Drive-by-wire conversion
 - (i) Steering subsystem

1.2 Background

1.2.1 Multi-vehicle Operations

Multi-vehicle operation has been extensively researched by the Air Force Research Laboratory (AFRL) Robotics Research Group, through support by the Force Protection Battlelab (FPB). The Remote Detection Challenge and Defense (REDCAR) system has been developed, and the AFRL and FPB wishes to employ the Joint Architecture for Unmanned Systems (JAUS) to develop heterogeneous robotic platforms comprised of the Mobile Detection, Assessment and Response System (MDARS), PackBot and SCOUT. An illustration of this program is given in Figure 1. During routine operation, this system will collaboratively assist with perimeter defense and security [8]. It is quite illuminating to compare research carried out by military laboratories with similar academic research. An area of research with rapidly growing popularity among many of the leading academic research facilities is the development of autonomous systems or robots in competition. Shown below in Figure 1 is a game of autonomous soccer that is part of a RoboCup competition. This is a common competition among colleges and universities. The contest consists of autonomous systems (4 legged or bipedal robots) that incorporate vision sensor inputs to recognize location on the field and target position, along with information from many motion sensors (including gyroscopes, accelerometers, etc.) to describe body position and movements, as well as a series of programmed moves to run, kick and trap a recognized ball on the field. Teams

compete and score on opponents by kicking the ball into the opposing goal while defending their own. Other academic and industrial competitions include RoboRescue, the DARPA Grand Challenge and DARPA Urban Challenge to name a few.

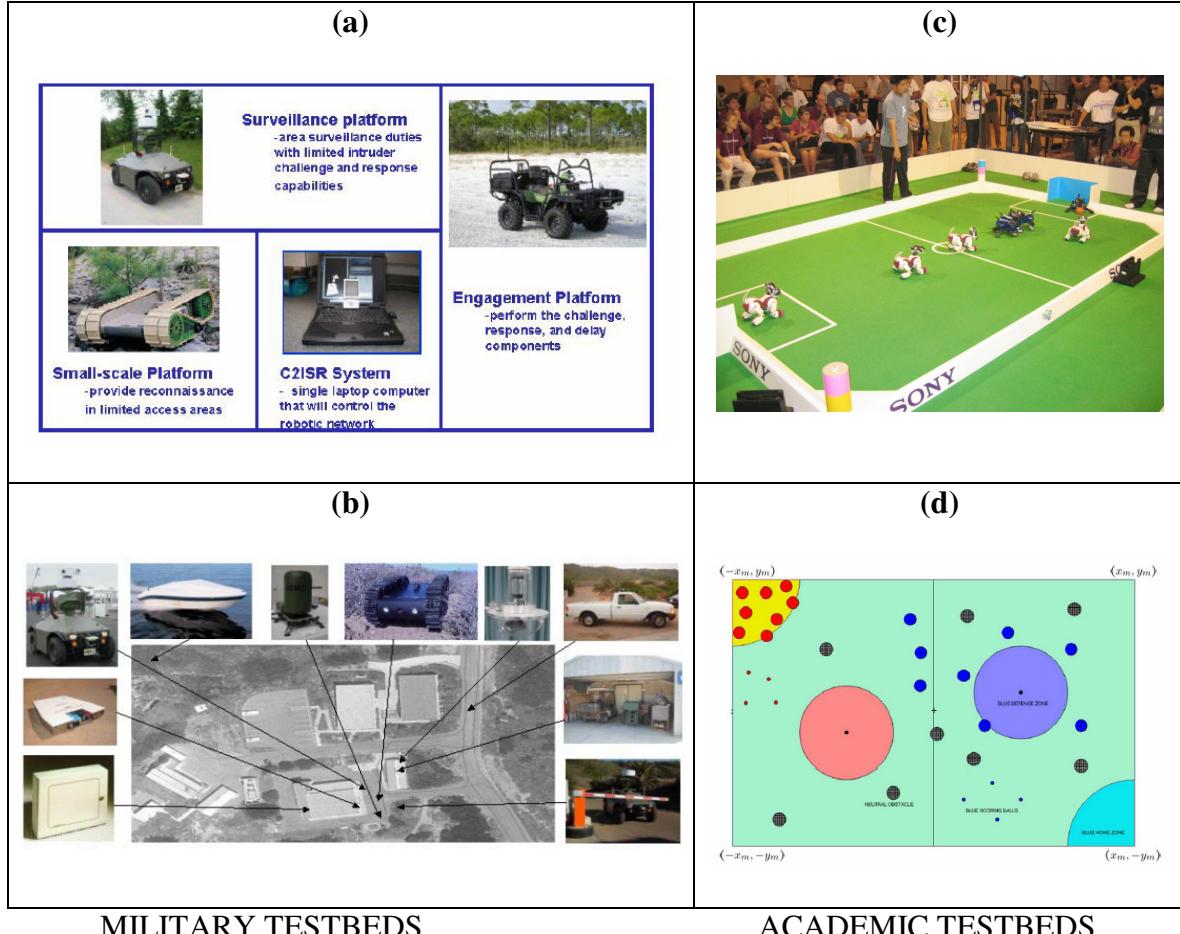


Figure 1 (a) Components of the REDCAR Program, (b) Components of the IFP Scenario developed by ROCC at SSC San Diego, (c) ROBOCUP team competition, (d) Mathematical abstraction of robotic soccer / steal the flag in academic research [16]. Image used by permission from original creator.

MDARS is the result of countless man-years of research for the Department of Defense (DoD) and commercial research and development companies (such as General Electric (GE), General Dynamics Robotics Systems (GDRS), Space and Warfare Systems Center in San Diego, US Army Aberdeen Test Center (ATC) in Maryland). These collaborative efforts have lead to the development of warehouse surveillance and intruder detection, as well as inspection of RF tagged inventory. Additionally the MDARS system has the capability of interpreting common traffic scenarios (such as intersections, railroad crossings, etc.) Figure 2 shows of the MDARS system, as well as the software interface that facilitates operation in an urban setting.

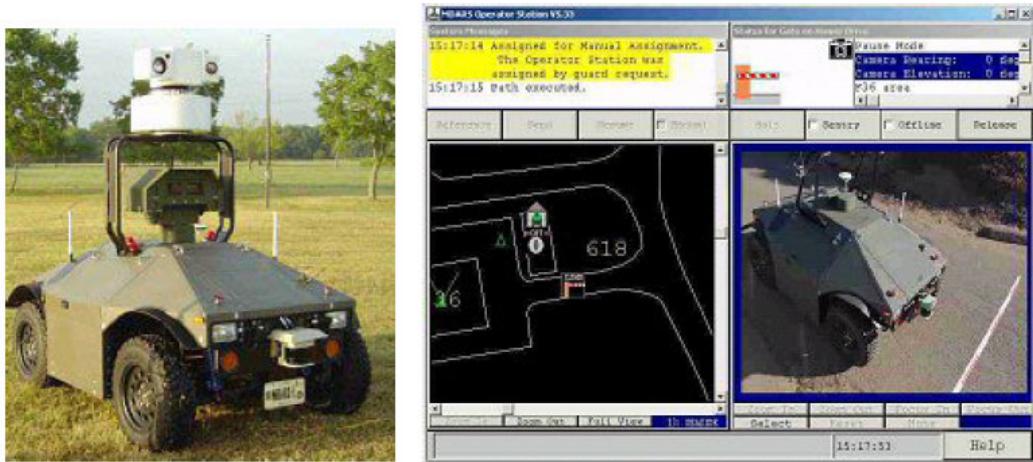


Figure 2 (Left) MDARS system, **(right)** Software interface illustrating urban operation [16]. Images used by permission from original creator.

The MDARS typifies modern developments in research at military laboratories. The primary concerns in military research laboratories include (but are not limited to) the ability to carry out highly specific concepts of operations (ConOps), reliability, interoperability, ruggedness, mobility, etc. For example, the development of

the Joint Architecture for Unmanned Systems (JAUS) is quite representative of products of research efforts at military laboratories. JAUS has been described as an open architecture over the domain of unmanned systems that meets mission objectives within the stated constraints. The architecture is the structure of an unmanned system, which includes software elements, externally visible properties of those elements, and the relationships among them [9]. In other words, JAUS is an accumulation of procedures and standards to unify the interface between control algorithms and autonomous technology. These procedures and standards stem from technological constraints and follow a common order to define a method of autonomy. Software utilizes information from positioning technology (GPS, localization networks, etc.), sensor acquisition systems and actuator feedback data to monitor current status and progress in tracking during a move. Interoperability is defined as the ability to exchange information between two or more systems and to use the exchanged information. This data is interoperable and is used in control algorithms to maintain its course and progressively complete a mission. Autonomous technology includes the developments of systems that aid in the ability to operate with little or no human interaction. These systems can range from GPS, to actuators, to sensor suites, to self-defense devices, to name but a few. Thus we see the emphasis is interoperability, reliability, functional capability, unified products and commonality.

Military research efforts are more product-based and oriented towards the creation of new operational systems. The goal of the work is to design and build systems. Academic research contrasts by focusing on a novel theory behind autonomy in order to develop new algorithms. Realizations of innovative technology, hardware, measurement,

communication and control principles are more sought than the application of these breakthroughs. The work presented in this thesis harnesses both efforts by implementing the technological hardware in a physical system and by accurately modeling control techniques for the specific system.

1.2.2 Drive-by-wire Conversion

Valuable concepts and ideas of drive-by-wire conversions can be found in [10] [11] [12] [13]. As we have seen in the last few sections, with so many advancements in the field of autonomous systems and so many unique drive-by-wire systems, a new focus of research has arisen. Efforts are beginning to transition towards enabling multi-vehicle collaboration to accomplish a common goal. The JOint Unmanned Systems, Test, Experimentation, and Research (JOUSTER) program has begun making initial progress for plans of research in this area. One advancement that JOUSTER has made came about recently. A Yamaha RMax gasoline powered helicopter has been converted to semi-autonomous flight. The ability to manually switch over to autonomous mode and hover while completely unmanned has been achieved. Another project in its earlier stages is the creation of a JAUS Enabled Ground Fleet. This fleet consists of five Ingersol Rand (IR) XRT 810 Electric Club Cars seen in Figure 3. This thesis studies and describes a part of this large, expansive project.



Figure 3 Ingersol Rand ClubCar XRT810. Photographed by author.

Key topics of this thesis include contributions towards the creation of the JAUS Enabled Ground Fleet. These contributions include the mechanical up-fit of the drive-by-wire steering subsystem and staging the mechanical up-fit to be JAUS enabled. In addition a substantial capability has been developed to derive and assess control methodologies for the individual vehicles and the fleet.

1.3 Contributions

1.3.1 Steering Subsystem

Development of an autonomous fleet has been an ongoing project for JOUSTER at Virginia Tech. This development includes an array of mechanical up-fits to convert a

traditional electric cart to complete drive-by-wire. One objective of this thesis is to design a steering subsystem that will enable one vehicle to operate autonomously. In other words, a system was designed to operate the existing steering wheel/shaft. Efforts focused on minimizing modifications to the existing method of steering the vehicle, enabling traditional, manual operation of the vehicle, and designing a simple, robust and reliable system. It was not desirable to completely replace the steering system and incorporate a joystick that has been done on previous autonomous ground vehicles. The introduction of an electric servo motor coupled to the steering shaft allowed for the existing steering wheel and shaft to remain operational on the vehicle. A 10:1 ratio gearhead was also used to provide the necessary torque to steer the vehicle. Tests were conducted under a series of different conditions to determine the necessary torque needed. Two pulleys were placed on the shaft of the gearhead and the steering shaft of the vehicle. A timing belt coupled the two pulleys. A sturdy, adjustable mounting plate is incorporated that utilizes the existing framework on the vehicle to sustain the back torque under normal conditions. This steering subsystem was designed for lower speed maneuvers because the project in this stage was focused on relatively slower overall operation (0-5 mph). This design constraint was set for the initial stage of development and staged the efforts to make a more agile system as needed.

1.3.2 Modified Simulation

A simulation has been developed by the author of this thesis and the fleet development team including Dr. Vahram Stepanyan, Dr. Andrew Kurdila, and Mr. Yu

Lei. The program is used to simulate autonomous ground vehicles as they navigate through a Military Operations in Urban Terrain (MOUT) site. An urban setting has been modeled that includes roads for a vehicle to navigate between mock buildings of the MOUT site.

Although the simulation, in its current state, is not JAUS enabled it was designed with a similar modular architecture. Each system in the simulation has its own separate block which allows for easy modification of individual systems, such as the steering system. Furthermore, it is generally applicable to a variety of vehicles. As the project develops and the team grows, an upgrade to utilize JAUS is a logical and potential next step. However, it is beyond scope of this thesis. Current efforts focus on accurately modeling the vehicle using the simulation created.

Modifications were made to this simulation to more accurately model the Ground Fleet. These modifications included altering the existing steering model to more closely match the steering system derived and developed for the drive-by-wire conversion. Steering motor voltage commands were replaced with steering angle commands. This choice was made since the controller hardware and software was obtained from the servo motor supplier Quicksilver. A traditional interface between the command software and steering actuators input a voltage command that the steering motor needed to achieve a desired steering angle. The Quicksilver motor has a built-in controller and feedback loop, which accepts steering angle commands. The conversion from voltage commands to steering angle commands has been carried out in the existing simulation. This directly links the simulation to the up-fitted steering system on the vehicle, putting the fleet one step closer to complete autonomous conversion with a detailed simulation.

Chapter 2 – Ground Vehicle Platform

This section discusses the initial conceptual design process and decisions that motivated the choice of the vehicle to convert to autonomous operation. Many factors went in to determining which vehicle would most efficiently suit the objectives of the project. These objectives included performance, power plant, payload, number and type of necessary modifications, cost and appearance, for example. Valuable experience and knowledge from the team members lead to the successful decision to use an Ingersoll-Rand (IR) 810 Electric (E) Club Car. The vehicle is to be up fitted with an array of sensors, including a GPS (Global Positioning System), an IMU (Inertial Measurement Unit), LIDAR (Laser Identification Detection and Ranging), and Compact Field Point, to name a few.

2.1 IR 810 E Club Car

Many different considerations went in to determining which vehicle platform to use. The IR 810 E Club Car offered many attributes that were favorable in this application. The overall size of the car was large enough to offer ample room for payloads and still fit in the back of a truck for easy transportation. A dump style bed offered 0.215 m^3 of volumetric storage space with a payload capacity of 136 kg not including driver or passengers. The two person, bench style seat also contributed valuable space protected by a fold down windshield and canopy. With such space, it was determined that little frame modifications would be required to provide support for

additional storage units. A total vehicle payload of 362 kg made it capable of hosting a variety of customer instrument packages [14].

The electric drive train was extremely favorable when considering the target velocity for the project. Focusing on low speed maneuvers, the electric drive train is extremely easy to control when compared to an internal combustion (IC) engine. It was also determined that an electric motor would offer better fidelity at lower speeds as compared to an IC engine. Lastly, there would be no exhaust to deal with during testing and up-fit.

2.2 Sensor Suite

To achieve complete autonomous operation, it is necessary for the vehicle to have the situational awareness sufficient to navigate from one point to another. To do this, the vehicle needs to know position (past, current and projected). It must be capable of obstacle avoidance and compensate for dynamic limitations to prevent harm done to the vehicle during operation

GPS is a very important tool for an autonomous vehicle. With a known desired endpoint and real time measurements of the vehicle path, algorithms can process the real time data and progressively orient the vehicle such that it reaches the desired location. An onboard compass will always give the vehicles bearings.

Not only does a vehicle need to know where it is in its surroundings relative to where it needs to go (globally), it needs to know its own orientation in an inertial frame. This knowledge enables the vehicle to avoid dangerous configurations or maneuvers

(e.g., if the vehicle is on a hillside, care must be taken to avoid rollover.). An IMU will provide the required information from a series of gyroscopes. The control system will adjust the vehicles' states (velocity, position) accordingly to prevent damage. It will also provide information about location using accelerometers and measuring rate of change of yaw, pitch and roll from onboard gyroscopes.

2.3 Other Features of Autonomous Operation

Autonomous vehicles possess the capability to complete pre-defined missions that could sometimes be otherwise life threatening. With the ability to operate individually or as a networked group for relatively long periods of time, it is possible to capture video imagery, thermal imagery, radar and other forms of surveillance of unknown environments. Properly equipped with video cameras, thermal imaging, laser scanners and radar, a vehicle can enter an unknown area and capture priceless information in unsafe situations. If something goes wrong, it would not be necessary to endanger human life.

Chapter 3 – Mechanical Up-fit

To modify a vehicle for complete autonomous operation involves designing actuators to replace the human driver who controls acceleration, deceleration, steering and emergency reactions. One goal of the project was to make as few modifications to the existing vehicle as possible. It was decided that the appearance and traditional human operation must remain as unaffected as possible during the modification for autonomous operation. This thesis concentrates on one specific subsystem that was modified during the conversion of the vehicle for autonomous operation. We discuss the changes to the steering subsystem in the next section. Below is a list of steering system requirements outlined by the team before design took place. The system was built to meet, but was not limited to, the following requirements:

- The autonomous steering system shall be designed to alter the existing steering procedure as little as possible.
- The autonomous steering system shall be designed to not impede any of the existing system operations of the vehicle (i.e. suspension system, braking system)
- The system shall be designed to make as few modifications to the existing vehicle framework, mounting apparatus, electrical circuits and visual appearance as possible.
- The autonomous steering system must deliver 180 in/lbs of torque necessary to turn wheels on terrain that may have high friction coefficients and/or is not flat (i.e. increase the tire contact patch). This specification is reasoned in the following sections.

- The autonomous steering mount system was designed to hold the engine/gearhead/sprocket assembly rigid enough to maintain necessary belt tension to deliver the necessary torque.
- The design shall be constrained to one degree of freedom; motion perpendicular to the steering shaft to allow for belt tensioning.
- The autonomous steering system was designed to enable one to tension the belt with as little trouble as possible.
- The autonomous steering system will operate for speeds from 0-5 mph accurately.

3.1 Steering Subsystem

The autonomous steering system is comprised of all physical elements necessary to activate the existing steering system. The goal has been to achieve the modification to autonomous operation with as few modifications as possible. One method that was used on other autonomous vehicles at Virginia Tech (during the development of the DARPA Grand Challenge and DARPA Urban Challenge Vehicles) was a conversion to electronic steering. This system is used today in some production automobiles (such as the Nissan Altima). This system relies completely on an electric system to steer the vehicle. By replacing the mechanical steering system with an electric system, if the electrical system fails (the batteries are dead, etc.) the vehicle cannot be steered. Additionally, in order to implement a joystick steering system, many alterations to the existing vehicle would have

to be made (include removing steering wheel and replacing it with a joystick) which defies the goal of preserving the factory operations and appearance as much as possible.



Figure 4 Virginia Tech's DARPA Grand Challenge Vehicle Cliff's Joystick

Steering. Photograph used by permission.

Consequently, an actuator that operates the steering system subject to these constraints was to be designed. By directly coupling the steering shaft to a servo motor, the rotational motion is possible and actuation is achieved as needed with one linkage. This yields an effective system. An appropriately sized and tuned servo motor produces very accurate positioning over a very wide range of accelerations. Virginia Tech's

DARPA Grand Challenge teams had experience and existing LabView codes to command the QuickSilver servo motors and controllers.

To determine which motor to use, it was determined that the steering would experience most resistance when initially at rest and initiating a change in path. Therefore, static torque measurements were taken in a variety of conditions. The vehicle was placed on a variety of terrain in a variety of different conditions (flat ground, small drainage ditches) to vary the friction with the ground that causes steering resistance. This steering resistance or feedback is the main resistance to the actuator, therefore it was necessary to see how much resistance the system may face. A torque wrench was couple to the steering shaft and tests of in increments of 10 in-lbs were conducted. A maximum torque was found to be 20.34 N-m (180 in-lbs). Under specific conditions incorporating a factor of safety of 1.5, it was determined that the necessary deliverable torque from the system would be 31.64 N-m (280 in-lbs). There was no servo motor that directly supplied a torque that large that would fit in this application, so a 10:1 ratio gearhead from Alpha gearheads was introduced in the new subsystem. The QuickSilver 34N-1 motor/encoder delivers 3.24 N-m, and with the Alpha gearhead, the system delivers 32.24 N-m. The recommended controller for the motor was the SilverNugget N3 controller/driver by QuickSilver. This was the lowest cost motor that met the torque specifications.

One issue with utilizing a servo motor and gearhead on the existing steering system was manually back-driving the motor. The manual operation now must overcome the resistance due to tire friction with the ground, linkage resistance, and the resistance created from backdriving the motor and gearhead. One solution to alleviate the resistance

involved designing a clutch system that could be engaged to operate autonomously and disengaged to operate manually. This approach would add another level of complexity, a potential source of failure, and additional cost to the problem. Experimental tests on backdriving the motor without a clutch showed an increase in resistance, but the decrease in performance was not sufficient to demand an alternative method (i.e. incorporating a clutch). Efforts to utilize the designed steering subsystem to act as a power assist to manual mode operation of the steering system were not looked into once the experimental tests showed it was possible to manually back drive the motor.

To couple the servo motor to the steering shaft, some modifications were necessary.

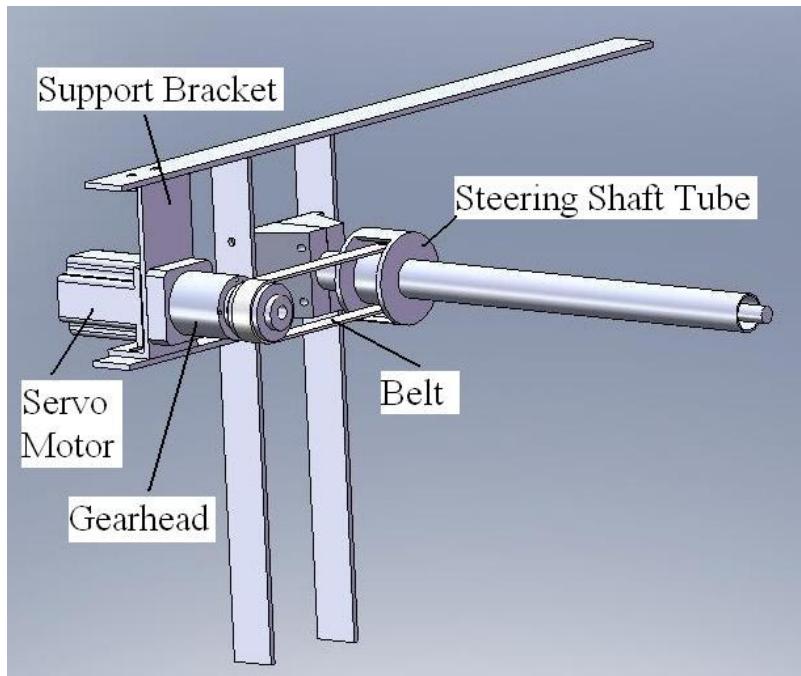


Figure 5 Designed Steering System. Image created by author.

First, to mount the motor near the steering shaft, the frame that supports the canopy/windshield and the dash board was used. An aluminum plate was fabricated to

mount the motor shaft parallel to the steering shaft. The mounting holes were slotted to allow for belt tensioning, while restricting motion to only one direction (perpendicular to the steering shaft) to maximize rigidity. The radial load on the motor cannot exceed 6.8 N (60 lbf), therefore tensioning of the belt cannot exceed 6 N. An aluminum safety tube designed to provide protection from the rotating steering shaft needed to be modified. To keep the system as nearly unmodified as possible, a section was removed below the dashboard and out of sight. A larger diameter section of aluminum tube was welded in place of the section that was removed. The existing tube had a diameter of 0.05 m (2 in) with .006 m (0.25 in) wall thickness. The section installed had a 0.1 m (4 in) diameter with the same wall thickness (.25 in). The gap created by the difference in diameters was filled with a machined piece of aluminum shaped as a washer. A piece of 0.006 m (.25 in) sheet of aluminum was fabricated to appear much like a very large washer that was welded to both sides of the large diameter section. This was then welded in place of the removed section on the steering tube itself. This housed a pulley fixed to the steering shaft. A matching pulley with a distributor-specified belt connected the two systems. A timing belt and pulley coupling system was used because a toothed belt and pulley offers a much more durable and dependable system than a smooth belt that has more tendency to slip. A chain and sprocket system was less desirable with all of the maintenance necessary with a chain. The slotted mounting holes made this system simple for an operator to tension the belt to specification with as little difficulty as possible. The tensioning specification came directly from the motor and gearhead radial and lateral for specifications. Please see the Figure 6 for reference.



Figure 6 Actual Steering System. Photographed by author.

3.2 Steering Characterization

Verifying the accuracy of the steering system entailed testing the motor/controller and testing the steering system response to different inputs. Software was supplied by QuickSilver (QCI software) to control the SilverNugget N3 controller/driver and 34N-1 motor/encoder was used to actuate the hardware. The software was very basic and operated by creating programs consisting of different commands to carry out certain maneuvers. Most tests were run using the controller command “MRT” (Move Relative, Time based). This command instructs the controller to move a certain number of counts relative to its current position in a specified amount of time. In order to test the steering system functionality, simple codes were generated to operate the steering in both directions, at different magnitudes at different times. Upon satisfactory inspection of the mechanical functionality of the system, further tests were conducted to verify the performance of the inner control loop built into the system.

Data were collected within the software that logged real time desired outputs into text files. The data for several command inputs were logged to files and subsequently plotted in a variety of formats. One of the most useful representations was the plot of the commanded trajectory and actual trajectory as a function of time. One way to think of the data is that of a history of what the motor is being told to do (move to “target position” for example) and another set is what the motor is actually doing (“actual position” for example). This data shows how accurate the feedback is and how the motor corrects itself to achieve input goals. Gains can be adjusted to modify input versus output. However initial experimentation exhibited acceptable error so that further operation continued to use the current settings. Below is the software user interface, the program supplied by QuickSilver.

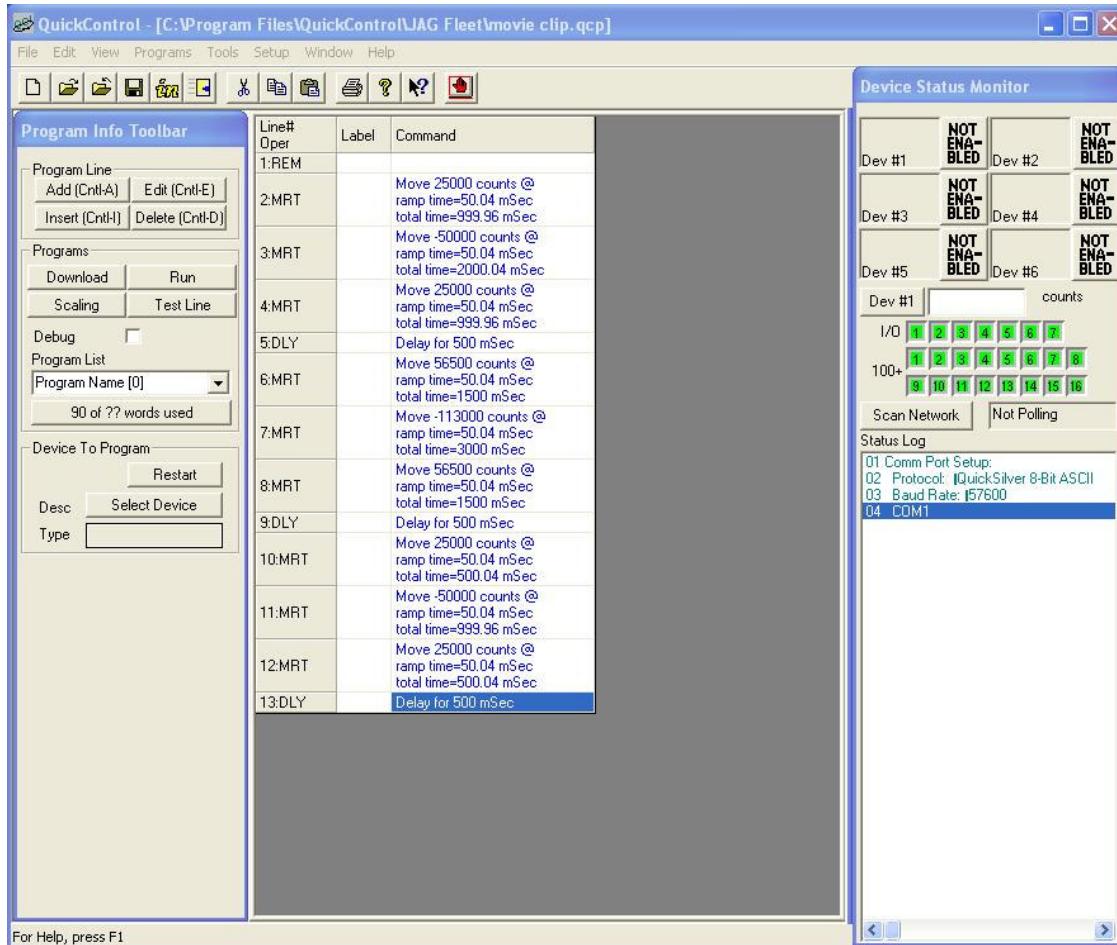


Figure 7 Software User Interface. Image created by author.

These plots showed under what conditions the greatest difference occurred. As expected, the greatest error was observed where the angle was changing most rapidly per unit time (greatest angular velocity). These results allowed for analysis of the percent error. Analysis showed a percent error within 1.3%, an error so small it was disregarded. It was not the objective of this thesis to fine tune the controller to decrease the error due to the existing controller dynamics. It was to design and create an operational steering system, and attain a simulation modeling this system.

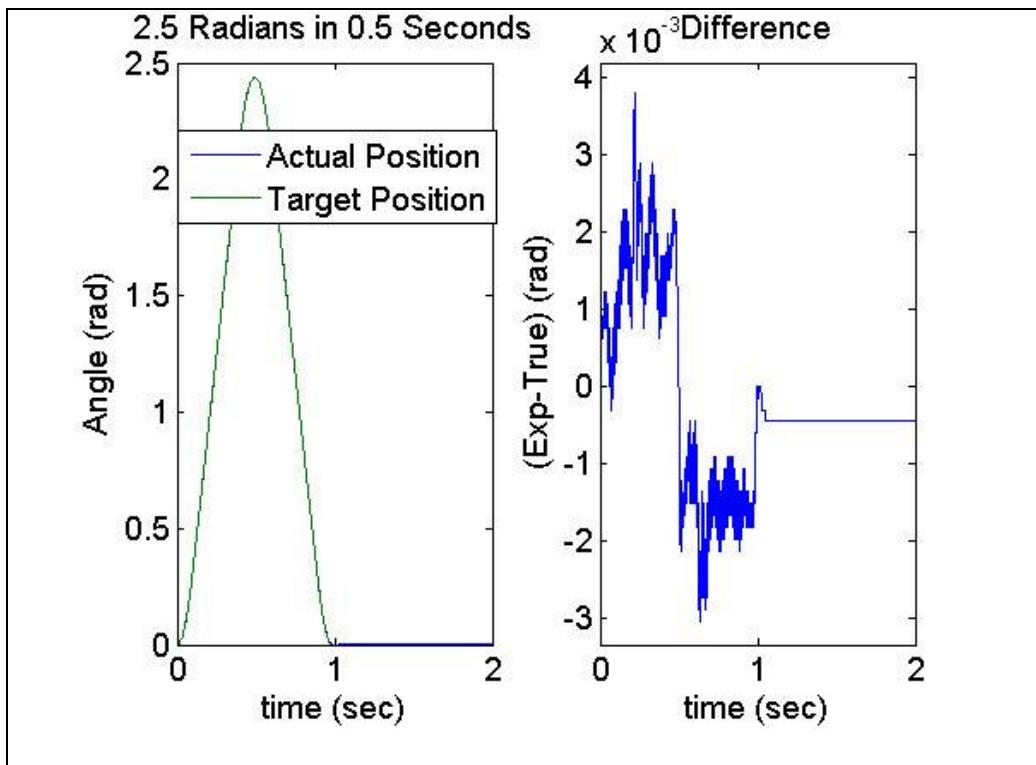
All of the following tests were conducted in a lab with concrete floors. The coefficient of friction was relatively low when compared to concrete or deep grassy environments. As a result of other tests of functionality in different environments no signs of delay or overworking the system were observed. Things like belt slippage and damage, as well as the motor stall during a test would have suggested the systems incapability. Because none of which were observed confidence was established in the system and in the data produced. Furthermore, the tests below were done statically, the vehicle remained still. The other functionality tests sufficed for verification of the dynamical operation of the system on the vehicle.

The series of tests discussed below were conducted to test the accuracy of the motor and controller. Commands and measurements were of the motor, not the wheels. Sensors in the motor relayed the “Actual” data, which was compared to the “Target” data the motor also recorded. The data collected was in counts of the servo motor. This was related to radians by determining that 4125 counts equaled one revolution of the motor. Knowing one revolution is 2π radians, the data was converted. The maximum angular range of motion of the steering wheel is 5 revolutions, from lock to lock. One suggested area of future work entails conducting tests to see how the error recorded affects the steering angle of the wheels (which takes into account how the gearhead affects this error, as well as the kinematic translation in the rack-and-pinion).

Valuable assistance was received from representatives with QuickSilver Control. Not only did they assist in familiarizing some team members with the operation of their software, but valuable input with suggestions of how to use their equipment to conduct verification and characterization tests. The technical support at QuickSilver suggested a

few series of tests to try which will produce the data sought. The team agreed on a few different series of tests, outlined below. To determine the initial angular velocity range that was to be tested ranges from 0.25 radians per second to 8 radians per second. Visual inspection of these dynamics suggested a range to stay within.

The first series of tests consisted of simple angle commands relative to time. The code issued control commands during each test where the goal was to achieve a specified command steering angle in a set time. Each of the three tests has a set target steering angle with a decreasing time interval to reach that angle. Also, the difference in the commanded steering angle (Target) was subtracted from the actual steering angle (Actual) to measure the difference as a function of time.



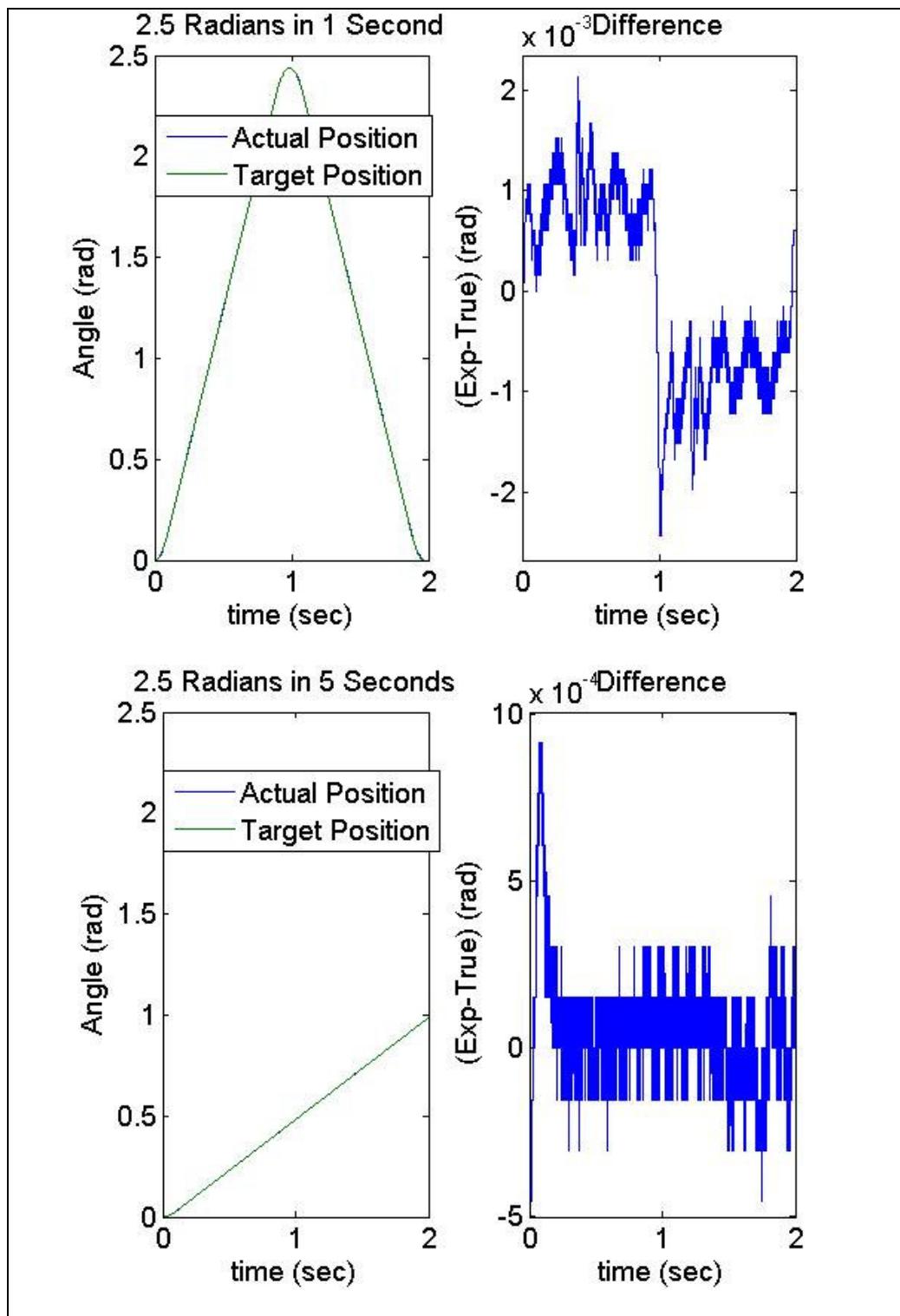
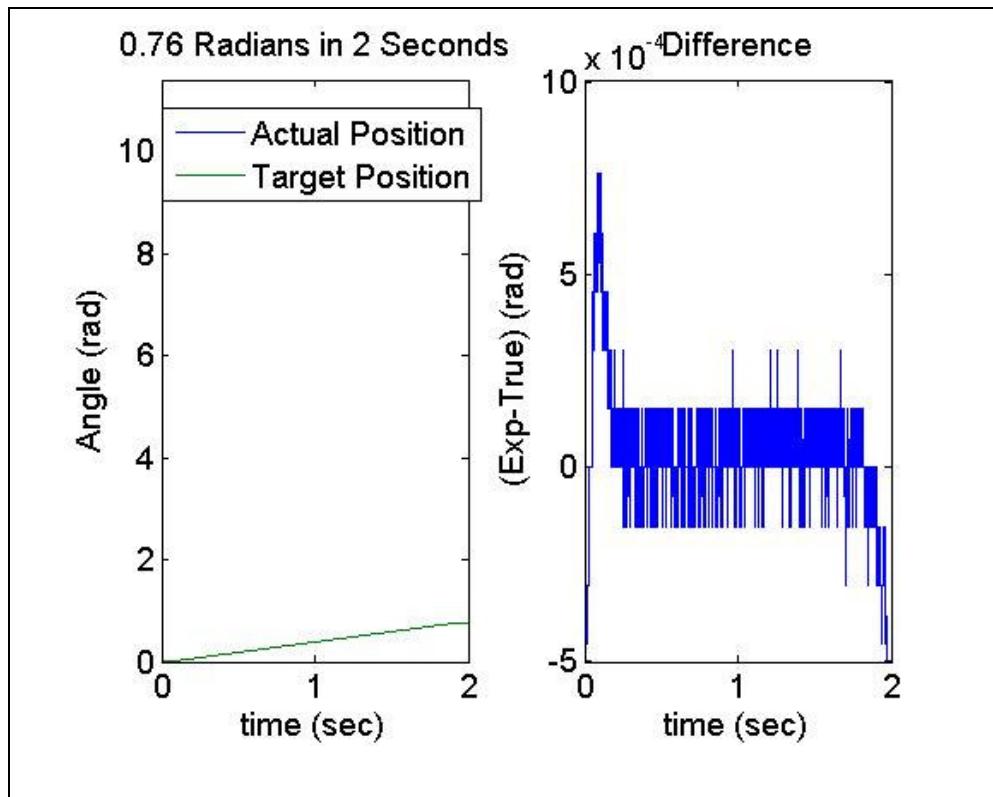
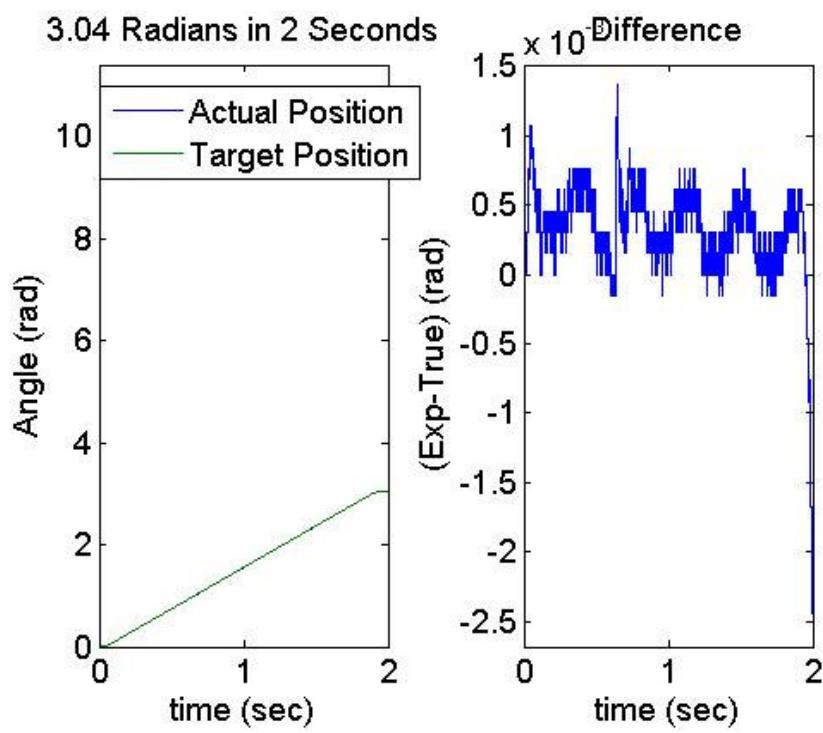
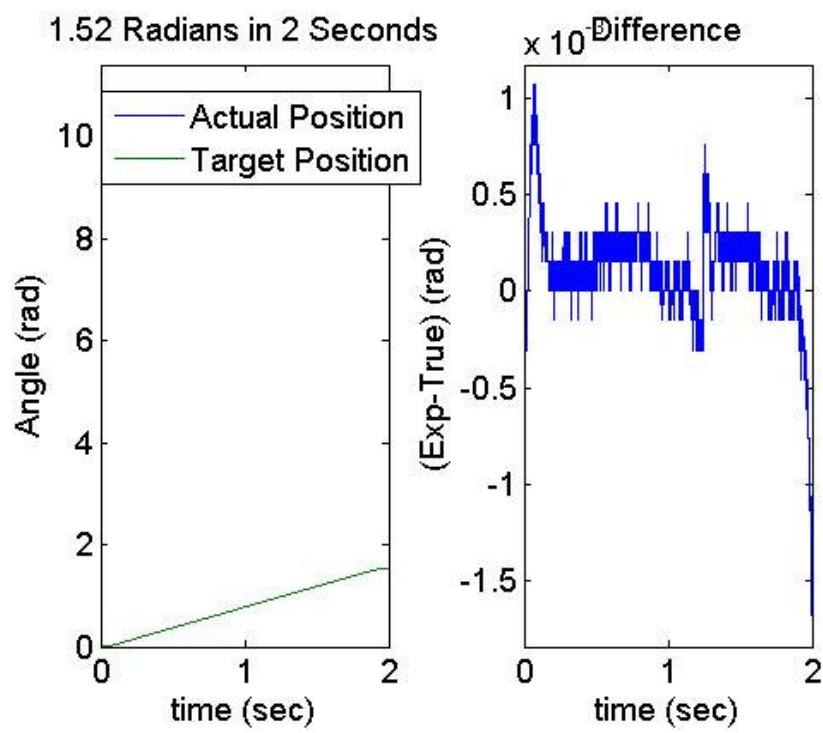


Figure 8 Test Series 1

The following tests consisted of a fixed time period while increasing the magnitude of steering motor angle. Again, analyzing the error was the main objective of this test. After learning that the error was less than 1.3% at the final time, it was concluded that the error was not significant enough to model the specific dynamics of the steering motor and controller system. In other words, it was decided that the combination of the steering motor and factory controller could be modeled as a perfect tracking subsystem. The percent error was calculated point by point by taking the difference between Actual and Target positions and divided by Actual data then multiplied by 100%

$$\left(\frac{\text{Actual} - \text{Target}}{\text{Actual}} \right) 100.$$





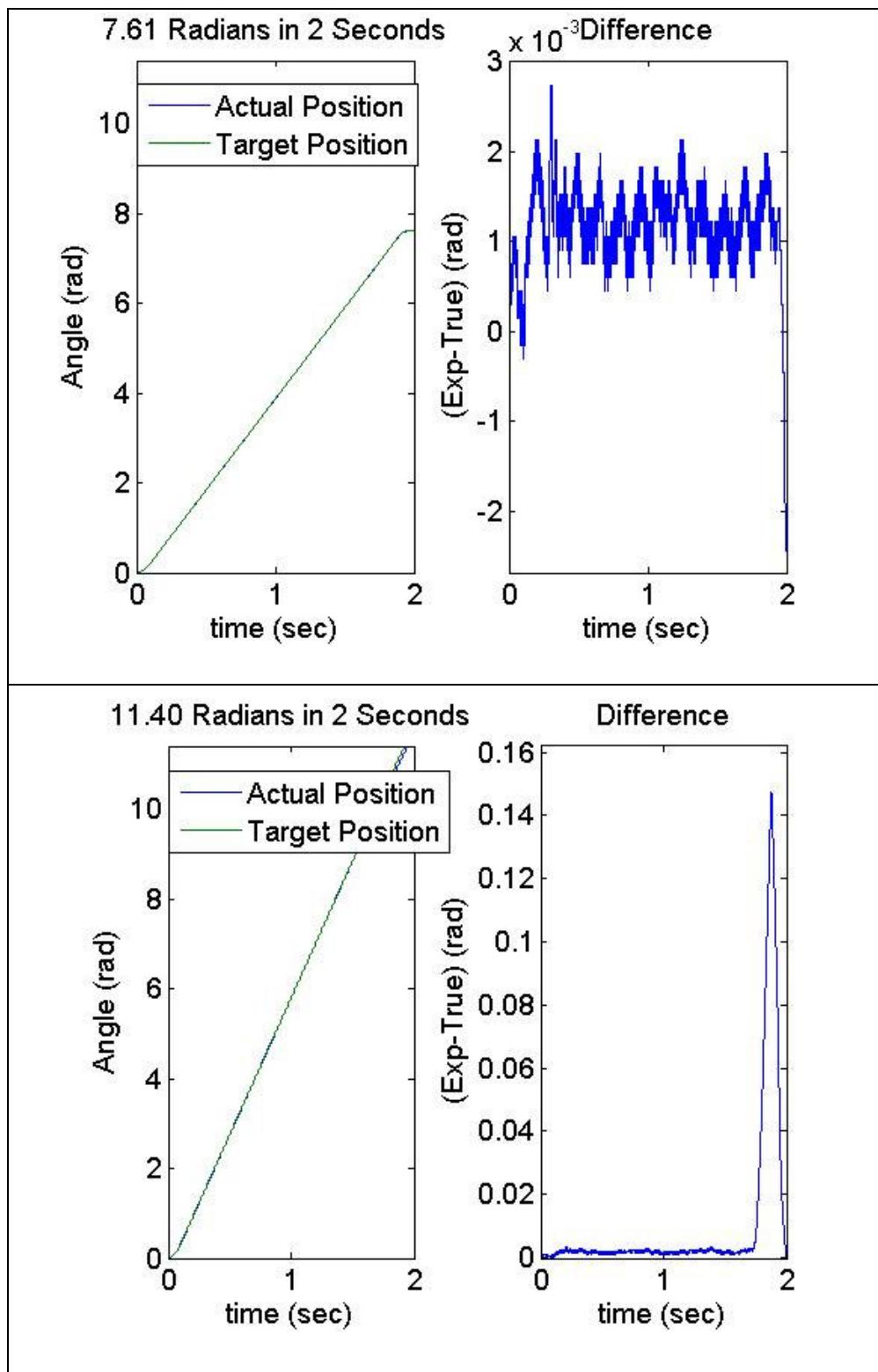


Figure 9 Test Series 2

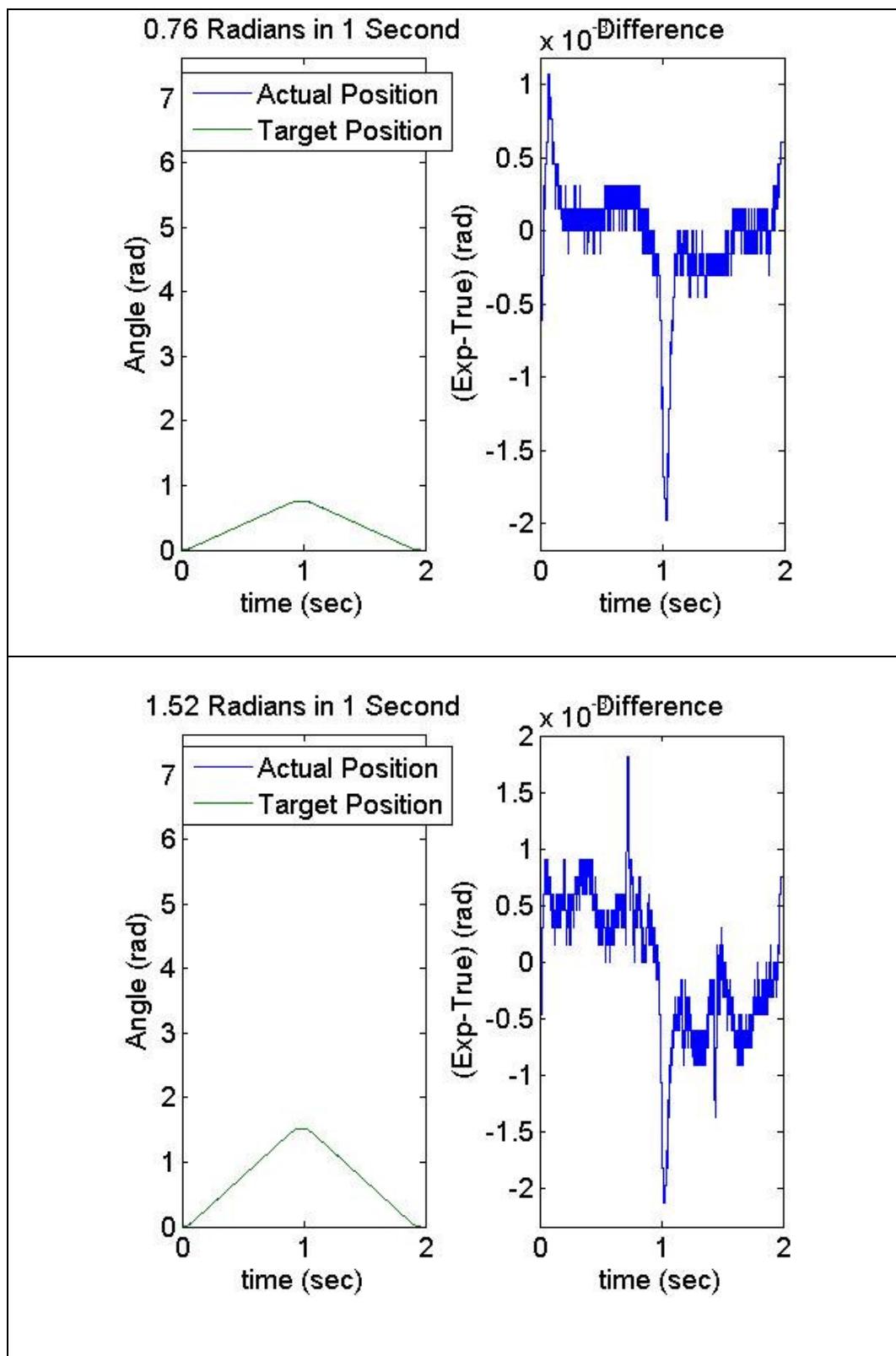
The following table displays the key numerical values analyzed for the second series of tests. The minimum and maximum difference in the input commands compared to the recorded outputs. Also the average and standard deviation of the data was taken and recorded.

Summary Table of Test Series 2					
	Min	Max	Mean	Std Dev	Max Error (%)
Test 4	0	0.0012	1.11E-04	1.61E-04	0.157894737
Test 5	0	0.0017	1.98E-04	2.06E-04	0.111842105
Test 6	0	0.0024	4.12E-04	2.69E-04	0.078947368
Test 7	0	0.0027	0.0013	4.11E-04	0.035479632
Test 8	0	0.1473	0.0099	0.0269	1.292105263
	Min	Max	Mean	Std Dev	
Series 2 (combined)	0	0.1473	0.002384278	0.011913385	

Table 1 Data Table of Test Series 2

Further tests were conducted to see how the system reacts to different magnitudes and rate of changes in the commanded steering angle. To get a better feel for how the system works, increasing target angle commands were sent to the motor with the same

duration times. This created the varying rate of changes of the steering angle. The maximum error, or the difference between the input (target) data and the output (actual) data, was analyzed.



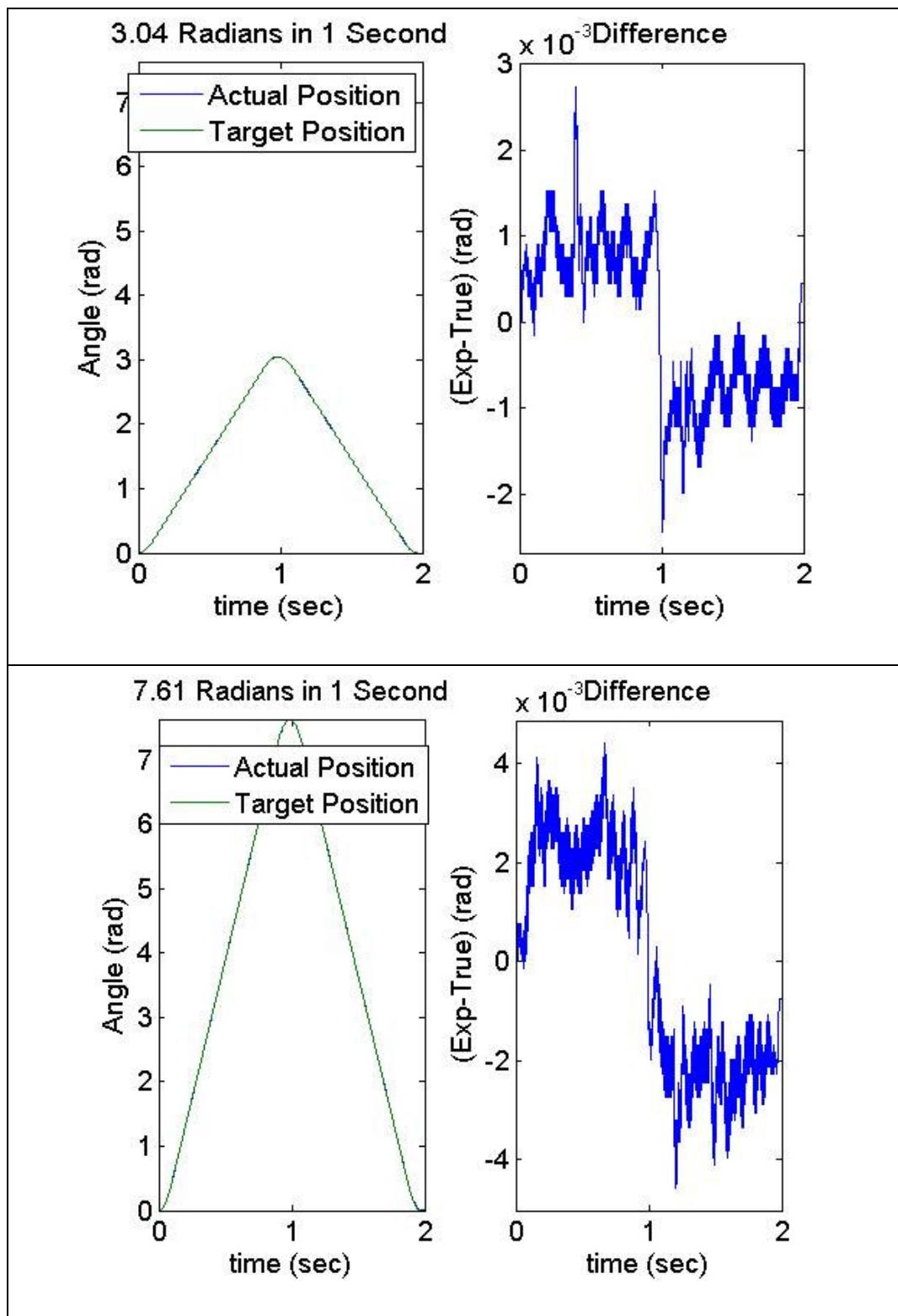


Figure 10 Test Series 3

The following table summarizes some of the important trends that can be observed in the plots above. Again, the minimum and maximum differences between the input command and the output data are displayed, as well as the mean and standard deviation of the data plotted. The maximum percent error was calculated and also displayed.

Summary Table of Test Series 3					
	Min	Max	Mean	Std Dev	Max Error %
Test 9	0	0.002	2.45E-04	3.11E-04	0.263157895
Test 10	0	0.0021	5.65E-04	3.48E-04	0.138157895
Test 11	0	0.0027	9.37E-04	3.81E-04	0.088815789
Test 12	0	0.0046	0.0024	8.47E-04	0.060446781
	Min	Max	Mean	Std Dev	
Series 2 (combined)	0	0.00285	0.0010364	0.000472	

Table 2 Data Table of Test Series 2

3.3 Conclusions

The following plot is data collected from a test run that commanded a target angle of 2.5 radians to be obtained in 5 seconds. Only 2 seconds of data can be collected in one test. In this test, the steering angle was commanded to vary linearly from 0 to 1 radian over a period of 2 seconds.

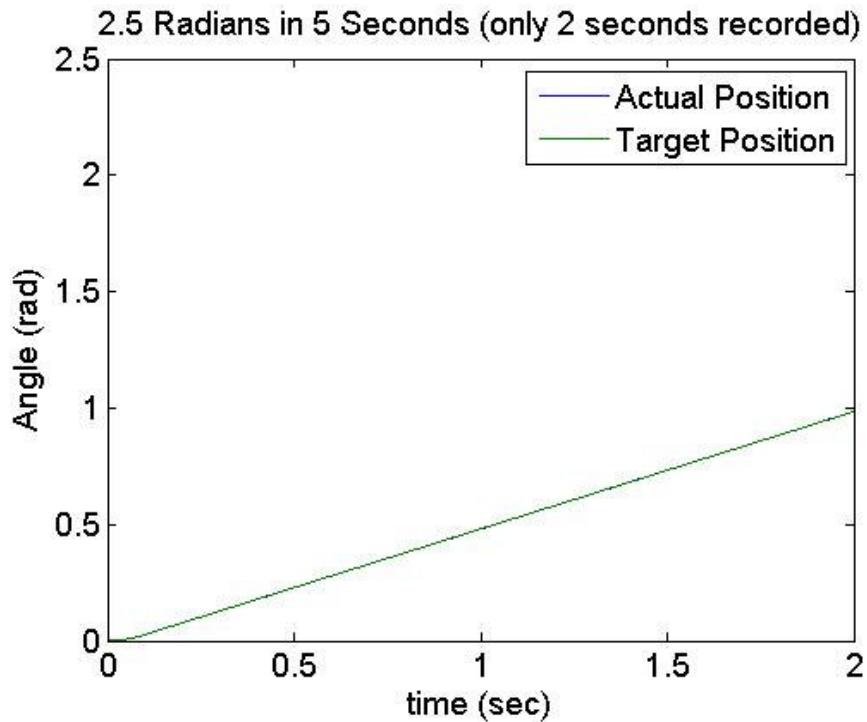


Figure 11 Linear Relationship Between Actual Position and Target Position

The difference in input versus output (error) was much less for a slower, less agile input, as expected. Figure 11 shows a virtually perfect relationship between the two at a relatively slow angular velocity. The error was about three times less than with the more agile, non-linear command function. This is important because this project was structured for lower speed maneuvers, not demanding quick movements requiring high

angular velocities of the steering subsystem. An example of a more agile, non-linear command function is the one used to run the test for which the data is plotted in the last plot of Figure 9. The command used in this test generated a larger steering angle in a shorter time period, greatly increasing the angular velocity. Keep in mind error was still within 1.3%.

Below is a plot of the difference between actual position and target position. This gives a sense of the small scale difference between the two. Roughly speaking, the more agile the commanded maneuver the higher the error. This agrees with intuition.

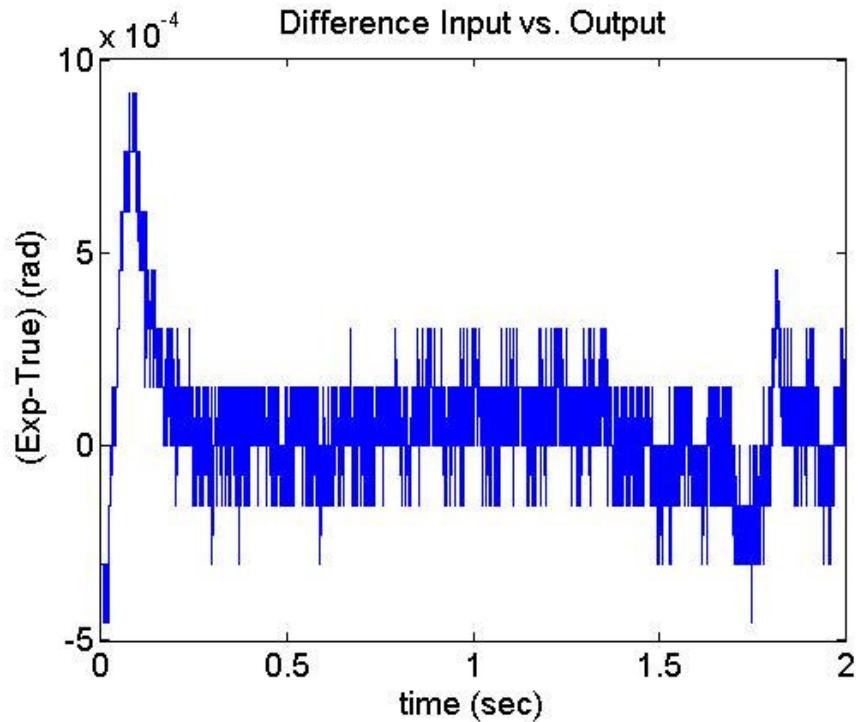


Figure 12 Difference of Linear Relationship

The maximum error was as great 0.32% as compared to 1.3% error. The difference in the programs was the total maneuver time. Both programs turned 2.5

radians; this program turned 2.5 radians in 5 seconds as compared to 0.5 seconds (ten times greater).

3.3.1 Discussion of Error Dynamics

Further data analyses lead to the inspection and characterization of the error. The dynamics of the system and the tracking error were further studied to get an idea of the noise properties for the tracking controller. From inspection of plots such as Figure 12, it was seen that different frequencies existed. The current model only captures the first low frequency. However, the error is so small that it can be considered insignificant for this application.

3.4 Further Tests

Further tests included a series of tests under different operating conditions for proper functionality. The vehicle was placed on different terrain (including asphalt, concrete, loose gravel, hard packed dirt, and deep grass conditions) in different situations and operational tests verified the following:

1. Full functionality of steering, from steering lock to steering lock
2. No loosening of hardware
3. No damage done to vehicle or steering subsystem
4. No belt slippage
5. Overall operation

No actual data was collected during these tests. They were simply tests to verify overall operation, no excess strain on any of the components. Upon completion of these tests, tear-down and inspection of the system showed no apparent damage or improper wear on any of the components. It is recommended to conduct further long-term tests to verify these results for an extended period of operating time.

Chapter 4 – Single Vehicle Operation: Control Simulation

Extensive work has been done to create an operational simulation of the proposed JAUS Enabled Ground Fleet. The program Simulink was used to create a block diagram simulation outlining the autonomous ground vehicle and all of its components. Simulink coordinates and manages these numerous components by creating a block diagram for each and arranging them in a hierarchical manner. The ability to wire, in a sense, each diagram together enables the operator to structure the simulation. This is one of the key attributes that weighed favorably in selecting to model the system with Simulink.

4.1 Equations of Motion (EOM)

The first step in constructing a simulation accurately modeling the autonomous vehicle was deriving the equations of motion for it. Below is a description of assumptions made and the derivation of equations of motion for the simulation developed in this thesis. This description outlines a derivation in [15].

For simplicity, the JAG fleet vehicle was modeled as a planar vehicle with four wheels. The front two wheels rotate to steer the vehicle. They are assumed to always be parallel, which is common in simple autonomous ground vehicle models. The back two wheels are also parallel to each other. They are perpendicular to the centerline of the vehicle which cuts the vehicle exactly in half in the direction it is oriented. The lateral centerline perpendicular to this splits the vehicle exactly in half, assumed to be the same distance from both the front axle and the rear axle. This assumption simplifies the derivation since the front and rear wheels can be collapsed into one front wheel and one rear wheel. For this discussion, the direction in which the vehicle is facing is along the longitudinal axis (the \bar{b}_1 direction in Figure 13). The lateral direction will be its perpendicular (the \bar{b}_2 direction in Figure 13).

In reality the assumption that the front two wheels remain parallel is not accurate. For a rigid body with two turning wheels to rotate about a fixed point, the distance between the front two wheels and the rear two wheels creates two different instantaneous centers of rotation if the wheels stay parallel. Two different instantaneous centers of rotation mean each wheel is turning along a different path. By turning the inside wheel slightly more, the instantaneous centers of rotation become meet at a common point. The bicycle model (common to many vehicle dynamics derivations, especially in autonomous control) avoids having to account for two different wheel turning radii was used in this derivation. Please refer to the Appendix A for a more detailed explanation of this assumption.

The assumption stating the front two wheels are parallel allows for one to collapse them into one wheel along the longitudinal axis. Additionally the rear wheels are

collapsed to one wheel also onto the longitudinal axis. This modeling assumption is commonly carried out for simple ground vehicle dynamics, particularly in the autonomous control literature. The following figure shows this conceptual vehicle in the XY plane. The position of the center of mass in the plane, the angle of the wheels with respect to the latitudinal direction, and the orientation of the vehicle itself within the XY plane will be denoted as x_c , y_c , ϕ (phi), and θ (theta) respectively.

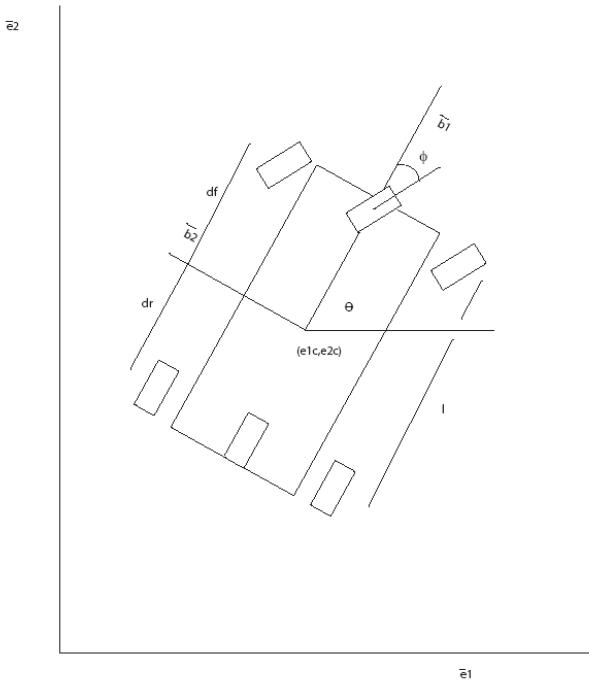


Figure 13 Vehicle Diagram. Image created by author.

The distance between the front and rear wheel is denoted as l . Denote the distance from the rear wheel to the center of mass as d_r . Denote the distance from the front wheel to the center of mass as d_f . The following equations can be written:

$$\begin{aligned}x_f &= x_c + d_f \cos \theta \\y_f &= y_c + d_f \sin \theta \\x_r &= x_c - d_r \cos \theta \\y_r &= y_c - d_r \sin \theta\end{aligned}$$

1

Differentiating with respect to time to yields the velocities

$$\begin{aligned}\dot{x}_f(t) &= \dot{x}_c(t) - d_f \dot{\theta}(t) \sin \theta(t) \\ \dot{y}_f(t) &= \dot{y}_c(t) + d_f \dot{\theta}(t) \cos \theta(t) \\ \dot{x}_r(t) &= \dot{x}_c(t) + d_r \dot{\theta}(t) \sin \theta(t) \\ \dot{y}_r(t) &= \dot{y}_c(t) - d_r \dot{\theta}(t) \cos \theta(t)\end{aligned}$$

2

The vehicle is assumed to obey no slip conditions at the interface of each wheel with the ground. The no-slip condition means that the wheels roll on the ground. Furthermore, the velocity of the point on the wheel that is in contact with the ground is equal to the

velocity of the ground. When we apply the no-slip conditions, we obtain the following equations:

$$-x_f(t) \sin(\theta(t) + \phi(t)) + y_f(t) \cos(\theta(t) + \phi(t)) = 0$$

$$-x_r(t) \sin \theta(t) + y_r(t) \cos \theta(t) = 0$$

3

Substituting Eq. 2 into Eq. 3 yields

$$-\left[x_c(t) - d_f \theta(t) \sin \theta(t) \right] \sin(\theta(t) + \phi(t)) + \left[y_c(t) + d_f \theta(t) \cos \theta(t) \right] \cos(\theta(t) + \phi(t))$$

$$= -x_c(t) \sin(\theta(t) + \phi(t)) + y_c(t) \cos(\theta(t) + \phi(t)) + d_f \theta(t) \cos \phi(t) = 0$$

$$-\left[x_c(t) + d_r \theta(t) \sin \theta(t) \right] \sin \theta(t) + \left[y_c(t) - d_r \theta(t) \cos \theta(t) \right] \cos \theta(t)$$

$$= -x_c(t) \sin \theta(t) + y_c(t) \cos \theta(t) + d_r \theta(t) = 0$$

4

Now define the vehicle's velocity terms in the rotational frame using (u, v) .

$$x_c(t) = u(t) \cos \theta(t) - v(t) \sin \theta(t)$$

$$y_c(t) = u(t) \sin \theta(t) + v(t) \cos \theta(t)$$

5

Substituting Eq. 5 into Eq. 4, we obtain the simple expressions

$$u(t) = \tan^{-1} \phi(t) l \dot{\theta}(t)$$

$$v(t) = d_r \dot{\theta}(t)$$

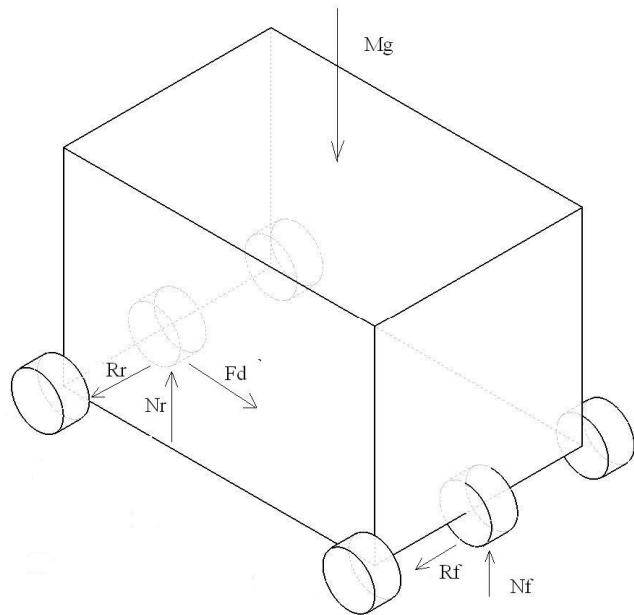
6

Finally, the kinematics of the vehicle necessary for the derivation can be summarized.

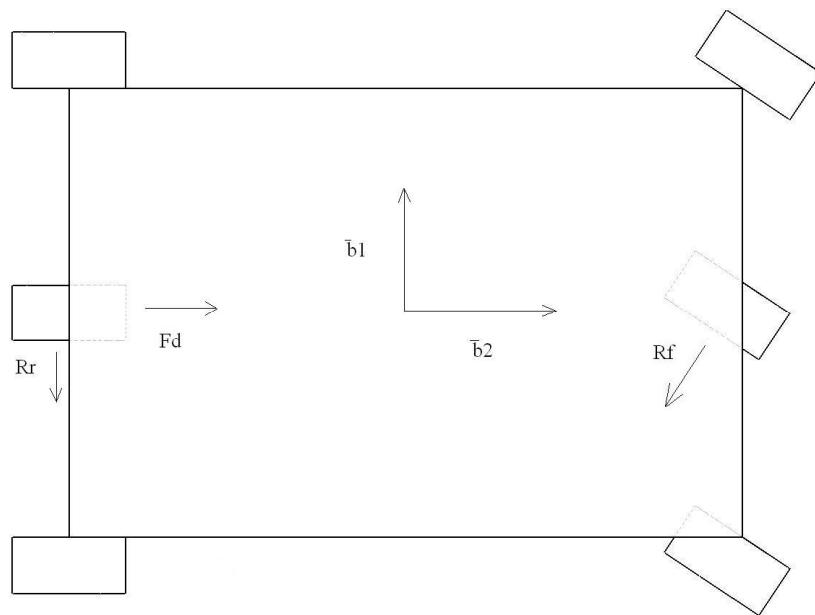
$$\begin{aligned} x_c(t) &= u(t) \cos \theta(t) - d_r \left(\frac{\tan \phi(t)}{l} \right) u(t) \sin \theta(t) \\ y_c(t) &= u(t) \sin \theta(t) + d_r \left(\frac{\tan \phi(t)}{l} \right) u(t) \cos \theta(t) \\ \dot{\theta}(t) &= \frac{\tan \phi(t)}{l} u(t) \end{aligned}$$

7

To derive the dynamics of the system, the Newton-Euler formalism is used. It is assumed that there is no friction between the wheels and the vehicle. More precisely, they are insignificant relative to the other forces, and therefore ignored. The free body diagram provides the following equations:



(a)



(b)

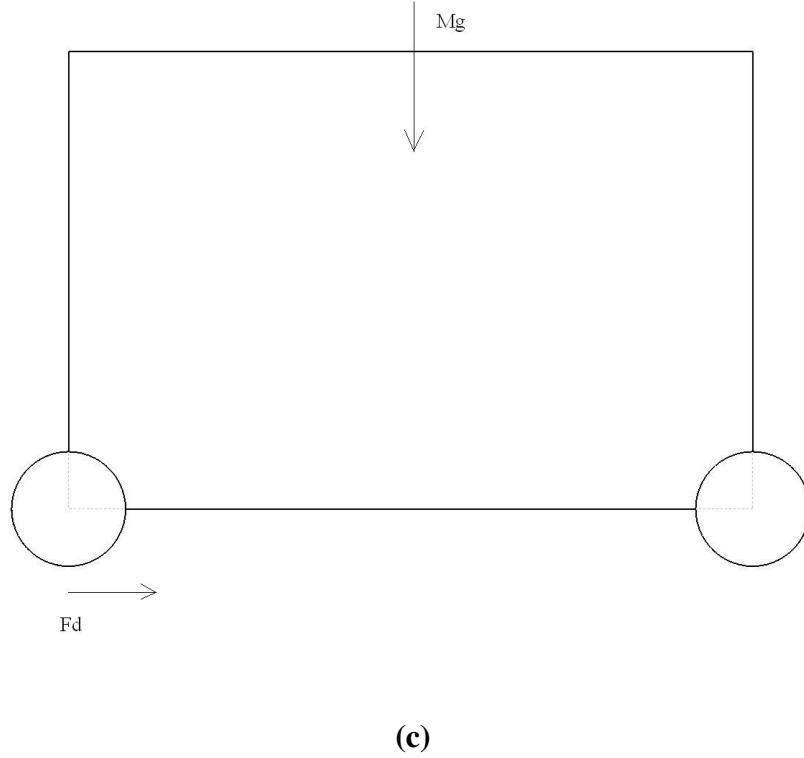


Figure 14 Free Body Diagram (a) Isometric View (b) Top View (c) Front View.

Images created by author.

The driving force, F_d , is the force from the drive motor applied to center of the rear wheel. The front and rear wheel reaction forces are denoted as R_f and R_r respectively. The yaw moment of inertia about the z axis (vertical) is denoted as J . The total mass of the vehicle is denoted as m .

$$m \left(\dot{u}(t) - v(t) \dot{\theta}(t) \right) = F_d - R_f \sin \phi(t)$$

$$m \left(\dot{v}(t) - u(t) \dot{\theta}(t) \right) = R_r + R_f \cos \phi(t)$$

$$J \ddot{\theta}(t) = d_f R_f \cos \phi(t) - d_r R_r$$

From the last equation in Eq. 8, R_r can be obtained.

$$R_r = \frac{d_f}{d_r} R_f \cos \phi(t) - \frac{J}{d_r} \ddot{\theta}(t)$$

Substituting R_r into the second equation of Eq. 8 yields

$$m \left(\dot{v}(t) - u(t) \dot{\theta}(t) \right) = \left(1 + \frac{d_f}{d_r} \right) R_f \cos \phi(t) - \frac{J}{d_r} \ddot{\theta}(t)$$

Differentiating the simplified constraint equations from Eq. 6; we get

$$\begin{aligned} u(t) &= \tan^{-1} \phi(t) l \ddot{\theta}(t) - \frac{u(t)}{\cos^2 \phi(t)} \dot{\phi}(t) \\ v(t) &= d_r \ddot{\theta}(t) \end{aligned}$$

Substituting the second equation in Eq. 11 into Eq. 10 and solving for the front reaction force provides us with the following equation.

$$R_f = \frac{md_r^2 + J}{l \cos \phi(t)} \ddot{\theta}(t) + \frac{md_r u(t)}{l \cos \phi(t)} \dot{\theta}(t)$$

12

Using the second equation in Eq. 6 and the third equation in Eq. 7, Eq. 12 can be substituted into the first equation in Eq. 8 to get

$$m \dot{u}(t) = F_d - \frac{md_r^2 + J}{l} \tan \phi(t) \ddot{\theta}(t)$$

13

Solving for $\ddot{\theta}(t)$ from the first equation in Eq. 11 in Eq. 13 yields

$$m \dot{u}(t) = F_d - \frac{md_r^2 + J}{l} \tan \phi(t) \left[\frac{\tan \phi(t)}{l} \dot{u}(t) + \frac{u(t)}{l \cos^2 \phi(t)} \dot{\phi}(t) \right]$$

14

And finally, solving for $u(t)$, we obtain

$$\dot{u}(t) = \frac{l^2}{\alpha(t)} F_d - \frac{m d_r^2 + J}{\alpha(t) \cos^2 \phi(t)} u(t) \dot{\phi}(t)$$

15

$$\text{where } \alpha(t) = m l^2 + (m d_r^2 + J) \tan^2 \phi(t).$$

The steering actuator is a servo motor coupled to the steering shaft. The output is the angle of the steering actuator $\phi(t)$. The actuator system is modeled as an open loop using the following first order system

$$\dot{\phi}(t) = -\frac{1}{\tau_s} \phi(t) + k_s v_s(t)$$

16

where τ_s is the time constant of the servo motor, k_s is the effectiveness of the servo motor, and v_s is the input voltage to the servo. The dynamics of the driving force, F_d , by the servo motor generated are given by the equation:

$$L_m \dot{I}_m(t) + R_m I_m(t) + k_m \omega_m(t) = v_m(t)$$

17

where L is the inductance of the motor, R is the resistance of the motor, k is the back emf of the motor, I is the current to the motor, ω is the angular rate of the motor shaft and v is the input voltage. The torque generated τ_m and the driving force F_d are proportional to the current I_m such that $\tau_m = c_m I_m$ and $F_d = \frac{c_m N}{r} I_m$ respectively. The constant r is the radius of the wheels, which are all the same in this case. Knowing the relationship $u = \frac{r\omega_m}{N}$, where N is the gear ratio between the motor crank and the drive shaft, the following formula can be written:

$$\dot{F}_d = -\frac{R_m}{L_m} F_d(t) - \frac{c_m k_m N}{r L_m} u(t) + \frac{c_m}{r L_m} v_m(t)$$

18

Lastly, a complete set of equations of motion for the JAG fleet vehicle can be written:

$$\begin{aligned}\dot{x}_c(t) &= u(t) \cos \theta(t) - d_r \left(\frac{\tan \phi(t)}{l} \right) u(t) \sin \theta(t) \\ \dot{y}_c(t) &= u(t) \sin \theta(t) + d_r \left(\frac{\tan \phi(t)}{l} \right) u(t) \cos \theta(t) \\ \dot{\theta}(t) &= \frac{\tan \phi(t)}{l} u(t) \\ \dot{u}(t) &= \frac{l^2}{\alpha(t)} F_d - \frac{m d_r^2 + J}{\alpha(t) \cos^2 \phi(t)} u(t) \dot{\phi}(t) \\ \dot{\phi}(t) &= -\frac{1}{\tau_s} \phi(t) + k_s v_s(t) \\ \dot{F}_d &= -\frac{R_m}{L_m} F_d(t) - \frac{c_m k_m N}{r L_m} u(t) + \frac{c_m}{r L_m} v_m(t)\end{aligned}$$

19

4.2 Simulation Introduction

Simulink interfaces with Matlab, which makes available the many computational capabilities that Matlab offers. Matlab is known not only for the creation of user algorithms, but also for data visualization and analysis. All of these features are necessary for the analysis and verification of the simulation. The interface also provides data monitoring and tracking, and the ability to produce visually appealing plots and graphs. Real time plots tracking the vehicle's location are just one of the many features available. Although interfacing with another program introduces another source of error, the Matlab debugging system offers additional trouble shooting techniques.

Simulink programs are created by building diagrams using over 200 different pre-existing blocks to represent each component of a system. The block diagrams consist of the essential parts of each system. In this case, they consist of the dynamics or derived equations for each system.

One example of a component comprising the overall system is the actuators. The actuators for the JAG Fleet are the mechanical devices used to up-fit each vehicle to operate remotely using traditional manual steering, braking and accelerating systems. A brief overview of how one component of the simulation can be constructed follows. Inputs to the Actuator Block consist of a desired steering angle and a desired voltage of the drive motor. These specify at which steering angle the steering wheel needs to be and at which voltage the electric drive motor must be to maintain the vehicles trajectory.

These inputs are supplied from the Controller Block. The outputs of the block consist of the desired steering angle and the desired torque of the drive motor. The outputs are supplied to the Unmanned Ground Vehicle (UGV) Dynamics Block. We provide a detailed discussion of these blocks in the following sections.

The test scenario for this simulation is Fort Benning which has been used in numerous ground vehicle tests and previous simulations. Much of JOUSTER unmanned system simulation is carried out using a Fort Benning model and testing is done at Fort Benning.

4.3 Original Model

The first attempt at modeling the JAG fleet in Simulink was done using the derived Equations of Motion (EOM) given in Eq.'s (19) representing the five degree of freedom model. This simulation was created by Dr. Varham Stepanyan. Although an actual vehicle does not have parallel front wheels due to a different turning radius for each front wheel (Ackerman's model) this assumption is common in control simulations of ground vehicles. Upon successful implementation and thorough accuracy testing of this model, one goal is to determine if this assumption is accurate in the present case. If these tests present sufficient data indicating the assumed five degree of freedom model does not suffice, modifications to the model could, and should, be made at that time. However, these validation tests are not considered in this thesis. They are scheduled for future development plans for the JAG Fleet.

Below is the top level in the Simulink model, or what is referred to as the global, Simulink model. It shows the overall schematic of the system as it was modeled.

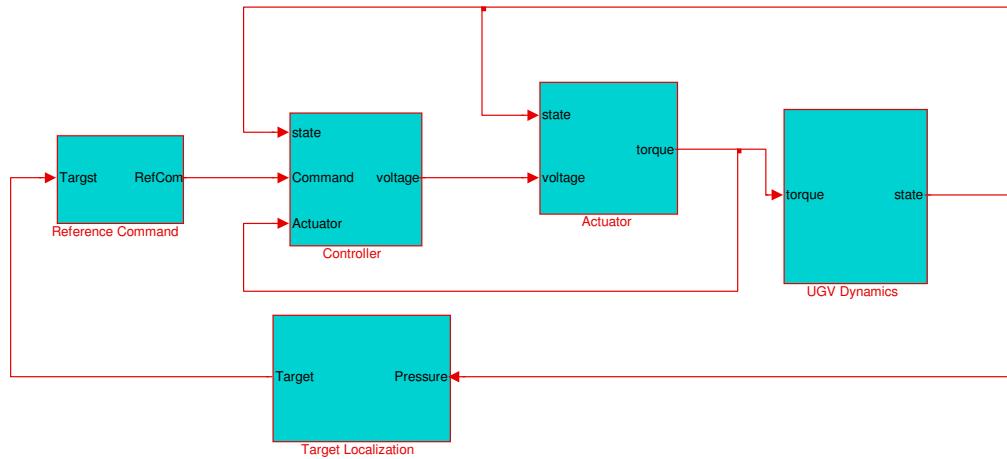


Figure 15 Simulation Block Diagram. Imaged created by author.

Figure 15 is the block diagram for the system. As shown, the system is broken into five components or blocks. Some of these components are comprised of defining equations or dynamics, such as the “Reference Command Block”; others are comprised of sub-components, such as the “Controller Block” which contain other sub-blocks that contain specific defining equations. Each block will be described by stating what the inputs are, what the outputs are and what is happening within the block to generate the outputs. Descriptions will be ordered from left to right, top to bottom, by the convention which Simulink recognizes and interprets the model.

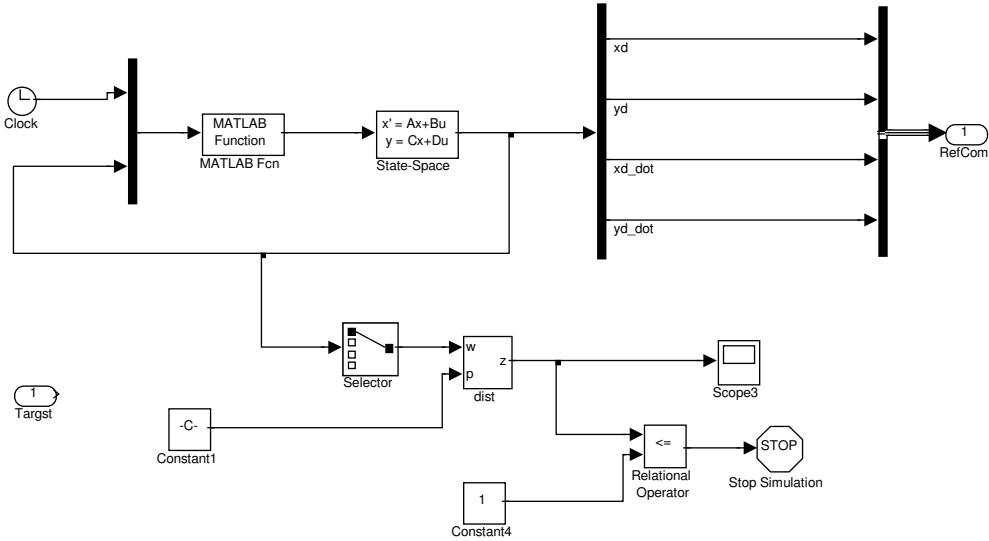


Figure 16 Command Block Diagram. Imaged created by author.

The “Reference Command Block” is known as the path planning block. It recognizes a target for the vehicle and determines its trajectory. It is a dynamical system consisting of a Matlab function, which takes on the current time, the target’s position and the current position of the system and generates control inputs for a linear time invariant (LTI) 4th order system according to some optimal scheme. The description of which is out of the scope of this thesis. The LTI system generates the desired trajectory, given by the coordinates xd and yd , and desired velocity, given by xd_dot and yd_dot . These variables are the outputs of the block. Also, the block contains a stopping mechanism that stops the simulation when the goal, i.e. the targets desired position, is reached.

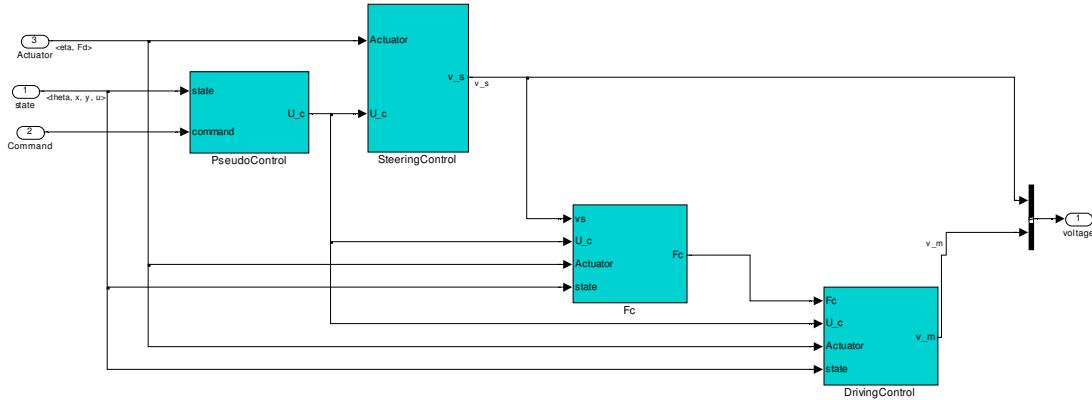


Figure 17 Controller Block. Imaged created by author.

The Controller block takes the supplied data and feeds the Actuator Block to update the state of the vehicles' operating components to maintain the desired trajectory. In other words, it receives an update of the vehicles position and instructs the actuators to make the necessary changes to maintain its course. There are three categories of inputs to the Controller Block. One category contains the states of the vehicle: the x and y position, the forward velocity (denoted as u), and the angular orientation of the vehicle (denoted as θ). The second category of inputs to the Controller Block is the command input. The command input gives the Controller Block the desired position and velocity states of the vehicle (xd, yd) and (xd, yd) respectively. The final category of inputs to the Controller consists of feedback from the Actuator Block, which is described in the following paragraph. The Actuator Block supplies the Controller Block with the steering wheel angle (η) and the driving force (F_d) of the vehicle. The output of the block is a necessary voltage for the steering motor and drive motor. An error measured from the difference between the desired drive force and steering angle to the actual drive force and

steering angle is used to generate a correction signal. The correction signal is necessary to make changes for the vehicle to maintain its desired course. To convert from the inputs described above to the necessary voltage for each motor (drive motor and steering actuator motor), there are many complex mathematical operations and equations being utilized in four blocks. These four blocks consist of the: Pseudo Control Block, Steering Control Block, Fc Block (desired force applied to rear axle), and Driving Control Block.

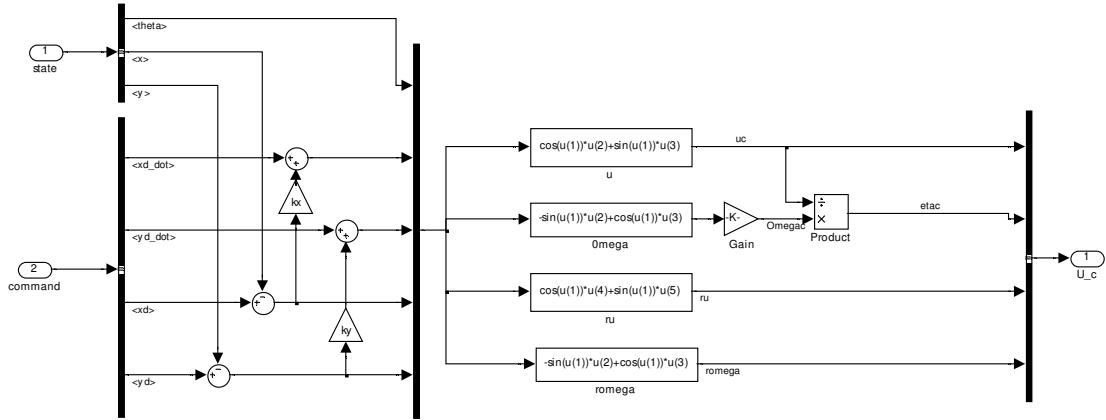


Figure 18 PseudoControl Block. Imaged created by author.

The Pseudo Control Block (PseudoControl) generates a command forward velocity and steering angle, as well as a “rotation translated” forward velocity and orientation angle of the vehicle. The rotation functions limit the vehicles’ turning and velocity by taking into account physical limitations of the vehicle (such as top speed and maximum turning angle). This prevents the vehicle from being instructed to maneuver in a way that is physically impossible. These channels are labeled ‘ U_c ’. The difference is the command functions represent the necessary changes to the vehicles current state to maintain the proper trajectory. The inputs used to generate these signals come from the State and Command inputs described above, both actual and desired position, velocity, as

well as orientation angle of the vehicle. The derivation for the equations in this block can be found in [15] and in [16]

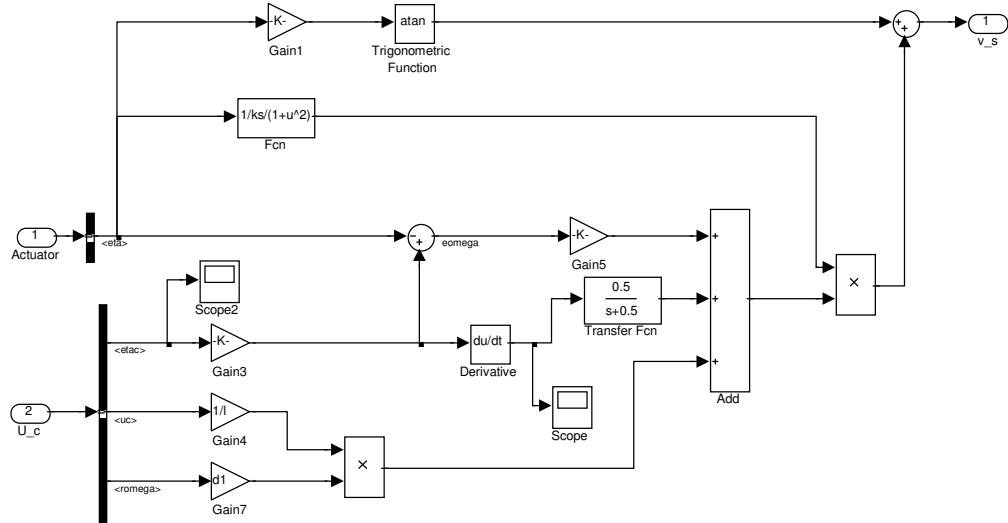


Figure 19 Steering Control Block. Image created by author.

The Steering Control Block generates a voltage command to send to the steering actuator motor. This voltage actuates the motor to change the steering angle to a desired angle, which in turn orients the vehicle as needed.

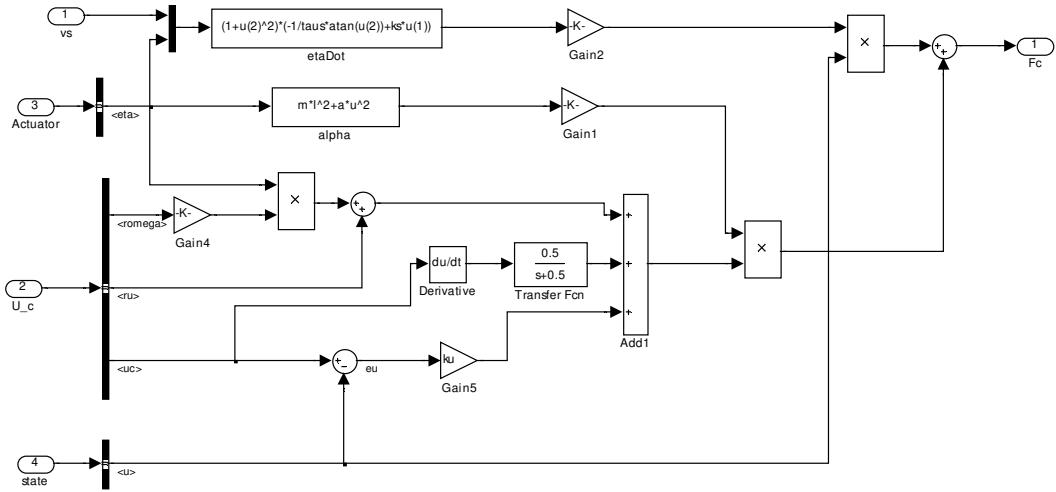


Figure 20 Desired Control Force (F_c) . Imaged created by author.

The Desired Force Block (F_c) generates a desired control force for driving the vehicle. Combining inputs from the steering voltage (vs), steering angle from the Actuator Block (η), and rotation transformed orientation angle of vehicle and velocity ($r\omega$ and ru respectively), command forward velocity (uc) and state velocity (u), is modeled [15]. The output of the block is the command signal for the desired force which is used in the Driving Control Block. The rotation transformed angle of vehicle and velocity are auxiliary signals here, used in the “back-stepping” control utilized in this derivation [15]. One simulation technique used in this block calculates the derivative of the forward command velocity. This generates high frequency error. By incorporating a low pass filter, the error is smoothed out. Although this does introduce a lag time, it does eliminate most of the unwanted high frequency. This is more properly known as a “pseudo-differentiator”.

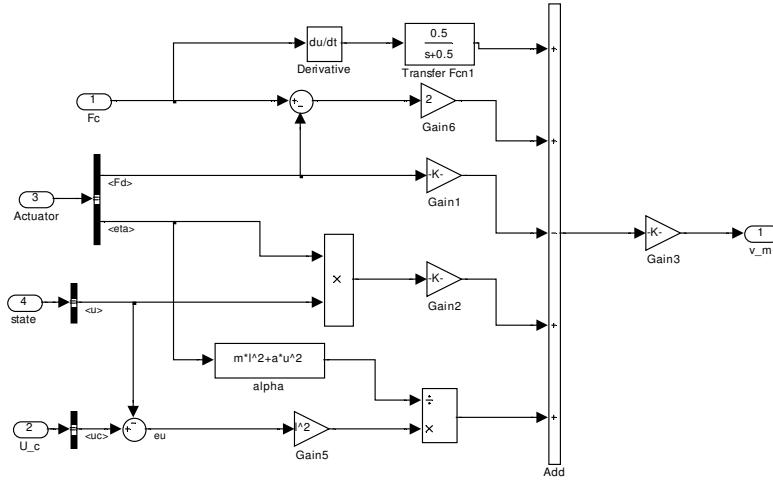


Figure 21 Driving Control Block. Image created by author.

The Driving Control Block generates a voltage for the drive motor and for the steering motor. Inputs are the desired force (F_c), Actuator, state (u) and changing rates of position and steering (forward and angular steering velocity) [15]. The outputs are:

1. The voltages that the steering motor needs to be at to accurately steer the vehicle along its path and
2. The voltage that the drive motor needs to be at to hold the correct forward velocity.

This block was designed to embody the inner control loop. Given a desired trajectory, the equations output commands to maintain the appropriate forward velocity to maintain its course.

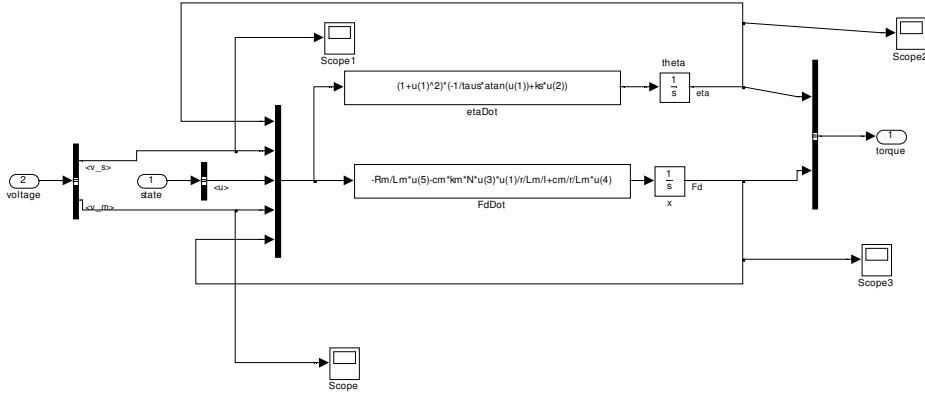


Figure 22 The Actuator Block. Imaged created by author.

The Actuator Block generates a steering angle for the vehicle and a driving force at the wheels of the vehicle from the voltage inputs coming from the Controller Block. The inputs to this block are the desired voltages for the drive motor and steering wheel actuator motor. The time derivative of the steering angle is calculated, and then transformed to the steering angle which is sent to the steering actuator. Also, the time derivative of the drive force (F_d) is calculated and also transformed to supply the motor with the needed drive force [15]. Therefore, the outputs of the system are the steering angle of the wheel and the drive force at the rear axel.

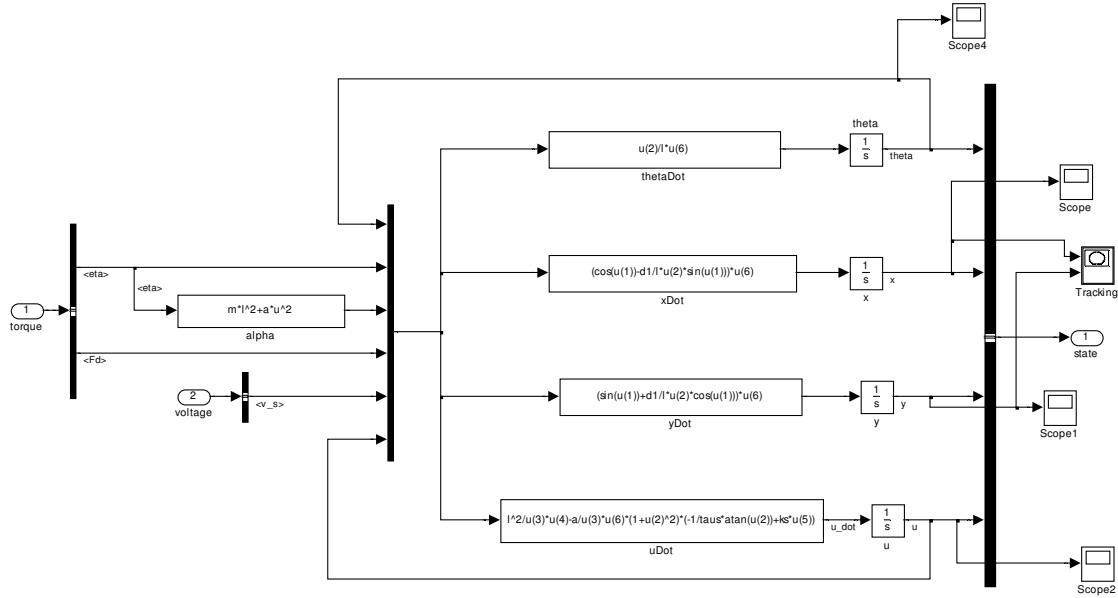


Figure 23 Unmanned Ground Vehicle (UGV) Dynamics. Imaged created by author.

The Unmanned Ground Vehicle (UGV) Dynamics Block represents the actual vehicle. All the other blocks represent sub-systems that enable the vehicle to drive by wire. The steering angle and drive force inputs supply the system of equations in this block to take on the steering wheel angle and the driving force signals. The equations convert them to states of the vehicle: x and y positions, forward velocity and angle of orientation of the vehicle [15]. The outputs of the system define the state of the vehicle for each time step. The vehicle's position is plotted on an X-Y plot as each time step passes, thereby tracking the vehicles progress.

4.4 Proposed Modifications

One contribution to this thesis was to create a more accurate simulation by modifying the subsystem equations in the original simulation. The original simulation was created by Dr. Varham Stepanyan, and assistance was provided in making the proposed modifications. To achieve this goal, modifications were made to the model in the Controller Block and Actuator Block. The up-fitted steering actuator system consists of a servo motor that is coupled to the steering shaft via a timing belt and two pulleys and a motor controller. The motivation behind the proposed modification to the system stemmed from the availability of a tailored controller designed expressly for the selected servo motor. The controller, manufactured by Quicksilver, supplies a simple user interface to control the servo motor. It provides a quick and user friendly way to communicate to the motor and test the mechanical operation of the up-fitted system. It was the availability of this controller that sparked the idea to command a steering angle as compared to using the traditional voltage control to command the servo motor. It is easier to analyze and process a steering angle than have to convert a voltage to the angle in trouble shooting scenarios, etc.

The Controller in the existing model contained a block for the steering control. Its inputs included the steering angle of the wheels and the driving force signals, as well as the desired forward velocity, desired steering wheel angle and rotation translated orientation ($r\omega$) of the vehicle discussed in the Pseudo Controller Block. It outputs a voltage for the steering motor to operate at to make the correct change in steering angle to maintain a desired trajectory [15].

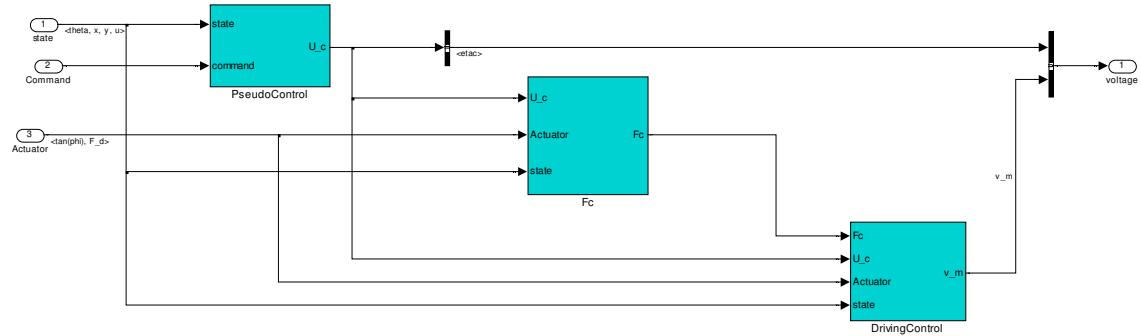


Figure 24 Modified Controller Block. Imaged created by author.

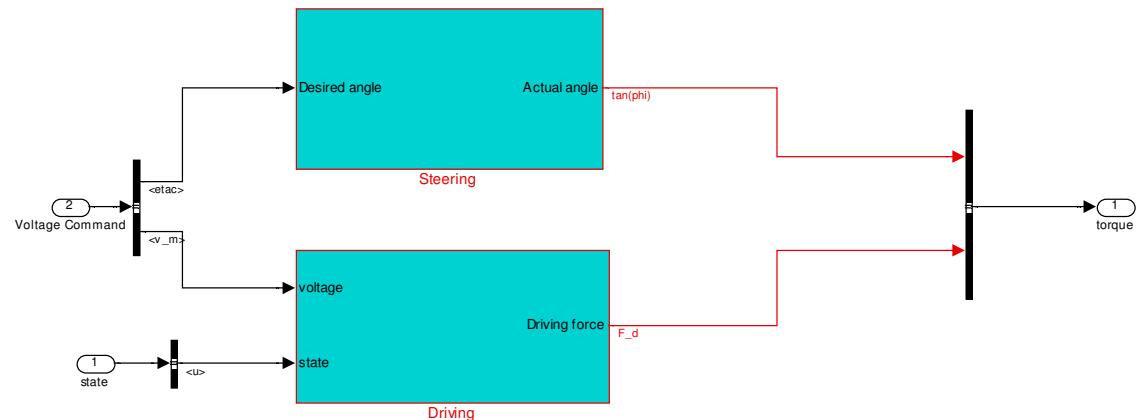


Figure 25 Modified Actuator Block. Imaged created by author.

The proposed change removes the Steering Control block from the Controller and places it into the Actuator Block. In reality, this makes more sense to have it in the Actuator Block for the steering block represents what is happening in the steering motor and controller itself. Also, if the system can accept a steering angle command, it is unnecessary to make a conversion to a voltage command input to the steering actuator.

With that, the Steering Block in the Actuator is simply a pass through signal for now. The output is the steering angle (ϕ). When it comes time for implementation a gain will be placed in there to convert from the steering angle to the angle of the wheels. For now, the simulation works properly as it is now because it will be a linear relationship.

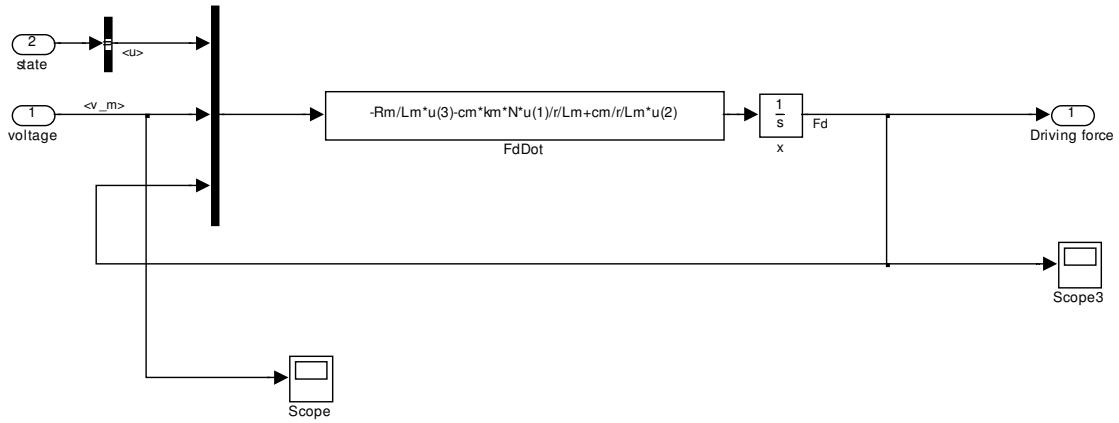


Figure 26 Driving Block inside Actuator. Imaged created by author.

The Actuator still contains the equation for generating the driving force as it did before. This is placed into its own block, labeled Driving now, to keep them separated as shown above. The command or desired voltage of the motor is input, along with the forward velocity signal [15].

4.5 Results

These modifications assume that the steering controller acts a black box. Through a series of accuracy tests on the target and actual steering angle, this approach has deemed sufficiently accurate. The majority of what little error (on a scale of 1% total

error) was found to be due to noise, or with the analog to digital conversion. The series of tests suggested that the error could drift as much as three counts in the inner controller loop. There are 41250 counts per revolution. One revolution is 2π radians. Thus, a three count error corresponds to 4.56×10^{-4} radians. Error on this small of a scale would never be noticed in this application.

The simulation is working as expected. Below is a diagram in Matlab outlining the test site used in the simulation.

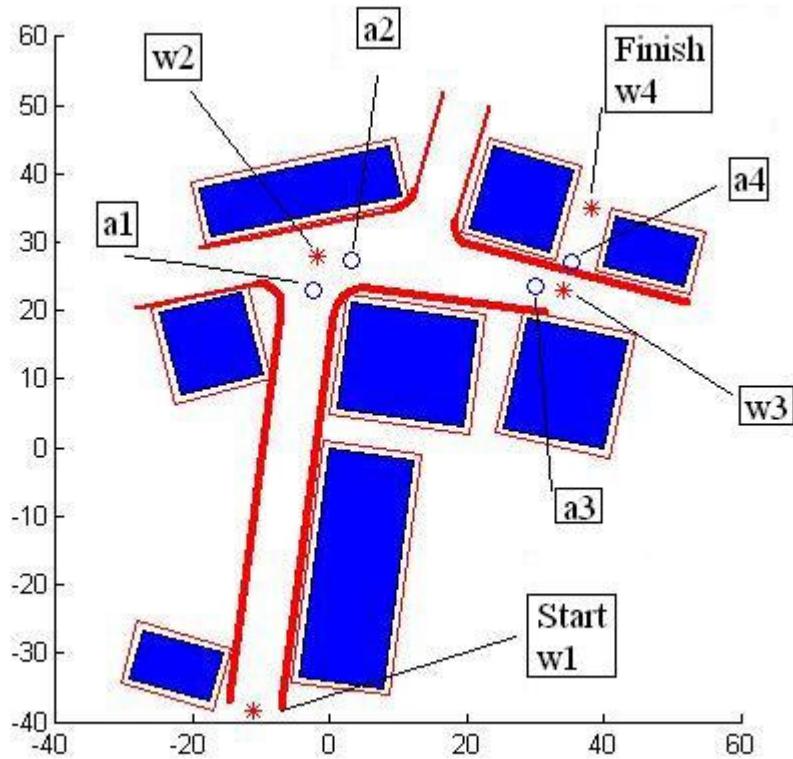


Figure 27 Annotated Fort Benning in Simulation. Imaged created by author.

Below is a figure of the simulation once the vehicle has reached its target location. The vehicle started from its set starting point, w1. This point can be seen in Figure 27. As can be seen, it traversed along the path between the blocks representing buildings,

navigating towards the way points (shown as the stars on the diagram w1-w4) and the target areas (shown as the circles along the path in the diagram a1-a4). The simulation stopped once it has completed, i.e. once the vehicle has reached its final waypoint located at w4.

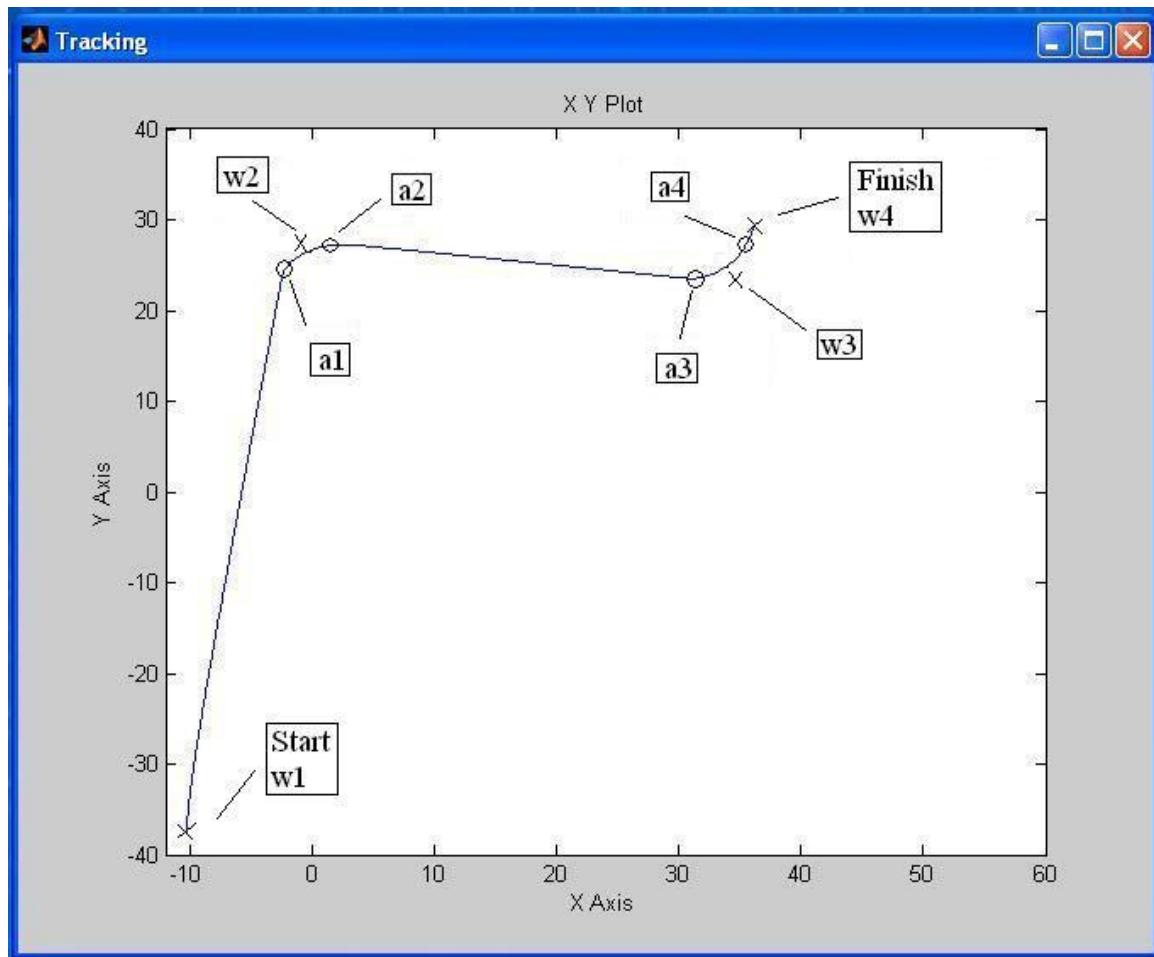


Figure 28 Annotated Tracking Diagram of Vehicles Traversed Path Upon Reaching its Target Location. Imaged created by author.

4.6 Notes on Simulation

As the simulation is constructed now, the initial and final locations are known. In other words the vehicle knows where it starts and where it finishes. Furthermore, in this thesis, a map of the obstacles in the environment is also known. This means a planned path or trajectory for the vehicle is predetermined. A series of waypoints are set up for the vehicle to pass through along this path. This desired trajectory data are fed into the simulation. Currently, the system generates an optimal trajectory utilizing the known start and end points, as well as the map of the environment.

Future work will focus on the implementation of JAUS. At the current state of the project and this thesis, enabling it with JAUS extends beyond the scope and main objectives of this specific thesis. Upon successful mechanical up-fit of the other necessary autonomous systems and simulations of them, the JAUS enabling will begin. As for now, this thesis has been focused on designing an autonomous steering subsystem and an accurate simulation depicting its operation.

Also, future work will focus on the implementation of a few other systems to replace the predetermined start and end points, as well as the map of the environment. The simulation is in the process of modification to implement an acoustic model that will detect a sniper gun shot. Pressure sensors will be mounted on the vehicle platform and will detect changes in pressure due to a sniper gun shot. The pressure sensing system will continuously search for more shots to better localize the general position of the sniper and direct the vehicle towards it. Real time updates from the GPS system will record the vehicles' progress from its initial location towards the location of the jump in

pressure. This system will replace the predetermined start and endpoints of the vehicle trajectory.

To replace the planned trajectory of the vehicle, a suite of vision sensors will also be implemented on the vehicle. Future work entails implementing already existing vision sensors to recognize solid obstacles in the vehicle desired path. Much research along these lines is being done by Virginia Tech in collaboration with the Naval Postgraduate School through the United States Special Operations Command (SOCOM)-sponsored Tactical Network Training program. This program includes quarterly exercises using high data-rate sensors (cameras) to develop and extract the distance from a vehicle to an object in its path. In addition, the cameras are used to determine the object's shape and orientation using shadow sensors and light change detection.

Also, a set of limitations is placed in the simulation to account for vehicle constraints. Maximum steering angle, maximum velocity and acceleration, turning radius, and torque (could be thought of as the product of the driving force and wheel radius) are a few of the constraints that the vehicle has that are accounted for in the simulation. This prevents the algorithm from creating a path or instructing the vehicle to maneuver in a way that could be dangerous or exceed the performance of the vehicle.

With these changes, the outer loop (path-planning) controller will no longer seek a globally optimal trajectory. Currently the simulation assumes that the vehicle knows its initial and final position. Additionally, making the assumption that the obstacles in its test field environment are known *a priori* allows for computation of an optimal and predetermined path. This means the optimal path is already defined for the whole simulation. The objective of the final project is not to have the globally optimal,

predetermined path to the final location. Sensors mentioned above will give information on a changing and/or an unfamiliar environment. The real time position and velocity feedback data will give the vehicle updates on progress. These updates will reset the simulation every time to calculate a new optimum path from its current position to its desired final position. An algorithm will be created based on the Receding Horizon Control (RHC) theory which will utilize these real time updates and regenerate its position and target location. The simulation will be sub-optimal only because small subintervals will be defined over which a local path is calculated optimally, rather than optimize the overall task. However, the vehicle will then not need prior knowledge of its environment and can be deployed anywhere and still complete its task.

Chapter 5 - Conclusion

Autonomous vehicles offer such a valuable contribution to military efforts, as well as are applicable in so many diverse industrial operations, that efforts made in the development of such systems is priceless. The ability to equip a vehicle (or any other manually operational machine for that matter) to conduct potentially unsafe missions will not only save back breaking hours, but possibly human lives. Technology discussed in this thesis is applicable to a diverse pool of systems. It specifically enables an electric car to be steered in a drive-by-wire mode. The implementation of other systems like this for accelerating and braking maneuvers will produce a drive-by-wire vehicle.

A steering system was designed, fabricated and up-fitted to an Ingersol-Rand XRT 810 E Club Car. This thesis has outlined success in meeting goals set forth to alter the existing system as little as possible were met, as well as functionality and proof of concept. The designed steering system was implemented and tested in desired, first-stage conditions. The vehicle can be steered utilizing the supplied software.

Characterization of the supplied controller and software show error small enough to be implemented in a simulation in Simulink and Matlab with dependable results. Tests acknowledge a percent error that is known, and show confidence in the relationship between an input command and its output. More specifically, these tests show the error between a desired angle input to the motor and the actual angle output to the steering shaft.

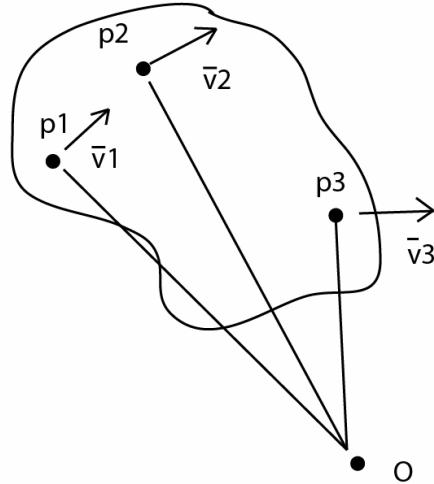
The Matlab/Simulink simulation has been modified to account for the control method of the designed steering system. It replaces the complex method of relating an

input voltage to an electric actuator motor to a steering angle output by utilizing the supplied control software that relates a desired steering angle input to the actual angle output to the steering shaft. This model simplifies the conversion from a voltage to a road wheel angle by incorporating a conversion from steering wheel angle to road wheel angle that can be directly related to a current road wheel angle through a kinematic translation of the vehicle. Furthermore, it is a more accurate model of the system outlined in this thesis.

The first step in converting a vehicle to drive-by-wire has been completed with the design and up-fit of a steering system to a vehicle. Supporting information about the system has been supplied. An operational simulation of this vehicle has also been developed. The stage has been set for future work to utilize this information to complete the conversion of a fully autonomous vehicle.

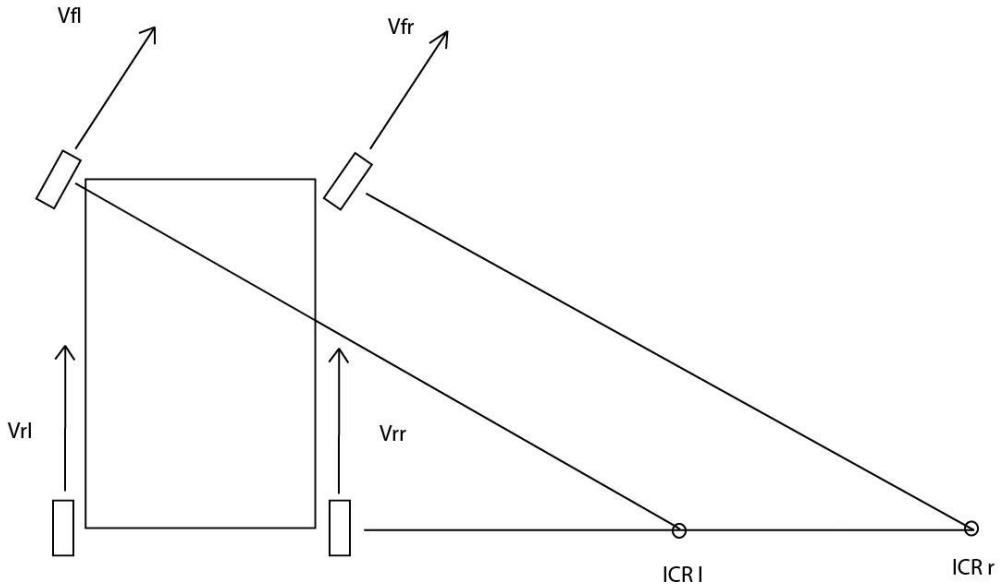
Appendix A – More Detailed Explanation of Parallel Wheel Assumption

By assuming the front and rear wheels are parallel, each pair can be collapsed into one front and rear wheel. This means the model only has one radius of curvature (versus two, one for each wheel in reality), which greatly reduces the complexity of the derivation of equations of motion for the vehicle. In reality, a vehicle with two front wheels separated by a distance must have the inside wheel turn sharper than the outside wheel. Furthermore, it must turn sharp enough so that both front wheels have the same instantaneous centers of rotation. If this was not the case, one wheel would be forced to skid, because both wheels would not be turning about a common point. By Euler's theorem, the rigid body motion at each time instant can be represented by an instantaneous rotation about an instantaneous center of rotation. Please see Appendix Figure 1.



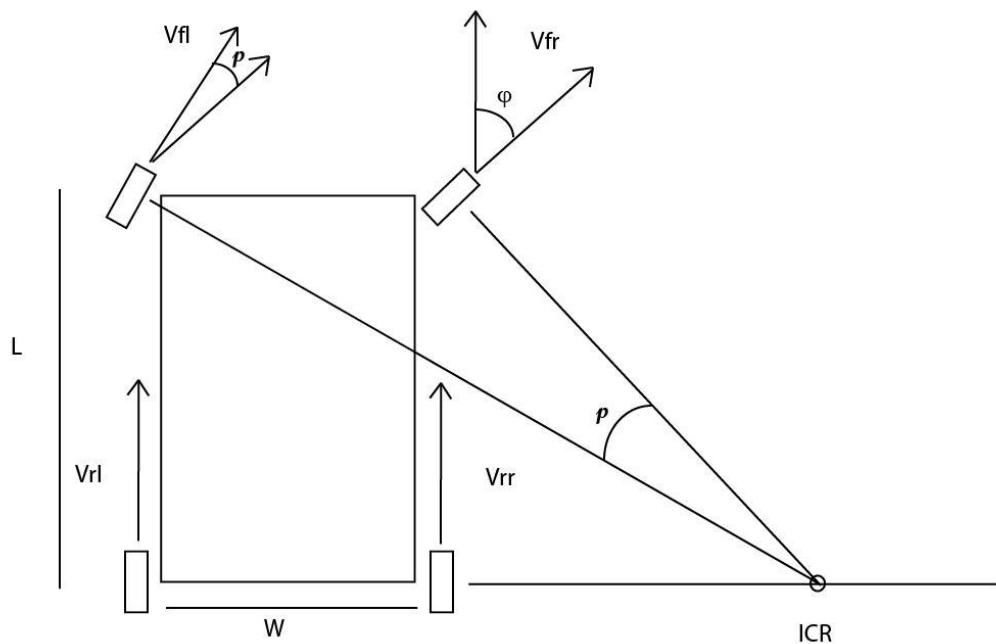
Appendix Figure 1 Instantaneous Rigid Body Rotation about Instantaneous Center of Rotation. Image created by author.

Therefore, the velocity \vec{v}_i of each point p_i is perpendicular to the instantaneous radius of rotation \vec{Op}_i (perpendicular to the line connecting point p_i to the instantaneous center of rotation O). By collapsing the front two wheels into one, there is no need to account for this difference in angle between each front wheel, greatly decreasing the kinematics of the model. This assumption is quite common in autonomous derivations of equations of motion for ground vehicles particularly for this simplification. In the first design stage of this project, this assumption has been accepted. Appendix Figure 2 shows a diagram of an example of a vehicle with parallel front wheels, highlighting the contradiction that it is physically impossible to have a rigid body rotate about two different instantaneous centers of rotation. In the diagram, the velocities at each of the wheels (v_{fl} , v_{fr} , v_{rl} , v_{rr}) and the instantaneous centers of rotation ($ICR l$, $ICR r$) are labeled.



Appendix Figure 2 Parallel Front Wheel Diagram. Imaged created by author.

Below is a figure of a vehicle with non-parallel front wheels. The inside wheel turns sharper to create the common instantaneous center of rotation (ICR). This also creates turning radii of the front wheels, which makes for a more complex derivation of equations of motion for the vehicle. ρ is the difference between the two turning angles, φ is the steering angle, L is the length of the vehicle and W is the width of the vehicle.



Appendix Figure 3 Non-parallel Front Wheel Diagram. Imaged created by author.

Appendix B – Matlab Code for Data Plots and Analysis

```
clear all
clc
% calling time scale
time_scale = xlsread('matlab_data', 'A4:A2621');

%*****NOTE**** full 2 seconds of data recorded every time

%2.5 radians in .5 seconds, return to 0 in .5 seconds
test1 = xlsread('matlab_data', 'C3:D2620');
%2.5 radians in 1 second, return to 0 in 1 second
test2 = xlsread('matlab_data', 'F3:G2620');
%2.5 radians in 5 seconds, return to 0 in 5 seconds - only 2 seconds of
data recorded
test3 = xlsread('matlab_data', 'I3:J2620');

%0.76 radians in 2 seconds
test4 = xlsread('matlab_data', 'L3:M2620');
%1.52 radians in 2 seconds
test5 = xlsread('matlab_data', 'O3:P2620');
%3.04 radians in 2 seconds
test6 = xlsread('matlab_data', 'R3:S2620');
%7.61 radians in 2 seconds
test7 = xlsread('matlab_data', 'U3:V2620');
%11.40 radians in 2 seconds
test8 = xlsread('matlab_data', 'X3:Y2620');

%0 to 0.76 to 0 radians in 1 second
test9 = xlsread('matlab_data', 'AA3:AB2620');
%0 to 1.52 to 0 radians in 1 second
test10 = xlsread('matlab_data', 'AD3:AE2620');
%0 to 3.04 to 0 radians in 1 second
test11 = xlsread('matlab_data', 'AG3:AH2620');
%0 to 7.61 to 0 radians in 1 second
test12 = xlsread('matlab_data', 'AJ3:AK2620');

%min and max values
min1 = min(abs(test1(:,2)-test1(:,1)));
max1 = max(abs(test1(:,2)-test1(:,1)));
mean1 = mean(abs(test1(:,2)-test1(:,1)));
std1 = std(abs(test1(:,2)-test1(:,1)));

min2 = min(abs(test2(:,2)-test2(:,1)));
max2 = max(abs(test2(:,2)-test2(:,1)));
mean2 = mean(abs(test2(:,2)-test2(:,1)));
std2 = std(abs(test2(:,2)-test2(:,1)));

min3 = min(abs(test3(:,2)-test3(:,1)));
max3 = max(abs(test3(:,2)-test3(:,1)));
mean3 = mean(abs(test3(:,2)-test3(:,1)));
std3 = std(abs(test3(:,2)-test3(:,1)));

min4 = min(abs(test4(:,2)-test4(:,1)));
max4 = max(abs(test4(:,2)-test4(:,1)));



```

```

mean4 = mean(abs(test4(:,2)-test4(:,1)));
std4 = std(abs(test4(:,2)-test4(:,1)));

min5 = min(abs(test5(:,2)-test5(:,1)));
max5 = max(abs(test5(:,2)-test5(:,1)));
mean5 = mean(abs(test5(:,2)-test5(:,1)));
std5 = std(abs(test5(:,2)-test5(:,1)));

min6 = min(abs(test6(:,2)-test6(:,1)));
max6 = max(abs(test6(:,2)-test6(:,1)));
mean6 = mean(abs(test6(:,2)-test6(:,1)));
std6 = std(abs(test6(:,2)-test6(:,1)));

min7 = min(abs(test7(:,2)-test7(:,1)));
max7 = max(abs(test7(:,2)-test7(:,1)));
mean7 = mean(abs(test7(:,2)-test7(:,1)));
std7 = std(abs(test7(:,2)-test7(:,1)));

min8 = min(abs(test8(:,2)-test8(:,1)));
max8 = max(abs(test8(:,2)-test8(:,1)));
mean8 = mean(abs(test8(:,2)-test8(:,1)));
std8 = std(abs(test8(:,2)-test8(:,1)));

min9 = min(abs(test9(:,2)-test9(:,1)));
max9 = max(abs(test9(:,2)-test9(:,1)));
mean9 = mean(abs(test9(:,2)-test9(:,1)));
std9 = std(abs(test9(:,2)-test9(:,1)));

min10 = min(abs(test10(:,2)-test10(:,1)));
max10 = max(abs(test10(:,2)-test10(:,1)));
mean10 = mean(abs(test10(:,2)-test10(:,1)));
std10 = std(abs(test10(:,2)-test10(:,1)));

min11 = min(abs(test11(:,2)-test11(:,1)));
max11 = max(abs(test11(:,2)-test11(:,1)));
mean11 = mean(abs(test11(:,2)-test11(:,1)));
std11 = std(abs(test11(:,2)-test11(:,1)));

min12 = min(abs(test12(:,2)-test12(:,1)));
max12 = max(abs(test12(:,2)-test12(:,1)));
mean12 = mean(abs(test12(:,2)-test12(:,1)));
std12 = std(abs(test12(:,2)-test12(:,1)));

figure(1)
clf
%subplot(1,2,1)
%overlay actual data on experimental data
plot(time_scale,test1(:,1),time_scale,test1(:,2)),title('2.5 Radians in
0.5 Seconds','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 2.5])
%plot difference absolute value of experimental-true
%subplot(1,2,2)

```

```

figure(2)
plot(time_scale,(test1(:,2)-
test1(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y1min = min((test1(:,2)-test1(:,1)))*1.1;
y1max = max((test1(:,2)-test1(:,1)))*1.1;
axis([0 2 y1min y1max])

figure(3)
clf
%subplot(1,2,1)
plot(time_scale,test2(:,1),time_scale,test2(:,2)),title('2.5 Radians in
1 Second','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 2.5])
%plot difference absolute value of experimental-true
figure(4)
%subplot(1,2,2)
plot(time_scale,(test2(:,2)-
test2(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y2min = min((test2(:,2)-test2(:,1)))*1.1;
y2max = max((test2(:,2)-test2(:,1)))*1.1;
axis([0 2 y1min y1max])

figure(5)
clf
%subplot(1,2,1)
plot(time_scale,test3(:,1),time_scale,test3(:,2)),title('2.5 Radians in
5 Seconds','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 2.5])
%plot difference absolute value of experimental-true
figure(6)
%subplot(1,2,2)
plot(time_scale,(test3(:,2)-
test3(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y3min = min((test3(:,2)-test3(:,1)))*1.1;
y3max = max((test3(:,2)-test3(:,1)))*1.1;
axis([0 2 y3min y3max])

figure(7)
clf
%subplot(1,2,1)

```

```

plot(time_scale,test4(:,1),time_scale,test4(:,2)),title('0.76 Radians
in 2 Seconds','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 11.4])
%plot difference absolute value of experimental-true
%subplot(1,2,2)
figure(8)
plot(time_scale,(test4(:,2)-
test4(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y4min = min((test4(:,2)-test4(:,1)))*1.1;
y4max = max((test4(:,2)-test4(:,1)))*1.1;
axis([0 2 y3min y3max])

figure(9)
clf
%subplot(1,2,1)
plot(time_scale,test5(:,1),time_scale,test5(:,2)),title('1.52 Radians
in 2 Seconds','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 11.4])
%plot difference absolute value of experimental-true
%subplot(1,2,2)
figure(10)
plot(time_scale,(test5(:,2)-
test5(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y5min = min((test5(:,2)-test5(:,1)))*1.1;
y5max = max((test5(:,2)-test5(:,1)))*1.1;
axis([0 2 y5min y5max])

figure(11)
clf
%subplot(1,2,1)
plot(time_scale,test6(:,1),time_scale,test6(:,2)),title('3.04 Radians
in 2 Seconds','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 11.4])
%plot difference absolute value of experimental-true
%subplot(1,2,2)
figure(12)
plot(time_scale,(test6(:,2)-
test6(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot

```

```

y6min = min((test6(:,2)-test6(:,1)))*1.1;
y6max = max((test6(:,2)-test6(:,1)))*1.1;
axis([0 2 y6min y6max])

figure(13)
clf
%subplot(1,2,1)
plot(time_scale,test7(:,1),time_scale,test7(:,2)),title('7.61 Radians
in 2 Seconds','fontsize',14), xlabel('time
(sec)', 'fontsize',14), ylabel('Angle
(rad)', 'fontsize',14), set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 11.4])
%plot difference absolute value of experimental-true
%subplot(1,2,2)
figure(14)
plot(time_scale,(test7(:,2)-
test7(:,1))),title('Difference','fontsize',14), xlabel('time
(sec)', 'fontsize',14), ylabel('(Exp-True)
(rad)', 'fontsize',14), set(gca,'fontsize',14)
%setting axis limits for plot
y7min = min((test7(:,2)-test7(:,1)))*1.1;
y7max = max((test7(:,2)-test7(:,1)))*1.1;
axis([0 2 y7min y7max])

figure(15)
clf
%subplot(1,2,1)
plot(time_scale,test8(:,1),time_scale,test8(:,2)),title('11.40 Radians
in 2 Seconds','fontsize',14), xlabel('time
(sec)', 'fontsize',14), ylabel('Angle
(rad)', 'fontsize',14), set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 11.4])
%plot difference absolute value of experimental-true
%subplot(1,2,2)
figure(16)
plot(time_scale,(test8(:,2)-
test8(:,1))),title('Difference','fontsize',14), xlabel('time
(sec)', 'fontsize',14), ylabel('(Exp-True)
(rad)', 'fontsize',14), set(gca,'fontsize',14)
%setting axis limits for plot
y8min = min((test8(:,2)-test8(:,1)))*1.1;
y8max = max((test8(:,2)-test8(:,1)))*1.1;
axis([0 2 y8min y8max])

figure(17)
clf
%subplot(1,2,1)
plot(time_scale,test9(:,1),time_scale,test9(:,2)),title('0.76 Radians
in 1 Second','fontsize',14), xlabel('time
(sec)', 'fontsize',14), ylabel('Angle
(rad)', 'fontsize',14), set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 7.61])
%plot difference absolute value of experimental-true

```

```

%subplot(1,2,2)
figure(18)
plot(time_scale,(test9(:,2)-
test9(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y9min = min((test9(:,2)-test9(:,1)))*1.1;
y9max = max((test9(:,2)-test9(:,1)))*1.1;
axis([0 2 y9min y9max])

figure(19)
clf
%subplot(1,2,1)
plot(time_scale,test10(:,1),time_scale,test10(:,2)),title('1.52 Radians
in 1 Second','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 7.61])
%plot difference absolute value of experimental-true
%subplot(1,2,2)
figure(20)
plot(time_scale,(test10(:,2)-
test10(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y10min = min((test10(:,2)-test10(:,1)))*1.1;
y10max = max((test10(:,2)-test10(:,1)))*1.1;
axis([0 2 y10min y10max])

figure(21)
clf
%subplot(1,2,1)
plot(time_scale,test11(:,1),time_scale,test11(:,2)),title('3.04 Radians
in 1 Second','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 7.61])
%plot difference absolute value of experimental-true
%subplot(1,2,2)
figure(22)
plot(time_scale,(test11(:,2)-
test11(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y11min = min((test11(:,2)-test11(:,1)))*1.1;
y11max = max((test11(:,2)-test11(:,1)))*1.1;
axis([0 2 y11min y11max])

figure(23)
clf

```

```

plot(time_scale,test12(:,1),time_scale,test12(:,2)),title('7.61 Radians
in 1 Second','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 7.61])
%plot difference absolute value of experimental-true
%subplot(1,2,2)
figure(24)
plot(time_scale,(test12(:,2)-
test12(:,1))),title('Difference','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('(Exp-True)
(rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y12min = min((test12(:,2)-test12(:,1)))*1.1;
y12max = max((test12(:,2)-test12(:,1)))*1.1;
axis([0 2 y12min y12max])

figure(25)
clf
plot(time_scale,test3(:,1),time_scale,test3(:,2)),title('2.5 Radians in
5 Seconds (only 2 seconds recorded)','fontsize',14),xlabel('time
(sec)','fontsize',14),ylabel('Angle
(rad)','fontsize',14),set(gca,'fontsize',14)
legend('Actual Position','Target Position')
axis([0 2 0 2.5])

figure(26)
clf
plot(time_scale,(test3(:,2)-test3(:,1))),title('Difference Input vs.
Output','fontsize',14),xlabel('time (sec)','fontsize',14),ylabel('(Exp-
True) (rad)','fontsize',14),set(gca,'fontsize',14)
%setting axis limits for plot
y1min = min((test3(:,2)-test3(:,1)))*1.1;
y1max = max((test3(:,2)-test3(:,1)))*1.1;
axis([0 2 y1min y1max])

```

References

- [1] Hughes, R., "Elbit unveils autonomous ground vehicle," *Jane's Defense Weekly*, October 2005.
- [2] Touchton, R., Kent, D., Galluzzo, Tom., Crane, C., Armstrong, D., Flann, N., Wit, J., Adsit, P., "Planning and Modeling Extensions to the Joint Architecture for Unmanned Systems (JAUS) for Application to Unmanned Ground Vehicles," Proceedings of SPIE – The International Society for Optical Engineering, v 5804, Unmanned Ground Vehicle Technology VII, 2005.
- [3] Gomi, T., Ide, K., Matsuo, H., "Development of a fully autonomous ground vehicle (FAGV)," Proceedings of IEEE Intelligent Vehicles Symposium, October 24-26 1994.
- [4] Albus, J., Barbera, A., Schlenoff, C., "Progress and Prospects for Collaborative Tactical Behaviors for Autonomous Ground and Air Vehicles," AUVSI's Unmanned Systems North America, Association for Unmanned Vehicle Systems International, 2004.
- [5] Webb, T., Prazenica, R., Kurdila, A., Lind, R., "Vision-based State Estimation for Autonomous Micro Air Vehicles," Journal of Guidance, Control, and Dynamics, v 30, n 3, May/June, 2007, American Institute of Aeronautics and Astronautics Inc.
- [6] Sasaki, M., Ogawa, K., Koike, T., Hosotani, N., "Development of Autonomous Underwater Vehicle for Observation of Water in Lake Biwa," *Mitsui Zosen Technical Review*, n 170, Mitsui Eng. & Shipbuilding Co, June 2000.
- [7] Stokey, R., Roup, A., von Alt, C., Allen, B., Norrester, N., Austin, T., Goldsborough, R., Purcell, M., Jaffre, F., Packard, G., Kukulya, A., "Development of the REMUS 600 Autonomous Underwater Vehicle," *Oceans 2005*, IEEE Cat. No. 05CH37711), 2005.
- [8] Department of Defense Joint Robotics Enterprise, <http://www.jointrobotics.com/>.
- [9] Joint Architecture for Unmanned Systems (JAUS) Reference Architecture, Version 3.0, <http://www.jauswg.org/>.
- [10] Donoghue, P., Larson, G., Dondoe, S., "Extending Technology to Drive-by-Wire Control," Proceedings of the 50th Annual Meeting – Institute of Navigation, 1994.
- [11] Zetterstrom, S., "Electromechanical Steering, Suspension, Drive, and Brake Modules," *56th IEEE Vehicular Technology Conference*, v 56, n 3, 2002.
- [12] Johnson, P., Chapman, K., Bay, J., "Navigation of an Autonomous Ground Vehicle using the Subsumption Architecture," Proceedings of the SPIE – The International Society for Optical Engineering, v 2903, 1997.
- [13] Ahmadi, J., Ghaffri, A., and Kazemi, R., "Fuzzy logic based vehicle stability enhancement through combined differential braking and active front steering," Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference - DETC2005: 5th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, p 2417-2423, September 24-28 2005.
- [14] Ingersol Rand Website and Online Technical Documents, <http://www.clubcar.com/>.
- [15] Varham, S., "Dynamics and Controls of the Car-Like Mobile Robot," 2007.
- [16] Kurdila, A., Roan, M., Johnson, M., Stilwell, D., Woosley, C., Schaub, H., "Heterogeneous Systems Technology for Perimeter Security of Conventional and Nuclear Weapons Facilities," Airbase Technologies: Solicitation Number MAATYN07001, Submitted to Department of the Air Force Air Education and Training Command, February, 2007.