



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Design of an Unmanned Sailing Vessel

by

Philip van Schalkwyk
20838816

Mechatronic Project 488

Final Report

Study leader: Prof D.J.J. Versfeld

July 2021

Plagiarism Declaration

1. Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.
2. I agree that plagiarism is a punishable offence because it constitutes theft.
3. I also understand that direct translations are plagiarism.
4. Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.
5. I declare that the work contained in this assignment, except where otherwise stated, is my original 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.

Student number	Signature
Initials and surname	Date

EXECUTIVE SUMMARY

Student: P. van Schalkwyk

Title of project
Design of an Unmanned Sailing Vessel.
Objectives
Design and develop a sailing vessel that has onboard micro-processor(s) to read sensor data, control actuators and dynamically make navigation decisions without the need for a crew.
What is current practice and what are its limitations?
Advanced sailing vessels are capable of semi-autonomous operation, by keeping a constant heading or maintaining a point of sail. Marine research vessels are not capable of running for extended periods due to harsh weather conditions, remote locations, the need for supplies and costly human labor.
What is new in this project?
The project retrofit an existing remote control sailing vessel with a control system to allow for sensor reading, autonomous sailing, and waypoint following.
Was the project successful? How will it make a difference?
The project was successful. It proved the concept of autonomous sailing by demonstrating the ability to adjust sails, control the rudder and keep a required heading. This will allow research to be conducted in remote locations without the need for human crew.
What were the risks to the project being a success? How were these handled?
The digital compass might provide inaccurate data to tilting and pitching of the boat or magnetic interference from onboard motors. The control logic might not work in practice due effects that were not accounted for. The compass headings will be tilt-compensated, and the control system can dynamically calculate new trajectories to correct for unaccounted effects.
What contributions have/will other students made/make?
The degree of autonomy can be improved by explicitly accounting for currents, fitting an obstacle avoidance system, and adding PV-panels to power the vessel.
Which aspects of the project will carry on after completion and why?
The project will be scaled and fitted to a sailing vessel to conduct research autonomously.
What arrangements have been/will be made to expedite continuation?
The entire project will be documented, detailed, and photographed to the finest extent to ensure that it is repeatable for future endeavors to improve the degree of autonomy. The entire project will also be made available to the National Research Foundation for further development.

Student

Date

Lecturer

Contents

Plagiarism Declaration	i
Executive Summary	ii
Contents	iii
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.2 Objectives	2
1.3 Motivation	2
2 Literature Review	3
2.1 Sail Theory	3
2.1.1 Sailing Vessel Terminology	3
2.2 Basic Sailing Manoeuvres	4
2.2.1 Tack	4
2.2.2 Gybe	4
2.2.3 Get Out of Irons	5
2.3 Unmanned Surface Vessels: A History	5
2.4 Initial Developments	5
2.5 Saildrone Commercial Development	5
2.6 Autonomous Sailing Vessel Design by Stenersen	6
2.6.1 System Sensors	7
2.6.2 Microcontroller Usage	7
2.6.3 Navigational Control	7
2.6.4 Directional Control	8
2.6.5 Modes of Operation	8
2.6.6 Triangle Path Test	9

2.6.7	Results Obtained	9
2.7	Autonomous Sailing Vessel Design by Ferl and Hills	10
2.7.1	System Sensors	10
2.7.2	Microcontroller Usage	10
2.7.3	Navigational Control	11
2.7.4	Directional Control	11
2.7.5	Sail Control	12
2.7.6	Modes of Operation	12
3	Design Requirements	13
4	Digital Compass Concept Development	14
4.1	Background	14
4.1.1	Axis of Reference	14
4.1.2	Traditional Magnetic Compass Mechanism	15
4.1.3	Digital Compass	15
4.2	Hardware	16
4.3	Software	17
4.4	Sensor Calibration Procedure	18
4.4.1	Uncalibrated Magnetometer Data	18
4.4.2	Calibration Factor Calculation	19
4.4.3	Calibrated Magnetometer	21
4.5	Compensation Algorithms	21
4.5.1	Calculation of Pitch Angle	21
4.5.2	Calculation of Roll Angle	22
4.5.3	Compensation for Linear Acceleration	23
4.5.4	Gyroscope-based Pitch and Roll	24
4.5.5	Complimentary Usage of Accelerometer and Gyroscope Data	25
4.5.6	Construction of the Tilt-Compensation Matrix	26
4.5.7	Magnetic Heading	27
4.5.8	True Heading	27
5	Additional Sensor Implementations	28
5.1	GPS Unit	28
5.1.1	Hardware	28
5.1.2	Software	29
5.1.3	Explanation of TinyGPS++	29
5.1.4	Explanation of GPS reading function	30
5.1.5	Required heading determination	30
5.2	Data Logger System	31
5.2.1	Hardware	31

5.2.2 Software	31
6 Physical Hardware Development	33
6.1 Windvane Development	33
6.1.1 Sense of Direction	33
6.2 PCB Design	34
6.3 Housing Design	35
7 Control System Design	36
7.1 Navigation	36
7.1.1 GPS Navigation	36
7.1.2 Compass Navigation	37
7.1.3 Waypoint Navigation	37
7.2 Sailing Limitations	37
7.3 Directional Control	38
7.3.1 Control Parameters	39
7.4 Propulsion Control	40
7.5 Servo Motor Control	40
7.6 Control System Summary	41
7.7 Limitations of the Control System	42
8 Testing	44
8.1 Digital Compass	44
8.1.1 Experimental Setup	45
8.1.2 Results	45
8.1.3 Discussion of Results	46
8.2 GPS Unit	47
8.3 Windvane	48
8.4 Control System	52
8.5 Integration of Entire System	53
8.5.1 Lake Test 1	54
8.5.2 Lake Test 2	55
8.5.3 Lake Test 3	56
9 Evaluation	58
9.1 Evaluation of Sub-System	58
9.1.1 Digital Compass	58
9.1.2 GPS Unit	58
9.1.3 Datalogger System	59
9.1.4 Windvane	59
9.2 Evaluation of Control System	59

9.2.1	Theoretical Working	59
9.2.2	Performance of Vessel in Water	59
9.3	Recommended Improvements and Future Work	60
10	Conclusion	61
	Bibliography	62

List of Figures

2.2.1	Points of sailing (Safe-skipper, 2018)	4
2.5.1	Saildrone vessel (Saildrone, 2020)	6
2.6.1	Prototype of autonomous sailing vessel (Stenersen, 2016)	7
2.6.2	Triangle path test (Stenersen,2016)	9
4.1.1	Directional axis of geography (Langley, 2003)	14
4.1.2	Magnetic Declination as a function of location (INSERT REF)	15
4.1.3	Explanation of heading	16
4.2.1	Digital compass setup	17
4.4.1	Magnetometer readings: X vs Y vs Z.	19
4.4.2	Magnetometer readings: X vs Y vs Z showing real compared to ideal	19
4.5.1	Tilt compensated axis system	21
4.5.2	Accelerometer vs gyro vs filtered pitch during vibration	24
4.5.3	Accelerometer vs gyro vs filtered pitch displaying gyro drift	24
4.5.4	Accelerometer vs gyro vs filtered pitch when inducing sudden pitch .	26
6.1.1	Windvane assembly	34
6.2.1	Schematic of PCB	35
6.3.1	Assembly of electronics housing	35
7.1.1	Navigation function block diagram	36
7.1.2	Compass function block diagram	37
7.2.1	Tacking navigation function block diagram	38
7.3.1	Turn function block diagram	39
7.3.2	Tack function block diagram	39
7.4.1	Sail control function block diagram	40
7.6.1	Standard operating control block diagram	42
7.6.2	Void loop block diagram	42
8.1.1	Digital compass test setup	44
8.1.2	Digital compass test 1	45
8.1.3	Digital compass test 2	46

8.1.4	Digital compass test 3	46
8.2.1	GPS accuracy test	48
8.3.1	Ideal I2C communication	49
8.3.2	Real I2C communication without mitigation attempts	49
8.3.3	Real I2C communication after mitigation attempts	50
8.3.4	Resulting I2C communication test	50
8.3.5	Shielded twisted pairs INSERT REF	51
8.4.1	Results of park test 1	52
8.4.2	Results of park test 2	53
8.5.1	Dragonflite95 during lake tests	54
8.5.2	Results of lake test 1	54
8.5.3	Results of lake test 2	55
8.5.4	Results of lake test 3	57

List of Tables

3.0.1	Technical Performance Measures for the design of the unmanned sailing vessel.	13
4.2.1	Hardware wiring for digital compass	17
4.4.1	Uncalibrated magnetometer maximum and minimum values	20
4.4.2	Results of calibration calculation	20
4.5.1	Magnetic heading calculation	27
5.1.1	Hardware setup of GPS	28
5.1.2	NMEA sentence identifier meanings	29
5.2.1	Hardware setup of datalogger system	31
7.5.1	Servo control PWM value range	41
8.1.1	Compass Test Results	45

Chapter 1

Introduction

1.1 Background

It is said that 70% of the earth's surface area is covered by oceans of which 80% is still unmapped, unobserved and unexplored (NOAA, 2021).

When considering oceanic research, extreme missions such as Antarctic Missions conducted by the South African National Antarctic Program (SANAP) and the Council for Scientific and Industrial Research (CSIR) aboard the SA Agulhas II, typically entails a 3-month voyage covering 15000km. These voyages are critically limited by factors such as fuel capacity and availability due to remote locations, extreme weather conditions and human psychological needs. Out of these limitations, stems the idea for the design and development of an unmanned sailing vessel. The project will focus on the design of a digital compass, GPS navigation system, development of a control-algorithm and application to an existing model sailing vessel as a proof of concept.

A sailing vessel can conduct self-sustained year-round oceanic research in remote locations and extreme conditions without the limitation of human capability. An unmanned sailing vessel will thus have the potential to collect oceanic-, marine- and geological data without the time-consuming and costly human input, excessive fuel costs and induced environmental pollution that is currently the norm.

The research is the concern of the Nasional Research Foundation (NRF) developed through the Department of Electrical and Electronic Engineering at Stellenbosch University by Mr. P van Schalkwyk under the guidance of Dr. J Versfeld.

1.2 Objectives

As stated, the project entails the design and development of an unmanned sailing vessel that is to be able to semi-autonomously operate by wind propulsion only. A DragonFlite95 remote control sailing vessel is bought out and retrofitted with designed sensors and a control system to achieve autonomous operation. Due to the time-constraint of this project, the objectives will be implemented modularly. The objectives are:

1. Design, and build a low-cost digital compass by using a 3-axis magnetometer that compensates accurately for pitch and tilt.
2. Design and build a suitable low-cost windvane which houses a digital compass. Retrofit the windvane to the DragonFlite95 sailing vessel.
3. Retrofit the digital compass, windvane, a GPS unit, and data logger system to the DragonFlite95 sailing vessel for navigation control and data collection.
4. Design a theoretical control-algorithm for autonomous operation of the sailing vessel to reach a destination.
5. Apply the control-algorithm through a microcontroller.
6. Test and verify the working of the control system by data analysis.

1.3 Motivation

The norm for environmental conservation research in remote oceanic locations entails the use of tremendous diesel and electric motors which contradictorily induces large scale pollution. If an unmanned sailing vessel is successfully developed, it could be a step towards the future of oceanic and marine research due to its renewable nature and low cost of operation.

The successful development of a scale model could lead to upscaling of the project to a real sailing vessel to be used for the benefit of truly environmentally friendly research. Future uses could include combatting illegal offshore trawling and protecting the South African coastline in remote locations.

This project combines knowledge obtained from the Department of Mechanical and Mechatronic Engineering such as design of a digital compass and its tilt-compensation algorithm, implementation of control hardware, design and programming of electronic systems, and validation of data acquired during testing.

Chapter 2

Literature Review

2.1 Sail Theory

The art of sailing is believed to have started in ancient Egypt around 3400BC for transporting cargo and travelling long distances. The sailboat, as known today, still relies on two very basic principles; wind-based propulsion and rudder-based directional control. Sails are trimmed to best capture the force of the wind over an area to induce motion through water. A rudder creates a motion unbalance at the rear of the vessel to induce a change in direction.

2.1.1 Sailing Vessel Terminology

- Hull: The body of the boat below the deck
- Deck: The upper surface of the hull
- Bow: The front of the boat
- Stern: The rear of the boat
- Port: The left side of the boat when facing the bow from the stern
- Starboard: The right side of the boat when facing the bow from the stern
- Rudder: The vertical blade at the stern of the vessel that provided direction to the boat
- Main sail: The largest sail on the vessel (at the stern of the vessel)
- Jib or Genoa sail: The smaller sail on the vessel (at the bow of the vessel)
- Wind vane: A device used to detect the angle of the apparent wind

2.2 Basic Sailing Manoeuvres

Figure 2.2.1 illustrates different points of sailing with specified angles of attack (angle to the wind). The no-sail zone illustrated in purple, also referred to as sailing in irons, is any angle within 45° from the direction of the wind where the sails do not generate sufficient force for accelerating the vessel depending on vessel characteristics (Good Hope Sailing Academy, 2020).

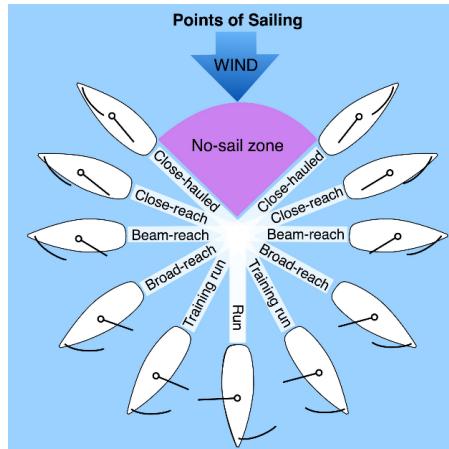


Figure 2.2.1: Points of sailing (Safe-skipper, 2018)

2.2.1 Tack

When sailing against the direction of the wind, it is not possible to sail in a straight line parallel to the wind direction. The act of iteratively sailing back and forth across the angle of the wind is referred to as tacking. This manoeuvre is performed by turning the bow through the wind so the direction of apparent wind on the sail is changed from one side to the other. This iterative process allows progress towards the desired waypoint (Good Hope Sailing Academy, 2020).

2.2.2 Gybe

When sailing downwind, a gybe is the act of turning the stern through the direction of the wind so the angle of wind on the sails is changed from one side to the other. Gybing can cause unintended course changes due to the extreme moments caused by the change in force direction (Good Hope Sailing Academy, 2020).

2.2.3 Get Out of Irons

When the vessel is pointing directly into the wind and there occurs a stall in momentum, a manoeuvre will have to be performed to get the boat moving again. This routine is referred to as “get out of iron”. The process entails moving the rudder away from the centre position quickly and moving it back to the centre position slowly to “push” the stern of the boat and change the heading. The second option is to push the mainsail out in the direction of which a change in heading is required (Good Hope Sailing Academy, 2020).

2.3 Unmanned Surface Vessels: A History

A review paper by Battelle Applied Coastal and Environmental Services notes that although unknown to society until recently, Unmanned Surface Vessels (USVs) have been a topic of research since 1993, and advanced into production for prototyping soon after. (Manley, 2008). The earliest prototype was a fishing trawler USV named ARTEMIS capable of demonstrating functioning navigational control systems. The project was led by the Massachusetts Institute of Technology and was developed for localized bathymetric data mapping in the Charles River.

2.4 Initial Developments

Following the successful development and implementation of ARTEMIS, several other research developments followed. More recently, better documented autonomous sailing vessel projects such as ASAROME, VAIMOS, and Saildrone have been conducted and the latter being currently commercially developed. The research of Stenersen, and Ferl et. al will be studied and reviewed in depth.

2.5 Saildrone Commercial Development

The Saildrone start-up company was founded in 2012 in Alameda, California near the San Francisco Bay Area (Saildrone, 2020). The company develops autonomous sailing vessels that are fitted with 42 meteorological and oceanographic sensors that allow the vessels to reach extreme locations and collect data regarding ice melt, oceanic carbon dioxide levels, and count marine population while removing the need for human crew. Saildrone offers institutions the opportunity to conduct

oceanic research at a fixed cost of \$2500 per day instead of using government vessels or satellites with huge capital and running costs (Yacowicz, 2017).

Saildrone aims to aid in the scaling of marine research by deploying fleets of vessels to form a network of dynamically allocated data sources. The vessels are also said be durable since the company has not sunken a vessel within the last four years of operation (Yacowicz, 2017).

As seen in figure 2.5.1, Saildrone vessels utilize a fixed wing design that is controlled by a wing flap. Due to the commercial applications of the Saildrone vessel, little information regarding its guidance- and control systems are known.



Figure 2.5.1: Saildrone vessel (Saildrone, 2020)

2.6 Autonomous Sailing Vessel Design by Stenersen

The following developments differ from previously discussed vessels in the sense that they are solely powered by wind propulsion, which is the part of the requirements of this project. This section will discuss the work of Stenersen and Ferl et. al in depth, highlighting the methodology, results, and key discussions.

The work of Stenersen aimed to develop a small scale-model of a sailing vessel that houses the needed components that can aid autonomous operation. Tasks included the design of a microcontroller-based control system that could log data, dynamically make high level navigational decisions, and implement low-level control of mechanical actuators. A simple control algorithm was suggested and verified during field tests.



Figure 2.6.1: Prototype of autonomous sailing vessel (Stenersen, 2016)

2.6.1 System Sensors

The system implemented includes a wind direction sensor, inertial measurement unit (IMU), GPS, a rudder servomotor, and multiple sail servomotors. The wind direction sensor is a wind vane that rotates on a potentiometer to obtain a relative wind direction due to a change in resistance (Stenersen, 2016).

2.6.2 Microcontroller Usage

The high-level control algorithm is executed on a Raspberry-Pi microcomputer that is running the Raspbian operating system. A controller area network (CAN bus) was setup to house all sensory devices on a single platform powered by a modified Atmega2560 microchip. The performance of the Atmega2560 microchip is limited by its sequential operating capability and 16 MHz clock speed, and therefore was programmed to mimic running concurrent operations by setting up threads and corresponding priorities used for scheduling tasks through use of FreeRTOS, a real time operating system that can be integrated into the sequential processor operation (Stenersen, 2016).

The Raspberry-Pi reads information via the CAN bus and executes the control logic, and then passes the required servo actuation requirements back to the Atmega via the CAN bus (Stenersen, 2016)

2.6.3 Navigational Control

The IMU consists of a 3-axis accelerometer and 3-axis magnetometer which is used to determine the rotational orientation of the vessel from the view of a North-East-Down (NED) coordinate frame. The accelerometer provides a roll and pitch angle since it can measure the gravitational acceleration vector (Stenersen, 2016).

After calibration, the magnetometer readings are used to calculate heading. Stenersen notes that a significant noise component is present and that a low pass filter was considered to filter out high frequency noise in the range of 1 to 3 degrees. The result however showed that a significant phase lag was introduced and thus an adaptive filter was rather used (Stenersen, 2016).

2.6.4 Directional Control

Due to the use of a model sailing vessel that consists of two independently controlled sails, the work of Stenersen focuses on directional control by inducing a generated lift caused by different sail angles in the mainsail and the Genoa. In this setup, the effect of the rudder is quite insignificant and is generally only used for fine tuning heading (Stenersen, 2016).

2.6.5 Modes of Operation

Semi-manual

In semi-manual mode, the user, through a graphical user interface (GUI) dynamically selects the desired heading and the control system keeps the boat in a state of movement by dynamically adjusting the sails, if the wind allows for sailing in that direction (Stenersen, 2016).

Heading Hold

In heading hold mode, the control system utilizes proportional control to hold the boat to a desired heading by changing the difference in sail angles between the mainsail and the Genoa, effectively generating a lift (Stenersen, 2016).

Waypoint Tracking

In waypoint tracking mode, the control system utilizes heading hold mode combined with a required bearing calculated from the current GPS position and the next required waypoint coordinate set. A waypoint is reached when coming within a certain radius of that point (Stenersen, 2016).

Path Following

In path following mode, the control system utilizes a combination of all the above-mentioned modes and tries to follow a straight line between its initial position and the waypoint. This mode of operation is based on Lookahead-based steering (Stenersen, 2016).

2.6.6 Triangle Path Test

A capability field test is used by Stenersen referred to as the triangle path test. This method consists of constructing a path that resembles an equilateral triangle. The test displayed the ability of the vessel to sail a 100-meter course along certain routes and achieved a maximum velocity of 0.61 m/s. The one side of the triangle must be parallel to the wind direction to ensure that a vessel can feasibly sail along the desired course (Stenersen, 2016).

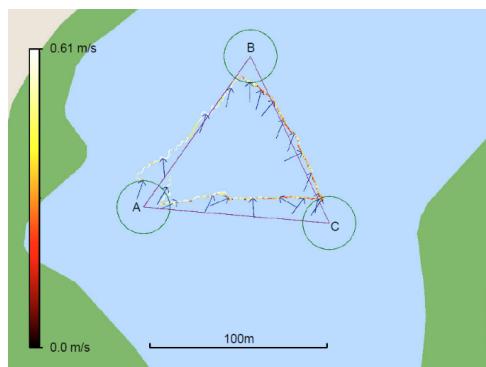


Figure 2.6.2: Triangle path test (Stenersen,2016)

2.6.7 Results Obtained

Stenersen found that the vessel was able to adequately perform in all the above-mentioned modes of operation and the triangle path test. The control system performed predictably, and all decisions were verified post-test by use of a data logger.

2.7 Autonomous Sailing Vessel Design by Ferl and Hills

The work of Ferl et al. aimed to develop a 1.2m autonomous sailing vessel that would be capable of completing a trans-Atlantic attempt. It required the development and fitting of sensors for location and weather observations, solar power systems for charging a battery and powering the system, and the design of a control system for autonomous navigation and a fixed wing to reduce energy consumption (Ferl and Hills, 2020).

Preliminary testing exhibited positive results but yielded sluggish rudder response times due to arduous calculations for the navigational sub-routine. The solution implemented was to use FreeRTOS based threading to mimic parallel execution on a sequential processor.

2.7.1 System Sensors

The system incorporated a Davis 6410 Vantage Pro2 Anemometer to sense apparent wind direction and speed, a Waveshare NEO-7M GPS chip, and an Adafruit LSM303 Magnetometer and Accelerometer. The Davis 6410 Anemometer provides an accurate wind direction and speed, but has a high cost of R2400-00 and is said to be non-rugged (Ferl and Hills, 2020).

The Waveshare NEO-7M GPS chip is used to determine the current location for navigational calculations. The Adafruit LSM303 IMU is used to determine the heading of the vessel by using a magnetometer and accelerometer to determine a tilt-compensated heading (Ferl and Hills, 2020).

2.7.2 Microcontroller Usage

The system uses an Arduino Uno for executing its control algorithm. The microcontroller communicates with the IMU via I2C, with the GPS chip via SPI, and utilizes its analog-to-digital converter to read wind speed and wind direction. The rudder and sail servo motors are controller via PWM outputs (Ferl and Hills, 2020).

The sluggish execution of the control algorithm was mitigated by using threads to mimic parallel processing of the GPS data, windvane data, and compass data. In effect, the code still executes sequentially, but pre-determined tasks are pulled from the control code and executed at a certain frequency between other tasks.

This prevents execution of computationally intensive code at each iteration of the loop. Global variables are then updated and can be used for decision making and actuating (Ferl and Hills, 2020).

2.7.3 Navigational Control

The vessel senses its current position using the NEO-7M GPS chip and calculates the desired heading required to reach its pre-programmed destination. The desired heading is then compared with the current heading received from the Adafruit IMU. Then an error or required course change is computed. If a course change is required, the control system will determine which type of manoeuvre is applicable and execute it (Ferl and Hills, 2020).

2.7.4 Directional Control

Initially, proportional control was implemented for rudder control based on the magnitude of the required change in, but since proportional control is unresponsive to different rates of change in direction, it yielded overshoot and oscillation in the course change response (Ferl and Hills, 2020).

The final rudder control was implemented with proportional-integral-derivate (PID) control. The P-term responds to the required change in heading, the I-term responds to previous change in heading thus increasing the rudder angle if no change is sensed, and the D-term responds by correcting the instantaneous rate of change to prevent overshoot and oscillation in the response (Ferl and Hills, 2020).

Since Arduino forces a discreet time system, a difference equation estimate is made for the I-term and D-term of the controller. Where $y(t)$ represents the rudder angle, $x(t)$ represents the required course change, and T is the sampling period of the compass. The 'n' functions represents the equivalent discrete time instance equations programmable on the microcontroller (Ferl and Hills, 2020).

$$y(t) = \frac{dx(t)}{dt} \approx y[n] = \frac{1}{T}(x[n] - x[n - 1])$$

$$y(t) = \int_{-\infty}^t x(\tau)d\tau \approx y[n] - y[n - 1] = Tx[n]$$

Final tuning of the PID coefficients were done after conducting water tests and downloading experimental data (Ferl and Hills, 2020).

2.7.5 Sail Control

Little mention was made of what control was applied to the sails. From the flowchart of the control logic, one could see that the sails were trimmed after each iteration of the code, but not how it was achieved.

2.7.6 Modes of Operation

The sailing vessel developed by Ferl et al., only operated in fully autonomous mode where it followed waypoints and trimmed the rudder and sails accordingly to achieve the waypoint. The control is most likely based on the angle of attack.

There was however mentioned that sometimes the vessel may get stuck with its nose pointing into the wind, unable to move (referred to as irons), then a “get out of irons” routine was performed to get the vessel moving and return to the regular control logic (Ferl and Hills, 2020).

Chapter 3

Design Requirements

The design requirements are presented as Technical Performance Measures (TPMs) that are deduced from the objectives in Chapter 1.2. The TPMs listed in Table 3.0.1 below are either expressed as quantitative measurements or qualitative attributes.

Concept Category	TPM ID	Description of Requirement	Value or Qualitative Attribute
Digital Compass	TPM1	Components used	Only magnetometer, accelerometer, and gyroscope
	TPM2	Degree of heading accuracy	Less than 10° error on average
GPS Unit	TPM3	Degree of positional accuracy	Less than 10 meters
Data Logger	TPM4	Ability to write to a MicroSD card	One complete write operation per second
Windvane	TPM5	Ability to indicate wind direction continuously	One sample per second
Control System	TPM6	Dynamically calculate the required trajectory	When course is deviated from, indicate a change in desired heading
	TPM7	Hold the required heading	Less than 20% overshoot
	TPM8	Reach the required destination	Within 10-meter radius

Table 3.0.1: Technical Performance Measures for the design of the unmanned sailing vessel.

Chapter 4

Digital Compass Concept Development

4.1 Background

4.1.1 Axis of Reference

The earth's geomagnetic field is generated from within its outer core. The geomagnetic field is a dipole field that has a large degree of symmetry, and this is used to indicate heading. The dipole magnetic poles are located near the Arctic and Antarctic Poles but are not the same the same geographical places. True North (along the X axis) is indicated in the geographical heading 90° from the direction of gravity (along the Z axis). The dipole axis indicating Magnetic North is offset about 10° from True North as indicated in 4.1.1 below.

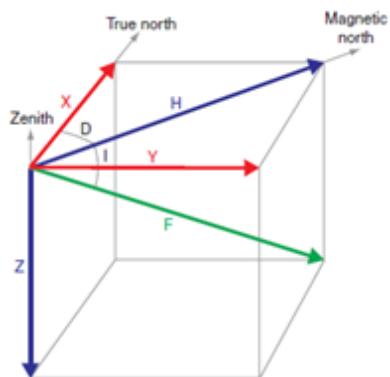


Figure 4.1.1: Directional axis of geography (Langley, 2003)

The difference in degrees between Magnetic and True North (the horizontal component of total force vector F) is referred to as magnetic declination indicated by D in Figure 4.1.1 and is a function of latitude and longitude as shown in Figure 4.1.2. The difference between Magnetic North in the horizontal plane and the total force vector F, is referred to as magnetic inclination.

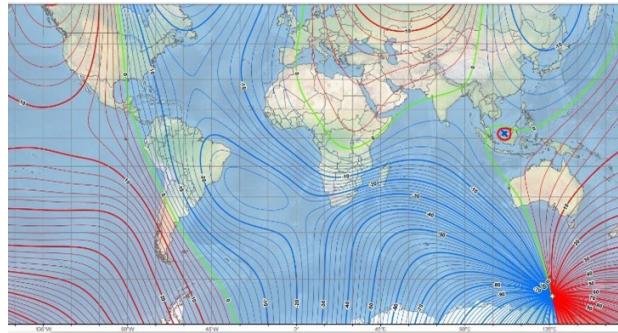


Figure 4.1.2: Magnetic Declination as a function of location (INSERT REF)

4.1.2 Traditional Magnetic Compass Mechanism

A traditional compass relies on the principle that opposite magnetic poles attract. The traditional compass needle too is a magnet, and since earth's geographic North Pole is the southern pole of the earth's magnetic field, a compass needle will always point to magnetic North. (Not geographical north, but in the same general direction barring magnetic declination). Magnetic declination is a function of latitude and longitude since the earth's magnetic field is non-uniform. This could cause a compass to deviate in severe cases up to 180° at extreme latitudes. Traditional compasses are weight balanced over its centre rotation point for its geographical location of use or placed on a gimble to ensure that the compass operates in the horizontal plane only as shown by the X-axis and Y-axis in Figure 4.1.1.

4.1.3 Digital Compass

A digital compass, as found in many vehicles, make use of 3-axis magnetometers to measure the earth's magnetic forces at its location along a 3-axis system. To indicate heading it measures two orthogonal components of the earth's magnetic field as shown in Figure 4.1.3 and calculates the heading by trigonometry using the equation below. The equation, however, is only valid when the magnetometer

is exactly in the horizontal plane. Otherwise, the magnetic field has a partial component B_z too.

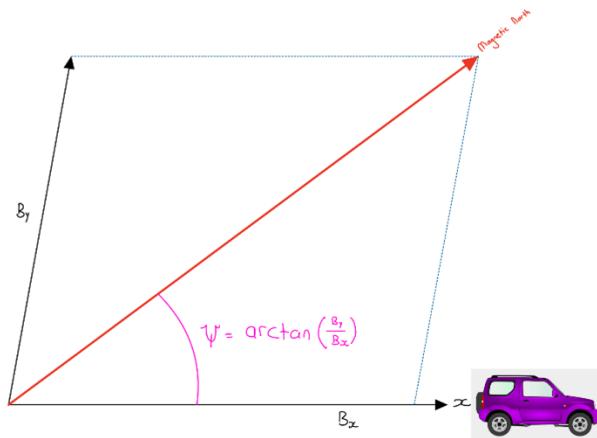


Figure 4.1.3: Explanation of heading

4.2 Hardware

Since the goal of the design is to keep cost and energy usage to a minimum, a low-cost microcontroller the EspressIf ESP-WROOM-32 was selected that has the required communication capabilities. The other reason it was selected is due to the 240 MHz clock speed compared to the Arduino Mega with a 16 MHz clock speed. This processing speed advantage might make operation possible without using two separate microcontrollers for all the functions. It is compatible with the Arduino IDE and is programmable in the C programming language. The ESP32 module has built in Wi-Fi, Bluetooth and I²C/SPI communication capabilities. It can be operated from a 3.7V LiPo battery, only consumes 27 mA at a lower clock speed of 160 MHz, and only costs R150-00.

Since a magnetometer and accelerometer is needed, a MPU9250-6500 9-axis sensor module was chosen that only costs R98-00 which contains a magnetometer, accelerometer, and gyroscope. To interface the two modules, four female-to-female jumpers are needed as well as a USB cable for powering and programming the ESP32 from a computer as illustrated below in Figure 4.2.1.

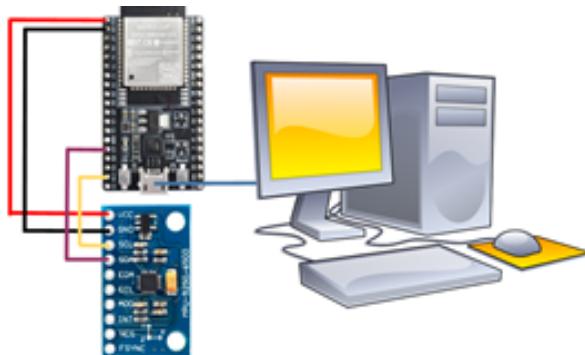


Figure 4.2.1: Digital compass setup

Color of Wire in Figure 4.2.1	ESP32	Digital Compass	Configuration
Red	3.3V	VCC	Positive power lead
Black	GND	GND	Negative power lead
Purple	I ² C (Pin 21)	SDA	Data transmission
Yellow	I ² C (Pin 22)	SCL	Clock synchronization
Blue	MicroUSB	-	Serial communication

Table 4.2.1: Hardware wiring for digital compass

4.3 Software

As mentioned previously, the ESP32 microcontroller can be coded using the Arduino IDE. The default 'math.h' header file is used for calculations and the 'MPU9250.h' by Bolder Flight Systems is used for ease of communication.

The module connected is created as an object that stores incoming I²C communication at a specific address (0x68). To test whether actual data is received, serial communication is utilized. The serial communication is read using Arduino IDE's built-in serial monitor. The ESP32 operates at a baud rate of 115 200.

During the setup function which only runs once before the main loop, a status variable is set to the value of the begin function of the object created. The value is then printed to the serial monitor to indicate the state of communication with the module. A value of zero indicated an unsuccessful communication attempt has been made, and a value of one indicated that successful communication has been established.

4.4 Sensor Calibration Procedure

Although Bolder Flight Systems wrote a calibration function for the MPU9250 module as produced by InvenSense, it is not compatible with the locally available MPU9250 used in this project. Thus, the calibration algorithm had to be designed and implemented.

The calibration procedure is required to account for hard iron and soft iron errors. The calibration procedure for flat plane operation is described in depth in the paper 'Application of Magnetic Sensors for Low-Cost Compass Systems' and is summarized below (Caruso, 2000). When placed in the horizontal field, and rotated about the Z-axis 360°, the output plot of x-data versus y-data should produce a near perfect circle centred around zero. However, if uncalibrated, the output plot will most likely resemble an oval rather than a circle which will not be centred. The work of Caruso however contains errors regarding the calibration method, since it assumes that all uncalibrated magnetometer values are greater than zero.

Since this project cannot guarantee horizontal plane operation, the process is adapted to ensure calibration of the z-axis of the magnetometer as well. The compass is swung around in the air in figure eight motions across all possible axis to record maximum and minimum values for all orientations for 60 seconds. The raw magnetometer data is recorded as 'X', 'Y', and 'Z' columns using comma delimiters to differentiate between each reading. The serial monitor data is saved to a CSV file which is then analysed in MS Excel.

The maximum and minimum of each column is calculated and are used to calculate offsets so that data can be centred, and scale factors, so that all axes are of the same radius. This translates to Caruso's concept of producing a perfect circle when representing the magnetometer data of two axes or a perfect sphere when representing magnetometer data of three axes.

4.4.1 Uncalibrated Magnetometer Data

As seen in Figure XX, Figure XX, and Figure XX, the uncalibrated magnetometer readings when represented by two axes, represent oval shaped data that are not centred about zero. As seen in Figure 4.4.1a, when the uncalibrated magnetometer readings are represented by three axes, it resembles an ellipsoid that is not centred about zero. The calibration ideal can be seen in Figure 4.4.2 showing a multi-colour sphere centred around zero.

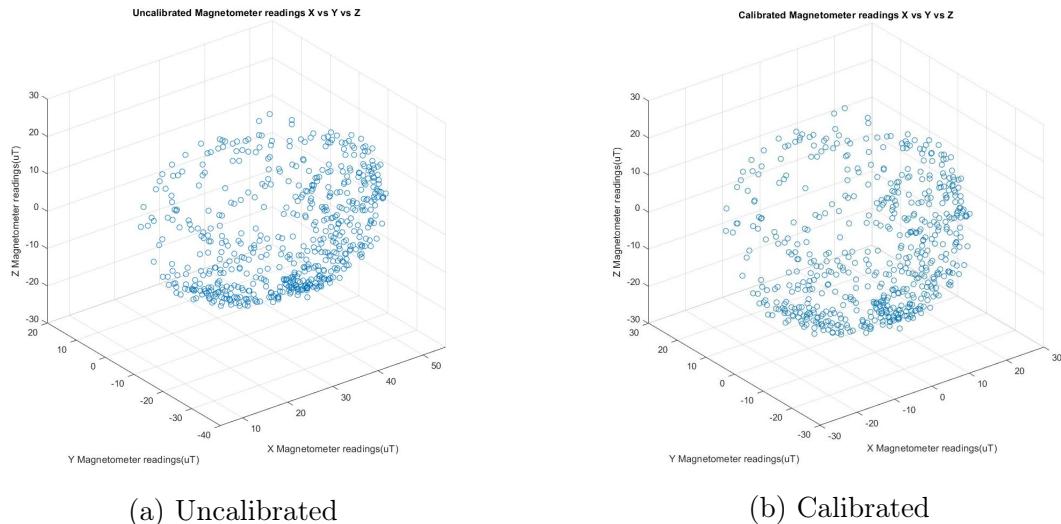


Figure 4.4.1: Magnetometer readings: X vs Y vs Z.

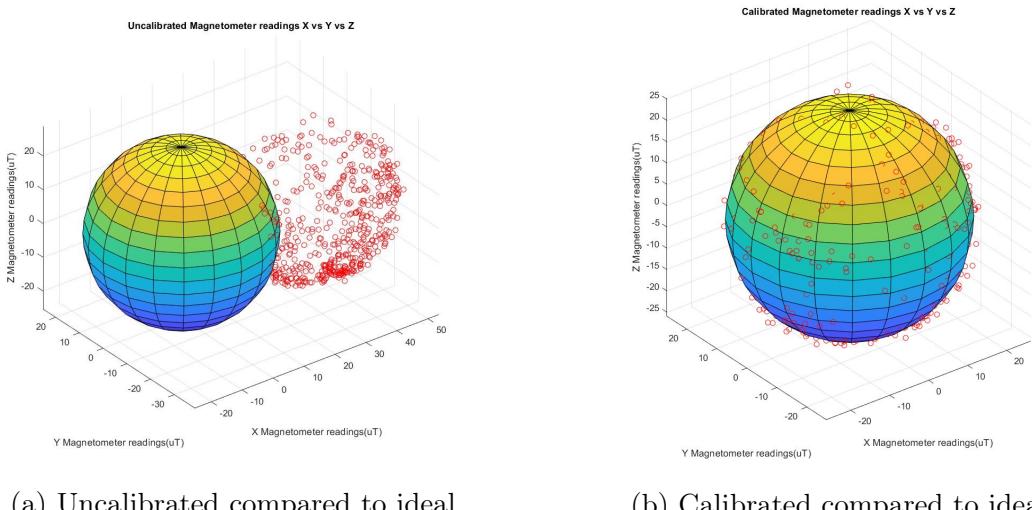


Figure 4.4.2: Magnetometer readings: X vs Y vs Z showing real compared to ideal

4.4.2 Calibration Factor Calculation

The factors to calibrate the respective axes, are calculated below.

The values of table 2 above do not correspond to the graphs shown in Figure XX, Figure XX, Figure XX, and Figure 4.4.1, because the readings were taken at different physical locations. The calibration procedure must be performed after every start up. The radius of each axis is calculated by:

Axis	Maximum Value (uT)	Minimum Value (uT)
X	73.271	24.951
Y	27.507	-25.91
Z	34.599	-16.703

Table 4.4.1: Uncalibrated magnetometer maximum and minimum values

$$Axis_{radius} = \frac{Axis_{maximum} - Axis_{minimum}}{2} \quad (4.4.1)$$

$$Radius_{average} = \frac{Radius_x + Radius_y + Radius_z}{3} \quad (4.4.2)$$

$$Axis_{offset} = \frac{Axis_{maximum} + Axis_{minimum}}{2} \quad (4.4.3)$$

$$Axis_{scaler} = \frac{Radius_{average}}{Axis_{radius}} \quad (4.4.4)$$

Axis	Radius of axis	Center of axis	Scaler
X	24.16	49.111	1.0557
Y	26.7085	0.7985	0.955
Z	25.651	8.9465	0.994

Table 4.4.2: Results of calibration calculation

The calibrated values are calculated by:

$$Axis_{calibrated} = (Axis_{measured} - Axis_{offset})Axis_{scaler} \quad (4.4.5)$$

The accelerometer is calibrated by the same general method, but since the maximum acceleration is $\pm 9.81 m/s^2$, this centre of axis should be zero and the radius of each axis should be equal to 9.81. The gyroscope has an autocalibration-feature that is activated upon power up, so no further calculations are needed.

4.4.3 Calibrated Magnetometer

As seen above in Figure 11, Figure 12, and Figure 13, the calibrated magnetometer data when represented by two different axes, form a near perfect circle. This confirms the success of the calibration process. When used to construct a three-dimensional shape, the data yields a near perfect sphere as seen in Figure 4.4.1b.

4.5 Compensation Algorithms

When the magnetometer is not exactly horizontal, i.e., there is an induced roll and/or pitch angle, the tilt must be compensated for to provide an accurate heading. To measure roll and pitch angles, an accelerometer is needed to sense the direction of gravity. As shown below, X-Y-Z is the fixed coordinate system and X'-Y'-Z' is the roll-pitch coordinate system. Clockwise rotation about the X-axis is defined as roll-angle (θ) and clockwise rotation about the Y-axis is defined as pitch-angle (ϕ) with orientations as shown in Figure 4.5.1 below.

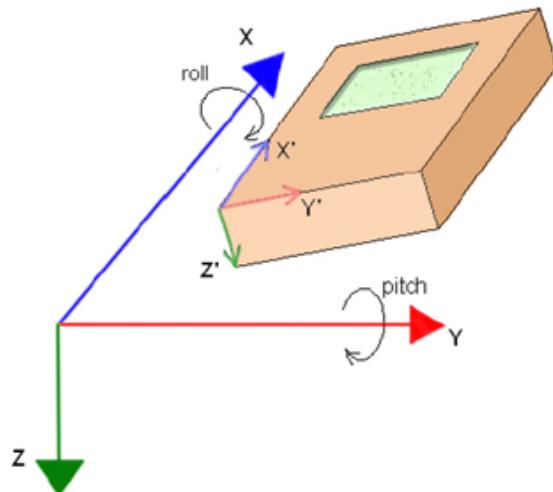


Figure 4.5.1: Tilt compensated axis system

4.5.1 Calculation of Pitch Angle

Since the acceleration of gravity is constantly 9.81 m/s^2 downwards (positive Z-direction), the accelerometer after calibration, when placed on a flat surface free of

any linear acceleration, should read 9.81 m/s² along its Z-axis.

When inducing pitch, the pitch angle causes components of the gravitational acceleration (Z) to be experienced along the X-axis. The pitch angle ϕ can be calculated by arctangent of measured acceleration along the X-axis divided by measured acceleration along the negative Z-axis.

$$\phi_{acc} = \arctan\left(-\frac{A_z}{A_x}\right) \quad (4.5.1)$$

The measured values are of pitched axes X', Y', and Z'. To rotate the pitched axes back to the principal axes X, Y, and Z, apply the following equations.

$$X = X' \cos(\phi_{acc}) + Z' \sin(\phi_{acc}) \quad (4.5.2)$$

$$Y = Y' \quad (4.5.3)$$

$$Z = Z' \cos(\phi_{acc}) - X' \sin(\phi_{acc}) \quad (4.5.4)$$

Thus, the equations characterized by a Pitch Rotation Matrix R_p :

$$R_p = \begin{bmatrix} \cos(\phi_{acc}) & 0 & \sin(\phi_{acc}) \\ 0 & 1 & 0 \\ -\sin(\phi_{acc}) & 0 & \cos(\phi_{acc}) \end{bmatrix} \quad (4.5.5)$$

So that:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_p \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (4.5.6)$$

4.5.2 Calculation of Roll Angle

The accelerometer after calibration, when free of any linear acceleration, is used to calculate roll angle. When inducing roll, the roll angle causes components of the gravitational acceleration (Z) to be experienced along the Y-axis. The Roll angle (θ) can be calculated as the arctangent of measured acceleration along the Z-axis divided by the measured acceleration along the Y-axis.

$$\theta_{acc} = \arctan\left(\frac{A_z}{A_y}\right) \quad (4.5.7)$$

The measured values are of rolled axes X', Y', and Z'. To rotate the rolled axes back to the principal axes X, Y, and Z, apply the following equations.

$$X = X' \quad (4.5.8)$$

$$Y = Y' \cos(\theta_{acc}) - Z' \sin(\theta_{acc}) \quad (4.5.9)$$

$$Z = Z' \cos(\theta_{acc}) + Y' \sin(\theta_{acc}) \quad (4.5.10)$$

Thus, the equations characterized by a Roll Rotation Matrix R_r :

$$R_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{acc}) & -\sin(\theta_{acc}) \\ 0 & \sin(\theta_{acc}) & \cos(\theta_{acc}) \end{bmatrix} \quad (4.5.11)$$

So that:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_r \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (4.5.12)$$

4.5.3 Compensation for Linear Acceleration

The accelerometer provides excellent long-term accuracy to pitch and roll angle measurements, since the gravitational acceleration (g) never changes. In the short-term however, it can be highly inaccurate since it mistakes any linear acceleration and vibration as rotation as seen in Figure 4.5.2 below.

As the Dragonflite95 vessel might experience vibration due to wind in the sails, compensation is necessary to ensure that effects of high frequency components are reduced. The simplest form of a digital filter, a difference equation, is implemented. The difference equation uses present signal values in conjunction with past signal values (feedback).

$$A_{n-acc} = 0.2 \cdot A_n + 0.8 \cdot A_{n-1} \quad (4.5.13)$$

$$\phi_{acc-filtered}[n] = 0.2 \cdot \phi_{acc}[n] + 0.8 \cdot \phi_{acc}[n-1] \quad (4.5.14)$$

$$\theta_{acc-filtered}[n] = 0.2 \cdot \theta_{acc}[n] + 0.8 \cdot \theta_{acc}[n-1] \quad (4.5.15)$$

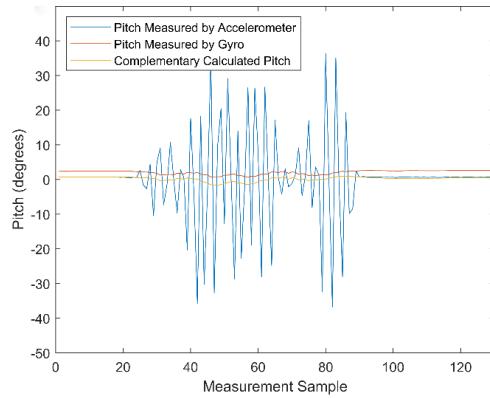


Figure 4.5.2: Accelerometer vs gyro vs filtered pitch during vibration

4.5.4 Gyroscope-based Pitch and Roll

The gyroscope provides excellent short-term accuracy since it can measure angular velocity and translate that into rotational angles. The gyroscope however has no sense of reference such as gravity, and thus displays drift when stationary over time as seen in Figure 4.5.3 below. The device is horizontal, it is then picked up, moved, and placed horizontal again. Within 50 samples, a drift larger than 10° is recorded.

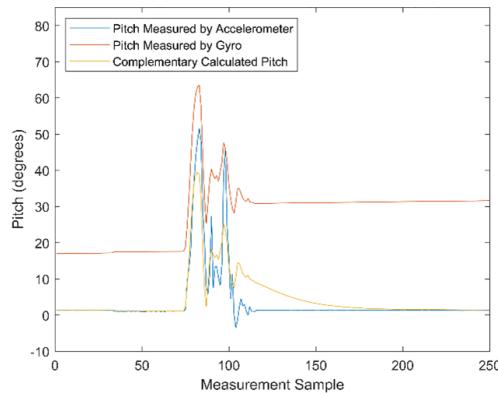


Figure 4.5.3: Accelerometer vs gyro vs filtered pitch displaying gyro drift

To translate measured data into pitch and roll angles, apply the following equations.

$$\phi_{gyro}[n] = \phi_{gyro}[n-1] + \omega_{gyro-about-y} \cdot dt \quad (4.5.16)$$

$$\theta_{gyro}[n] = 0.2 \cdot \theta_{gyro}[n] + \omega_{gyro-about-x} \cdot dt \quad (4.5.17)$$

Where n represents the current angle, n-1 represents the previous angle, ω represents the angular velocity of the respective axis, and dt represents the time elapsed between consecutive samples.

4.5.5 Complimentary Usage of Accelerometer and Gyroscope Data

By using a combination of both the accelerometer and gyroscope, the long-term accuracy of the accelerometer can be kept while also maintaining short-term accuracy of the gyroscope. The function for best combination of both inputs so that there are also less high frequency components were found to be:

$$\phi_{comp}[n] = 0.95 \cdot (\phi_{comp}[n - 1] + \omega_{gyro-about-y} \cdot dt) + 0.05 \cdot \phi_{acc-filtered}[n] \quad (4.5.18)$$

$$\theta_{comp}[n] = 0.95 \cdot (\theta_{comp}[n - 1] + \omega_{gyro-about-x} \cdot dt) + 0.05 \cdot \theta_{acc-filtered}[n] \quad (4.5.19)$$

This difference equation gives a 95% weight to the previous pitch or roll angle plus change in angle as determined by the gyroscope, and a 5% weight to the filtered pitch or roll angle as determined by the accelerometer. This will ensure that the short-term response is dictated by the gyroscope, and the drift will be eliminated since it will strive to the long-term value produced by the accelerometer. The effect of these factors can be seen below in Figure 4.5.4 when inducing a sudden pitch. The complementary function follows the short-term response of the gyroscope but tends towards the long term accuracy of the accelerometer.

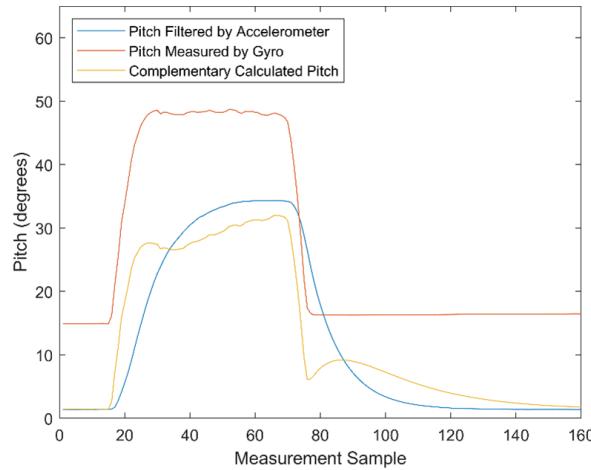


Figure 4.5.4: Accelerometer vs gyro vs filtered pitch when inducing sudden pitch

4.5.6 Construction of the Tilt-Compensation Matrix

The total translation of the rotated coordinate system back to the principal axes can be calculated using the pitch- and roll rotation matrix as shown below.

So that:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_p \ R_r \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (4.5.20)$$

Thus,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos(\phi_{comp}) & 0 & \sin(\phi_{comp}) \\ 0 & 1 & 0 \\ -\sin(\phi_{comp}) & 0 & \cos(\phi_{comp}) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{comp}) & -\sin(\theta_{comp}) \\ 0 & \sin(\theta_{comp}) & \cos(\theta_{comp}) \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (4.5.21)$$

Thus,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos(\phi_{comp}) & \sin(\phi_{comp})\sin(\theta_{comp}) & \sin(\phi_{comp})\cos(\theta_{comp}) \\ 0 & \cos(\theta_{comp}) & -\sin(\theta_{comp}) \\ -\sin(\phi_{comp}) & \sin(\theta_{comp})\cos(\phi_{comp}) & \cos(\phi_{comp})\cos(\theta_{comp}) \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (4.5.22)$$

4.5.7 Magnetic Heading

The heading angle is measured clockwise from Magnetic North in the horizontal plane. The localized heading is given by:

$$\text{Localheading} = \arctan\left(\left|\frac{Y}{X}\right|\right) \quad (4.5.23)$$

Sign of tilt-compensated X-magnetometer data	Sign of tilt-compensated Y-magnetometer data	Magnetic heading
+	+	Local heading
-	+	Local heading + 90°
+	-	360° - Local heading
-	-	180° + Local heading
0	+	90
0	-	270

Table 4.5.1: Magnetic heading calculation

4.5.8 True Heading

True heading differs from magnetic heading by a location-specific magnitude called magnetic declination. For simplification, this project will assume that magnetic declination is constant. This simplification will not have a great effect on performance since the vessel will only be tested on a small lake. The magnetic declination is determined to be -25.72° from www.magnetic-declination.com. The effect of the negative sign is that true North lies West of magnetic North. Thus:

$$\text{True heading}(\psi) = \text{Magnetic Heading} - 25.72^\circ \quad (4.5.24)$$

Chapter 5

Additional Sensor Implementations

5.1 GPS Unit

A Global Positioning System (GPS) is required for the system to be able to determine its location. This function is not only required for the post-test verification of the system, but also to determine the required change in heading to keep to the course.

5.1.1 Hardware

Due to spatial constraints and monetary limitations, the Neo M8N GPS chip was selected for its low price, and small size and claimed 2.5m horizontal position accuracy (INSERT DATASHEET SRC). It required only four connections and communicates with the microcontroller via UART and has a dedicated TX and RX channel, while constantly outputting strings of data at a baudrate of 9600. A limitation of the GPS is that it only returns latitude and longitude data containing six decimal places, which may result in a large error if used over small distances.

Color of Wire	ESP32	Neo M8N	Configuration
Red	3.3V	VCC	Positive power lead
Black	GND	GND	Negative power lead
Brown	RX2	TX	RX_TX
Orange	TX2	RX	TX_RX

Table 5.1.1: Hardware setup of GPS

5.1.2 Software

The libraries 'SoftwareSerial.h' and 'TinyGPS++.h' by Mikal Hart are used for GPS communication with the microcontroller. As the GPS operates at a baudrate of 9600, and the microcontroller operates at a baudrate of 115200, the microcontroller's second UART channels are used, configured at a rate of 9600. The serial TX and RX lines are defined in code by specifying the appropriate pin numbers. An object 'gps' is also created for use in code with the TinyGPS++ library.

Two character-arrays are also defined. The first array, GPSheader, consists of 80 characters, allowing for definition of all data that will be read in, such as time, speed, latitude, or longitude. The second character array, GPSfile, consisting of 200 characters, contains character casted float type data received from the GPS.

5.1.3 Explanation of TinyGPS++

Without the use of a library, slave-mode-only serial communication is initiated without stop. There is a constant influx of string type data. The data 'sentences' are labelled by use of pre-determined identifiers of the Nasional Marine Electronics Association (NMEA). The identifiers can be seen below.

NMEA identifier	Meaning of identifier
\$GNGGA	Time, position, and GPS fix of receiver
\$GNGLL	Time, position and GPS fix status
\$GNGSA	ID of satellites connected to
\$GPGSV	Elevation and azimuth heading
\$GNRMC	Time, date, position, course, and speed
\$GPTXT	GPS antenna status

Table 5.1.2: NMEA sentence identifier meanings

To translate these sentences, a search function is needed to search the serial communication for the correct identifier, and according to its format, save a certain range of bits at a pre-determined offset into a variable for use on the microcontroller. To expedite the process, TinyGPS++ is used as it contains pre-authored functions to return float type data which can be used for calculations. The float type data can be saved to a variable by using the 'object.variable()' function.

5.1.4 Explanation of GPS reading function

The GPS function consists of a while loop with a nested if statement and a condition flag. This ensures that one updated GPS location is saved, and that it occurs only once before returning to the main function at least once every second by using a timer operation.

While there is not serial communication between the GPS chip and the microcontroller, serial data is read. If there is an update from its previous location or change in time, write a string of data that comprises of the required data by use of the `sprint()` function to write data to file, or '`object.variable()`' function for saving of float type data for calculation purposes.

5.1.5 Required heading determination

A pre-specified location is programmed into the microcontroller memory by providing its latitude and longitude. To calculate distance between two geographic points, the Haversine formula is used. The Haversine formula is required to calculate the distance over the surface of a sphere between two points on the surface of the sphere. The Haversine formula will however result in an error since the earth does not resemble a perfect sphere, but the error is less than 0.5% in distance calculation (INSERT SOURCE). (Vincenty model is computationally intensive and iterative, i.e. it does not guarantee convergence).

Let A resemble the current boat coordinate point, and B resemble the pre-determined destination. Coordinates consist of latitude (Lat) and longitude (Lon). Let β resemble bearing. Then as shown before:

$$\beta = \arctan\left(\frac{Y}{X}\right) \quad (5.1.1)$$

But now, X and Y are representative of distances along the surface of the earth.

$$X = \cos(B_{Lat}) \cdot \sin(A_{Lon} - B_{Lon}) \quad (5.1.2)$$

$$Y = \cos(A_{Lat}) - \sin(A_{Lat}) \cdot \cos(B_{Lat}) \cdot \cos(A_{Lon} - B_{Lon}) \quad (5.1.3)$$

5.2 Data Logger System

As this project is control-system orientated, and real-time parameter tuning is not possible due to the vessel being sealed for operation in water, post-processing of data is required to verify the control system's ability to identify a scenario-specific mode of operation and dynamically alter its heading and sail angle to reach its goal.

5.2.1 Hardware

Due to the size constraint of the small opening in the deck of the DragonFlite95 vessel through which all electronics must fit to be housed within the hull or on top of the deck, a microSD card breakout board with the smallest physical footprint will be selected. A Sparkfun Level Shifting breakout board was chosen that features operation on 3.3V logic level without a voltage rectifier.

Color of Wire	Meaning of identifier	Sparkfun MicroSD	Configuration
Red	3.3V	VCC	Positive power lead
Black	GND	GND	Negative power lead
Orange	D5	CS	Chip select
Yellow	D23	DI	MOSI
Blue	D18	SCK	Serial clock
Green	D19	DO	MISO

Table 5.2.1: Hardware setup of datalogger system

5.2.2 Software

The communication with the SD card module is configured in SPI mode. This requires a connection to VCC for power, GND for ground, CS for chip select, SCK for clock synchronization, DI for Master In Slave Out (MISO), and DO for Master Out Slave In (MISO). The software requires the use of the standard Arduino Corporation SD.h, SPI.h, and FS.h libraries in the functions written. The SD card also requires formatting to FAT32 file system before use.

writeFile() adapted from Rui Santos

The writeFile function is used to write data to a file initially once it is created. It takes the address of the file system, the root path of the file and a data message string that should be written to file as input arguments.

appendFile()

The appendFile function is used to write data to a file once it has been created and the data column headers has already been writted by the writeFile() function. It too takes the address of the file system, the root path of the file and a data message string as input arguments.

SD.open()

The SD.open() function is used to create a file which can be written to. It takes the file path of the desired file with the corresponding name and file type as a single string input argument. The function is housed in the SD.h library.

Chapter 6

Physical Hardware Development

6.1 Windvane Development

A requirement of the proposed project is that the vessel may only be driven by harnessing propulsion power from wind. To harness wind power and apply the harnessed power to the correct sail angle, the direction of the apparent wind needs to be known.

The easiest method of achieving this objective is by using a windvane. Since this project focusses on the use of a windvane rather than the design of a windvane, the windvane was designed by integrating key knowledge from previous research and by mimicking the Davis windvane used by Ferl and Hills (Ferl and Hills, 2020).

6.1.1 Sense of Direction

To sense direction, a calibrated endless potentiometer or digital compass can be utilized. Since this project already requires the development of a digital compass, it was decided to use another digital compass that would fit onto the windvane. This would provide the same results as the Davis windvane, but at a much lower cost.

The windvane base would be integrated with a 6-wire slipring to ensure conductance of the signal while not limiting rotation.

The size of the wing at the rear of the windvane was determined experimentally to produce enough force at low wind speeds to ensure that the static friction of the slipring is overcome, and a turn is induced in the direction of the wind.

The separate components were 3D-printed with PLA, assembled, and fitted to the DragonFlite95 vessel. The windvane comprises of two 3D printed mounting structures that fits over the slipring base and shaft. The bottom structure mounts onto the mast of the DragonFlite95, and the top structure allows the digital compass to be mounted on top.

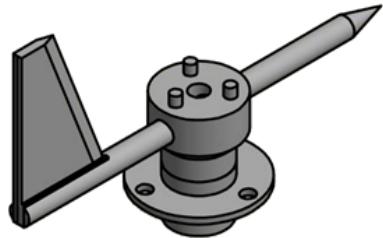


Figure 6.1.1: Windvane assembly

6.2 PCB Design

To fix all components into place, ensure that no space is wasted by wires, and ensure contact without breadboard capacitance, a PCB was designed using easyEDA, and printed at the E&E Faculty Workshop, Stellenbosch. The design can be found in Appendix XXX.

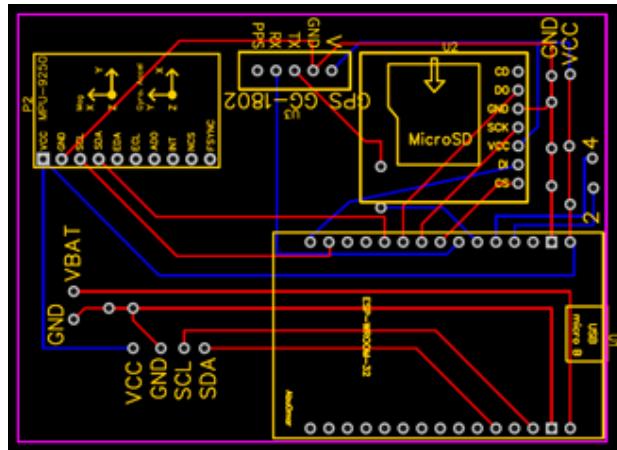


Figure 6.2.1: Schematic of PCB

6.3 Housing Design

Spatial constraints in the hull of the DragonFlite95 vessel caused the need for a housing for the electronics and battery. This housing must be watertight, but still non-permanent as data needs to be analysed between water tests. The housing was designed using dimensions of the PCB, allowing space for the battery to fit below the PCB. The drawings for the full design can be seen in Appendix X.X.

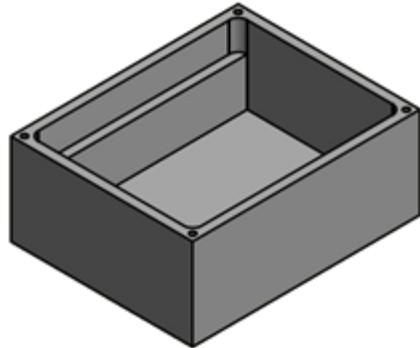


Figure 6.3.1: Assembly of electronics housing

Chapter 7

Control System Design

The design of a theoretical control system to allow for autonomous sailing entails reviewing sailors' logic, sailing techniques and application thereof, to remove the need for the sailor.

The control of a sailing vessel can be divided into four main parts - navigation, sailing limitations, directional control, propulsion control, and motor control.

7.1 Navigation

7.1.1 GPS Navigation

The controller code contains preprogrammed waypoint(s) and/or destination. From the GPS, the current latitude, longitude, and velocity is received at a rate of 1 Hz. The controller then compares the current position (latitude and longitude) with the desired position. The desired heading and distance to the waypoint is calculated using the Haversine formula.



Figure 7.1.1: Navigation function block diagram

7.1.2 Compass Navigation

The current heading is calculated with data obtained from the IMU as explained in chapter XX. The raw magnetometer, accelerometer, and gyroscope data is sampled from the IMU at a rate of 6 Hz, is tilt-compensated, and then the current true heading is determined.

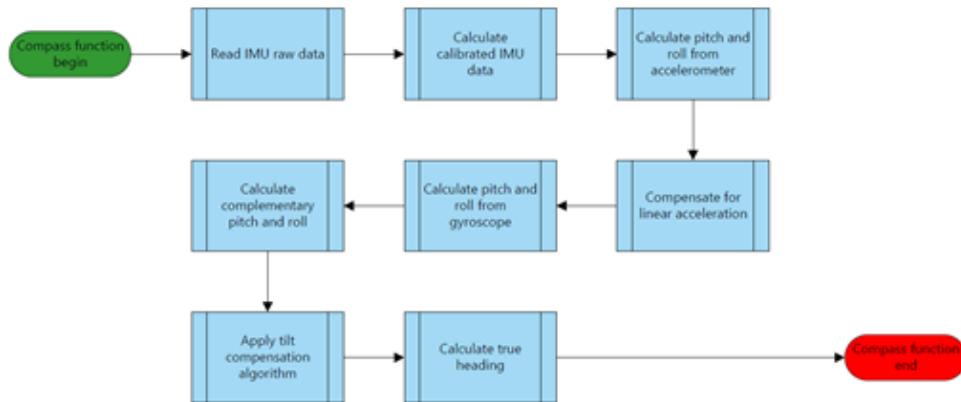


Figure 7.1.2: Compass function block diagram

7.1.3 Waypoint Navigation

If the distance from the current position to the current waypoint is less than 10 meters, the waypoint must be updated to the next waypoint to commence navigation to the next waypoint. If there are no other waypoints programmed, the vessel control system will shut down. In future research, a function can be developed to keep circling the current waypoint, or to turn helm into the dead zone and attempt to hold the vessel stationary.

7.2 Sailing Limitations

The traditional sailing vessel, since only utilizing wind energy for propulsion, has certain limitations as to the direction it can travel in with respect to the wind direction. A no-sail zone or dead zone is defined as the range about the angle of wind into which the vessel cannot feasibly sail. The dead zone magnitude is vessel specific, but a safe assumption of 45° toward either side is made since the goal of this project is not to sail fast, but to sail feasibly.

If a desired heading is found to be within the dead zone, the vessel cannot achieve the desired heading without applying the tacking technique discussed in chapter XX. The limits to both sides of the dead zone are calculated as the limit $\pm 5^\circ$ and one of two desired headings are set on an alternating basis. The current position is recorded as the intermediate waypoint location. A flag is set to disregard the desired waypoint calculated by the GPS.

If the distance between the current position and the intermediate waypoint location exceeds 50 meters, then the desired heading is set to the limit on the far side of the dead zone. This process is repeated until the desired heading as calculated by the navigation function is not within the dead zone, or until the vessel is within 10 meters of its next waypoint.

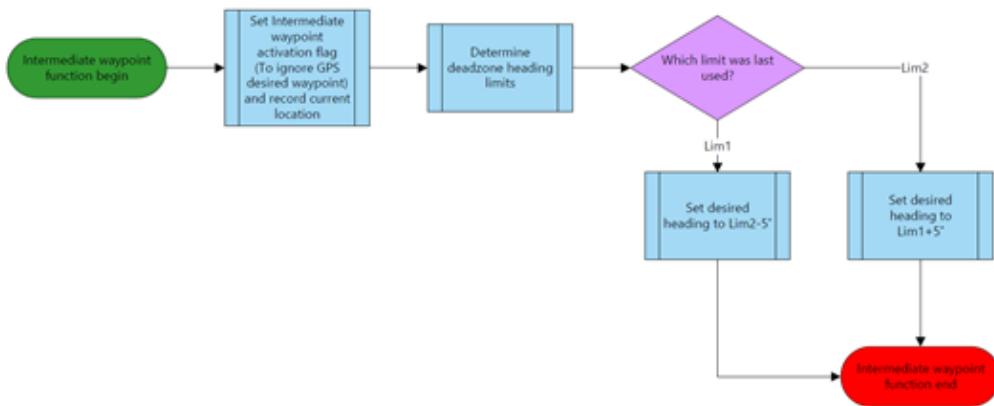


Figure 7.2.1: Tacking navigation function block diagram

7.3 Directional Control

The heading of a sailing vessel is mainly influenced by two factors assuming there are no currents present: the position of the rudder, and the moment generated by a difference in lift force on the main sail and the jib. Due to the sail-setup of the DragonFlite95, controlling the generated moment is not possible since sails can only be retracted, not pushed out. Thus, directional control will be applied through actuation of the rudder.

7.3.1 Control Parameters

The Desired Heading (setpoint) is calculated dynamically by the navigation function, the Current Heading is calculated dynamically by the compass function (input), and the error signal is calculated as the absolute value of the difference between these two, respectively.

The navigation function calculates desired heading as a numerical float ranging from 0° to 360° while the compass function calculated current heading a numerical float ranging from -180° to 180° . Therefore, a check is added to see whether the error signal is greater than 180° and if so, 360° is subtracted so that heading error can be determined to be less than 180° anticlockwise or clockwise from North.

If the error signal is greater than zero, a steer to port side is required and if the error signal is less than zero, a steer to starboard side is required. Using proportional control, a rudder angle proportional to the error signal is calculated and mapped to by PWM as derived in chapter XX.

$$\text{RudderAngle} = K_p \cdot \text{ErrorSignal} \quad (7.3.1)$$

The low velocity of the vessel, its smooth operation in water, and the high sample rate of the control code should allow the use of a proportional controller only.

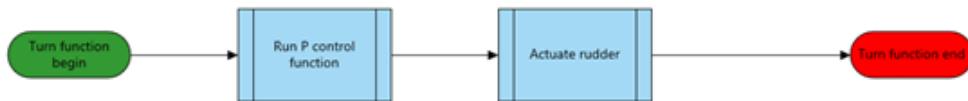


Figure 7.3.1: Turn function block diagram

If a tack is required, then a more aggressive P control function will be run in order to ensure crossing of the dead zone without getting stuck in irons.



Figure 7.3.2: Tack function block diagram

7.4 Propulsion Control

The propulsion of the DragonFlite95 is solely due to wind forces exerted on the main sail and jib. As such, the angle of attack plays a great role in the ability of the vessel to move forward. The goal of this project is not to achieve and operate at maximum velocity, but to prove the concept of autonomous sailing, so the point of sail will be determined and classified to perform one of 3 sail trims.

If the angle of attack is less than 60° , the point of sail is close hauled, and sails are released to 10° . If the angle of attack is greater than 60° and less than 130° , the point of sail is close reach, beam reach or broad reach, and the sails are released to 45° . If the angle of attack is greater than 130° , the point of sail is training or training run, and the sails are released to its maximum at 90° .

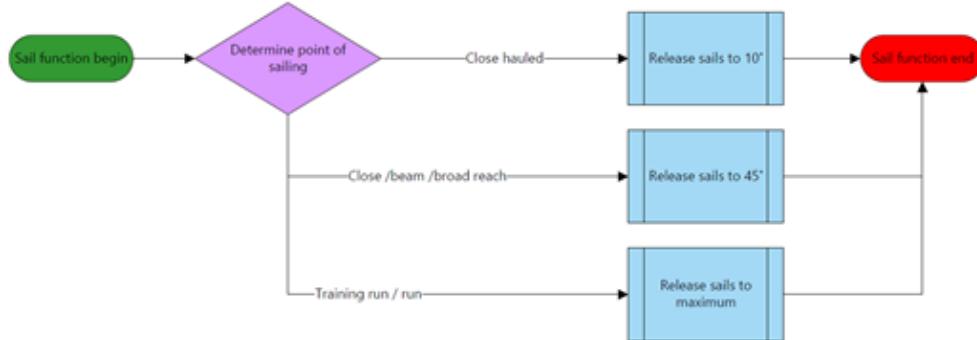


Figure 7.4.1: Sail control function block diagram

7.5 Servo Motor Control

The DragonFlite95 sailing vessel houses two 6V servo motors to control the rudder and sail winch. The servo motors are powered by the 6V battery pack inside the hull and are controlled by PWM signals.

Using the ESP32, a specific pulse width is sent to the motor as coded in the Arduino IDE. A specific pulse width translates to a specific position of a servo motor. The rudder is connected to the servo via a connecting shaft, so the physical rotation of the servo is not equal to the physical rotation of the rudder. This length of the connecting shaft is adjustable so that the minimum servo position entails a rudder position of -60° , and that the maximum servo position induces a rudder position of 60° . The winch servo utilizes a gear system, so that the minimum servo

position has the sails fully retracted (0°), and the maximum servo position has the sails fully released (90°).

The specific PWM value required to achieve these positions are servo-specific and unknown for the DragonFlite95 servos since the manufacturer of the DragonFlite95 did not intend for the vessel to be used for non-RC applications. The values are determined experimentally by sending the servo sequentially incremented PWM values and recording the rudder or sail movement.

The 'PWMWriteValue' and physical movement of the system is related by a linear factor and is determined as shown:

Servo application	Minimum position($^\circ$)	Minimum 'PWMWriteValue'	Maximum position($^\circ$)	Maximum 'PWMWriteValue'
Rudder	-60	75	60	130
Sail	0	90	90	200

Table 7.5.1: Servo control PWM value range

Thus, the relationship between 'PWMWriteValue' and rudder position can be determined to by:

$$PWM_{WriteValue} = Pos \cdot \frac{PWM_{max} - PWM_{min}}{Pos_{max} - Pos_{min}} + \frac{PWM_{max} + PWM_{min}}{2} \quad (7.5.1)$$

And the relationship between 'PWMWriteValue' and sail position can be determined by:

$$PWM_{WriteValue} = Pos \cdot \frac{PWM_{max} - PWM_{min}}{Pos_{max} - Pos_{min}} + PWM_{min} \quad (7.5.2)$$

7.6 Control System Summary

The block diagram of the control system during a normal (no tack required) sailing voyage is demonstrated below in Figure 7.6.1.

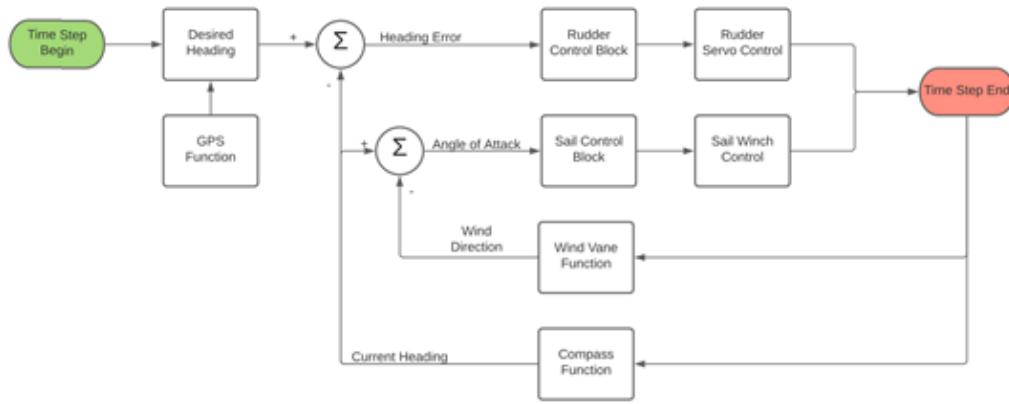


Figure 7.6.1: Standard operating control block diagram

The pseudo-code flow diagram for the control system can be seen below. The code is executed sequentially using for-loops. After the setup and initialization, the void loop is executed as shown below in Figure 7.6.2.

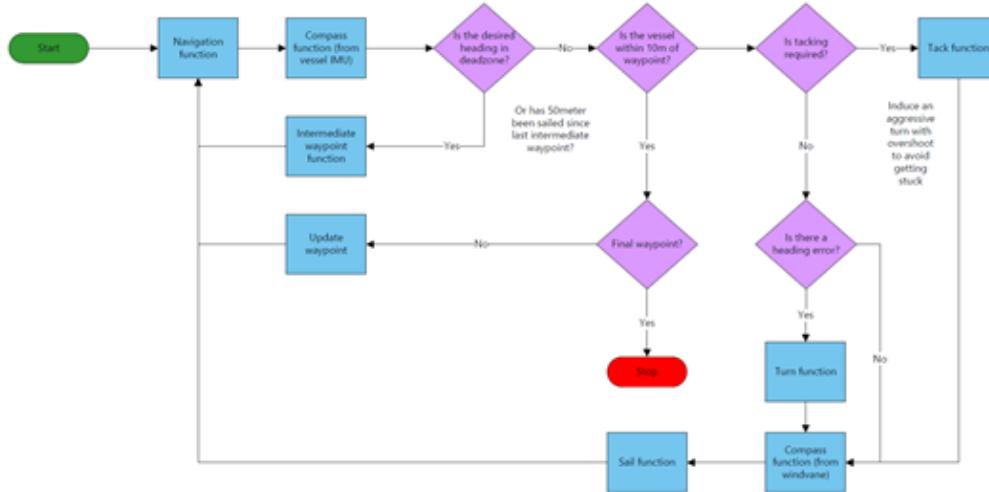


Figure 7.6.2: Void loop block diagram

7.7 Limitations of the Control System

While reviewing the control system design and implementing it via the Arduino IDE on the ESP32, certain limitation surfaced.

Since the ESP32 operates in a discrete time-system, it is necessary to discretize

discrete time-system equivalents for integral and derivative control as suggested by Ferl and Hills. The equations are estimated using the difference equations below.

$$PID_{output} = K_p \cdot ErrorSignal + K_i \cdot ErrorSum + K_d \cdot ErrorRate \quad (7.7.1)$$

Where

$$ErrorRate = \frac{ErrorSignal - PrevErrorSignal}{\Delta T} \quad (7.7.2)$$

$$ErrorSum = \sum_{i=1}^n ErrorSignal \quad (7.7.3)$$

The first limitation is that the low-cost sensors available at the time, do not provide good accuracy or repeatability. This entails that when tracking the cumulative error, an error loss will occur defying the working of the integral control in the system.

The second limitation is that sail winch servo actuations require a large delay of 200ms due to the slow change rate of the winch servo. This causes a significant reduction in sample rate of the sensors and refresh rate of the program. It was therefore decided to only allow three point of sail classification, thereby reducing the amount of sail position changes to rather give processing time to the navigational and directional control functions.

Chapter 8

Testing

Before verifying the working of the entire system in an aquatic environment, individual tests will be conducted on each of the sub-systems to verify its working and ensure that any integration issues that may arise, are remediable by software in the combined program.

8.1 Digital Compass

A test rig as shown below in Figure 8.1.1 was set up to verify the working of the digital compass. A handheld magnetic compass was used as an absolute reference, an iPhone 11 was used as a first digital comparison, and a Samsung Galaxy S8+ was used as a second digital comparison.



Figure 8.1.1: Digital compass test setup

8.1.1 Experimental Setup

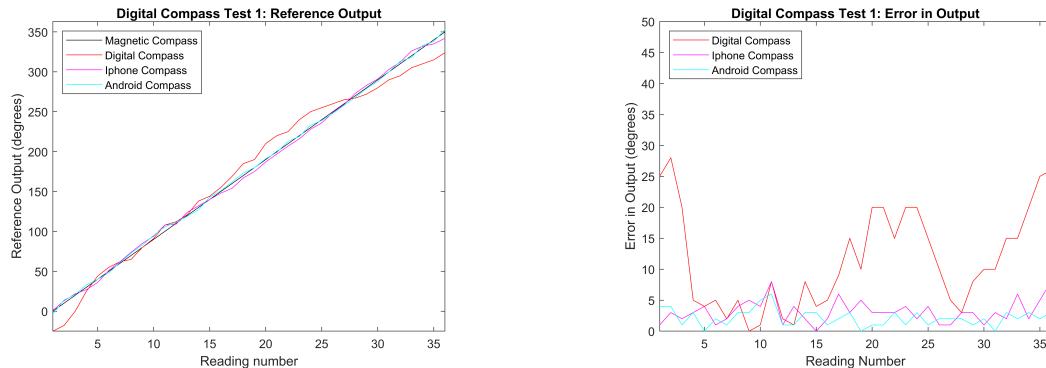
The four devices were spaced out evenly along the length of the wooden beam leaving 300mm space between devices. This distance was found experimentally during testing as the minimum distance where devices have no effect on each other's readings.

Three tests were conducted at the same location, on different days, by the same tester. The test rig was aligned to face magnetic North as indicated by the magnetic compass, and heading readings were taken from the digital compass, iPhone, and android. The heading of the test rig was increased at increments of 10° and readings recorded. A summary of the readings obtained can be seen below in table XX.

8.1.2 Results

Source of error-> Test number:	iPhone error	Android error	Digital compass error
1	3.1081	2.1892	11.8649
2	10.7297	9.7297	4.2432
3	5.2162	4.3243	5.8432
Average	6.3513	5.4144	7.3171

Table 8.1.1: Compass Test Results



(a) Test 1: Reference output

(b) Test 1: Error signal

Figure 8.1.2: Digital compass test 1

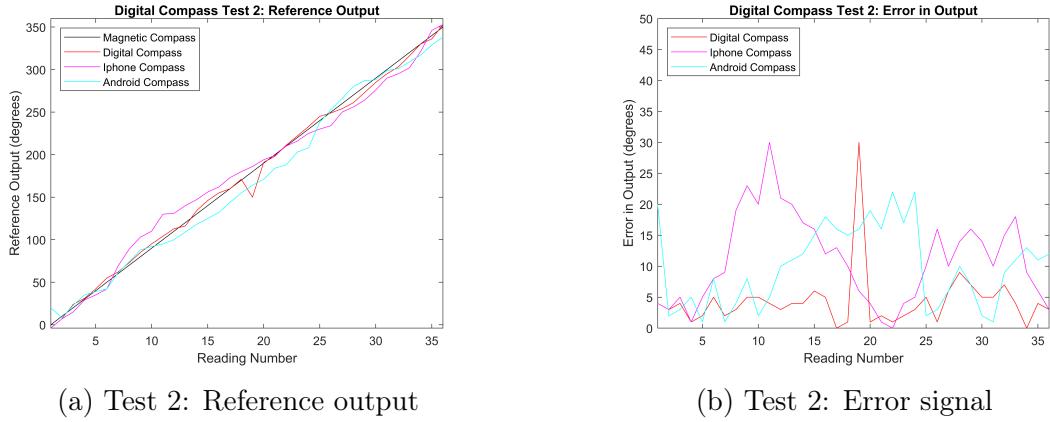


Figure 8.1.3: Digital compass test 2

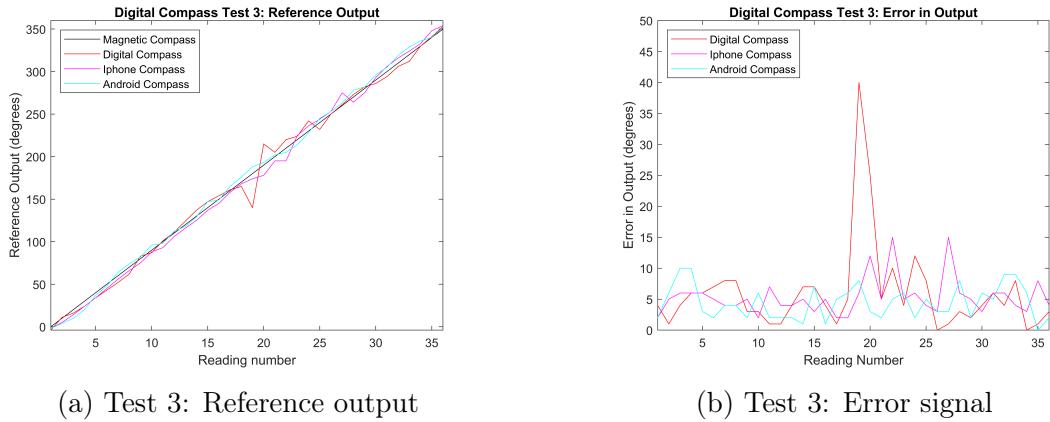


Figure 8.1.4: Digital compass test 3

8.1.3 Discussion of Results

Test 1

Test 1 showed that the iPhone, Android, and digital compass exhibited average errors of 3.11° , 2.19° , and 11.87° respectively. These results are the result of insufficient calibration. When not conducting an intensive calibration effort, the extreme minimums and maximums of each axis is not recorded, and during testing — values that exceed the incorrectly recorded minima and maxima — are amplified disproportionately resulting in extreme inaccuracies at certain degree ranges. The iPhone and Android compasses thus provided a fairly accurate estimate, but the digital compass, although providing a good estimation of general direction, did not provide an accurate heading.

Test 2

Test 2 showed that the iPhone, Android, and digital compass exhibited average errors of 10.73° , 9.73° , and 4.24° respectively. After conducting an intensive calibration test, the digital compass displayed an average error of only 4.24° which is an accurate estimation of heading. The spike in error signal between reading number 17 and 20 can be attributed to the filtering process when handling accelerometer data. Since the compass measured from -180° to 180° , the filtering process when crossing the magnetic South point, averages values that are both negative and positive in sign, thereby causing a rapid change in measured heading. The spike cannot be eliminated, only mitigated by reducing the effectiveness of the filtering process, since it will then be less reliant on previous values and more reliant on current values, but this choice will negatively affect noise presence.

Test 3

Test 3 showed that the iPhone, Android, and digital compass exhibited average errors of 5.22° , 4.33° , and 5.84° respectively. iPhone, Android, and digital compass readings during test 3 were accurate indication of heading compared to the magnetic heading. Once again, a spike in error was recorded during the transition from 180° to -179° across the South point.

8.2 GPS Unit

The GPS unit represents an integral part of the system's setpoint calculation ability for the controller of the rudder, furthermore, enabling the user to visualize the movement of the vessel post-voyage for data analysis and tuning purposes.

To test the accuracy of the GPS, a 2.5km round trip was conducted at a pace of 10km/h, a velocity greater than what the DragonFlite95 will reach, thus ensuring that the sample rate is sufficient. The GPS unit is set up to sample data at a rate of 1 Hz and write its latitude and longitude to a MicroSD card in csv format. Post testing, the data is downloaded and imported into MATLAB then plotted using the 'geoplot' function to show the route perceived.

The round trip is conducted around the Van Riebeeckshof Road circle in Bellville as shown below in Figure 8.2.1.

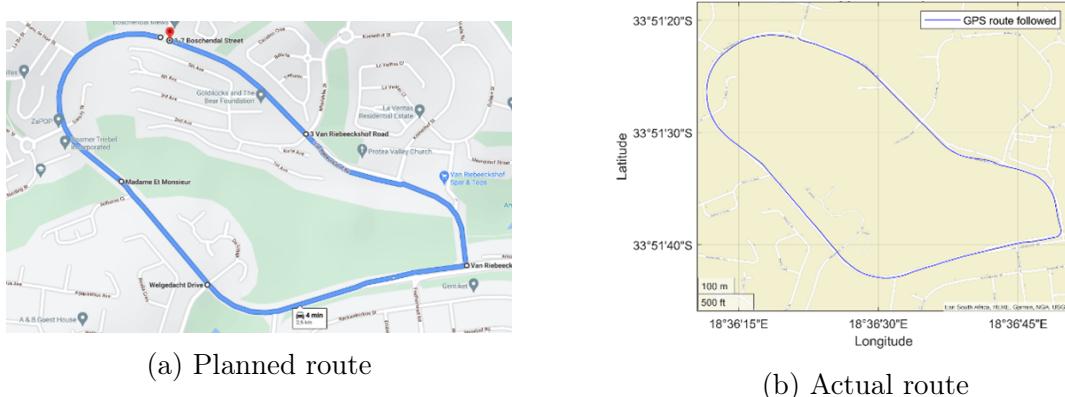


Figure 8.2.1: GPS accuracy test

The GPS data logged is represented above in Figure 8.2.1b. When comparing the GPS route logged to the planned route above in Figure 8.2.1a, it can be seen that the GPS logged data accurately resembles the true movement of the user.

It must be noted that the GPS system takes a few seconds to establish an initial link to satellites signalled by a constant red LED changing to flashing mode, but that during bad weather, characterized by visible cloud coverage and rain, the system takes almost a minute to establish a connection and sometimes loses signal during operation. When a signal is lost, the GPS feeds the microcontroller the last received position so that operation can continue but presents an error message to notify the user of the lost connection.

8.3 Windvane

The windvane's ability to accurately indicate the direction of apparent wind is a function of the digital compass' accuracy, and the ability of the windvane to turn its nose into the wind at low wind velocities. The digital compass' accuracy is determined, documented, and discussed in Chapter 8.1. The windvane's tail was experimentally constructed to allow correct movements at low wind velocities.

A single factor, however, was not considered during the design of the windvane – the distance between the windvane and the microcontroller. Communication between the compass and the microcontroller utilizes I2C master–slave synchronous serial communication. I2C is a short distance protocol intended for communication between devices on a single PCB. The microcontroller, acting as the master, generates clock pulses via the SCL line to enable synchronization, and the compass,

acting as the slave, transmits data via the SDA line, which is read by the master. Ideally, over a short distance, with little resistance and capacitance, the SDA line would appear as depicted below in Figure 8.3.1 displaying instantaneous changes from logical low (0V) to logical high(3.3V).

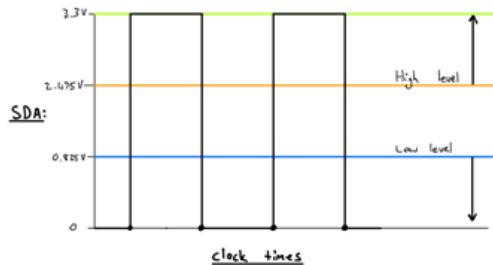


Figure 8.3.1: Ideal I2C communication

Due to conductors being non-ideal, a waveband of voltages can be accepted as a logical high or logical low. Any voltage greater than 2.475V is interpreted as a logical high and any voltage less than 0.825V is interpreted as a logical low. In the non-ideal environment, made even less ideal using long wires (1.8m) from the windvane to the microcontroller, large magnitudes of capacitance are present resulting in a large logical high rise time. This results in the SDA line looking like Figure 8.3.2 below. The rise time to logical high is not fast enough and even though a change from logical low to logical high is induced by the slave, it is not being interpreted as a logical high by the master.

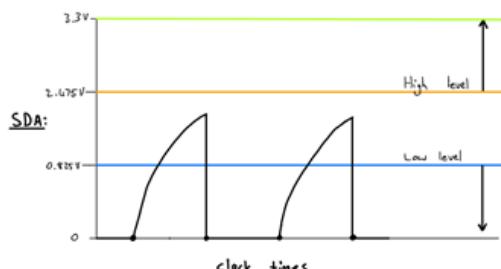


Figure 8.3.2: Real I2C communication without mitigation attempts

To mitigate the effects of the excessive capacitance, the root cause of the issue must be addressed. The source of the capacitance can be attributed to the characteristics of the 4 wires, namely VCC, GND, SDA, and SCL, and their orientation

with respect to each other. I2C tolerates a maximum capacitance of 400 pF in standard mode and the maximum rise time is a function of the serial communication rate.

Firstly, the rate of serial communication was lowered from 400 kbit/s to the minimum allowed rate of 5 kbit/s as mandated by the ESP32. This did not make a significant enough difference as the communication output was still perceived as zeros.

Next, two pull up resistors were added to the SCL and SDA lines ($2.2k\Omega$ each), to enhance the ability to overcome the capacitance and enable communication by pulling the line high if the slave does not pull it to ground. This change established successful communication between the master and the slave as the logical high is reached before being pulled low again as depicted below in the logic diagram in Figure 8.3.3 and the magnetometer reading diagram in Figure 8.3.4.

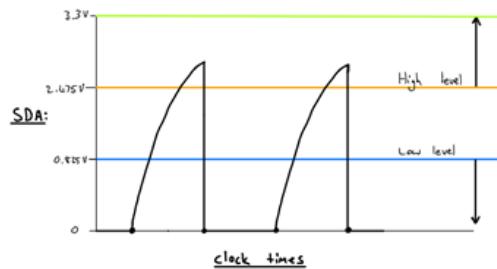


Figure 8.3.3: Real I2C communication after mitigation attempts

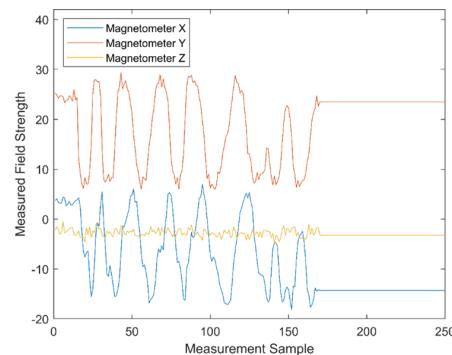


Figure 8.3.4: Resulting I2C communication test

However, consistently after about 150 samples the windvane compass produces constant readings as depicted above in Figure 8.3.4. This is the result of an inadequate rise time most likely caused by a rise in individual wire resistance as wires heat up or the effect of crosstalk between wires, or a possible momentary disconnect in the cable. Crosstalk can be defined as the unintended effects one cable has on another cable's communication transmission. To possibly mitigate the effect of increased resistance and crosstalk, a higher quality cable is used. The 4 loose insulated copper wires are replaced with an individually and overall shielded twisted pair cable that also contains thicker copper than the previous wire. The new wire setup is depicted below in Figure 8.3.5.

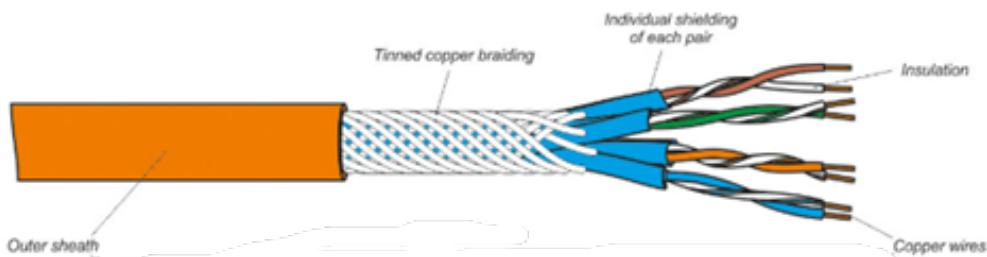


Figure 8.3.5: Shielded twisted pairs INSERT REF

The VCC and GND lines are doubled and paired with themselves, and the SCL and SDA lines are twisted with VCC lines to ensure that possible crosstalk is minimized. This mitigation however did not cease the occurrence of the communication stop.

A software attempt was also made to mitigate the issue by resetting the I2C communication protocol once 2 samples of the same X, Y, and Z field strengths are received, since this is next to impossible. Communication would then be reinitialized and could continue, but this caused the microcontroller to reinitialize communication once every 10 seconds which was not feasible.

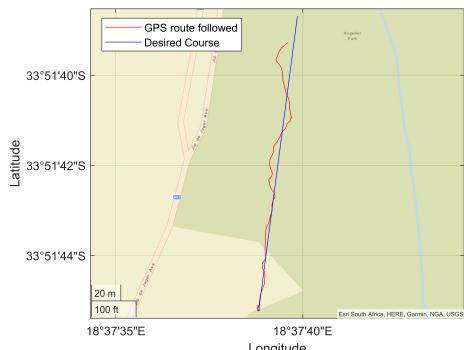
At the time when this was realized, the option for changing the design to use another form of communication or add an additional microcontroller closer to the slave device was not feasible due to the little time that was left and spatial constraints on the model sailing vessel. (The additional microcontroller would then communicate with the main microcontroller via Bluetooth). It is also not possible to move the windvane to a different location, since at any location other than the top of the mast, the apparent wind senses would be influenced by the position of

the sails.

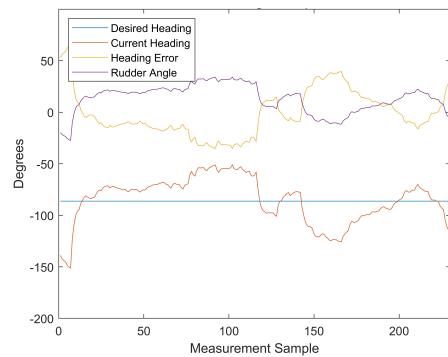
It was therefore decided, that to verify the functionality of the rest of the system, a wind direction would be read at the test location using the windvane and then assumed to be constant throughout the test. Future work would include further attempting to mitigate the issue or implementing another microcontroller.

8.4 Control System

Before testing the ability of the vessel to sail a course in an aquatic environment, the ability of the controller to keep a course on dry land will be tested. This will be tested by programming a physical landmark as destination into the system which is visible from the testing location in a park. The control system will be programmed to respond with a large K_p value so that visible rudder movement is present, and the test user can turn the helm of the vessel as mimicking the response in water. The response exhibited by the control system is visible to the test user and all control system data is logged to a MicroSD card for post-test data analysis. The visualization of the data is seen below in Figures 8.4.1.



(a) GPS park test data



(b) Control system data

Figure 8.4.1: Results of park test 1

As seen above in Figure 8.4.1a, during park land test 1, the vessel attempted to follow the straight-line course illustrated by the blue line. As seen above in Figure 8.4.1b, the heading error illustrated in yellow, is constantly countered by the rudder angle illustrated in purple, thereby keeping the current heading illustrated in orange centred about the desired heading illustrated in blue.

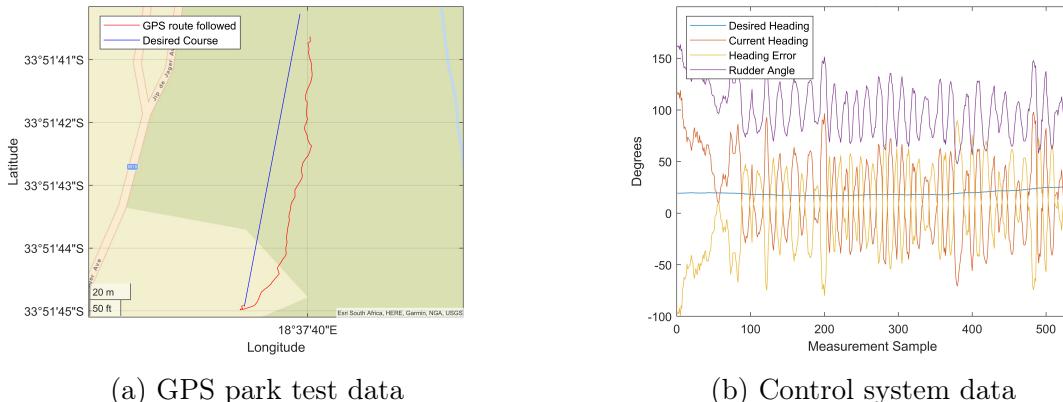


Figure 8.4.2: Results of park test 2

During the second park test as seen above in Figures 8.4.2, a deliberate deviation from the straight-line desired course is made to exhibit the dynamically calculated desired heading which could compensate for a current drift.

After the initial deviation from the course (at the bottom of Figure 8.4.2a), there is little change in desired heading, but as the vessel approaches the destination (at the top of Figure 8.4.2a)), there is a larger change in desired heading. As seen above in figure 8.4.2b), the heading error illustrated in yellow, is constantly countered by the rudder angle illustrated in purple, thereby keeping the current heading illustrated in orange centred about the desired heading illustrated in blue. Although there is significant overshoot present during land test 2, it demonstrates the control system's ability to keep to the desired heading on average.

8.5 Integration of Entire System

After testing all the individual sub-systems and successfully completing the land-based verification tests, the entire vessel with the embedded sub-systems will be setup to sail in the aquatic environment.

The Bellville Golf Course dam was selected a testing environment due to its low-gradient slope at the edges allowing a safe launch of the vessel by the user. The environment is also known for its mostly windy conditions which is ideal for testing. A picture of the DragonFlite95 vessel during a test can be seen below in Figure 8.5.1.



Figure 8.5.1: Dragonflite95 during lake tests

8.5.1 Lake Test 1

A wind originating from 170° (SSE) is recorded during the test setup as indicated by the arrow on the GPS data log below. A GPS destination is set so that points of sail are not in the dead zone, and thus no tacking is required. The proportional control parameter K_p is set as 4, the same value as used during park tests. The results of the first lake test can be seen below in Figure 8.5.2.

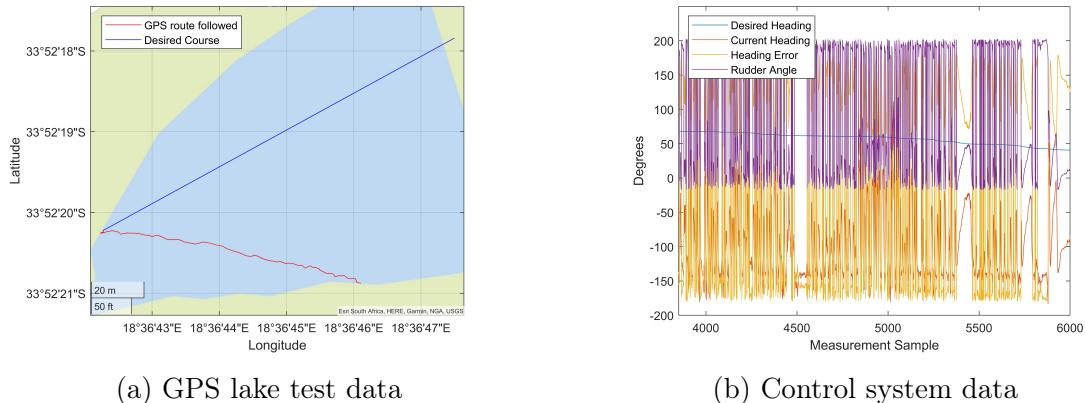


Figure 8.5.2: Results of lake test 1

The GPS data logged (as illustrated in red) during the first lake test can be seen in Figure 8.5.2a. This test illustrates that the vessel did not remotely keep

to the straight-line desired course. The desired heading as shown in blue in Figure 8.5.2b, however shows that that the control system still aimed to point the helm of the vessel to the destination. The inability to reach the destination was caused by extreme overshoot and oscillation of the controller as seen in the rudder angle (illustrated in purple) in Figure 8.5.2b. During the next test, the value of K_p must be reduced significantly to avoid extreme overshoot and oscillation.

The maximum velocity of the sailing vessel recorded during this test was 0.94 m/s and the average velocity recorded during this test was 0.05 m/s over 80 meters. The low average velocity recorded was due to constant turning of the vessel from side to side caused by the oscillation seen in figure 8.5.2b.

This test, even though not successful in reaching the desired destination, illustrated the ability of the vessel to sail at a point of sail of 40° , showing that the assumed dead zone of 45° to either side of the apparent wind is a safe assumption. Future work would include determining the dead zone limits to optimally propel the vessel.

8.5.2 Lake Test 2

A wind originating from 180° (S) is recorded during the test setup as indicated by the arrow on the GPS data log below. The GPS destination is kept the same to allow sailing in broad reach without the need for tacking. The value of K_p is decreased to 2. The results of the second lake test can be seen below in Figure 8.5.3.

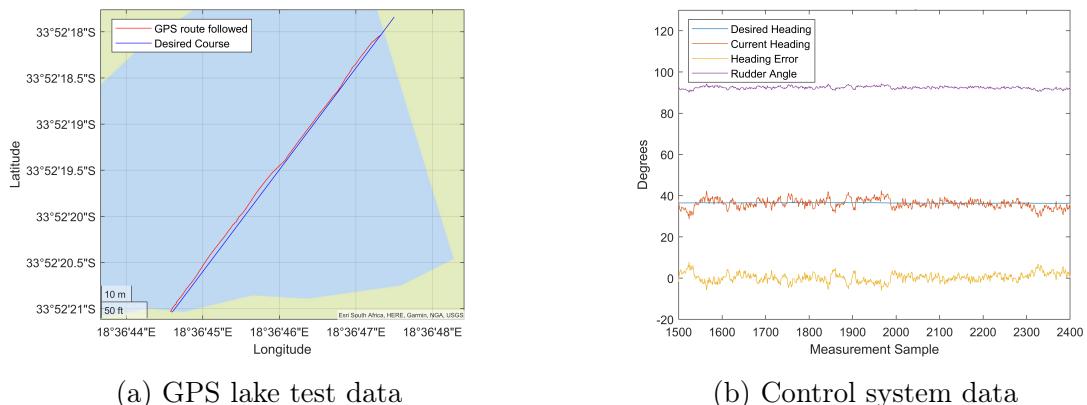


Figure 8.5.3: Results of lake test 2

The GPS data logged (as illustrated in red) during the second lake test can be seen in Figure 8.5.3a. This test illustrates that the vessel near-perfectly kept the straight-line desired course thereby proving the system's ability to sail at a constant heading, and sail to a specific destination.

The desired heading (as illustrated in blue in Figure 8.5.3b) is kept nearly constant as there is not much deviation of the GPS route followed (as illustrated in red in Figure 8.5.3a) from the desired straight-line course (as illustrated in blue in Figure 8.5.3a).

The current heading (as illustrated in orange) closely tracks the desired heading (as illustrated in blue) in Figure 8.5.3b. Although there is still overshoot and oscillation, the magnitude of overshoot is comparably much smaller than during the previous test. The heading error oscillates around 0° and has a maximum of absolute error of 6° during the test.

The maximum velocity of the sailing vessel recorded during this test was 1.4 m/s and the average velocity recorded during this test was 0.4 m/s over 120 meters. The maximum and average velocity is greatly increased from the previous test due to the reduce magnitude of overshoot.

During the next test, the value of K_p will be kept constant to see whether a similar result is achieved and if the controller responds predictably.

8.5.3 Lake Test 3

A wind originating from 190° (SbW) is recorded during the test setup as indicated by the arrow on the GPS data log below. The GPS destination is kept the same to allow sailing in a training run without the need for tacking. The value of K_p is left as 2. The results of the third lake test can be seen below in Figures 8.5.4.

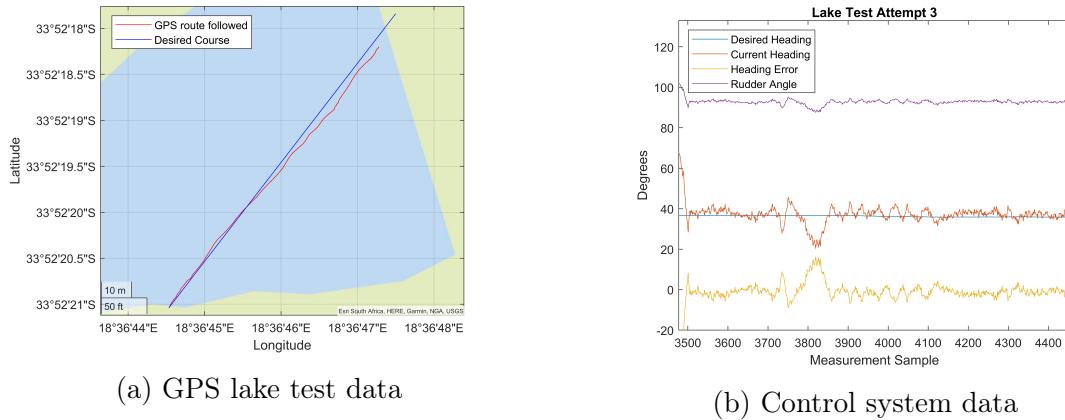


Figure 8.5.4: Results of lake test 3

The GPS data logged (as illustrated in red) during the third lake test can be seen in Figure 8.5.4. This test again illustrates that the vessel near-perfectly kept the straight-line desired course thereby proving the system's ability to sail at a constant heading, and sail to a specific destination, as well as showing that the control system outcome is predictable and repeatable.

The desired heading (as illustrated in blue in Figure 8.5.4b) is kept nearly constant as there is not much deviation of the GPS route followed (as illustrated in red in Figure 8.5.4a) from the desired straight-line course (as illustrated in blue in Figure 8.5.4a).

The current heading (as illustrated in orange) closely tracks the desired heading (as illustrated in blue) in Figure 8.5.4b, but there are some deviations from the desired heading causing the occasional spike in heading error.

The initial large error is due to explicitly placing the vessel in water facing a different direction than the desired heading to test the ability of the stationary vessel to start moving in the correct direction. The heading error is immediately countered by the rudder. The second set of large errors could possibly be caused by a wind gust in a different direction than the initial measured wind direction. The heading error is countered and mitigated without much overshoot after the second rudder change from one side to the other.

The maximum velocity of the sailing vessel recorded during this test was 1.3 m/s and the average velocity recorded during this test was 0.25 m/s over 120 meters. The maximum and average velocity is comparable to that of the previous test.

Chapter 9

Evaluation

9.1 Evaluation of Sub-System

9.1.1 Digital Compass

The tests conducted to verify the working of the digital compass showed promising results. The average error obtained by measuring direction from the digital compass compared to a magnetic compass was 5.04° . The design requirement of achieving an average error of less than 10° is achieved, but it needs to be mentioned that multiple tests indicated that there is no repeatability in low-cost digital measurements and cumulative error may be 'leaked' due to lack of repeatability. The effect of a slight error in heading is mitigated by use of GPS data to calculate required heading dynamically. The designed digital compass provides a fairly accurate estimation of the heading.

Future work would include to improve the accuracy of the digital compass, mitigate the spike in error, and design a complimentary function that utilizes digital compass heading and GPS determined heading while moving.

9.1.2 GPS Unit

The GPS unit performed as expected and provided good positional accuracy. The accuracy of the GPS however is not crucial to the working of the vessel, since over a few kilometres - a positional error of 10 meters will barely have any significance. The only issue to be mentioned is the poor signal locking ability of the unit when cloud coverage is abundant.

9.1.3 Datalogger System

The data logger performed excellent and enabled the user to review, visualize and tune the performance of the vessel post-voyage without any fault.

9.1.4 Windvane

The designed windvane did not function as intended. A fatal design flaw was the sheer distance between the sensor and the microcontroller. The result was excessive bus capacitance which caused the communication to be unusable. After attempting to mitigate the effects of added capacitance, successful communication was established, but malfunctions were still frequent. Subsequently, a constant wind direction had to be assumed during testing.

9.2 Evaluation of Control System

9.2.1 Theoretical Working

During land tests, the control system displayed success in reacting to rectify the heading error by actuating changes in the rudder to the opposite direction. Large magnitudes of overshoot were present causing oscillation. The system also displayed success in the ability to dynamically calculate new required headings due to a deviation off course that may occur due to current drift in the ocean. The controller also displayed the ability to keep to a desired heading on average, thus establishing successful 'heading hold' on average albeit oscillation.

Due to leakage of error of the digital compass, it would not be possible to implement integral control in a discrete time-system using a cumulative error estimation. Therefore, it would never be possible to eliminate overshoot and oscillation. A reduced K_p however lead to extremely successful tests by minimizing the magnitude of overshoot.

9.2.2 Performance of Vessel in Water

Water tests over distances between 80 meters and 120 meters, confirmed the working of the control system. It was seen that a controller with significant overshoot and oscillation completely inhibited the vessel from achieving waypoint following. Once the controller was better tuned, the system displayed excellent results. It closely

followed the desired straight-line course - much better than the result achieved by Stenersen during testing while also achieving a higher average and maximum velocity even though the focus of this study was not optimal speed achievement.

9.3 Recommended Improvements and Future Work

Future work would include to improve the accuracy of the digital compass, mitigate the spike in error, and design a complimentary function that utilizes digital compass heading and GPS determined heading while moving.

The GPS unit could be replaced with a more expensive sensor to improve performance in cloudy or rainy conditions. A longer antenna could also be added which can be placed on top of the mast instead of on the deck.

The windvane design may be kept as is, but the mode of communication must be changed to a more suitable mode for communication over longer distances such as RS-232. Another option would be to install an I2C serial bus booster or add a secondary ESP32 microcontroller just below the sliring that could communicate with the main microcontroller on the deck via Bluetooth.

Due to weather and time constraints, more water tests or tests over longer distances could not be conducted to verify the ability of the control system to perform tacks. A test needs to be set up where the required destination is upwind from the current position in the dead zone, thus creating an infeasible course, forcing the need for tacking manoeuvres.

The performance of the system could also be recorded and documented when using a P, PI, and PID controller to increase the performance of the control system by reducing overshoot and oscillation.

Ultimately, the vessel would also need to be made self-sufficient, so an investigation into renewable alternatives to battery power could also be conducted to allow a vessel to stay at sea for extreme durations.

Chapter 10

Conclusion

In recent years, the STEM industry has seen rapid development of unmanned vehicles in air- and road environment, but hardly any commercialized development has been made for unmanned vessels in the oceanic environment. The proposed project will serve as a proof of concept for an unmanned sailing vessel by designing and developing a fixed wing, fitting appropriate sensors, and implementing a control system to an RC model sailing vessel.

This project has the potential to serve as a benchmark for the future of environmentally friendly oceanic research with low operating costs and little human input.

The project is expected to be completed within sixteen weeks at a total labour cost of R150 000 and a total capital expenditure of R8500. Ample risk mitigation strategies are implemented to prevent budget overruns.

Bibliography

- Amorina Kingdon (2019). The South Pointing Spoon. *Hakai*. Available at: <https://www.hakaimagazine.com/article-short/the-south-pointing-spoon/>
- Bars, F.L. and Jaulin, L. (2015). *Robotic Sailing 2014*. ISBN 9783319022758.
- Caruso, M.J. (2000). Applications of magnetic sensors for low cost compass systems. *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 177–184.
- Ferl, T. and Hills, S. (2020). A Control System for a Small Autonomous Sailing Vessel.
- Ferreira, H., Almeida, C., Martins, A., Almeida, J., Dias, N., Dias, A. and Silva, E. (2009). Autonomous bathymetry for risk assessment with ROAZ robotic surface vehicle. *OCEANS '09 IEEE Bremen: Balancing Technology with Future Needs*, pp. 1–6.
- Good Hope Sailing Academy (2020). *Competent Crew Course*.
- Jin, J., Zhang, J., Shao, F., Lyu, Z. and Wang, D. (2018). A novel ocean bathymetry technology based on an unmanned surface vehicle. *Acta Oceanologica Sinica*, vol. 37, no. 9, pp. 99–106. ISSN 0253505X.
- Lavalle, S.M. (2006). Planning Algorithms. Available at: <http://planning.cs.uiuc.edu/node/102.html>
- Legursky, K., Downing, D., Ewing, M. and Young, B. (2013). System Identification and the Modeling of Sailing Yachts.
- Manley, J.E. (1997). Development of the autonomous surface craft ‘ACES’. *Oceans Conference Record (IEEE)*, vol. 2, no. November, pp. 827–832. ISSN 01977385.
- Manley, J.E. (2008). Unmanned surface vehicles, 15 years of development. *Oceans 2008*, , no. October 2008.
- Manley, J.E., Marsh, A., Cornforth, W. and Wiseman, C. (2000). Evolution of the autonomous surface craft AutoCat. *Oceans Conference Record (IEEE)*, vol. 1, no. November, pp. 403–408. ISSN 01977385.

- NOAA (2021). How much of the ocean have we explored?
Available at: <https://oceanservice.noaa.gov/facts/exploration.html>
- Plumet, F., Pêtrès, C., Romero-Ramirez, M.A., Gas, B. and Ieng, S.H. (2015). Toward an Autonomous Sailing Boat. *IEEE Journal of Oceanic Engineering*, vol. 40, no. 2, pp. 397–407. ISSN 03649059.
- Rodriguez-ortiz, C.D. (1996). Automated Bathymetry Mapping Using an Autonomous Surface Craft. , no. June 1996.
- Safe-skipper (2018). Points of Sailing.
Available at: <https://www.safe-skipper.com/points-of-sail/>
- Saildrone (2020). Saildrone - About Us.
Available at: <https://www.saildrone.com/about>
- Santos, D., Junior, A.G., Negreiros, A., Boas, J.V., Alvarez, J., Araujo, A., Aroca, R.V. and Gonçalves, L.M. (2016). Design and implementation of a control system for a sailboat robot. *Robotics*, vol. 5, no. 1, pp. 1–24. ISSN 22186581.
- Stenersen, H.S. (2016). Construction and Control of an Autonomous Sail Boat. *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 524–531. ISSN 24058963.
Available at: <http://dx.doi.org/10.1016/j.ifacol.2016.10.489>
- Strömbeck, C. (2017). Modeling, Control and Optimal Trajectory Determination for an Autonomous Sailboat. *Lund University Libraries*.
Available at: <https://lup.lub.lu.se/student-papers/search/publication/8923407>
- Talat, O. (2015). Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors. *Freescale Semiconductor Application Note*, vol. AN4248.
Available at: <https://www.nxp.com/docs/en/application-note/AN4248.pdf>
- Tretow, C. (2017). Design of a free-rotating wing sail for an autonomous sailboat. pp. 10–16.
- Vazquez-cuervo, J., Gentemann, C., Tang, W., Carroll, D., Zhang, H., Menemenlis, D., Gomez-valdes, J., Bouali, M. and Steele, M. (2021). Using Saildrones to Validate Arctic Sea-Surface Salinity from the SMAP Satellite and from Ocean Models. pp. 1–16.
- Yacowicz, W. (2017). The Most Exciting Drones Aren't in the Air- They're in the Ocean. Inc.
- Zakki, A.F., Triwiyatno, A., Sasmito, B. and Budiono, K. (2019). Design and control of autonomous surface vehicle to support bathymetry survey in the coastal environment.