Design and Development of a Marine Data Acquisition System for Inertial Measurement in Wind Powered Yachts

By

Alexandre Bergeron

Thesis submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Applied Science (Engineering)

University of Ottawa

Ottawa, Ontario, Canada

Abstract

This thesis presents the design of an inertial measurement data acquisition system intended for use in sailboats. The variables of interest are 3-axis acceleration, 3-axis rotation, GPS position/velocity, magnetic compass bearing and wind speed/direction. The design focus is on low-cost micro-electromechanical systems (MEMS) based technology and demonstrating the validity of these technologies in a scientific application. A prototype is constructed and submitted to a series of tests to demonstrate functionality and soundness of the design. These tests range from bench tests to full scale application. Contributions of this thesis include the novel application of inertial measurement unit (IMU) technology to a sailboat racing application, the integration of all instrumentation, creative ruggedised packaging and emphasising the use of low-cost commercial off-the-shelf (COTS) technology.

Acknowledgements

Successfully completing the work and documentation required for this thesis would have proven impossible without the aid, support and encouragement of a large number of individuals. Firstly, a large portion of this recognition goes to the thesis supervisor, Prof. Natalie Baddour, who was instrumental in guiding the direction and focus of the work while being astonishingly expedient when requests for review or consultations were thrown in her direction. Her creative insights have always managed to help me engineer a way out of any technical hurdle.

The rest of the recognition goes to my significant other, family and friends: all of whom were instrumental in upholding my motivation, inspiration and perseverance.

Table of Contents

Abstract **2**

Acknowledgements **3**

List of Tables **11**

List of Figures **13**

List of Acronyms **16**

Chapter 1 Introduction **1**

1.1. Problem Definition 1

1.2. Sailing Yachts 1

1.2.1. Brief Historical Context *2*

1.2.2. Design and Operating Methods *2*

1.2.3. Racing *3*

1.2.4. Summary of Sailing Yachts *4*

1.3. Problem Introduction 4

1.4. Thesis Contributions 5

1.5. Thesis Outline 5

Chapter 2 Hardware Review **7**

2.1. Data Acquisition Systems 7

2.1.1. Sensors *7*

2.1.2. Signal Processors *8*

2.1.3. Analog to Digital Converters *8*

2.2. Inertial Measurement Units 11

2.2.1. Typical Components *11*

2.2.2. Typical Applications *13*

2.2.3. Euler Angles *13*

2.2.4. Filtering *15*

2.2.5. Sensor Fusion *16*

2.3. Inertial Measurement Sensors 19

2.3.1. Accelerometers *19*

2.3.2. Gyroscopes *20*

2.3.3. Magnetic Compass *22*

2.3.4. Global Positioning System *23*

2.3.5. Sensor Performance Characterisation *27*

2.4. Data Acquisition Platforms 29

2.4.1. Personal Computer *29*

2.4.2. Microcontroller *30*

2.4.3. Standalone Data Loggers *33*

2.5. Commercial Inertial Measurement Products 34

2.5.1. Crossbow 320 Series IMU *34*

2.5.2. Xsens MTi 6 DOF Measurement Unit *35*

2.6. Existing Products for Sailing Yachts 37

2.6.1. Navigation Systems *37*

2.6.2. Tacktick *39*

2.6.3. Velocitek SpeedPuck *41*

2.6.4. Limitations of current systems *42*

2.7. Wind 42

2.7.1. Wind Direction *43*

2.7.2. Wind Speed *43*

Chapter 3 Research Objectives 45

3.1. General Objectives 45

3.2. Specific Objectives 46
3.3. Intended Application 47
3.4. Design Requirements 47
3.4.1. Variables of Interest 47
Chapter 4 Concept Generation and Selection 49
4.1. Design Specifications 49
4.1.1. Design Specification Summary 51
4.2. Concept Generation 52
4.2.1. Independent Sensor Network 52
4.2.2. Microcontroller Based Centralised Sensor Network 53
4.2.3. Microcontroller Based Wireless Sensor Network 53
4.3. Concept Selection 54
4.4. Core Hardware Selection 57
4.4.1. LEGO Mindstorms NXT 57
4.4.2. Axon II 58
4.4.3. Arduino Mega 2560 59
4.4.4. Microcontroller Comparison 60
4.5. Summary 61
Chapter 5 Hardware Selection and Integration 63
5.1. System Architecture 63
5.2. Hardware Components 65
5.2.1. Microcontroller 65
5.2.2. Six Degrees of Freedom Razor – Ultra-Thin IMU 65
5.2.3. Digital Compass 68
5.2.4. Global Positioning System 70

5.2.5. Wind Instruments *72*

5.2.6. Secure Digital Storage Media Interface *74*

5.3. Hardware Integration 74

5.3.1. Physical Connections *75*

5.3.2. Microcontroller to Device Interfacing *77*

5.3.3. PC to AxonII Interfacing *78*

5.4. PC Interface Software 79

5.4.1. Axon II Programming *81*

5.4.2. Matlab Interface *82*

5.5. Standalone Interface Software 84

5.5.1. Axon II Programming *84*

5.5.2. PC Data Interface *86*

5.6. AxonII Code Optimisation 88

5.6.1. PC Interface *88*

5.6.2. Standalone Interface *89*

5.7. Packaging 90

5.8. Batteries 92

Chapter 6 Validation 95

6.1. Bench Testing 95

6.1.1. Justification *95*

6.1.2. Sensor Orientation Verification *95*

6.1.3. GPS Reception and Position *98*

6.1.4. Compass Heading *99*

6.1.5. Battery Life *100*

6.1.6. Wind Speed *100*

6.1.7. Wind Direction *103*
6.1.8. Data Rate *104*
6.1.9. Bench Testing Summary *104*
6.2. In-Car Testing 105
6.2.1. Justification *105*
6.2.2. Challenges *106*
6.2.3. Test Circuit and Procedure *106*
6.2.4. GPS Position *111*
6.2.5. GPS Speed *113*
6.2.6. Accelerometer *114*
6.2.7. Gyroscope *121*
6.2.8. In-Car Testing Summary *125*
6.3. Bicycle Testing 125
6.3.1. Justification *125*
6.3.2. Challenges *125*
6.3.3. Test Circuit and Procedure *126*
6.3.4. Roll Angle Estimation *128*
6.3.5. Roll Angle Estimation Results *131*
6.3.6. Bicycle Testing Summary *138*
6.4. Sailboat Testing 138
6.4.1. Justification *139*
6.4.2. Challenges *139*
6.4.3. J/24 ..... *140*
6.4.4. Installation *142*
6.4.5. Operation *146*

6.4.6. Environmental Factors *147*

6.4.7. Sailboat Testing Summary *149*

6.5. Summary 149

Chapter 7 Summary & Conclusion 150

7.1. Thesis Summary 150

7.2. Prototype Data Acquisition System Final Specifications 151

7.3. Contributions 152

7.4. Future Work and Suggested Improvements 152

7.4.1. Hardware *152*

7.4.2. Data Gathering *153*

7.4.3. Data Analysis *154*

7.5. Conclusion 155

References 157

Appendix A - Device Specifications 169

A.1. Society of Robots AxonII Specifications 169

A.2. Analog Devices® ADXL335 Specifications 172

A.3. STMicroelectronics™ LY530ALH Specifications 174

A.4. STMicroelectronics™ LPR530AL Specifications 177

A.5. Honeywell HMC6352 Specifications 180

A.6. Venus 634LPx Specifications 181

A.7. ONSHINE GPS Antenna ANT-555 Specifications 183

A.8. Specifications of the Argent Data Systems Weather Sensor Assembly 185

Appendix B - Programming and Scripts 187

B.1. PC Interface 187

B.1.1. AxonII Program *187*

B.1.2. Matlab Script *189*

B.2. Standalone Interface 192

B.2.1. AxonII Program *192*

B.3. Post Processing Matlab Scripts 196

B.3.1. SD Card Raw Data Importation *196*

B.3.2. SD Card Conversion *197*

B.3.3. Wind Speed Conversion *200*

Appendix C - Hardware and Wiring 202

C.1. Component List 202

C.2. Wiring Diagram 204

C.3. Connectors 205

C.4. Prototype Cost 206

Appendix D - AxonII Related Software & Drivers 207

D.1. Virtual COM port drivers 207

D.2. AVR Studio 207

D.3. Webbotlib 207

D.4. AxonII Bootloader & Procedures 209

Appendix E - Sailing 210

E.1. Sailboat Anatomy 210

E.2. Nautical Terminology 211

List of Tables

Table 1: Accelerometer Displacement Measurement Techniques 20
Table 2: GPS Capabilities 26
Table 3: Lassen iQ GPS Module Accuracy [43] 26
Table 4: Selected Arduino Microcontroller Comparison 32
Table 5: Crossbow IMU 320 Summary Information [58] 35
Table 6: Xsens MTi Summary Information [60] 36
Table 7: Variables of Interest 48
Table 8: Design Specifications 49
Table 9: Design Specification Summary 51
Table 10: Concept Selection Matrix 56
Table 11: Microcontroller Comparison 61
Table 12: ADXL335 Condensed Specifications 66
Table 13: Gyroscope Condensed Specifications 68
Table 14: Honeywell HMC6352 Condensed Specifications 69
Table 15: Venus 634LPx Condensed Specifications 71
Table 16: Weather Vane Direction to Resistance Equivalence 73
Table 17: Interpreted Raw Axon II Data String 80
Table 18: AxonII Program Instructions for PC Interface 81
Table 19: Matlab Script Instructions for PC Interface 82
Table 20: AxonII Program Instructions for Standalone Interface 84
Table 21: Wind Heading Equivalencies 87
Table 22: Expected Device Power Draws 93
Table 23: Average Measured Accelerations during Static Testing 97
Table 24: Average Measured Rotational Rates during Static Testing 98
Table 25: Battery Life Test Results 100
Table 26: Wind Speed Test Results 102
Table 27: In-Car Test Circuit Description 110
Table 28: In-Car Testing X-Axis Accelerometer Summary 117
Table 29: Event 10 Y-Axis Acceleration Analysis Results 120
Table 30: In-Car Testing Gyroscope Summary 122

Table 31: Z-Axis Gyroscope Event 10 Analysis Results 124
Table 32: Bicycle Test Circuit Information 127
Table 33: Estimated Bicycle Roll Angle for Turn 1, While Riding 131
Table 34: Estimated Bicycle Roll Angle for Turn 2, While Riding 132
Table 35: Estimated Bicycle Roll Angle for Turn 3, While Riding 134
Table 36: Estimated Bicycle Roll Angle While Riding in a Straight Line 135
Table 37: Estimated Bicycle Roll Angle for Turn 1, While Pushed 137
Table 38: Estimated Bicycle Roll Angle for Turn 3, While Pushed 137
Table 39: Summary of Estimated Bicycle Roll Angles 138
Table 40: J/24 Dimensions [102] 142
Table 41: Prototype Data Acquisition System Specifications 151
Table 42: AxonII Specifications [83] 169
Table 43: ATmega 640 Specifications [104] 170
Table 44: Parts List 203
Table 45: Data Acquisition Prototype Component Cost 206

List of Figures

Figure 1: Offset Error [7] 9
Figure 2: Full-Scale Error [7] 9
Figure 3: ADC Quantisation Error [7] 10
Figure 4: Typical Body Frame Axes for Ground Vehicle 12
Figure 5: Euler Angles [24] 14
Figure 6: Basic Complementary Filter [29] 18
Figure 7: Structure of an accelerometer [4] 19
Figure 8: Vibratory Gyroscope Operating Principle [32] 21
Figure 9: Simplified Trilateration Example [38] 24
Figure 10: Simplified Trilateration Example Showing Signal Time Errors [38] 25
Figure 11: Atmel ATxmega16A4U Microcontroller [49] 31
Figure 12: AT90CAN128 AVR Microcontroller and Header Board [50] 31
Figure 13: HOBO U12 4-Channel External Data Logger [55] 33
Figure 14: Crossbow IMU320 [56] 34
Figure 15: Xsens MTi 6 Degree of Freedom [60] 36
Figure 16: Garmin eTrex® 30 [62] 38
Figure 17: Lowrance HDS-10® Fishfinder/GPS Chartplotter [63] 38
Figure 18: TackTick T210 Multifunctional Wireless Maxi Display [64] 40
Figure 19: Velocitek SpeedPuck [65] 41
Figure 20: Independent Sensor Network 52
Figure 21: Microcontroller Based Centralised Sensor Network 53
Figure 22: Microcontroller Based Wireless Sensor Network 54
Figure 23: Proposed System Architecture 64
Figure 24: 6 DOF Razor IMU [86] 66
Figure 25: Honeywell HMC6352 [88] 69
Figure 26: Venus 634LPx GPS [89] 70
Figure 27: ONSHINE GPS Antenna ANT-555 [90] 71
Figure 28: Argent Data Systems Weather Sensor Assembly [91] 72
Figure 29: Breakout Board for SD-MMC Cards [93] 74
Figure 30: Data Acquisition Prototype Wiring Diagram 76

Figure 31: Pelican® 1120 Case 91
Figure 32: Internal Packaging Layout 91
Figure 33: Deutsch DT Series Connectors [95] 92
Figure 34: Typical 7.4V 100mAh Lithium-Polymer Battery Pack [96] 93
Figure 35: Orthogonal Orientations 96
Figure 36: X-Axis Acceleration during Bench Testing 96
Figure 37: Y, Z-Axis Accelerations during Bench Testing 97
Figure 38: Rotational Rates during Bench Testing 98
Figure 39: Compass Heading during Bench Testing 99
Figure 40: Suunto A-10 Needle Compass [98] 100
Figure 41: Wind Speed Test Setup 101
Figure 42: WindMate WM-200 Portable Anemometer [99] 101
Figure 43: Wind Speed Test Results 102
Figure 44: Wind Heading, Vane Spinning 103
Figure 45: In-Car Testing Track 107
Figure 46: Test Circuit Event Locations 109
Figure 47: Event 10 Turn Radius Estimation 111
Figure 48: In-Car Testing GPS Track Plot 112
Figure 49: In-Car Testing GPS Plot Overlaid on Map 112
Figure 50: In-Car Testing GPS Speed 113
Figure 51: In-Car Testing X-Acceleration 114
Figure 52: In-Car Testing Y-Acceleration 115
Figure 53: X-Acceleration Test 3 Event 5 116
Figure 54: X-Axis Acceleration for Test 3 Event 1 118
Figure 55: Yaw Axis Rotational Rate 121
Figure 56: Z-Axis Rotational Rate for Test 3 Event 7 123
Figure 57: Bicycle Test Circuit 127
Figure 58: Heel Angle Estimation Diagram 129
Figure 59: Centripetal Acceleration [34] 130
Figure 60: Turn 1 Test 10 [100] 132
Figure 61: Test 10 Turn 2 [100] 133

Figure 62: Test 10 Turn 3 [100] 134
Figure 63: Test 10 Straight Line [100] 136
Figure 64: J/24 Dimensions [102] 141
Figure 65: Prototype Data Acquisition System J/24 Mounting Location 143
Figure 66: GPS Antenna J/24 Mounting Location 144
Figure 67: Wind Instruments J/24 Mounting Location 144
Figure 68: Wind Instruments J/24 Cable Management 145
Figure 69: Portable Computer J/24 Mounting Location 146
Figure 70: Hourly Wind Speed for September 15, 2011 Ottawa [103] 148
Figure 71: Molex WM2512-ND Pin [110] 205
Figure 72: Molex Connector Plastic Housings [111] [112] [113] 205
Figure 73: Webbotlib Project Designer [119] 208
Figure 74: AxonII Bootloader [118] 209
Figure 75: Anatomy of a Sailboat [120] 210

List of Acronyms

GPS: Global Positioning System

ADC: Analog to Digital Converter

IMU: Inertial Measurement Unit

MEMS: Micro-Electro Mechanical System

WAAS: Wide-Area Augmentation System

DGPS: Differential Global Positioning System

NMEA: National Marine Electronics Association

PCB: Printed Circuit Board

SD/MMC: Secure Digital/Multi-Media Card

SPI: Serial Parallel Interface

SMA: Sub-Miniature version A (Connector)

USB: Universal Serial Bus

Chapter 1 Introduction

Sailing and the pursuit of speed have long been historic truths for humanity. An old mariner's saying is "a single sailboat cruises, two sailboats race" and at no other time in history does this ring truer than it does today. Over the last century a virtual explosion of yacht clubs, sailboat racing series and competitions has appeared. This has resulted in numerous innovations which ceaselessly repel the technical boundaries and capabilities of sailing yachts: always further, always faster. This same climate also opens the possibility for further improvements.

As a basic requirement for any engineering design, it is important to properly define the loads, conditions and requirements in which a particular machine must operate. In the case of sailing machines however, this has always proved problematic: simply quantifying the complex motion of a vessel underway is not readily accomplished.

This chapter will serve to introduce several key concepts such as data acquisition systems and inertial measurement units as well as contextual information in order to establish the relevance of the work undertaken.

1.1. Problem Definition

The problem addressed by this thesis is the lack of a freely available tool or method which can be used to quantify and measure the movement of a sailing yacht while underway.

1.2. Sailing Yachts

Sailing yachts, commonly referred to as sailboats are water borne vessels which employ the wind as their primary method of propulsion. This is in the vast majority of cases done by employing a series of cloth aerofoils which are hoisted vertically along a mast to produce aerodynamic lift. These aerofoils can be trimmed according to prevailing wind conditions, desired boat heading and sea conditions.

1.2.1. Brief Historical Context

For several thousand years, sail powered craft were the primary means for trade, travel, warfare and exploration. From their humble beginnings as small river and lake boats to their largest incarnation as steel clippers used for tea trade in the latter part of the 19th century, sailing yachts have enjoyed a long and useful life in humanity's history.

With the development of modern scientific and engineering processes, sailboat design and naval architecture remained a few steps behind. With the advent of the steamship and further progress in propulsion, sails lost their role as prime mover for the world's ocean shipping. This hindered development simply because the interest was nonexistent and the majority of the research and development was focused on commercial interests. Yachting, for a time, remained the domain of wealthy or dedicated enthusiasts from various backgrounds. This continued until the invention and widespread use of fibreglass as a hull construction material in the 1960s, when sailboats came within the financial reach of the more average consumer.

In the present day, sailboat design can be divided into two major categories; recreational or cruising boats and racing boats. Although many of the top level racing boats employ sophisticated designs, these have yet to filter down completely to the "average" consumer level.

1.2.2. Design and Operating Methods

Historically, sailing vessels have always been designed to suit their purpose. They are survivors of an era before the advent of scientific engineering processes - when boats were designed by a good eye and what was tried and true. A sensible naval architect would base his designs on what was previously done and what worked.

With the advent of the modern era, materials for building boats evolved from wood to iron, aluminium and composites. Boat design methods also faced the need to change: design techniques sought to bridge the gap between traditional design rules and newer ideas. Practical experience requires further bolstering by hard fact and scientific data. What took centuries to refine now needs to quickly adapt. In the words of Dennis Conner, a renowned America's Cup sailor on the topic of boat design: "It basically was an art before. We're just starting to scratch it into a science" [1].

In the wake of this massive change, many design standards, although still functional, were no longer optimal.

The same holds true for the way in which sailboats are handled by their crew; traditional methods for boat handling and sail trim still prevail despite major changes in sailing technology in the past century. Crews rely on past experience and seat of the pants determinations for decision making. Because of the highly subjective nature of personal experiences, there exists a wide variety of sailing methods and techniques, each undoubtedly with their own merit. Comparing these methods is more akin to a political debate than a scientific one.

In this spirit, there is a need to collect quantifiable data to better define the conditions in which sailboats operate and to better define the manner in which they are operated. By accomplishing this, it will be possible to use this data to objectively contrast crew operating methods.

1.2.3. Racing

Racing, in itself is quite a simple concept: to get from point A to point B faster than your opponent. It is a concept nearly as old as time itself, early recorded history mentions foot racing as a popular form of racing. With the advent of other means of transport came new forms of racing: whether on horseback, chariot all the way through to the automobile, racing has evolved and remains a popular occupation, spectator sport and tremendous source of revenue.

For scientific interests, racing is often worthy of further study simply because it creates conditions of extremes. Often, machines are pushed to their limits by the crew in the quest to extract every last ounce of performance. The presence of these extreme conditions is thus an ideal opportunity to highlight the minute differences between competitors which may give them an edge.

This driving influence on optimising racing machines can also have repercussions on more pedestrian products: technological or procedural innovations originally devised for racing can trickle down to consumer products. In the automotive world, anti-lock braking systems or crash safety measures can be cited as examples. Successful 20^{th} century industrialist Soichiro Honda (Honda Motor Company) held the position that “racing improves the breed” [2], by which the pursuit of racing is generally beneficial to the entire realm from an engineering point of view.

In the sailing world, racing is often a big-budget affair; during the 2008-2009 Volvo Ocean race, a single team's race budget exceeded 30 million euros while the known record for an America's Cup campaign is 130 million $US [3]. This serves to demonstrate that there is a commercial interest to improve on the performance envelope of both sailing yachts and their crews.

1.2.4. Summary of Sailing Yachts

In a practical vein, given a hypothetical racing scenario, where two identical boats are competing on the same course in identical conditions, the question becomes: which is fastest? Why? This premise has been the bane of competitors in all manner of racing sports, no matter the method of locomotion. For sailors, racing has always been a fact of life, whether it was two fine merchant ships from the golden age of sail trying to best each other to port to buy the goods or two small dinghies duelling for bragging rights on a quiet lake.

How can we better quantify the reasons why one is better than the other? What methods make a particular crew superior in terms of manoeuvring or boat handling? The causes and reasons can be better explored through quantifiable measurement of the motion of a sailing yacht. Using measurements, a scientific approach can be used to settle the metaphorical argument in a more definitive manner.

1.3. Problem Introduction

Quantifying or measuring the motion of a vehicle is a technique that is employed in numerous fields as a means to improve the underlying design, engineering or use of the vehicle. With quantified knowledge of its motion, it should be possible to derive information on optimal approaches to vehicle handling, as well as ultimately calculating applied loads and conditions.

The present work seeks to apply the same technique to the domain of sailing yachts; *to quantifiably measure the motion of a sailboat while underway*. Particular attention is given to the requirements of sailing, such as the marine environment and the wind as the driving force.

1.4. Thesis Contributions

The current thesis contributes the design and development of a measurement instrument in order to remedy the lack of openly available data on the actual motion that sailing yachts undergo during a race. The format of the solution is a prototype marine data acquisition system incorporating electronic sensors and a recording device. A key consideration is to minimise total cost while still being able to fulfill the proposed function.

The nature of the intended application requires the consideration of environmental conditions present in a sailing context. This includes factors such as quantifying the main driving force, i.e. wind, and also withstanding hazards such as waves, impacts, water, corrosion, etc. Crucial to this is the integration of wind instrumentation and ruggedised packaging.

Although electronic devices which measure and record the motion of objects have found numerous applications in other types of vehicles for control or navigation purposes, there is no documented evidence in the open literature of such applications to sailing yachts.

In this context, the primary contribution of this thesis is the design and development of a marine data acquisition device which successfully acquires relevant data on the motion of a sailboat during a race. The device is designed with the intent to form the basis of a platform for further experimentation on the motion of a sailboat. The specific focus is to provide an instrument which will serve as the basis for the objective comparison of sailboat crew performance.

It is important to keep in mind the context for the possible later use of such as device: information recording can subsequently be used for state estimation by filtering or data fusion techniques. Although this is not the goal of the present work, it does serve as the framework behind several design choices and decisions.

1.5. Thesis Outline

This thesis will describe the process of the design and development of a data acquisition system for inertial measurement for use in sailing yachts. Chapter 2 presents an overview of current and relevant technologies to sailboats and data acquisition systems. Chapter 3 will outline research

objectives and different possible solution concepts will be enumerated and selected. Chapter 4 gives in-depth technical details on the hardware components used, their assembly and the software necessary for the creation of a functional prototype. Prototype testing and validation is described in Chapter 6 including on-the-water tests. Finally the conclusion and a description of future work is given in Chapter 7

Chapter 2 Hardware Review

This chapter reviews technologies and methods relevant to inertial measurement, data acquisition and their application to sailing yachts.

2.1. Data Acquisition Systems

Data acquisition systems are devices that measure physical quantities and convert them into a digital format for storage and processing by computers [4]. The basic constituents of a data acquisition system are (i) sensors which convert the measurement into a signal, (ii) signal processors which treat the signal and (iii) analog to digital converters which convert measurement signals into a digital format.

2.1.1. Sensors

Sensors are at the most basic of measurement devices, simply a detector which measures a quantity in the physical realm and converts it into a signal that can be read by an observer. A basic example is the mercury thermometer, which measures the value of temperature and converts it into a visual signal; in this case a distance which can be interpreted visually by the user.

Modern sensors come in a variety of shapes, sizes and operating principles. They are commonly found in everyday life as devices which we interact with: buttons, switches or the monitoring equipment present in automobiles, machinery, medical apparatus and so on. They can function in an analog manner such as a thermometer or voltmeter or they can function in a native digital state, such as a two way switch.

Important concepts with sensors, regardless of function or measured quantity, are those of accuracy and precision. Accuracy is a measure of how close the measurement is to the actual value and precision is a measure of how unchanged the measurement is under unchanged conditions [5]. The vast majority of sensors are often calibrated by comparing measurements against known standards. Another important concept is that of resolution, which is the least amount of change in measurement quantity which the sensor can detect [5].

2.1.2. Signal Processors

Signal processors are components of a data acquisition system which perform mathematical operations on sensor signals. There are several possible objectives for such processors, including but not limited to: filtering, smoothing, amplification and conversion [4]. The nature of signal processors can vary from mechanical, electrical, electronic or software.

2.1.3. Analog to Digital Converters

An analog to digital converter, or ADC, is a component which transforms a continuous analog measure into a digital discrete quantity [6]. In practise, an ADC reads an analog quantity as measured by a sensor and converts it into a digital equivalent which can be processed by a computer.

The ADC is the last crucial link in a data acquisition system; it allows the computer a means to interface with the real, analog world. A sensor measures a physical quantity, converts it into an analog signal such as a voltage which can then in turn be converted into a digital value.

There are several types of ADCs which all accomplish the same function [4], [6] but simply employ different methods to accomplish this task. Some common types are:

- Successive-approximation
- Ramp-comparison
- Integrating
- Counter-ramp
- Direct-conversion

Most ADCs have a linear response to their input signals, that is their output is proportional to their input [6]. The choice of ADC depends on the type of input signal and its expected characteristics such as speed, frequency, rate of change and so on.

Quantifying ADC performance is accomplished by the following parameters [7] [8]:

- Resolution: the resolution of an ADC is a measure of how many digital values appear as the output over a given range of input signal. Specifically, it is defined for an ADC as a

binary number, since the output is digital. As an example, an 8-bit ADC would have a resolution equivalent to 2^8 or 256.

- Offset and full-scale error: details on offset error are shown in Figure 1 while full-scale error is shown in Figure 2.

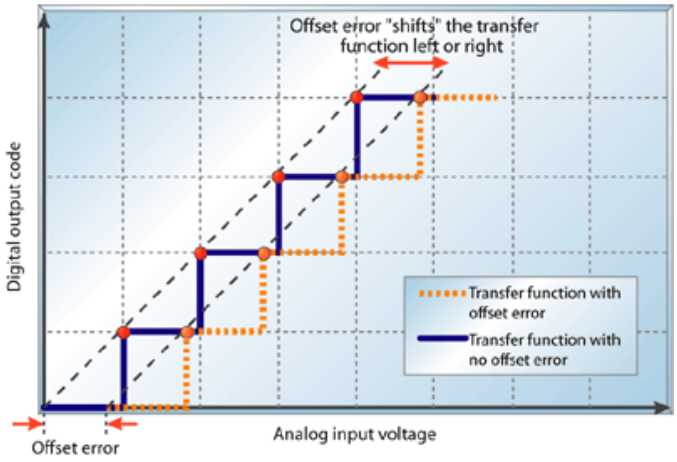


Figure 1: Offset Error [7]

Offset error is caused by the need for the input signal to reach an initial threshold before triggering a change in the ADC output.

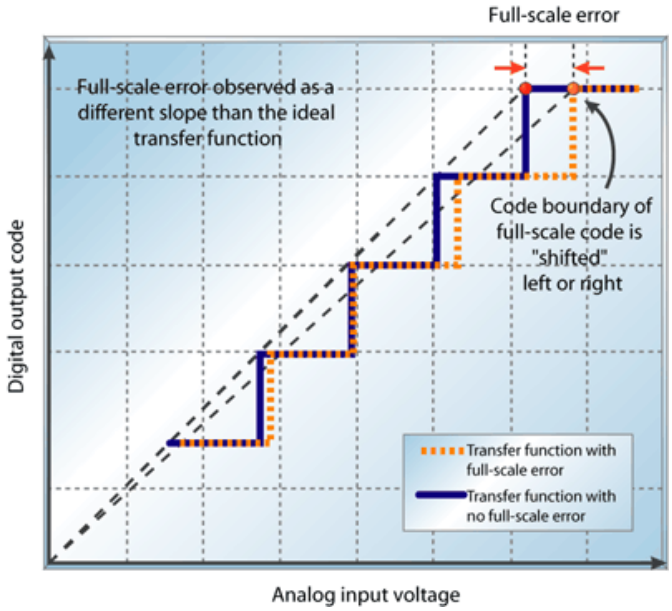


Figure 2: Full-Scale Error [7]

Full-scale error is similar to offset error but is at the other end of the scale: when the input signal reaches the upper limit of the measurable scale of the ADC.

- Nonlinearity. As previously mentioned, an ideal ADC would have a linear output with respect to the input signal. Nonlinearity is a measure of how the actual output signal deviates from this ideal linear relationship. One way to quantify nonlinearity is by differential nonlinearity (DNL) which is defined in equation (2.1) as

$$DNL = \frac{(V_{n+1} - V_n)}{V_{LSB}} \tag{2.1}$$

Here, V_{n+1} is the voltage at the next step, V_n is the voltage at the current step and V_{LSB} is the voltage of the least significant bit (LSB) or the smallest detectable change in voltage.

The final important source of error inherent to ADC is the quantisation error. This particular error can be calculated as the difference between the ADC input signal and the ADC output. Since the output signal is digital in nature (i.e. appears as 'steps'), it can never truly represent an analog signal entirely.

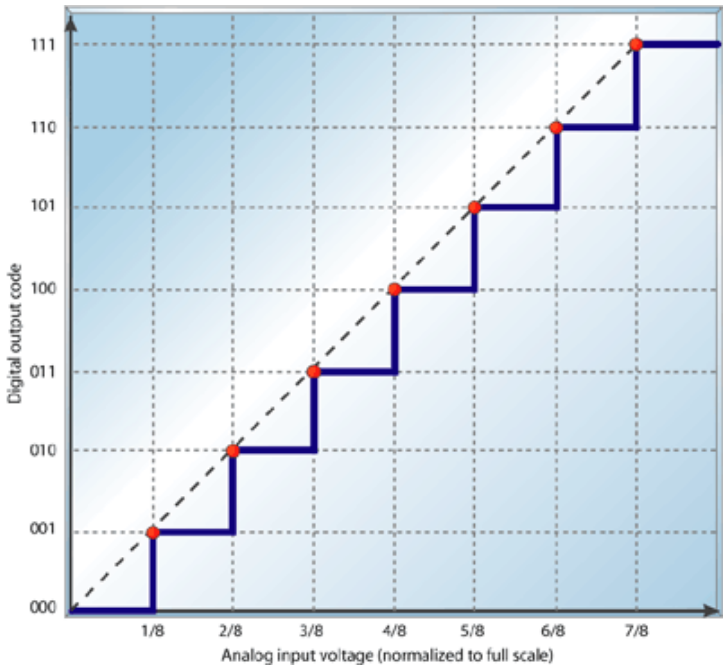


Figure 3: ADC Quantisation Error [7]

Figure 3 illustrates the concept of quantisation error: the dotted line represents a smooth analog ramp signal, the blue line corresponds to the step-like digital output of the ADC.

The dynamic performance of ADCs can be defined much like that of any sensor, by parameters which are defined later in section 2.3.5. These parameters are important in selecting an ADC which is expected to perform at a specified frequency.

2.2. Inertial Measurement Units

An Inertial Measurement Unit (IMU) is a device which quantifies the motion of an object as it moves about the Earth [9]. The can be used either as information gathering devices or for control of the object.

2.2.1. Typical Components

Inertial measurement is in itself usually accomplished by measuring accelerations or velocities, either linear or rotational [10]. Since the world exists in a three dimensional space, linear acceleration is, by convention, measured on three linearly independent orthogonal axes based on the Cartesian coordinate system. On ground based moving vehicles, X-direction is typically oriented forwards, Z-direction is typically upwards and Y-axis towards the left hand side when facing forwards as shown in Figure 4. Further details on the conventions used to describe the orientation of a rigid body in a three-dimensional space are given in 2.2.3.

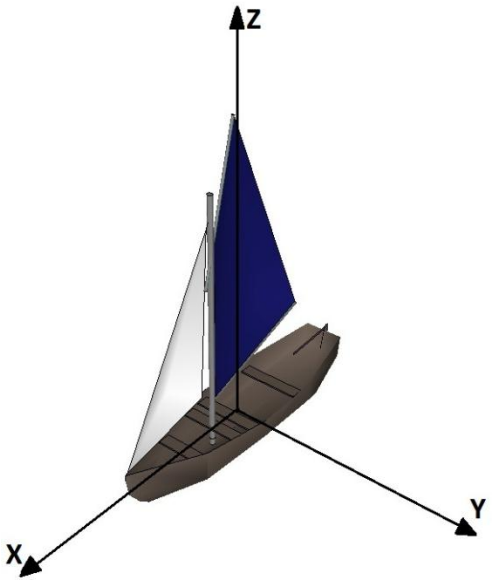


Figure 4: Typical Body Frame Axes for Ground Vehicle

Quantifying linear accelerations is performed with an accelerometer. An accelerometer is an instrument which is somewhat related to a strain gauge; it measures, conceptually speaking, the displacement of a mass in a spring, mass and damper system [4].

Measurement of rotational velocity is undertaken by Micro Electro Mechanical Systems (MEMS) based gyroscopes which are also known as vibrating structures gyroscopes. These take advantage of the Coriolis Effect [4].

Some inertial measurement devices augment the basic accelerometer/gyroscope combination with magnetometers, which provide an indication of the orientation of the Earth's magnetic field. These function like a traditional needle compass, "pointing" North.

Another very powerful technological advance that is becoming omnipresent in inertial measurement is global positioning system technology, or GPS. This technology is based on a constellation of satellites orbiting the planet which provide signals that can be used by a receiver to triangulate position along the surface of the Earth. GPS devices also supply heading and speed while underway and are generally regarded as having good accuracy.

2.2.2. Typical Applications

Inertial measurement technology has a large number of applications in today's world and is integrated into many devices used every day such as:

- Smart phones, such as the iPhone® [11]
- Gaming devices, such as the Nintendo Wii® [12]
- Consumer navigational aids, such as the TomTom® GPS [13]
- Automotive safety devices, such as stability control systems [14]
- Camera stabilisation technology [15]

Other, more sophisticated devices for robotics, aerospace and military applications pioneered the use of inertial measurements and developed some of the technologies involved. IMUs have been incorporated into machines and devices such as:

- Cruise missiles and smart bombs [16]
- Unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) [17]
- Tactical drones and smart munitions [18]

A great deal more applications of inertial measurement devices are employed in research areas to quantify the motion of an object or vehicle, in fields such as:

- Heavy freight and maritime shipping [19]
- Trains and railway transport [20]
- Tunnel mining [21]
- Human kinetics measurement such as athlete motion capture [22]
- Automotive applications [23]

Other fields of interest and novel applications appear frequently in the literature.

2.2.3. Euler Angles

Euler angles are a set of three angles commonly used to describe the orientation of a rigid body in space. They are a specific formulation of the three parameters required to describe orientation

is a 3D Euclidian space. Several definitions of Euler angles exist, but a common one is stated here. It is important to note that this example is only valid if all coordinate systems respect the same handedness.

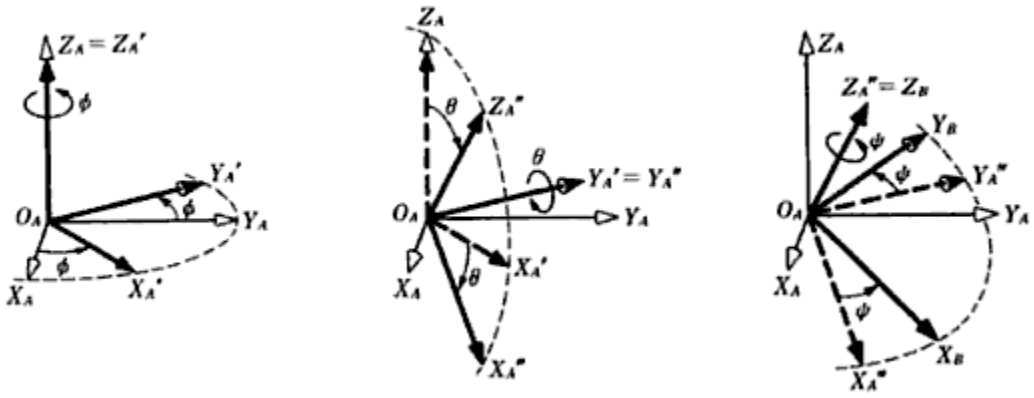


Figure 5: Euler Angles [24]

- The initial orientation of the frame of reference Σ_A is shown by (O_A – X_A,Y_A,Z_A) in Figure 5.
- The frame $\Sigma_{A'}$ is obtained by a rotation of an angle φ about the axis Z_A, shown by (O_A – $X_{A'}$,$Y_{A'}$,$Z_{A'}$) in Figure 5.
- The frame $\Sigma_{A''}$ is obtained by a rotation of an angle θ about the axis $Y_{A'}$, shown by (O_A – $X_{A''}$,$Y_{A''}$,$Z_{A''}$) in Figure 5.
- The frame Σ_B is obtained by a rotation of an angle ψ about the axis $Z_{A''}$, shown by (O_A – $X_{A'''}$,$Y_{A'''}$,$Z_{A'''}$) in Figure 5.

The relationship between the orientation of the rotated frame Σ_B with respect to the initial frame Σ_A is represented by the three angles φ, θ, and ψ. These are known as Euler angles [24]. This relationship is expressed in the form of rotation matrices [24]:

$$ {}^{A}R_{A'} = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2} $$

$$ {}^{A'}R_{A''} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.3} $$

$$ {}^{A''}R_{B} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4} $$

The final relationship between the final frame Σ_B with respect to the initial frame Σ_A is given by [24]:

$$ {}^{A}R_{B} = {}^{A}R_{A'}\,{}^{A'}R_{A''}\,{}^{A''}R_{B} \tag{2.5} $$

$$ {}^{A}R_{B} = \begin{pmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta \\ \sin\phi\cos\theta\cos\psi - \cos\phi\sin\psi & -\sin\phi\cos\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{pmatrix} \tag{2.6} $$

In most cases, it is possible to generalise the above relationship. If R(W,α) describes the rotation matrix between a given frame of reference and its counterpart rotated about an axis W by an angle α, we can generalise equation (2.4) into the following format [24]:

$$ {}^{A}R_{B} = R(Z_A, \phi)R(Y_{A'}, \theta)R(Z_{A''}, \psi) \tag{2.7} $$

From the point of view of inertial measurement, it is possible to obtain instrumentation which directly or indirectly provides Euler angles which can be recorded over time to produce a repeatable history of the motion of an object.

Similar conversions can be used to match local measurements on a vehicle for example, with measurements based on another frame of reference, such as GPS data.

2.2.4. Filtering

Filtering in the context of signal processing is a mathematical construct by which undesirable features are removed from a measured signal. Undesirable features may be elements such as noise or interference. Typical filters include [25] high pass and low-pass filters.

The high pass and low pass filters are the two most basic, they may be combined to form a band-pass filter.

The high-pass filter allows signal frequencies which are above the filter's cut-off frequency to "pass through" whereas any signal frequencies which are too low are cut out [26]. Similarly, the low-pass filter allows signal frequencies which are below the filter's cut-off frequency to "pass through" whereas any signal frequencies which are too high are cut out [26]. The band-pass filter is a combination of a high-pass and low-pass filter with different cut-off frequencies. This filter permits only a range of frequencies to "pass through" [26].

Filters may be specified by family and band-form, leading to certain characteristics of the filter and thus its subsequent effect on the input signal.

2.2.5. Sensor Fusion and State Estimation

Sensor fusion is a combination of numerous sources of data in a way that improves the quality of the information. "Multisensor fusion is a natural extension of using multiple sensors for verification" [25], it is also a way of estimating the state of a system. The state of a system is defined as "the smallest set of variables which determine the behaviour of a system" [26].

Two examples of sensor fusion are the Kalman filter and the complementary filter. The Kalman filter is "the commonest approach to data fusion from separate sources" [25]. A brief description of both the Kalman and the complementary filter is presented here for illustrative purposes.

Kalman Filter

The Kalman Filter [27] is a general filter which "addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time process" [28]. In practise, a Kalman filter estimates a process state at a given (future) time then uses feedback on the actual state of the system through physical measurements. These portions are known as the predictor and corrector equations respectively. The predictor portion is as follows:

$$x_{k+1} = A_k x_k + B u_k + w_k \tag{2.8}$$

where x_{k+1} is the estimation of the next time, A_k is the state transition matrix (Kalman gain), x_k is the current state of the system, u_k is a process control signal, B is the control input matrix and w_k is a process noise signal. The random noise term is assumed to be a Gaussian random variable.

The next portion of the Kalman filter, the corrector is as follows:

$$z_k = Hx_k + v_k \tag{2.9}$$

where z_k is the measurement, H_k is the measurement noise covariance matrix, x_k is the current state of the system and v_k is the measurement noise signal. It is assumed that both noise signals v_k and w_k are independent and possess normal probability distributions [27].

The process is iterated until it converges to a solution in the following sequence [28]:

Calculate the state at k+1:

$$\bar{x}_{k+1} = A_k x_k + Bu_k \tag{2.10}$$

Project the error covariance:

$$P_{k+1}^{-} = AP_kA^T + Q \tag{2.11}$$

Compute the Kalman Gain:

$$A_{k+1} = P_{k+1}^{-}H^T(HP_{k+1}^{-}H^T + R)^{-1} \tag{2.12}$$

Update the estimate using equation:

$$x_{k+1} = \bar{x}_{k+1} + K_{k+1}(z_{k+1} - H\bar{x}_{k+1}) \tag{2.13}$$

Update the error covariance:

$$P_k = (I - K_{k+1}H)P_k^{-} \tag{2.14}$$

This sequence is iterated until the solution converges.

Typically for practical applications, the matrices A_k, B_k and H_k can be modelled as numerical values. The Kalman filter serves as a recursive solution to state estimation using multiple sensor inputs.

Complementary Filter

A complementary filter is a steady-state version of the Kalman filter applied to a specific class of filtering problems [29]. The complementary filter does not consider the statistics of the noise corrupting the signals. The filter is obtained by in the frequency domain [29]. A block diagram on the basic complementary filter is shown in Figure 6.

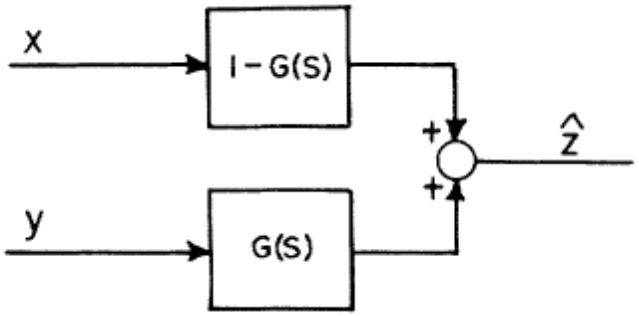


Figure 6: Basic Complementary Filter [29]

Figure 6 shows x and y, noisy measurement signals of a signal z and $\hat{z}$ is an estimation generated by the filter. The first assumption is that the noise inherent to the measurement signal y is high-frequency in nature. The second assumption is that the noise inherent to the measurement signal x is low-frequency in nature. The transfer function G(s) is a low-pass filter which filters the high frequency noise in the measurement signal y. 1-G(s) is a high-pass filter, the complement of the low-pass filter G(s), which filters the low-frequency noise in the measurement signal x [29]. This use of G(s) and its complement 1-G(s) gives the complementary filter its name.

A practical application of a complementary filter is to combine linear accelerometer data with rotational rate gyroscopes in order to estimate orientation [30]. This filter is suitable since accelerometers typically present high frequency noise and rate gyroscope measurements typically present low frequency noise [30]. In this case, the signals in Figure 6 would represent x as a numerically integrated rate gyroscope measurement signal and y as an accelerometer

measurement signal. The relation used to combine these measurement signals into an estimate of the actual state might have the following form:

$$Angle = A(IntegratedAngularVelocity) + B(LinearAcceleration) \tag{2.15}$$

where $\sum A + B = 1$.

Although it is not the intent of the present work to perform any data fusion or filtering, it is important to understand these concepts because they form the basis for the projected use of any prototype device.

2.3. Inertial Measurement Sensors

This section describes sensors typically in use today for the purpose of inertial measurement.

2.3.1. Accelerometers

Several different types of accelerometers are available on the market, the most popular being based on MEMS. These MEMS accelerometers can have a large variety of ranges and sensitivities to accommodate many different application requirements [4]. The frequency response of accelerometers can vary "between zero and a high value" [4] and provide output signals which can easily be integrated into velocity and displacement measurements.

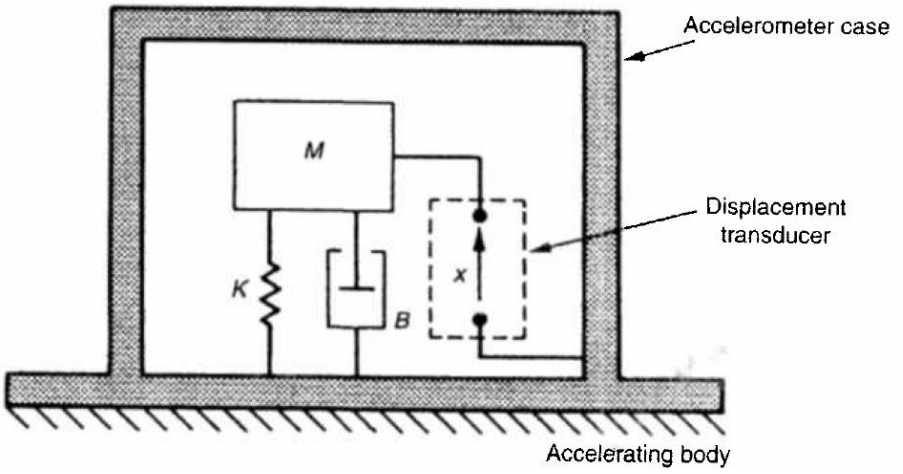


Figure 7: Structure of an accelerometer [4]

The operating principle behind most MEMS accelerometers is the spring-mass-damper system as shown in Figure 7. In this particular case, the spring and damping coefficients are known, as is the mass. From this system, a second order equation of motion for a damped system can be derived as:

$$-Kx - C\dot{x} = M\ddot{x} \tag{2.1}$$

where x is the displacement of the mass, the dot denotes a time derivative, K is the stiffness of the spring and C is the value of the damping coefficient.

The actual measurement portion within the accelerometer is one of displacement, and this is where variety is introduced between accelerometer designs [4]. Some displacement measurement techniques along with their typical range and accuracy [4] are shown in Table 1.

Table 1: Accelerometer Displacement Measurement Techniques

Type	**Range (in multiples of gravitational constant g)**	**Accuracy (+/- % of full scale range)**
Resistive Potentiometer	0 - 50	1
Strain Gauges	0 - 200	1
Piezo Resistive	0 - 200	1
Linear Variable Differential Transformer	0 – 700	1
Variable-inductance	0 – 40	0.25

The choice of accelerometer type often has more complex criteria, such as the expected form of the measured signal, the instrument packaging requirements, desired physical size of the accelerometer, power consumption and cost.

2.3.2. Gyroscopes

A gyroscope is a device which measures the orientation of an object [31], traditionally the attitude or orientation. Gyroscopes can also be used to detect the rate of change of orientation.

The classical gyroscope is comprised of a spinning disc suspended on a set of gimbals to give it multiple degrees of freedom. When the disc is spun at a high rate, the angular momentum of the disc gives it a tendency to maintain its initial attitude when the gimballed assembly is tilted.

Modern gyroscopes can be hybrids of the classical mechanical gyroscopes with electronic sensors or a newer type of electronics sensor: the MEMS angular rate gyroscope.

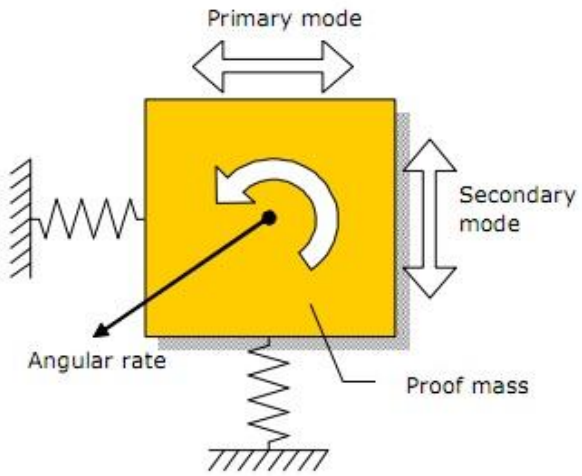


Figure 8: Vibratory Gyroscope Operating Principle [32]

Figure 8 illustrates the basic operating principle of a MEMS vibratory gyroscope. A proof mass is elastically suspended with two degrees of freedom. The sensitive element is vibrated at chosen amplitude, known as the primary mode [33]. When the entire device is subjected to rotation about the sensor axis, the resulting Coriolis force modifies the induced oscillation of the proof mass. This external excitation creates secondary modes present in the oscillation of the proof mass. From the position of the mass, measured in much the same way as in the MEMS accelerometer, it is possible to deduce the rotational rate of the gyroscope from its equations of motion [32]. This is typically accomplished directly within the MEMS gyroscope itself, modern devices simply outputting the measurement directly.

The Coriolis force is caused by the Coriolis acceleration. It is observed in a rotating frame of reference. The Coriolis acceleration is given by [34]:

$$\overrightarrow{a_c} = 2\vec{v} \times \vec{w} \tag{2.2}$$

where $\overrightarrow{a_c}$ is the Coriolis acceleration, $\vec{v}$ is the velocity of the elastically suspended proof mass and $\vec{w}$ is the angular velocity of the rotating body. The cross product of the $\vec{v}$ and $\vec{w}$ terms yields the Coriolis acceleration in an orthogonal direction [35].

Since the intended measured term is $\vec{w}$, it is not feasible to impose a constant velocity on the elastically suspended proof mass in the system. This constraint explains the imposed vibration on the system which, in ideal circumstances matches the natural frequency of the elastically suspended mass.

2.3.3. Magnetic Compass

Magnetic compasses have always been conceptually simple devices and most people are familiar with their use. Of interest here is the more recent innovation of the solid-state magnetic compass, a specialised form of magnetometer magnetic field sensor. Based on the Hall Effect [36], the modern solid-state magnetometer produces a change in voltage that is proportional to the applied magnetic field.

The Hall Effect states that when an electrical conductor is subjected to a magnetic field, the electrons carrying the charge are also subjected to a Lorentz force perpendicular to the direction of the magnetic field and the current. The resultant action of the force on the conductor is to cause a measurable electrical potential difference between the different surfaces of the conductor. For the simplest case of a metallic plate of known dimensions [36]:

$$V_H = -\frac{IB}{ned} \tag{2.3}$$

where V_H is the Hall voltage, I is the current flowing across the metallic plate, B is the magnetic field perpendicular to the plate, d is the plate thickness, n is the charge carrier density and e is the electron charge.

By measuring this voltage difference across a conductor of known geometry and current flow, it is possible to establish the relative magnitude of a magnetic field.

Depending on the application, two or three linear magnetometers are positioned orthogonally, to form a set of reference axes in either a two-dimensional or three-dimensional space. Each sensor

provides a scalar measurement of the intensity of the magnetic field according to its particular orientation [37]. Further integrated circuitry processes these values and performs basic trigonometry to calculate the direction of the magnetic field vector.

2.3.4. Global Positioning System

The global positioning system or GPS is a technology initially developed for military use by the United States in the late 20th century. It was later opened up for public use in 1996 because of its immense potential for scientific and consumer applications. Physically, GPS is comprised of three segments: (i) the space segment, (ii) the control segment and (iii) the user segment [38].

The space segment is composed of a constellation of satellites, minimum 24 currently in operation, orbiting the Earth at roughly 20 200km [39]. Their orbits are configured such that a minimum of six are always visible on most points on the planet's surface. These satellites continually broadcast their satellite identification, position and time of transmission.

The control segment is located in Colorado Springs, Colorado, USA and is tasked with maintaining control over the satellite network [39]. It also monitors the positions of the satellites and calculates the necessary updates with respect to the information broadcast by the satellites in orbit.

The user segment is the one of most interest as it is the receiver, antenna and processing devices required to make use of the GPS satellites. The user segment performs all of the necessary calculations to provide the end-user with position, time, speed and course over ground.

The actual principles behind the calculations required are relatively simple in principle; the GPS satellites provide stationary reference points to the receiver. By comparing the time at which one particular satellite sends a given message with the time at which it was received by the GPS receiver, it is possible to calculate the distance between the satellite in question and the receiver. Repeating this procedure with multiple satellites, the current position of the receiver on the Earth's surface can be calculated by trilateration or "the determination of a distance from three points" [38] [39].

In a simplified example, trilateration in the context of GPS is performed by the GPS receiver which receives a signal from several satellites simultaneously. By knowing how much time was required for the GPS signal to reach the receiver, a relative distance between the satellite and the receiver can be calculated via

$$distance = c \times t \tag{2.4}$$

where c is the speed of light and t is the time for the signal to travel between the satellite and the receiver. By knowing the effective distance between the receiver and multiple satellites of a known position in space as shown in a simplified example in Figure 9, it is possible to determine the position of the receiver

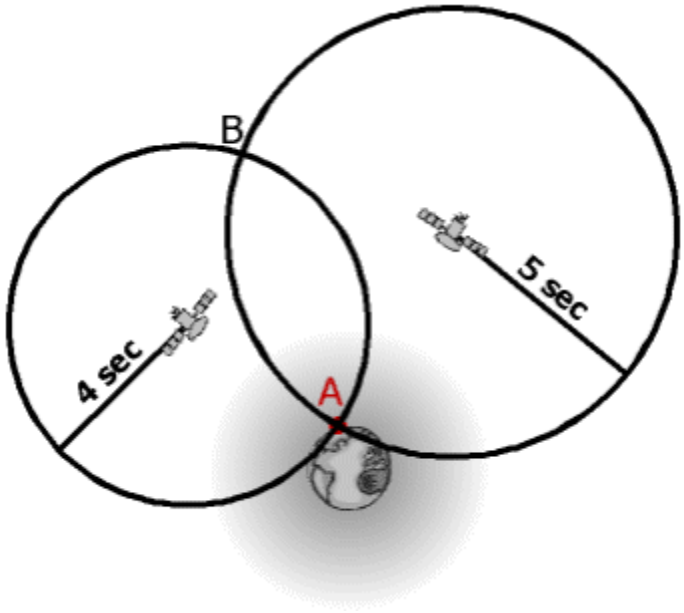


Figure 9: Simplified Trilateration Example [38]

The system of equations for solving this problem is based on the equations for spheres of the type [40]:

$$\begin{aligned} r_1^2 &= x^2 + y^2 + z^2 \\ r_2^2 &= x^2 + y^2 + z^2 \\ r_3^2 &= x^2 + y^2 + z^2 \end{aligned} \tag{2.5}$$

where r_1, r_2, r_3 represent the distances between the receiver and three satellites.

In practise, the technology is far more complex than would appear, and a large number of technical hurdles appear. Factors such as the need for ultra-precise clocks on the satellites and the receivers as well as constant monitoring and tracking of the satellites themselves are key elements which have been overcome in order to allow GPS to function. Overcoming the problem of clock precision on the receiver end is typically done by incorporating signals from more than 3 satellites. Since there is considerable imprecision on the accuracy of the timing, it is oftentimes assumed that the location of the receiver is at or near the surface of the Earth which allows certain boundary conditions to be set.

The final calculation of the position of the receiver should be identical regardless of which three satellite signals are used. This process allows for correction of the imprecision of the measurement by the GPS receivers of the signal travel time. Figure 10 illustrates the error range for a simplified 2D trilateration.

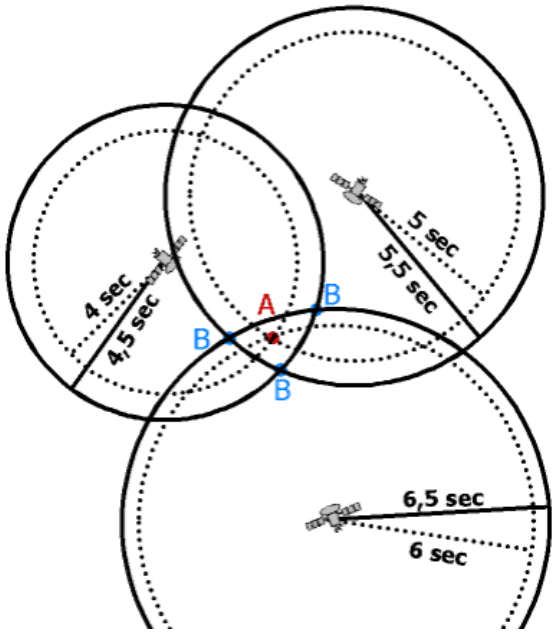


Figure 10: Simplified Trilateration Example Showing Signal Time Errors [38]

There are several possible mathematical solutions to this problem in two dimensions; the circles may intersect at multiple locations (points B on Figure 10). In reality it is known that the receiver should lie at one location, thus by iteration it is possible to refine the solution to obtain a single point within a very accurate tolerance.

Many improvements have been made to GPS technology since its inception to allow for faster location times and increased precision. One such improvement is the Wide-Area Augmentation System or WAAS. This is a series of 38 ground-based reference stations and 3 master stations [41] which serve to enhance accuracy and reliability of GPS.

The end result of this system is the availability of the following measurement capabilities to the receiver device, given in Table 2 [39] [42].

Table 2: GPS Capabilities

Measurement	**Accuracy**
Latitude	100 metres
Longitude	100 metres
Height	156 metres
Time	340 nanoseconds

However, these capabilities do not account for what is now possible, due to the WAAS or other GPS measurement techniques such as differential global positioning systems (DGPS). Some manufacturers of GPS receivers claim better results. For example, the accuracy specifications on the Lassen iQ GPS module [43] are much better and are given in Table 3.

Table 3: Lassen iQ GPS Module Accuracy [43]

Measurement	**Accuracy**
Latitude	8 metres
Longitude	8 metres
Height	16 metres

GPS receivers themselves also perform other calculations, such as those required to determine velocity and heading. This is accomplished by analysing the change in position over a brief period of time.

GPS receivers output the information they calculate according to a standardised format specified by the National Marine Electronics Association (NMEA). The particular standard is known as NMEA-0183 and some relevant details can be found in [44].

2.3.5. Sensor Performance Characterisation

In a general sense, similar sensors can have a wide variety of performance. Standard performance metrics allow the designer to differentiate between them:

- Sensitivity. This is a proportionality constant (scale) describing the relationship between the measured quantity and the output of a given sensor [4]. Typically, a good sensor should present a nearly linear sensitivity in its operating range. As such, the sensitivity represents the slope of this linear approximation of the true behaviour of a sensor. Since the output of an electronic sensor is usually a voltage, the sensitivity constant is given as the ratio of the output voltage over the measured quantity.

- Environmental Sensitivity. Most electronic devices can be subjected to variances in the output signal caused by an environmental factor such as temperature, air pressure or humidity. This environmental sensitivity is expressed as a percentage deviation of the output voltage with respect to a quantity representing the environmental factor. As a specific example, MEMS gyroscopes and accelerometers are affected by temperature. This particular environmental sensitivity is expressed as: $\frac{\%Sensitivity}{Temperature}$.

- Quality Factor (Q-factor). This relates to accelerometers and gyroscopes which incorporate damped spring-mass systems. The Q-factor is the value of the amplitude ratio at the resonance (natural) frequency of the system [45]. Assuming a small damping value less than 0.05, the Q-factor can be given by equation (2.6) where ξ is the damping coefficient of the system:

$$Q = \frac{1}{2\zeta} \tag{2.6}$$

Given a large Q-factor, a sensor system will have a large amplitude at the natural frequency but have a sharp drop-off in amplitude as the forcing frequency is moved away from the natural frequency. In the case of a sensor system with a low Q-factor, the amplitude will be small but the drop-off in amplitude as the forcing frequency is moved away will also be small.

- Bandwidth. This represents the range of frequencies at which the measured variable is not considered small. In the case of a MEMS gyroscope for example, the bandwidth represents a range of frequencies where the proof mass displacement is at least half of the peak value at resonance. For this, the bandwidth is given as:

$$\Delta f = f_o / Q \tag{2.7}$$

Where Δf is the bandwidth, f_o is the natural frequency and Q is the Q-factor [45]. It is important to note that the same low damping value assumption used for the Q-factor must also be applied in equation (2.7).

- Signal to noise ratio. The limitation of what is detectable in the output signal is traditionally quantified as the standard deviation of the ambient (white) noise present in the instrument. The signal to noise ratio is therefore typically given as $\frac{Measured\ Quantity}{\sqrt{Hz}}$ where $\sqrt{Hz}$ is the standard deviation of the magnitude of the ambient noise [4].

Other standard performance characteristics include operating parameters such as:

- Measurement range, example: +/- 1000 degrees/second
- Operating temperature
- Input voltage
- Power consumption

All of these quantities allow a designer to select a given component over another depending on the application and design constraints.

2.4. Data Acquisition Platforms

This section will detail the types of platforms - the basic building blocks - on which a typical inertial measurement data acquisition system is constructed. All of these platforms are based on digital computing principles. The core function of any digital computing device is "the manipulation of data" [4]. In order to function, computing devices require three basic elements: memory, an input/output interface and a central processing unit.

Where the following platforms differ is in their architecture, capabilities and scale. The possible platforms are (i) personal computer based, (ii) microcontroller based data acquisition platforms and (iii) the standalone data logger.

2.4.1. Personal Computer

The Personal Computer (PC) is defined as follows: "any general-purpose computer whose size, capabilities, and original sales price make it useful for individuals and which is intended to be operated directly by an end-user with no intervening computer operator" [46]. The capabilities and form factor of the PC have greatly evolved in the past few decades; what started as the ubiquitous beige box sat atop a desktop has morphed into sleek and portable tablet computers.

Several specialist data acquisition software packages exist which are geared specifically for use with PCs. An example of such a product is National Instruments™ LabVIEW, described as "a graphical programming environment used by millions of engineers and scientists to develop sophisticated measurement, test, and control systems" [47]. This type of software allows the user to connect and interface several sensors while allowing the capability to design and test an interface for a data acquisition system.

As the basis for a mobile data acquisition device, the preferred form factor is one that is portable and compact. This makes laptops or their smaller counterparts' netbooks, interesting. An example of a typical netbook is the Asus Eee PC 1001PXD. This device features a 10.1" screen, Intel® Atom processor, 1GB of RAM and claimed 8 hour battery life [48]. This particular model also includes Universal Serial Bus connectivity.

Personal computers have several advantages as data acquisition platforms, chief among which is their powerful computing abilities. In most cases, post processing of data is done on personal computers, so it is logical to record the data to them directly. The easy user interface and ability to visualise data on the fly are other assets which make the personal computer an attractive choice.

Interfacing of personal computers with sensors can be done using several types of connections including:

- Data acquisition boards
- Parallel ports
- Universal Serial Bus (USB)
- Bluetooth wireless connection

The biggest downside of using a personal computer as a data acquisition platform is that it does not easily support measurements of voltage or analog signals. All of this needs to be accomplished using supplementary hardware.

These supplementary pieces of hardware add to the base cost and complexity of any potential data acquisition system using a personal computer as a platform.

2.4.2. Microcontroller

A microcontroller is "a single chip microcomputer for embedded control applications" [6], a typical example is shown as the black integrated circuit visible in the centre of Figure 11. It includes many of the same basic components as a full sized PC but with far reduced computing and memory capabilities. These components include a central processing unit (CPU), memory and input/output capability (I/O). Microcontrollers are intended for use on embedded applications for a great many modern devices such as appliances, mobile electronics and machine control systems.

Figure 11: Atmel ATxmega16A4U Microcontroller [49]

Microcontrollers are also popular in robotics applications where they are interfaced with sensors and motors to implement different control strategies. For this end, consumer devices are available which simplify the use of the highly miniaturised integrated circuits such as microcontrollers by packaging them on development or "header" boards such as the one shown in Figure 12.

Figure 12: AT90CAN128 AVR Microcontroller and Header Board [50]

Header boards are named simply because they enlarge the very small connector format used by the integrated circuit into a larger, more user-friendly format. Most microcontrollers packaged this way allow for easy connectivity with separate sensors and electronic storage units, making them a good choice as the basis of a data acquisition platform.

Interfacing of sensors with microcontrollers can be readily accomplished using built-in ADCs (see section 2.1.3 for further details). Certain mid-range or higher microcontrollers support other communication protocols such as:

- Inter-Integrated Circuit (I^2C) [51] [52]. This protocol is a multi-master serial single ended bus which incorporates a serial clock ((SCL), serial data line (SDA) and
- Universal Asynchronous Receive Transmit (UART) [52]. This protocol is allows the communication of parallel signals in a serial format. It physically consists of a transmit and receive input.
- Serial-Parallel Interface (SPI) Bus [53]. This protocol is a synchronous serial data link which incorporates a serial clock output (SCLK), a master input/slave output (MISO) and master output/slave input (MOSI) connection.

The main advantage of microcontrollers is their relatively low cost; the AVR microcontroller shown in Figure 12 above retails for 23.96 $US [50].

One popular type of microcontroller used for research or experimental purposes is the Arduino family [54]. As an example of the wide range of the performance envelope available to the researcher, Table 4 provides a comparison of selected members of the Arduino family.

Table 4: Selected Arduino Microcontroller Comparison

Arduino Model	Clock Speed (MHz)	Flash Memory (kilobyte)	Number of Analog Inputs	Number of Digital I/O pins	Number of UART ports	Typical Price in $US (2012) [54]
Mega2560	16	256	54	16	4	58.42
Duemilanove	16	32	14	14	1	22.47
Leonardo	16	32	12	25	1	23.97
Uno	16	32	6	14	1	29.96
Fio	8	32	8	14	1	24.95

Even though the Arduino family of microcontrollers is targeted at a specific group of customers, there is a wide variety in performance criteria such as computational capability, memory, I/O capability and price. Depending on the intended application, each model is a compromise on a different type of performance criteria. It remains to the designer to select the microcontroller solution which best fits the design requirements.

2.4.3. Standalone Data Loggers

Standalone data loggers are a class of commercial product of specialised electronics developed with the specific intent and purpose of powering sensors and storing their measurements for later analysis. Internally, they are usually composed of ADCs, power supply, storage medium and computational unit. A typical example is shown in Figure 13.

Figure 13: HOBO U12 4-Channel External Data Logger [55]

Standalone data loggers are often packaged with software to read and interpret the data from the sensors while performing basic analysis. Retail prices are much higher than microcontrollers alone, in large part due to the packaging and research and development invested in sensor interfacing. The HOBO data logger pictured above retails for 215.00 $US [55].

2.5. Commercial Inertial Measurement Products

Several commercial inertial measurement units are available on the market today. Many are directed at the unmanned aerial vehicle market and scientific researcher. Generally speaking, these products are instruments only, meant to be physically connected to a personal computer which would record the necessary data. Two typical products are presented here, the Crossbow 320 series and Xsens MTi.

2.5.1. Crossbow 320 Series IMU

Crossbow Technology is a subsidiary of Moog Aircraft group, which is a leading supplier of "low-cost, smart-sensor technology" [56]. Crossbow markets several lines of inertial measurement units intended for control of either land-based or aerial vehicles.

The 320 series IMU as shown in Figure 14 is advertised as an entry-level system for "land vehicle control environments", "vehicle dynamics test instrumentation and general purpose inertial measurement applications" [57]. It is based on micro-electro mechanical systems (MEMS) sensor technology and incorporates full 3-axis linear acceleration and rotational rate measurement with a 100Hz data rate [58].

Figure 14: Crossbow IMU320 [56]

Summary information on the Crossbow IMU320 is given in Table 5.

Table 5: Crossbow IMU 320 Summary Information [58]

Sensors	**Measurement range**	**Measurement Unit**
Roll rotational rate	+/- 150	Degrees/second
Pitch rotational rate	+/- 150	Degrees/second
Yaw rotational rate	+/- 150	Degrees/second
X, Y, Z acceleration range	+/- 39.24	Metres/second^2
Physical dimensions		
Length	108	mm
Width	89	mm
Height	150	mm
Weight	0.45	Kg
Power & Connectivity		
Input voltage range	9 - 42	Volts
Output format	RS-232	

Crossbow IMU devices are bundled with Nav-View 2.0 configuration and display software which is meant to be installed on a personal computer; the IMU320 is meant to be tethered to a personal computer in order to function. The bundled software provides facilities to display and record data with the ability to perform basic analysis on the data.

The suggested retail price for the Crossbow 320 series IMU is over 1,000.00 $US according to the manufacturer's policy, however the least expensive obtainable retailer quote was for 1,391.00 $CAD [59].

2.5.2. Xsens MTi 6 DOF Measurement Unit

Xsens is a motion capture and inertial measurement specialist based in the Netherlands. They market a line of inertial measurement sensors with a similar design brief as the Crossbow IMUs. The product of interest is the MTi 6 degree of freedom inertial measurement unit shown in Figure 15.

Figure 15: Xsens MTi 6 Degree of Freedom [60]

The MTi is also a MEMS based package, complete with 3-axis linear acceleration and rotational rate measurement capability. It also includes a 3-axis digital magnetometer to maintain a reference heading with the Earth's magnetic North pole; claimed maximum data rate is 256 Hz [60]. Technical details are summarised in Table 6.

Table 6: Xsens MTi Summary Information [60]

Sensors	**Measurement range**	**Measurement Unit**
Roll rotational rate	+/- 300	Degrees/second
Pitch rotational rate	+/- 300	Degrees/second
Yaw rotational rate	+/- 300	Degrees/second
X,Y,Z acceleration range	+/- 50	Metres/second^2
3-axis magnetometer	0.1	Degrees
Physical dimensions		
Length	58	mm
Width	58	mm
Height	22	mm
Weight	0.05	Kg
Power & Connectivity		
Input voltage range	4.5 - 30	Volts
Output format	RS-232, USB adapter	

Xsens devices are available as a development kit bundle complete with software development kit, hardcopy user's manual and necessary cables. As with the Crossbow, the MTi is also only an instrument and is intended to be tethered to a personal computer for research purposes.

Typical prices depend on the application although one reference puts it at about €1,750.00, or approximately 2,760 $CAD [61].

2.6. Existing Products for Sailing Yachts

This section will describe some of the typical products available to the average consumer in terms of inertial measurement, performance review and navigational aids intended for use in sailing yachts.

2.6.1. Navigation Systems

Navigation systems intended for use on sailing yachts vary wildly in terms of features, cost and use. The simplest devices are the classical compass, map and a pair of divider callipers. Modern devices tend to centre on global positioning system (GPS) technology. GPS units can range from simple handheld displays to built-in computers with mapping, logging and even auto piloting functionality.

Another popular technology used either as a standalone device or in conjunction with a GPS map is a sonar device. Many low-cost uniaxial devices are marketed for sports fishermen while other complex devices integrate them with navigational charts with real-time position display.

An example of a simple GPS navigation device is the Garmin eTrex® 30 shown in Figure 16. The eTrex® 30 is a portable, battery powered handheld GPS device [62]. It can be outfitted with navigational charts loaded on microSD storage devices, this allows the user to set waypoints or destinations and record the track. It offers other useful display functions such as magnetic compass, average speed, current speed and current heading.

Figure 16: Garmin eTrex® 30 [62]

The Garmin eTrex® 30 is waterproof down to 3 metres and is powered by two AA batteries with an advertised 25 hours of battery life [62].

On the other end of the scale, a more complex and permanent device is the Lowrance HDS-10® Fishfinder/GPS chartplotter as seen in Figure 17. This unit combines a sonar and water temperature transducer with a GPS receiver which feed their information into a 264mm LCD display unit [63].

Figure 17: Lowrance HDS-10® Fishfinder/GPS Chartplotter [63]

Inside the display unit is a computer device which can overlay different readings from the sensors and combine them with the navigational charts stored in the unit or in up to two

SD/MMC card readers. Installation of the HDS-10 unit is meant to be permanent, affixed to a console and powered by the vessel's on-board battery supply.

The navigational features of the Lowrance HDS-10® are expanded on the basic features of a handheld unit such as the Garmin eTrex® 30, such as inputting a pre-planned travel itinerary and more detailed maps.

Devices such as these are typical in the arsenal of tools used by sailing yacht skippers to manoeuvre their vessels either during cruising or long-range tripping.

2.6.2. Tacktick

Tacktick is a family of products developed by Suunto Oy of Finland and now owned and marketed by Raymarine Marine Electronics of the United States [64]. The premise behind this family of products is that of a wireless marine sensor and display network.

The Tacktick family is comprised of transducers and displays, the systems can be tailored depending on the budget of the end-user and the application. Transducers include:

- Water temperature
- Water depth
- Hull speed in water
- Combination hull speed and water depth
- Combination hull speed, water depth and water temperature
- Wind speed and direction
- Magnetic compass

Transducers are typically fully waterproofed and self-contained. Some are even solar powered while others require 12V shipboard battery power. The transducers send out the data which they measure on a wireless network, and the data is then received and displayed by a display unit.

Displays follow similar power requirements as the transducers, some are 12V battery powered, others are solar powered. Display units are targeted for different applications, some for the

cruising sailor, while others are designed and marketed for the serious race competitor. An example is the Tacktick T210 multifunctional wireless maxi display as shown in Figure 18.

Figure 18: TackTick T210 Multifunctional Wireless Maxi Display [64]

The T210 is solar powered and displays up to 50mm digits [64] and can display a multitude of raw data or calculated information such as:

- Relative and true wind speed
- Water depth
- Tidal drift
- Course to steer
- Wind shifts
- Race leg timing

This information is of course dependant on the transducers available on the wireless network. The user is free to purchase and install transducers according to his or her timeframe and budget.

The Tacktick family of products is designed as a performance enhancing tool and provides the user with real-time information which can be used to better decide which actions to take. This means that the data provided to the user is of use only immediately and its value is nearly instantaneous; there is no recording function in the Tacktick system. There is also a lack of GPS data and functionality incorporated into the system, likely due to the proliferation of separate GPS systems for use in boats.

2.6.3. Velocitek SpeedPuck

The SpeedPuck is a single product marketed by Velocitek of Hawaii [65]. This company markets a number of GPS-based display products intended for sailors and yacht racers. The SpeedPuck is a large, circular shaped LCD display with integrated GPS as shown in Figure 19. This entire unit is housed in a waterproof enclosure and battery powered to simplify installation and use.

Figure 19: Velocitek SpeedPuck [65]

The GPS chipset incorporated to the SpeedPuck is a u-blox LEA-4A which is a "low power, low cost, high sensitivity GPS receiver that ideally fits the needs of mass market products" [66]. Basic information that can be displayed on the 27mm LCD display is speed in knots, heading (course over ground) and maximum speed. These features are simplified versions of what any typical GPS system will provide the end user.

The main difference and a key feature of the Velocitek SpeedPuck is its ability to record data; it is marketed as a GPS based data-logger. The unit records latitude, longitude, time, speed and heading at a frequency of 0.5Hz with a total storage capacity of 30 hours [65]. This data can then be downloaded to a personal computer through a USB cable.

Another SpeedPuck innovation in the sailing world is the SpeedPlay replay software, which simplifies reviewing the recorded data. SpeedPlay software downloads data from multiple devices and overlays it on Microsoft Virtual Earth map overlay. This allows the user to visually replay the course during a race and compare it with the course of fellow competitors. This can be quite a useful tool to compare and contrast tactical differences and choices made during a race.

2.6.4. Limitations of current systems

Many of the existing navigational aids marketed for use in sailboats have sold in large numbers and have good reviews in the relevant popular press. Whether these are general marine devices marketed at boat owners or specific electronics targeted at racers, most of these devices seem to be oriented towards providing the user with real-time information in the form of a display.

A great many devices are GPS based, because it has become a common and inexpensive technology to integrate into an electronic display. GPS data is also easy to log or record in order to be able to review the data chronologically. Most common GPS devices will indeed record at least a track history of the route travelled. Other, more specialised devices for the sailing market include more relevant sensors such as wind speed and direction, yet no consumer devices currently on the market allow for the recording of data other than that provided by the GPS.

2.7. Wind

Monitoring the weather can be done on both a large and small scale, large scale being the weather information that is distributed through numerous forms of media and smaller scale being that which is of most interest to the current application. Specifically, of interest is the monitoring and recording of the main force for sailboat propulsion: the wind. Wind in a sailing context can be defined as a vector field existing over the surface of the water, neglecting the vertical component of the field, thus rendering it two-dimensional.

Specifically, the two relevant quantities are wind speed and wind direction. Several factors can affect the orientations and magnitudes of the wind vector field including obstructions such as other boats, shore effects or weather effects such as clouds and the sun's rays. It is therefore important for sailors to keep track on the ever changing wind conditions as their vessel traverses on the water's surface.

It is also important to note that when discussing wind speed and wind direction on moving sailboats that these measurements are taken using the moving vessel as the frame of reference. Wind speed and direction are therefore relative to the boat, rather than the more usual "static"

reference frame of the Earth, as is conventional in weather reports. The relative wind speed and direction are referred to as apparent wind.

2.7.1. Wind Direction

Quantifying the wind direction can be done simply, using visual inspection of a flag which can ascertain its rough direction; this has long been the traditional method. This eventually evolved into the traditional decorative weather vane.

The operating principle behind the weather vane is that the indicator arrow is allowed to freely rotate about a vertical axis and will tend to align itself with the direction of the wind. The observer can determine the wind heading from the fixed reference markers, aligned with the Earth's cardinal points.

Refining the concept of the traditional weather vane, modern electronic versions incorporate a rotational position sensor to determine the orientation of the vane. Different types of rotational position sensors have been used including, rotary potentiometers, mechanical switches and optical encoder [6]. All of these output the position of the vane using electric signals.

2.7.2. Wind Speed

An instrument which measures the speed of the wind is known as an anemometer [67]. Two primary categories exist, speed-based anemometers and pressure-based anemometers. The first type, speed based anemometers directly measure or infer the velocity of the wind. Different types exist such as the rotating cup, windmill and ultrasonic anemometer. All of these measure wind speed directly.

Pressure-based anemometers employ Bernoulli's equation to establish a relation between pressure and velocity [68]. A particular type of pressure anemometer is known as the Pitot tube, commonly found on aircraft because the high velocities encountered make speed-based anemometers impractical.

Commonly used in maritime application is the rotating cup type velocity anemometer. This device rotates around an axis when subjected to moving air and there is a relation between the

rotational rate of the anemometer and the wind speed. Further details and information on a rotating cup type velocity anemometer are given in section 5.2.5.

Chapter 3 Research Objectives

In this chapter, the objectives and general goals of this research will be explained in detail, along with an overview of their intended application.

3.1. General Objectives

One of the goals of this research is to develop a system that will permit the quantification of the movement of a sailboat under racing conditions. This quantification can demonstrate several key points.

First, this will help to create a better understanding of the loads to which the hull is subjected while underway. These loads are typically modelled as hydrodynamic effects such as waves, turbulence or drag. In practise, one cannot ignore the effects of other factors, such as sail and rig loading, crew movement and inertial effects. The combined effects could potentially be better understood by measuring the motion of the sailboat subjected to all these effects.

The ability to quantify the impact of a sailboat cresting over a large wave while being heeled over by a strong gust of wind is something which structural, aerodynamic or hydrodynamic modelling is not able to accomplish. There is no comprehensive model which accounts for all of these factors in the literature, likely due to the complexity and large number of parameters that need to be modelled. As such, the first goal is to generate a broader understanding of the combined effects of all loads which affect a sailboat while underway through the ability to measure the motion of the sailboat.

A second goal is to create the foundation of a tool that can be used by sailors to improve their performance in a race – in other words, to address the question of what makes a particular crew or boat faster than another.

This tool would allow the user to record his or her boat's performance and grant the ability to review and analyse data retroactively. Through comparison with prior performances or competitor's performances, a team can determine which changes to make to their techniques in

order to improve their performance. This method of data recording for later analysis is currently applied in aeronautical and automotive disciplines with great success [69] [70].

Thus, the goals of the research are as follows:

- To design a prototype inertial measurement data acquisition system which can quantifiably measure and record the motion of a sailing yacht while under racing conditions.
- To validate this prototype under several test conditions.
- To successfully demonstrate that the prototype can record data on a full scale test during a sailboat race.

3.2. Specific Objectives

The general objectives of the prototype marine data acquisition system are to provide a means by which to quantify the motion of a sailing yacht. Specifically, this includes:

- Accelerations
- Position relative to the Earth
- Orientation and heading
- Velocities and speeds

The prototype should also provide some indication as to the state of that most basic driving force of sail boats: the wind. Without knowledge of the wind, the measurements of the boat's motion lack context. Due to practical considerations and the physical limitations of packaging a system onto a boat, the measurement of wind will be taken from onboard the moving boat: relative wind speed and relative wind heading.

Furthermore, all measurements should be recorded electronically for later analysis. The system should provide a simple means by which to transfer this data to a personal computer for such an analysis.

Finally, the system should be able to stand up to the rigors of the environment found on a sail boat, including the potential presence of water (salt or fresh) and shocks or impacts.

3.3. Intended Application

The intended application for the proposed device is that of a sailboat under racing conditions. Racing is a relevant application for inertial measurement because it provides the greatest concentration of extreme conditions. During the course of a race, boats are pushed to their limits in terms of speed, manoeuvres and weather conditions. Sailboat racing is thus a favourable environment for which to apply inertial measurement because of its potentially extreme nature.

3.4. Design Requirements

The design requirements for the proposed marine data acquisition system can be outlined as follows:

- Correctly record the motion of a sailing yacht while underway.
- Determine and record the spatial position of the sailing yacht at any time in customary units of latitude and longitude.
- Provide contextual information regarding wind conditions.
- Withstand the rigors of the environment found on any boat, such as the possibility of water or the impact of waves.
- Minimize the cost of parts of the prototype.

The actual measurement variables of interest are outlined in 3.4.1.

3.4.1. Variables of Interest

The variables of interest along with the relevant sensors are shown in Table 7. This table also correlates the variable of interest with the proper sensor and desired measurement units.

Table 7: Variables of Interest

Variable	Sensor	Units
3-axis linear acceleration	Accelerometer	Metres/second^2
3-axis rotational rate	Gyroscope	Degrees/second
Magnetic heading	Compass	Degrees
Wind speed	Anemometer	Metres/second
Wind direction	Weather Vane	Degrees
Global position	GPS	Radians
Global heading	GPS	Degrees
Planar velocity	GPS	Knots
GPS satellite time	GPS	Seconds
Measurement Interval	Timer	Microseconds

The prototype data acquisition system must therefore fulfill all of the research objectives outlined above.

Chapter 4 Concept Generation and Selection

This chapter will provide a full description of the specifics of selecting the concept on which the prototype data acquisition system was built.

4.1. Design Specifications

Detailed design specifications for all instruments are summarised in Table 8.

Table 8: Design Specifications

Measurement	Measurement Range	Sensitivity
Linear Acceleration	+/- 30 metres/second^2	0.1 metres/second^2
Rotational Velocity	+/- 200 degrees/second	5 degrees/second
Magnetic Heading	0-360 degrees	1 degree
Relative Wind Speed	0-50 knots	1 knot
Relative Wind Heading	Usual Cardinal references (N, NE, E, etc.)	π/8 radians
GPS Heading	0-360 degrees	1 degree
GPS Speed	0-50 knots	0.1 knot
GPS Position	N/A	+/- 50 metres
GPS Time	N/A	1 second
Measurement Duration	0-50000 microseconds	1 microsecond

The accelerometer specifications are based on what is typically used for measurement in automobiles. Airbag control accelerometers are typically triggered at 50g (490 m/s^2) [71] [72], while the maximal acceleration seen under braking or cornering by a racing car is typically around 1.5 to 2g (14.7 to 19.6 m/s^2) [73] [74]. This threshold of 2g (19.6 m/s^2) is typically used in automotive applications [75] [74]. As a sailboat moves much more slowly than the average automobile, this limit will be used as a guideline. Some room is left in the threshold to accommodate for the possibility that the actual accelerations experienced by a sailboat may be higher as they remain largely unknown.

The gyroscope range is based on what is considered a truly worst case scenario: catastrophic capsize of a boat. Regardless of the cause, prior experience shows that this rare occurrence tends to happen rather quickly, on the order of 0.5 seconds [76]. Since this event has serious repercussions to the crew's performance, it is important that a prototype data acquisition system be able to record this event. Although it is hard to obtain quantified information on capsizing, it is estimated that it involves the rotation of the boat by approximately 90 degrees in a very short interval of 0.5 seconds. This equates to 180 degrees/second: the measurement range leaves some margin on this estimation.

The GPS specifications are based on what is normal for commercial GPS modules subjected to civilian restrictions [39].

The sailing community's convention is to measure wind speed in knots. Typically, sailing is possible from between 5 to ~ 40 knots [77] and only in extreme case does a boat skipper care to venture out in storm-force winds. Since wind speed measurements will be relative to the moving craft, allowance is made to account for the possibility that the relative wind speed could be null or higher than the true wind speed as measured from a stationary point.

Wind direction in weather forecasts is conventionally given as a cardinal point, such as North (0 degrees) or North-East (45 degrees). These usually represent rough headings in increments of approximately 45 degrees. In maritime conventions, a further measure of precision is given by dividing the headings, for example North-North-East (22.5 degrees). By the same convention, full circle headings are divided into 16 equal portions of $2\pi/16$ radians or 22.5 degrees. Since the measurement of wind direction on a sailboat is affected by a multitude of factors, not the least of which is due to the motion of the craft itself, it is assumed that there is no need to have any greater resolution than 22.5 degrees for relative wind direction measurement in order to provide sufficient contextual information.

Battery life requirements are set to exceed the duration of the average club race. In practise, most club races last for one evening or a weekend afternoon [78] and should account for travel time to and from docking points. With this in mind, the minimum battery life for the device should be 6 hours.

Finally, since this design is to serve as a prototype, it should be able to accommodate modifications and changes easily. Eventual testing or practical use may highlight the need to modify or add sensors according to needs which may become apparent later.

4.1.1. Design Specification Summary

The design specifications are outlined in Table 9.

Table 9: Design Specification Summary

Specification	**Description**
Basic Device Measurement Requirements	
	Measures linear acceleration in all three axes
	Measures rotational velocity in all three axes
	Measures magnetic heading with respect to North
	Measures relative wind heading
	Measures relative wind speed
	Measures GPS location, speed, heading
	Measures timing interval
Simple to assemble	
	Minimal number of components
	Availability of components
Packaging	
	Battery life duration is 6 hours
	Water-resistant
	Shock-resistant
	Device is portable
Cost	
	Lowest total cost of major components
Flexibility	
	Can accommodate future software modifications
	Can accommodate future hardware modifications

4.2. Concept Generation

This section describes the considered concepts which could satisfy the design requirements and specifications.

4.2.1. Independent Sensor Network

The independent sensor network relies on the premise that each sensor can function on its own or independently from other sensors. A schematic of this concept is shown in Figure 20.

An individual sensor will record each of its measurements along with the necessary timing information. Later, the data from all of these independent sensors would be transferred to a personal computer for post-processing.

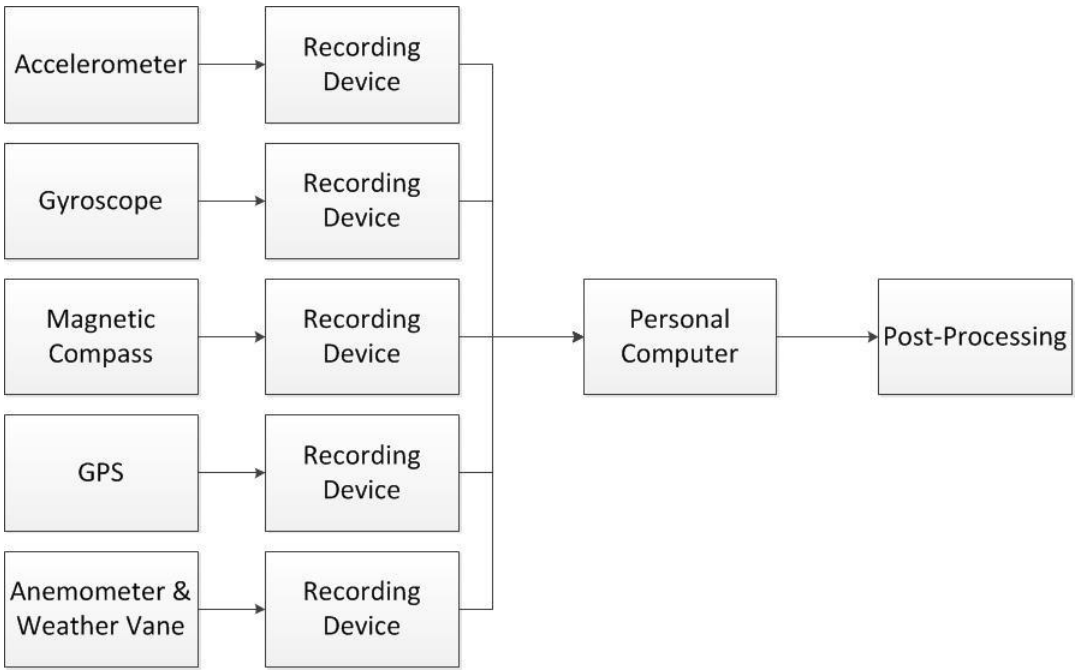


Figure 20: Independent Sensor Network

The post-processing would unavoidably need accomplish the synchronisation and overlay of each separate measurement into one coherent and continuous dataset.

4.2.2. Microcontroller Based Centralised Sensor Network

The microcontroller-based centralised sensor network concept is to have all of the required sensors physically connected through a wired network, all feeding into a central point: a microcontroller. A schematic of this concept is shown in Figure 21.

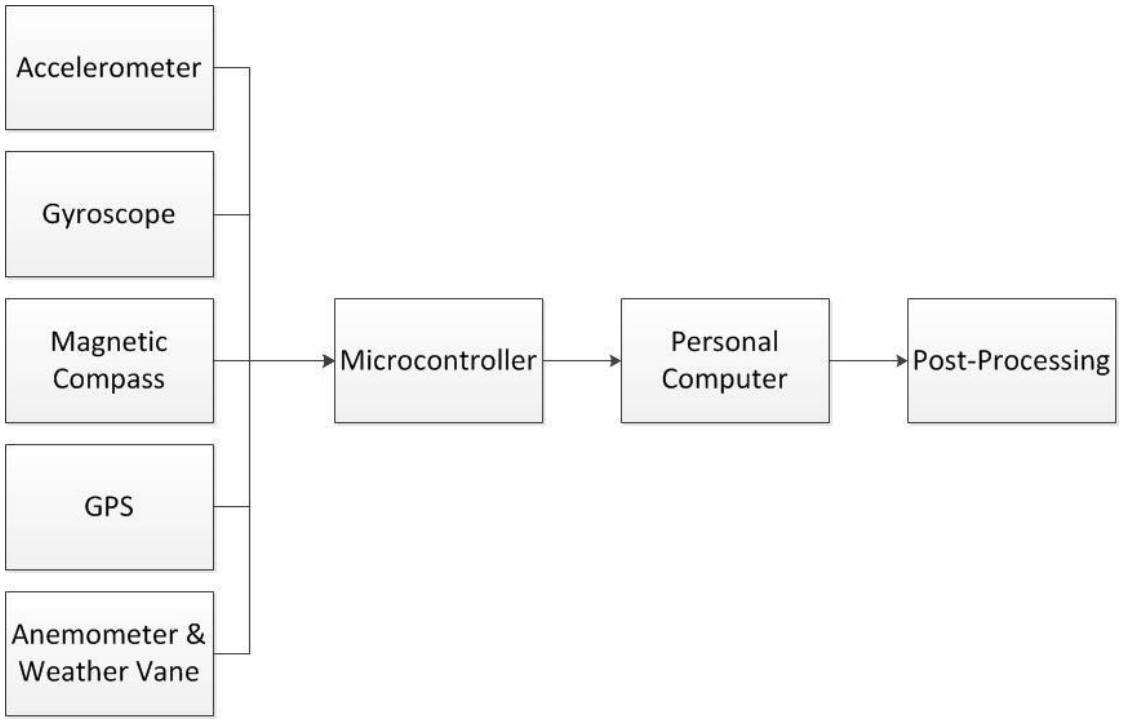


Figure 21: Microcontroller Based Centralised Sensor Network

This microcontroller will handle the polling and synchronisation of the sensor data as it occurs. Recording the measurements can be handled either through the microcontroller or through a personal computer. Post-processing will again be handled on a personal computer with the data already synchronised and continuous.

4.2.3. Microcontroller Based Wireless Sensor Network

This concept is similar to that which is described in section 4.2.2 with the critical difference that the link between the sensors and the microcontroller is wireless. A schematic of this concept is shown in Figure 22.

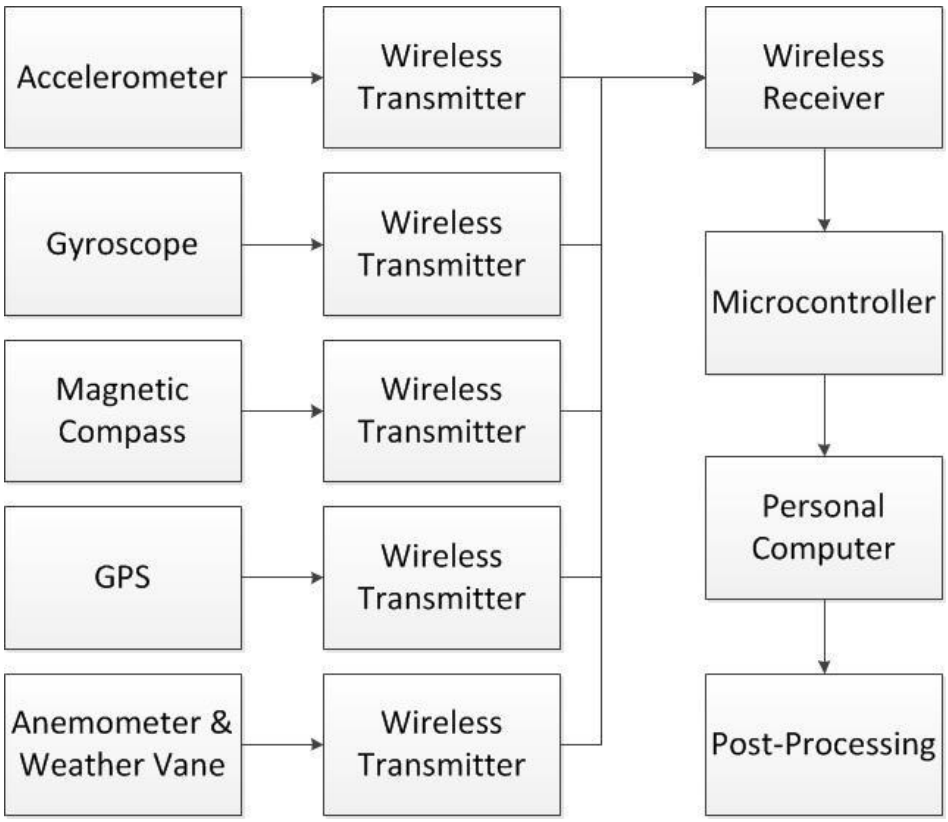


Figure 22: Microcontroller Based Wireless Sensor Network

4.3. Concept Selection

In order to select the concept which best fills the desired role for a prototype data acquisition system, a weighted rating matrix was prepared and is given in Table 10. Each selection criteria is based on those given in 4.1.1, although refined to highlight the contrasts between each possible concept. These criteria are weighted according to their perceived importance; subsequently, each concept is assigned a subjective rating. The selection criteria are introduced here, with each criterion given a (percentage) weight in accordance with its perceived importance in the overall concept: The sum total of all the weightings is 100.

1. The ability to fulfill all of the basic measurement requirements given in section 4.1.1 is necessary in order for the system to accomplish the intended function. Although it would appear obvious, this selection criterion is important in order to differentiate potential

solutions on their facility to achieve the basic requirements. As such, the chosen weighting for this is 10%.

2. The simplicity of assembly is a factor which is considered for two reasons. First the facility to construct and deconstruct the prototype in order to test and debug. Secondly, this can be used as a metric for the ease with which the prototype can be replicated. A lower weight of 5% is given since this is considered relatively minor.
3. The number of components is a good metric of the complexity of a system; the fewer the number of parts, the simpler the system. This can affect reliability and cost and also benefits assembly and design. Because this is a major consideration in selecting a prototype that will be easier to design, construct and troubleshoot, a weight of 15% is assigned to it.
4. The availability of components used in the device is used as a metric of the ease of parts acquisition during the construction phase and also gives an indication of how common they are in the marketplace and how well supported they are by the vendor or user community. This is also considered a minor factor therefore a weight of 5% is assigned.
5. The portability of the concept is the notion of how easily it can be physically transported and how it can be installed on a sail boat. This is a useful criterion only during the actual use of a fully developed system; as such a minor weight of 5% is assigned.
6. The total cost of major components is a major criterion due to a constrained development budget; it is weighted accordingly at 20%.
7. Flexibility in use is meant to quantify how easily the concept will be to actually employ and use; how many components to mount separately, how difficult it will be to turn on all the necessary devices and later send the data to a personal computer. Similarly to portability, this criterion impacts the end-use of the fully developed system and is considered minor with a weight of 5%.
8. One major criterion is the ability of the concepts to accommodate software changes and modifications. Since the entire premise of the design is to function as a prototype, it should be somewhat over-specified in this respect to facilitate future experimentation. This is considered a major criterion and is assigned a weight of 15%.

9. The final criterion is the ability to accommodate hardware modifications or upgrades. Due to the nature of prototypes, it is wise to be able to accommodate changes such as supplementary sensors for the evolving needs of the project. This is considered a major criterion: Since hardware modifications are generally more effort than software, this criterion is weighted slightly higher at 20%.

Having established ratings, the three concepts presented in section 4.2 are given a subjective rating of 1 to 5 for each selection criterion (1 being poor and 5 being ideal) according to the designer's experience, prior research and preliminary testing. The final ratings are presented in Table 10.

Table 10: Concept Selection Matrix

		Concepts					
Selection Criteria		**Independent Sensor Network**		**Microcontroller, Wired**		**Microcontroller, Wireless**	
	Weight	Rating	Weighted	Rating	Weighted	Rating	Weighted
Basic measurement requirements	**10**	3	30	3	30	3	30
Simple to assemble and build	**5**	2	10	4	20	3	15
Least number of components	**15**	0	0	5	75	3	45
Availability of components	**5**	2	10	4	20	3	15
Portability	**5**	5	25	2	10	4	20
Lowest total cost	**20**	0	0	4	80	2	40
Flexibility in use	**5**	4	20	3	15	5	25
Future software modifications	**15**	2	30	3	45	3	45
Future hardware modifications	**20**	2	40	5	100	2	40
Total 100		20	**165**	33	**395**	28	**275**

Although the unweighted scores of both the microcontroller-based wired and wireless networks are very close, the intent for this initial design selection is to maintain simplicity, therefore the

principal differentiating factor between these two concepts is the added layer of complexity of the wireless portion. This differentiating factor is highlighted through the weighted scores.

The chosen concept is consequently the microcontroller based centralised sensor network, with a weighted score of 395, against 165 for the independent sensor network and 275 for the microcontroller-based centralised remote sensor network.

4.4. Core Hardware Selection

This section outlines the microcontrollers which were initially considered as design options for an inertial measurement data acquisition system prototype. There are three prime candidates; the LEGO Mindstorms NXT, the Axon II and the Arduino.

4.4.1. LEGO Mindstorms NXT

The LEGO Mindstorms is an educational product marketed by the LEGO company in numerous versions since 1998 [79]. At the heart of the product is a programmable microcontroller capable of accepting several different types of sensor inputs and generating motor outputs. The user is meant to incorporate this functionality with LEGO Technic system bricks, which have several mechanical components such as gears, shafts, axles, wheels to create simple robots or automated machines. The innovation of this product was combining it with a simple visual programming language intended for children or the more powerful National Instruments LABView based language.

The latest version of LEGO Mindstorms is the NXT, which has a 32-bit ARM7 microcontroller, 3 motor output ports, 4 sensors input ports and Bluetooth capability [80]. Power is supplied either by 6 x 1.5V AA batteries or a rechargeable Li-Ion power pack. Programming and computer interface is typically done through USB connection with a PC.

LABView software is a powerful tool and industry standard for computer interfacing with measurement systems and is fully compatible with the Mindstorms NXT's USB or Bluetooth connections. Several higher-end sensors are available from third party vendors.

HiTechnic is an official third party sensor supplier for the LEGO Mindstorms line of products [81]. There are several sensors which could be applicable including:

- 3-axis accelerometer
- Single axis gyroscope
- Compass

The primary limitation of the Mindstorms NXT is the restriction to four sensor inputs and no GPS device readily available. While this lack of GPS could be remedied with a supplemental device, the limitation to four sensor inputs cannot be overcome using a single Mindstorms NXT unit.

4.4.2. Axon II

The Axon II is the second generation of a microcontroller product intended for use for amateur or scientific robotics prototyping. It is based around Atmel's ATmega640 processor, which incorporates a relatively large number of sensor inputs and device outputs. It was originally intended to fill a void in the market for consumers wishing to have a single device controlling multi-limbed robots or robot arms [82] as existing devices did not incorporate sufficient inputs and outputs and thus required "no additional messy breadboard wiring or expensive shields" [82] to interface devices.

The Axon II incorporates the following features [83]:

- 16 Analog to digital converters (ADC)
- Support for 25+ Servos
- I2C, SPI
- 3 Universal Asynchronous Transmit Receive (UART) + USB
- Up to 8 external interrupts
- 15 PWM Channels
- 64KB Flash, 4KB EEPROM, 8KB SRAM
- 16 MIPS throughput at 16 MHz
- 6 Timers (four 16-bit, two 8-bit)
- Pre-programmed with a bootloader - no programmer required
- Numerical LED display
- Built in 3.3V, 5V, and unregulated power buses

Programming the Axon II is accomplished through Atmel's AVR Studio, a C based programming language. Numerous open-source libraries exist which offer support for sensors and motors. Chief among these is Webbotlib, which is specifically designed for robotics applications. Webbotlib also includes a portion tailored specifically to the Axon II and several popular sensors including gyroscopes, digital compasses, accelerometers and GPS modules.

This versatility and the availability of a wide range of devices seem to be both the primary advantage and biggest downfall of the Axon II. While it is tempting to imagine that a wide variety of combinations of sensors and devices have been interfaced successfully due to the open-source nature of the Axon II, the reality is that these interfaces are often not always optimal. Nevertheless, it is well supported through a very active web-based community [84].

4.4.3. Arduino Mega 2560

The Arduino family is "an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software." [54]. It is also based on AVR's ATmega series of microprocessors. The Arduino platform lists some of its advantages as being inexpensive and having a "simple, clear programming environment" based on AVR's C programming language.

One of the more powerful members of the family, the Arduino Mega 2560 was considered as a possible candidate. The Arduino Mega 2560 incorporates the following features [85]:

- 16 Analog to digital converters (ADC)
- Support for 14 Servos
- I2C, SPI
- 4 UART
- Up to 6 external interrupts
- 14 PWM Channels
- 256KB Flash, 4KB EEPROM, 8KB SRAM
- 16 MHz clock speed

As previously mentioned, programming the Arduino can be accomplished in a few different ways, first with "Arduino Integrated development environment" [54]. This is a cross platform Java application which is intended as a more introductory programming language for the

inexperienced programmer. Since these microcontrollers are also from the ATmega family, they can be programmed using Atmel's AVR Studio environment.

Several open-source libraries exist which support various sensors and motors intended for robotics applications. The Webbotlib library, similarly to the Axon II, also supports ATmega based Arduino boards. Webbotlib supports several popular sensors including those of interest; accelerometer, gyroscopes, digital compasses and GPS modules.

The Arduino Mega 2560 is a popular board and is generally highly regarded by scientific and general users alike [54]. It is flexible in design and has been used in a large number of projects thanks to its low cost. It does require the end-user to design and build some hardware-to-interface sensors, particularly when it comes to powering the Arduino since the Mega does not include a built-in regulated power bus.

The Arduino hardware model is based on the use of modular "shields" or supplementary boards which overlay one atop the next in order to interface the desired components. The shield model is practical and helps to drive down costs in order to keep the basic Arduino simple and clean. The downside is that interfacing devices does incur additional costs and complexity in terms of buying and installing these shields.

4.4.4. Microcontroller Comparison

A brief comparison of the selected microcontrollers is given in Table 11.

Table 11: Microcontroller Comparison

	LEGO Mindstorms NXT	Axon II	Arduino Mega 2560
Processor			
Flash memory (KB)	256	64	256
EEPROM (KB)	N/A	4	4
SRAM (KB)	64	8	8
No. of timers	N/A	6	N/A
No. of interrupts	N/A	8	6
Programming language	LabView	C	C / Arduino IDE
Power & Connectivity			
5V Regulated output voltage	No	Yes, 1500mA max	500mA max
3.3V Regulated output voltage	No	Yes, 90mA max	Yes, 50mA max
Recommended input voltage range	9	6 - 7.2	7.0 -12
USB Support	Yes	Yes	Yes
Dimensions			
Maximum width (mm)	144.8	64.5	101.6
Maximum thickness (mm)	61	20	20
Maximum length (mm)	96.5	65.5	53.3
Weight (g)	250	40	40
Retail Price ($US)	279.99	108.00	58.95
Inputs			
No. of analog to digital converters	N/A	16	16
I2C	Yes	Yes	Yes
SPI	No	Yes	Yes
No. of UART ports	0	3	3

4.5. Summary

The chosen design concept is a microcontroller based centralised sensor network. In application, this is a collection of the sensors required to gather the desired data, wired directly to a microcontroller. This microcontroller will read, interpret and synchronise the measurements into a coherent data output to be recorded. The information gathered will then be transferred to a PC for further post processing using technical computing software tools.

Several possible candidates for a microcontroller have been described in this chapter, including the LEGO Mindstorms NXT, Axon II and the Arduino Mega 2560. An objective comparison between the different microcontroller choices was presented. Microcontroller and further hardware selection is discussed in the next chapter.

Chapter 5 Hardware Selection and Integration

5.1. System Architecture

The proposed system architecture is shown in Figure 23. The system architecture is an expansion of the concept chosen in section 4.3, the microcontroller-based centralised remote sensor network. The chosen sensors are the Razor 6DOF IMU for acceleration and rotational rate measurements, the HMC6352 Digital Compass for magnetic heading, the Venus 634LPx GPS receiver and an anemometer and wind vane to measure the wind information.

Two options are provided to link the AxonII microcontroller with a personal computer; these modes are USB output and SD (Secure Digital) cards. The reasoning and implementation of these methods of linking to a PC are further clarified in sections 5.3 and 5.5 respectively.

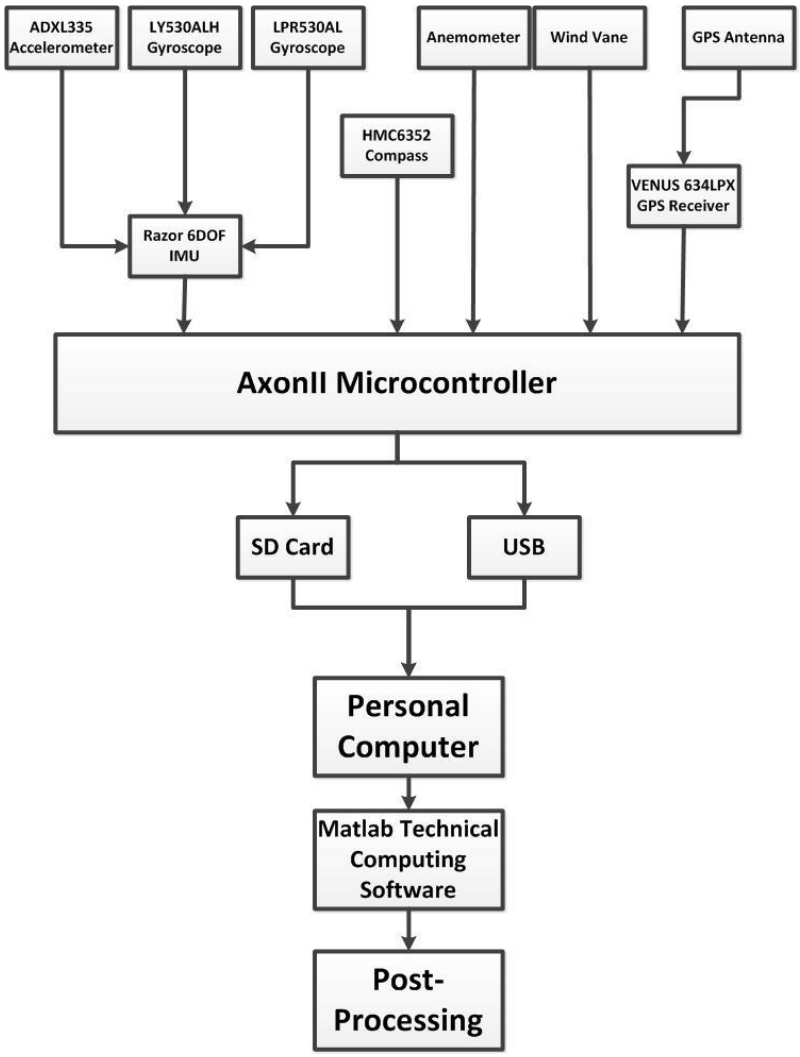


Figure 23: Proposed System Architecture

A further description of each component in the system is provided in section 5.2 and how they are physically interfaced is explained in section 5.3. Further details on programming the microcontroller and Matlab software are given in sections 5.3 and 5.5. Finally, other considerations such as packaging and batteries are considered in sections 5.7 and 5.8 respectively.

5.2. Hardware Components

The hardware components described here were chosen because they were deemed to best fit the system specifications and are known to be compatible [53]. The system design specifications have been previously established in section 4.1.

5.2.1. Microcontroller

As previously mentioned, the chosen microcontroller to build the proof-of-concept setup is the Society of Robot's Axon II. This particular microcontroller was chosen for several key reasons:

- Ease of hardware integration and wiring
- Bigger regulated power supply
- Community support [84]

Finally, the AxonII microcontroller has the simplest format by which to add sensors, requiring no shields and minimal external wiring. This factor is important in the prototyping phase to accommodate on-the-fly modifications or address any additional hardware requirements which may arise during testing.

Complete specifications for the AxonII microcontroller can be found in Appendix A.1.

5.2.2. Six Degrees of Freedom Razor – Ultra-Thin IMU

The Razor 6 Degree of Freedom – Ultra-Thin inertial measurement unit is a product marketed by Sparkfun Electronics of Boulder, Colorado, United States. It is a sensor combination board designed for hobbyists and electronics enthusiasts. The Razor IMU integrates commercial MEMS accelerometers and gyroscopes which are packaged as chips with a much more user-friendly breakout board format [86], Figure 24. This format allows the end-user to use and experiment with MEMS chips which are intended to be used in high-volume original equipment manufacturing applications.

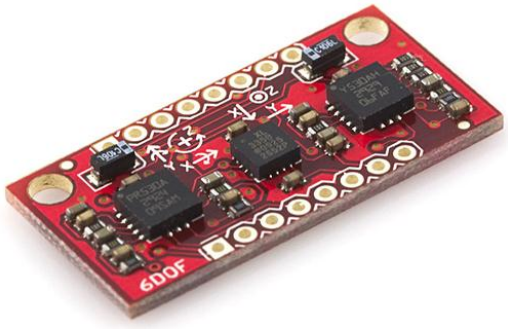


Figure 24: 6 DOF Razor IMU [86]

The Razor IMU incorporates three MEMS devices, an Analog Devices® ADXL335 3-axis accelerometer, a single axis STMicroelectronics™ LY530ALH gyroscope and a dual axis STMicroelectronics™ LPR530AL gyroscope [86]. These are mounted on a printed circuit board with 0.1" pitch standard size headers [86] which are large enough to be soldered and manipulated with hand-held tools.

5.2.2.1. Accelerometer

The MEMS based accelerometer included in the Razor IMU is the Analog Devices® ADXL335 3-axis device; it can be seen as the centremost chip on the Razor IMU in Figure 24. This is a newer version of the accelerometer chip used on the Nintendo® Wii remote [87]. Complete specifications for the ADXL335 can be found in Appendix 0 and a condensed form is presented in Table 12.

Table 12: ADXL335 Condensed Specifications

		Units
Measurement Range	+/- 30	Metres/second^2
Power Supply	1.8 – 3.6	Volts

The ADXL335 accelerometer is capable of measuring acceleration on three orthogonal axes with a range of +/- 30 m/s^2. Determining the sensitivity and resolution is done by first calculating the microcontroller's ADC resolution using (5.1) [6] .

- Assuming a 10-bit ADC, N = 1024 steps.

- V_{cc} = 5V, ADC reference voltage.
- 3.3V is peak accelerometer output from supply voltage.

$$ADC\,Resolution\left[\frac{mV}{step}\right] = \frac{V_{cc}}{N} = 4.8828125\left[\frac{mV}{step}\right] \tag{5.1}$$

From the ADXL335 specifications given in appendix A.2, for an accelerometer supply voltage of 3.3V, the accelerometer sensitivity is 330 [mV/g]. Converting this into conventional units [mV/m/s^2] is shown here:

$$Sensitivity\left[\frac{mV}{m/s^2}\right] = 330\left[\frac{mV}{g}\right] \times \frac{1}{9.81}\left[\frac{g}{m/s^2}\right] = 33.6391\left[\frac{mV}{m/s^2}\right] \tag{5.2}$$

Finally, combining the results of (5.1) and (5.2), sensor resolution for all three accelerometer axes is given by:

$$Sensor\,resolution\left[\frac{m/s^2}{step}\right] = \frac{ADC\,Resolution\left[\frac{mV}{step}\right]}{Sensitivity\left[\frac{mV}{m/s^2}\right]} = 0.14513 \tag{5.3}$$

The accelerometer resolution is therefore 0.14513 m/s^2 per ADC step.

5.2.2.2. Gyroscopes

Two MEMS-based gyroscopes are packaged within the Razor IMU, a single axis STMicroelectronics™ LY530ALH (Y stands for yaw) gyroscope and a dual axis STMicroelectronics™ LPR530AL (PR stands for pitch and roll) gyroscope. The LY530ALH is visible in Figure 24 as the rightmost chip while the LPR530AL appears as the rightmost chip on the Razor IMU's Printed Circuit Board (PCB).

The complete specifications for the LY530ALH are given in Appendix A.3 and specifications for the LPR530AL are found in Appendix A.4. Some relevant information is summarised in Table 13.

Table 13: Gyroscope Condensed Specifications

	LY530ALH	LPR530AL	Units
1x Output Measurement Range	+/- 1200	+/- 1200	Degrees/Second
1x Sensitivity	0.83	0.83	mV/Degrees/Second
4x Output Measurement Range	+/- 300	+/- 300	Degrees/Second
4x Sensitivity	3.33	3.33	mV/Degrees/Second
Power Supply	2.7 – 3.6	2.7 – 3.6	Volts

Both gyroscopes can output in two modes, an unamplified 1x output and amplified 4x output. This choice affects the possible measurement range and instrument sensitivity. For this application, the 4x amplified output mode was used.

Determining the sensitivity and resolution of the gyroscopes is similar to the process described in 5.2.2.1. Starting with the results for the microcontroller's ADC resolution shown in (5.1) [6] and using the same assumptions, an identical result is obtained.

From the gyroscope specifications given in A.3 and A.4 for a 4x amplified output, an identical sensitivity is obtained for both sensors: 3.33 mV/Degrees/Second. Combining the sensitivity and microcontroller ADC resolutions is shown in equation (5.4).

$$Sensor\ resolution\left[\frac{Degrees/s}{step}\right]=\frac{ADC\ Resolution\left[\frac{mV}{step}\right]}{Sensitivity\left[\frac{mV}{Degrees/s}\right]}=1.4663 \qquad (5.4)$$

The gyroscope resolution is therefore 1.4663 Degrees/Second per ADC step for all gyroscopes.

5.2.3. Digital Compass

The magnetic compass implementation uses a Honeywell HMC6352 solid state magneto-resistive based device. This is also packaged on a PCB provided by Sparkfun Electronics, Figure

25. The PCB includes a standard 0.1” pitch header which makes using the HMC6352 possible with standard soldering and handling equipment.

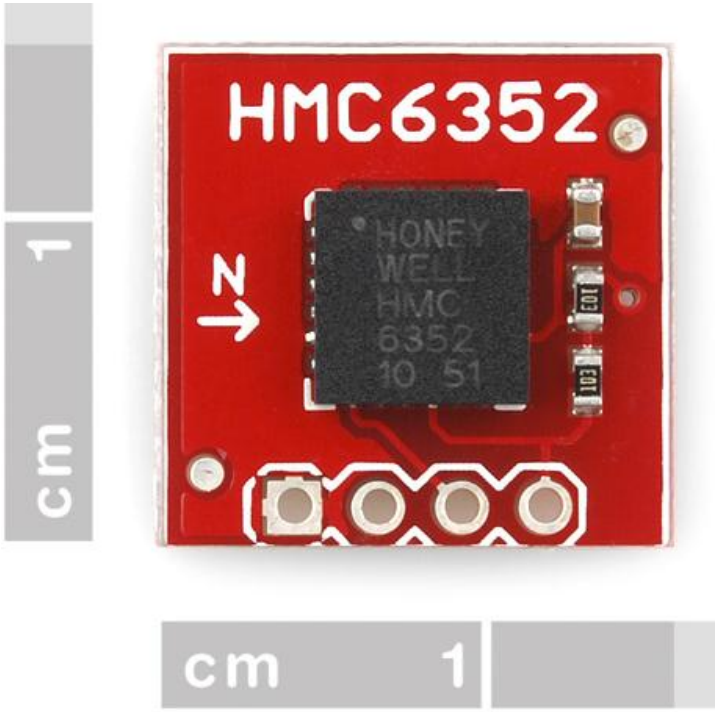


Figure 25: Honeywell HMC6352 [88]

The integrated circuit does all of the necessary interpretation and converts the output to a magnetic heading directly, with a resolution of 0.5 degrees and 1 degree of repeatability. This heading value is outputted as an integer corresponding to the bearing in degrees and the outgoing signal is relayed to the Axon II through an I^2C connection. Complete specifications for the Honeywell HMC6352 are found in Appendix 0 and summarised in Table 14Table 14: Honeywell HMC6352 Condensed Specifications.

Table 14: Honeywell HMC6352 Condensed Specifications

		Units
Measurement Range	0 - 360	Degrees
Sensitivity	0.5	Degrees
Power Supply	2.7 – 5.2	Volts

This solid state magnetic compass is subject to the usual limitations of the traditional needle compass, such as interference effects from large metal structures or nearby magnetic fields.

5.2.4. Global Positioning System

The GPS receiver device used is based on the Venus 634LPx manufactured by SkyTraq Technology Inc. of Taiwan. It is mounted to a PCB and provided with an external SubMiniature version A (SMA) for an external antenna mount. The entire device is sold by Sparkfun Electronics as the Venus GPS with SMA Connector [89], as shown in Figure 26.

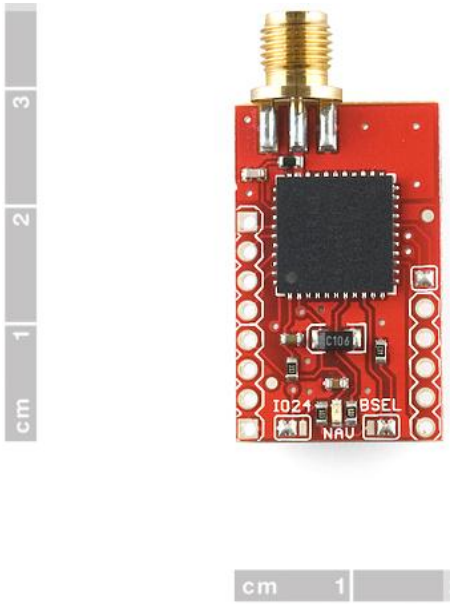


Figure 26: Venus 634LPx GPS [89]

Complete specifications for the Venus GPS receiver can be found in Appendix A.6 and are condensed in Table 15.

Table 15: Venus 634LPx Condensed Specifications

		Units
Possible GPS satellites tracked at once	14	
Maximum update rate	10	Hz
Power Supply	2.7 – 3.3	Volts
Claimed accuracy	2.5	Metres
Hot Start duration	1	Seconds
Cold Start duration	29	Seconds

The GPS receiver returns several values in the format of a standardised string at an adjustable data rate. The standardised messages outputted by the Venus GPS are defined by the National Marine Electronics Association (NMEA) standard [44]. The useful data are latitude and longitude positions, time, course over ground (bearing) and speed over ground. These messages are then interpreted and read by the AxonII microcontroller.

The SMA connector mounted on the device PCB requires the connection of an external antenna, in this case one with a magnetic mount and 5m cable, shown in Figure 27. The antenna has a gain of 26dB and a voltage standing wave ratio (VSWR) of less than 2.0 [90].

Figure 27: ONSHINE GPS Antenna ANT-555 [90]

Out of convenience, this particular antenna has a magnetic base for easy attachment and is fully waterproof for outdoor use. Its long cable, compact size and waterproof enclosure make it a good choice for a temporarily mounted antenna for use in a marine environment.

5.2.5. Wind Instruments

Both weather instruments used are originally from a kit imported by Argent Data Systems as the "Weather Sensor Assembly p/n 80422" [91], shown in Figure 28. On the left hand side of Figure 28, the anemometer is visible while the weather vane is on the right hand side. The basic unit also comes with a rain gauge which was not used.

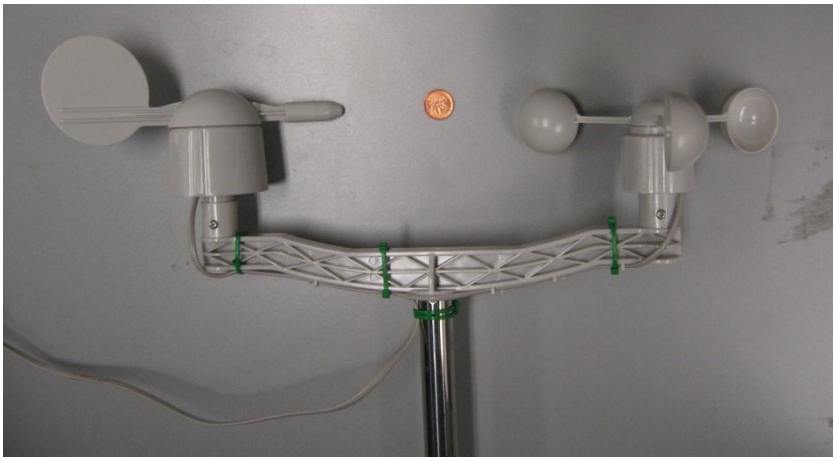


Figure 28: Argent Data Systems Weather Sensor Assembly [91]

Design specifications for the anemometer and weather vane are given in section 4.1. Actual specifications and instructions for the Weather Sensor Assembly can be found in section 5.2.5.

5.2.5.1. Anemometer

The anemometer in the Weather Sensor Assembly is a rotating three cup device; the force of moving air causes the rotation of the cups about a vertical axis. Inside the anemometer, a reed switch is activated twice every rotation by completing an electrical circuit. This generates a series of pulses, the frequency variations of which can be related to speed over a short interval of time, (in this case one second) by the relation given in equation (5.5) [92].

$$\frac{Measured\ Pulses}{[second]} \times 1.2 \left[\frac{kmh}{pulses} \right] \times 0.5366 \left[\frac{knots}{kmh} \right] = Measured\ Windspeed\ [knots] \qquad (5.5)$$

The actual counting and conversion is performed during post-processing as described in appendix B.3.3.

5.2.5.2. Weather Vane

The weather vane is a flat vane type, free to rotate about a vertical axis to maintain alignment with the wind's direction. Internally, the wind vane uses a voltage-divider type of sensor and measures wind direction in increments of $1/16^{th}$ of a full circle (22.5 degrees increments). The voltage outputted by the weather vane is a function of the varying resistance of the sensor [92] which is shown in Table 16.

Table 16: Weather Vane Direction to Resistance Equivalence

Direction (Degrees)	Resistance (KΩ)
0	33
22.5	6.57
45	8.2
67.5	0.891
90	1
112.5	0.688
135	2.2
157.5	1.41
180	3.9
202.5	3.14
225	16
247.5	14.12
270	120
292.5	42.12
315	64.9
337.5	21.88

The AxonII simply records the ADC channel value and interpretation of the raw data is performed later during post-processing.

5.2.6. Secure Digital Storage Media Interface

To accommodate the possibility of having on-board storage, a Secure Digital (SD) or Multi-Media Card (MMC) interface is needed. This is provided in the form of the Breakout Board for SD-MMC Cards with Push-Push type socket [93] shown in Figure 29.

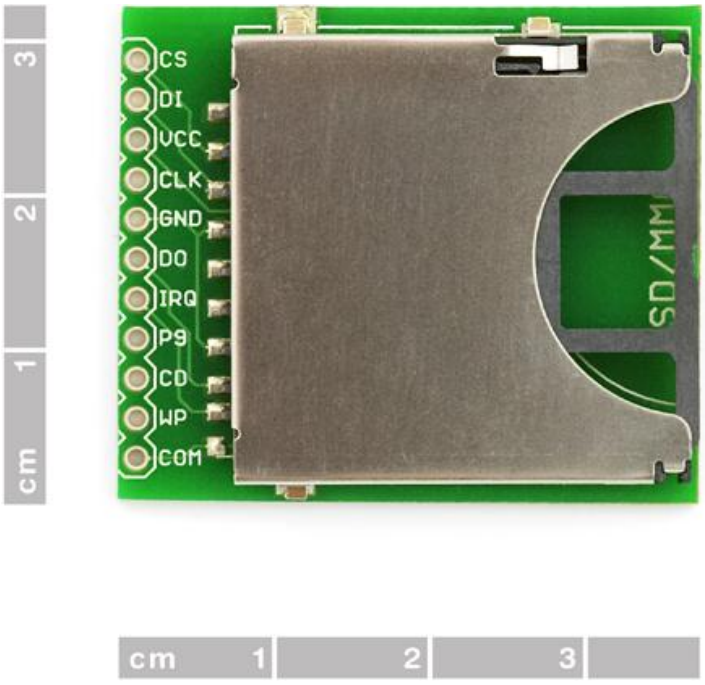


Figure 29: Breakout Board for SD-MMC Cards [93]

This breakout board features standardised 0.1" pitch headers for connections [93]. It can be interfaced with the AxonII's Serial-Parallel Interface (SPI) [53] and can accommodate varying capacities.

5.3. Hardware Integration

An overview of the all the components used is presented in appendix C.1. Physical assembly is possible using basic electronic soldering and wiring tools:

- Soldering iron, lead-free solder, flux

- Diagonal cutting pliers
- Wire crimpers
- Wire insulation strippers

Physical connections and interfacing are described below.

5.3.1. Physical Connections

All of the hardware components described in section 5.2 are interfaced to the AxonII following the wiring schematic shown in Figure 30. The HMC6352 compass, the Razor 6DOF IMU and the Venus 634LPx are all soldered on a 0.100" pitch prototyping board using standardised pin headers (shown in Figure 32) which also provides structural mounting for the components. Wiring headers are soldered onto the prototyping board which allow for the connection of wires between the AxonII and the sensor components.

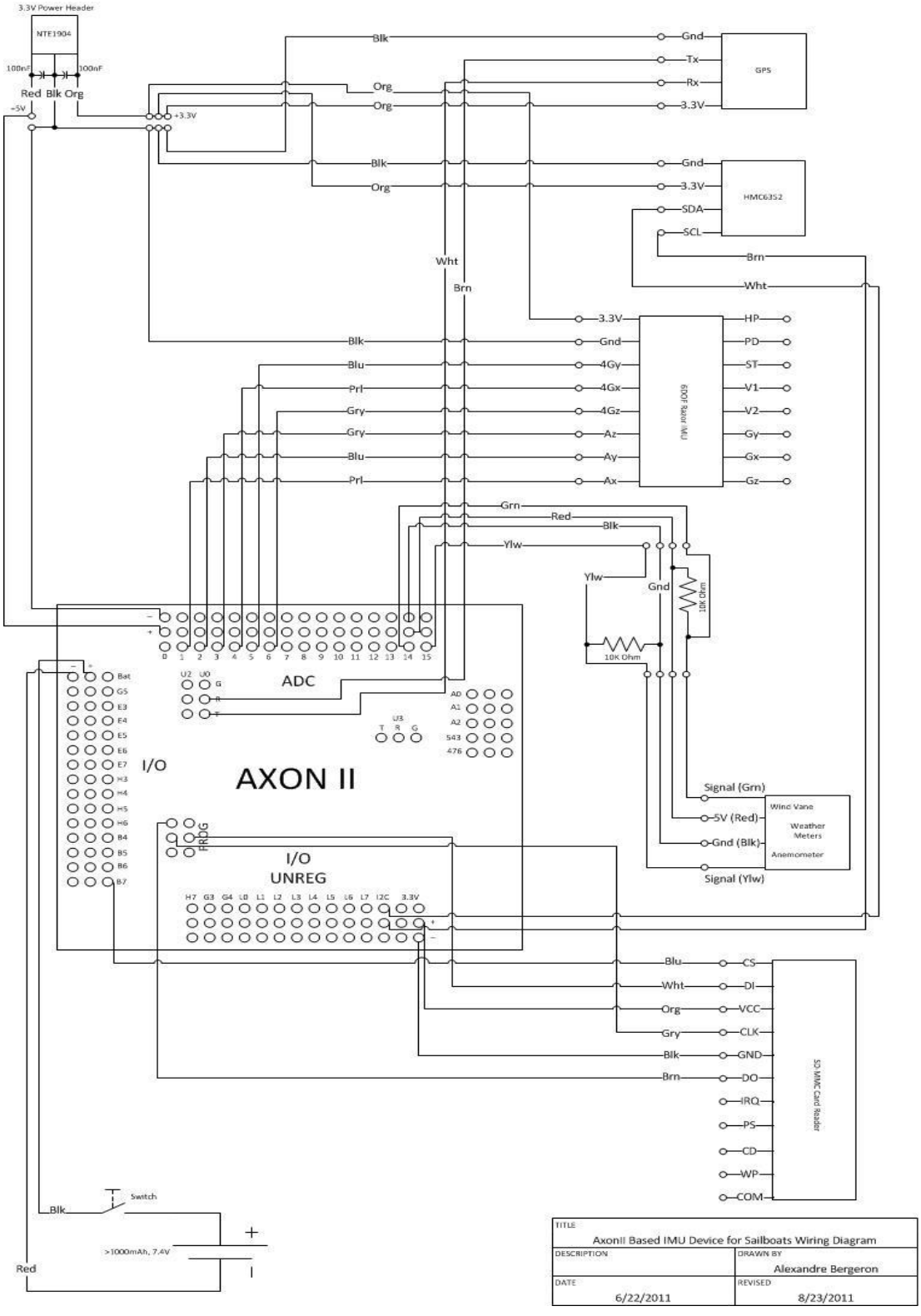


Figure 30: Data Acquisition Prototype Wiring Diagram

The connectors used are standard Molex 0.100" pitch connectors described in appendix C.3.

Other components including the power switch, battery and Secure Digital Storage Media Interface are connected to the AxonII using similar connectors but are not physically soldered to the prototyping board for ease of access.

One limitation of the AxonII microcontroller is in the small number of available 3.3V outputs, in fact there is only 73mA of current available [83]. To overcome this limitation, it was necessary to put into place a separate 5V to 3.3V conversion integrated circuit using an NTE1904 Positive 3 Terminal Voltage Regulator. This is powered by one of the AxonII's 5V outputs and is able to accommodate power for the compass, GPS and Razor IMU. The onboard 3.3V ports are used only to power the SD card. Details on how this circuit is laid out are shown in Figure 30.

5.3.2. Microcontroller to Device Interfacing

The GPS device is designed to send standard NMEA message strings at a pre-set Baud rate through a generic UART port. Since the Venus 634 LPx can function at several Baud rates, it is important to configure the AxonII UART port Baud rate to match the GPS output Baud rate. First, it is necessary to perform a one-time configuration of the Venus GPS to match its output Baud rate to the AxonII's default 115200. At the same time, the GPS refresh rate is set to its maximal possible value of 10Hz. The procedure required to accomplish these changes is described in [53] [89].

The GPS NMEA string messages are outputted to the AxonII according to the standard format described in NMEA standard 0183 [44]. Webbotlib provides functions which interpret NMEA strings from a generic GPS device. This standard library function was modified to optimise performance as described in section 5.6.

The Honeywell HMC6352 digital compass is set up to use an I^2C interface as a slave component. Internally it is accessed at the address 0X42 and comes from the factory set up to simply output its reading continuously. It is interfaced using functions from Webbotlib which:

- Create an I^2C bus.
- Add and initialise a device called compass with the correct properties.

- Receive and interpret the I^2C message and return the value for use in the main program.

This pre-built library simplifies the programming required on the AxonII in order to interface the sensor correctly.

However, some further set-up is required to send a message to the compass in order to modify its default refresh rate from 1Hz up to 20Hz. This implies a one-time configuration of the compass by sending it the correct message as described in [53] [51].

Interfacing of the accelerometers in the Razor 6DOF IMU is achieved through an ADC sampling of the output voltages. This requires proper configuration and identification of the AxonII's built-in ADC channels: i.e. channel 1 is x-axis acceleration, channel 2 is yaxis acceleration, etc. This channel numbering is shown on the wiring diagram in Figure 30.

SPI for SD CARD

The SD Card used for the standalone interface mode described in section 5.5 requires the AxonII microcontroller to treat it as a disc. This is done through an SPI bus interface set up using Webbotlib functions which allow the setup of an SPI bus using a standard UART port. Data is stored in a buffer until sufficient data is accumulated in the buffer to fill one block on the SD card. This buffer is also set up using standard Webbotlib functions.

5.3.3. PC to AxonII Interfacing

PC connectivity for the PC interface mode outlined in section 5.3 is accomplished through a standard mini B to type A USB connector. The Axon II uses a Silicon Labs CP210x USB-to-UART bridge chip [83] to convert its lower level UART signals to a more convenient USB format.

On the computer end, drivers create a virtual COM port, which Windows PCs treat like an older style 9-pin (RS-232) serial port. Details on how to correctly install the required drivers are shown in appendix D.1. It is important to correctly set the PC COM port Baud rate to match the AxonII's rate of 115200.

For programming the AxonII microcontroller, the AVR Studio integrated development environment (IDE) is used. Installation details for this particular IDE are given in appendix D.2. As previously stated in section 4.4.2, AVR Studio is a C based environment and is used for programming all generic Atmel microcontrollers.

In order to tailor AVR Studio to the particular hardware layout of the AxonII, templates are used from the Webbotlib library. The AxonII specific template contains information and presets which allow the use of AxonII hardware such as:

- UART to USB conversion chip
- 7-segment LED display
- Built-in push-button

Webbotlib also contains libraries which serve as the basis to interface:

- Standard GPS devices
- I^2C devices
- SPI for SD card output

Details on the installation of the Webbotlib library are outlined in appendix D.3.

Once a particular version of code is compiled by AVR Studio, it is outputted in a .hex file format. This file needs to be transferred to the AxonII via the UART to USB conversion chip. There is a specific sequence to follow in order to program or "flash" the AxonII microcontroller which entails the use of a bootloader program. Further information on "flashing" can be found in appendix D.4.

5.4. PC Interface Software

Intended as the initial mode of recording or capturing the data, the data acquisition system prototype can be interfaced directly with a computer through USB cable. This requires both a program to be running on the AxonII and on Matlab PC software simultaneously.

The advantages of employing this mode are:

- Ease of troubleshooting and debugging
- On-screen feedback of functionality
- Immediate saving of data in Matlab structure format for later analysis and review.

The data measured by the sensors is collected and parsed by the AxonII as a series of continuous strings, each containing 15 comma-separated numbers, one for each measurement. For example, this appears in its raw form as:

230154.230,45.3757,-75.8533,5.1,349.2,80,339,321,404,245,257,255,786,0,10069

The corresponding interpreted results are shown in Table 17.

Table 17: Interpreted Raw Axon II Data String

Variable	**Value**	**Units**
GPS Time	230154.23	Seconds of GPS week
GPS Latitude	45.3757	Radians
GPS Longitude	-75.8533	Radians
GPS Speed	5.1	Knots
GPS Heading	349.2	Degrees
Compass Heading	357	Degrees
X-Axis Acceleration	339	ADC Steps (* / 1024)
Y-Axis Acceleration	321	ADC Steps (* / 1024)
Z-Axis Acceleration	404	ADC Steps (* / 1024)
X-Axis Rotational Rate	245	ADC Steps (* / 1024)
Y-Axis Rotational Rate	257	ADC Steps (* / 1024)
Z-Axis Rotational Rate	255	ADC Steps (* / 1024)
Wind Heading	786	ADC Steps (* / 1024)
Wind Speed	0	ADC Steps (* / 1024)
Interval	10069	Microseconds

Further interpretation of this data is carried out by a Matlab script running on the PC, the details of which are presented in section 5.4.2.

Outlined in the following sections are details on the programs running on both the AxonII microcontroller and the Matlab scrip which interfaces with it.

5.4.1. Axon II Programming

The code executed on the AxonII is fairly low-level and is executed in a continuous loop. A simplified sequence of instructions in the program is described in Table 18.

Table 18: AxonII Program Instructions for PC Interface

1. Begin timer
2. Output carriage return to USB
3. Read GPS data to obtain: latitude, longitude, speed and track from UART interface
a. Output latitude to USB as 8-digit floating point value
b. Output longitude to USB as 8-digit floating point value
c. Output speed to USB as 4-digit floating point value
d. Output GPS heading to USB as 3-digit floating point value
4. Read HMC6352 Compass sensor from I^2C interface
a. Output compass heading to USB as 3-digit floating point value
5. Read ADXL335 accelerometer as an ADC value
a. Output x-axis acceleration to USB as step count from ADC
b. Output y-axis acceleration to USB as step count from ADC
c. Output z-axis acceleration to USB as step count from ADC
6. Read LPR530AL gyroscope as an ADC value
a. Output x-axis rotational to USB rate as step count from ADC
b. Output y-axis rotational to USB rate as step count from ADC
7. Read LY530AL gyroscope as an ADC value
a. Output z-axis rotational to USB rate as step count from ADC

8. Read weather vane as an ADC value
a. Output weather vane orientation to USB as step count from ADC
9. Read anemometer as an ADC value
a. Output anemometer position to USB as step count from ADC
10. End timer
a. Output total time in microseconds needed to perform loop iteration to USB

As previously mentioned, these instructions are repeated continuously while the AxonII and its sensors are powered. It is continuously sending out measurements regardless of whether or not a PC is connected to the USB cable or recording measurements.

For further details, the complete C code is provided in Appendix B.1.1.

5.4.2. Matlab Interface

The Matlab script is the core of PC interface software; it is what the user of the prototype data acquisition system interacts with to manipulate, store and visualise data. A detailed copy of the Matlab script used is provided in Appendix B.1.2. The simplified sequence of instructions is given in Table 19.

The Matlab script also needs to account for infrequent but random errors where the AxonII fails to output a measurement line correctly. It needs to determine whether the measurement interval is salvageable or not and take appropriate actions to compensate. This is done by averaging the prior and next measurement.

Table 19: Matlab Script Instructions for PC Interface

1. Determine recording properties
1.1. Desired recording duration
1.2. Test number (1, 2, 3…)
1.3. Calculate number of measurement lines required
2. Define the parameters necessary to connect to the AxonII
2.1. COM port number

2.2. Baud rate
2.3. Buffer size
2.4. Line terminator (carriage return)
3. Pre-size working matrices for efficiency
3.1. Calculate size of matric based on desired length and pre-set data rate
4. Initiate connection and dump the two first lines to ensure proper synchronisation
5. Begin a loop which records the measurement
5.1. Read line
5.2. Ensure that line is free of errors
5.2.1. *If there is an error, note its position for correction later*
5.3. Record measurement line to working matrix
5.4. Display every i^{th} measurement for on-screen feedback
6. Terminate the connection
7. Add up measurement interval to create a time vector
8. Correct the erroneous measurements (if any)
8.1. Attempt to salvage last digit (measurement interval)
8.2. Take the average of previous and next measurements and replace in erroneous line
8.3. Account for the possibility of error lines at the very beginning or very end of all measurements
9. Re-assign data to a Matlab structure
9.1. Create a structure for this test, with its own index to allow for multiple test runs
9.2. Every variable measured has its own sub-structure for ease of access later

Further Matlab scripts are required to perform post-processing of the data. Some are simple modules to convert the accelerometer and gyroscope values into proper engineering units by multiplying the values with the correct conversion factors, as shown in section 5. This modular script format is desirable to permit experimentation with post-processing methods.

5.5. Standalone Interface Software

In order to maximise the portability of the inertial measurement data acquisition system prototype, it can be configured to eliminate the need for a PC to record the data during the actual testing. This is desirable, especially on rough seas where water ingress or shock can create problems with a traditional computer or laptop.

This standalone configuration removes the need for a PC by using Multi-Media Cards (MMC) or Secure Digital (SD) cards as the destination to which the Axon II sends measurement data to be recorded.

Data is written to the digital storage cards in the form of .txt files, which store each sequence of measurements as a line of comma-separated variables representing a specific measurement. This is similar to the format described in section 5.3. These lines are interpreted in the same way as presented in Table 17. Once one line representing a sequence of measurements is complete, the next sequence is stored on the next line of the .txt file.

Further interpretation is needed to convert the accelerometer, gyroscope, wind heading and wind speed into useable measurements. This is carried out during post-processing on a computer. The code for this conversion is given in appendix B.3.3.

5.5.1. Axon II Programming

The code executed on the AxonII is fairly low-level and executed in a continuous loop. A simplified sequence of instructions in the program is described in Table 20.

Table 20: AxonII Program Instructions for Standalone Interface

1. Detect and initialise SD card interface
a. Display error message if the card is not present, unformatted or corrupt
b. If no errors, display standby message to indicate readiness
2. Wait until proper input to begin recording
a. User must press button to begin
b. Display message to show recording has begun

3. Begin Timer
4. Output carriage return to USB
5. Read GPS data to obtain: latitude, longitude, speed and track from UART interface
a. Store latitude as 8-digit floating point value
b. Store longitude as 8-digit floating point value
c. Store speed as 4-digit floating point value
d. Store GPS heading as 3-digit floating point value
6. Read HMC6352 Compass sensor from I^2C interface
a. Store compass heading as 3-digit floating point value
7. Read ADXL335 accelerometer as an ADC value
a. Store x-axis acceleration as step count from ADC
b. Store y-axis acceleration as step count from ADC
c. Store z-axis acceleration as step count from ADC
8. Read LPR530AL gyroscope as an ADC value
a. Store x-axis rotational rate as step count from ADC
b. Store y-axis rotational rate as step count from ADC
9. Read LY530AL gyroscope as an ADC value
a. Store z-axis rotational rate as step count from ADC
10. Read weather vane as an ADC value
a. Store weather vane orientation as step count from ADC
11. Read anemometer as an ADC value
a. Store anemometer position as step count from ADC
12. End timer
a. Calculate total time in microseconds needed to perform loop iteration and store
13. Print all stored values to the SD card
a. Print all measurement values in the correct order
b. Verify that disk is not full, if so display error message and cease recording

For further details, the complete C code is provided in Appendix B.2.1.

5.5.2. PC Data Interface

The PC interface for the standalone mode is relatively simple; it does not need to do any monitoring at all during the test. It must simply be able to read the data from the storage media in the form of a .txt file and interpret accordingly.

On the hardware side, interfacing with SD or MMC cards is accomplished using a reader, such as those commonly found on laptop or portable computers or simply using an adapter.

For software interpretation of data, Matlab software incorporates a data importation wizard tool which can be configured to read comma-separated variables stored in .txt files.

Matlab software must then perform a few simple post-processing tasks to convert the data into useable units. There are three conversions to be made, first for the acceleration measurements, second for the rotational rate measurements and finally for the wind heading measurements.

The acceleration measurements are recorded in raw form as ADC steps as explained in section 5.2.2.1, and need to be transformed into conventional engineering units of acceleration, m/s^2. The conversion is done according to the following formula:

$$Acceleration\left[{}^{m}/_{s^2} \right] = ADC\,reading\left[Steps \right] \times Sensor\,resolution\left[\frac{{}^{m}/_{s^2}}{step} \right] \tag{5.6}$$

where the sensor resolution is given as 0.14513 m/s^2 per step. This conversion formula is applied to each measurement in turn.

The rotational rate measurements are also ADC steps in raw form, having no direct meaning as explained in section 5.2.2.2. These are transformed into a conventional engineering unit, degrees/second. The conversion is done according to the following formula:

$$Rotational\,rate\left[{}^{Degrees}/_{s} \right] = ADC\,reading\left[Steps \right] \times Sensor\,resolution\left[\frac{{}^{Degrees}/_{s}}{step} \right] \tag{5.7}$$

where the sensor resolution is given as 1.4663 Degrees/Second per step. This conversion formula is also applied to each measurement.

The wind heading measurement raw values are also meaningless without proper conversion. A Matlab script parses through the measured raw values and assigns a wind bearing according to the following equivalencies given in Table 21.

Table 21: Wind Heading Equivalencies

Direction (Degrees)	Resistance (KΩ)	Lower Threshold (ADC Steps)	Upper Threshold (ADC Steps)
0	33	744	806
22.5	6.57	346	432
45	8.2	432	530
67.5	0.891	73	87
90	1	87	108
112.5	0.688	0	73
135	2.2	154	213
157.5	1.41	108	154
180	3.9	266	346
202.5	3.14	213	266
225	16	615	667
247.5	14.12	530	615
270	120	914	1024
292.5	42.12	806	856
315	64.9	856	914
337.5	21.88	667	744

Each raw measurement situated between the defined lower and upper thresholds is equivalent to the direction specified in the table.

5.6. AxonII Code Optimisation

5.6.1. PC Interface

This particular mode lent itself best to programming and troubleshooting because it removed the complexity of coding the AxonII to write data to an SD card. As such, it was the first to be developed. Initially, a functional version of the AxonII program was constructed to test out the hardware's functionality and to validate the wiring connections.

The initial functional version was also used as the basis on which to construct the Matlab scripts which provide the interface between the AxonII's USB output and saving the recorded data to the PC's disc.

This initial code outputted the data at a relatively slow data rate because it used several Webbot libraries, such as those used for floating point calculations. The initial version also carried out several unit conversions directly on board the AxonII including:

- Accelerometers. The built-in Webbot libraries converted the as-read voltages into multiples of the gravitational constant (G's). This involved relatively complex floating point math operations.
- Gyroscopes. The built-in Webbot libraries converted the as-read voltages into degrees/second. This also involved relatively complex floating point math operations.
- GPS. The built-in Webbot libraries interpreted the standard NMEA message strings from the GPS into full-length floating point variables for all measured quantities.
- Wind direction. Conversion from the voltage into a heading was done directly using integer math operations.

In the quest to reduce the computational load on the AxonII, given its intrinsically lower capabilities in comparison to the PC, several iterations of this program were made. The end goal was to try to optimise the speed at which this particular program could run. Mainly, this involved moving all of the unit conversions from on-board the AxonII to the post-processing stage done on the PC with Matlab. Changes from the initial to the final version of the AxonII program presented were:

- Accelerometers. Removing the use of the built-in Webbot libraries which read the accelerometer output voltages and performed unit conversions. This was replaced by reading the value on the proper ADC channels, which removed several complex floating point operations and removed the use of a large portion of the Webbot library, thus reducing overall program size. The conversion from ADC values to metres/second^2 is now performed during post-processing.
- Gyroscopes. Removing the use of the built-in Webbot libraries which read the gyroscope output voltages and performed unit conversions. This was replaced by simply reading the value on the proper ADC channels. This change also removed several complex floating point operations and removed the use of a large portion of the Webbot library, reducing overall program size. The conversion from ADC values to degrees/second is now performed during post-processing.
- GPS. The use of the built-in Webbot libraries to interpret the standard GPS NMEA output messages was maintained. The implemented change was to reduce the number of digits used to store and communicate the GPS speed from 8 to 4 and for the GPS heading from 8 to 3. This change was validated by practical testing with the GPS: the dropped digits did not actually provide more precision as this is limited by the GPS specifications.
- Wind direction. The wind direction conversion from ADC value to heading is now performed during post-processing. This change removed a lot of the computational burden (specifically if statements) on the AxonII program.

5.6.2. Standalone Interface

The standalone interface was the next program to be developed after a thorough debugging of the PC interface program. The primary change here was to introduce the new SD card hardware. The focus of the optimisation done to this version of the AxonII program was primarily to ensure user-friendliness and robustness of the system. The initial version of this program simply started writing to disc as soon as the AxonII was powered on. This proved problematic when it was time to stop the recording as the AxonII was simply powered off thereby not closing the writing session correctly with an ensuing loss of data: this was highly undesirable.

It was finally found that the best solution was to allow the user to interact with the program in the way that is currently implemented in section 5.5.1: using the built-in push button and seven segment LED display. Some built-in redundancy was also implemented in the form of disc full or disc error checking.

5.7. Packaging

In order to primarily ensure the waterproofing of the data acquisition prototype, all electronics and internal components were mounted in a waterproof Pelican® 1120 case, seen in Figure 31. This case is rated fully waterproof to 30 metres, incorporates a pressure relief valve, neoprene seal and double-latched cover. Pelican® cases have a reputation as solid containers and are typically used by photographers to protect expensive camera equipment from shock, dust, moisture or water. Physical internal dimensions of the Pelican® 1120 are 184 x 121 x 78 mm [94].

The rigidity of the Pelican® case also provides multiple ways in which to mount the case to the test vessel. It is strong enough to be clamped into position, can be temporarily fixed using adhesive gaffer's tape or physically tied to a mounting point using plastic cable ties. Having flat sides also facilitates alignment of the accelerometer, gyroscope and compass axes with the movement axes of the sailboat.

Inside the 1120 case are mounted the accelerometers, gyroscopes, digital compass, GPS receiver, SD card interface, Axon II microcontroller and battery. The sensors are all mounted perpendicular to the external surfaces of the case. Mounting is accomplished by adhesive Velcro™ strips and double sided adhesive tape to hold the on/off switch in position. Details on the mounting arrangements can be seen in Figure 32.

External components are the waterproof GPS antenna, wind instrumentation and optional USB cable. This necessitates two or three external wires; one is for the GPS antenna, the other to connect the wind instrumentation and the optional USB cable. Physically, this meant drilling a small hole near the corner of the Pelican® 1120 case to allow the wires to pass through. Re-sealing the hole is accomplished by silicone caulk. This undoubtedly compromises the water

tightness and the ability to submerge the case, however sufficient protection remains against shock and water ingress.

Figure 31: Pelican® 1120 Case

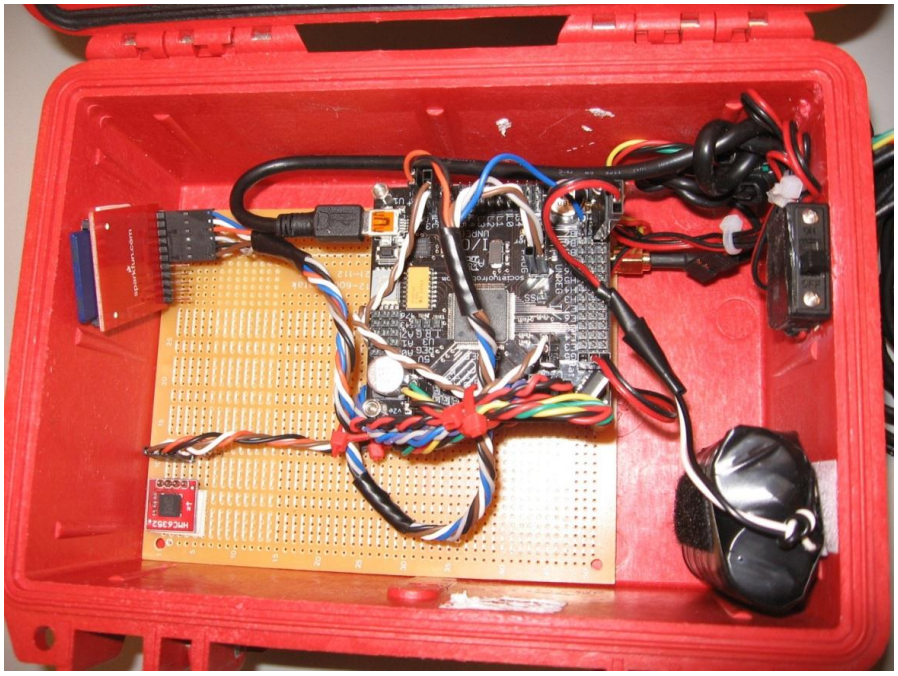


Figure 32: Internal Packaging Layout

The connection between the GPS receiver and antenna cannot be disconnected easily; this is not a hindrance since the antenna cable length is sufficiently long to allow positioning of the antenna away from the Pelican® case to a suitable location for good signal reception. Similarly, the optional USB cable needs only to find its way to a laptop PC, which is never too far from the data acquisition prototype. Thus, it is not removable.

Figure 33: Deutsch DT Series Connectors [95]

However, the wind instrumentation does need to be positioned somewhere appropriate, which implies that the cabling is long and too cumbersome to stay permanently attached to the Pelican® case. In order to make this cable removable and still maintain waterproofness, a 4-prong Deutsch® DT series connector is used, as shown in Figure 33.

5.8. Batteries

The AxonII based system can accommodate a supply voltage from 5.35V to 16V [83] however the maximum recommended voltage is between 6V and 7.2V for use with servo motors. Typical for use in small robotics projects is a two-cell 7.4V lithium-polymer 100mAh battery pack, for example as shown in Figure 34, this type of battery pack retails for 7.00 to 10.00 $US [96] [97].

Figure 34: Typical 7.4V 100mAh Lithium-Polymer Battery Pack [96]

The expected energy consumption of the data acquisition prototype can be estimated using the known power consumption of all devices used. The expected power draw available for some devices is shown in Table 22, compiled from each device's datasheet found in Appendix A -.

Table 22: Expected Device Power Draws

Device	Power Consumption (mA)
ADXL335 Accelerometer	10
LY530ALH Gyroscope	5.5
LPR530AL Gyroscope	6.8
HMC6352 Compass	10
Venus 634LPx GPS	28
Total	**60.3**

As a very rough estimation of the estimated battery life using a 7.2V 1000mAh battery is shown in equation (5.8) [6], using the following assumptions:

- Microcontroller power requirements are approximately 40mA including what is necessary to operate the wind instrumentation and SD card functionality
- System efficiency is 100%

$$Battery\,life\left[hours\right] = \frac{Capacity\,Rating\,of\;Battery\left[milliAmperes - hours\right]}{Current\,consumption\,of\;the\,device\left[milliAmperes\right]} \tag{5.8}$$

$$= \frac{1000\ [milliAmperes - hours]}{60 + 40\ [milliAmperes]}$$

The estimated battery life is thus approximately 10 hours for a 1000mAh 7.2V battery.

In application, the battery packs used are 7.2V 3-cell 1800mAh, which should provide ample power reserves and ensure sufficient battery life, as per the design requirements outlined in section 4.1.1.

A potential concern regarding the use of any batteries is their ability to function in cold weather. Recreational sailing is generally a fair-weather sport and even the most die-hard enthusiasts rarely take out their boats when the temperature is below 5°C. Because of this, the battery pack is not expected to function in temperatures lower than 5°C.

Chapter 6 Validation

The prototype data acquisition system was submitted to several tests designed to be implemented incrementally so as to establish full functionality of the prototype. Initial tests are simple bench tests carried out in the laboratory, followed by ground vehicle tests which introduce motion and finally actual measurement in a sailing yacht.

6.1. Bench Testing

Bench testing is the first portion of validating the prototype which is carried out after construction and assembly.

6.1.1. Justification

This first phase of testing is the most logical in terms of assessing basic functionality of the prototype, serving as a crude verification of instrument calibrations and orientations. Bench testing answers the following elementary questions:

- Do all of the electronics function as they are intended?
- Do the measurements output correctly?
- What is the actual battery life?
- Are the orientations of the sensor axes correct?

6.1.2. Sensor Orientation Verification

The assembled box was placed statically on all six faces such that each of the three orthogonal accelerometer axes was subjected to gravity in both positive and negative directions, as shown in Figure 35. Although this test was repeated for up to several hours in each orientation, only a small portion of the results is presented here to illustrate the accelerometer's typical behaviour.

Figure 35: Orthogonal Orientations

As an illustrative example, the case where the X-axis is positive towards gravity is used. Figure 36 shows the X-axis accelerometer measurements over a period of approximately 5 seconds.

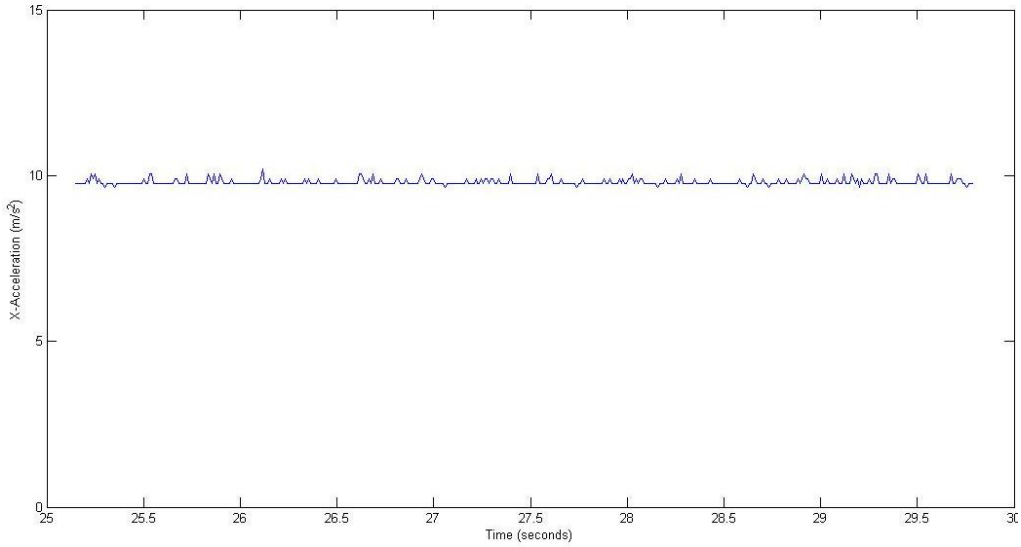


Figure 36: X-Axis Acceleration during Bench Testing

The average value of X-axis acceleration over this period of time is 9.8103 m/s^2 compared to an expected value of 9.80665 m/s^2 [34].

Figure 38 shows the values of the Y and Z axes during the same test, over the same period of time.

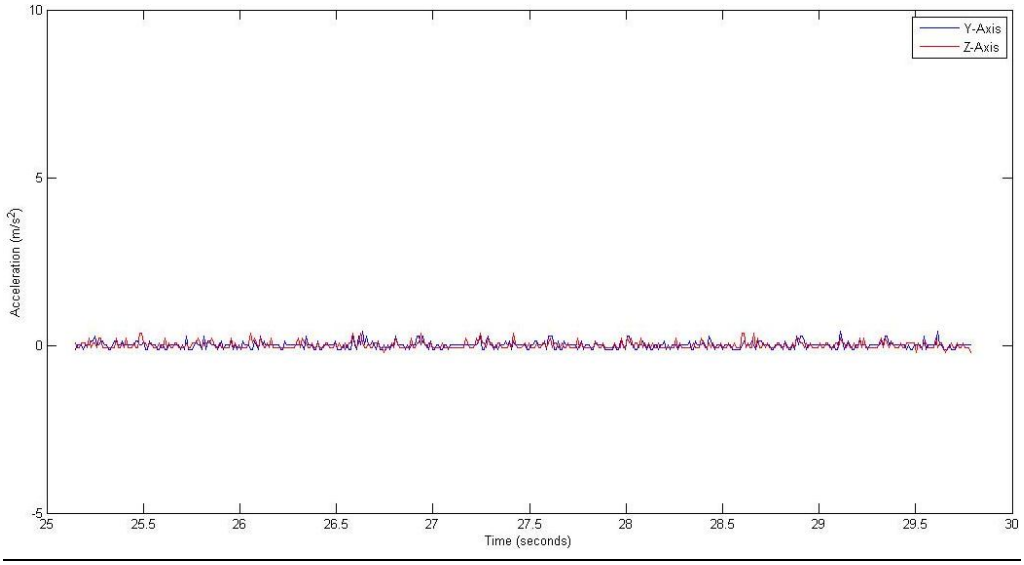


Figure 37: Y, Z-Axis Accelerations during Bench Testing

The expected values for both Y and Z axes are both zero whereas the average values for Y and Z are 3.2168×10^{-4} m/s^2 and -3.3836×10^{-4} m/s^2 respectively. These values are within the expected tolerance.

A summary of the measured accelerations is given in Table 23.

Table 23: Average Measured Accelerations during Static Testing

Axis	**X**	**Y**	**X**
Average Measured Value (m/s^2)	9.810	0.000322	-0.000338
Standard Deviation (m/s^2)	0.0869	0.115	0.110

The average measured values are in agreement with their respective expected values, standard deviations illustrate the inherent noise in the sensor.

Figure 38 shows the values of the rotational rates, as measured by the gyroscopes, in all axes during the same time period as previously used.

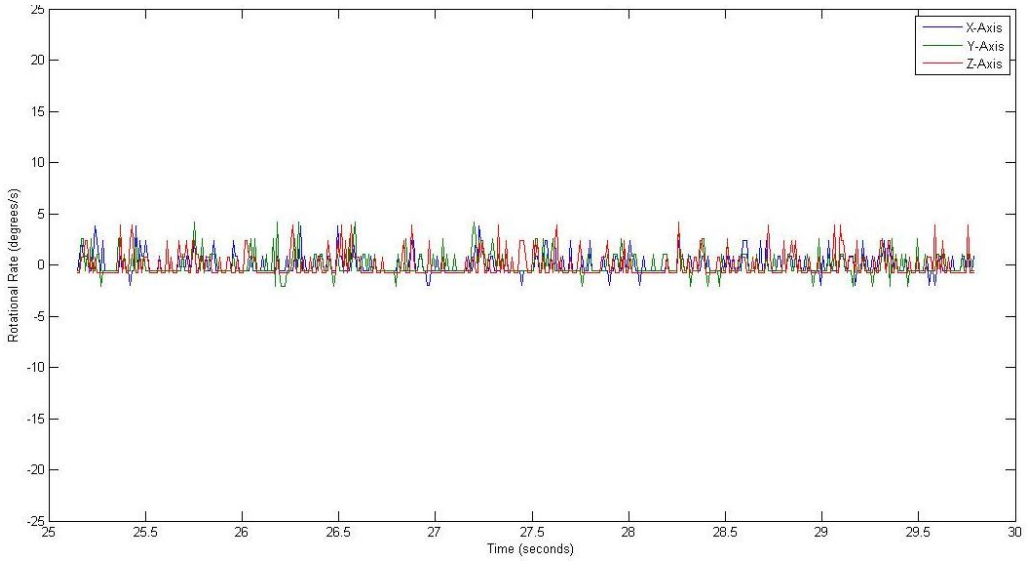


Figure 38: Rotational Rates during Bench Testing

Since this is a static test, the expected value of the rotational rates is zero. Measured averages and standard deviations are given in Table 24.

Table 24: Average Measured Rotational Rates during Static Testing

Axis	**X**	**Y**	**X**
Average Measured Value (degrees/s)	0.00017	-0.000079	-0.000413
Standard Deviation (degrees/s)	1.02	1.13	1.22

The measured gyroscope averages are within the expected tolerance to the expected value of zero but due to the inherent noise, have a relatively large standard deviation.

6.1.3. GPS Reception and Position

Powering up the unit with the GPS antenna positioned with a clear view of the sky illustrates the time needed to acquire a GPS signal.

As a secondary objective of this test, the position reading as given by the GPS can be compared to the known position of the antenna as referenced on a map.

6.1.4. Compass Heading

To test the compass heading returned by the HMC 6352 digital compass, the prototype data acquisition was made to perform a full circle on a horizontal plane. Sensor readouts for this test are shown in Figure 39.

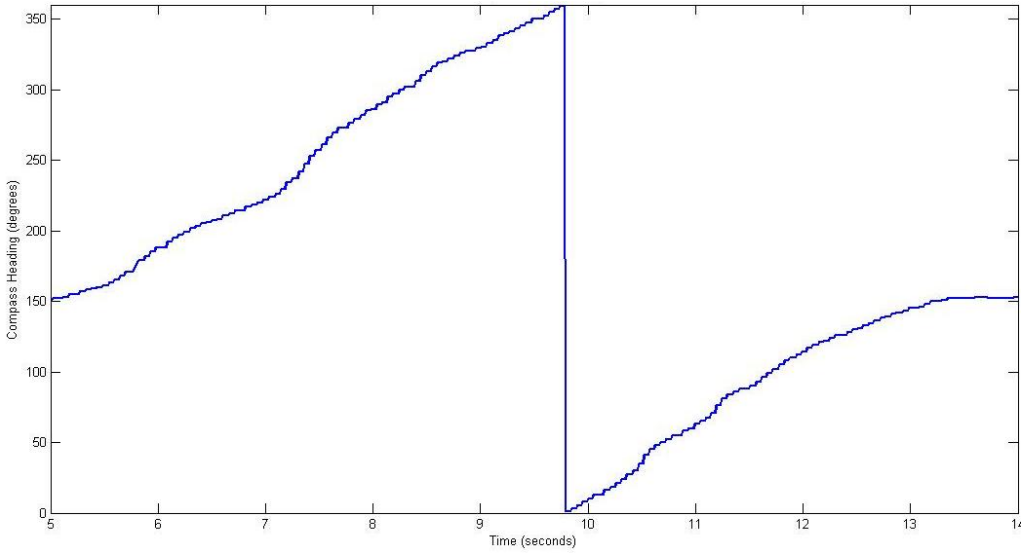


Figure 39: Compass Heading during Bench Testing

This first test demonstrates that the compass can measure through the full 0-360 degree range. This also verifies that the compass is not subjected to significant magnetic interference which would skew the measurement results and prevent the compass from fully cycling through its full range.

The next portion of the test was to compare the compass reading when the data acquisition system is oriented against a known bearing. Random headings were chosen and marked out on the test surface using a traditional needle compass shown in Figure 40. These marks were then re-measured using the HMC 6352 digital compass.

Figure 40: Suunto A-10 Needle Compass [98]

In all test cases, the HMC 6352 digital compass sensor accurately displayed identical results to the Suunto A-10 needle compass.

6.1.5. Battery Life

This portion of the test was to simply measure the time taken for a fully charged battery to lose sufficient charge as to no longer be able to power the data acquisition system. Practically, the unit was left powered on and the status of the data output verified every fifteen minutes. This test was repeated a total of three times for a maximal duration of 555 minutes or 9 hours, 15 minutes. The results of this test are condensed in Table 25.

Table 25: Battery Life Test Results

	Battery Duration
Test 1	540 minutes
Test 2	> 555 minutes
Test 3	> 555 minutes

The measured battery life for the data acquisition prototype is at least nine hours.

6.1.6. Wind Speed

In order to accurately verify the wind speed, the anemometer was mounted against a constant wind source in the form of a desktop oscillating fan, as shown in Figure 41.

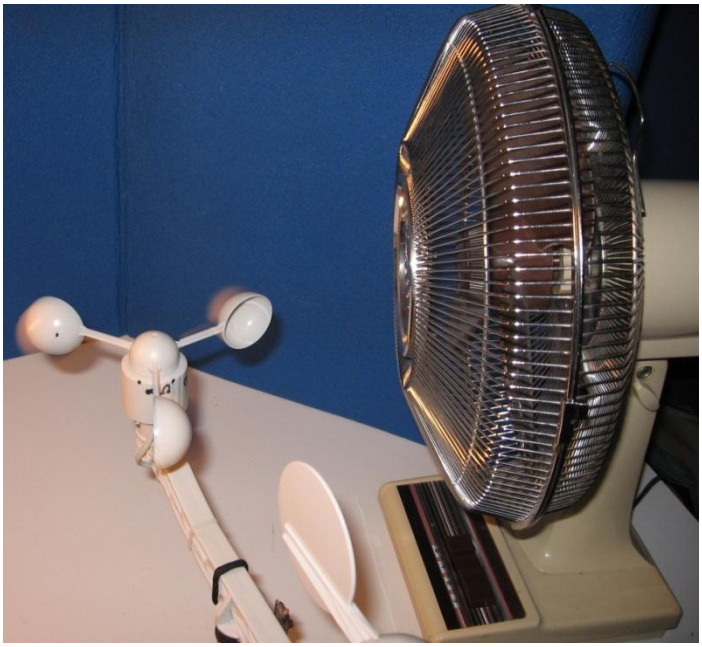


Figure 41: Wind Speed Test Setup

By using the fan's controls, low and fast speeds were used as two different test cases, the intermediate speed producing little noticeable change in wind speed. In order to obtain comparative values, a separate wind instrument was used to obtain independent speed values. The SpeedTech™ WindMate® WM-200 portable anemometer, shown in Figure 42 was used to measure the fan-produced wind speeds. This particular anemometer has a claimed accuracy of +/- 3% [99].

Figure 42: WindMate WM-200 Portable Anemometer [99]

The wind speeds, as measured by the prototype data acquisition system for each test case, are given in Figure 43. Test 1 is the low fan speed and test 2 is the high fan speed.

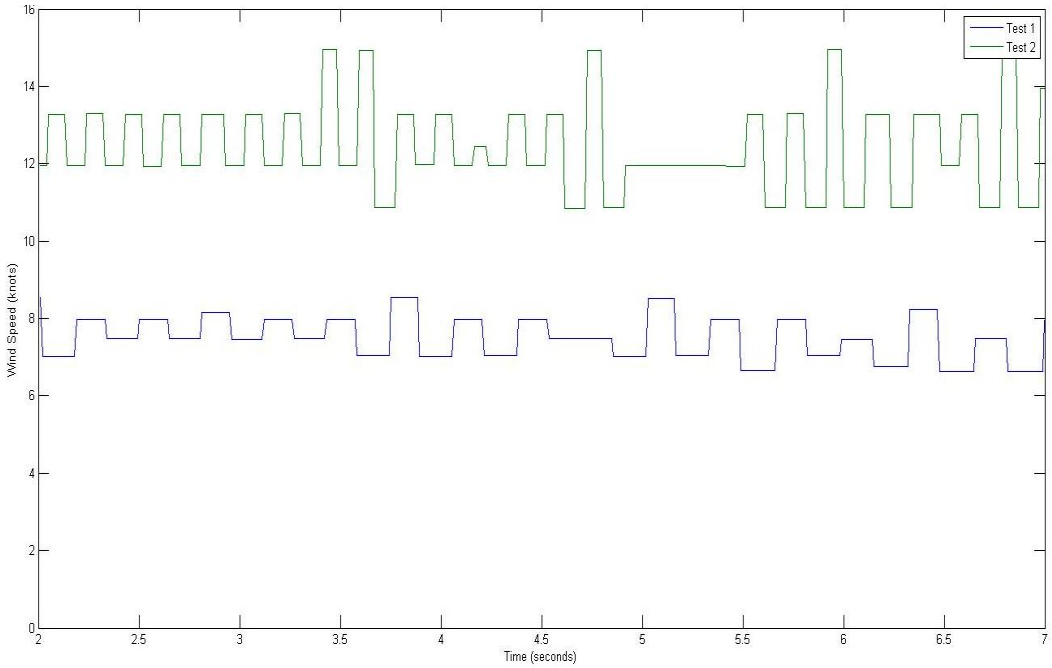


Figure 43: Wind Speed Test Results

It can be seen that the prototype-measured wind speeds are step-like in nature: this is due to the averaging of the values by Matlab post-processing, as described in appendix B.3.3.

A comparison of results of both the portable and prototype's anemometers is shown in Table 26. All measured wind speeds are averaged over a period of 5 seconds.

Table 26: Wind Speed Test Results

	Test 1	**Test 2**
WM-200 Wind Speed (knots)	7.9	13.1
Anemometer Average Wind Speed (knots)	7.5	12.4
Anemometer Standard Deviation (knots)	0.54	1.15
% Difference	**5.25**	**5.39**

The prototype anemometer is consistently within 5.5% of the WM-200 wind speeds and underestimates the wind speed by this amount.

6.1.7. Wind Direction

Validating the wind direction is accomplished by rotating the weather vane through all possible headings, as given in Table 16. This is done by rapidly spinning the weather vane and allowing it to rotate freely, artificially producing motion. Resulting wind heading measurements for one such spinning test are given in Figure 44.

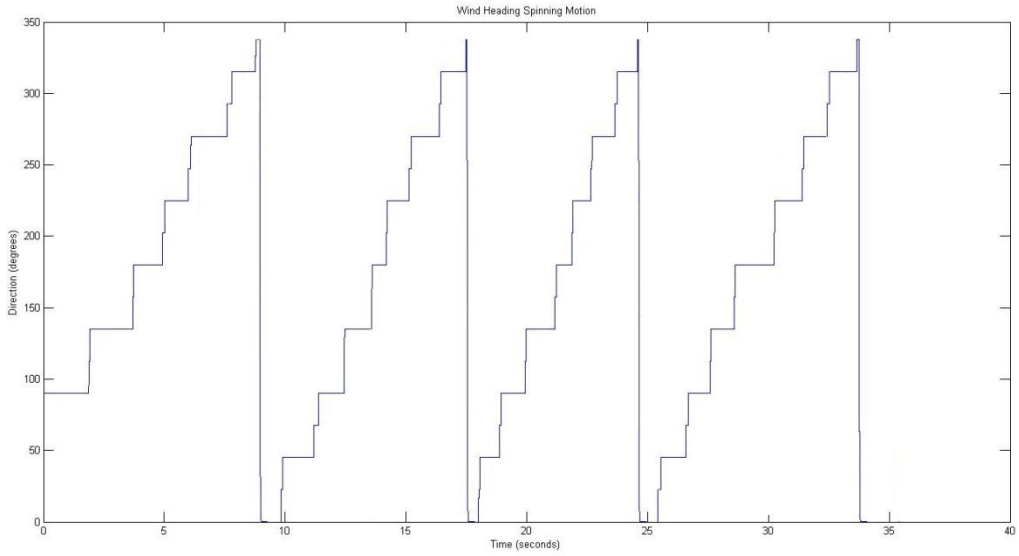


Figure 44: Wind Heading, Vane Spinning

The measured wind heading shows an increasing heading over time, which peaks at 337.5 degrees before returning to zero. This process is periodical because of the spinning motion of the weather vane.

Some of the 22.5 degree interval steps are short, for example the 202.5 degree step has a short duration. Although the rotational rate of the weather vane in this test is fairly constant, two factors contribute to short steps. First is the rapid spinning motion and second, lower quality wind instruments are not evenly divided across a 360 degree range. For example, the weather vane registers 202.5 degrees over a much smaller period of time than it does for its neighbouring steps of 180 and 225 degrees.

6.1.8. Data Rate

While in PC (USB) interface mode, the prototype data acquisition system is left to record data over a period of one hour with the purpose of examining the length of the measurement interval. This is the time required for the AxonII to record, process and output each measurement as measured by the microcontroller`s internal clock.

The average measurement interval for this case is found to be 0.01031 seconds with a standard deviation of 0.000215 seconds. This equates to measurement frequency of 97.08 Hz for the PC interface mode.

In standalone (SD card) interface mode, the same test was repeated to examine the measurement interval. This is measured in the same manner as for the PC interface mode. The average measurement interval for the standalone mode is found to be 0.01028 seconds with a standard deviation of 0.00354 seconds. This equates to a measurement frequency of 97.22 Hz.

The standard deviation is one full order of magnitude greater for the standalone interface mode than for the PC interface mode because of the SD card itself. Data can only be written to the SD card in 512 byte blocks [53] and must be stored in an internal buffer until sufficient bytes are accumulated for one writing cycle. This means that not every measurement interval includes an actual write operation to the SD card action and this affects the duration of that particular measurement interval, thereby accounting for the greater standard deviation.

Also of relevance to the standalone (SD card) interface mode is the file-size increase rate, or how much information in megabytes is being added to the storage medium over a period of time. Experimentally, this figure was calculated by dividing the size of the measurement data file by the duration of the measurements. It was found that the file size increased by 21.34 Mb/hour.

A typical 1 GB (1024 Mb) capacity SD card could therefore theoretically store approximately 48 hours of measured test data.

6.1.9. Bench Testing Summary

Bench testing confirmed the proper functionality of the prototype, such as basic instrument calibration and data output. The electronics and wiring showed no unexpected issues. Each of the

sensor orientations were as expected and static measurements read correctly. The battery life was found to exceed the basic requirement of 6 hours, as outlined in section 4.149, by 3 hours. Finally, the measured data rate, for the standalone mode and the USB mode, was approximately 97Hz.

6.2. In-Car Testing

The in-car testing is motivated by the need to subject the data acquisition system to motion which is more controllable and repeatable than is possible on a sailing yacht, so that repeatability of measurements may be established.

6.2.1. Justification

The automobile is a convenient way to subject the prototype data acquisition system to a moving environment. The chosen track or path is easily repeated since conditions do not change and the road does not move. The main advantage of following a single path along a road is that it makes it relatively easy to subject the prototype to repeatable test conditions.

However, as a downside in-car testing does not provide all of the expected conditions to which a sailing yacht is subjected:

- A road is much smoother than wind-swept water, thus the oscillation due to wave effects is not present
- Automobiles do not heel (roll) or pitch to the same degree as boats. Rolling and pitching contribute to passenger discomfort and are therefore undesirable in automobiles [73].
- The typical legal and practical speeds of the automobile in Canada are at least a full order of magnitude greater than those an average sailing yacht typically experiences. Whereas a typical sailing yacht operates in a range of 0 to 10 km/h, most cars operate at speeds of 0 to 100 km/h.

As such, the reasoning behind the in-car testing is chiefly to subject the data acquisition system prototype to a moving environment and verify that the results yield sensible and most importantly, repeatable results.

6.2.2. Challenges

The primary challenge in designing and implementing this test was to find a location which is suitably quiet and easily accessible while possessing the desired geometry. The minimal presence of vehicular or pedestrian traffic was critical in order to minimise error sources and facilitate repeatability. Multiple other potential locations were scouted before finally settling on the chosen test circuit.

The second challenge is finding a driver who is capable of performing multiple test runs in a very repeatable manner: typically in automobiles this is done using professional drivers who are highly skilled in their work and intimately familiar with the test track. Five different test drivers were assessed before deciding on one. The assessment was based on several factors including:

- Visual confirmation by the author of the vehicle speeds at known locations during manoeuvres
- Smoothness at the car control inputs (steering and braking principally)

The other critical element in assessing potential test driver`s performance was completing several timed runs around the circuit. The selected driver was able to complete over a dozen circuits and maintain the duration of the test runs within 5 seconds of each other.

6.2.3. Test Circuit and Procedure

The chosen test circuit displays several traits in order to better approximate the behaviour of a sailboat. Firstly, the test circuit is chosen so as to maintain relatively low speeds, on average below 40 km/h, which is the enforced speed limit in residential areas in the testing location. Secondly, the chosen track has several sections of easily repeatable manoeuvres, such as straight motion, right angle turns, sweeping turns and circular motion. Also, the track contains a constant speed circle in the form of several full loops in a dead-end.

The actual chosen track is shown in Figure 45. This location also proved convenient due to its relatively low traffic and proximity to the laboratory.

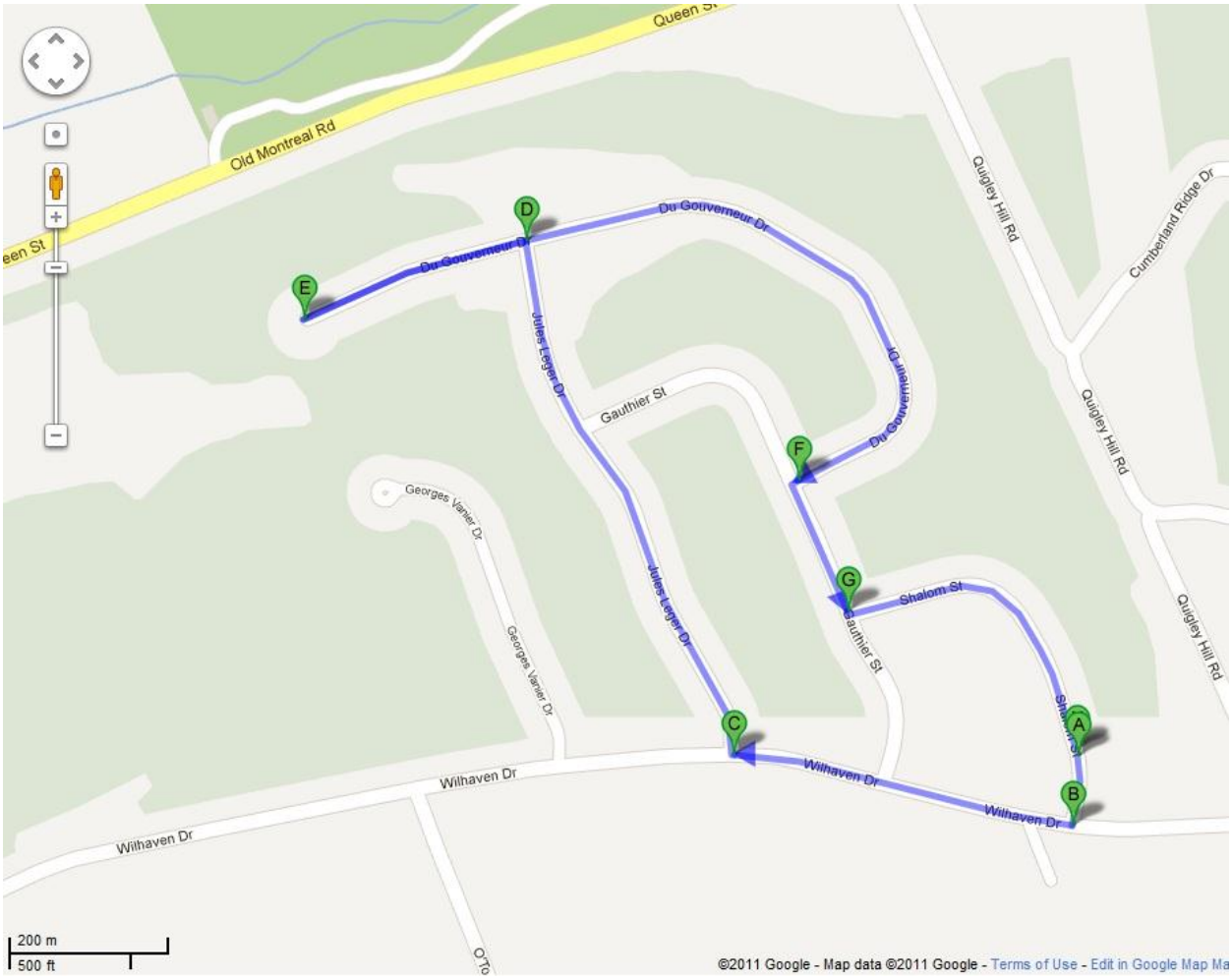


Figure 45: In-Car Testing Track

- The test runs begin at point "A", where the vehicle is stationary. The vehicle then accelerates towards point "B" where it must come to a complete stop.
- At point "B", the vehicle accelerates while performing a ninety degree right hand turn. After the turn is completed, the vehicle accelerates to a constant speed of approximately 60 km/h. Once sufficient distance has been covered, the driver decelerates in preparation of another right angle, right hand corner at point "C".
- At point "C", the vehicles proceeds directly to point "D" at a maximum velocity of 40 km/h, the maximum permissible legal limit. The driver attempts to maintain as constant

of a velocity as possible while accounting for road conditions. Point "D" requires the driver to slow down in preparation for a left hand square corner.

- At point "D", the vehicle performs a left hand turn then accelerates to point "E" which is located at a dead end. At this point, the driver slows down to a comfortable speed and proceeds to complete two full loops around the dead end, while trying to maintain a constant speed and steering angle.
- From point "E" to point "F", the driver simply follows the road as it bends and curves. There is a gentle right hand sweeping curve of constant radius when approaching point "F". After the previously mentioned curve, the car must come to a full stop at point "F".
- From Point "F", the car accelerates while performing a left hand ninety degree turn. Once the car is pointing in the right direction, the driver accelerates to a maximum speed of 40 km/h. After, the driver decelerates in preparation for a left hand turn at point "G".
- At point "G", the driver performs a left hand ninety degree turn and proceeds to come to a full stop once the car has returned to point "A".

In order to minimise environmental factors and other conditions which might affect test results, test runs were all carried out back to back, by the same driver, on the same day. Several testing days were performed.

For the sake of clarity, a series of three test runs made on the same day will be employed for comparative purposes. For analysis, the test circuit was divided into 12 distinct events as shown in Figure 46.

Due to the nature of the vehicle's movement, the measured data shows trends primarily in two dimensions. The car does not exhibit much tendency to roll or pitch during low speed motion and the terrain of the test circuit is primarily flat due to very little topographic deviations.

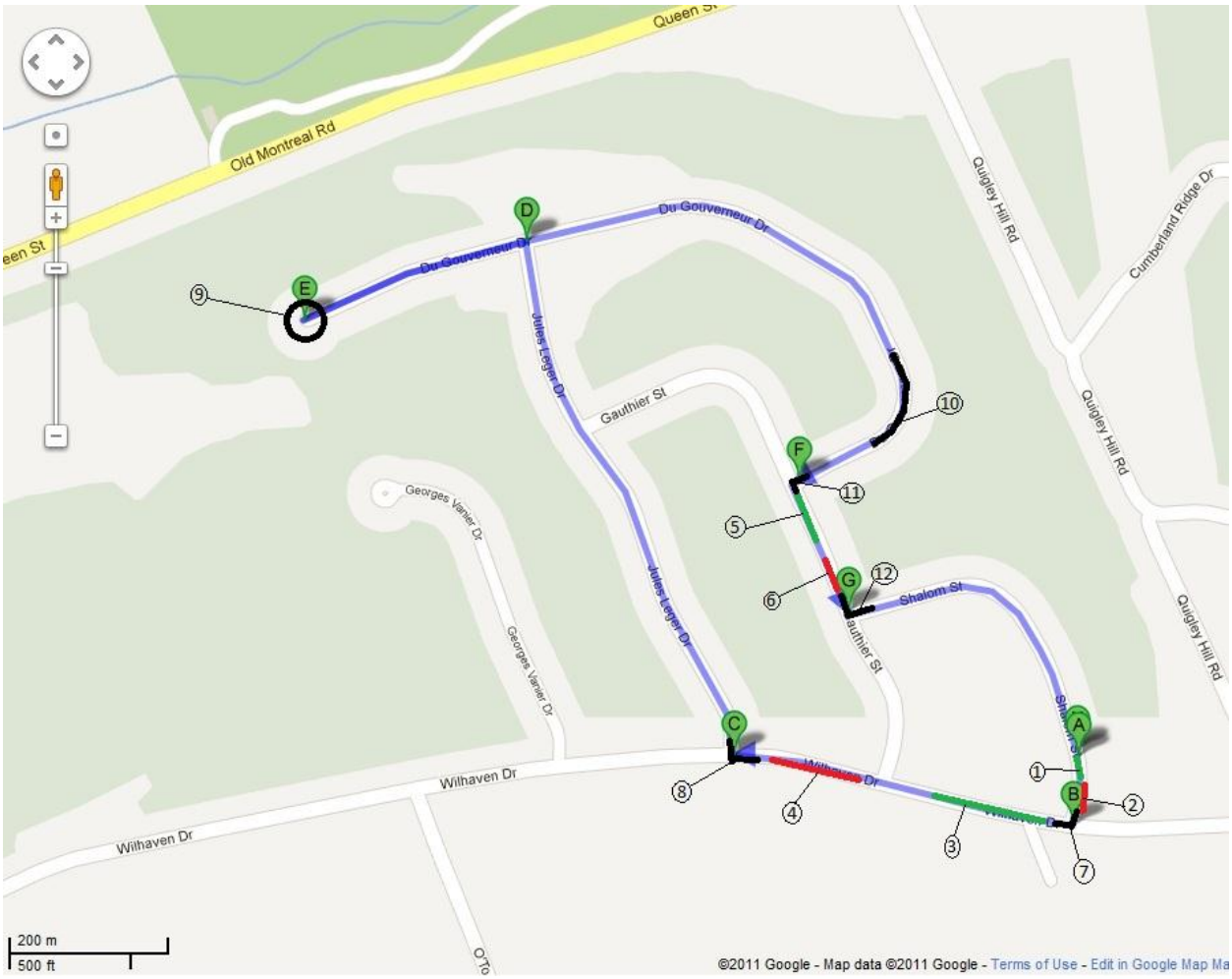


Figure 46: Test Circuit Event Locations

The test circuit events are further divided in two main categories, events which affect primarily fore-aft acceleration (X-Axis) and events which affect side-to-side acceleration and yaw rates (Y-Axis acceleration and Z-Axis yaw rate). These events are summarised in Table 27.

Table 27: In-Car Test Circuit Description

Event	Description	Acceleration Axes	Rotational Rate Axes
1	Accelerate after A	X	N/A
2	Decelerate before turn B	X	N/A
3	Accelerate after turn B	X	N/A
4	Decelerate before turn C	X	N/A
5	Accelerate after turn F	X	N/A
6	Decelerate before turn G	X	N/A
7	Right hand turn at B	Y	Z
8	Right hand turn at C	Y	Z
9	Full counter-clockwise circle at E	Y	Z
10	Sweeping left hand turn before F	Y	Z
11	Left hand turn at F	Y	Z
12	Left hand turn at G	Y	Z

Although numerous test runs were carried out the results of three runs are presented here.

For later calculations, it is necessary to obtain an estimate of the radius of the turn located at event 10. This was achieved through geometric measurements carried out in Google Earth™, as shown in Figure 47.

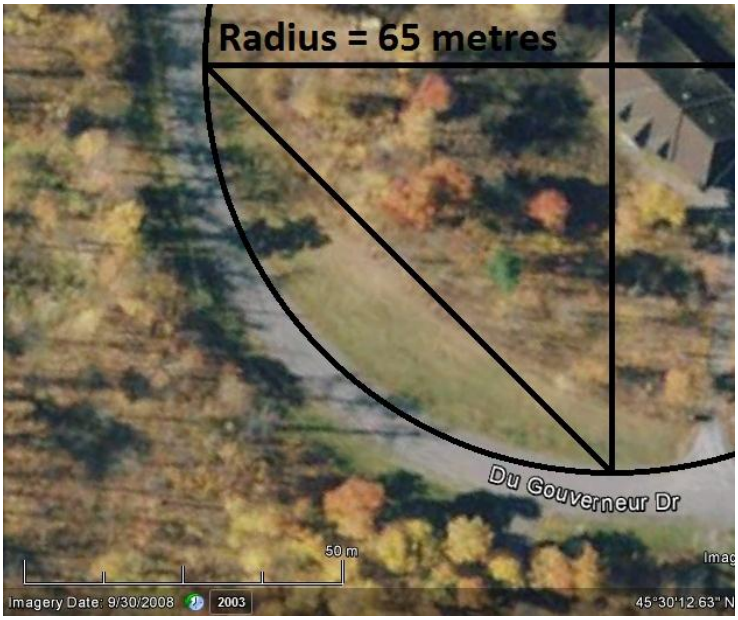


Figure 47: Event 10 Turn Radius Estimation

The turn curvature radius for event 10 is estimated to be approximately 65 metres.

6.2.4. GPS Position

The recorded GPS readings of latitude and longitude are shown in Figure 48 to form a trace of the vehicle's track. Three tests, labelled 1, 2 and 3 are displayed using the colours blue, green and red. It is quite apparent that the GPS position data shows little deviation from the expected uniform and smooth shape of the test circuit.

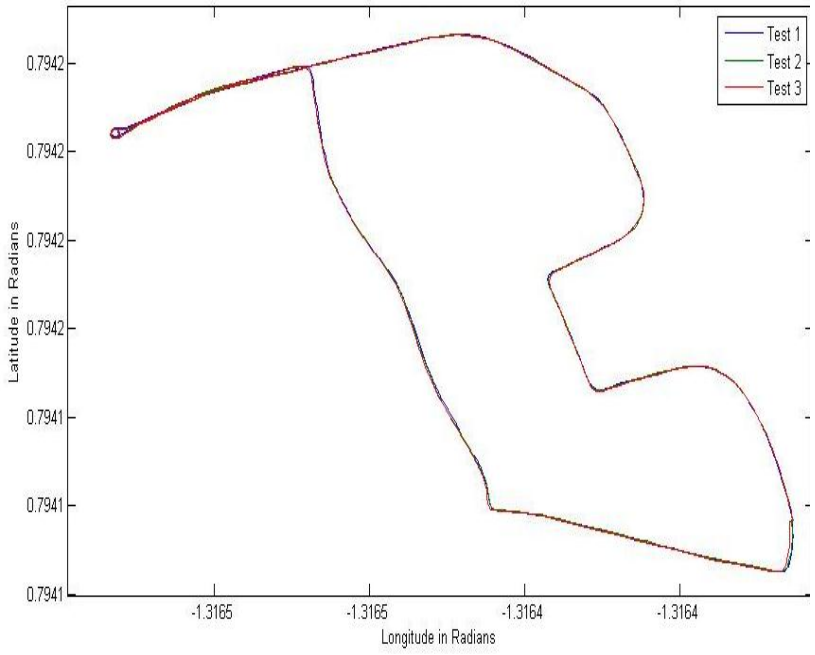


Figure 48: In-Car Testing GPS Track Plot

The recorded GPS position plot is overlaid on the test circuit map in Figure 49. In order to make this possible, the size of the GPS position plot needed to be adjusted to the same scale as that of the map, but no other modification was necessary.

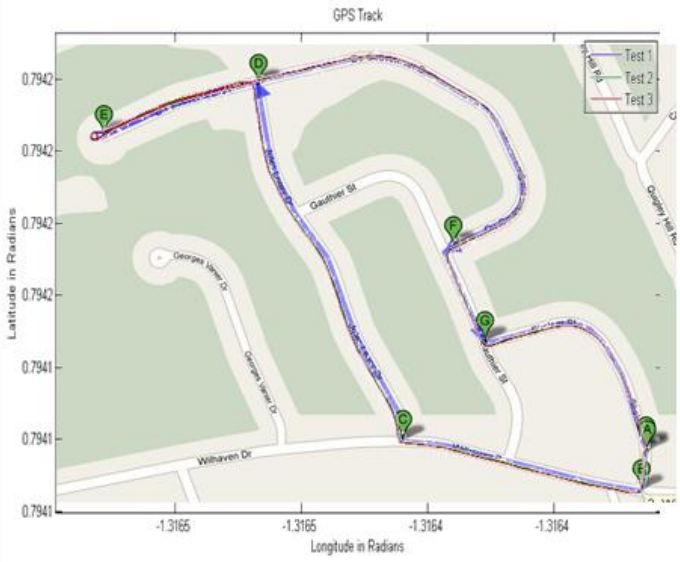


Figure 49: In-Car Testing GPS Plot Overlaid on Map

From the figure, if can be seen that the GPS track correlates quite highly with the test circuit map, deviating little from where the confines of the road are illustrated on the map.

6.2.5. GPS Speed

The GPS speed measurement, provided by the Venus 634LPx, has a manufacturer specified accuracy of 0.1m/s as shown in Appendix A.6. The GPS speed plotted against time is shown in Figure 50 for the three chosen test runs.

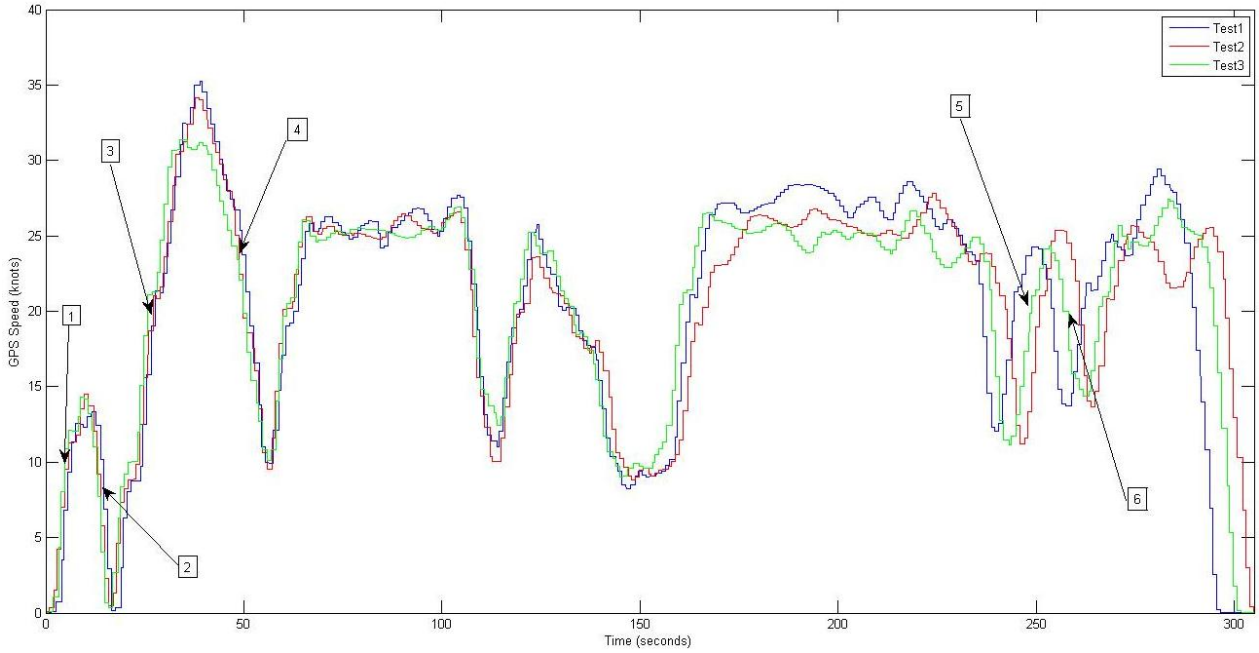


Figure 50: In-Car Testing GPS Speed

Labelled on Figure 50 are approximate occurrences for events 1 through 6 where the accelerations and decelerations take place. Each test run shows speed peak speeds within the expected range of the permitted speed limits where applicable, for example 32 knots is approximately 60km/h, which is the road speed limit between points B and C as shown on Figure 45. The car travels between points B and C approximately between 15 and 60 seconds.

The overall shape of the speed traces remains fairly constant between all tests. Any differences can be justified by variations of the human driver's performance - the duration of the test varies by several seconds. This is because the driver does not always accelerate or brake neither at the

same rate nor at exactly the same points along the test circuit. When overlaid as in Figure 50, these differences in GPS speed traces become more and more apparent as time progresses.

6.2.6. Accelerometer

Due to the relatively flat nature of the test course and the fact that the automobile exhibited little roll during the test session, meaningful acceleration data was only recorded on the fore-aft and lateral axes, X and Y respectively. Shown in Figure 51 and Figure 52 are the acceleration measurements over time for all three chosen test runs in both the X and Y axes.

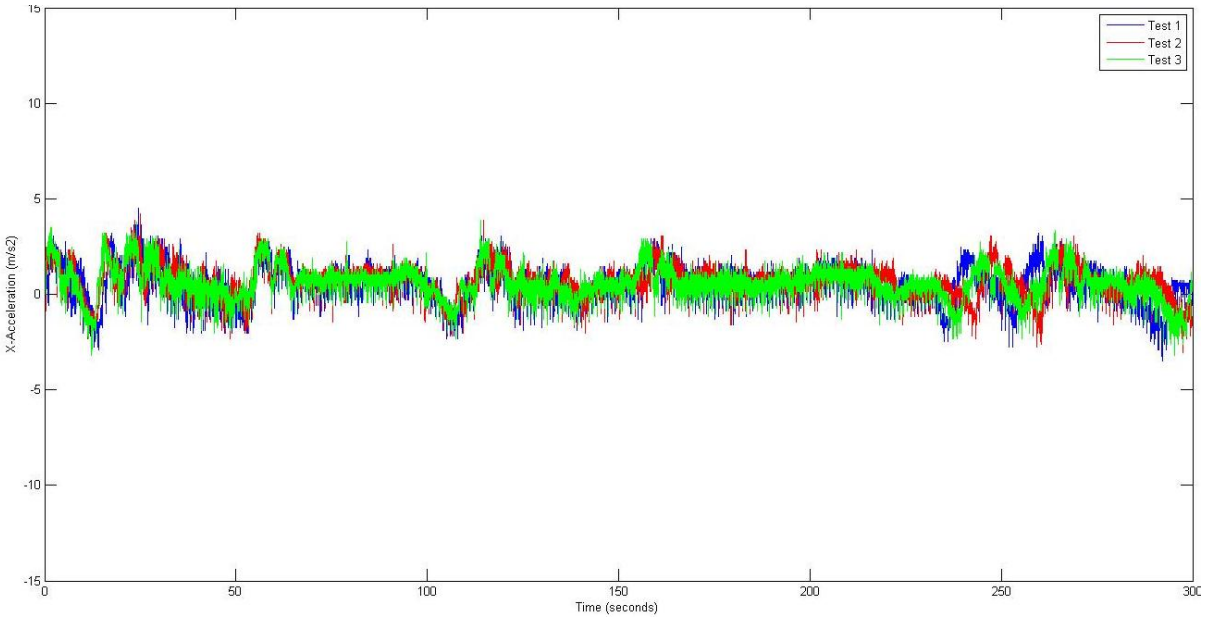


Figure 51: In-Car Testing X-Acceleration

Figure 51 shows the X-axis acceleration, scaled to +/- 15m/s^2 or half of the accelerometer's possible measurement range. It can be seen that the signal measured rarely exceeds +/- 3m/s^2, which correlates with the gentle acceleration provided by the driver. Visually, the repeatability of the X-axis accelerometer measurement is good, with similarly shaped peaks that correlate with known braking and acceleration points along the test circuit.

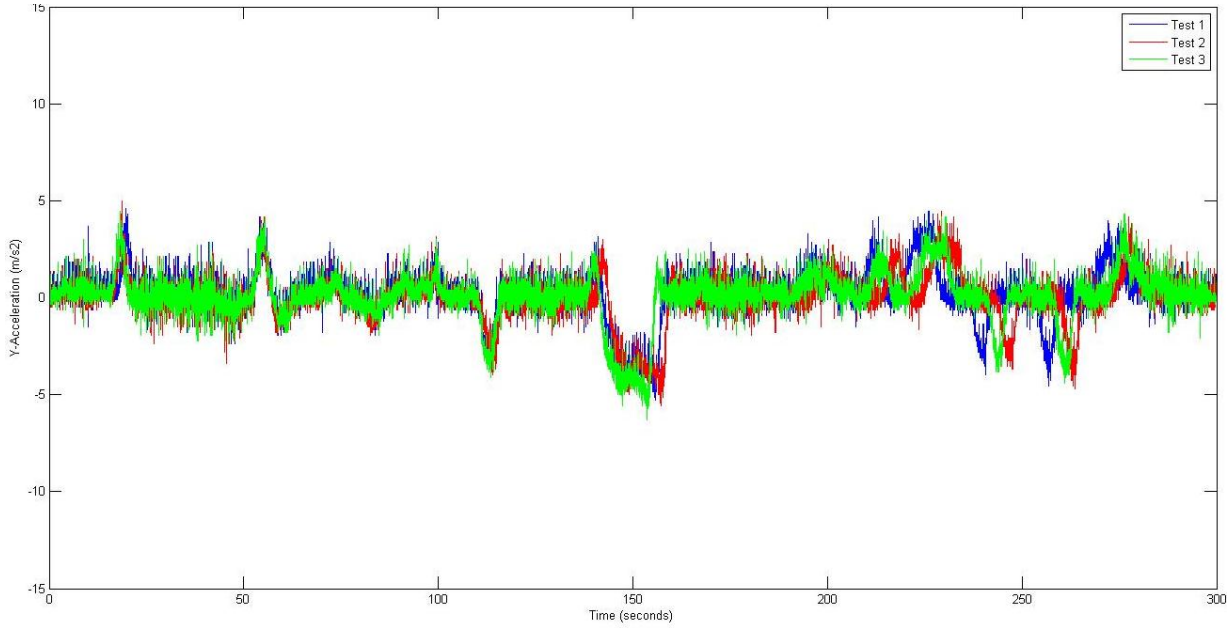


Figure 52: In-Car Testing Y-Acceleration

Figure 52 shows the Y-axis acceleration, also scaled to +/- 15m/s^2. Positive values imply acceleration towards the left hand side of the driver and negative values towards the right hand side of the driver. Typical measured acceleration during manoeuvres is less than 5m/s^2, which is considered typical for everyday driving [73].

Again, by visual inspection, the repeatability of the Y-axis accelerometer measurement is comparable to that observed in the X-axis measurements. The difference in the occurrence of peaks as time progresses is consistent with the difference in test run durations described in section 6.2.5.

In order to try to better quantify the accuracy of the measurements and to compare them to an expected value, a basic analysis was carried out looking at the specific events described in section 6.2.3. For X-Axis accelerations, events 1 through 6 are relevant whereas for Y-axis accelerations, event 10 is used.

X-axis Acceleration Analysis

The approximation used to verify linear accelerations is based on the GPS measurements. By selecting a starting and ending position for each of the events using the latitude and longitude as

triggering values, it is possible to correlate the positions with a corresponding start and end time. Of course, the chosen starting and ending positions are positioned such that they occur in between the acceleration and deceleration peaks as seen in Figure 50, in order to accommodate any variations by the driver on the actual start and end points of the manoeuvre.

Once the start and end times are known, an average of the accelerometer-measured accelerations between the start and end times is computed. This is compared to acceleration calculated from the change in measured GPS speed between the start and end times by assuming a constant linear acceleration. A summary of the results of all 6 events over the same 3 tests is given in Table 28.

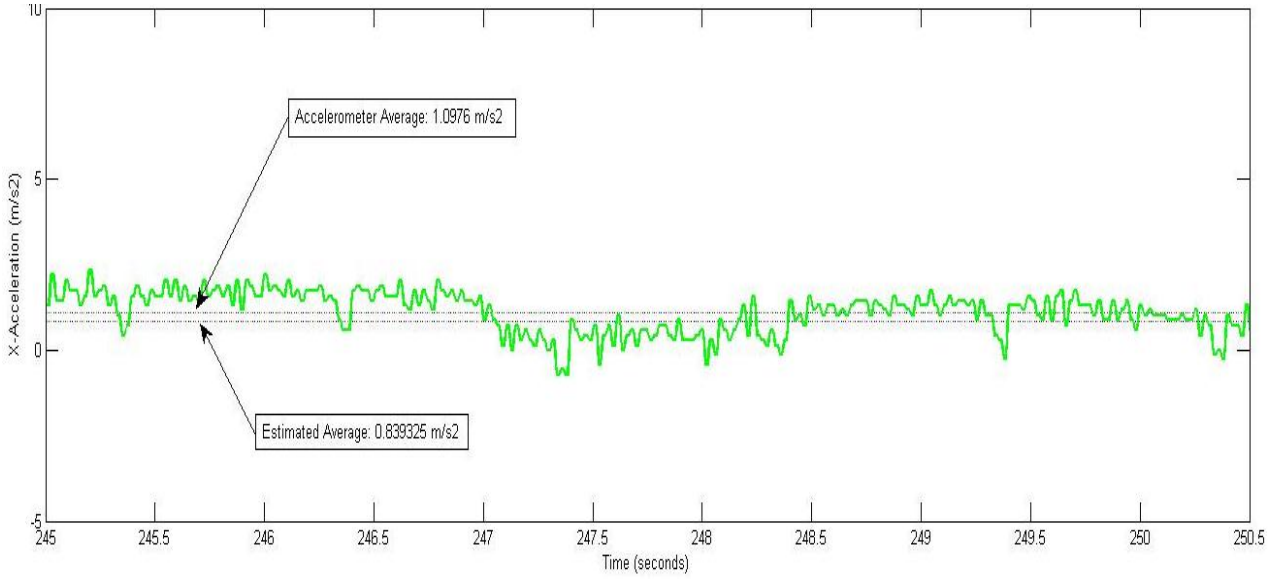


Figure 53: X-Acceleration Test 3 Event 5

As an example, looking at event 5 of test 3, a brief acceleration in a straight line is observed following a left hand corner with a mild change in speed, from approximately 13 to 23 knots. The accelerometer-measured acceleration during this event is shown in Figure 53 in green. The average accelerometer-measured acceleration is 1.0976m/s^2, whereas the linear acceleration estimated from GPS speed is calculated as 0.839m/s^2.

Table 28: In-Car Testing X-Axis Accelerometer Summary

Test 1						
Event	**1**	**2**	**3**	**4**	**5**	**6**
Event Duration (seconds)	8.26	3.00	8.00	9.00	5.00	3.00
Start Speed (m/s^2)	0.031	6.39	8.1	15.49	7.48	11.90
End Speed (m/s^2)	6.44	1.53	15.96	8.54	11.56	7.64
Calculated Average Acceleration (m/s^2)	0.78	-1.62	0.98	-0.77	0.82	-1.42
X-Axis Accelerometer Measured Average Acceleration (m/s^2)	1.15	-1.62	1.21	-0.41	1.09	-1.11
% Difference	33	0.1	19	88	25	28
Test 2						
Event	**1**	**2**	**3**	**4**	**5**	**6**
Δ Time (seconds)	8.31	3.28	8.29	9.85	5.09	3.29
Start Speed (m/s^2)	0.046	6.21	8.02	15.70	6.86	12.04
End Speed (m/s^2)	6.96	1.19	15.63	8.28	12.67	7.22
Calculated Average Acceleration (m/s^2)	0.83	-1.53	0.92	-0.75	1.14	-1.46
X-Axis Accelerometer Measured Average Acceleration (m/s^2)	1.35	-1.35	1.27	-0.51	0.98	-1.28
% Difference	38	13	28	49	17	14
Test 3						
Event	**1**	**2**	**3**	**4**	**5**	**6**
Δ Time (seconds)	7.96	3.69	8.54	10.11	5.91	6.24
Start Speed (m/s^2)	0.015	6.79	6.17	15.65	6.76	12.03
End Speed (m/s^2)	6.71	2.08	15.78	7.92	11.74	7.40
Calculated Average Acceleration (m/s^2)	0.84	-1.27	1.12	-0.77	0.84	-0.74
X-Axis Accelerometer Measured Average Acceleration (m/s^2)	1.27	-1.47	1.37	-0.57	1.10	-0.71
% Difference	34	14	18	35	24	4
Overall Average % Difference	**27%**					
Overall Standard Deviation	**19%**					

For all of the events in each test, a percent difference is calculated between the actual accelerometer-measured value and the approximate value calculated via the GPS measurement.

On average there is a 27% difference between the actual and calculated values, with a standard deviation of 19%.

The differences between the actual accelerations and the estimation provided by the linear acceleration based on GPS speed measurements can be explained by several factors. Primarily, the actual measured accelerations are not constant therefore the use of a constant acceleration to estimate the acceleration from GPS speed will introduce errors. However, the assumption of a constant acceleration is sufficient to confirm that the measurements produced by the accelerometer are of the correct order of magnitude and close to an expected value. Additionally, the GPS speed measurements are refreshed at a much slower rate (10Hz) than the accelerometer, which will also introduce some minor errors.

A typical example that highlights the non-linear nature of the measured X-axis accelerations is illustrated in Figure 54. This figure shows a portion of the measured acceleration as the car starts from a stationary position and increases its speed to approximately 13 knots. Between 4.5 and 6 seconds, a decrease or dip is present in the acceleration where the vehicle's measured acceleration is actually negative. The dip itself is due to the driver upshifting from first to second gear.

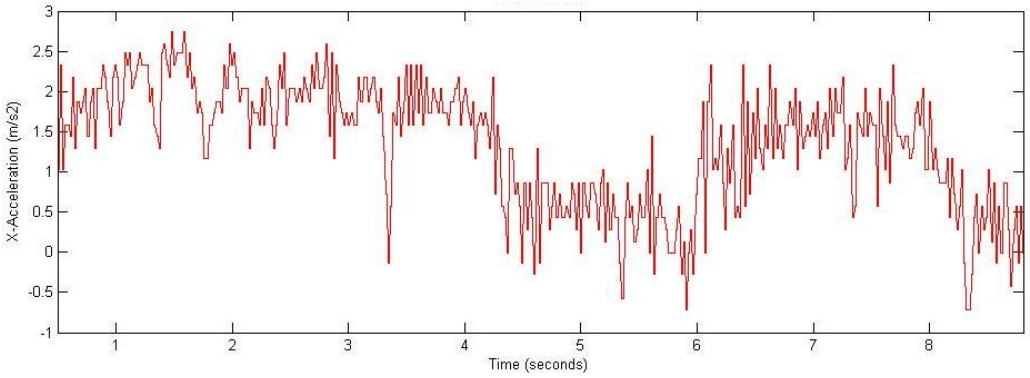


Figure 54: X-Axis Acceleration for Test 3 Event 1

The other detrimental factor to the linear approximation is the slower refresh rate of the GPS compared to that of the accelerometer. Since the refresh rate of the GPS is 10Hz and the accelerometer functions at approximately 100Hz in the case of the standalone (SD card) mode,

there are 10 different acceleration measurements associated with 10 identical GPS speed values. This can result in some discrepancies in the approximation depending on how much the actual speed of the vehicle has changed in between the GPS refresh time of 0.1 seconds.

Y-axis Acceleration Analysis

As previously stated, Y-axis acceleration analysis focuses on event 10, a constant radius right hand turn. The accelerometer-measured Y-axis accelerations are compared to an estimate based on the average speed carried by the vehicle through the bend. For a constant radius turn, the inward-pointing Y-acceleration is expected to satisfy $a_y = v^2/r$, where v is the speed of the vehicle and r is the radius of the turn [34]. The turn radius was estimated via Google earth™ as described above. Summarised analysis results are given in Table 29.

Table 29: Event 10 Y-Axis Acceleration Analysis Results

Event 10			
	Test 1	**Test 2**	**Test 3**
Turn Radius (m)	65	65	65
Average speed (m/s)	13.18	12.93	12.77
Calculated Average Y-Axis Acceleration (m/s^2)	2.67	2.57	2.51
Average Measured Y-Axis Acceleration (m/s^2)	2.33	2.20	2.11
% Difference	15	17	19
Average % Difference	**17%**		
Standard Deviation	**2%**		

The average difference between the average measured lateral acceleration and the calculated (expected) approximation is 17% with a 2% standard deviation.

Deviations between the estimate and the actual measurements can be explained by a few factors. First, the speed of the car during the turn is not constant for the duration of the turn, therefore nor is the lateral acceleration. However, the formula $a_y = v^2/r$ holds for any instant in time so could be applied at any point in time [34]. Further, any minor misalignment of the x,y,z axes of the prototype box with respect to forward/lateral/vertical axes with respect to the ground implies that the x,y,z axis accelerations that are measured by the prototype are in directions other than what might be considered to be "true" forward, lateral or vertical directions with respect to the ground. For example, it was assumed that the car does not roll, however any minor rolling of the car will affect the measured y-axis accelerations.

The average GPS speed value employed in the estimate is calculated over the duration of the event. It reality it is not constant: Figure 50 illustrates this fact. Even during intervals of relatively constant speed, a degree of variability is still present. Once the road speed limit has been reached, there are still some variations introduced by driver error and road conditions. These same errors carry over in the estimation.

The very nature of the GPS instrument is such that it has a much slower refresh rate. If the data acquisition prototype functions in standalone mode, there are approximately 16 different successive accelerometer readings for each change in the recorded GPS speed. When averaging out the measured GPS speed over a period of time, this can introduce some error with respect to the average speed used.

6.2.7. Gyroscope

Figure 55 shows the Z-axis (Yaw) rotational rate as a function of time for the three selected test runs. The figure is scaled to the full gyroscope measurement range of +/- 300 degrees/s. Each of the peaks corresponds to a certain manoeuvre or turn on the test circuit. Most notable is the large positive to negative spike at around 150 seconds. This is the full circle event 9 at point E as described in Figure 46.

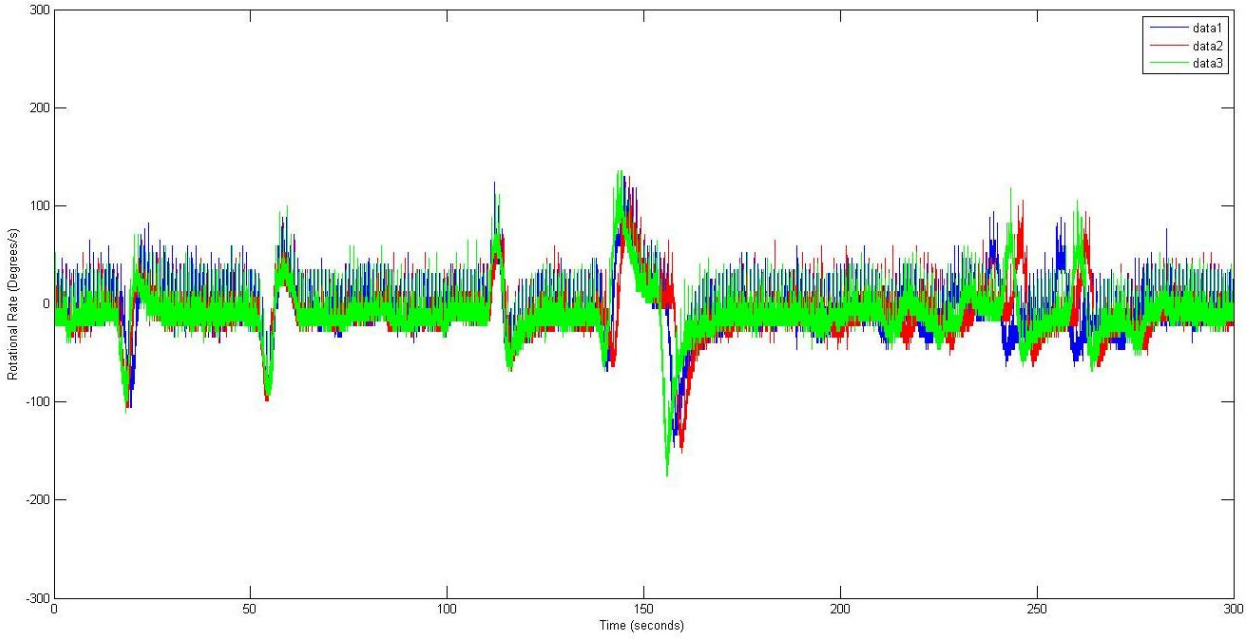


Figure 55: Yaw Axis Rotational Rate

Visually, the Z-axis (yaw) rotational rate measurement trace shows good repeatability over all three tests. The measured yaw rate peaks and their shapes are highly similar from test to test. The time at which these major events occur correlates with known turns or events in the test circuit as described in section 6.2.3.

In order to quantify the accuracy of the measurements, it is necessary to compare them to an expected value. This is again done based on the GPS, in a manner analogous to the X-axis accelerometer estimates described in section 6.2.6. Events 7 through 12 are analysed.

Event start and end times are chosen using the vehicle's position as the trigger. The calculated estimate is based on the GPS heading change over the chosen start and end points of a given manoeuvre: for example, event 7 sees the car change heading by 81 degrees. Using the time stamps for a given event's start and end time, it is possible to calculate how much time was required to complete the manoeuvre. Therefore, an average rotational (yaw) rate can be estimated by dividing the heading change with the total time needed to complete the manoeuvre. This particular estimation also assumes that the vehicle is yawing at a constant rate. A summary of the comparison between the actual measured and GPS-estimated rotational rates is given in Table 30. A percentage difference between the measured and estimated values is also listed to for all 6 events in the three separate tests.

Table 30: In-Car Testing Gyroscope Summary

General						
Event	**7**	**8**	**9**	**10**	**11**	**12**
Start Heading (degrees)	195	273	N/A	157	243	158
End Heading (degrees)	276	347	N/A	244	156	73
Test 1						
Event	**7**	**8**	**9**	**10**	**11**	**12**
Duration (seconds)	5.04	4.39	8.20	9.70	7.51	6.49
Calculated Average Rotational Rate (degrees/s)	16.06	16.84	45.85	8.97	11.59	13.09
Measured Z-Axis Gyroscope Average	15.49	40.05	38.32	9.26	36.31	18.05
% Difference	4	58	20	3	68	28
Test 2						
Event	**7**	**8**	**9**	**10**	**11**	**12**
Duration (seconds)	7.95	4.88	8.63	10.01	7.54	5.28
Calculated Average Rotational Rate (degrees/s)	10.57	15.97	44.17	8.79	11.68	15.90
Measured Z-Axis Gyroscope Average	22.05	29.56	36.17	9.98	12.77	17.51
% Difference	52	46	22	42	9	9

Test 3						
Event	**7**	**8**	**9**	**10**	**11**	**12**
Duration (seconds)	9.81	5.73	7.81	11.02	7.75	5.19
Calculated Average Rotational Rate (degrees/s)	8.56	13.43	47.22	8.17	11.36	16.56
Measured Z-Axis Gyroscope Average	16.73	30.15	43.69	10.92	9.66	14.52
% Difference	49	56	8	25	18	14
Overall Average % Difference	**28**					
Overall Standard Deviation	**21**					

Overall, the average difference between the estimated values and the measured values is 28% with a standard deviation of 21%.

- The principle differences between the actual measured rotational rate and the estimates based on the GPS heading can be attributed to similar factors to those outlined in section 6.2.6: the non-constant actual nature of the yaw rate and the lower GPS refresh rate.

To illustrate how a given yaw rate is not truly linear; Figure 56 shows the yaw rate during event 7, a square right hand turn. This manoeuvre is completed fairly quickly and it not representative of a "steady state" corner [73]. There is a periodically repeating signal spike which appears roughly every second which is attributed to experimental error: the test vehicle had an improperly balanced wheel on the front axle, causing a noticeable vibration in the steering input from the driver: This deficiency was visible on the video recording and eventually corrected.

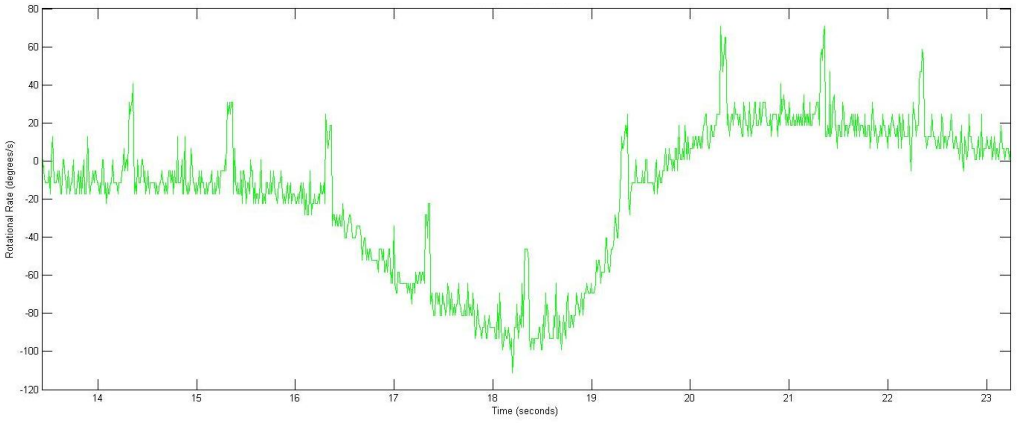


Figure 56: Z-Axis Rotational Rate for Test 3 Event 7

The constant radius right hand turn at event 10 can also be analysed from a different point of view. By using the gyroscope's rotational rate measurement and maintaining the assumption that the turn at event 10 has a constant radius of 65 metres, it is possible to calculate the speed at which the car is travelling through the turn by using the following relation $v = \omega \times r$ [34]. The resultant speed can then be compared with the actual GPS average speed for event 10 and this is shown in Table 31.

Table 31: Z-Axis Gyroscope Event 10 Analysis Results

Event 10			
	Test 1	**Test 2**	**Test 3**
Turn Radius (m)	65	65	65
GPS Average speed (m/s)	13.18	12.93	12.77
Average Y-Axis Measured Rotational Rate (radians/s)	0.16	0.153	0.14
Calculated Average speed (m/s)	10.18	9.97	9.26
% Difference	23	23	27
Average % Difference	**24%**		
Standard Deviation	**2%**		

Overall, the average percent difference for this particular comparison is 24%, with a standard deviation of 2%.

Any difference can also be explained by the same factors as the previous gyroscope approximations, GPS refresh rate and the non-constant nature of the actual yaw rate.

6.2.8. In-Car Testing Summary

In-car testing successfully demonstrated the prototype data acquisition system's ability to operate in a moving environment. Secondly, it demonstrated that under repeated test conditions, the instruments recorded repeatable results that were close to expected values for the manoeuvres.

The GPS proved to be very repeatable regarding measured position and was useful as the basis for verifying that the accelerometer and gyroscopes yielded sensible results. By successfully basing estimates on measured GPS speeds or headings, it was possible to show that both the accelerometer and gyroscopes actual measurements were within a reasonable range of the estimates.

6.3. Bicycle Testing

Bicycle testing is another test designed to assess the repeatability of the instrumentation in a moving environment, much like the automobile. The principal difference is that the bicycle typically operates at much slower speeds and tends to roll more than a car.

6.3.1. Justification

The bicycle is a convenient way to subject the prototype data acquisition system to a moving environment, much like the automobile. Using a pre-determined test circuit and running identical test laps, the bicycle is also subjected to easily repeatable conditions. However, the bicycle test offers the additional benefits (i) that bicycle speeds are similar to those of a slow moving sailboat (0 to 5 knots) and (ii) bicycles can be made to roll or lean in much the same way as a sailboat heels.

Using a road-going bicycle, the test circuit is still as smooth as the road used for the in-car testing, thus the oscillations caused by waves cannot be simulated.

6.3.2. Challenges

The main challenge in the bicycle test is determining a suitable mounting location for the prototype data acquisition system. This problem is compounded by the need for interaction with the prototype to initiate the data recording. The test operator must initiate the recording and then

give the rider a signal to proceed with the test. It was attempted to mount the device at several locations on the bicycle which would allow ease of access to the AxonII's pushbutton to start and stop SD card recording. These locations proved unsuitable because they were not correctly aligned with the direction of travel of the bicycle and proved too flimsy.

The final decision was to use the PC interface mode and to have the laptop placed in a pannier bag. The AxonII mounting solution is presented in section 6.3.3.

The tight radius of the turns used in the test circuit also proved difficult for many riders: it was highly challenging to remain on the track while maintaining a steady equilibrium and speed during turns. This is part of the reason why two modes were used: riding and walking, as described in section 6.3.3.

6.3.3. Test Circuit and Procedure

The prototype data acquisition system is mounted to the rear of a common road-going bicycle atop a pannier style luggage rack. This provides a flat mounting surface which is easy to align with the bicycle's direction of travel.

The test circuit itself was physically marked on the surface of a paved car park with chalk, making use of painted lines normally used to align parked vehicles. The actual layout is shown in Figure 57. The chosen test circuit has pairs of identical constant radius turns in the left and right directions.

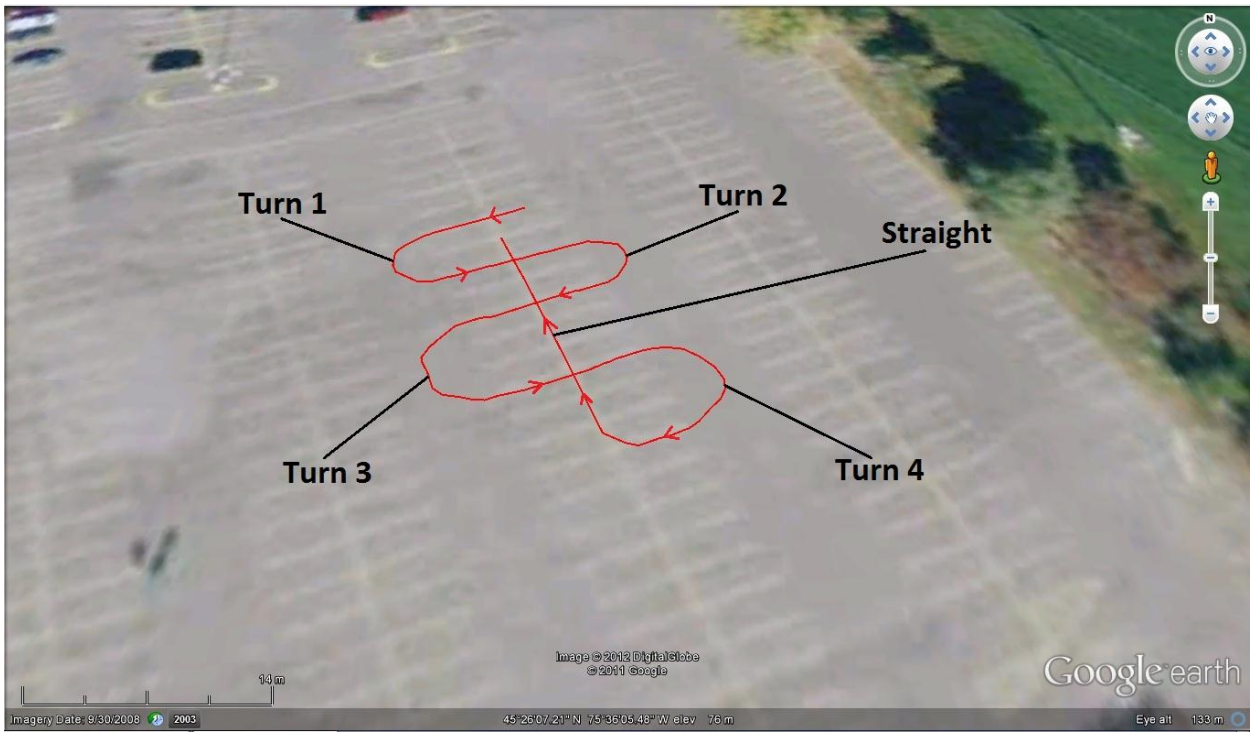


Figure 57: Bicycle Test Circuit

Each turn is 180 degrees except and arranged in matching left and right pairs: turns 1, 2 are small radius turns and turns 3, 4 have radii 50% larger than the first pair. Detailed measurements of turn radii and directions are given in Table 32. The approximate total length of the test circuit, as measured by Google earth™, is 113 metres.

Table 32: Bicycle Test Circuit Information

Event	Description
Turn 1	Left hand turn, 2.75m radius
Turn 2	Right hand turn, 2.75m radius
Turn 3	Left hand turn, 4.13m radius
Turn 4	Right hand turn, 4.13m radius
Straight	Straight line

Several test runs were carried out in two different ways: first using a rider who is tasked with following the prescribed track as closely as possible and second by pushing the bicycle around the circuit while on foot.

Riding the bicycle allows for higher speeds down the straight portion of the track and wider radius turns 3 and 4 when compared to pushing the bicycle. However, riding the bicycle implies that it is harder to maintain a smooth and precise heading especially on the tight portions of the course. Pushing the bicycle allows the user to follow the track much more precisely and smoothly while generally maintaining a constant roll angle through turns. It was also possible to increase the roll of the bicycle through the corners when pushing it.

The test numbers used from now are in reference to the order in which they were carried out. Video recordings of each test run can be seen in the video at [100]. The camera was mounted to the chest of the cyclist.

6.3.4. Roll Angle Estimation

Due to the bicycle's inherent leaning into corners, it is possible to employ the accelerometer's measurements to estimate the heeling (roll) angle. It is assumed that the bicycle does not pitch and that the roll angle is stable once the bicycle has transitioned from a vertical to rolled orientation.

Figure 58 represents the bicycle performing a left hand turn when viewed from the rear. Gravity is represented as a vector pointing downwards. The vertical axis is aligned as pointing towards the sky and the horizontal is positive to the rider's right hand side, when seated on the bicycle facing forwards. The vectors $\mathbf{A_y}$ and $\mathbf{A_z}$ represent the data acquisition system's y and z accelerometer axes, respectively.

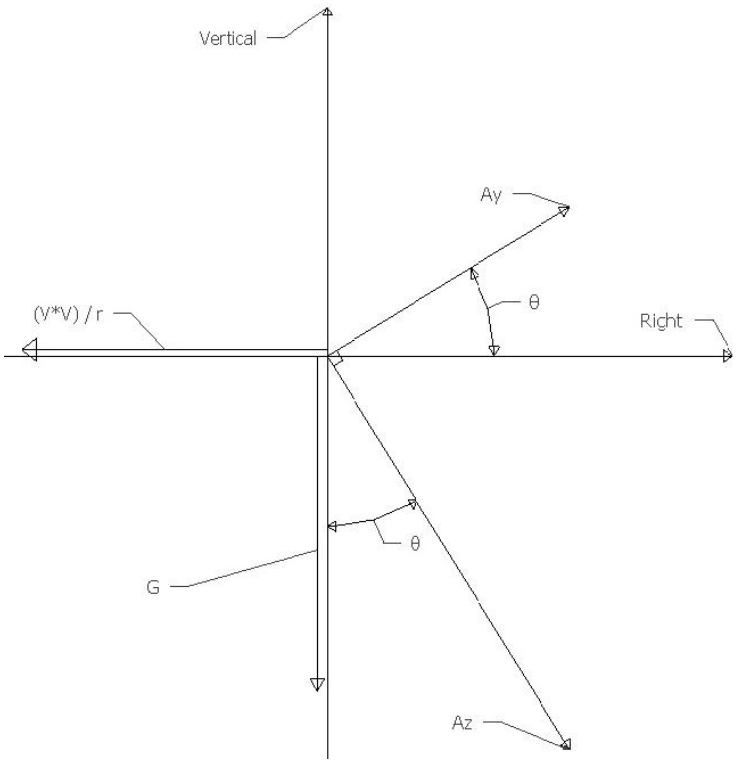


Figure 58: Heel Angle Estimation Diagram

Based on the concepts described in [34] and shown in Figure 59, centripetal acceleration $\mathbf{a_n}$ has magnitude v^2/ρ and is in the direction of $\mathbf{e_n}$, where v is the speed of the bicycle through a curve of radius ρ. The unit vector $\mathbf{e_n}$ is the inward pointing unit normal. This same centripetal acceleration is present for the case of the bicycle performing a left hand turn and is shown as a vector pointing left towards the centre of the left-hand turn.

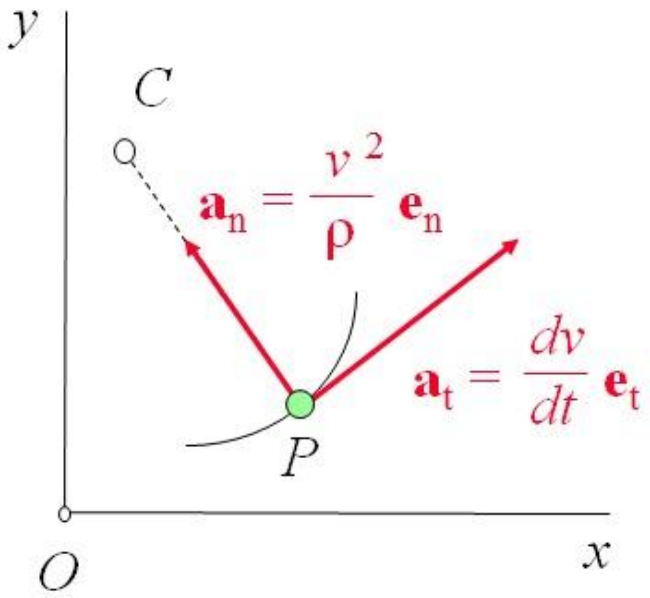


Figure 59: Centripetal Acceleration [34]

For the case of the bicycle, it is possible to define the vectors $\mathbf{A_z}$ and $\mathbf{A_y}$ as follows:

$$A_z = +g\cos(\theta) - \frac{v^2}{r}\sin(\theta) \tag{6.1}$$

$$A_y = -g\sin(\theta) - \frac{v^2}{r}\cos(\theta) \tag{6.2}$$

Multiplying $\mathbf{A_z}$ by cos(θ) and $\mathbf{A_y}$ by sin(θ) we obtain:

$$A_z\cos(\theta) = +g\cos^2(\theta) - \frac{v^2}{r}\sin(\theta)\cos(\theta) \tag{6.3}$$

$$A_y\sin(\theta) = -g\sin^2(\theta) - \frac{v^2}{r}\sin(\theta)\cos(\theta) \tag{6.4}$$

Subtracting equation (6.3) by equation (6.4) yields:

$$A_z\cos(\theta) - A_y\sin(\theta) = \left(+g\cos^2(\theta) - \frac{v^2}{r}\sin(\theta)\cos(\theta)\right) - \left(-g\sin^2(\theta) - \frac{v^2}{r}\sin(\theta)\cos(\theta)\right) \tag{6.5}$$

Simplifying equation (6.5):

$$A_z \cos(\theta) - A_y \sin(\theta) - g = 0 \tag{6.6}$$

Equation (6.6) can be numerically solved for values of the roll angle θ given measured acceleration values $\mathbf{A_z}$ and $\mathbf{A_y}$.

6.3.5. Roll Angle Estimation Results

Using the heel angle estimation equation discussed in section 6.3.4, roll angles were numerically calculated using actual test data, averaged out over approximately 0.5 seconds. Although each turn lasts for at least 4 seconds, a half second window proved adequate to estimate a roll angle.

Test numbering is clarified in section 6.3.3.

6.3.5.1. Cyclist Riding the Bicycle

For tests 10, 12 and 13, the cyclist is riding the bicycle in the usual fashion. First is turn 1, a 2.75m radius left hand turn, results are given in Table 33.

Table 33: Estimated Bicycle Roll Angle for Turn 1, While Riding

Turn 1 Riding			
Test Number	**10**	**12**	**13**
Turn 1 Duration (s)	0.5	0.5	0.5
Average Measured Y-Axis Acceleration(m/s^2)	-0.67	-0.87	-0.57
Average Measured Z-Axis Acceleration(m/s^2)	7.9	8.6	8.1
Estimated Roll Angle (Degrees From Horizontal)	4.9	5.8	4.0
Average Angle (Degrees)	**4.97**		
Standard Deviation (Degrees)	**0.87**		

The average estimated roll angle for turn 1 is 4.97 degrees. Figure 60 shows a snapshot of the bicycle's attitude during turn 1. Qualitatively, the bicycle is exhibiting some roll through the corner.

Figure 60: Turn 1 Test 10 [100]

Following turn 1 are the estimated roll angles for turn 2, which is the right hand analog of turn 1. Results are shown in Table 34.

Table 34: Estimated Bicycle Roll Angle for Turn 2, While Riding

Turn 2 Riding			
Test Number	**10**	**12**	**13**
Turn 2 Duration (s)	0.5	0.5	0.5
Average Measured Y-Axis Acceleration(m/s^2)	-0.24	-1.6	-1.1
Average Measured Z-Axis Acceleration(m/s^2)	8.6	9.1	8.7
Estimated Roll Angle (Degrees From Horizontal)	1.6	9.1	7.0
Average Angle (Degrees)	**5.90**		
Standard Deviation (Degrees)	**3.9**		

The average estimated roll angle for turn 2 is 5.90 degrees. Figure 61 shows a sample view from the chest mounted camera. It can be seen that the bicycle is leaning slightly to the right hand

side. Turn 2 also proved to be the most difficult for the rider to execute smoothly, as is visible in [100]. Because the rider is able to carry more speed entering turn 2 compared to turn 1, he must compensate by rolling the bicycle to a greater extent compared to turn 1. The downside of this is the rider's tendency to overcompensate by exaggerating the roll. This instability results in more oscillations in the bicycle's roll angle which is problematic with the basic assumption of the estimation: that the roll angle remains steady after the rider has transitioned from his vertical orientation to a leaned orientation.

Figure 61: Test 10 Turn 2 [100]

Turn 1 and 2 exhibit very similar characteristics, both have identical radius thus both have a similar ideal speed at which the rider should approach them. They are relatively tight for a bicycle to handle smoothly but it is reasonable to expect similar roll angles for both cases. Since turn 1 is taken earlier on the test circuit, the rider has less time to build up speed; therefore the roll angle is slightly less than that of turn 2.

The following estimations are for turn 3, a 4.13m radius left hand turn for which results are given in Table 35.

Table 35: Estimated Bicycle Roll Angle for Turn 3, While Riding

Turn 3 Riding			
Test Number	**10**	**12**	**13**
Turn 3 Duration (s)	0.5	0.5	0.5
Average Measured Y-Axis Acceleration(m/s^2)	-1.6	-1.1	-0.56
Average Measured Z-Axis Acceleration(m/s^2)	9.5	8.9	8.8
Estimated Roll Angle (Degrees From Horizontal)	9.8	6.8	2.8
Average Angle (Degrees)	**6.46**		
Standard Deviation (Degrees)	**3.5**		

The average estimated roll angle for turn 3 is 6.46 degrees, slightly more than the roll angle of turns 1 and 2. Figure 62 shows a sample view of turn 3

Figure 62: Test 10 Turn 3 [100]

Because turn 3 has a larger radius than the two previous corners, it can be taken a higher speed and increased roll angle.

The final estimation is intended for comparative purposes with a situation in which the bicycle is supposed to be upright and have no roll angle. Table 36 shows the roll angle estimation when the bicycle is being ridden in a straight line.

Table 36: Estimated Bicycle Roll Angle While Riding in a Straight Line

Straight Line Riding			
Test Number	**10**	**12**	**13**
Straight Line Duration (s)	0.5	0.5	0.5
Average Measured Y-Axis Acceleration(m/s^2)	-0.27	-0.15	-0.18
Average Measured Z-Axis Acceleration(m/s^2)	9.8	9.5	9.3
Estimated Roll Angle (Degrees From Horizontal)	1.6	0.90	1.1
Average Angle (Degrees)	**1.2**		
Standard Deviation (Degrees)	**0.33**		

The average estimated roll angle for riding in a straight line is 1.2 degrees. The nature of a bicycle is that riding in straight lines is smoother as speed increases and there is an inherent rolling oscillation caused by the rider shifting his weight side-to-side while pedalling.

Figure 63: Test 10 Straight Line [100]

As Figure 63 illustrates, the bicycle does indeed exhibit oscillation which explains why the roll angle estimation yields non-zero results.

6.3.5.2. Cyclist Pushing the Bicycle

For the tests numbering 14 through 16, the cyclist is pushing the bicycle while walking on the right hand side of the bicycle when facing forward. First is turn 1, a 2.75m radius left hand turn, the results of which are shown in Table 37.

Table 37: Estimated Bicycle Roll Angle for Turn 1, While Pushed

Turn 1 Pushing			
Test Number	**14**	**15**	**16**
Turn 1 Duration (s)	0.5	0.5	0.5
Average Measured Y-Axis Acceleration(m/s^2)	-2.1	-2.4	-2.3
Average Measured Z-Axis Acceleration(m/s^2)	9.5	9.1	8.7
Estimated Roll Angle (Degrees From Horizontal)	12.4	14.8	14.8
Average Angle (Degrees)	**14**		
Standard Deviation (Degrees)	**1.36**		

The average estimated roll angle for turn 1 is 14 degrees.

Next is turn 3, a 4.13m radius left hand corner for which estimated roll angle results are given in Table 38.

Table 38: Estimated Bicycle Roll Angle for Turn 3, While Pushed

Turn 3 Pushing			
Test Number	**14**	**15**	**16**
Turn 3 Duration (s)	0.5	0.5	0.5
Average Measured Y-Axis Acceleration(m/s^2)	-1.9	-1.8	-1.9
Average Measured Z-Axis Acceleration(m/s^2)	8.5	8.9	9.3
Estimated Roll Angle (Degrees From Horizontal)	12.8	11.4	11.8
Average Angle (Degrees)	**12**		
Standard Deviation (Degrees)	**0.72**		

The estimated roll angle for turn 3 is 12 degrees. This is a similar result as obtained for turn 1 under the same conditions.

6.3.6. Bicycle Testing Summary

Summarised in Table 39 are the average estimated roll angles for turns 1 through 3 and the straight line for both the pushing and riding test cases.

Table 39: Summary of Estimated Bicycle Roll Angles

Average Estimated Roll Angles		
	Riding	**Pushing**
2.75m radius turns	5.44	14
4.13m radius turns	6.46	12
Straight line	1.2	N/A

Generally, the estimation results agree with the corresponding video logs and test notes: more roll was induced when the rider was pushing the bicycle than was observed while riding it. The overall magnitude of the estimated roll angles is very slight, which makes sense given the tight radius turns of the test circuit.

To further support the roll angle estimation results, a portion of a straight line was analysed. The estimated roll angle in a straight line is effectively zero given the fact that pedalling the bicycle does induce a slight amount of vertical oscillation.

6.4. Sailboat Testing

Sailboat testing is an important part of the prototype validation and serves to verify several important points but chiefly to demonstrate one key point: the prototype is capable of functioning in its intended environment. By employing the prototype data acquisition system in-situ it is possible to establish that it successfully fulfills its design requirements.

6.4.1. Justification

Having demonstrated that the prototype fulfills all of the measurement and electronics related design requirements outlined in section 3.4 with previous tests, on the water testing serves to verify the last remaining criteria:

- Withstand the rigors of the environment found on any boat, such as the possibility of water or the impact of waves.

This criterion was determined to be water resistance and shock resistance or having the ability to operate and survive on a sailboat during a race. Finally, sailboat testing should serve to provide a good measure of the unit's portability, ease of installation and ease of operation.

6.4.2. Challenges

The sailboat testing proved challenging in that it was difficult to find a suitable mounting location for any device. This problem is compounded by the fact that the installation has to remain temporary. It is also critical to minimise the setup time required prior to any testing, due to the nature of evening club racing: access to the boats is restricted until owners are present, and they can only be present after working hours. This leaves a very small window of opportunity to complete equipment set-up while the crew is preparing the boat.

Several attempts were made to attach the prototype data acquisition system using gaffer tape or finding an area where it would be wedged into place. The final solution to this problem was to use c-clamps and this is presented in section 6.4.4.

Wiring and attaching the wind instrumentation also proved difficult: the wires need to be kept well out of the way of the boat's running rigging and areas where the crew needs access during a race. At the same time, not creating a safety hazard with these wires is essential.

Using a laptop on a sailboat during racing conditions proved to be very difficult: several attempts were made to mount it at a location where it would remain shielded from potential water ingress. There was also a need to provide heavy padding to help absorb the severe shocks present when the boat is crashing through waves.

The final hurdle to overcome during sailboat testing is human limitation: the test operator must remain below deck to monitor the test equipment while underway. Sailboat cabins can be cramped, present poor ventilation and are subject to shocks and rough movement. Every time the crew performs a manoeuvre such as a tack or gybe, the test operator must shift his or her position to the upwind side of the boat to help maintain the proper hull trim.

6.4.3. J/24

The chosen type of sailboat for testing is an international J/24 shown in Figure 64, a very popular type of racing keelboat [101]. It is called the J/24 because its overall length is 24 feet (7.32m) and is known as a "One-Design" boat, meaning that all J/24s are constructed according to stringent rules and regulations.

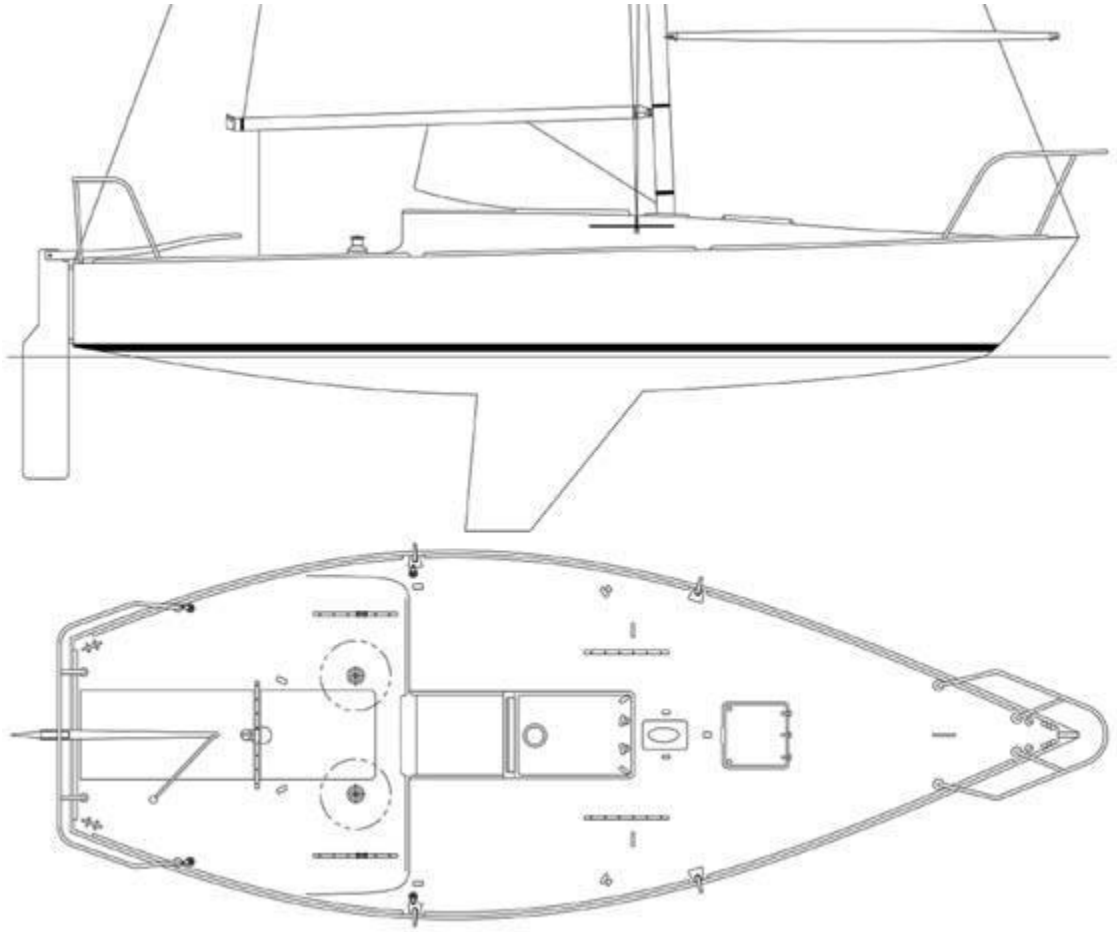


Figure 64: J/24 Dimensions [102]

Summary measurements and dimensions of the J/24 are shown in Table 40. The J/24 is a very good candidate for sailboat testing because it is a relatively fast and manoeuvrable craft which is used in a common racing format. It is accessible, with numerous examples available locally all of which are involved in a fairly competitive club racing scene.

Table 40: J/24 Dimensions [102]

Description	US	Metric
LOA	24.00 ft	7.32 m
LWL	20.00 ft	6.10 m
Beam	8.90 ft	2.71 m
Draft	4.00 ft	1.22 m
Displacement	3,100 lb	1,406 kg
Lead Keel	950 lb	431 kg
Engine	Outboard 4 hp	

Numerous club races happen during the summer sailing season and the majority of testing occurred during evening and weekend races. A total of 8 test sessions were carried out from July to October 2011.

6.4.4. Installation

Physical installation of the prototype data acquisition unit proved challenging, bearing in mind that the entire installation needed to be temporary and require that no permanent changes be made to the test vessel. Additional installation considerations require that the unit be aligned properly with the sailboat's forward and lateral axes while being horizontal.

The most suitable location for aligning the unit correctly proved to be the flat floor intended as the fore berthing area in the J/24 shown in Figure 65. The flat floor provides good support and in combination with the vertical bulkhead aligns the unit properly. Rubberised plastic C-clamps were used to secure the unit into position without the need for permanent fasteners or other damaging methods.

Figure 65: Prototype Data Acquisition System J/24 Mounting Location

The GPS antenna should ideally be mounted with a clear line of sight to the sky in order to maximise satellite signal reception. Although it is fully waterproof as explained in section 5.2.4, the mounting location for the antenna needed to be protected from crew movement, therefore the underside of the foredeck hatch proved ideal. The foredeck hatch as shown in Figure 66 is made of transparent Poly(methyl methacrylate) and did not prove problematic with respect to signal reception. The antenna was affixed with residue-free painter's tape.

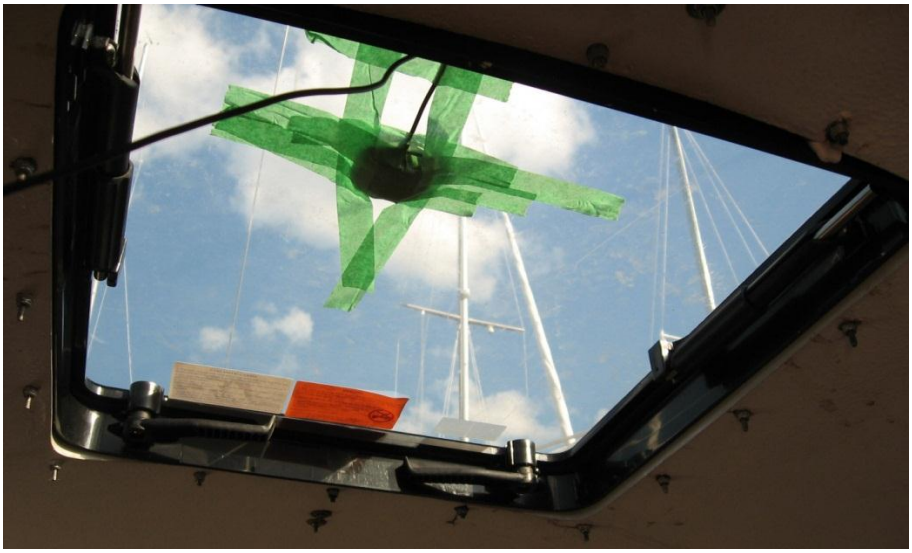


Figure 66: GPS Antenna J/24 Mounting Location

The other peripheral devices which require external mounting are the wind instruments. Ideally, these are placed at the very top of the mast, in order to get the best possible wind readings, unimpeded by the disturbances caused by sails, waves, rigging or other boats. Due to the difficulty in actually accessing the top portion of the mast to install temporary instruments, these were positioned on the safety railing which lines the periphery of the deck as visible in Figure 67. Temporary mounting is accomplished using plastic cable ties.

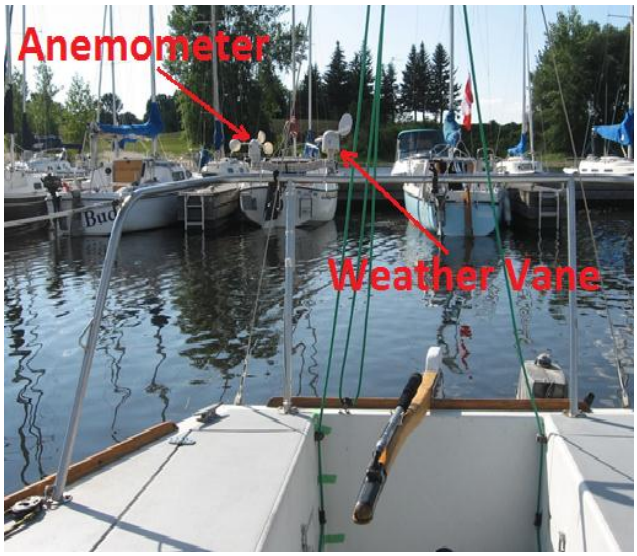


Figure 67: Wind Instruments J/24 Mounting Location

Cable management for the wind instruments is important due to the fact that they must traverse the J/24's cockpit area without impeded operation of the boat or being accidentally removed. It was not possible to route the cable through the hull so the wiring was attached to the side of the cockpit again using residue-free painter's tape as shown in Figure 68.

Figure 68: Wind Instruments J/24 Cable Management

The final installation consideration regards the USB interface mode for the prototype data acquisition system: finding a suitable location for a portable computer. This needed to be shock proof and absolutely dry since only a standard netbook format computer with no waterproofing

was used. A larger Pelican 1400® case containing the computer was used and this fitted neatly in the J/24 galley's sink as is visible in Figure 69.

Figure 69: Portable Computer J/24 Mounting Location

Practically, the Pelican 1400® case containing the computer was left open to allow the USB wire to remain connected with the prototype data acquisition system.

6.4.5. Operation

Operating the prototype data acquisition can be performed in one of two modes, PC interface (USB) mode or standalone (SD card) mode as described in sections 5.3 and 5.5, respectively.

In the PC interface mode, operation is relatively simple: power up prototype data acquisition system and laptop, verify USB connection and activate the Matlab script. In practise, the PC interface mode is much harder to implement correctly because of limitations imposed by the laptop. These limitations include

- Finding a proper location to stow the laptop
- Submitting the laptop to shocks
- Ensuring the laptop remains dry at all times
- Laptop battery life is less than that of the prototype data acquisition system

Mounting the laptop is difficult in the J/24 because the laptop is fairly large and needs to be accessible to the user. It must be powered up and the Matlab script activated and this is generally done while underway. Interacting with the laptop is not easily accomplished in due to the motion of a sailboat. After completing the tests, the user must be able to interact with Matlab to save the test data before the laptop battery runs out of charge.

In practise, the chosen netbook portable computer exhibited a battery life of approximately 2 hours, severely limiting the duration of the tests. This required the measurement time to be set at 1.5 hours in order to have sufficient time to save the test data. Failure to save the data on time means that the all measurement data is simply lost resulting in wasted test time: a severe limitation of the PC interface (USB) mode.

The other severe limitation is that most portable computers cannot withstand shock loads: on several occasions, the netbook suffered crashes due to hard drive related errors. This necessitated a system reboot and all gathered test data was lost. The danger with this situation is that once the laptop is stowed, the user may not notice the exact moment of failure, which again results in lost testing time.

The standalone (SD card) mode proved not to be susceptible to the same issues as its PC interface counterpart because it eliminated the need for an on-board personal computer. In fact, the standalone mode is much easier to use, the user simply powers up the prototype and initiates the recording before setting off and is able to focus on other tasks confident in knowing that the device has at least 6 hours of battery life and recording time. Once the race is over, the recording was simply stopped and the unit powered down.

6.4.6. Environmental Factors

The two main environmental factors encountered by the data acquisition system are shock loading and water.

As an illustrative example, the September 15th 2011 test provided the highest wind speeds encountered during the series of sailboat tests. As can be seen in Figure 70, the average wind speed at the Ottawa International Airport between the hours of 16:00 to 18:00 was around 30km/h. This average wind speed qualifies as a fresh breeze on the Beaufort scale [77], with

typical wave height between 2 to 3 metres (half the total length of the J/24). This average wind speed also included several gusts exceeding 50km/h.

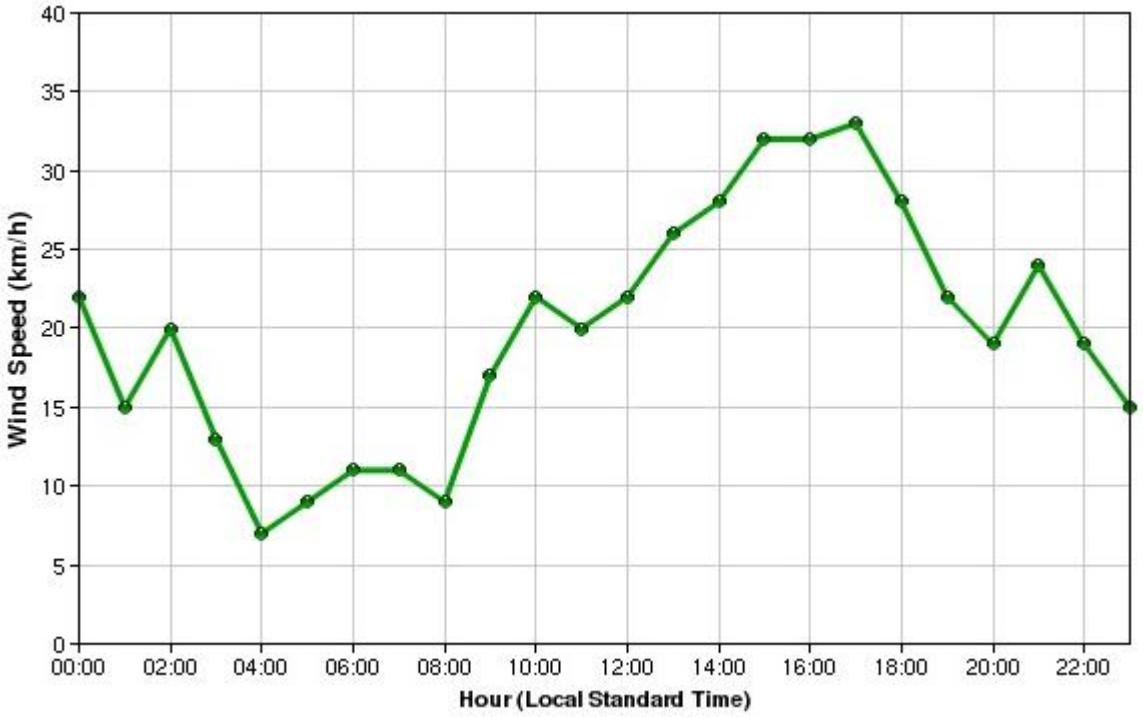


Figure 70: Hourly Wind Speed for September 15, 2011 Ottawa [103]

As experienced on the boat, the occasional wave did wash across the bow of the J/24 and depending on the boat's heading relative to the wind and waves; the crashing of the boat through the waves did become uncomfortable and disorientating below deck. These repeated impacts over several hours did not adversely affect the electronics.

The installation of the prototype data acquisition system is very near to the mast, as is visible in Figure 65 as the vertical aluminium member. There is a large gap where the mast goes through the upper deck structure that allows for water ingress when waves wash across the deck. This conveniently landed directly atop the Pelican® case repeatedly and proved harmless to the device contained within.

6.4.7. Sailboat Testing Summary

The sailboat testing successfully demonstrated the prototype's ability to successfully operate in its intended environment. It survived the hazards of wave impacts and water splashing without fail thus fulfilling the last of the design criteria to be verified by the testing program.

In practise, the ease of the temporary installation and removal proved adequate, requiring one trained person approximately 20 minutes to complete. During the sailboat testing, it became readily apparent that the standalone (SD card) mode of operation was highly preferable due to the numerous disadvantages afforded by the portable computer.

6.5. Summary

The entire testing program successfully demonstrated that the prototype data acquisition system is fully functional in its intended environment: the sailboat. Through static bench testing, the basic functions were tested individually and proven to be reliable, showing no faults and predictable behaviour while at rest.

Mobile testing, both in-car and on a bicycle demonstrated the prototype`s ability to function in a moving environment. The measurement results proved to be repeatable and yielding sensible results. Both the accelerometer and gyroscope data was compared to reasonable estimates, either based on observation or reliable GPS data. These comparisons verified that the instrumentation behaved correctly and yielded results in the expected range.

Finally, sailboat testing established the data acquisition system`s ability to function in its intended environment properly and without fault. It effectively survived the conditions present in a marine environment and was easy to deploy and use in-situ under the standalone (SD card) mode of operation.

Chapter 7 Summary & Conclusion

The initial goals of the present research were set in chapters 3, after which the successful design, construction and validation of a prototype were described in chapters 4, 5 and 6 respectively. The present chapter will summarise the extent of the work undertaken while exploring possible future directions for this research. Finally, aspects of the prototype data acquisition system which could be enhanced will be suggested.

7.1. Thesis Summary

A prototype data acquisition system was successfully constructed. This prototype measures the following:

- Linear accelerations
- Rotational rates
- Position relative to the Earth
- Compass heading
- Velocity
- Wind heading and speed

This system satisfies the original research objective of designing and constructing a device which can quantifiably measure and record the motion of a sailing yacht while under racing conditions, as outlined in section 3.1.

The prototype was then validated by a series of tests including static, in-car and bicycle mounted scenarios. These served to establish the proper functionality of the constructed device under controlled conditions. It exhibited predictable, repeatable behaviour with consistent results. In several analyses with comparative estimates, the data recorded by the data acquisition system is found to be within a reasonable margin to the expected values.

Ultimately, the prototype data acquisition successfully recorded data during a full scale test on a sailboat under racing conditions.

This fully functional prototype was also assembled for a total component cost of 409.69 $US which fulfills the major design consideration of minimising financial expense. This demonstrates the possibility of constructing a fully featured IMU for the purpose of measurement and analysis for a relatively low cost.

7.2. Prototype Data Acquisition System Final Specifications

The final system specifications of the prototype data acquisition system are summarised in Table 41.

Table 41: Prototype Data Acquisition System Specifications

Specification		
Measurement Refresh Rate	97	Hz
Battery		
Voltage	7.2	Volts
Charge	1800	mAh
Minimal Duration	9	Hours
Accelerometer		
Range	30	m/s^2
Resolution	0.14513	m/s^2
Compass		
Range	0 - 360	Degrees
Resolution	0.5	Degrees
Gyroscope		
Range	+/- 300	Degrees/Second
Resolution	1.4663	Degrees/Second
Anemometer		
Resolution	0.1	Knots
Weather Vane		
Vane Range	0 - 360	Degrees
Vane Resolution	22.5	Degrees
GPS		
Refresh Rate	10	Hz
Cold Start Time	29	Seconds
Positional Accuracy	2.5	Metres
Velocity Accuracy	0.1	m/s

This device has also been successfully packaged in a waterproof container with external anemometer, weather vane and GPS antenna. This packaging form factor has also been shown to function in its intended environment: the racing sailboat.

Table 41 can be compared with Table 8 to highlight the following points:

- Final prototype exceeded specified anemometer resolution from 1 knot to 0.1 knot
- Final prototype exceeded specified measurement refresh rate from 20Hz to 97Hz

7.3. Contributions

The concept of applying inertial measurement technology to the domain of sailing yachts for the express purpose of creating a tool for crew performance enhancement and training is novel.

The specific combination for each of the sensors used in conjunction with the AxonII microcontroller is also new; this includes the SD card interface and recording format used to transfer and store the measured data.

The specific layout, electronic design and packaging of the prototype data acquisition system is original.

The Matlab scripts used for connecting the PC interface mode between the AxonII through the USB, the importation function of the prototype's data from an SD card and all of the post-processing code are a novel contribution.

All of the experimental testing and design validation of the prototype data acquisition system were original work designed and implemented by the author.

7.4. Future Work and Suggested Improvements

7.4.1. Hardware

The current prototype packaging meets the stated design objectives but could be improved for field-use. Even though the Pelican® case provides a decent flat mounting surface which allows for easy clamping, it has sometimes been problematic. A flat surface, such as a bulkhead or flat

floor, which is suitable to use as a clamping location is oftentimes difficult to find. This is due to the relative rarity of ideal mounting places on an actual sailboat, better provisions for mounting brackets which can be mounted with screws or bolts would be an improvement to the present design.

Incorporating more weather sensors may be useful in providing context to the recorded motion data. Typical weather measurements are:

- Ambient air temperature
- Water temperature
- Atmospheric pressure

These would serve to bolster the weather trends typically recorded by shore weather stations with data taken on the sailboat.

It would also be possible to reduce the overall size of the device packaging to a smaller Pelican® case or similar enclosure. A miniaturised version, with integrated GPS antenna and external switchgear would be much easier to mount on a small dinghy, hydrofoil or even on a sailboard/windsurfer. This could be achieved by a small production run of a specific printed circuit board incorporating all of the hardware into one specific device.

Further developments could include a display function which would allow the user to monitor the data in real-time. This could be tied in to future data analysis work: as an example, manoeuvre detection. When the device detects that a given manoeuvre, such as a tack, is occurring, a display would allow the user to see that the device has detected it. Specific benchmarks or performance metrics for the manoeuvre could be displayed to allow the user to gauge his or her performance as it happens.

7.4.2. Data Gathering

The next phase of the project is the acquisition of more test data in order to build up a database for post-processing and analysis methods. This can be attempted in multiple ways:

- Using same vessel and crew, varying the race and weather conditions

- Multiple identical vessels simultaneously
- Different vessels simultaneously
- Different vessels, varying weather conditions

Each testing strategy can be used for different purposes, either determining the loads and forces on a particular boat or for the purpose of crew training and improvement. Having sufficient test data is a solid foundation for the next logical step in the project: analysis.

7.4.3. Data Analysis

Data analysis will provide additional opportunity for innovation and furthering the understanding of the dynamics of sail boat motion. The data gathered with the prototype data acquisition tools can be processed and filtered to determine the orientation of the sailboat at any point in time. Standard techniques for accomplishing this exist in other fields such as robotics and UAVs.

Any attempt at using filtering or data fusion techniques on measured data should be preceded by error compensation or mitigation. Practical experience with the prototype data acquisition system shows that the many of the sensors involved in inertial measurement are very noisy. For the purpose of validation testing, a simple averaging was employed to address this issue and this proved satisfactory for the type of analyses used. Further analysis or any attempt to increase its complexity would require more advanced methods to compensate for inherent or induced errors. Several methods to accomplish this exist already, especially from the UAV domain, which should be applied to this particular application.

One key hardware modification which could potentially improve the usability of the gyroscopic measurements is replacement of the HMC 6352 digital compass with a three axis magnetometer. This is in essence, switching a two-dimensional compass for a three-dimensional one. This three axis magnetometer would reduce the error that the current compass exhibits due to the heeling of the boat. In addition, it would provide a constant heading reference to the Earth's magnetic North, which could potentially be very useful when integrating sensor measurements to obtain positional information

The gathered data could also be used to quantify the forces and dynamic effects on the sailboat. By determining loads, optimisation of the structural components of a sailboat can be improved. Should an instrumented boat experience a catastrophic failure, the data collected prior to and during the failure event can potentially identify the cause. Knowing the cause of the failure is beneficial to boat designers and operators alike. Designers could use this information to improve their designs and boat owners could potentially learn to avoid the conditions that caused the failure.

Finally, the data acquisition device could be used as a crew training and performance improvement tool. A performance analysis should be conducted, specifically focusing on typical individual manoeuvres such as tacking and gybing. A series of repeated but identical manoeuvres in similar weather conditions should be undertaken and compared in order to establish a baseline performance. Then, various changes could be made to the way the manoeuvre is performed in order to determine which is optimal for use during a race.

In the same spirit, synchronising the prototype data acquisition system's measurements with recorded video would be most beneficial. Video replay is a standard and effective tool in providing a visual record of an event that occurred on deck, which can be useful in crew training. During post-processing of the data, video recordings would also be useful in correlating events to measurements. From this, it should be possible to eventually generate an algorithm to recognise these events purely from the recorded data.

7.5. Conclusion

The goals outlined for this research have been met successfully, allowing for the beginning of the next phase of the project. With the sailboat test data already gathered, a post-processing tool is currently being developed. While the development of such a tool is likely to be a long iterative process, the current data acquisition prototype will continue to serve as the foundation for the next phase of the project. It has already been replicated in preparation for the next sailboat testing season in order to maximise the amount of data that can be gathered during a given race. It is entirely feasible that further evolutions of the current prototype will be developed to

eventually achieve a consumer grade product which can be used by sailors and yacht designers to maximise performance.

References

1. **Conner, Dennis.** *ThinkExist.com*. [Online] 2011. [Cited: 12 04, 2011.] http://thinkexist.com/quotation/it_basically_was_an_art_before-we-re_just/203073.html.

2. **Honda, Soichiro.** SearchAutoParts.com. [Online] [Cited: 12 1, 2011.] http://motorage.search-autoparts.com/motorage/Technical/Does-racing-really-improve-the-breed/ArticleStandard/Article/detail/712103.

3. Story Lead: Financial. *Volvo Ocean Race Offical Press Information*. [Online] [Cited: 11 17, 2010.] http://press.volvooceanrace.com/?p=142#more-142.

4. **Alan S. Moris, Reza Langari.** *Measurement and Instrumentation.* San Diego : Elsevier Inc., 2012.

5. **Douglas C. Montgomery, George C. Runger.** *Applied Statistics and Probability for Engineers.* New Jersey : John Wiley & Sons, Inc., 2010. 978-0-470-05304-1.

6. **Necsulescu, Dan Sorin.** *Mechatronics.* New Jersey : Prentice-Hall, 2002.

7. **Staller, Len.** Understanding Analog to Digital Converter Specifications. *Embedded Systems Design*. 2005.

8. **Walden, R.H.** Analog-to-Digital Converter Survey and Analysis. *IEEE Journal on Selected Areas in Communications*. 1999, Vol. 17, 4.

9. *IMU Data Generation from the Defined Motion Using the NDI Controller Design Technique.* **Atesoglu, Ozgur.** San Diego : IEEE/ION, 2006.

10. *MEMS IMU for AHRS Applications.* **W. Geiger, J. Bartholomeyczik, U. Breng, W. Gutmann, M. Hafen, E. Handrich, M. Huber, A. Jäckle,.** Monterey, California : Position, Location and Navigation Symposium, 2008 IEEE/ION, 2008.

11. **Apple Inc.** Apple iPhone 4S Technical Specifications. *Apple.com*. [Online] [Cited: 01 08, 2011.] http://www.apple.com/iphone/specs.html.

12. **WiiBrew.** WiiMote. *WiiBrew.org*. [Online] [Cited: 01 08, 2012.] http://wiibrew.org/wiki/Wiimote.

13. **TomTom.** GO 2535 WTE Series Technical Specifications. *TomTom.com*. [Online] [Cited: 10 28, 2011.] http://www.tomtom.com/en_ca/products/car-navigation/go-2535-tm-wte/index.jsp#tab:specifications:technical-specs.

14. **Robert Bosch GmbH.** Bosch Electronic Stability Control Technical Specifications. *bosch-escential.com*. [Online] [Cited: 01 08, 2012.] http://www.bosch-escential.com/us/language1/technical_features.html.

15. **Samsung Group.** TL500 10 Megapixel Rotating LCD Digital Camera. *Samsung.com*. [Online] [Cited: 01 15, 2012.] http://www.samsung.com/us/photography/digital-cameras/EC-TL500ZBPBUS-features.

16. **Encyclopaedia Britannica.** Smart Bomb. *Encyclopaedia Britannica*. [Online] [Cited: 01 08, 2012.] http://www.britannica.com/EBchecked/topic/549443/smart-bomb.

17. —. Unmanned Aerial Vehicle (UAV). *Encyclopaedia Britannica*. [Online] [Cited: 01 08, 2012.] http://www.britannica.com/EBchecked/topic/497716/unmanned-aerial-vehicle-UAV.

18. **Airforce-Technology.com.** Predator RQ-1/MQ-1/MQ-9 Reaper -, United States of America. [Online] [Cited: 01 08, 2012.] http://www.airforce-technology.com/Projects/predator-uav/.

19. *On the development of guidance system design for ships operating in close proximity*. **Pedersen, E., Shimizu, E. and Berg, T.E.** Monterey, California : Position, Location and Navigation Symposium, 2008 IEEE/ION , 2008.

20. **Colibrys SA.** Inertial Sensing. *colibrys.ch*. [Online] [Cited: 01 08, 2012.] http://www.colibrys.ch/e/page/151/.

21. **General Electric Oil & Gas.** GE Oil & Gas. *MagneScan IMU Inspection Tool*. [Online] [Cited: 01 08, 2012.] http://site.ge-energy.com/businesses/ge_oilandgas/en/prod_serv/serv/pipeline/en/inspection_services/mapping_and_caliper/magnescan_imu/index.htm.

22. *Development of a MEMS based wearable motion capture system*. **Abbate, N., et al.** 2009 : Human System Interactions, 2009. HSI '09. 2nd Conference on , Catania.

23. **Racelogic.** VBOX (Velocity Box). *velocitybox.co.uk*. [Online] [Cited: 01 08, 2012.] http://www.velocitybox.co.uk/.

24. **Yoshikawa, Tsuneo.** *Foundations of Robotics: Analysis and Control.* s.l. : The MIT Press, 1990. 0262240289.

25. **Ellery, A.** *An Introduction to Space Robotics.* New York : Springer-Verlag, 2000. 185233164X.

26. **Ogata, K.** *Modern Control Engineering.* s.l. : Prentice-Hall, 2000. 0130609072.

27. **Kalman, R.E.** A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering.* 1960.

28. **Welch, G., Bishop, G.** An Introduction to the Kalman Filter. *SIGGRAPH 2001 Course 8.* North Carolina : University of North Carolina at Chapel Hill, Department of Computer Science, 1997.

29. **Higgins, Walter T. Jr.** A Comparison of Complementary and Kalman Filtering. *IEEE Transactions on Aerospace and Electronic Systems.* 1975, Vol. 11, 3.

30. *A Complementary Filter for Attitude Estimation of a Fixed-Wing UAV*. **Euston, M., Coote, P., Mahony, R., Kim, J., Hamel, T.** Nice : IEEE International Conference on Intelligent Robots and Systems, 2008. 978-1-4244-2057-5.

31. **Encyclopaedia Britannica.** Gyroscope. *Encyclopaedia Britannica*. [Online] [Cited: 11 18, 2011.] http://www.britannica.com/EBchecked/topic/250498/gyroscope.

32. **Apostolyuk, Vladislav.** Theory and Design of Micromechanical Vibratory Gyroscopes. *Astrise Corporation.* [Online] [Cited: 12 06, 2011.] http://www.astrise.com/research/library/index.html.

33. *Efficient Design of Micromechanical Gyroscopes.* **Apostolyuk, Vladislav.** s.l. : Journal of Micromechanics and Microengineering, 2002, Vol. 12.

34. **Ferdinand P. Beer, E. Russell Johnston, William E. Clausen.** *Vector Mechanics for Engineers: Dynamics (SI Units).* New York : McGraw-Hill, 2008. 0071273603.

35. *Micromachined Inertial Sensors.* **N. Yazdi, F. Ayazi, K. Najafi.** 8, s.l. : Proceedings of the IEEE, 1998, Vol. 86.

36. *On a New Action of the Magnet on Electric Currents.* **Hall, Edwin Herbert.** Baltimore : American Journal of Mathematics, 1879, Vol. 2.

37. **Honeywell International Inc.** Technical Articles: Magnetic Sensor Overview. *MagneticSensors.com*. [Online] 1998. [Cited: 12 09, 2011.] http://www.magneticsensors.com/literature.php?jumpMenu2=http%3A%2F%2Fwww.ssec.honeywell.com%2Fmp3%2Fmagneticsensors_compassing_applications.mp3&go_button2.x=9&go_button2.y=9&go_button2=Go.

38. **kowoma.de.** Position Determination with GPS. [Online] kowoma.de. [Cited: 11 1, 2011.] http://www.kowoma.de/en/gps/positioning.htm.

39. **U.S. Department of Homeland Security.** USCG Navigation Center. [Online] [Cited: 09 18, 2011.] http://www.navcen.uscg.gov.

40. *Efficient Solution and Performance Analysis od 3-D Position Estimation by Trilateration.* **Manolakis, Dimitris E.** 4, s.l. : IEEE Transactions on Aerospace and Electronic Systems, 1996, Vol. 32.

41. **T. Walter, P. Enge.** *SP-1303 EGNOS: A Cornerstone of Galileo.* Paris : European Space Agency, 2006.

42. **Time Service Department.** USNO NAVSTAR Global Positioning System. [Online] United States Navy, 09 12, 2011. http://tycho.usno.navy.mil/gpsinfo.html.

43. **Trimble Navigation Limited.** Lassen iQ GPS Module Datasheet. Sunnyvale : s.n., 2004.

44. **National Marine Electronics Association.** NMEA Reference Manual revision 1.3. San Jose : s.n., 2005.

45. **Rao, Singeresu S.** *Mechanical Vibrations.* Upper Saddle River New Jerson : Pearson Prentice Hall, 2004. 0-13-048987-5.

46. **wikipedia.org.** Personal Computer. [Online] 11 25, 2011. [Cited: 11 26, 2011.] http://en.wikipedia.org/wiki/Personal_computer.

47. **National Instruments.** National Instruments LabVIEW: What is NI LabVIEW? [Online] National Instruments, 2011. [Cited: 10 14, 2011.] http://www.ni.com/labview/whatis/.

48. **ASUSTeK Computer Inc.** Eee PC 1001PXD Specifications. [Online] [Cited: 12 03, 2011.] http://ca.asus.com/en/Eee/Eee_PC/Eee_PC_1001PXD/#specifications.

49. **Atmel Corporation.** ATxmega16A4U. *Atmel.com*. [Online] [Cited: 01 08, 2012.] http://www.atmel.com/dyn/products/product_card.asp?part_id=17327&category_id=163&family_id=607&subfamily_id=1965.

50. **Sparkfun Electronics.** Header Board for AT90CAN128. *Sparkfun.com*. [Online] [Cited: 01 08, 2012.] http://www.sparkfun.com/products/655.

51. **Honeywell International Inc.** 2-Axis Compass with Algorithms HMC 6352. *honeywell.com*. [Online] 02 2009. [Cited: 10 08, 2011.] http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Missiles-Munitions/HMC6352.pdf.

52. **WebbotLib.org.uk.** WebbotLib. *webbot.org.uk*. [Online] [Cited: 06 15, 2011.] www.WebbotLib.org.uk.

53. —. WebbotLib C Documentation. [Online] [Cited: 12 5, 2011.] http://webbot.org.uk/WebbotLibDocs/.

54. **Arduino Software.** *Arduino.cc*. [Online] [Cited: 08 15, 2011.] http://arduino.cc/en/.

55. **Onset Computer Corporation.** HOBO U12 4-Channel External Data Logger - U12-008. *onsetcomp.com*. [Online] [Cited: 01 08, 2012.] HOBO U12 4-Channel External Data Logger - U12-008.

56. **Moog Crossbow.** Moog Crossbow Overview. [Online] [Cited: 11 3, 2011.] http://www.xbow.com/corporate/about/index.html.

57. **Crossbow Technology Inc.** Crossbow Technology Introduces 320-Series Inertial Systems Starting at <$1,000. San Jose : s.n., 03 30, 2009.

58. —. IMU 320 6DOF Inertial Measurement Unit Datasheet. San Jose : s.n., 2009.

59. **Hoskin Scientific.** Private Communication from Hoskin Scientific: Quotation. Ottawa : s.n., 09 09, 2009.

60. **Xsens Technologies B.V.** MTi Technical Specifications. [Online] [Cited: 10 10, 2011.] http://www.xsens.com/en/general/mti.

61. **Douxchamps, Damien.** A small list of IMU/INS/INU. [Online] 04 16, 2011. http://damien.douxchamps.net/research/imu/.

62. **Garmin Ltd.** eTrex 30 Specifications. [Online] [Cited: 12 03, 2011.] https://buy.garmin.com/shop/shop.do?cID=145&pID=87774&ra=true.

63. **Lowrance Electronics.** Specifications for HDS-10 GPS Chartplotter. [Online] [Cited: 12 03, 2011.] http://www.lowrance.com/Products/Marine/HDS-High-Definition-System/HDS-10-FishfinderGPS-Chartplotter/Specifications/.

64. **Raymarine UK Ltd.** Raymarine's Tacktick Range. [Online] [Cited: 11 23, 2011.] http://www1.raymarine.eu/view/?id=1237.

65. **Velocitek.** Speedpuck Technical Specifications. [Online] [Cited: 12 03, 2011.] http://www.velocitek.com/speedpuck/tech-specs/.

66. **u-Blox AG.** Low Cost, Low Power, High Sensitivity: u-Blox Launches the LEA-4A GPS Module. 2006.

67. **Encyclopaedia Britannica.** Anemometer. *Encyclopeadia Britannica.* [Online] [Cited: 01 08, 2012.] http://www.britannica.com/EBchecked/topic/24295/anemometer.

68. **Pitot, Henri.** Description d'une machine pour mesurer la vitesse des eaux courantes et le sillage des vaisseaux. *Histoire de l'Académie royale des sciences avec les mémoires de mathématique et de physique tirés des registres de cette Académie.* 1732.

69. **Corrigan, Kirsten.** Data Analysis Key Focus for Hannes Arch and Team Abu Dhabi. *RedBullAirRace.com*. [Online] 04 27, 2010. [Cited: 01 08, 2012.] http://www.redbullairrace.com/cs/Satellite/en_air/Article/Data-analysis-key-focus-for-Hannes-Arch-and-Team-Abu-Dhabi-021242837984758.

70. **Fédération Internationale de l'Automobile.** Formula 1 Electrical Systems Technical Regulations. *formula1.com*. [Online] [Cited: 01 08, 2012.] http://www.formula1.com/inside_f1/rules_and_regulations/technical_regulations/8701/fia.html.

71. **al., Galvin et.** *Micromechanical Accelerometer for Automotive Applications. 6,149,190* United States of America, 11 21, 2000.

72. **al., Goodenough et.** Airbags boom when IC accelerometer sees 50g. *Electronic Design.* 1991.

73. **William F. Milliken, Douglas L. Milliken.** *Race Car Vehicle Dynamics.* s.l. : Society of Automotive Engineers, 1995. 1560915269.

74. **MacDonald, G.A.** A Review of Low Cost Accelerometers for Vehicle Dynamics. *Sensors and Actuators A: Physical.* 1990, Vol. 21, 1-3.

75. *Dual-Axes Capacitive Inclinometer/Low-g Accelerometer for Automotive Applications.* **Lapadatu, D., Habibi, S., Reppen, B., Salomonsen, G., Kvisteroy, T.** Interlaken : IEEE, 2001. 0-7803-5998-4.

76. **Sheppard, Mr. Van.** Private Communication. Ottawa : s.n., 2008.

77. **United Kingdom Meteorological Office.** Beaufort wind force scale. [Online] [Cited: 12 11, 2011.] http://www.metoffice.gov.uk/weather/marine/guide/beaufortscale.html.

78. **Nepean Sailing Club.** Racing, What NSC is All About! [Online] 2011. [Cited: 12 03, 2011.] http://nsc.ca/racing.

79. **Dave Baum, Michael Gasperi, Ralph Hempel, Luis Villa.** *Extreme Mindstorms: an Advanced Guide to LEGO Mindstorms.* New York : Springer-Verlag New York, Inc., 2002.

80. **LEGO Group.** LEGO Mindstorms NXT Hardware Developer Kit V1.00 Manual. 2006.

81. **Hitechnic Division, Dataport Systems, Inc.** *hitechnic.com*. [Online] [Cited: 01 08, 2012.] http://www.hitechnic.com/.

82. **Palmisano, John: Society of Robots.** AxonII. *Society of Robots.* [Online] [Cited: 11 24, 2010.] http://www.societyofrobots.com/axon2/.

83. **Palmisano, John.** AxonII Datasheet. [Online] [Cited: 10 1, 2011.] http://www.societyofrobots.com/axon2/axon2_datasheet.shtml.

84. **Society of Robots.** Robot Forum. [Online] http://www.societyofrobots.com/robotforum/.

85. **Arduino Software.** Arduino Mega 2560. *Arduino.cc*. [Online] [Cited: 08 15, 2011.] http://arduino.cc/en/Main/ArduinoBoardMega2560.

86. **Sparkfun Electronics.** IMU Analog Combo Board Razor - 6DOF Ultra-Thin IMU. [Online] [Cited: 11 10, 2011.] http://www.sparkfun.com/products/9431.

87. **Analog Devices, Inc.** Analog Devices and Nintendo Collaboration Drives Video Game Innovation with IMEMS Motion Signal Processing Technology. *Analog.com*. [Online] 05 09, 2006. http://www.analog.com/en/press-release/May_09_2006_ADI_Nintendo_Collaboration/press.html.

88. **Sparkfun Electronics.** Compass Module - HMC6352. [Online] [Cited: 9 25, 2011.] http://www.sparkfun.com/products/7915.

89. —. Venus GPS with SMA Connector. [Online] [Cited: 10 4, 2011.] http://www.sparkfun.com/products/9133.

90. —. Antenna GPS 3V Magnetic Mount SMA. [Online] [Cited: 9 13, 2011.] http://www.sparkfun.com/products/464.

91. —. Weather Meters. [Online] [Cited: 9 16, 2011.] http://www.sparkfun.com/products/8942.

92. **Argent Data Systems.** Weather Sensor Assembly p/n 80422. *Argent Data Systems*. [Online] [Cited: 11 24, 2010.] http://www.sparkfun.com/datasheets/Sensors/Weather/Weather%20Sensor%20Assembly.pdf.

93. **Sparkfun Electronics.** Breakout Board for SD-MMC Cards. [Online] [Cited: 9 21, 2011.] http://www.sparkfun.com/products/204.

94. **Pelican Products.** Pelican Products 1120 Case. *Pelican.com*. [Online] [Cited: 01 08, 2012.] http://pelican.com/cases_detail.php?Case=1120.

95. **Compagnie Deutsch GmbH.** DT Series Connectors. [Online] [Cited: 5 16, 2011.] http://www.deutsch.net/deutsch-product-search.aspx?page=3&pid=589.

96. **HobbyKing.comTurnigy nano-tech 1000mAh 2S 25-50C LiPo Pack.** [Online] [Cited: 12 3, 2011.] http://www.hobbyking.com/hobbyking/store/__11900__Turnigy_nano_tech_1000mah_2S_25_50C_Lipo_Pack.html.

97. **Sparkfun Electronics.** Polymer Lithium Ion Battery7 - 1000mAh 7.4V. [Online] [Cited: 12 3, 2011.] http://www.sparkfun.com/products/10472.

98. **Suunto Oy.** Suunto A-10 CM. *Suunto.com*. [Online] [Cited: 01 08, 2012.] http://www.suunto.com/global/en/products/Fieldcompasses/suunto-a-10/suunto-a-10-cm.

99. **Campbell Scientific, Inc.** WM-200 WindmMate with Wind Direction. *WeatherHawk.com*. [Online] [Cited: 01 18, 2011.] http://www.weatherhawk.com/wm200.

100. **Bergeron, Alexandre.** Alex's Bike Tests. *youtube.com*. [Online] 05 11, 2011. http://www.youtube.com/watch?v=8aDK9njbZoc.

101. **International Sailing Federation.** *2009 International J/24 Class Association.* 2009.

102. **J/24 Class Association.** J/24 Dimensions. *jboats.com*. [Online] [Cited: 11 28, 2011.] http://www.jboats.com/j24/j24dimensions.htm.

103. **National Climate Data and Information Archive.** Hourly Wind Speed for September 15, 2011 Ottawa Macdonald-Cartier Int'l Airport. *www.climate.weatheroffice.gc.ca*. [Online] 09 15, 2011. [Cited: 09 16, 2011.] http://www.climate.weatheroffice.gc.ca/climateData/generate_chart_e.html?StationID=4337&Month=9&Year=2011&timeframe=1&Day=15&type=line&MeasTypeID=windspd.

104. **Atmel Corporation.** ATmega640 Overview. [Online] [Cited: 10 15, 2011.] http://www.atmel.com/dyn/products/product_card.asp?PN=ATmega640.

105. **Analog Devices, Inc.** ADXL335: Small, Low Power, 3-Axis ±3 g Accelerometer. *Analog.com*. [Online] 01 2010. [Cited: 11 08, 2011.] http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf.

106. **STMicroelectronics N.V.** LY530ALH MEMS motion sensor: high performance ±300 °/s analog yaw-rate gyroscope. *ST.com*. [Online] 07 2009. [Cited: 10 08, 2011.] http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00237186.pdf.

107. **STMicrolectronics N.V.** LPR530AL MEMS motion sensor: dual axis pitch and roll ±300°/s analog gyroscope. *ST.com*. [Online] 07 2009. [Cited: 10 08, 2011.] http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00237209.pdf.

108. **On Shine Enterprise CO., Ltd.** ONSHINE G.P.S. Antenna ANT-555. *onshine.com.tw*. [Online] [Cited: 07 01, 2011.] http://php2.twinner.com.tw/files/onshine/ANT555-2006-NEW.pdf.

109. **Mountain Equipment Co-op.** Pelican 1120 Case. [Online] [Cited: 11 29, 2011.] http://www.mec.ca/AST/ShopMEC/Paddling/BagsPacksCases/Cases/PRD~4009-434/pelican-1120-case.jsp.

110. **Digi-Key Corporation.** WM2512-ND Pin. [Online] [Cited: 12 14, 2011.] http://search.digikey.com/ca/en/products/16-02-0103/WM2512-ND/115063.

111. —. WM2800-ND 2 Pin Connector Housing. [Online] [Cited: 12 14, 2011.] http://search.digikey.com/scripts/dksearch/dksus.dll?vendor=0&keywords=WM2800-ND+.

112. —. WM2801-ND 3 Pin Connector Housing. [Online] http://search.digikey.com/scripts/dksearch/dksus.dll?vendor=0&keywords=WM2801-ND.

113. —. WM2802-ND 4 Pin Connector Housing. [Online] http://search.digikey.com/scripts/dksearch/dksus.dll?vendor=0&keywords=WM2802-ND.

114. **Newark Electronics.** NTE1904. [Online] [Cited: 12 14, 2011.] http://www.newark.com/jsp/search/productdetail.jsp?SKU=31C3394&CMP=AFC-SF-T11.

115. **Palmisano, John.** Electronics Wire Connector Tutorial. [Online] [Cited: 12 14, 2011.] http://www.societyofrobots.com/electronics_wire_connector.shtml.

116. **Silicon Laboratories Inc.** CP210x USB to UART Bridge VCP Drivers. *SiLabs.com*. [Online] [Cited: 11 28, 2011.] https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx.

117. **Atmel Corporation.** AVR Studio 4. *Atmel.com*. [Online] [Cited: 11 05, 2011.] http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725.

118. **Palmisano, John.** AxonII Getting Started. *SocietyOfRobots*. [Online] [Cited: 06 15, 2011.] http://www.societyofrobots.com/axon2/axon2_setup1.shtml.

119. **WebbotLib.org.uk.** WebbotLib Project Designer. *webbot.org.uk*. [Online] [Cited: 01 08, 2012.] http://webbot.org.uk/iPoint/37.page.

120. **White Mountain Sailing.** Parts of the Sailboat. *whitemountainsailing.com*. [Online] 04 15, 2009. http://whitemountainsailing.com/parts.aspx.

121. **Laven, Kate.** telegraph.co.uk. [Online] 11 23, 2011. [Cited: 11 23, 2011.] http://www.telegraph.co.uk/sport/othersports/sailing/volvo-ocean-race/8911061/Volvo-Ocean-Race-2011-Puma-Ocean-Racing-skipper-denies-crew-pushed-Mar-Mostro-too-hard.html.

122. **VolvoOceanRace.** Ticking off the to-do list. [Online] 11 30, 2011. http://www.volvooceanrace.com/en/news/4247_Ticking-off-the-to-do-list.html.

Appendix A - Device Specifications

A.1. Society of Robots AxonII Specifications

The specifications for the AxonII microcontroller are given in Table 42 and specifications for the Atmel ATmega 640 chip are shown in Table 43. For additional information on the AxonII microcontroller see [83].

Table 42: AxonII Specifications [83]

58 I/O Total
16 ADC
25+ Servos
I2C, SPI
3 UART + USB
Up to 8 external interrupts
15 PWM Channels
64KB Flash, 4KB EEPROM, 8KB SRAM
16 MIPS throughput at 16 MHz
6 Timers (four 16-bit, two 8-bit)
pre-programmed with a bootloader - no programmer required
numerical LED display
built in 3.3V, 5V, and unregulated power buses
external memory support (port A)
***all* software is free**
100% open source, large support community
Windows, Mac, and Linux compatible

Table 43: ATmega 640 Specifications [104]

Parameter	Value
Flash (Kbytes):	64
Pin Count:	100
Max. Operating Frequency:	16
CPU:	8-bit AVR
# of Touch Channels:	16
Hardware QTouch Acquisition:	no
Max I/O Pins:	86
Ext Interrupts:	32
Quadrature Decoder Channels:	0
USB Transceiver:	0
USB Speed:	No
USB Interface:	No
SPI:	5
TWI (I2C):	1
UART:	4
CAN:	0
LIN:	0
SSC:	0
Ethernet:	0
SD / eMMC:	0
Segment LCD:	0
Graphic LCD:	no
Video Decoder:	no
Camera Interface:	no
ADC Channels:	16
ADC Resolution (bits):	10
ADC Speed (ksps):	15

Analog Comparators:	1
Resistive Touch Screen:	no
DAC Channels:	0
DAC Resolution (bits):	0
Temp. Sensor:	no
SRAM (Kbytes):	8
EEPROM (Bytes):	4096
Self Program Memory:	yes
External Bus Interface:	0
DRAM Memory:	no
NAND Interface:	no
picoPower:	no
Temp. Range (deg C):	-40 to 85
I/O Supply Class (V):	1.8 to 5.5
Operating Voltage (Vcc):	1.8 to 5.5
FPU:	no
MPU / MMU:	no / no
Crypto Engine:	no
Timers:	6
Output Compare Channels:	16
Input Capture Channels:	4
PWM Channels:	15
32kHz RTC:	yes
Calibrated RC Oscillator:	yes

A.2. Analog Devices® ADXL335 Specifications

These specifications are extracted from [105].

http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf

SPECIFICATIONS

T_A = 25°C, V_S = 3 V, C_X = C_Y = C_Z = 0.1 µF, acceleration = 0 *g*, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		±3	±3.6		*g*
Nonlinearity	% of full scale		±0.3		%
Package Alignment Error			±1		Degrees
Interaxis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity[1]			±1		%
SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	V_S = 3 V	270	300	330	mV/*g*
Sensitivity Change Due to Temperature[3]	V_S = 3 V		±0.01		%/°C
ZERO *g* BIAS LEVEL (RATIOMETRIC)					
0 *g* Voltage at X_{OUT}, Y_{OUT}	V_S = 3 V	1.35	1.5	1.65	V
0 *g* Voltage at Z_{OUT}	V_S = 3 V	1.2	1.5	1.8	V
0 *g* Offset vs. Temperature			±1		mg/°C
NOISE PERFORMANCE					
Noise Density X_{OUT}, Y_{OUT}			150		µg/√Hz rms
Noise Density Z_{OUT}			300		µg/√Hz rms
FREQUENCY RESPONSE[4]					
Bandwidth X_{OUT}, Y_{OUT}[5]	No external filter		1600		Hz
Bandwidth Z_{OUT}[5]	No external filter		550		Hz
R_{FILT} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF-TEST[6]					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		µA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	−150	−325	−600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	V_S = 3 V		350		µA
Turn-On Time[7]	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		−40		+85	°C

[1] Defined as coupling between any two axes.
[2] Sensitivity is essentially ratiometric to V_S.
[3] Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.
[4] Actual frequency response controlled by user-supplied external filter capacitors (C_X, C_Y, C_Z).
[5] Bandwidth with external capacitors = 1/(2 × π × 32 kΩ × C). For C_X, C_Y = 0.003 µF, bandwidth = 1.6 kHz. For C_Z = 0.01 µF, bandwidth = 500 Hz. For C_X, C_Y, C_Z = 10 µF, bandwidth = 0.5 Hz.
[6] Self-test response changes cubically with V_S.
[7] Turn-on time is dependent on C_X, C_Y, C_Z and is approximately 160 × C_X or C_Y or C_Z + 1 ms, where C_X, C_Y, C_Z are in microfarads (µF).

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration (Any Axis, Unpowered)	10,000 *g*
Acceleration (Any Axis, Powered)	10,000 *g*
V_S	−0.3 V to +3.6 V
All Other Pins	(COM − 0.3 V) to (V_S + 0.3 V)
Output Short-Circuit Duration (Any Pin to Common)	Indefinite
Temperature Range (Powered)	−55°C to +125°C
Temperature Range (Storage)	−65°C to +150°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

A.3. STMicroelectronics™ LY530ALH Specifications

The specifications for the STMicroelectronics™ LY530ALH are found in [106].

http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00237186.pdf

LY530ALH

MEMS motion sensor:
high performance ±300 °/s analog yaw-rate gyroscope

Preliminary data

Features

- 2.7 V to 3.6 V single-supply operation
- Wide operating temperature range (-40 °C to +85 °C)
- High stability overtemperature
- Analog absolute angular-rate output
- Two separate outputs (1x and 4x amplified)
- Integrated low-pass filters
- Low power consumption
- Embedded power-down
- Embedded self-test
- High shock and vibration survivability
- ECOPACK® RoHS and "Green" compliant (see *Section 5*)

Applications

- Pointing devices, remote and game controllers
- Motion control with user interface
- GPS navigation systems
- Industrial and robotics

Description

The LY530ALH is a low-power single-axis micromachined gyroscope capable of measuring angular rate along yaw axis.

It provides excellent temperature stability and high resolution over an extended operating temperature range (-40 °C to +85 °C).

LGA-16L (5x5x1.5mm)

The LY530ALH has a full scale of ±300 °/s and is capable of detecting rates with a -3 dB bandwidth up to 140 Hz.

The gyroscope is the combination of one actuator and one accelerometer integrated in a single micromachined structure.

It includes a sensing element composed by single driving mass, kept in continuos oscillating movement and able to react when an angular rate is applied based on the Coriolis principle.

A CMOS IC provides the measured angular rate to the external world through an analog output voltage, allowing high level of integration and production trimming to better match sensing element characteristics.

ST's gyroscope family leverages on robust and mature manufacturing process already used for the production of micromachined accelerometers.

ST is already in the field with several hundreds million sensors with excellent acceptance from the market in terms of quality, reliability and performance.

LY530ALH is provided in plastic land grid array (LGA) package. Several years ago ST pioneered successfully the usage of this package for accelerometers. Today ST has the widest manufacturing capability and strongest expertise in the world for production of sensor in plastic LGA package.

2.1 Mechanical characteristics

Table 3. Mechanical characteristics @ Vdd = 3 V, T = 25 °C unless otherwise noted(1)

Symbol	Parameter	Test condition	Min.	Typ.(2)	Max.	Unit
FSA	Measurement range	4x OUT (amplified)		±300		°/s
FS		OUT (not amplified)		±1200		°/s
SoA	Sensitivity(3)	4x OUT (amplified)		3.33		mV/ °/s
So		OUT (not amplified)		0.83		mV/ °/s
SoDr	Sensitivity change vs temperature	Delta from 25°C		0.05		%/°C
Voff	Zero-rate level(3)			1.23		V
Vref	Reference voltage			1.23		V
OffDr	Zero-rate level change Vs temperature	Delta from 25°C		0.05		°/s/°C
NL	Non linearity	Best fit straight line		±1		% FS
BW	Bandwidth(4)			140		Hz
Rn	Rate noise density			0.035		°/s / $\sqrt{Hz}$
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3 V. The operational power supply range is specified in *Table 4*.
2. Typical specifications are not guaranteed
3. Sensitivity and zero-rate level are not ratiometric to supply voltage
4. The product is capable of measuring angular rates extending from DC to the selected BW.

2.2 Electrical characteristics

Table 4. Electrical characteristics @ Vdd =3 V, T=25 °C unless otherwise noted(1)

Symbol	Parameter	Test condition	Min.	Typ.(2)	Max.	Unit
Vdd	Supply voltage		2.7	3	3.6	V
Idd	Supply current	PD pin connected to GND		5	5.5	mA
IddPdn	Supply current in power-down mode	PD pin connected to Vdd		1	5	µA
V_{ST}	Self-test input	Logic 0 level	0		0.2*Vdd	V
		Logic 1 level	0.8*Vdd		Vdd	
V_{PD}	Power-down input	Logic 0 level	0		0.2*Vdd	V
		Logic 1 level	0.8*Vdd		Vdd	
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3 V
2. Typical specifications are not guaranteed

2.3 Absolute maximum ratings

Stresses above those listed as "Absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 5. Absolute maximum ratings

Symbol	Ratings	Maximum value	Unit
Vdd	Supply voltage	-0.3 to 6	V
Vin	Input voltage on any control pin (PD, ST)	-0.3 to Vdd +0.3	V
A	Acceleration	3000 *g* for 0.5 ms	
		10000 *g* for 0.1 ms	
T_{STG}	Storage temperature range	-40 to +125	°C
ESD	Electrostatic discharge protection	2 (HBM)	kV

This is a mechanical shock sensitive device, improper handling can cause permanent damage to the part

This is an ESD sensitive device, improper handling can cause permanent damage to the part

A.4. STMicroelectronics™ LPR530AL Specifications

The specifications for the STMicroelectronics™ LPR530AL are found in [107].

http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00237209.pdf

LPR530AL

MEMS motion sensor:
dual axis pitch and roll ±300°/s analog gyroscope

Preliminary data

Features

- 2.7 V to 3.6 V single-supply operation
- Wide operating temperature range (-40 °C to +85 °C)
- High stability overtemperature
- Analog absolute angular-rate output
- Two separate outputs for each axis (1x and 4x amplified)
- Integrated low-pass filters
- Low power consumption
- Embedded power-down
- Embedded self-test
- High shock and vibration survivability
- ECOPACK® RoHS and "Green" compliant (see *Section 5*)

Applications

- Pointing devices, remote and game controllers
- Motion control with user interface
- GPS navigation systems
- Industrial and robotics

Description

The LPR530AL is a low-power dual-axis micromachined gyroscope capable of measuring angular rate along pitch and roll axes.

It provides excellent temperature stability and high resolution over an extended operating temperature range (-40 °C to +85 °C).

LGA-16L (5x5x1.5mm)

The LPR530AL has a full scale of ±300 °/s and is capable of detecting rates with a -3 dB bandwidth up to 140 Hz.

The gyroscope is the combination of one actuator and one accelerometer integrated in a single micromachined structure.

It includes a sensing element composed by single driving mass, kept in continuous oscillating movement and able to react when an angular rate is applied based on the Coriolis principle.

A CMOS IC provides the measured angular rate to the external world through an analog output voltage, allowing high level of integration and production trimming to better match sensing element characteristics.

ST's gyroscope family leverages on robust and mature manufacturing process already used for the production of micromachined accelerometers.

ST is already in the field with several hundreds million sensors with excellent acceptance from the market in terms of quality, reliability and performance.

LPR530AL is provided in plastic land grid array (LGA) package. Several years ago ST pioneered successfully the usage of this package for accelerometers. Today ST has the widest manufacturing capability and strongest expertise in the world for production of sensor in plastic LGA package.

2.1 Mechanical characteristics

Table 3. Mechanical characteristics @ Vdd = 3 V, T = 25 °C unless otherwise noted(1)

Symbol	Parameter	Test condition	Min.	Typ.(2)	Max.	Unit
FSA	Measurement range	4x OUT (amplified)		±300		°/s
FS		OUT (not amplified)		±1200		°/s
SoA	Sensitivity(3)	4x OUT (amplified)		3.33		mV/ °/s
So		OUT (not amplified)		0.83		mV/ °/s
SoDr	Sensitivity change vs temperature	Delta from 25°C		0.03		%/°C
Voff	Zero-rate level(3)			1.23		V
Vref	Reference voltage			1.23		V
OffDr	Zero-rate level change Vs temperature	Delta from 25°C		0.05		°/s/°C
NL	Non linearity	Best fit straight line		±1		% FS
BW	Bandwidth(4)			140		Hz
Rn	Rate noise density			0.035		°/s / $\sqrt{Hz}$
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3 V. The operational power supply range is specified in *Table 4*.
2. Typical specifications are not guaranteed
3. Sensitivity and zero-rate level are not ratiometric to supply voltage
4. The product is capable of measuring angular rates extending from DC to the selected BW.

2.2 Electrical characteristics

Table 4. Electrical characteristics @ Vdd =3 V, T=25 °C unless otherwise noted(1)

Symbol	Parameter	Test condition	Min.	Typ.(2)	Max.	Unit
Vdd	Supply voltage		2.7	3	3.6	V
Idd	Supply current	PD pin connected to GND		6.8		mA
IddPdn	Supply current in power-down mode	PD pin connected to Vdd		1	5	µA
V_{ST}	Self-test input	Logic 0 level	0		0.2*Vdd	V
		Logic 1 level	0.8*Vdd		Vdd	
V_{PD}	Power-down input	Logic 0 level	0		0.2*Vdd	V
		Logic 1 level	0.8*Vdd		Vdd	
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3 V
2. Typical specifications are not guaranteed

2.3 Absolute maximum ratings

Stresses above those listed as "Absolute maximum ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 5. Absolute maximum ratings

Symbol	Ratings	Maximum value	Unit
Vdd	Supply voltage	-0.3 to 6	V
Vin	Input voltage on any control pin (PD, ST)	-0.3 to Vdd +0.3	V
T_{STG}	Storage temperature range	-40 to +125	°C
A	Acceleration	3000 *g* for 0.5 ms	
		10000 *g* for 0.1 ms	
ESD	Electrostatic discharge protection	2 (HBM)	kV


This is a mechanical shock sensitive device, improper handling can cause permanent damage to the part


This is an ESD sensitive device, improper handling can cause permanent damage to the part

A.5. Honeywell HMC6352 Specifications

The specifications for the Honeywell HMC6352 magnetic compass are obtained from [51].

http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Missiles-Munitions/HMC6352.pdf

HMC6352 SPECIFICATIONS

Characteristics	Conditions (1)	Min	Typ	Max	Units
Supply Voltage	Vsupply to GND	2.7	3.0	5.2	Volts
Supply Current	Vsupply to GND				
	Sleep Mode (Vsupply = 3.0V)		1		µA
	Steady State (Vsupply = 3.0V)		1		mA
	Steady State (Vsupply = 5.0V)		2	10	mA
	Dynamic Peaks				mA
Field Range (2)	Total applied field	0.10	-	0.75	gauss
Heading Accuracy	HMC6352		2.5		degRMS
Heading Resolution			0.5		deg
Heading Repeatability			1.0		deg
Disturbing Field	Sensitivity starts to degrade. Enable set/reset function to restore sensitivity.	20			gauss
Max. Exposed Field	No permanent damage and set/reset function restores performance.			10000	gauss
Operating Temperature	Ambient	-20		70	°C
Storage Temperature	Ambient	-55		125	°C
Peak Reflow Temperature	For Lead-Free SMT Reflow	230	-	240	°C
Moisture Sensivity	Max 240°C		MSL3		-
Output	Heading, Mag X, Mag Y				
Size	6.5 x 6.5 x 1.5				mm
Weight			0.14		grams

(1) Tested at 25°C except stated otherwise.
(2) Field upper limit can be extended by using external resistors across CA1/CA2 and CB1/CB2.

A.6. Venus 634LPx Specifications

The specifications for the SkyTraq Venus 634LPx are obtained from [89].

http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/GPS/Venus638FLPx.pdf

Data Sheet **SKYTRAQ**

VENUS634FLPx 65 Channel Low Power GPS Receiver --- Flash

FEATURES

- GPS receiver in 10mm x 10mm x 1.1mm size
- Tests 8 million time-frequency hypothesis per sec
- Open sky cold start 29 second
- Signal detection better than -161dBm
- Accuracy 2.5m CEP
- ~28mA in tracking and navigation mode
- Multipath detection and suppression
- Data logging with external SPI serial Flash
- Supports active or passive antenna
- 1 SPI interface and 7 GPIO lines
- Flash-based, firmware customizable
- LGA44 package with 0.8 pitch
- Pb free RoHS compliant

The Venus634FLPx is a module-in-a-chip design targeting mobile consumer and cellular handset applications. It offers very low current consumption, high sensitivity, and best in class signal acquisition and time-to-first-fix performance.

The Venus634FLPx contains all the necessary components of a complete GPS receiver module, includes GPS RF front-end, GPS baseband signal processor, 0.5ppm TCXO, 32.768kHz RTC crystal, RTC LDO regulator, and passive components. It requires very low external component count and takes up only 100mm^2 PCB footprint.

Dedicated massive-correlator signal parameter search engine within the baseband enables rapid search of all the available satellites and acquisition of very weak signal. An advanced track engine allows weak signal tracking and positioning in harsh environments such as urban canyons and under deep foliage.

Programmable flash-based memory makes it ideal for applications requiring customized firmware.

TECHNICAL SPECIFICATIONS

Receiver Type	L1 Frequency GPS C/A code SBAS Capable 51 Channel Acquisitions 14 Channel Tracking
Accuracy	Position 2.5m CEP Velocity 0.1m/sec Timing 300ns
Open Sky TTFF	Hot start 1 second Cold start 29 seconds average
Reacquisition	< 1s
Sensitivity	Tracking -161dBm
Update Rate	1Hz standard
Dynamics	4G
Operational Limits	Altitude < 18,000m[*1] or Velocity < 515m/s[*1]
Datum	Default WGS-84
Interface	UART LVTTL level
Baud Rate	4800 / 9600 / 38400 / 115200 software configurable (9600 as default)
Protocol	NMEA-0183 V3.01, GGA, GLL, GSA, GSV, RMC, VTG (default GGA, GSA, GSV, RMC, VTG) SkyTraq Binary
Main Supply Voltage	2.8V ~ 3.6V
Backup Voltage	1.5V ~ 6V
Current Consumption	~28mA tracking
Operating Temperature	-40 ~ +85 deg-C
Storage Temperature	-40 ~ +125 deg-C
Package	LGA44 10mm x 10mm x 1.1mm, 0.8mm pitch

A.7. ONSHINE GPS Antenna ANT-555 Specifications

The specifications for the ONSHINE GPS Antenna ANT-555 are obtained from [108].

http://php2.twinner.com.tw/files/onshine/ANT555-2006-NEW.pdf

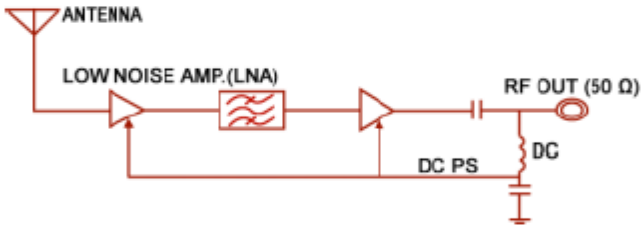
■ Features

- High Gain and Low Noise Figure
- Ultra Low Power Consumption
- Varied Cable Length and Connectors are available
- Compact Construction
- Excellent Temperature Stability
- Magnetic Mount Base

■ Application

- Car Navigation System
- Hand-held GPS system

■ Block Diagram



■ Specifications

Patch Antenna

Type	
Frequency	1575.42MHz +/-3MHz
Ground Plane	70x70mm
Bandwith(at10dB return loss)	10MHz min
Gain at Zenith	+4.5dBi typ
Axial Ratio	3.0 dB max
VSWR	1.5 dB max
Impedance	50Ω
Operating Temperature	-40℃~+85℃

TEL:886-2-22496895 FAX:886-2-22496175 http:// www.onshine.com.tw

LNA

Type	
System Gain without cable	27 dB typ @2.7V(+25℃)
Noise Figure	1.5 dB typ (+25℃)
Output VSWR	2.0 dB max
Out of band rejection	Fo = 1575.42 MHz
	Fo ± 50 MHz 25db typical
	Fo ± 100 MHz 35db typical
Voltage	2.7V~5V
Current Consumption	8 mA typical@2.7V
Cable Length	RG-174-2, 3 , 5.. meters
Connector	SMA , MCX , MMCX...
Operating Temperature	-40℃~+85℃

Others

Type	
Dimension	L: 46mm W:46mm H:13.7mm
Weight	80.0g (Cable Length 2M)
Recommended Storage Condition	Temperature -20℃ to 45℃
	Humidity 80% max
Waterproof	100% waterproof
Plastic color	Black

A.8. Specifications of the Argent Data Systems Weather Sensor Assembly

The specifications and user instructions for the Argent Data Systems Weather Sensor Assembly are lifted from [91] and [92]

http://www.sparkfun.com/datasheets/Sensors/Weather/Weather%20Sensor%20Assembly..pdf.

Anemometer

The cup-type anemometer measures wind speed by closing a contact as a magnet moves past a switch. A wind speed of 1.492 MPH (2.4 km/h) causes the switch to close once per second.

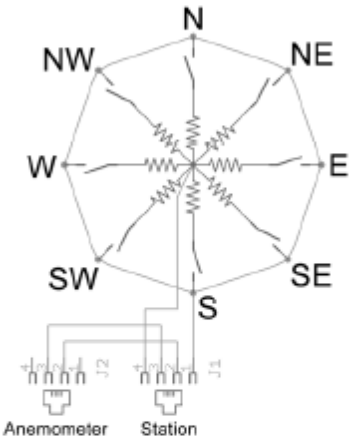
The anemometer switch is connected to the inner two conductors of the RJ11 cable shared by the anemometer and wind vane (pins 2 and 3.)

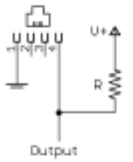
Wind Vane

The wind vane is the most complicated of the three sensors. It has eight switches, each connected to a different resistor. The vane's magnet may close two switches at once, allowing up to 16 different positions to be indicated. An external resistor can be used to form a voltage divider, producing a voltage output that can be measured with an analog to digital converter, as shown below.

The switch and resistor arrangement is shown in the diagram to the right. Resistance values for all 16 possible positions are given in the table.

Resistance values for positions between those shown in the diagram are the result of two adjacent resistors connected in parallel when the vane's magnet activates two switches simultaneously.





Example wind vane interface circuit. Voltage readings for a 5 volt supply and a resistor value of 10k ohms are given in the table.

Direction (Degrees)	Resistance (Ohms)	Voltage (V=5v, R=10k)
0	33k	3.84v
22.5	6.57k	1.98v
45	8.2k	2.25v
67.5	891	0.41v
90	1k	0.45v
112.5	688	0.32v
135	2.2k	0.90v
157.5	1.41k	0.62v
180	3.9k	1.40v
202.5	3.14k	1.19v
225	16k	3.08v
247.5	14.12k	2.93v
270	120k	4.62v
292.5	42.12k	4.04v
315	64.9k	4.78v
337.5	21.88k	3.43v

Appendix B - Programming and Scripts

B.1. PC Interface

B.1.1. AxonII Program

The user generated code portion of the program running on the AxonII for the USB interface mode is presented here. The function hardware.h is defined by Webbotlib depending on the hardware attached to the AxonII board.

```
#define RPRINTF_FLOAT
#include "hardware.h"

//Declaring and initialising  the timer variables
TICK_COUNT StartTimer;
TICK_COUNT EndTimer;

// Initialise  the hardware
void appInitHardware(void)
{
        initHardware();
}

// Initialise  the software
TICK_COUNT appInitSoftware(TICK_COUNT loopStart)
{
        return 0;
}
// This is the main loop
TICK_COUNT appControl(LOOP_COUNT loopCount, TICK_COUNT loopStart)
{
        // -------- Start GPS -------

        gpsNMEAprocess(&MyGPS);
        if(MyGPS.gps.info.valid)
        {
                // We have got a signal
                // Data is stored in gps.info - see Sensors/GPS/gps.h
                rprintfFloat(6, MyGPS.gps.info.fixTime);
                rprintf(","); rprintfFloat(8,MyGPS.gps.info.latitude);
                rprintf(","); rprintfFloat(8,MyGPS.gps.info.longitude);
                rprintf(","); rprintfFloat(4,MyGPS.gps.info.speed);
                rprintf(","); rprintfFloat(3,MyGPS.gps.info.track);
        }
```

```
else
{
        rprintf("0,0,0,0,0");
}
// -------- End   GPS -------

// -------- Start HMC6352 Compass-------
// Read the HMC6352 Compass and store results
compassRead(compass);
// The bearing is stored in compass.compass.bearingDegrees;
rprintf(",%d",  compass.compass.bearingDegrees);
// -------- End   HMC6352 Compass-------

// -------- Start ADXL335 3 Axis Accelerometer-------
uint16_t  Ax = a2dConvert10bit(ADC_CH_ADC1);
uint16_t  Ay = a2dConvert10bit(ADC_CH_ADC2);
uint16_t  Az = a2dConvert10bit(ADC_CH_ADC3);
rprintf(",%d,%d,%d",  Ax, Ay, Az);
// -------- End   ADXL335 3 Axis Accelerometer-------

// -------- Start LPR530AL 2 axis-------
uint16_t  Gx = a2dConvert10bit(ADC_CH_ADC4);
uint16_t  Gy = a2dConvert10bit(ADC_CH_ADC5);
rprintf(",%d,%d",  Gx, Gy);
// -------- End   LPR530AL 2 axis-------

// -------- Start LY530AL 1 axis-------
uint16_t  Gz = a2dConvert10bit(ADC_CH_ADC6);
rprintf(",%d",  Gz);
// -------- End   LY530AL 1 axis-------

// -------- Start ADC 14 Wind Vane-------
uint16_t  WindDirection = a2dConvert10bit(ADC_CH_ADC14);
rprintf(",%d",WindDirection);
// -------- End   ADC 14 Wind Vane-------

// -------- Start ADC 15 Anemometer -------
uint16_t  WindSpeed = a2dConvert10bit(ADC_CH_ADC15);
rprintf(",%d",WindSpeed);
// -------- End   ADC 15 Anemometer -------

//Get timer final value in microseconds
EndTimer = clockGetus();

//Output interval and size of interval for debugging
rprintf(",%u",(EndTimer  - StartTimer));
```

```
        //Carriage return
        rprintfCRLF();

        //Get initial timer value in microseconds
        StartTimer = EndTimer;

        return 0;
}
```

B.1.2. Matlab Script

This Matlab script reads the data from the AxonII for a defined time period and stores it to a Matlab Structure format.

```
%% Defining script variables

%USER INPUT VARIABLES HERE
%***********************************************************
CaptureTimeCheck = exist('CaptureTime','var'); % Check if CaptureTime exists.
if CaptureTimeCheck == 0 % Zero if it does not.
   CaptureTimeStr = input('Enter the desired capture time in minutes: ','s');
   CaptureTime = str2num(CaptureTimeStr);
   while isempty(CaptureTime)
      disp('Invalid entry!');
      CaptureTimeStr = input('Enter the desired capture time in minutes: ','s');
      CaptureTime = str2num(CaptureTimeStr);
   end
end

TestIndexCheck = exist('TestIndex','var'); % Check if TestIndex exists.
if TestIndexCheck == 0 % Zero if it does not.
   TestIndex = 0; % Create it and set to zero.
end
TestIndex = TestIndex + 1; % Increment to next index.
%***********************************************************

%Number of quantities being measured.
%Last measured quantity is assumed to be the time interval.
MeasuredQuantities = 15;

%Serial port related parameters.
Axon = serial ('COM3', 'BaudRate', 115200); % Setting the COM port number and BaudRate
Axon.InputBufferSize = 2000; % COM port input buffer size
Axon.Terminator = ('CR'); % Set terminator property to "Carriage Return", this shows the end of the lines
sent out by the Axon

%Calculating matrix sizes for presizing later.
MeasurementFrequency = 100;   %Axon measurement frequency in Hz (anticipated or estimated)
```

```
n = CaptureTime * 60 / (1/MeasurementFrequency); % n is the number of readings to take

%% Opening the connection and reading two lines, to ensure synchronisation
fopen(Axon);
Line = fgets(Axon);
Line = fgets(Axon);
% Create the error counter.
Test(TestIndex).AxonError = 0;
Test(TestIndex).AxonErrorLine = zeros (n,1);

%% Reading data from the serial port and assigning it to Datastore
Datastore = zeros(n, MeasuredQuantities); % Create a matrix of zeros to which data will be recorded
MatlabInterval = zeros (n,1); %Create a vector of zeros to record the tic/toc intervals
for ii = 1:n
   tic; %Begin timer
   Line = fgets(Axon); %Read a line of string data from the Axon
   Data = str2num(Line); % Convert the string data to numbers
   if (~isempty(Data)) && (length(Data)== MeasuredQuantities)
      Datastore(ii,:) = Data; % Assign the line read to the datastore matrix
   else
      Test(TestIndex).AxonError = (Test(TestIndex).AxonError) + 1;
      Test(TestIndex).AxonErrorLine(ii) = 1;
   end
   Test(TestIndex).MatlabInterval(ii) = toc; %End timer and store. This counts the amount of time needed
for Matlab to make one reading
   if mod(ii,MeasurementFrequency) == 0 % Displays every "MeasurementFrequeny"ith lines for user
feedback purposes
      disp(Line);
   end
end

%% Closing the connection
fclose(Axon);

%% Adding up the intervals to create time vectors
Test(TestIndex).AxonInterval = Datastore(:, MeasuredQuantities) / 1000000; %Store the amount of time
needed for the Axon to return one full line of data and convert from microseconds to seconds
Test(TestIndex).AxonTime = zeros (n,1); %Create a vector of zeros to add the Axon time intervals. This
is the time vector to which all other measurements will be compared with
Test(TestIndex).AxonTime(1) = Test(TestIndex).AxonInterval(1); %Setting the initial Axon time

Test(TestIndex).MatlabTime = zeros (n,1); %Create a vector of zeros to add the Matlab time intervals.
This is for comparison purposes only!
Test(TestIndex).MatlabTime(1) = Test(TestIndex).MatlabInterval(1); %Setting the initial Matlab time

for oo = 2:n
   Test(TestIndex).AxonTime(oo) = Test(TestIndex).AxonInterval(oo) + Test(TestIndex).AxonTime(oo-
1); %Adding up the intervals to create the Axon time vector
```

```
   Test(TestIndex).MatlabTime(oo) = Test(TestIndex).MatlabInterval(oo) +
Test(TestIndex).MatlabTime(oo-1); %Adding up the intervals to create the Matlab time vector
end

%% Correcting for Errors
for ss = 1:n
   if (Test(TestIndex).AxonErrorLine(ss) == 1)
      cb = ss-1; % Closest good data before.
      ca = ss+1; % Closest good data after.
      % Find closest good data before.
      while cb > 0
         if (Test(TestIndex).AxonErrorLine(cb) == 0)
            break;
         else
            cb = cb-1;
         end
      end
      % Find closest good data after.
      while ca < n+1
         if (Test(TestIndex).AxonErrorLine(ca) == 0)
            break;
         else
            ca = ca+1;
         end
      end
      % Account for errors on edges.
      if ((cb == 0) && (ca < n+1))
         cb = ca;
      elseif ((ca == n+1) && (cb > 0))
         ca = cb;
      elseif ((cb == 0) && (ca == n+1))
         disp('Warning! Every line is an error line!');
      end
      % Average the closest good data.
      for tt = 1 : MeasuredQuantities
         Datastore(ss,tt) = (Datastore(cb,tt) + Datastore(ca,tt))/2;
      end
   end
end

%% Re-assigning Datastore to a structure
Test(TestIndex).GPSTime = Datastore(:,1);
Test(TestIndex).Latitude = Datastore(:,2);
Test(TestIndex).Longitude = Datastore(:,3);
Test(TestIndex).GPSSpeed = Datastore(:,4);
Test(TestIndex).GPSHeading = Datastore(:,5);
Test(TestIndex).CompassHeading = Datastore(:,6);
Test(TestIndex).Ax = Datastore(:,7);
Test(TestIndex).Ay = Datastore(:,8);
```

```
Test(TestIndex).Az = Datastore(:,9);
Test(TestIndex).Gx = Datastore(:,10);
Test(TestIndex).Gy = Datastore(:,11);
Test(TestIndex).Gz = Datastore(:,12);
Test(TestIndex).WindHeading = Datastore(:,13);
Test(TestIndex).WindSpeed = Datastore(:,14);
```

B.2. Standalone Interface

B.2.1. AxonII Program

Presented here is the user generated code portion of the program running on the AxonII for the standalone interface mode. The function hardware.h is defined by Webbotlib depending on the hardware attached to the AxonII board.

```
#include <string.h>
#include <stdio.h>
#include "hardware.h"

//Declaring and initialising the timer variables
TICK_COUNT StartTimer;
TICK_COUNT EndTimer;
uint8_t writing = 0;
uint8_t error = 0;
FATFILE datafile;

// Initialise the hardware
void appInitHardware(void)
{
	initHardware();
	if(!disk_sdCard.initialised)
	{
		// ERR1: Error initalizing the card.
		// Causes:	1) No card present.
		//			2) Card not formatted.
		error = 1;
		segled_put_char(&led_display,'1');
	}
}

// Initialise the software
TICK_COUNT appInitSoftware(TICK_COUNT loopStart)
{
	if(!error)
	{
```

```
		segled_put_char(&led_display,'-');
	}
	return 0;
}
// This is the main loop
TICK_COUNT appControl(LOOP_COUNT loopCount, TICK_COUNT loopStart)
{
	if(!writing  && SWITCH_pressed(&button) && !error)
	{
		uint16_t  fileNumber;
		char fileNumberStr[6];
		char fileName[11];
		for (fileNumber = 1; fileNumber <= UINT16_MAX; fileNumber++)
		{
			itoa(fileNumber,fileNumberStr,10);
			fileName[0]='\0';
			strcat(fileName,"/");
			strcat(fileName,fileNumberStr);
			strcat(fileName,".txt");
			if(!fileExists(&disk_sdCard,fileName))
			{
				break;
			}
		}
		if(fileOpen(&disk_sdCard, &datafile, fileName,'w'))
		{
			// ERR2: Disk is full.
			// Causes: Disk is full.
			error = 2;
			segled_put_char(&led_display,'2');
		}
		else
		{
			writing = 1;
			segled_put_char(&led_display,'0');
			delay_ms(500);
			while(SWITCH_pressed(&button));
			delay_ms(500);
		}
	}
	else if(writing  && SWITCH_pressed(&button) && !error)
	{
		fileClose(&datafile);
		writing = 0;
		segled_put_char(&led_display,'-');
		delay_ms(500);
		while(SWITCH_pressed(&button));
		delay_ms(500);
	}
```

```
else if(writing  && !error)
{
        char buffer[128];
        // -------- Start GPS -------
        gpsNMEAprocess(&gps);
        double  gpsTime  = 0;
        double  gpsLat = 0;
        double  gpsLong  = 0;
        double  gpsSpeed = 0;
        double  gpsTrack = 0;
        if(gps.gps.info.valid)
        {
                gpsTime  = gps.gps.info.fixTime;
                gpsLat = gps.gps.info.latitude;
                gpsLong  = gps.gps.info.longitude;
                gpsSpeed = gps.gps.info.speed;
                gpsTrack =  gps.gps.info.track;
        }
        // -------- End   GPS -------

        // -------- Start HMC6352 Compass-------
        // Read the HMC6352 Compass and store results
        compassRead(compass);
        // The bearing is stored in compass.compass.bearingDegrees;
        // -------- End   HMC6352 Compass-------

        // -------- Start ADXL335 3 Axis  Accelerometer-------
        uint16_t  Ax = a2dConvert10bit(ADC_CH_ADC1);
        uint16_t  Ay = a2dConvert10bit(ADC_CH_ADC2);
        uint16_t  Az = a2dConvert10bit(ADC_CH_ADC3);
        // -------- End   ADXL335 3 Axis Accelerometer-------

        // -------- Start LPR530AL 2 axis-------
        uint16_t  Gx = a2dConvert10bit(ADC_CH_ADC4);
        uint16_t  Gy = a2dConvert10bit(ADC_CH_ADC5);
        // -------- End   LPR530AL 2 axis-------

        // -------- Start LY530AL 1 axis-------
        uint16_t  Gz = a2dConvert10bit(ADC_CH_ADC6);
        // -------- End   LY530AL 1 axis-------

        // -------- Start ADC 14 Wind Vane-------
        uint16_t  WindDirection  = a2dConvert10bit(ADC_CH_ADC14);
        // -------- End   ADC 14 Wind Vane-------

        // -------- Start ADC 15 Anemometer -------
        uint16_t  WindSpeed = a2dConvert10bit(ADC_CH_ADC15);
        // -------- End   ADC 15 Anemometer -------
```

```
            //Get timer final value in microseconds
            TICK_COUNT timeInterval = 0;
            while(timeInterval < 10000)
            {
                  EndTimer = clockGetus();
                  timeInterval = EndTimer - StartTimer;
            }


      sprintf(buffer,"%1.3f,%1.4f,%1.4f,%1.2f,%1.2f,%u,%u,%u,%u,%u,%u,%u,%u,%u,%lu\r\n",
                  gpsTime,
                  gpsLat,
                  gpsLong,
                  gpsSpeed,
                  gpsTrack,
                  compass.compass.bearingDegrees,
                  Ax,
                  Ay,
                  Az,
                  Gx,
                  Gy,
                  Gz,
                  WindDirection,
                  WindSpeed,
                  timeInterval);

            if(fileWriteNext(&datafile,strlen(buffer),buffer)  != strlen(buffer))
            {
                  // ERR2: Disk is full.
                  // Causes: Disk is full.
                  error = 2;
                  segled_put_char(&led_display,'2');
                  fileClose(&datafile);
                  writing = 0;
            }

            //Get initial  timer value in microseconds
            StartTimer = EndTimer;
      }

      return 0;
}
```

B.3. Post Processing Matlab Scripts

B.3.1. SD Card Raw Data Importation

This script is used to import raw data from the standalone (SD card) in the form of text files. First, import the files into Matlab by following the File/Import Data… menu options. Import the appropriate file into the Matlab workspace. The imported filename must correspond to the filename defined in the script. This script automatically converts the text data into the same structure format used in the PC interface mode.

```
% Raw data importation into Test file
% to be used with SDcard mode, imports and converts the data into a
% similar format to the USB interface

%%Define imported data filename
filename = XXXXXX; %Change filename here
%%
%Index check

TestIndexCheck = exist('TestIndex','var'); % Check if TestIndex exists.
if TestIndexCheck == 0 % Zero if it does not.
   TestIndex = 0; % Create it and set to zero.
end
TestIndex = TestIndex + 1; % Increment to next index.

n = numel(filename)/15;%counting the number of elements in a column in imported file for use later


%% Adding up the intervals to create time vectors
Test(TestIndex).AxonInterval = filename(:,15) / 1000000; %Store the amount of time needed for the
Axon to return one full line of data and convert from microseconds to seconds
Test(TestIndex).AxonTime = zeros (n,1); %Create a vector of zeros to add the Axon time intervals. This
is the time vector to which all other measurements will be compared with
Test(TestIndex).AxonTime(1) = Test(TestIndex).AxonInterval(1); %Setting the initial Axon time


%% Re-assigning imported data to a structure
Test(TestIndex).GPSTime = filename(:,1);
Test(TestIndex).Latitude = filename(:,2);
Test(TestIndex).Longitude = filename(:,3);
Test(TestIndex).GPSSpeed = filename(:,4);
Test(TestIndex).GPSHeading = filename(:,5);
Test(TestIndex).CompassHeading = filename(:,6);
Test(TestIndex).Ax = filename(:,7);
Test(TestIndex).Ay = filename(:,8);
```

```
Test(TestIndex).Az = filename(:,9);
Test(TestIndex).Gx = filename(:,10);
Test(TestIndex).Gy = filename(:,11);
Test(TestIndex).Gz = filename(:,12);
Test(TestIndex).WindHeading = filename(:,13);
Test(TestIndex).WindSpeed = filename(:,14);

%%cleanup
clearvars n
```

B.3.2. SD Card Conversion

This Matlab script is to be used after importing the raw data from the standalone (SD card) in order to convert it to the appropriate engineering units so the output format is the same between the standalone and PC interface modes. It also adapts to calibration constants and offsets determined by static testing.

```
%% Constants from calibration.
AxZero = 3.362646354166667e+02;  %steps
AyZero = 3.379853645833333e+02;  %steps
AzZero = 3.383023437500000e+02;  %steps

AxSens = 0.142926193489032;  %(m/s^2)/step
AySens = 0.140771941262186;  %(m/s^2)/step
AzSens = 0.145519161791881;  %(m/s^2)/step

GxZero = 2.444007083333333e+02;  %steps
GyZero = 2.525703750000000e+02;  %steps
GzZero = 2.508863541666667e+02;  %steps

GxSens = 1.474572791281662;  %(deg/s)/step
GySens = 1.578517811907089;  %(deg/s)/step
GzSens = 1.558676011990455;  %(deg/s)/step

AxStaticAlignment  = 1.624; %(m/s^2)
AyStaticAlignment  = 1.127; %(m/s^2)
AzStaticAlignment  = 1.924; %(m/s^2)
GxStaticAlignment  = 14.726; %(deg/s)
GyStaticAlignment  = 15.435; %(deg/s)
GzStaticAlignment  = 19.661; %(deg/s)


%% Time Vector Construction
```

```
n = numel(Test);
for ii = 1:n
   nn = numel(Test(ii).AxonInterval);
   Test(ii).AxonTime = zeros (nn,1); %Create a vector of zeros to add the Axon time intervals. This is the
time vector to which all other measurements will be compared with
   Test(ii).AxonTime(1) = Test(ii).AxonInterval(1); %Setting the initial Axon time
   for jj = 2:nn
      Test(ii).AxonTime(jj) = Test(ii).AxonInterval(jj) + Test(ii).AxonTime(jj-1); %Adding up the
intervals to create the Axon time vector
   end
end

%% IMU Conversion

n = numel(Test);
for ii = 1:n
   Test(ii).Ax = ((Test(ii).Ax-AxZero)*AxSens ) - AxStaticAlignment;
   Test(ii).Ay = ((Test(ii).Ay-AyZero)*AySens ) - AyStaticAlignment;
   Test(ii).Az = ((Test(ii).Az-AzZero)*AzSens ) - AzStaticAlignment;
   Test(ii).Gx = ((Test(ii).Gx-GxZero)*GxSens ) - GxStaticAlignment;
   Test(ii).Gy = ((Test(ii).Gy-GyZero)*GySens ) - GyStaticAlignment;
   Test(ii).Gz = ((Test(ii).Gz-GzZero)*GzSens ) - GzStaticAlignment;
end

%% Wind Speed Conversion

n = numel(Test);
for ii = 1:n
   nn = numel(Test(ii).WindSpeed);
   ClicksIndex = 1;

   for gg = 2:nn
      if ((Test(ii).WindSpeed(gg) > 500) && (Test(ii).WindSpeed(gg-1) <= 500))
         Clicks(ClicksIndex) = gg;
         ClicksIndex = ClicksIndex + 1;
      end
   end

   if (exist('Clicks','var') == 1)
      if numel(Clicks) > 1
         for dd = 1:(numel(Clicks)-1)
            IntervalWindspeed = 1.2 / (Test(ii).AxonTime(Clicks(dd+1)) - Test(ii).AxonTime(Clicks(dd)));
            for rr = Clicks(dd):Clicks(dd+1)
               Test(ii).WindSpeed(rr) = IntervalWindspeed;
            end
         end
         for rr = 1:Clicks(1)-1
            Test(ii).WindSpeed(rr) = Test(ii).WindSpeed(Clicks(1));
         end
```

```
        for rr = Clicks(end)+1:nn
            Test(ii).WindSpeed(rr) = Test(ii).WindSpeed(Clicks(end));
        end
    end
    clearvars Clicks;
  end
end

%% Wind Heading Conversion

n = numel(Test);
for ii = 1:n
  nn = numel(Test(ii).WindHeading);
  for yy = 1:nn
    if (Test(ii).WindHeading(yy) <= 73 )
      Test(ii).WindHeading(yy) = 112.5;
    elseif (Test(ii).WindHeading(yy) > 73 && Test(ii).WindHeading(yy) <= 87 )
      Test(ii).WindHeading(yy) = 67.5;
    elseif (Test(ii).WindHeading(yy) > 87 && Test(ii).WindHeading(yy) <= 108 )
      Test(ii).WindHeading(yy) = 90;
    elseif (Test(ii).WindHeading(yy) > 108 && Test(ii).WindHeading(yy) <= 154 )
      Test(ii).WindHeading(yy) = 157.5;
    elseif (Test(ii).WindHeading(yy) > 154 && Test(ii).WindHeading(yy) <= 213 )
      Test(ii).WindHeading(yy) = 135;
    elseif (Test(ii).WindHeading(yy) > 213 && Test(ii).WindHeading(yy) <= 266 )
      Test(ii).WindHeading(yy) = 202.5;
    elseif (Test(ii).WindHeading(yy) > 266 && Test(ii).WindHeading(yy) <= 346 )
      Test(ii).WindHeading(yy) = 180;
    elseif (Test(ii).WindHeading(yy) > 346 && Test(ii).WindHeading(yy) <= 432 )
      Test(ii).WindHeading(yy) = 22.5;
    elseif (Test(ii).WindHeading(yy) > 432 && Test(ii).WindHeading(yy) <= 530 )
      Test(ii).WindHeading(yy) = 45;
    elseif (Test(ii).WindHeading(yy) > 530 && Test(ii).WindHeading(yy) <= 615 )
      Test(ii).WindHeading(yy) = 247.5;
    elseif (Test(ii).WindHeading(yy) > 615 && Test(ii).WindHeading(yy) <= 667 )
      Test(ii).WindHeading(yy) = 225;
    elseif (Test(ii).WindHeading(yy) > 667 && Test(ii).WindHeading(yy) <= 744 )
      Test(ii).WindHeading(yy) = 337.5;
    elseif (Test(ii).WindHeading(yy) > 744 && Test(ii).WindHeading(yy) <= 806 )
      Test(ii).WindHeading(yy) = 0;
    elseif (Test(ii).WindHeading(yy) > 806 && Test(ii).WindHeading(yy) <= 856 )
      Test(ii).WindHeading(yy) = 292.5;
    elseif (Test(ii).WindHeading(yy) > 856 && Test(ii).WindHeading(yy) <= 914 )
      Test(ii).WindHeading(yy) = 315;
    elseif (Test(ii).WindHeading(yy) > 914 )
      Test(ii).WindHeading(yy) = 270;
    end;
  end
end
```

```
%% Compass Heading Conversion

n = numel(Test);
for ii = 1:n
   nn = numel(Test(ii).CompassHeading);
   for jj = 1:nn
      % Compass north is 90 degrees offset from box positive X axis.
      Test(ii).CompassHeading(jj) = Test(ii).CompassHeading(jj) - 90;
      if Test(ii).CompassHeading(jj) < 0
         Test(ii).CompassHeading(jj) = 360 + Test(ii).CompassHeading(jj);
      end
   end
end


%% Clean Up

clearvars AxZero AyZero AzZero AxSens AySens AzSens GxZero GyZero GzZero GxSens GySens
GzSens n ii nn ClicksIndex Clicks IntervalWindspeed gg jj yy ZeroThreshold AxStaticAlignement
AyStaticAlignement AzStaticAlignement GxStaticAlignment GyStaticAlignment GzStaticAlignment;
```

B.3.3. Wind Speed Conversion

This script converts the raw wind speed data from single ADC readings to averaged wind speeds between sensor output pulses as described in section 0.

```
%% Wind Speed Conversion

n = numel(Test(1,1));
for ii = 1:n
   nn = numel(Test(ii).WindSpeed);
   ClicksIndex = 1;

   for gg = 2:nn
      if ((Test(ii).WindSpeed(gg) > 500) && (Test(ii).WindSpeed(gg-1) <= 500))
         Clicks(ClicksIndex) = gg;
         ClicksIndex = ClicksIndex + 1;
      end
   end

   if (exist('Clicks','var') == 1)
      if numel(Clicks) > 1
         for dd = 1:(numel(Clicks)-1)
            IntervalWindspeed = 1.2 / (Test(ii).AxonTime(Clicks(dd+1)) - Test(ii).AxonTime(Clicks(dd)));
            for rr = Clicks(dd):Clicks(dd+1)
               Test(ii).WindSpeed(rr) = IntervalWindspeed;
```

```
            end
        end
        for rr = 1:Clicks(1)-1
            Test(ii).WindSpeed(rr) = Test(ii).WindSpeed(Clicks(1));
        end
        for rr = Clicks(end)+1:nn
            Test(ii).WindSpeed(rr) = Test(ii).WindSpeed(Clicks(end));
        end
    end
    %clearvars Clicks IntervalWindspeed ClicksIndex dd gg ii n nn rr;
  end
end
```

Appendix C - Hardware and Wiring

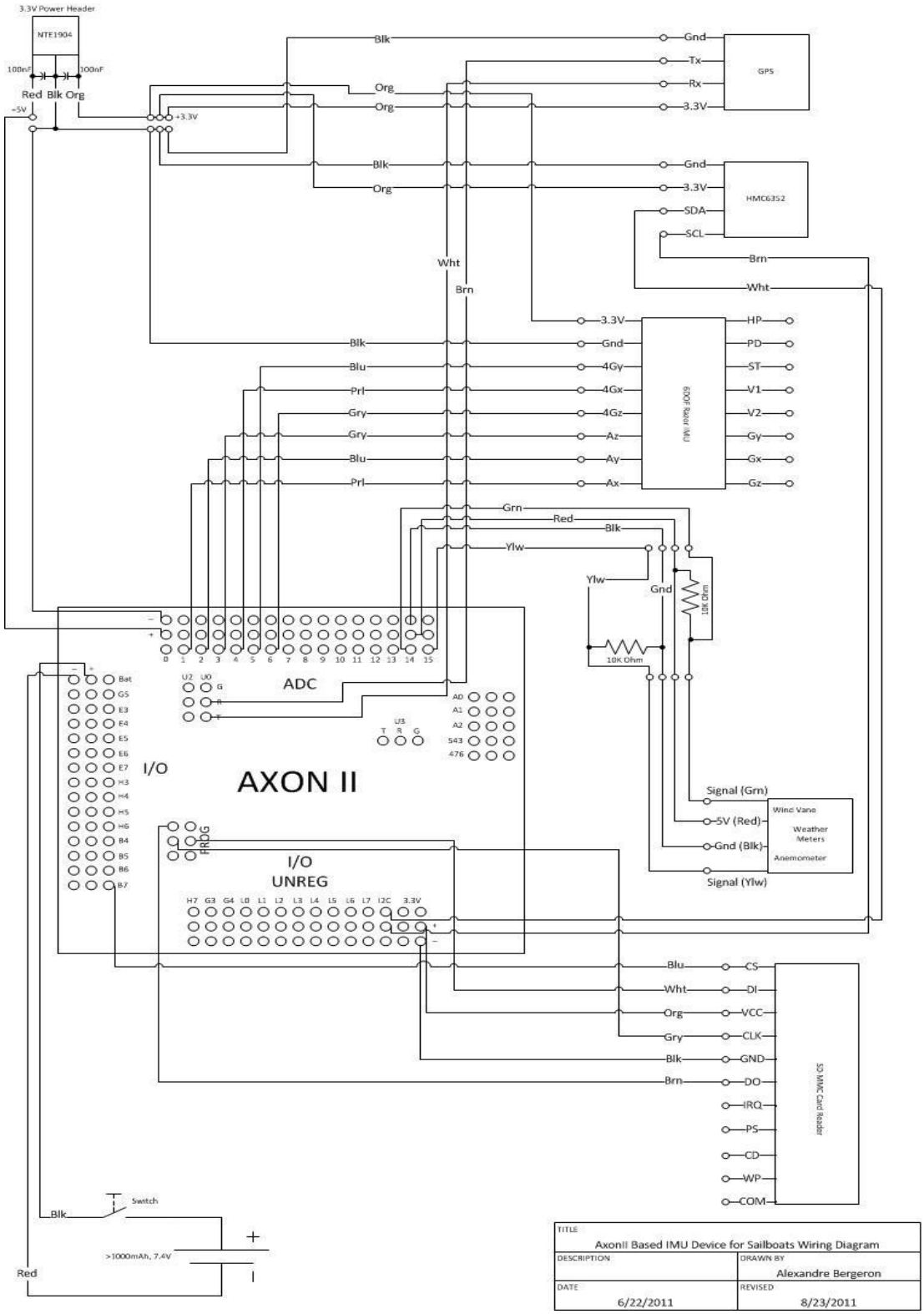
C.1. Component List

The list of required parts and components needed to construct the marine data acquisition prototype for inertial measurement is shown in Table 44.

Table 44: Parts List

Component Name	Retailer	Quantity Required
AxonII Microcontroller	Society of Robots [82]	1
6DOF Razor IMU	Sparkfun Electronics [86]	1
HMC6352 Compass	Sparkfun Electronics [88]	1
Venus 634LPx GPS	Sparkfun Electronics [89]	1
ANT-555 GPS Antenna	Sparkfun Electronics [90]	1
Weather Sensor Assembly	Sparkfun Electronics [91]	1
SD Card Interface	Sparkfun Electronics [93]	1
Battery	Sparkfun Electronics [97]	1
Pelican 1120 Case	Mountain Equipment Co-op [109]	1
WM2512-ND Pin	Digi-Key Corporation [110]	52
WM2800-ND 2 Pin Housing	Digi-Key Corporation [111]	7
WM2801-ND 3 Pin Housing	Digi-Key Corporation [112]	16
WM2802-ND 4 Pin Housing	Digi-Key Corporation [113]	3
Voltage Regulator NTE1904	Newark Corporation [114]	1
Wire, 22AWG, Black	Local Supplier	N/A
Wire, 22AWG, Red	Local Supplier	N/A
Wire, 22AWG, Orange	Local Supplier	N/A
Wire, 22AWG, White	Local Supplier	N/A
Wire, 22AWG, Blue	Local Supplier	N/A
Wire, 22AWG, Purple	Local Supplier	N/A
Wire, 22AWG, Grey	Local Supplier	N/A
Wire, 22AWG, Brown	Local Supplier	N/A
Wire, 22AWG, Yellow	Local Supplier	N/A
Resistance, 10KΩ	Local Supplier	2
Capacitor, 100nF	Local Supplier	2
0.100 Inch Header Pins	Local Supplier	104
Prototyping Board	Local Supplier	1

C.2. Wiring Diagram



C.3. Connectors

Most connectors used in the wiring of the prototype data acquisition system are Molex® 0.100" Grid Stackable Single Row connectors, in two, three and four pin varieties. The pins required are Molex® WM2512-ND, seen in Figure 71, and can be crimped on 22 or 24 gauge wire.

Figure 71: Molex WM2512-ND Pin [110]

These pins are designed to lock into position in their plastic housings. The two pin housing is a Molex®WM2800-ND, the three pin is a Molex® WM2801-ND and the four pin is Molex® WM2802-ND. All are visible in Figure 72.

Figure 72: Molex Connector Plastic Housings [111] [112] [113]

These connectors are typically used in connecting PC hardware, hobby servo-motors and low-cost electronics. A basic tutorial on their use, installation and re-use can be found in (http://www.societyofrobots.com/electronics_wire_connector.shtml) [115].

C.4. Prototype Cost

One of the more important design criteria specified in Chapter 4 was the total cost of the data acquisition prototype. A breakdown of the actual value of each component used is described below.

The individual cost for each component required to assemble the data acquisition prototype is shown in Table 45.

Table 45: Data Acquisition Prototype Component Cost

Component	Retailer	Cost ($US)
AxonII Microcontroller	Society of Robots [82]	108.00
6DOF Razor IMU	Sparkfun Electronics [86]	59.95
HMC6352 Compass	Sparkfun Electronics [88]	34.95
Venus 634LPx GPS	Sparkfun Electronics [89]	49.95
ANT-555 GPS Antenna	Sparkfun Electronics [90]	12.95
Weather Sensor Assembly	Sparkfun Electronics [91]	9.95
SD Card Interface	Sparkfun Electronics [93]	9.95
Battery	Sparkfun Electronics [97]	6.95
Connectors & Wiring	Digikey.ca [110] [111] [112] [113]	20.00
Pelican 1120 Case	Mountain Equipment Co-op [109]	37.04
Total		**409.69**

The total component cost is 409.69 $US, not including local taxes and shipping. In practise, it is safe to say that even with the inclusion of shipping and taxes, it is possible to construct this data acquisition device for under US$500.00. The cost of soldering equipment, tools and other minor consumables was not included in this total.

Appendix D - AxonII Related Software & Drivers

D.1. Virtual COM port drivers

The AxonII uses a Silicon Labs CP210x USB to UART bridge integrated circuit to convert the microcontroller's low-level communications to the PC-friendly USB format. On the PC end, a driver is needed to create a virtual serial (COM) port that in reality is a USB device. The necessary driver is found at [116].

(https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx)

D.2. AVR Studio

Atmel AVR Studio® is the integrated development environment used to create programs for Atmel microprocessors. The version used for the design and development of the marine data acquisition prototype is AVR Studio 4.19. This is available as a free download from Atmel (http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725) [117]. Instructions pertaining to the download, installation and programming for the AxonII can be found here (http://www.societyofrobots.com/axon2/axon2_setup1.shtml) [118].

D.3. Webbotlib

The library used in conjunction with AVR Studio is Webbotlib, which contains templates and functions specific to the AxonII and the sensors. Instructions on the download and installation of Webbotlib can be found at (http://www.societyofrobots.com/axon2/axon2_setup1.shtml) [118] and in (http://www.webbotLib.org.uk) [52]. Complete documentation on Webbotlib is found at [53].

Another feature used is the Webbotlib Project Designer; the main interface is shown in Figure 73. Project Designer allows the user to visually "connect" all of the peripheral devices onto a large number of Atmel AVR based microcontrollers or development boards [119]. Once all of the required devices and batteries are connected, the project designer exports an AVR Studio

project file complete with all of the proper definitions and pin identifiers used. This simplifies prototyping and programming for the end-user.

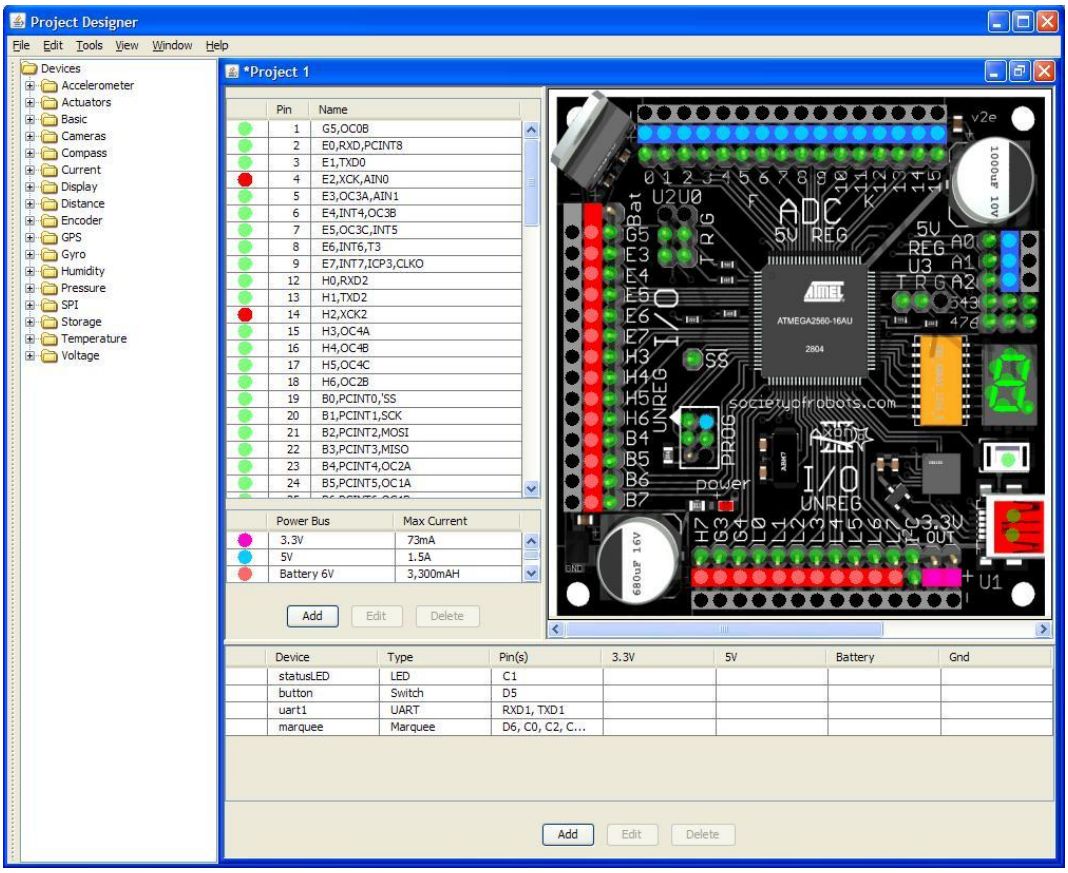


Figure 73: Webbotlib Project Designer [119]

D.4. AxonII Bootloader & Procedures

One key feature of the AxonII is the ability to be programmed through its USB port rather than requiring a specialised hardware programmer. USB programming is done by using a bootloader utility as shown in Figure 74.

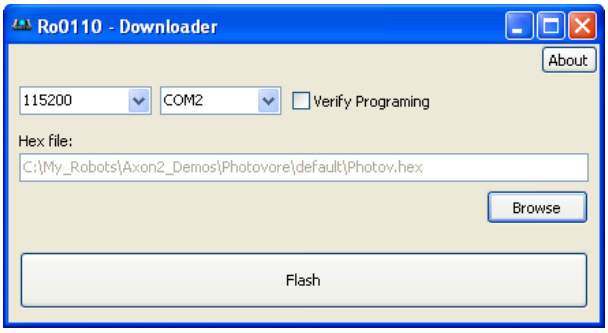


Figure 74: AxonII Bootloader [118]

Specific instructions on how to download and use the AxonII bootloader are given in (http://www.societyofrobots.com/axon2/axon2_setup1.shtml) [118]. Care must be taken to make sure that the AxonII is unpowered while the bootloader is initialised and that power is not interrupted during the programming sequence.

Appendix E - Sailing

E.1. Sailboat Anatomy

To facilitate the reader that may be unfamiliar with sailboat layout, parts and nautical terminology, Figure 75 illustrates the principal relevant components.

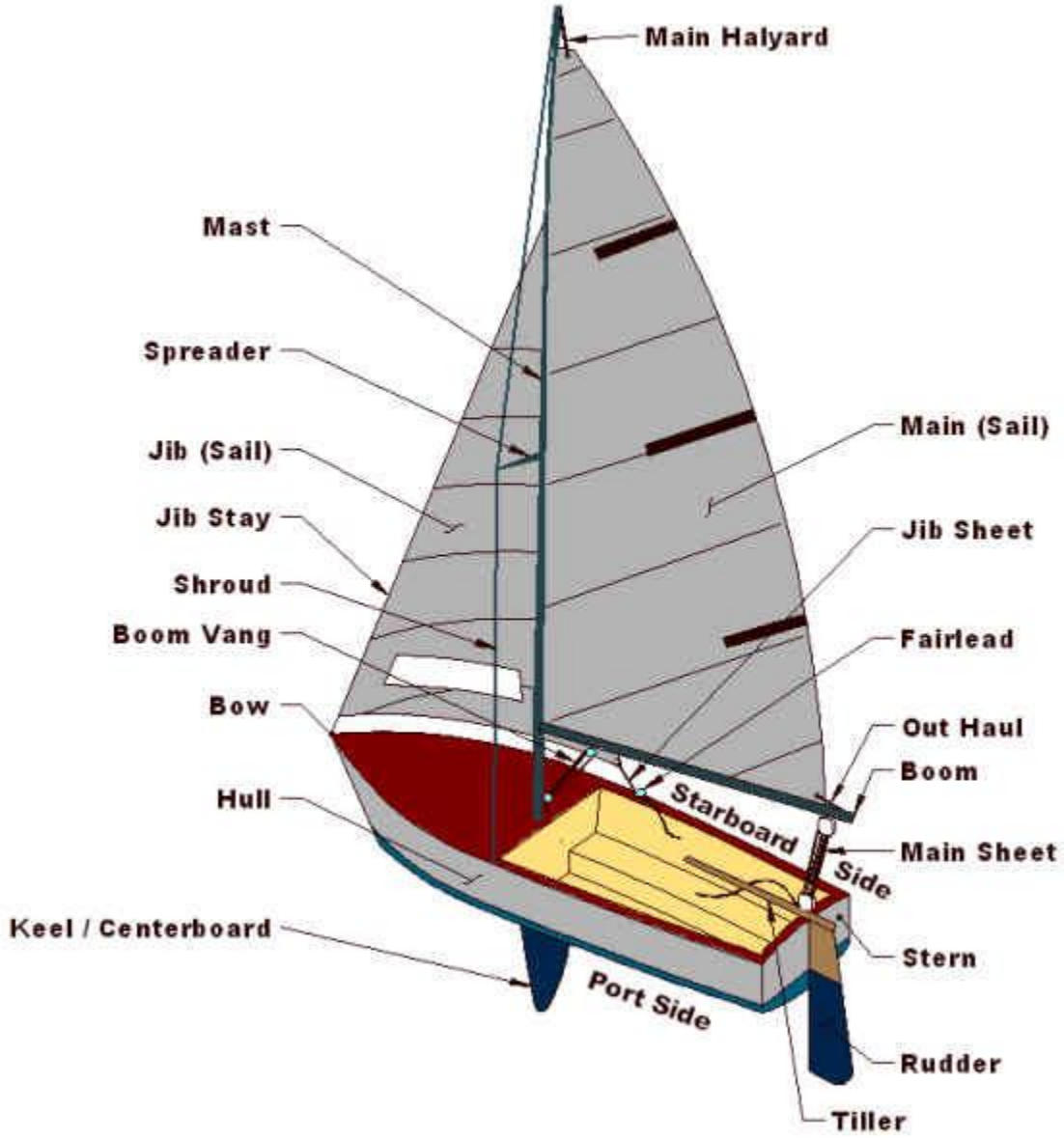


Figure 75: Anatomy of a Sailboat [120]

E.2. Nautical Terminology

- Block: nautical term for pulley
- Bow: front of boat
- Chain plate: through hull mounting for a shroud
- Gybing: manoeuvre where the sailboat crosses the wind stern first
- Halyard: cable or rope with a function relating to hoisting or pulling upwards or on a sail
- Heavy air: high winds
- Heeling: lateral leaning or the hull, or rotation about the bow to stern axis
- Light air: low winds
- Reef: Attachment point designed to reduce the overall area of the sail. Used during heavy air situations lower the sail.
- Shroud: Cables designed to stiffen the mast
- Spreaders: Devices which spread the shrouds. Usually straight bars.
- Stern: rearmost portion of the boat
- Tacking: manoeuvre where the sailboat crosses the wind bow first