UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Navigation and Control of an Unmanned Surface Vessel

by

L.E.V. Kingwill
20725728

Progress Report

Study leader: Prof Jaco Versfeld

July 2022

PLAGIARISM DECLARATION

I have read and understand the Stellenbosch University Policy on Plagiarism and the definitions of plagiarism and self-plagiarism contained in the Policy [Plagiarism: The use of the ideas or material of others without acknowledgement, or the re-use of one's own previously evaluated or published material without acknowledgement or indication thereof (self-plagiarism or text-recycling)].

I also understand that direct translations are plagiarism, unless accompanied by an appropriate acknowledgement of the source. I also know that verbatim copy that has not been explicitly indicated as such, is plagiarism.

I know that plagiarism is a punishable offence and may be referred to the University's Central Disciplinary Committee (CDC) who has the authority to expel me for such an offence.

I know that plagiarism is harmful for the academic environment and that it has a negative impact on any profession.

Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully (acknowledged); further, all verbatim copies have been expressly indicated as such (e.g. through quotation marks) and the sources are cited fully.

I declare that, except where a source has been cited, the work contained in this assignment is my own work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

I declare that have not allowed, and will not allow, anyone to use my work (in paper, graphics, electronic, verbal or any other format) with the intention of passing it off as his/her own work.

I know that a mark of zero may be awarded to assignments with plagiarism and also that no opportunity be given to submit an improved assignment.

Signature: ....
L.E.V. Kingwill
20725728

Date: ....

EXECUTIVE SUMMARY

Student: L.E.V Kingwill

Title of Project
Navigation and Control of an Unmanned Surface Vessel.
Objectives
The development of an independent navigation and control system that can be implemented on an unmanned surface vessel that uses electrical thrusters for propulsion and steering.
What is new in this project?
A new control system is going to be created to control the power to the thrusters and thereby steer the vessel. Building on this a navigation system will be created so that the vessel can navigate to a designated point autonomously.
If the project is successful, how will it make a difference?
With a successful navigation and control system, the system could be moved to vessels with better range and seafaring ability and these unmanned vessels can be used for research data collection, patrolling and search and rescue.
What contributions have/will other students made/make?
N.A.
Which aspects of the project will carry on after completion and why
For the vessel to be completely autonomous, a further project should add an obstacle avoidance system. This will be beneficial to avoid other sea vessels as well as fixed obstacles such as rocks and shore.
What arrangements have been/will be made to expedite contiuation?
All the research and project documents will be archived with the university and the code-base will be thoroughly commented for ease of understanding.

Student	Date	Lecturer

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

As technology has improved over the years, processes and systems have become more automated. Initially factories were replacing manual labour with automated machines but recently companies have been investigating self-driving cars and trucks. All over industries tasks are being automated or done remotely with fewer human involvement.

The ocean is the perfect area for unmanned surface vessels (USV) to be used as many of the issues faced with autonomous land vehicles such as self-driving cars are mitigated by open water. On the open water one gets a 360Â° of the surroundings of the vehicle and although there can still be high volumes of traffic in certain areas such as commercial shipping lanes, due to the expanse of the ocean these high traffic areas are avoidable. Finally, and probably the most desirable mitigating factor is that where a surface vehicle would need to look where the road surface is to follow it, an ocean vessel can move directly from point to point on any piece of water.

In South Africa there is a growing need to USVs with regards to ocean research and conservation. There has been a growing use of acoustic sensory systems to track dolphins and whales around the world. By combining this with the technology of USVs, a far larger area can be surveyed.

1.2 Objectives

This project will focus on the navigation and propulsion control of the USV. This is the building block of the USV upon which a future project can build by adding an obstacle avoidance system or renewable power sources to keep the USV operational for longer. This project will have the following objectives:

1. Design and manufacture an electric surface vessel.

2. Designing and manufacturing the control system that will give a pilot manual control over the electric surface vessel.

3. Building on the manual control and implementing navigation control so that the electric surface vessel.

1.3 Motivation

Currently the marine community is using these acoustic systems as stationary systems. By using the USV in conjunction with the USV the area that is studied can be greatly increased with fewer acoustic platforms as have been used in passed projects. Furthermore, the technology can be adapted for use in other industries such sonar surveying, defence and search and rescue. The use of USVs is becoming more prominent as a USV can be cheaper to operate and therefore organisations can either save costs in the case of sonar and acoustic research or in the case or marine patrols and search and rescue, USVs can be used to fill up the ranks of vessels and close the possible.

The tasks previously mentioned are often time consuming and the crew of the assigned vessel need time to rest whereas a fully autonomous USV can operate constantly, stopping only to replenish its energy source and with further developments such as solar charging, USVs could begin to operate indefinitely, having to only come in for services or if there is a problem with the system.

Chapter 2

Literature Review

2.1 GPS

2.1.1 History of GPS

Global Positioning System (GPS) is an everyday thing in our lives today and has become a luxury that most take for granted. There is GPS in our phones, laptops and even cars. We are using it to find directions on our commutes, hail taxis or ride shares and even for recreational sport, tracking how far we travelled.

Figure 2.1: Early GPS receivers were large, heavy devices. (USAF) (1978)

The origins of GPS or rather any global satellite navigation system begins with the space race. It starts, in 1957, with the first satellite to successfully orbit the earth, the Russian satellite Sputnik. During its orbiting flight of the earth, Sputnik was emitting a radio signal which could be picked up on earth. During this orbit Scientists from John Hopkins University in America were monitoring the radio signals emitted by the Sputnik satellite when they saw the Doppler Effect in action with the radio signals, as the satellite drew closer, the radio signal frequency increased and vice versa. These scientists theorized that if they could determine the location of the satellite based on its signal frequency, the opposite would also be true, they could determine the location of a receiver on the ground given the satellites location. Aerospace (2021)

The first instance of a global satellite navigation system was the Transit. It was developed in 1958 by the Advanced Research Projects Agency and the first satellite was launch in 1960. The Transit satellites were mostly used by the military, specifically the Navy's missile submarines. The program was transferred to the Navy during the mid-1960s. During this time there were further Transit satellites launched and by 1968 the entire constellation of Transit satellites was operational, a total of 36 satellites. Aerospace (2021)

There was plenty of other research that was being conducted around the same time to improve on the current Transit. One such researcher was Phillip Diamond. Diamonds concept, from his study in 1963, lead to the Air Force forming a new satellite navigation program which he called 621-B. Further studies were undertaken by James Woodford and Hideyoshi Nakamura, which completed in 1966, proposed using four satellites. The use of four satellites would mean that the receivers no longer needed to be equipped with high-accuracy clocks. This was the first step in reducing the size and cost of the receivers. Aerospace (2021)

There was a range of technological advancements that help progress the satellite navigation systems such as new bandwidth utilization techniques, advancements in computer and the introduction of solid-state microprocessors. These technological advancements helped reduce the size and weight of the GPS receivers to what we now know today, figure 2.1 shows how large and cumbersome the early GPS receivers were. However, one significant technological advancement was the development of atomic clocks. This development led to another satellite navigation system known as Timation (Time Navigation). The third of three Timation satellites launched in 1974, became the first satellite equipped with an atomic clock, the previous two contained crystal oscillator clocks. The use of the atomic clock led to vast improvements in the accuracy of the navigation system and provided three-dimensional location coverage. Aerospace (2021)

There were now three satellite navigation systems, and so when in the 1970s, the Department of Defence wanted a robust and stable system, the project team developed a new concept by cherry-picking the best aspects of all three, Transit, Timation and 621-B. This system was designated, Navigation System with Timing and Ranging (NAVSTAR), this was later changed to GPS I, the precursors to the GPS system we know today. The first NAVSTAR satellite was launched in 1978 and further satellites were launched in the following years, the system reaching its fully operational state with 24 satellites in 1993. Mai (2017)

Although the satellite navigation systems were operational and orbiting the earth, they were still used mostly by the military and the receivers were expensive. However, this began to change in 1983 when President Ronald Reagan authorized commercial airlines use of the NAVSTAR system. This was the start of civilian use of GPS. His (2011)

2.1.2 Modern GPS

The cost of GPS receivers began to decrease in the late-1990s, early-200s, the first cell phone containing GPS technology was released in 1999. The cost reduction can be attributed to the American government approving more non-military singnals as well as the technological advances in processors that was leading to cheaper processing chips. And naturally from the cheaper access, GPS use began to grow and putting more tax on the system which although upgraded to GPS II was not equipped to handle the modern requirments. In 2000 a plan was formed to add new signals to satellites that had not yet been launched in order to handle the increased use. Furhtermore, a new system was to be developed, GPS III, that could fully meet the modern requirements. The first of the GPS III satellites was launched in 2018 with a couple more in the following years and the remaining 6 to be launched by 2023. Aerospace (2021)

2.1.3 How GPS works

There are a total of 31 GPS satellites currently sitting in a medium earth orbit. These are the satellites that are sending the radio signals that a GPS receiver can use to determine its location.

The signal that the satellites broadcast has a range of information that is used

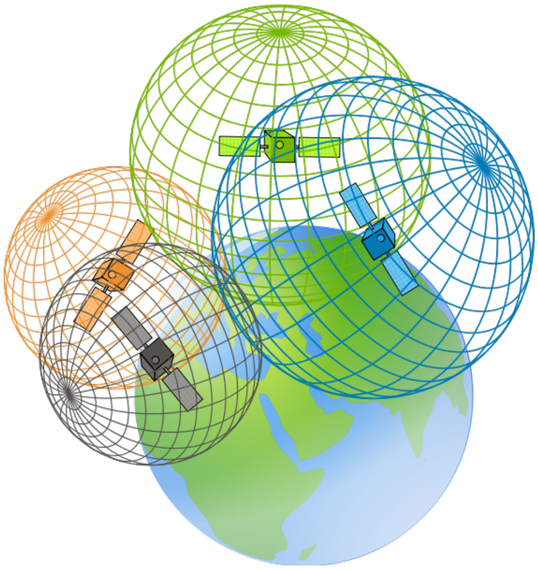

Figure 2.2: Distance spheres around each satellite intersect at one point

by receivers, this information contains data needed to determine the location of the satellite as well as the time that the signal broadcast, using the satellites atomic clock. Based on the time taken for the signal to reach the receiver and corrected for propagation delays or delays from the signal passing through the ionosphere and troposphere, the receiver can calculate the distance between itself and the satellite. This creates a sphere around the satellite upon which the receiver must lie. By adding in a second and third satellite and their distance spheres, there will be only two points of intersection between the three spheres. The one will be the receiver's location, while the other will be impossible location in space. However, to accurately calculate the distance, the receiver would have to have a synchronized atomic clock to determine exactly how long the signal takes to reach it. As it was mentioned earlier, highly accurate clocks were taken out of the receivers by adding a measurement from a fourth satellite to ensure that the distance calculation is accurate. Figure 2.2 illustrates the concept of the distance spheres and their intersection being the location of the GPS receiver. (Federal Aviation Administration)

2.2 Digital Compass

Compasses have been used extensively over the past centuries for navigating, surveying, and map-making. The compass is thought to have been in use from around

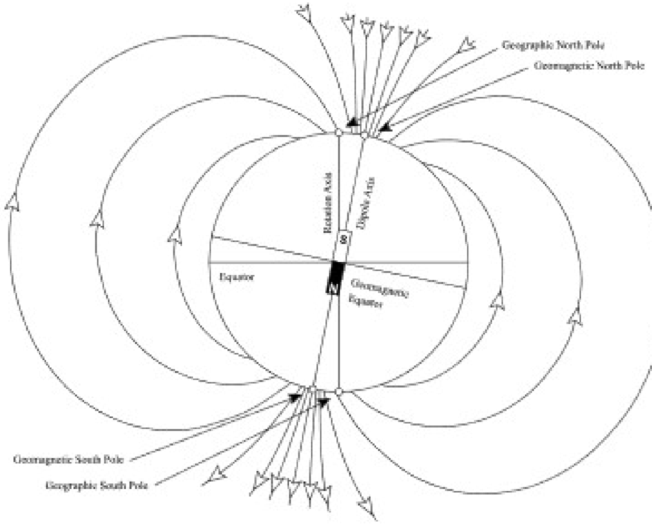

Figure 2.3: The differentiation between magnetic and true north

the 12th century in Europe and possibly earlier in east Asia Jones (2019). Although as many things have over the years been digitalized, so has the compass. The digital compass uses a technology called magneto-induction. This allows the digital compass to electronically detect the earth's magnetic field. Being as sensitive as it is an embedded microcontroller is needed to filter out any magnetic fields from ferro-magnetic materials or other electrical systems that are creating a magnetic field. (Advanced Safety Devices) (2013)

2.2.1 What is magnetic north

True north is always fixed and is the direction that is directly in line with the north pole. However, compasses do not point to true north, they point to magnetic north. This is because a compass aligns itself with the magnetic field caused by the earth's magnetic core. The distinction between true north and the magnetic field at magnetic north is shown in figure 2.1. To further complicate the matter however, the earth's magnetic core experiences changes and these cause small shifts in the magnetic field around the earth. Jones (2019)

2.3 PWM

Pulse Width Modulation (PWM) is a technique of using a digital signal to represent an analogue signal which is used to control analogue systems. The cost of switching a digital circuit between on (high) and off (low) is a cheaper alternative

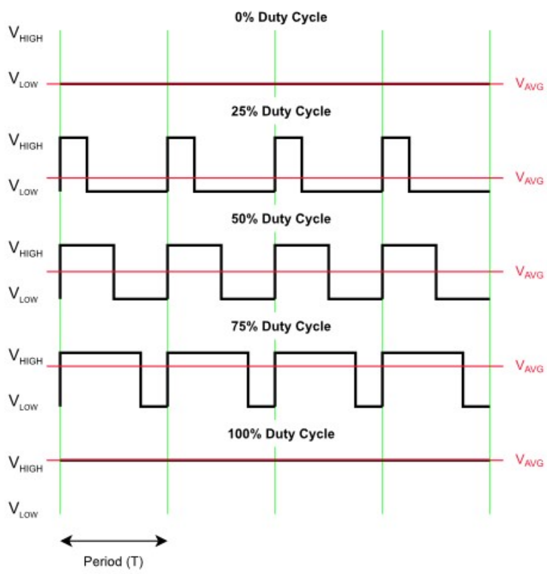

Figure 2.4: Duty Cycle of PWM Signal

to creating an analogue circuit that will incur not drift over time. These PWM signals are mostly used in speed control of DC motors or controlling the brightness of lightbulbs. Christ and Wernli (2014)

PWM is a digital signal that is switched between high and low, leading to the generation of a square wave signal. The time that the signal goes high can be modulated to vary the power delivered to the system. Typically, microcontrollers are used to generate and control the PWM to power an external system. There are a few signal parameters that will be highlighted in this explanation of a PWM signal. The signal amplitude, this is the maximum voltage that can be supplied to the external system. If the microcontrollers output voltage is insufficient for the external system, the signal can be passed through an amplifying circuit to provide the required voltage. Secondly is the signal period, and therefore the frequency as they are inversely proportional, and is the total time for one signal wave to propagate. The frequency is set depending on the requirements of the system, but this frequency will be needed later to help with the calculation of the duty cycle. The duty cycle is the final parameter. The duty cycle is the ratio between the time the signal is high and the time the signal is low. It is always a value between 0 and 1, however, the duty cycle is often expressed as a percentage.

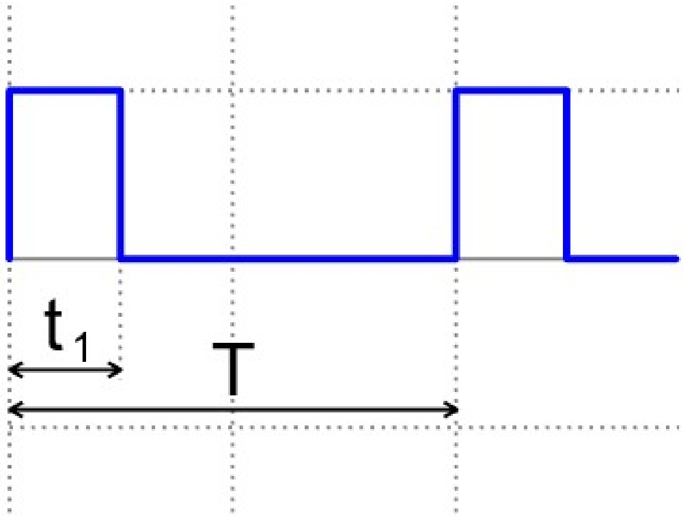

Figure 2.5: The relation between the time the signal is high and the signal period.

A PWM varies the voltage supplied to the system by varying the duty cycle. A small duty cycle means that the signal is high for a short portion of the signal period while a large duty cycle means that the signal is low for a large portion of the signal period. The system that is being supplied then uses the average voltage of this period. Therefore, a low duty cycle, a short high signal followed by a long low signal, would lead to a low average voltage. The variation in duty cycle and the associated average voltage is shown in figure 2.4. Ibrahim (2014)

To determine how long the signal must go high, the duty cycle is multiplied by the signal period. The duty cycle is often expressed as a percentage and so the duty cycle is the percentage of time that the signal is high. Therefore, by multiplying the duty cycle with the period gives the time for which the signal is pushed high (t1). Figure 2.5 shows the relationship between t1, the time the signal is high, and T, the signal period.

2.4 Analogue vs Digital Signals

Signals are used to convey data and information from point to point. For this project only electrical signals will be used although there are plenty of other mediums through which signals can be sent. There are two predominant signals that

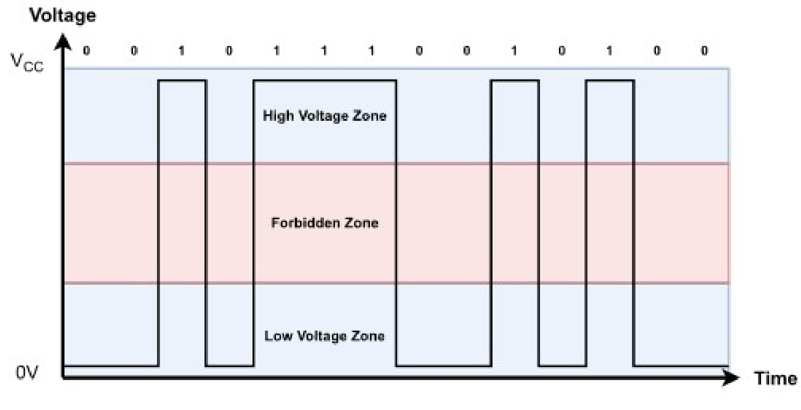

Figure 2.6: A digital signal and its three zones.

are used when regarding electrical signals, analogue and digital.

A digital signal, most simply represents discrete values, more precisely 2 discrete values. This makes digital signals perfect for conveying data in a binary data format but slightly more troublesome when more than two values are required. It will transmit a signal as either a low voltage, a zero, or a high voltage, a one. The low voltage is generally 0V while the high voltage is the voltage supply of the driving device. However, because voltages can have small fluctuations and will therefore not always be exactly 0V or equal to the nominal voltage, a range is preset whereby the receiving device can denote the value as low or high. A buffer zone is also incorporated, a voltage range around half the value of the nominal voltage, to prevent a small fluctuation in the voltage possibly altering the value of the signal. This buffer zone along with the area in which the signal can be read as high or low is shown in figure 2.6. This buffer is called the forbidden zone any signal received within the forbidden zone is considered floating and will be randomly assigned as either high or low.

An analogue signal on the other hand is continuous and where the digital signal ranged from 0 to an upper voltage, an analogue signal ranges from a low voltage to a high voltage. Typically, $\pm V_C C$, the voltage of the microcontroller, is used for these upper and lower limits. An example of a continuous analogue signal between $\pm V_C C$ is shown in figure 2.7. An analogue signal can therefore transmit an infinite number of values between these limits. By assigning an upper and lower limit to the sensor that will transmit the data, a max min transformation can be computed, and the transformed value transmitted along the analogue signal. Because the analogue

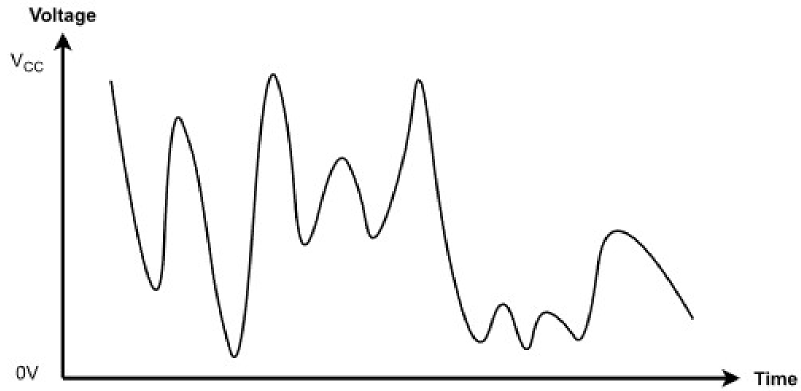

Figure 2.7: Analogue signal

signal is continuous, it can also be used in tracking the change in a value over time by computing the integral of the signal wave.

Chapter 3

Methodology and Design

This chapter will describe the methodology used in the project. The project is mostly software based and so that will be the focus. However, there is a physical element in that of the vessel and the electrical hardware. The mechanical design and fabrication of the thruster mounts is outside the scope of this project and so it will be described only in terms of how it interfaces with the electrical hardware.

3.1 Software Environment

The microcontroller that was chosen for this project is a Arduino DUE. An Arduino microcontroller was chosen as there is a wide range of Arduino libraries available online, as well as several forums and code examples. Furthermore, the Arduino environment uses the C language.

3.2 System Design

3.2.1 Objective

The system needs to be designed to be deployed for long periods of time between servicing. The future addition of power regeneration such as solar or wind can be used to improve the deployable range and period of the vessel but the energy storage must be designed to handle an extended period of 'dark time', any time when the power regeneration is negligible.

The vessel must be autonomous and having received a set of points before deployment, navigate between these points until retrieval. Even under non ideal circumstances the vessel should be able to correct its course and continue to navigate to the set points.

This system is a proof of concept that is designed to be able sized up to a larger vessel. Therefore the prototype vessel should be able to handle any conditions that could be encountered in testing and all electronics should be sufficiently sealed so that no damage is incurred. It is not expected that the prototype vessel can handle rough and storm weather conditions.

3.2.2 Engineering Requirements

The prototype is a proof of concept that can be scaled up to a larger vessel and so a small vessel that can accommodate at least 2 people is required. This is preferred to a smaller vessel that cannot accommodate the weight of a person as the weight to power ratio of a small vessel would not be as representative and this could influence the steering capability of the vessel and therefore the control system. Furthermore, a working vessel is going to require a large battery bank and this is easily represented in a larger vessel. The energy source can then be scaled up by adding cells in both parallel and series to create the required power supply for the working vessel.

The autonomous nature of the vessel means that the a electronic control system is required to control the vessel. Furthermore, there are several elements that are required to make up the system however, these are not integral to the autonomous nature of the system but integral to the system as a whole.

3.3 System Description

This section will give a broad overview of the system as a whole and describe the different sub-systems and their interactions. Several of these subsystems are outside the scope of the project but have been included to provide a foundation upon which the applicable detailed descriptions will build.

3.3.1 Hardware

Vessel

The vessel is outside the scope of this project as it was acquired before the start of the project. The project was designed around the use of the vessel for testing only as in actual uses the vessel would be upgraded to handle severe weather conditions and hold an array of sensory equipment. The vessel referred to in this project, pictured in figure 3.1 is a Spider 3, a small single hulled fibreglass boat. The vessel measures $1.3\,\mathrm{m} \times 3.2\,\mathrm{m}$ and is rated to carry 4 people and a 15hp traditional outboard motor.

Figure 3.1: The vessel, a spider 3.

Thrusters

Although the thrusters are electrical they are a complete unit together with the ESC, and all electronic interfacing was done with the ESC which then drove the thrusters. Therefore the thrusters are being considered general hardware and similarly to the boat were acquired prior to the project and are only there as a means of testing the control system. They are therefore also out of the scope of the project but are integral to the performance of the system overall.

The propulsion system consists of two electric thrusters mounted at the back of the vessel. These thrusters are each capable of producing up to 18 kg of thrust. An aliminium mount designed and manufactured by the Electrical engineering workshop allows for the thrusters to be raised during the launching and retrieval of the boat so as not to foul on the trailer. The mounts are removable and are removed for transport. Each thruster has a integrated ESC that regulates the power supplied to the electric motor and therefore the thrust provided. The ESC is detailed later in the report.

Power Supply

The initial design was to use a bank of 4 Lithium Iron Phosphate cells to form a 12 V battery. This was also procured before the start of the project. However upon testing, it was seen that 12 V was not enough voltage to provide the thrusters with enough power to move the boat at a reasonable rate. Each cell has a voltage of

3.3 V and a capacity of 100 A h. Therefore the next course of action was to source more cells and increase the size of the battery. However, due to the high cost of these cells and a lack of suppliers this was not plausible. LiFePO cells are expensive because they are designed to have a deeper life cycle than standard lead acid cells. These cells in particular were also expensive due to their large capacity.

Finally, it was decided that in order to test the control system, the power source would be changed to a battery of two lead acid cells. The lead acid cells were each 12 V and had a capacity of 50 A h. A lead acid battery can generally be drained to about 80 % capacity without doing much damage to the cell, a LiFePO cell has a deep cycle of about 40 %-50 %. Therefore these cells would be used for short tests and recharged between tests. However, for a working system it would be recommended that a large LiFePO battery bank were used.

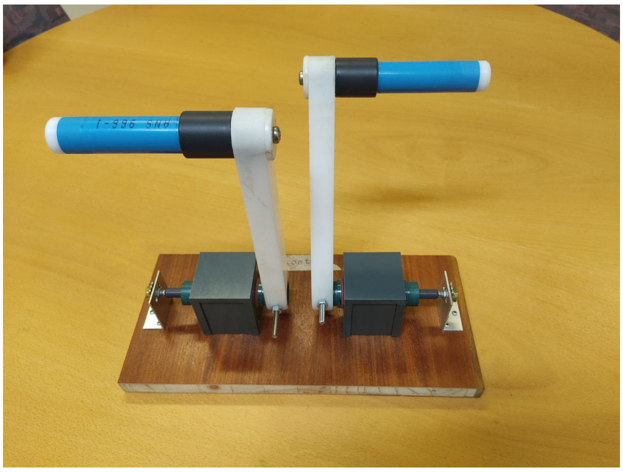

Figure 3.2: The final throttle system with two POTs on either end.

Throttle

The initial concept was to use two throttles that can move independently from each other and purchased to be used. However, when these components arrived, they were much smaller than they had appeared and they had very small range of movement. An alternate solution was designed and consisted of two throttle arms that could each turn a shaft. This shaft was then mounted to a linear potentiometers which would provide the required analogue input from the throttle. The throttle is shown in figure 3.2. An initial concept design was given to the electrical engineering department who then refined and manufactured the design.

PCB

3.3.2 Electronics

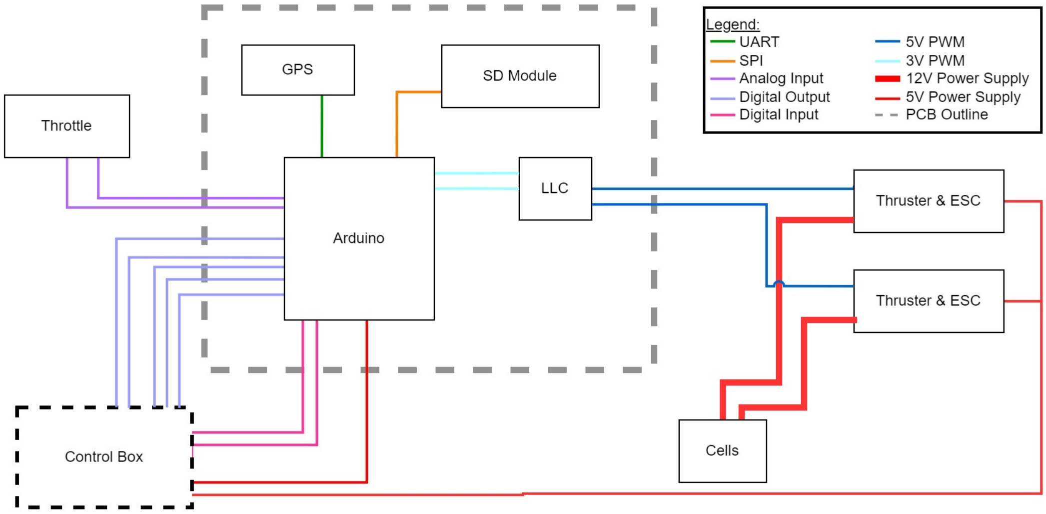

Figure 3.3: Wiring Diagram for the System

Arduino Due

The microcontroller was selected by considering the initial design and possible peripherals that would be used in the system. The minimum requirements for the microcontroller to be used with the peripherals of the GPS, SD card modules and the ESC and POTs as well as several inputs and outputs were as follows:

1. 2 PWM pins.
2. Voltage regulators of 3 V and 5 V.
3. 5 digital IO pins.
4. 4 analogue input pins.
5. 1 SPI connection.
6. 1 UART connection.
7. 256 kB programmable flash memory.

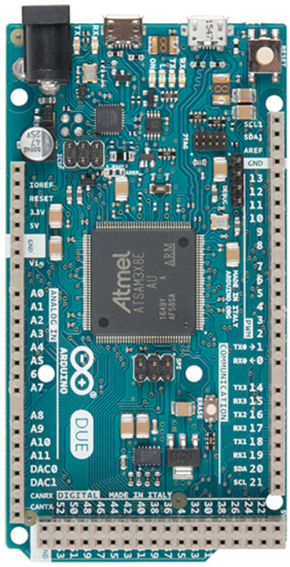

Figure 3.4: The microcontroller used, an Arduino DUE

Based on these requirements, the Arduino Uno was considered. It is the standard Arduino board used in projects and meets most of the requirements. However, the Arduino Uno has only one UART connection and although this does meet the minimum requirements it does not allow for any possible design alterations that might need a second UART connection. Finally, the Arduino only has 32 kB of programmable flash memory which does not meet the requirements.

The next consideration was the Arduino DUE. This has 4 UART connections and plenty of digital IO pins and several analogue inputs. This offers a range of versatility to any design progression or alterations that might occur. Furthermore, the DUE has 512 kB of programmable flash memory which is double that of the minimum requirement. The one flaw with the Arduino DUE is that its IO pins operate a 3.3 V as opposed to the generally standard 5 V. However, this was easily overcome by implementing a logic level converter to shift the required signals to 5 V while keeping the signals shape. Keeping the signals shape is particularly important for the PWM signal controlling the ESCs. The final microcontroller selected was the Arduino DUE.

SD Card Module

The SD card is used as an external storage device. Data can then be written to the SD card during operation and the data can be downloaded for analysis. The SD card module is a standard SD card module that is attached to the microcontroller as shown in the wiring diagram figure 3.3. The SD card uses SPI communication and there are built in libraries that are available for use.

GPS Module

The GPS module used is a PmodGPS and it Initially a PmodGPS was used as the GPS module. However the PmodGPS has a built in antenna and there were signal strength issues. The GPS was slow at acquiring a GPS fix when the control box was closed. Therefore an alternate GPS module was sourced that has an external antenna that can be fed out of the control box and placed in a position where a strong signal can be received. Both GPS modules use UART to communicate the data to the microcontroller. The GPS modules send a string of characters along the UART connection and the microcontroller must then decode the the characters. The GPS module is used to return the current longitude, latitude, date and time, speed and heading. The UART is set-up to use a baud rate of 9600, 8 data bits, no parity and 1 stop bit. has two connectors J1, which has 6 pins and J2 which has 2 pins. J1 is used to power the GPS module as well as connect to the MCU using UART communication with a baud rate of 9600, 8 data bits, no parity and 1 stop bit.

ESC

There are two ESC, one for each thruster and are submersible and therefore they are attached to the thrusters. The ESC has two inputs, the control input and the power input. Th power input can range between DC12 V and DC50 V and a maximum constant current of 100 A and in this project the input is 24 V supplied by the battery cells. The control input is a 5V signal that is used to control the speed of the thrusters. This is a PWM signal whose duty cycle determines the thrusters speed and direction.

POT

Table 3.1: Measured physical limits and neutral range of the throttle POTs

Throttle Position	**Left POT**	**Right POT**
Full Forward	655	720
Neutral	440 485	505 572
Full Reverse	285	352

The POT is a standard 10 kΩ potentiometer that has three pins as shown in figure 3.5, high voltage, ground and the output. The high voltage and ground are connected to the connections on either side and the output is connected to the middle. As the shaft of the potentiometer is turned the resistance varies from from almost no resistance to the full 10 kΩ which causes the voltage to vary from input to almost ground voltage.

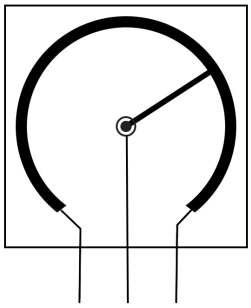

Figure 3.5: Fundamental illustration of a linear potentiometer

The output is connected to analogue input pins on the microcontroller which uses an ADC, with a 10 bit resolution, to convert the signal to a value between 0 and 1024. However, the potentiometer has a 300° range and the throttle only has about a 150° range of motion and so the software corrects this after having determined the extends of the motion through trial. The extends are measured at the two extremes and the middle point and offers a neutral range buffer for ease of use of the throttle. The measured extends are listed in table 3.1.

Switches and LEDs

The control box shown in figure 3.7 is made up by 3 switches and 5 state display LEDs. Two of the switches are configured using pull down resistor configuration with the output of the pull down circuit connected to a digital input on the microcontroller. Whereas, the LEDs all use a simple LED circuits that are driven by the digital outputs of the microcontroller. The final switch is the power switch and it is wired to cutoff the 5 V power supply to the microcontroller. The control box circuit is shown in figure 3.6 and can also been seen in the overall wiring diagram of the system in figure 3.3. A description of each switch and LED is given in table 3.2.

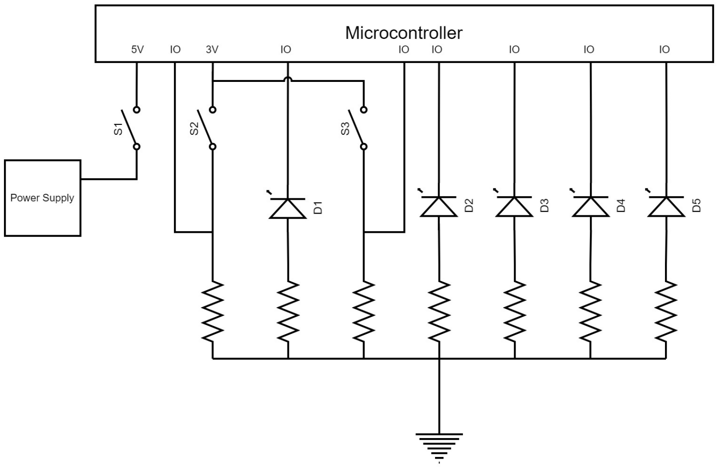

Figure 3.6: The wiring circuit inside of the control box.

3.3.3 Software

PWM Signal

The thrusters are each controlled by a 5 V PWM signal. There is limited information on the datasheet for the ESC and the signal boundaries, full forward, full reverse and neutral positions are described in the unconventional terms of the time

Figure 3.7: The control box with its 3 switches and 4 LEDs.

Table 3.2: Description of the switches and LEDs on the control box.

Component	Designation	Description
Power Switch	S1	This switch is used to cutoff the power being supplied to the microcontroller.
Navigation Mode Switch	S2	This switch allows the user to quickly switch between the manual navigation mode and the autonomous navigation mode.
Contingency Switch	S3	This switch is not used and was included and wired in case a design development required another switch.
On LED	D1	This LED indicates that the microcontroller is powered on.
Active LED	D2	This LED blinks on and off during the operation of the program and indicates that the program is alive and active and there has not been a crash in the program.
GPS LED	D3	This LED is turned on when GPS has a valid connection.
M LED	D4	This LED indicates that the system is in manual control mode.
A LED	D5	This LED indicates that the system is in autonomous navigation mode.

that the signal is high. The values are shown in table 3.3. Initially it was thought that the ESC operated at 50 Hz, however there was no response from the thruster at any duty cycle.

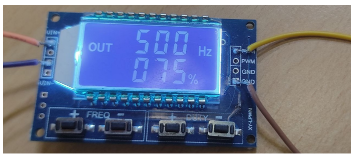

Figure 3.8: The PWM signal generator used to ensure the ESC was working.

In order to be sure that the correct PWM signal was being sent through and to be able to quickly vary both the frequency and the duty cycle, a PWM signal generator IC, shown in figure 3.8, was connected to ESC and trial and error was conducted to determine that the ESC began responding to a signal above 60 Hz.

Table 3.3: ESC boundaries and PWM duty cycle at various frequencies.

Position	Time (µs)	Duty Cycle @		
		50 Hz	60 Hz	500 Hz
Full Forward	2000	10 %	12 %	100 %
Neutral	1500	7.5 %	9 %	75 %
Full Reverse	1000	5 %	6 %	50 %

It was then decided to push the frequency up to the maximum of 500 Hz as this would offers the finest control because it has the maximum allowable duty cycle difference between the signal boundaries.

The Arduino libraries contain a function that can output a PWM wave, given the duty cycle and the pin to output on as a parameter however, the default frequency of the Arduino DUE PWM pins is 1000 Hz. Therefore, the frequency had to be manually changed by changing the timer settings driving the PWM signal.

The information needed to change the timer settings is available in the Arduino Datasheet. The process to configure the PWM outputs is as follows. First, the peripheral clocks for timer channels 6 and 7 were enabled. Secondly, the pins input output controller on peripheral A needed to be disabled and the pins switch to peripheral type B. Then came the configuration of the timer itself. The channel mode was set to waveform mode using clock 1 with the counter being incremented on the rising edge. Furthermore, the waveform was set to UP mode (signal being set high) being triggered when the counter reached the register C (RC) value and the signal being cleared when the counter reached the register A (RA) value. Once the timer is configured, the RA and RC values can be filled and the interrupt set to trigger when the counter reaches RC. Finally, the channel control register is set to perform a software trigger, reset the counter and start the clock. There is also a interrupt handler that needs to be added where in the status register is read. This is done because the flags in the status register are automatically reset when the status register is read, at the end of every period.

The signal is being cleared when the counter reaches RA and set high when the counter reaches RC and so RC can be equated to the period of the signal and RA to the time when the signal is high. Using this and that the clock is set to clock 1 which is the MCU clock (MCK) divided by 2, the RA and RC can be expressed in terms of MCK, frequency and duty cycle as shown in equations 3.1 and 3.2. RC is kept constant throughout while RA is changed to change the duty cycle of the

PWM signal sent to the ESC and therefore control the ESC and thruster.

$$RC = \frac{\frac{MCK}{2}}{Frequency} \tag{3.1}$$

$$RA = RC \times DutyCycle \tag{3.2}$$

ADC

As mentioned previously, the ADC is used to convert the analogue signal from the throttles potentiometers to a digital signal between 0 and 1024. Furthermore, using table 3.3 and 3.1 a relationship can be created to determine the RA value for a given throttle position. Because the forward operation is in the duty cycle range 75 % to 100 %, the analogue input from the POT needs to be linearly mapped to represent an equivalent duty cycle in this range. Equation 3.3 shows how this is done for the forwards operation. The same logic can be used to determine the duty cycle for reverse as equation 3.4 shows. For neutral however, the RA value needs to exact and this is difficult to achieve from the throttle, therefore, a neutral range is used on the potentiometer whereby any value in the range will be seen as neutral and the neutral value will be written to the RA register.

$$DutyCycle_{Forward} = 100 - (100 - 75)(\frac{POT_{MAX} - POT_{Input}}{POT_{MAX} - POT_{Neutral}}) \tag{3.3}$$

$$DutyCycle_{Reverse} = 75 - (75 - 50)(\frac{POT_{Neutral} - POT_{Input}}{POT_{Neutral} - POT_{MIN}}) \tag{3.4}$$

GPS

The GPS module transmits a series of 5 sentences containing various information. Each sentence begins with a '$GP' and then the specific message ID. The data is then sent through comma separated and finally ends with a checksum and the end of line characters <CR><LF>. For this project the 'RMC' sentence is the only sentence of interest as it contains all the necessary information, longitude, latitude and speed over ground in knots. An example of the 'RMC' sentence is shown below.

$GPRMC,064951.000,A,2307.1256,N,12016.4438,E,0.03,165.48,260406,3.05,W,A*55<CR><LF>

The entire sentence is read by the MCU before it starts to pick out the specific data out of the string using the commas and full stops as the guide to what array position is what data. The latitude and longitude are given in the format 'DDmm.mmmm' and must first be converted to decimal degrees as this is most easily used in calculations. The equation 3.5 shows this conversion. Each piece of data is then added to an instance of a GPS structure that can then be easily parsed to the navigation algorithm and analysed. The variables within the GPS structure are shown in table 3.4.

$$DecimalDegrees = Decimal + \frac{Minutes}{60} \tag{3.5}$$

Table 3.4: Variables and their types within the GPS structure.

Name	Type	Description
UTC	int	Coordinated Universal Time in the format (hhmmss).
latDecimal	int	The decimal portion of the latitude.
n_s	char	A character indicating North or South.
longiDecimal	int	The decimal portion of the longitude.
e_w	char	A character indication East or West.
knots	float	The speed over ground in knots.
course	float	The course over ground in degrees.
date	int	The date in the format (ddmmyy).

3.3.4 Control System

The control system is the scope of this project and the part of the project that should not require much alteration to scale the system up to a larger system. The control logic is all implemented through the software on the microcontroller. This section will detail the control logic with the aid of flow diagrams and pseudo code.

State Machine

The system makes use of a state machine to switch between tasks. There are four different states in which the system can be, waiting for valid GPS, manual navigation, autonomous navigation and halt. Figure 3.9 shows the relationship between these states. The algorithm shown is executed in the main while loop of the microcontroller and shows one cycle. The loop starts by checking the user input. The user can select to switch between the manual and autonomous operation at any point. However, if autonomous navigation is selected and the GPS data is valid, the system will only enter manual control once the cycle has been completed. The psuedo code for this loop is shown in algorithm 1.

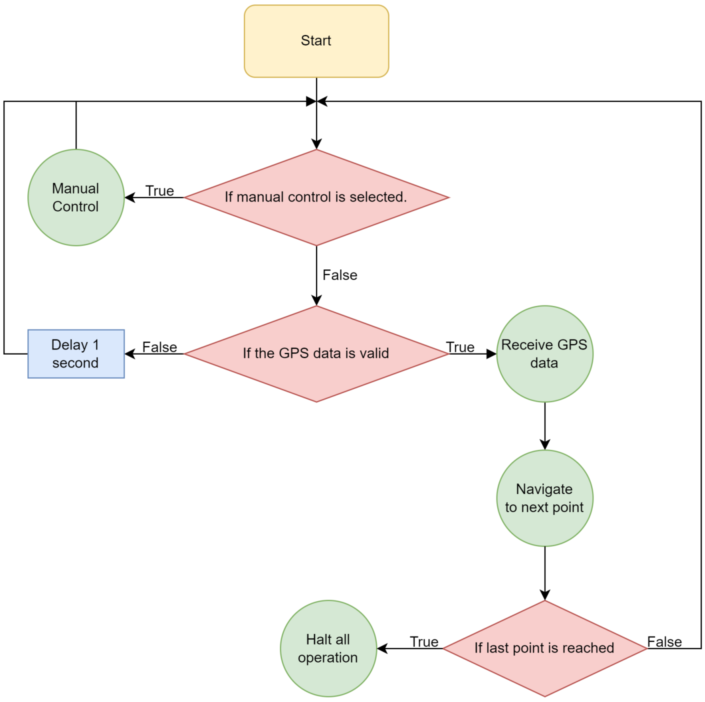

Figure 3.9: Flow diagram of the main states in the system.

Algorithm 1 State Algorithm

Require: Declare variables and initialize PWM, SPI, I^{2}C and UART

```
loop:
if manual control = true then
    call updatePWM()
    call powerThruster()
else
    call receivedGPSdata()
    if GPS data is valid then
        call navigate()
    else
        Delay 1 second
    if Last point has been reached then
        Halt receiving GPS data
goto loop.
```

Manual Control

The manual control involves receiving an input from the user and translating this to correctly power the thrusters. As seen in algorithm 1, the manual control involves the call of two functions, the first where most of the processing is done and the second is updating the RA register to produce the correct PWM signal. Firstly, the flow diagram of the first one, *updatePWM()* is shown in figure 3.10. This shows the process to read one sides input and calculate the required duty cycle for that sides thruster. This is carried out for both sides in each cycle. As mentioned, the second function *powerThruster()* simply writes the required value to RA register for the appropriate duty cycle. The psuedo for both functions are shown in the algorthims 2.

Algorithm 2 Pseudo code for manual control

```
procedure UPDATEPWM
    analog ← analogRead(analogPin)
    if analog > maximum threshold then
        duty cycle ← 95;
    else if analog > neutral threshold then
        analog ratio ← ratio forward (equ. 3.3) asfloat
        difference ← (95 − 75) × analog ratio asfloat
        duty cycle ← 95 − difference
    else if analog < minimum threshold then
        duty cycle ← 50;
    else if analog < neutral threshold then
        analog ratio ← ratio reverse (equ. 3.4)as float
        difference ← (75 − 50) × analog ratio asfloat
        duty cycle ← 75 − difference
    else
        duty cycle ← 75;
procedure POWERTHRUSTERS
    RA ← RC × duty cycle%;
```

Receive GPS Data

The GPS data is received with the serial UART communication and is a string of characters as shown previously. In order to receive the correct data line, the microcontroller receives each line and then checks that it is the required line. If so, the line is processed and the struct is updated with the newest data. A few

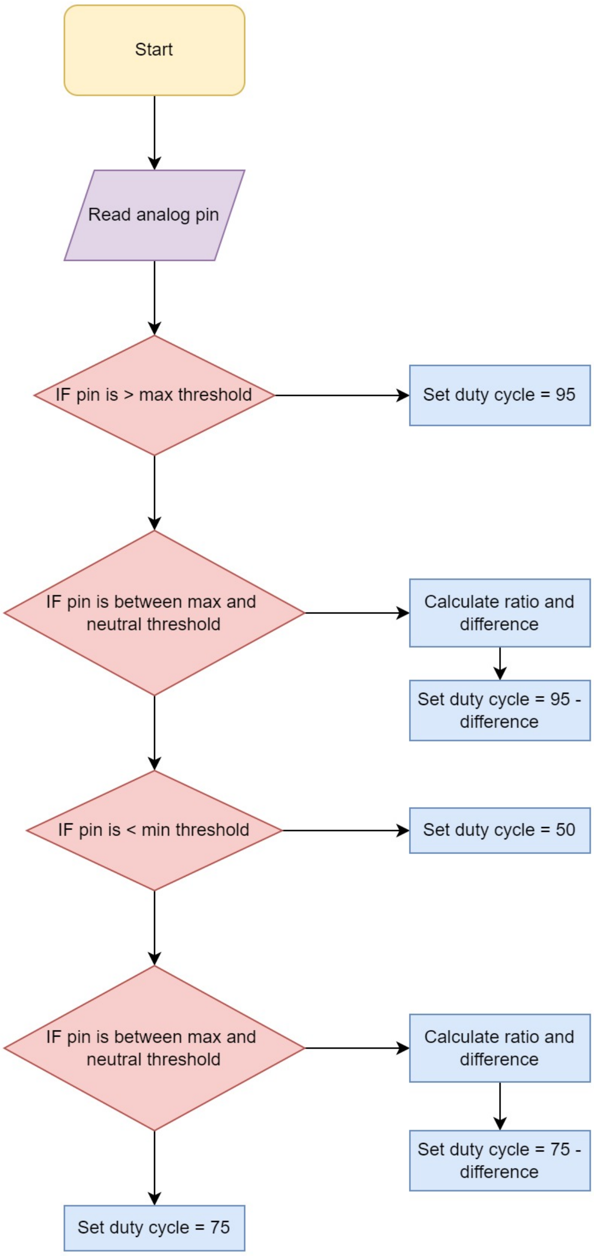

Figure 3.10: Flow diagram of *updatePWM()*

small tools were written to assist in this. They are all relatively simple and so their pseudo is unnecessary but a brief description follows for context.

findAllInstacesOf()

This function takes an integer array, a character array to search, a character to search for and the size of the character array to search as parameters. It returns the integer array with the index positions of the character searched for.

charToInteger()

This function takes a character array and a start and end index as integers. It returns an integer as it would be seen as the character string. For example, the character string '4862' would be returned as 4862 as an integer.

charToDecimals()

This function is similar to the integer version but it returns a float with the numbers as the decimals. For example a character string of '4862' would be returned as 0.4862.

The overall process to received the GPS data and populate the struct is shown in the flow diagram of figure 3.11. The populating of the struct is just using the arrays of full stop and comma indexes to split up the character array into the necessary information. This is easily done as the string from the GPS has a standard form.

Algorithm 3 Pseudo code for receiving GPS data

```
msgReceived ← false
rxCnt ← 0
while !msgRecieved do
    while rxByte! = 10 do
        rxByte ← SerialRead()
        if rxByte! = −1 then
            rxData(rxCnt) ← rxByte
            rxCnt ← rxCnt + 1
    if rxData(3) = 'R' AND rxData(4) = 'M' AND rxData(5) = 'C' then
        call updateStructRMC(rxData, rxCnt)
        msgReceived ← true
    rxCnt ← 0
    rxByte ← 0
```

Autonomous Navigation

The autonomous navigation will only begin executing when valid GPS data has been received. Therefore there is no need to check the data when the navigation algorithm begins. The target points for the navigation are written into a text file

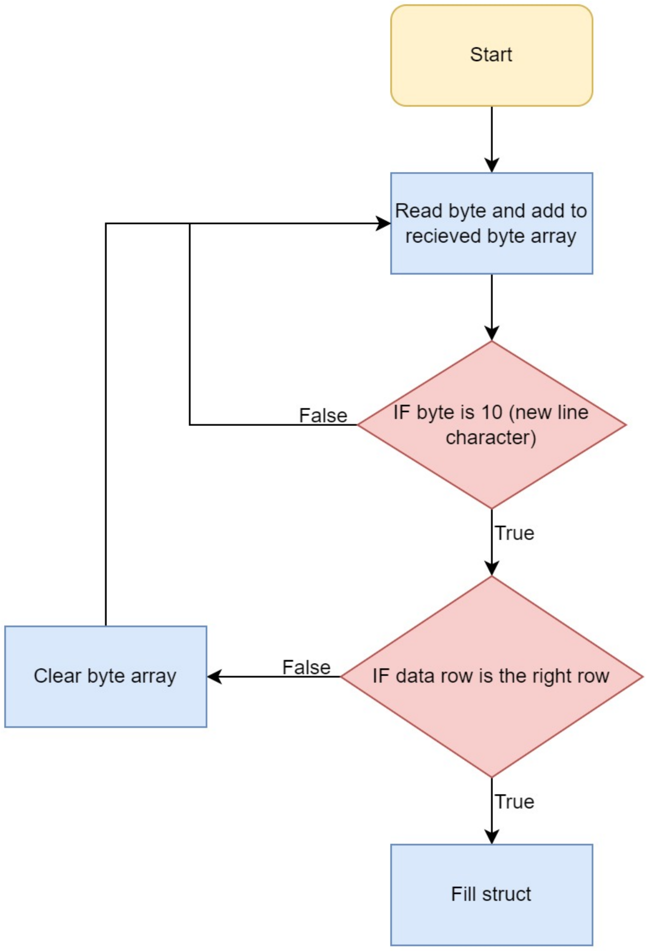

Figure 3.11: The flow diagram of receiving the GPS data.

prior to launching. The program reads these points into a struct at the start of operation to be ready for the autonomous navigation. The autonomous control calculates the distance and bearing to its destination and the error between its current bearing and its target bearing. If the distance and bearing error are above a set threshold, the system will move at full power using full steering. However, once it is within the distance threshold, it will begin to slow down proportional to how close it is. The amount of throttle as a percentage is determined by a ratio of the remaining distance and the distance threshold. Furthermore, once the distance is reduced below the arrival threshold, the point is marked as 'passed' and the system will begin navigating to the next point or half operation if there are no more points. Similarly for the steering, the amount of steering as a percentage is a ratio of the bearing error and the steering threshold. Finally the direction to steer is determined using an equation that accounts for the 360°/0° turn over point. If the target bearing is anywhere within 180° to the right of the current bearing, it will turn right and visa versa. This logic and pseudo code can be seen in the flow diagram of figure 3.12 and algorithm 4. In the pseudo code only the procedure for *steerLeft()* is shown as the procedure for *steerRight()* follows the exact same logic but the thruster sides are inverted.

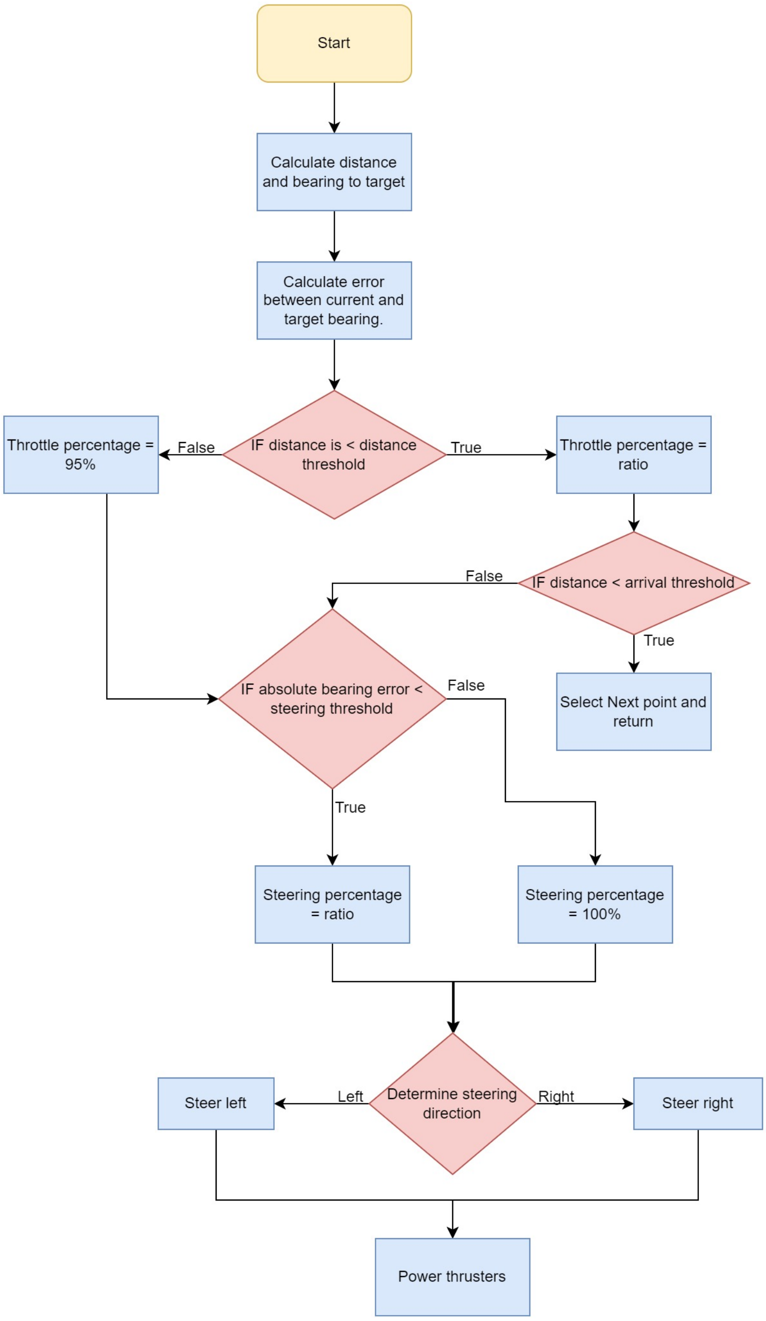

Figure 3.12: Flow diagram of the autonomous navigation

Algorithm 4 Pseudo code for the autonomous navigation

Require: Read the target GPS points from SD card.

procedure NAVIGATE()
    Calculate distance to target
    Calculate bearing to target
    Calculate the error between current and target bearing
    if $distance < distance\ threshold$ **then**
        $throttle\ percentage \leftarrow \frac{distance}{2 \times distance\ threshold}$
        if $distance < arrival\ threshold$ **then**
            call NextPoint()
            return
    else
        $throttle\ percentage \leftarrow 95$
    if $abs(bearing\ error) < steering\ threshold$ **then**
        $steering\ percentage \leftarrow \frac{abs(bearing\ error)}{steering\ threshold}$
    else
        $steering\ percentage \leftarrow 100$
    $maxBearing \leftarrow max(bearing\ error,\ target\ bearing)$
    $minBearing \leftarrow min(bearing\ error,\ target\ bearing)$
    if $(maxBearing = target\ bearing) XOR (maxBearing - minBearing) > 180)$ **then**
        call steerLeft()
    else
        call steerRight()
    call powerThrusters()

procedure NEXTPOINT()
    $points(targetIndex).passed \leftarrow true$
    $targetIndex \leftarrow targetIndex + 1$
    if *points.targetIndex does not exists* **then**
        $state \leftarrow Halt\ navigation$

procedure STEERLEFT()
    $throttle\ difference = 2 \times throttle\ percentage$
    $rightSide.duty\ Cycle \leftarrow 75 + (25 \times (throttle\ percentage\%))$
    $leftSide.duty\ Cycle \leftarrow rightSide.duty\ Cycle - (25 \times ((throttle\ difference \times (steering\ percentage\%))/100))$

Chapter 4

Testing and Results

List of References

(2011). History of the GPS Program.
Available at: https://www.aiaa.org/docs/default-source/uploadedfiles/about-aiaa/press-room/videos/iaf-60th-anniv-gps-nomination.pdf?sfvrsn=9bc64bfa_0

(Advanced Safety Devices) (2013). How Does a Digital Compass Work.
Available at: https://www.safety-devices.com/how-does-digital-compass-work-a-12.html

Aerospace (2021). A brief History of GPS.
Available at: https://aerospace.org/article/brief-history-gps

Christ, R.D. and Wernli, R.L. (2014). Power and Telemetry. In: *The ROV Manual*, pp. 141–161. Elsevier.

(Federal Aviation Administration) (). Satellite Navigation - GPS - How It Works.
Available at: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks

Ibrahim, D. (2014). Advanced PIC32 Projects. In: *Designing Embedded Systems with 32-Bit PIC Microcontrollers and MikroC*, pp. 359–442. Elsevier.

Jones, E. (2019). True North and magnetic north: whats the Difference.
Available at: https://www.rmg.co.uk/stories/topics/true-north-magnetic-north-whats-difference

Mai, T. (2017). Global Positioning History.
Available at: https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html

(USAF) (1978). GPS Early User Equipment 2.
Available at: https://www.losangeles.spaceforce.mil/News/Photos/igphoto/2000044571/