
Design, Build and Test of a Smart-Wing for UAV

Authors:

Anthony Hill, Fang Hwa Ting,

Roshan Pasupathy, Chris Bunn,

Fintan Healy, Nikolay Tarasov, Louis Winter

Supervisors:

Dr. T Glyn Thomas

Dr. Stephen D Prior

April 2017

This report is submitted in partial fulfilment of the requirements for the Degree of Master of Engineering, Faculty of Engineering and the Environment,

The University of Southampton

Declaration

We the undersigned confirm that the material presented in this project report is all our own work. References to, quotations from, and the discussion of work of any other person have been correctly acknowledged/cited within the report in accordance with University of Southampton guidelines on academic integrity.

Name: Anthony HILL

Signature: _____

Name: Fang Hwa TING

Signature: _____

Name: Roshan PASUPATHY

Signature: _____

Name: Chris BUNN

Signature: _____

Name: Fintan HEALY

Signature: _____ 

Name: Nikolay TARASOV

Signature: _____

Name: Louis WINTER

Signature: _____

Acknowledgements

Our greatest appreciation goes to the following for their effort, support and guidance during this project:

Dr T Glyn Thomas for his invaluable guidance and overseeing of the project.

Dr Stephen D Prior for his supervision and insight during project meetings.

Mr Mehmet Ali Erbil for his much-needed help and guidance during UAV building and wind tunnel testing.

Paul Heckles (Pilot) for his excellent piloting skills which enabled our UAV to collect and gather data whilst in flight.

Staff of the Engineering Design and Manufacturing Centre for their hard work and dedicated service.

Abstract

This report details the work undertaken as part of a final year engineering, Group Design Project in which a pressure sensor system that could fit inside an Unmanned Aerial Vehicle (UAV) wing was designed, built and tested. The purpose of this is to measure the pressure distribution across a wing and can predict the lift acting on said wing. With this information, it is feasible to implement a control system on the flaps of the wing to respond to a gust. This project was split into three areas: creating a pressure sensing system capable of real time data acquisition; predicting lift from pressure data across the profile of the wing; and testing the system in a wind tunnel with test rig, and UAV flight situation. The sensor board was built to be compact, housed in a parametric box with individual sensors separated by a gasket, able to fit inside a relatively thin wing and provide a sampling rate of 100Hz. The test rig provided a means of accurately predicting the aerodynamic forces that will act on the wings during flight. The test rig was also a means of testing and calibrating the pressure measurement system before applying it on the actual UAV. Simulations were run through XFOIL across a two-dimensional (2D) aerofoil to optimise pressure sensor locations along the wing for minimum error in readings. Further simulations were run to investigate multiple methods of approximating lift from the pressure distribution. In the future, the theoretical 2D model can be expanded into a three-dimensional (3D) finite model to provide more accurate results. This project proved the feasibility of implementing a pressure measurement system along a wing in real time flight. Future work will be focussed on adding a control system to respond to changes in gust and hence dynamically controlling lift on a wing.

SUMMARY REPORT

Design, Build and Test of a Smart-Wing for UAV

Authors:

Anthony Hill, Fang Hwa Ting,

Roshan Pasupathy, Chris Bunn,

Fintan Healy, Nikolay Tarasov, Louis Winter

Supervisors:

Dr. T Glyn Thomas

Dr. Stephen D Prior

April 2017

This report is submitted in partial fulfilment of the requirements for the Degree of Master of Engineering, Faculty of Engineering and the Environment,

The University of Southampton

1. Introduction

Typically, accelerometers are used to measure changes in lift in modern aircraft. However, lift and changes in lift can be represented by a pressure distribution across the upper and lower surfaces of the aircraft. The ability to measure and control this pressure distribution can lead to increased accuracy in control and measurements systems.

This project looks to design a system which could measure the pressure distribution across a wing in flight and perform analysis to approximate the lift in real time. Although there are many applications and uses for such a system, the focus for this project is on moving control surfaces to maintain constant lift.

a. Aims & Objectives

i. Aims

This project investigates a method to test for the feasibility and accuracy of measuring the pressure across a finite wing using a wind tunnel load measurement device. In addition to this, a set of a pressure sensors across an unmanned air vehicle wing were to be implemented and tested in flight conditions. Furthermore, methods of determining the lift and correcting to maintain constant lift from a pressure distribution are investigated.

ii. Objectives

The objectives specific to this project are:

- Develop a system to measure the pressure distribution across a main wing.
- Choose suitable sensing locations to measure the pressure distribution across the wing.
- Design and build a test rig for wind tunnel testing to validate the pressure distribution data.
- Design and build a UAV to gather live pressure data during flight
- Implement a control system to maintain constant lift across the main wing of a UAV.

2. Resources

a. Starting Funds

Each GDP group is given an initial allowance of £300 to spend on their project, for such things as materials and components. Each student is additionally allowed to spend £80 of their own money for the project, with the university reimbursing any funds utilised. This brought the groups initial budget up to £860.

From the beginning of the project it was unknown what the final total cost would be. This was due to this project being a new concept put forward by the university, thus no estimates could be gained from previous groups. It was predicted that the bulk of funds will be allocated to the UAV itself, for purchasing of the fuselage and landing gear, with a smaller proportion available for the test rig. To further decrease spending, parts were sourced and borrowed from either the university or the group's supervisor Dr. Glyn Thomas. This was done well in advance to ensure their availability and to prevent any delays.

During one of the early group meetings, it was decided that the group will have enough funds to adequately complete the aims of this project. Thus, there was no need to do the elevator pitch to get an extra sponsor.

b. Third-party Sponsor

The group did have the help of a third-party sponsor, who do not wish to be named. The sponsor provided a means for the group to produce the pressure sensor board.

3. Approach

The development of the smart wing was comprised of a multitude of phases; an initial design phase, testing manufacturing, wind tunnel testing, additional wing design, final manufacturing and build followed lastly by UAV implementation testing. The initial design phase encompassed four separate areas which were researched and designed in parallel, this included control system design,

pressure sensing design, test rig design and test wing design. Control system design involved creating a system that could take pressure readings from embedded pressure sensors and construct a wing lift profile from this data. The pressure sensor system design developed a way to have many sensors take and collate this data at a useful speed and transmit this data live via a Wi-Fi set up. As wind tunnel testing was planned, a test rig was needed. This involved designing a test rig that would be able to fit in the plume of a wind tunnel. This test rig could separate lift and drag forces to help validate the control system previously mentioned. A test wing was also needed for this so one was designed that would fit into the test rig and wind tunnel plume and would be able to house the pressure sensors and be scaled up to a full-size wing for use on a UAV. In the final manufacturing and build phase, the UAV platform that had been purchased needed to be constructed and the flight control system for this designed and put together. Full wings for the UAV were also constructed and ensured that they could be securely fitted to the platform.

4. *Constraints*

Manufacturing and design constraints were two of the major limitations that had to be considered throughout the project. While the budget also limited some of the planned ideas that were to be implemented, it did not limit the group from finishing the bulk of planned tasks. Problems arose from long waiting times for parts to be manufactured or delivered. Further limitations arose from the assumptions used in 2D modelling, not being fully applicable in a 3D environment.

Limited access to testing space was also a problem. Long queues to use wind tunnel facilities in Boldrewood, limited the time available for calibrations of the pressure sensor system. Due to the size of the UAV, an airfield was required for flight testing. Flight testing also required an experienced pilot who was only available for a limited amount of flight days. This naturally limited the testing schedule and capped progress at certain stages of the project.

5. Project Management

a. Group Management

Successful projects are characterised by the utilisation of efficient management. Within this section, the roles of individual group members are discussed. The tasks of the project were sorted to relevant sub-categories and each member of the group was assigned tasks to be completed.

i. Group Members



Anthony HILL



Chris BUNN



Roshan PASUPATHY



Fintan HEALY



Fang Hwa TING



Nikolay TARASOV



Louis WINTER

ii. Group Roles

In the first meeting the group members introduced each other and gave a brief summation of their skill-set, providing an idea as to the best role that they could be assigned. While the roles that are listed are assigned to a given group member, they were not limiting. Typically, multiple group members would work together to finish a given task. The different roles of the group members are listed in Table 1 and Table 2.

Table 1 - Administrative Roles

Administrative roles	
Project Leader	Fintan Healy
Treasurer	Nikolay Tarasov
Final Report Editor	Anthony Hill

Table 2 - General Group Roles

Control System Design and Electronics	
Test rig calibration	Roshan Pasupathy
UAV flight controller design	Anthony Aldo Hill
Simulation and Aerodynamics	
Wing simulation	Christopher Bunn
Pressure sensor design	Fintan Healy
Parts Design and Manufacturing	
Test rig design and manufacture	Fang Hwa Ting
Test wing design and manufacture	Louis Winter
UAV manufacture	Nikolay Tarasov

6. Important Results

Although the test rig and wind tunnel results were incomplete, the live model proved a success. The pressure measurement system received reliable data over Wi-Fi which lines up with data from the Pixhawk that shows altitude and roll characteristics of the UAV. Figure 1 shows the data below. This proves that the concept to install a pressure measurement system inside the profile of the wing was successful.

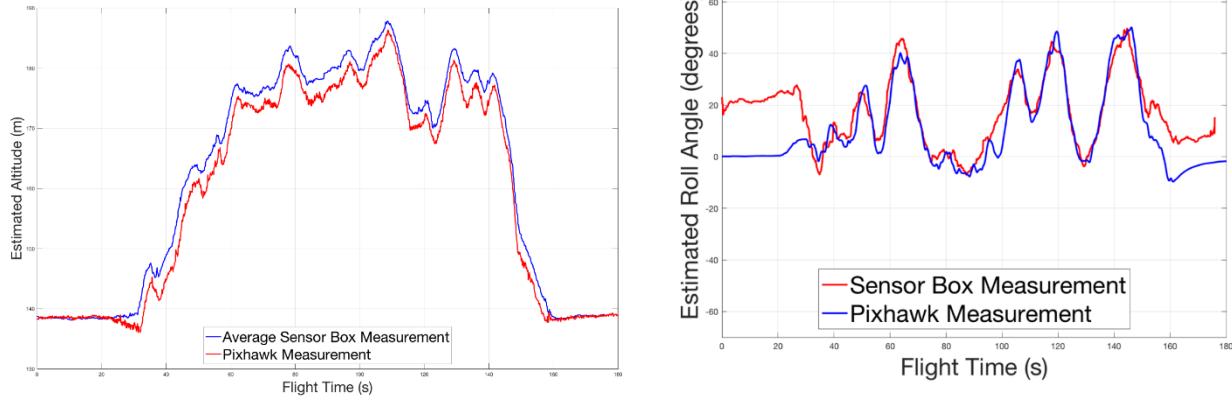


Figure 1 – (left) The estimated altitude of the aircraft using the static pressure measured using the Sensor boards, compared to that measured by the Pixhawk, during the first flight test. (right) The estimated roll angle of the UAV during the first flight test using 2 sources; the inertial measurement unit in the Pixhawk flight controller and a trigonometric relation between the altitude of the two sensor boxes.

7. Conclusion and Future Work

The ability to measure the pressure along a wing section and predict lift has been successful. The pressure has been analysed and lift has been predicted accurately. More data could have been taken to prove without doubt the stability and repeatability of the system. At the time of writing, the system is specific to the wing type and dimensions set out by the size of the sensing box. However, future editions can work a solution to make it more of a modular design for use on any system. Furthermore, a control system can be implemented to react to gusts seen by the UAV.

It was shown that the application of multiple BMP280 pressure sensors were a viable method of measuring the pressure in-flight. It was possible to fit many sensors in a small area, giving potential for many future applications. Measurements from these sensors could be recorded in real time and provide an accurate result, giving similar estimations to that of the Pixhawk flight controller in terms of altitude and roll.

MAIN REPORT

Design, Build and Test of a Smart-Wing for UAV

Authors:

Anthony Hill, Fang Hwa Ting,

Roshan Pasupathy, Chris Bunn,

Fintan Healy, Nikolay Tarasov, Louis Winter

Supervisors:

Dr. T Glyn Thomas

Dr. Stephen D Prior

April 2017

This report is submitted in partial fulfilment of the requirements for the Degree of Master of Engineering, Faculty of Engineering and the Environment,

The University of Southampton

Contents

Declaration.....	2
Acknowledgements.....	3
Abstract.....	4
SUMMARY REPORT	6
MAIN REPORT.....	14
1. Introduction.....	28
1.1. Aims & Objectives.....	29
1.2. Project Management	29
2. Pressure Measurement System – Electronic Development	34
2.1. Industrial Motivation and Brief	34
2.2. The Bosch BMP280 sensor.....	37
2.3. Design of the Sensor Board PCB.....	46
2.4. Design of a box to enclose the PCB.....	51
2.5. Micro-controller Hardware and Software design	51
2.6. Final Hardware setup for UAV for the flight tests	57
3. Pressure Distribution Analysis.....	59
3.1. Wing Sizing.....	59
3.2. Methods to approximate Lift from Pressure	64
3.3. Feedback	75
3.4. Pressure Sensor Quantity and Locations.....	78
4. Calculating Lift from Pressure.....	85
4.1. Method 1 – Lookup Table	85
4.2. Method 2 – Integration.....	90
4.3. Revised Weight	94

5.	Test wing design and manufacture.....	95
5.1.	Section methodology	95
5.2.	Spar Analysis	96
5.3.	Flap Actuation Mechanism	96
5.4.	Servo Selection	97
5.5.	Pressure Tapping Section Design.....	97
5.6.	Manufacture	98
6.	Final Wing Development and Manufacture.....	99
6.1.	Spar Sizing	99
6.2.	Connection of Constructed Wings to Fuselage	101
6.3.	Changes to Pressure Tapping Section	103
6.4.	Sectional Layout	104
6.5.	Servo Calculations	105
6.6.	Wing Manufacture	106
7.	Test Rig Development.....	108
7.1.	Main Purpose and Specification	108
7.2.	Load Measurement Device	109
7.3.	Test Rig Design.....	112
7.4.	Flow Perturbation.....	125
8.	Test Rig Calibration	130
8.1.	Single-axis load cell.....	130
8.2.	Signal Processing	131
8.3.	Calibration Theory	134
8.4.	Calibration Procedure	139
8.5.	Calibration Results.....	142
9.	UAV Test Bed Development.....	145

9.1.	Fuselage construction.....	145
9.2.	Avionics purpose and requirements	151
9.3.	Flight controller set-up.....	154
10.	Testing & Results.....	162
10.1.	Wind Tunnel Testing	162
10.2.	Wind Tunnel Results	166
10.3.	UAV Flight Tests.....	168
10.4.	UAV Flight Results	170
11.	Discussion & Conclusion.....	175
11.1.	Discussion	175
11.2.	Conclusion	179
11.3.	Recommendations and Further Work	180
	Bibliography	184
	Appendix A	187
	Appendix B	188
	Appendix C	197
	Appendix D	198
	Appendix E	200
	Appendix F.....	201
	Appendix G.....	204
	Appendix H.....	212

List of Figures

Figure 1 – (left) The estimated altitude of the aircraft using the static pressure measured using the Sensor boards, compared to that measured by the Pixhawk, during the first flight test. (right) The estimated roll angle of the UAV during the first flight test using 2 sources; the inertial measurement unit in the Pixhawk flight controller and a trigonometric relation between the altitude of the two sensor boxes.....	13
Figure 2 - Measurement Specialities ESP64-HD, a 64 port pressure scanner.....	35
Figure 3 - Scanivalves MPS4264, a 64 port pressure scanner	35
Figure 4 - A diagram of a typical I2C bus with two slave devices	38
Figure 5 - A diagram of a typical SPI bus with two slave devices.....	38
Figure 6 - A diagram illustrating how a single SPI transfer would work between a master and a slave device	39
Figure 7 - Simple schematic for the electrical setup of four BMP280 sensors.....	41
Figure 8 - The Adafruit Huzzah Microcontroller.....	41
Figure 9 - A simple flowchart of the process followed in the main method of the initial code....	41
Figure 10 - A simple flowchart of the process followed in the main method of the initial code, included the synchronous measurement of pressure across all sensors	41
Figure 11 - A simple schematic showing the final design for each cell of 8 BMP280 sensors	42
Figure 12 - A diagram illustrating how shift registers can be used as the SS line for multiple devices	43
Figure 13 - Exploded view of an exemplar sensor sealing design philosophy	44
Figure 14 - Section view of the sealed chambers created for each sensor using the exemplar sensor sealing design philosophy	44
Figure 15 – An example of a Molex pico-lock, board to wire, connector	47
Figure 16 – An exemplar view of the proposed layout of a cell of 4 sensors.....	47
Figure 17 – Bill of Materials sent to ALS for the design and manufacture of the Sensor Board .	47
Figure 18 – Final sensor box dimensions with 2.5 mm thick walls.....	47
Figure 19 – Final sensor box dimensions with 4 mm thick walls.....	47
Figure 20 – Top view of the manufactured Sensor PCBs	48
Figure 21 – Isometric view of the manufactured Sensor PCB.....	48

Figure 22 – Final electrical schematic sent to ALS to design and manufacture the Sensor Board	49
Figure 23 – Schematic of the Proposed initial component layout	50
Figure 24 – A render of an assembled Sensor Box	52
Figure 25 – A render of the exploded view of the Sensor Box Design	52
Figure 26 – A simple flowchart of the process followed in the main method on the microcontroller to collect sensor data then distribute it via UDP.	52
Figure 27 – A basic diagram, showing the network setup used on the flight days to collect pressure sensor data	55
Figure 28 – A simple diagram showing how a multiplexer chip can be used to select different part of the MISO bus	57
Figure 29 – A diagram showing the modularity of the sensor box wiring in the final UAV design	57
Figure 30 – Symmetric NACA aerofoil sizing comparisons for minimum required chord and 35% flap. (Top to bottom: Clark-Y, NACA0010, NACA0012)	61
Figure 31 – Clark Y Pressure Coefficient against Angle of Attack at position (0.24, 0.08783) ...	62
Figure 32 – Initial Wing Size Calculations	63
Figure 33 – Flow Chart for Lookup Table Method	66
Figure 34 - Variation of stagnation position with Angle of Attack [12] [13]	68
Figure 35 - Pressure Coefficients at sampling points within 2.5 cm of the leading edge. Higher Sampling point density	69
Figure 36 - Pressure Coefficients at sampling points within 2.5 cm of the leading edge. Lower Sampling Point Density	69
Figure 37 - Fit Obtained by using a three term Gaussian Model	70
Figure 38 - Fit Obtained by using a two-term exponential model	70
Figure 39 - Fit obtained by using a custom exponential model for fitting	71
Figure 40 - Fit obtained by using smoothing-spline fit	71
Figure 41 - Matlab script calcB used to calculate the vector b	75
Figure 42 – Flowchart of Control Loop using Pressure Sensors and Flap Angle as inputs	77
Figure 43 – Pressure distribution across NACA0012 aerofoil for $Re = 271810$ and flap = 20% chord	79

Figure 44 – Comparison of Change in Pressure Coefficient and Average Number of Solutions across a set of angles of attack and flap angles for each coordinate on the aerofoil. Ideal solution lies at top-left.	80
Figure 45 – Change in Angle of Attack with Pressure Coefficient for Various points across NACA0012. Small pressure coefficient spikes can be seen between 0 and 5 degrees for 10%, 50% and 66% chord and between 5 and 10 degrees for 10% chord.	81
Figure 46 – Average Error between approximated case and ideal case against Total Number of Sensors.	83
Figure 47 – Initial Pressure Tap Positions for Optimal Integration across a 2D NACA0012 aerofoil	83
Figure 48 – Final Pressure Tap Positions.....	84
Figure 49 – Weighting of sensor taps across chord.	88
Figure 50 – Average Error against Standard Deviation for the Weighting Factor over 100 iterations.	90
Figure 51 – Pressure Distribution across the flap on a NACA0012 aerofoil. Comparison of approximation of distribution from sensors compared to actual distribution.	91
Figure 52 – Comparison of different sets of sensors for Lift Coefficient against Angle of Attack for a Flap Angle of -12 degrees.....	92
Figure 53 – Elliptical Lift Distribution compared to an equivalent constant Cl across semi-span with Sensor Positions highlighted. Where the blue curve is the elliptical lift distribution.....	93
Figure 54- Test wing assembly	95
Figure 55- Cross-section of wing-part C	95
Figure 56- Cross-section of wing-part A &C	95
Figure 57- Flap Actuation Mechanism	96
Figure 58 - Isometric view of pressure tapping piece first iteration	97
Figure 59 - Isometric view of pressure tapping section second iteration	98
Figure 60 - Isometric view of pressure tapping section third iteration.	98
Figure 61 - View showing re-routing of pressure channels through spar supports.....	98
Figure 62 - Xfoil analysis of traced wing section	102
Figure 63 - Side view of outboard wing connection.....	103
Figure 64 - Side view of middle wing connection.....	103
Figure 65 - Plan view of outboard wing connection.....	103

Figure 66 - Side view of the final pressure tapping section.....	104
Figure 67 - Isometric view of the final pressure taping section.....	104
Figure 68 - Plan view of final wing assembly	105
Figure 69 - Side view of section A.....	105
Figure 70 - Side view of section B.....	105
Figure 71 - Side view of section E & H.....	105
Figure 72 - Aerodynamic Force Measurement System and Strain Gauges Placement [20]	109
Figure 73 - Single-force load cell (Left) and force measurement system by Samardžić et al. (Right) (External Six-Component Strain Gauge Balance for Low Speed Wind Tunnels [21].....	111
Figure 74 - (Left to right) Bending Beam load cell, compression load cell and S-Beam load cell [22].....	113
Figure 75 - First design of uniaxial force transmitting strut	115
Figure 76 - Improved force decoupling strut design with normal hinges.....	116
Figure 77 - Pillow bearing (right) and the force decoupling strut design with the use of pillow bearing (left).....	117
Figure 78 - Setup for the finite element analysis	118
Figure 79 - Total deformation of the strut structure.....	119
Figure 80 - Stress concentration location on the strut.....	119
Figure 81 - Overall test rig design configuration.....	120
Figure 82 - Overall test rig design configuration (view from side)	121
Figure 83 Total deformation for the drag arm system's rigid structure	121
Figure 84 - Angle of attack changing system.....	122
Figure 85 - Test wing mounting configuration	122
Figure 86 - Images of our final test rig	123
Figure 87 - Image of our final test rig with upper section and wing attached	124
Figure 88 - Initial concept of 'gust' initiator system.....	125
Figure 89 - Isometric view of flap system.	126
Figure 90 - Flaps mounted on flange of wind tunnel. 1- Flange, 2- Perspex end of tunnel, 3- Flap holder, 4- 3D printed connector, 5- Actuating rod, 6- 3D printed rod connector, 7- Flap.	127
Figure 91 - Flap system mounted on end of wind tunnel.	128
Figure 92 - Flaps actuated by a member of group.	129
Figure 93 - Load cell used in test rig.	130

Figure 94 - a) Load cell wiring (left) [28] and b) HX711 chip (right).....	130
Figure 95: Schematic of data acquisition system using the INA125P instrumentation amplifier.	131
Figure 96 - Roughness parameter vs diff from equation (17).....	133
Figure 97 - A simplified representation of the application of a media filter [24]	134
Figure 98 - calc_CW.m	137
Figure 99 - Theoretical workflow of the calibration process [26].	138
Figure 100 - Data flow during calibration	139
Figure 101 - Side view of pulley mechanism for drag load application.....	140
Figure 102 - Top view of mechanism for applying lift and drag forces	141
Figure 103 - Before building, the group members constructing the fuselage set out all parts to visualise the building process.	145
Figure 104 - Close up of front cut board after Epoxy has set.....	146
Figure 105 - Landing gear glued into place with wires of servo threaded through hole in undercarriage.....	147
Figure 106 - Underside of horizontal stabilizer with steps 4 and 5 completed.....	147
Figure 107 - Widened access for all wing wires in top of fuselage.	148
Figure 108 - Hole for the webcam before it was installed.....	148
Figure 109 - Motor shaft channel in right half of centre wing.....	149
Figure 110 - The Completed UAV fuselage before all control system and all internal wiring were installed.....	150
Figure 111 - Servo Locations with Flaperons Included on Wing	153
Figure 112 - Servo Locations with Dual Ailerons	153
Figure 113 - Depiction of Pixhawk and Components Wiring Schematic	155
Figure 114 - Pixhawk Servo Output Rail - With Indication of Useful Pins.....	156
Figure 115 - Laser Cut Board to Hold Electronics Inside Fuselage	157
Figure 116 - Layout of Components on Development Board.....	157
Figure 117 - Compass Calibration Screen in Mission Planner	158
Figure 118 - RC Receiver with Indication of the Intended Bind Plug Location	159
Figure 119 - Radio Calibration Screen in Mission Planner	159
Figure 120 - Diagram Showing the Difference Between Mode 1 and Mode 2 on a Radio Transmitter	160

Figure 121 - Data flow during wind tunnel testing.....	165
Figure 122 - Approximate two-dimensional Lift Coefficient equivalent against Angle of Attack from the wind tunnel test data.....	167
Figure 123 - Waypoint Data Including Altitude from Flight 1 Loaded into Google Earth.....	170
Figure 124 - Height Above Ground Versus Time	171
Figure 125 - Waypoint Data from Flight 1 Loaded into Google Earth.....	171
Figure 126 - Ground Speed of our Aircraft. Three Spikes Represent Flying with the Wind.....	172
Figure 127 - Waypoint Data Including Altitude from Flight 2	173
Figure 128 - Waypoint Data from Flight 2 Loaded into Google Earth.....	173
Figure 129 - Ground Speed of our Aircraft. Three Spikes Represent Flying with the Wind.....	174
Figure 130 - Height Above Ground Versus Time	174
Figure 131 - A render showing the raised areas on the lower side of the lid created to increase the potential of the lid to seal each pressure sensor effectively	176
Figure 132 - A plot of the estimated altitude of the aircraft using the static pressure measured using the developed Sensor boards, compared to that measured by the Pixhawk, during the first flight test.....	177
Figure 133 - A plot of the estimated roll angle of the UAV during the first flight test using 2 sources; the inertial measurement unit in the Pixhawk flight controller and a trigonometric relation between the altitude of the two sensor boxes.....	178
Figure 134 - Extra bracket on rigid stand for future test rig	182

List of Tables

Table 1 - Administrative Roles	12
Table 2 - General Group Roles	12
Table 1 - Summarised budget for group 13, October 2016 to March 2017	32
Table 4 - A Comparison of 3 MEM's based absolute pressure sensors.....	36
Table 5 – A comparison of the transmission rate achievable with different numbers of sensors on the same SPI bus, using two different Master Devices.....	55
Table 6 – Initial Weight Estimation	59
Table 7 – Pareto Optimal Solutions for Pressure Coefficient Difference against Average Number of Solutions	80
Table 8 – Comparison of approximation using a limited set of sensors compared to a set of all possible sensors.....	91
Table 9 – Revised Maximum Take Off Weigh.....	94
Table 10 - Deflection as shown by Ansys and by hand calculations	100
Table 11 - Deflections of a one spar system	100
Table 12 - Deflections of a two-spar system.....	100
Table 13 - X-position and percent of chord length of each tapping.....	104
Table 14 - Common load cell types [22].....	112
Table 15 - C_I Coefficient Matrix for measuring the load applied in grams	142
Table 16 - Total root-mean-square error (in N) using C_I matrix for estimating the 40 loading conditions used for calibration.....	142
Table 17 - C_{vw} Coefficient Matrix for measuring the load applied in grams.....	143
Table 18 - Total root-mean-square error (in N) using C_{vw} matrix for estimating the 40 loading conditions used for calibration.....	143
Table 19 - C_{wI} Coefficient Matrix for measuring the load applied in grams	143
Table 20 - Total root-mean-square error (in N) using C_{wI} matrix for estimating the 40 loading conditions used for calibration.....	144
Table 21 - table to show channel number on Pixhawk and intended purpose	156
Table 22 - Pixhawk parameters relating to the intended function of each channel and descriptions	161
Table 23 - Comparison of Simulation Lift Coefficient to Experimental Lift Coefficient.	168

Nomenclature

UAV	– Unmanned Aerial Vehicle
MEMS	– Microelectromechanical Systems
PCB	– Printed Circuit Board
GDP	– Group Design Project
SPI	– Serial Peripheral Interface
I²C	– Inter-Integrated Circuit
SCL	– Serial Clock Line
CLK	– Clock Signal
SS	– Slave Select
MISO	– Master In Slave Out
MOSI	– Master Out Slave In
SIFO	– Serial In Parallel Out
ADC	– Analogue to Digital Converter
BOM	– Bill of Materials
PLA	– Polylactic Acid
TCP/IP	– Transmission Control Protocol/Internet Protocol
UDP	– User Datagram Protocol
GPIO	– General Purpose Input / Output
Cl	– Coefficient of Lift
SLSQP	– Sequential Least Squares Programming
CAD	– Computer Aided Design
EDMC	– Engineering Design and Manufacturing Centre
GPS	– Global Positioning System
RC	– Radio Control
ESC	– Electronic Speed Controller
BEC	– Battery Eliminator Circuit
PWM	– Pulse Width Modulation
PPM	– Pulse Position Modulation
2D	– Two-Dimensional
3D	– Three-Dimensional

Symbols

x	– Position along the chord from leading edge
$T(x)$	– Thickness at x
t	– Maximum thickness
c	– Chord Length
n	– Number of sensors
α	– Angle of attack
γ	– Flap Angle
ϵ	– Error difference
w_i	– Weighting factor for a sensor
w	– Upper to lower surface prediction weighted average ratio
r	– Ratio of the span occupied by the flap
Cl	– Overall lift coefficient
Cl_γ	– Lift coefficient at the flap angle of γ
$Cl\gamma_0$	– Lift coefficient at zero flap.
ν	– Maximum deflection
W	– Total load applied
l	– Length of the beam
E	– Modulus of Elasticity
I	– Moment of inertia of the beam
D	– Outer diameter of the carbon fibre tube
d	– Inner diameter of the carbon fibre tube

1. Introduction

In any aircraft, the role of the wing is to generate a lifting force sufficient enough to keep the aircraft in the flight. This upwards force can be represented as a pressure difference between the upper and lower surfaces of the wing. The pressure distribution across each surface varies with the shape and angle of attack of the wing and can be analysed to predict the lift.

Pressure measurement experiments over surfaces are typically seen in wind tunnel experiments [1] in which neither the test subject or sensors are required to move. However, aerodynamic sensing across unmanned air systems have been studied to increase accuracy in flight control sensing and stall detection at high angles of attack [2]. However, this was only performed with a small selection of sensors. An increase in the number of sensors could further improve the accuracy of aerodynamic sensing and increase the number of potential industrial applications.

With the introduction and increasing popularity of Microelectromechanical Systems (MEMS), components are decreasing in size with pressure sensors included [3]. These are ideal for measuring the pressure distribution across the surface of a wing in-flight requiring minimal space and without contributing any significant mass. Knowledge of the pressure distribution over the wing during flight can allow for accurate in-flight analysis and control to be performed.

A primary use for this would be to maintain constant trajectory in flight. An aircraft which cruises with constant lift would be expected to maintain a constant trajectory. However, during flight the wing will be subject to gusts and ground effects which alter the pressure distribution and subsequently the lift. This change in lift can result in a significant change in pitch and altitude and cause the aircraft to deviate from this ideal constant trajectory.

The ability to anticipate or simply measure these changes in lift and oppose them should allow for an aircraft to maintain a close to constant trajectory.

1.1.Aims & Objectives

1.1.1. Aims

This project investigates a method to test for the feasibility and accuracy of measuring the pressure across a finite wing using a wind tunnel load measurement device. In addition to this, a set of a pressure sensors across an unmanned air vehicle wing were to be implemented and tested in flight conditions. Furthermore, methods of determining the lift and correcting to maintain constant lift from a pressure distribution are investigated.

1.1.2. Objectives

The objectives specific to this project are:

- Develop a system to measure the pressure distribution across a main wing.
- Choose suitable sensing locations to measure the pressure distribution across the wing.
- Design and build a test rig for wind tunnel testing to validate the pressure distribution data.
- Design and build a UAV to gather live pressure data during flight.
- Implement a control system to maintain constant lift across the main wing of a UAV.

1.2. Project Management

1.2.1. Weekly Group Meetings

Typically, the members of the Group had meetings at least once a week to update each other on their progress. The meetings provided a good opportunity for the group to measure their progress against the Gantt chart and discuss any ideas. They allowed group members to give input into other areas of the project that were not directly under their role. Most meetings were also attended by the supervisor, who was updated on the progress of the project and provided any required help and input to the members of the group.

1.2.2. Media

To facilitate the sharing of files a google drive was created, enabling the group to have a single place to share any files, photos or code. A Facebook group was also created to enable the group members to stay in contact. This social media platform enabled the team to plan meetings and provide a way for members to quickly get in contact with each other.

1.2.3. Time Management

Time management of a project is an integral part, enabling the group to track the progress of the project against the final deadline on the 21st April 2017. The project was spread over 26 weeks with specific milestones set by the university. The group schedule was discussed at length by the team in the first month of the project.

1.2.3.1. Gantt Chart

A Gantt chart was subsequently chosen as the method for keeping track of tasks. During the projects duration, more tasks were added to the Gantt chart when they arose. This chart was then uploaded onto the groups google drive. This enabled each of the group members to see and edit the chart as needed, providing a reminder of which tasks still needed to be done. The presence of the chart helped streamline the workload and make the project more efficient. The full Gantt chart is shown in Appendix E. The main categories of the Gantt chart are described below.

- **Report deadlines** – This category included any deadlines set by the University of Southampton, such as forms and presentations. These deadlines ensure that the project is completed to the Universities standards.
- **Design** – This project was based around designing the wings, the systems that fit inside and a test rig that can collect data inside a wind tunnel. This area also includes any coding and simulation. These tasks were the foundation on which the rest of the project would then build upon.
- **Manufacture** – The next stage of the project is the manufacture of the designs. The

manufacturing of some components was to be done by the EDMC, this introduced the possibility of delays thus enough time was set aside in preparation. The required assemblies included the test rig, UAV fuselage, wing sections and the control systems. As there were two scheduled flight days, it was imperative that all assembly pertaining to the UAV was completed in the week before, allowing for calibrations.

- **Testing** – This category includes testing of wing sections inside the wind tunnel and the gathering of data during the flight days. Due to the busy schedule of the wind tunnel time slots were booked well in advance.

1.2.4. Bookkeeping

A spreadsheet was created at the beginning of the project to keep track of funds left, as well as to see what components were brought. A small part of the funds was set aside for emergencies, such as replacement of broken components or extra materials, this would prevent any unnecessary delays incurred while finding extra funds. To keep accurate records of purchases members sent copies of their invoices and receipts to the treasurer to have physical copies in one place. Below in Table 3 is a summarised chart of the budget. The full list with all columns is available in Appendix A.

Table 3 - Summarised budget for group 13, October 2016 to March 2017.

Group 13 Budget					
Part description	Company	Quantity	Total	Who	Date Ordered
V-Slot Linear Rail - 20x20x500mm	Technobots Ltd	2			
V-Slot Linear Rail - 20x20x1000mm	Technobots Ltd	1	£26.44	A. Hill	29/11/2016
6x CNBTR aluminium YZC-131 Kitchen scale weighing sensor load cell 5Kg	CNBTR	1	£17.06	R. Pasupathy	17/11/2016
2x HX711 weighing sensors AD Module load cell	FlyFun Tech.	1	£9.98	R. Pasupathy	17/11/2016
Replacement electronic scale 0-20Kg range weighing sensor load cell	Sourcingmap	1	£8.03	R. Pasupathy	17/11/2016
Collar One Piece Screw, Bore 10mm, OD 20mm, W 10mm, Steel	RS Components	2	£3.29	A. Hill	02/12/2016
Test Rig materials without work hours	EDMC	1	£55.00	Felicia	14/12/2016
Metric mounted ball bearing 8mm	S. Pawlick	10	£49.90	Felicia	28/11/2016
Pultruded CB tube 10mm 1m	Easy Composites	1			
Pultruded CB tube 8mm 1m	Easy Composites	1	£28.26	L. Winter	16/12/2016
Flow perturbation rig material	EDMC	1	£20.00	N. Tarasov	03/01/2017
UAV Fuselage	Bormatec	1			
Landing Gear	Bormatec	1	£474.76	L. Winter	13/02/2017
Counter sunk Machine screws x25	AHC Ltd	1	£3.49	Felicia	16/02/2017
String	K's Market	1	£2.34	Felicia	22/02/2017
10mm Carbon fibre tube - 1.2m	Easy Composites	2			
8mm Carbon fibre tube - 1.2m	Easy Composites	2	£110.54	L. Winter	14/03/2017
2x Brushless Outrunner Motor		1	£71.97	A. Hill	17/03/2017
Diamond sRH771 antenna	Radioworld	1	£30.44	A. Hill	15/03/2017
Rx 8 channel PPM receiver	Root	1	£26.99	A. Hill	14/03/2017
	Total		£938.49	Budget of Project	Funds left
				£860.00	-£78.49

In the weeks before the two flight days the group still required a new pair of motors, as the original Nes-591 motors were too heavy for use in the test rig, so servos were considered and then ordered. Furthermore, the group also required a new antenna and a receiver as the faculty did not have any to spare at short notice. These new purchases would have put the group over budget. After speaking to the supervisor, he was willing to cover the extra funds for the parts, as he wished to use them for future faculty projects.

1.2.5. Reflection

The task schedule was not followed exactly, as some tasks were started ahead of schedule while others were delayed. The most incurred delays were to do with manufacturing, mostly due to replacement of broken/defective components, waiting times to use the facilities in the 3D printing centre and for the EDMC to finish the group's orders. Overall most scheduled tasks were completed on time, with the test rig being finished in time for testing and the UAV being flight ready for the flight days. Due to the prioritisation of other groups needing to use the wind tunnel facility we were not able to get an extra slot for addition pressure readings in time for the final report, but there are slots available in May to get results in time for the group presentation.

The group's utilisation of the budget has been good for the duration of the experiment. Most of the expenditure occurred for the fuselage and landing gear which accounted for more than half of the spending. In reflection, it would have been possible to utilise the foam cutting facilities to make a fuselage from scratch, but due to the workload being focused on the design and testing of the wings, this would introduce delays in weighting calculations and queues for the foam cutter. Overall all brought products were utilised efficiently and only a few breakages occurred, for instance the bearings for the test rig needed reordering.

2. Pressure Measurement System – Electronic Development

2.1.Industrial Motivation and Brief

The design and testing of the pressure measurement system was done in collaboration with a 3rd party partner who wished to remain anonymous in the final report. As such, they will be referred to as our ‘3rd party partner’ for the rest of the report.

With the high paced development of MEMS based sensor technology in recent years - which has been pushed by the mobile revolution - small, accurate and cheap digital aneroid barometers have come to market for primary use in mobile devices, such as your smart phone.

The latest edition of these sensors can measure absolute pressure at frequencies of approximately 100 Hz, have an accuracy in the order of Pascal’s, take up a planform area of less than 9 mm² and cost around £2 per sensor. As such, the capabilities of these sensors mean their use is not just restricted to mobile devices, and they are beginning to find use in both the automotive and aerospace industries.

For example, in experimental wind tunnel testing, in which many pressures readings are generally required, traditionally a pressure scanner such as Measurement Specialities® ESP64-HD, or Scanivalves® MPS4264, are placed within the test object and pneumatic tubing is run from these, to all the pressure tapings. If these tapings lie in components which are too thin to house the pressure scanner (such as in the wings), the scanner is placed somewhere else, and long lengths of tubing are run between the scanner and the tapings. This has a few key drawbacks:

- The length of tubing between the pressure tapping and the scanner effects the response measured by the scanner
- Running hundreds of tubes through components is time consuming, volumetrically inefficient and can make fault finding extremely difficult
- It may not always be possible to run tubing to the desired location, limiting testing possibilities.
- These devices are expensive so the use of multiple is generally impractical

MEM's barometric sensors can avoid these problems by enabling you to measure the pressure directly on the surface - utilising flexible PCB technology - Or, by manufacturing a 'miniaturised pressure scanner' that can be placed closer to the surface of interest, significantly reducing the effect of tube length. Per sensor these devices are also much cheaper than the ESP64-HD and the MPS4264, meaning the measurement of 100's of pressure tapping can be achieved at a fraction of the cost. Details of three of the latest generation of MEM's based barometers can be found in Table 4.



Figure 2 - Measurement Specialities ESP64-HD, a 64 port pressure scanner



Figure 3 - Scanivalves MPS4264, a 64 port pressure scanner

Table 4 - A Comparison of 3 MEM's based absolute pressure sensors

Name	MS5611	LPS25H	BMP280
Picture			
Dimensions (mm)	5.0 x 3.0 x 1.0	2.5 x 2.5 x 1.0	2 x 2.5 x 0.95
Measurement Noise (mbar)	0.03	0.03	0.03
Range (mbar)	10 to 1200	260 to 1260	300 to 1100
Interface	I ² C or SPI up to 20 MHz	I ² C or SPI up to 10 MHz	I ² C or SPI up to 20 MHz
Measurement time (ms)	1	3	5
Unit Cost (if purchasing 100 on 04/04/2017)	£6.71	£1.98	£1.65

One of our group members had already worked with our 3rd party partner, during their year in industry, integrating these types of sensors into a ‘close to surface’ measurement application. This work consisted of placing 5 Bosch® BMP280 sensors inside an aerofoil section. Both our group member and the 3rd party partner were interested in continuing this research through the GDP. In particular, our 3rd party partner had a requirement to develop a small ‘pressure scanner’ that could easily fit within a very confined space. This requirement had a good synergy with the aim of this GDP to “measure the pressure distribution across a UAV wing”. After liaising with our 3rd party partner an agreement was reached that meant:

- The GDP would develop a ‘pressure scanner box’ with a minimum of 32 individual barometric pressure sensors.
- The Designs would incorporate the Bosch® BMP280 sensor.
- The Device, including manifolds will not exceed the dimensions 11×45×100 mm.
- The Device can be powered from a supply input range of 3 V – 24 V.

- The GDP would design electrical schematics, and component layout DXF files, for the required PCB boards.
- PCB routing and manufacture would be outsourced to ‘Advanced Layout Solutions’, which the 3rd party partner would pay for.

2.2. The Bosch BMP280 sensor

At the start of this GDP, the Bosch® BMP280 barometric sensor was Bosch Sensortec’s® latest edition to their barometric pressure sensor range, (In January 2017 they announced the BMP380). If like many in this GDP group, you have an iPhone 6®, Samsung S6®, or later, your phone uses a BMP280 sensor to record altitude information, such is there popularity and accuracy. The key facts of the BMP280 are listed in Table 4.

The BMP280 sensor is a digital absolute pressure measurement sensor. It has on-board temperature and pressure sensing elements, which are converted into stored logical values using the on-board, 20 bit, ADC. A master device, such as a microcontroller, can communicate with a BMP280 over a choice of 2 communication protocols; the Serial Peripheral Interface Bus protocol (SPI) and the Inter-Integrated Circuit Bus protocol (I²C).

2.2.1. *Communication Protocol Selection*

I²C is a multi-master two wire communication protocol. The two I²C wires are the ‘serial data’ line (SDA), which is what all the data is transferred across, and the ‘serial clock’ line (SCL), which enforces the communication speed on the bus. All devices connected to the bus have unique 7-bit address, which are self-assigned. These addresses are used by the ‘master’ devices to communicate with specific ‘slave’ devices on the bus (like a BMP280 sensor). To ensure the hardware can be this simple (only two wires), the protocol itself is quite complex, meaning communication speeds are enforced at either 100 kbps, 400 kbps or 3.4 Mbps. A typical I²C bus is shown in Figure 4. Comparatively SPI is quite straightforward, it is a single master, 4 wire, communication bus. The 4 wires are:

- The clock signal (CLK), which defines the communication speed and is sent from the master device.

- The Slave select line (SS), individual to each slave, it is used by the master to select the slave it wishes to communicate with
- The first data line, *Master Out, Slave In* (MOSI), which is what the master uses to send data to slaves.
- The second data line, *Master In, Slave Out* (MISO), which is what the slaves use to send data to the master.

As the SPI protocol only ever has one master, the protocol is much simpler than I²C, meaning the entire protocol can be implemented in software quite easily. Due to this simplicity, the bus can also be driven at any clock speed, up to a limit stated by the manufacturer (for the BMP280 this is 10 MHz). To address multiple devices the CLK, MOSI and MISO lines are shared between all devices, but each slave has a separate SS line, as shown in Figure 5. It should be noted here the BMP280 sensor can only hold one of two I²C addresses, and you can switch between the two by asserting one of its connections ‘high’ or ‘low’. Meaning in effect each device would need a slave select line for the I²C setup as well.

Considering all of this, SPI was chosen as the protocol that would be used to communicate with the BMP280 sensor, primarily due to its faster communication speeds (as the problem of communicating with multiple sensors at once, was initially thought to be limited by communication speeds)



Figure 4 - A diagram of a typical I²C bus with two slave devices

Figure 5 - A diagram of a typical SPI bus with two slave devices

2.2.2. Basic Principles to Communicate with a Single BMP280 Sensor

In this section, the basic principle to communicate with a BMP280 sensor are outlined. It is assumed a SPI bus has been created between the sensor and a master controller.

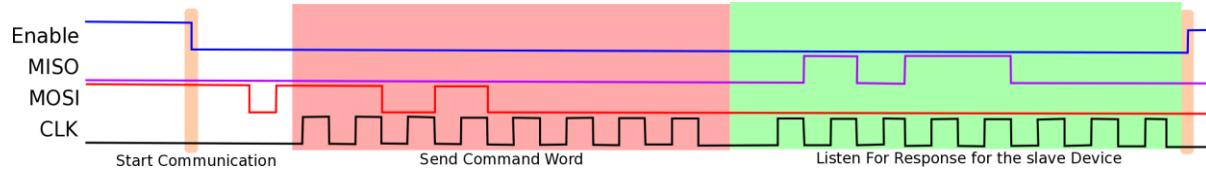


Figure 6 - A diagram illustrating how a single SPI transfer would work between a master and a slave device

First a typical SPI transfer (such as that shown in Figure 6) consists of:

- The master pulls the SS line of the device it wishes to speak to ‘low’
- The master then sends out an eight-bit command word, which the slave reads
- If required, the slave then sends back an eight-bit response
- All values are read on the rising edge of the clock line (SPI mode 00)
- The master brings the SS line high to end the communication

To be able communicate with a BMP280 the master needs to do some initial setup:

1. Firstly, the master checks if a BMP280 exists on the current slave select line. To do this the master does an SPI transfer of the command word 0xD0. And checks the response matches the hex value 0x58. If it does, the master has found a BMP280.
2. Secondly, each BMP280 is calibrated individually by the manufacturer, and 12 calibration coefficients are subsequently stored in the BMP280’s non-volatile memory. These coefficients are read over a series of SPI transfers and stored in the master device for later use.
3. The device mode is set as ‘forced’ (the reason for this is explained in a later section)

Once the initial setup has been complete you can begin to take measurements from the BMP280.

To do this you need to:

1. Measure data - Send the command word 0x25 to ask the BMP280 to record a new set of data.
2. Read Data - Use a series of SPI transfers to read the latest stored values for both the temperature and the pressure.
3. Use these measurements, in conjunction with the pre-acquired calibration coefficients and the algorithms supplied in the BMP280 Datasheet [4], to calculate a temperature compensated pressure measurement.

2.2.3. Prototyping – Communicating with 4 BMP280 Sensors

Initial prototyping was done using an Adafruit® Huzzah as the master device. The Huzzah is a breakout board for the esp8266 chip, which is a small, 2.4 GHz wifi enabled, 3.3 V logic, 80 MHz micro-controller that can be programmed using the Arduino programming environment [5]. An Image of one can be seen in Figure 8.

Four BMP280 sensors were connected to the microprocessor as described in Figure 7. The initial code to communicate with these sensors was based on the BMP280 driver files [4], which are freely distributed by Bosch Sensortec®, but they were largely modified remove unnecessary content and enable it to communicate with multiple sensors.

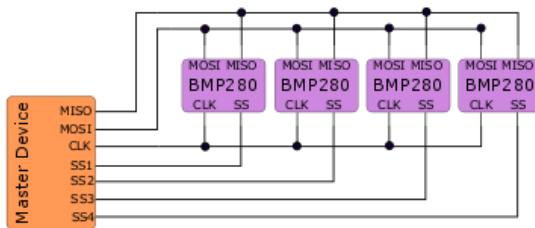


Figure 7 - Simple schematic for the electrical setup of four BMP280 sensors

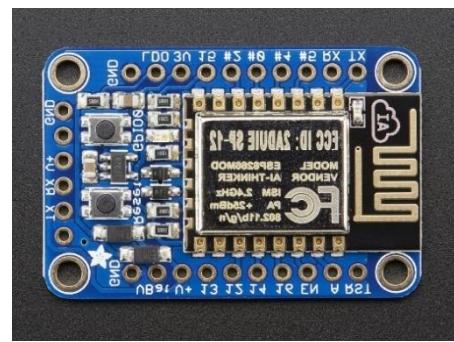


Figure 8 - The Adafruit Huzzah Microcontroller

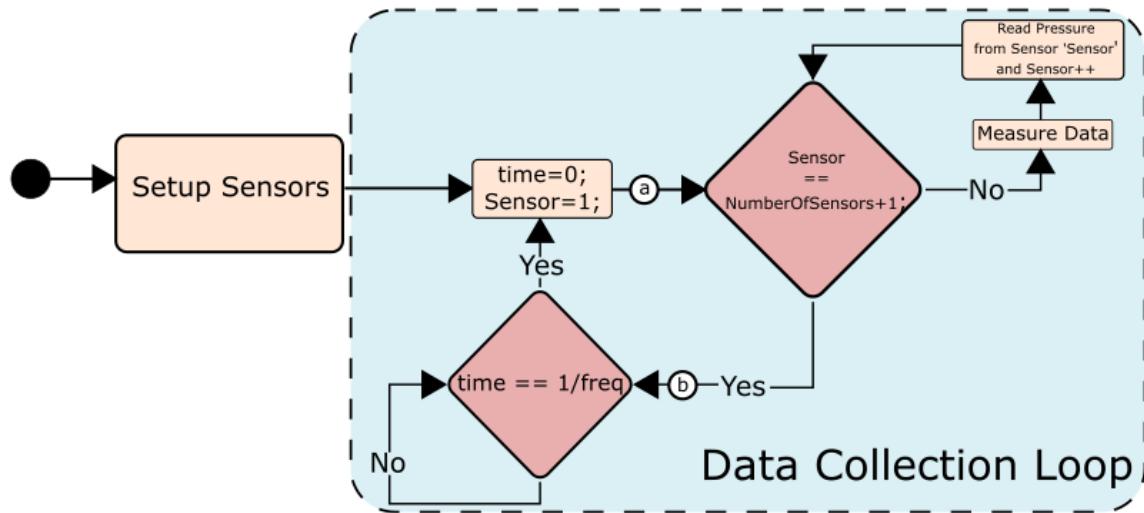


Figure 9 - A simple flowchart of the process followed in the main method of the initial code

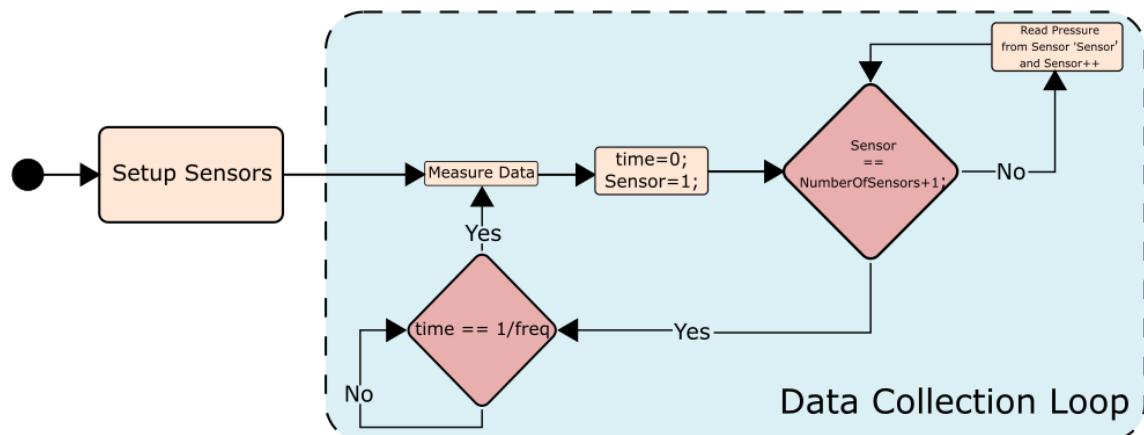


Figure 10 - A simple flowchart of the process followed in the main method of the initial code, included the synchronous measurement of pressure across all sensors

In the final version, a header file (BMP280.h) now exposes the class ‘*BMP280*’, when you initialise an object of this class, you provide the communication speed required and the pin for the SS line. The object then initialises the BMP280 sensor, as described in the previous section, and stores the calibration coefficients in a private variable. The class also exposes 2 public methods, *MeasureData()*, which tells the BMP280 to collect new data, and *ReadPressure()*, which returns a temperature compensated pressure measurement. The main method, which initially followed the process described in Figure 9, used this class to collect sensor data at a set frequency. By using Arduino’s internal timing methods, the time taken to go between markers ‘a’ and ‘b’ in Figure 9, was found to be approximately 1 ms, meaning the measurement process could effectively be run at a sampling rate approximately 1 kHz, with 4 sensors.

At this point it was realised that by setting the Sensors to ‘forced mode’ (which means the ‘Measure command’ must be sent to the sensors for them to take a new measurement) and then pulling all the SS lines ‘low’ simultaneously, the ‘Measure command’ could then be sent to all devices simultaneously, meaning they would all record a pressure synchronously. This meant the system could measure multiple BMP280’s synchronously, then send the data asynchronously from each sensor to the master. Eradicating the time delay between the measurements of different pressure sensors. The new process can be seen in the flow chart in Figure 10.

2.2.4. Developing a scalable Sensor selection strategy

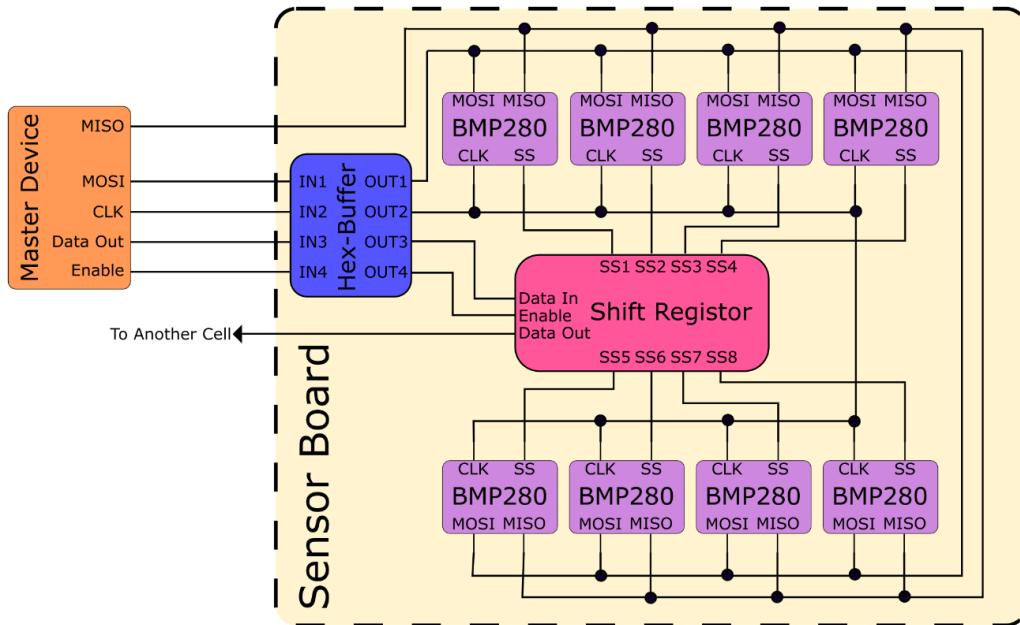


Figure 11 - A simple schematic showing the final design for each cell of 8 BMP280 sensors

In all previous work, all devices had separate SS lines, however this was not very scalable, as a system with 32 sensors in would require 32 free I/O ports on the master. Hence, a new strategy was needed to ensure each sensor could be selected individually. It was found that ‘Serial in, Parallel out’ (SIPO) shift registers are commonly used to increase the I/O ports available to microcontrollers.

A SIPO shift register is a cascade of digital flip-flops that a binary bit is pushed through on each rising edge of the clock input. The first flip flop is ‘fed’ a binary bit from the *Data-in* pin on the shift register. Hence by systematically pushing through a single binary 0, and using the enable pin on the shift resistor as the masters’ slave select pin, different BMP280’s can be selected, as shown in Figure 12. From this figure it is obvious that if you fill the shift register with binary zero’s all the sensors can be selected at once, as is required for the call to *MeasureData()*.

The value in the final flip-flop of a shift register is exposed via the *Data-out* pin, by connecting this to the *data-in* pin on another shift register you can effectively increase the size of your cascade infinitely, allowing for the individual selection of *limitless* sensors.

2.2.5. Splitting the micro-controller and sensors onto separate Boards

At this point it was decided that in the final product the microcontroller and sensing equipment would be split onto separate boards. This meant stock microcontrollers could be used and the design of the ‘Sensor Board’ would be vastly simplified, this is shown conceptually by the hashed box in Figure 11.

Following on from this decision it was realised that because the sensor board needed to operate with a wide input voltage range, and that it would have no knowledge of the microcontroller being used (in particular it would not know at what logic level the bus lines would be operating at), the Sensor board should be able to regulate the logic level on the incoming SPI bus, to do this a Hex buffer was added to the prototype as shown in Figure 11.

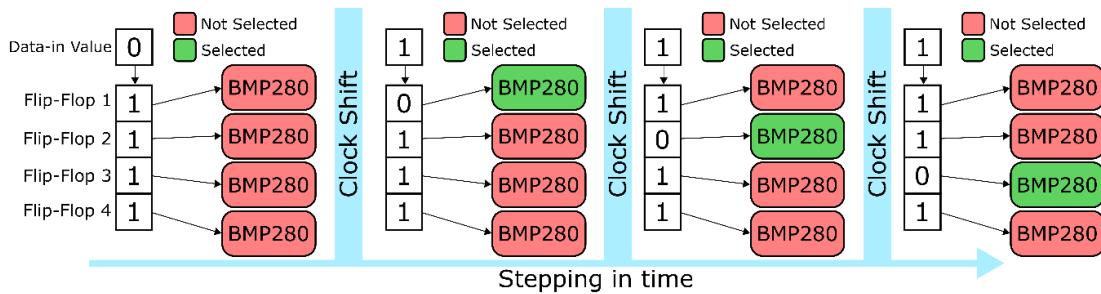


Figure 12 - A diagram illustrating how shift registers can be used as the SS line for multiple devices

2.2.6. Final Sensor Board Schematic

Once the prototype with 4 sensors, a shift register, and a hex buffer worked effectively, it was decided to proceed and design the schematic for the final sensor board. As shift registers commonly come with 8 outputs it was decided to build the final design up in cells of eight sensors. The data-in / data-out pins on each shift register would then be used to connect each of the cells together. SPI bus lines would also be common. A schematic of an eight-sensor cell can be seen in Figure 11. Other key changes and additions to the final schematics were:

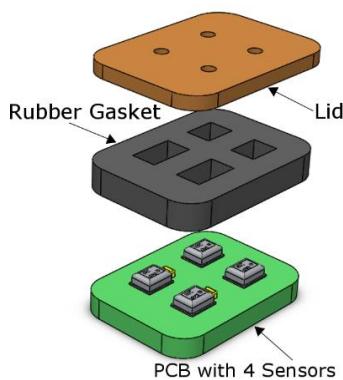


Figure 13 - Exploded view of an exemplar sensor sealing design philosophy

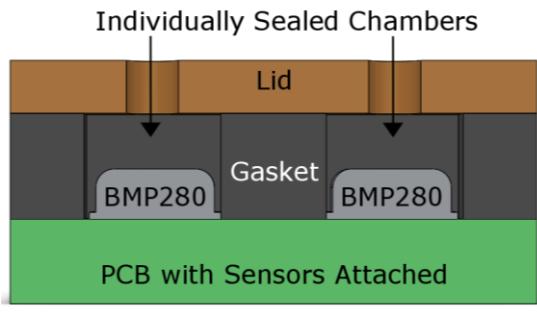


Figure 14 - Section view of the sealed chambers created for each sensor using the exemplar sensor sealing design philosophy

- In total, it was decided to have 64 BMP280's on each sensor board, the rational for which is described in the next section.
- To save board space only two hex-buffers chips (8 buffers) were used in the final design. This meant bus lines were shared between 16 sensors rather than 8.
- All connections between the sensor board and the master device were bundled into junctions
- The *Data out* pin of the final cell of sensors was also added to the junction so that sensor boards could be 'daisy chained' together
- To comply with an agreement with our 3rd party partner a voltage regulator was added to the board that would have a broad input spectrum.

- To increase the accuracy of the measurement, a decoupling capacitor was added to every other sensor in the aim to prevent the input voltage fluctuating when the on-board ADC digitised sensor readings.

The final Schematic, which includes all of the above points can be seen in Figure 22.

2.2.7. PCB component Selection and layout

To select a number of sensors, we first had to consider how each sensor would be independently sealed. The basic concept was to extend a strategy implemented on similar projects by our 3rd party partner. Firstly, the sensors would be evenly spaced on the same side of a PCB. A laser cut rubber gasket would then be placed on top of the sensors, which would in turn be compressed by a lid. This would give each sensor and individually sealed chamber that could be connected to an external pressure source via a hole in the lid. This concept is illustrated diagrammatically in Figure 13 and Figure 14.

It was decided to use a rectilinear pattern of sensors of the final board, the minimum spacing between sensors to ensure an effective seal was unknown. In the past 3 mm spacing had been used effectively by our 3rd party partner, so it was agreed we would trim this down to 2 mm in our design to push the boundaries of this technology. Additionally, in the proposed design a decoupling capacitor was included with every other sensor. These had to be placed as close as possible to the sensors to have the desired effect. Placing the capacitors on the opposite side of the PCB was explored however this would have dramatically increased the manufacturing cost, so was quickly discounted. This meant the capacitor had to be placed on the same surface as the sensor, so needed to be included in the sealed chamber of every other sensor. To minimise disruption, size 0602 capacitors were used, which have dimensions 1 x 0.5 x 0.5 mm, meaning they could easily be placed alongside the BMP280's. The final layout of a cell of 4 sensors, including the decoupling capacitors, can be seen in Figure 16.

The initial brief was to include 32 sensors on the board. With 2 mm spacing, the planform area of a grid of 8 sensors by 4 sensors was either 36 x 21 mm or 40 x 19mm (depending on the orientation), which were well within both ours and our partners size requirements. Therefore, the decision was taken to double the number of sensors to 64, which would take up an area of 36 x

42 mm, which was still well within our space requirements, leaving plenty of room for the additional components. This drastically improved the versatility of the final product.

All other components were chosen to be as small as possible to reduce the height of the final board, the chosen connectors were from the Molex® Pico-lock range, which have a 1 mm pitch, and were the smallest positive lock connectors we could find, an example of such a sensor can be seen in Figure 15. All final board components can be found in the bill of materials (BOM) in Figure 17. This gave an initial estimate for the cost of components, per board, as approximately £137.

Once all the components were selected an initial proposal for the board layout was designed, as seen in

Figure 23. Additionally, it was to be stipulated that:

- All the sensors would have the same orientation (to minimise issues that if the board flexed the sensor may receive different offset errors if they see different strains).
- All via's were to be filled to ensure an effective seal could be produced on the top surface

2.3.Design of the Sensor Board PCB

The BOM, initial layout and electronic schematic, were first approved internally and then with our 3rd party partner. At which point they were sent to Advanced Layout Solutions® so that they could design and manufacture the PCB's.

Through a collaborative effort between the team and Clive Thorn from ALS, a final PCB design was agreed and sent to manufacture. The design costs were £960, and the cost to manufacture 4 boards was £2422, making the total cost per board £845. The final board is a 6-layer PCB, images of which can be seen in Figure 20 and Figure 21. Our 3rd party partner did not wish to have the final dimensions and exact layout of the sensor board presented in this report so they have not been included.

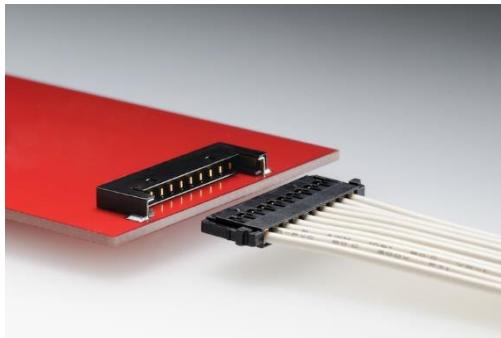


Figure 15 – An example of a Molex pico-lock, board to wire, connector

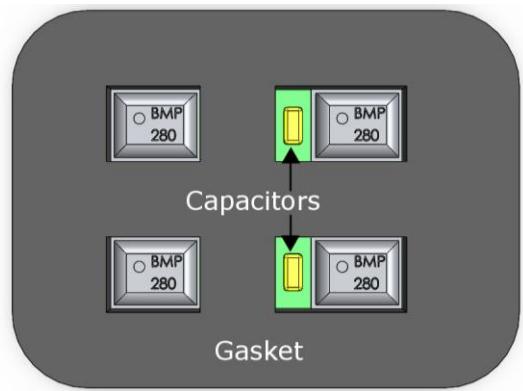


Figure 16 – An exemplar view of the proposed layout of a cell of 4 sensors

Part	Description	Manufacturer	Manufacturer part Num.	Value	Size	Rs stock Num.	Quantity	Cost Per Unit	Cost Per Board
BMP1-64	SMT Pressure Sensor	Bosch	BMP280	-	805	849-6187	64	£2.00	£128.00
C1-33	SMT Capacitor	TDK	CGA2B3X7R1H104K050BB	100nF	402	788-2878	33	£0.03	£1.09
C34	SMT Capacitor	TDK	C1608X5R1A106K080AC	10uF	603	788-2893	1	£0.13	£0.13
IC1-8	VQFN shift register	NXP	74HC595BQ	-	VQFN	791-8753	8	£0.15	£1.20
IC9-10	Hex Buffer converter	Texas Instr.	CD74HC4050PWR	-	TSSOP	662-7131	2	£0.24	£0.48
IC11	Voltage regulator	Microchip	MIC5239-3.3YS	-	SOT-223	910-1840	1	£2.08	£2.08
J1-2	4 pin pico-lock 1mm	molex	503763-0491	-	-	839-9850	2	£0.45	£0.89
J3	2 pin pico-lock 1mm	molex	503763-0291	-	-	839-9856	1	£0.38	£0.38
R1-64	SMT Resistor	Vishay	CRCW040210K0FKEDHP	10 k Ohm	402	812-1562	64	£0.04	£2.37
Total Cost Per Board									£136.62

Figure 17 – Bill of Materials sent to ALS for the design and manufacture of the Sensor Board

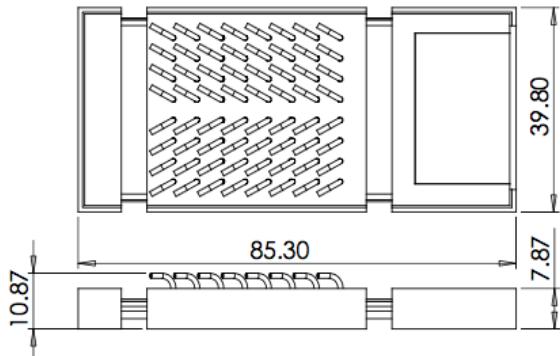


Figure 18 – Final sensor box dimensions with 2.5 mm thick walls.

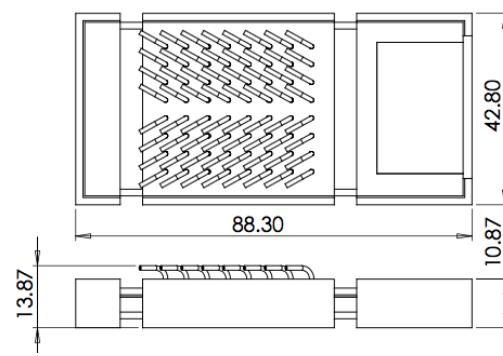


Figure 19 – Final sensor box dimensions with 4 mm thick walls.

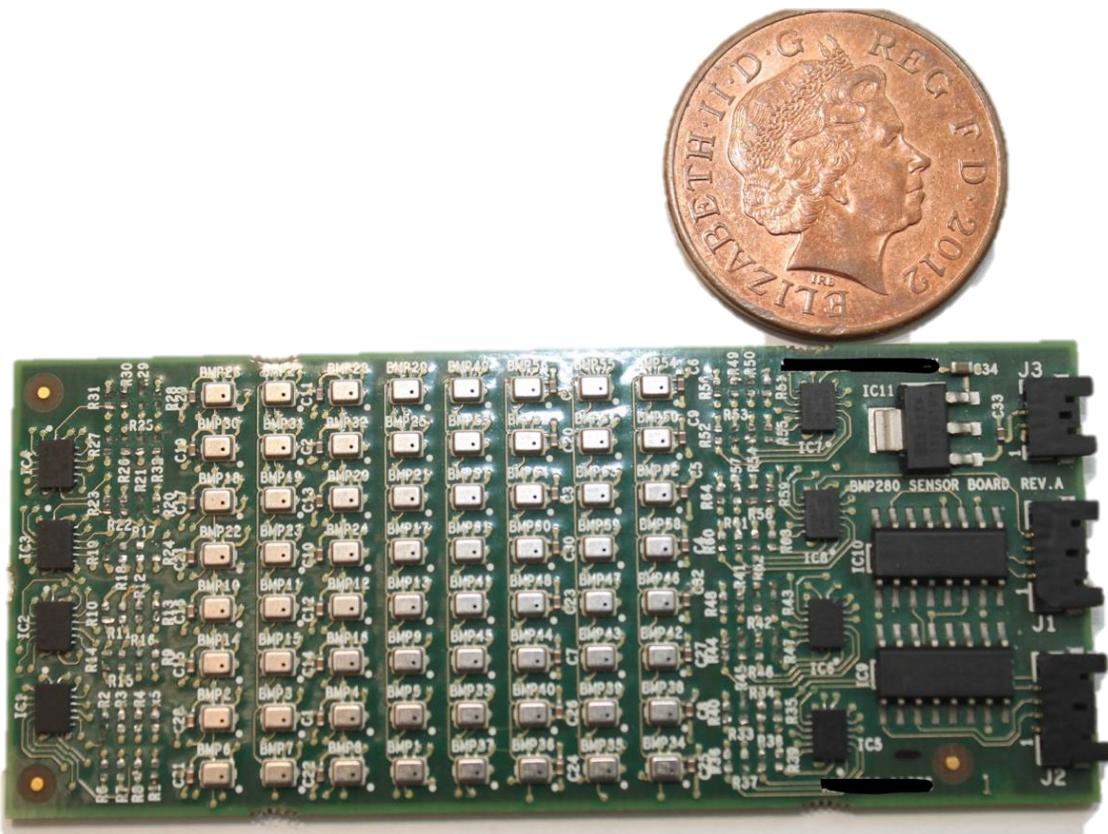


Figure 20 – Top view of the manufactured Sensor PCBs

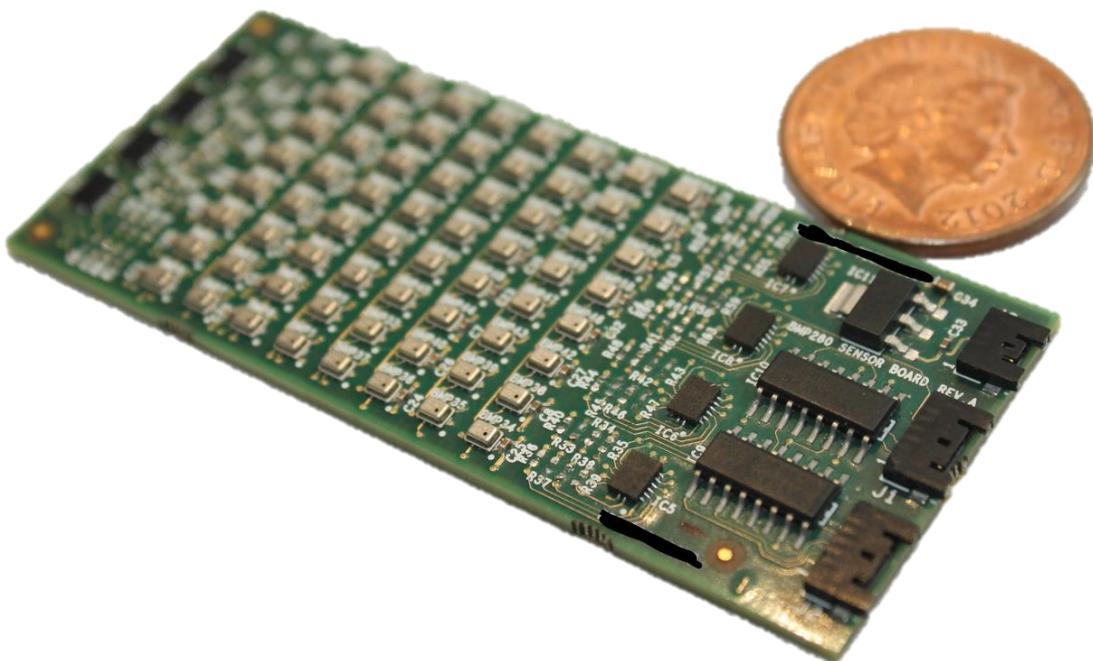


Figure 21 – Isometric view of the manufactured Sensor PCB

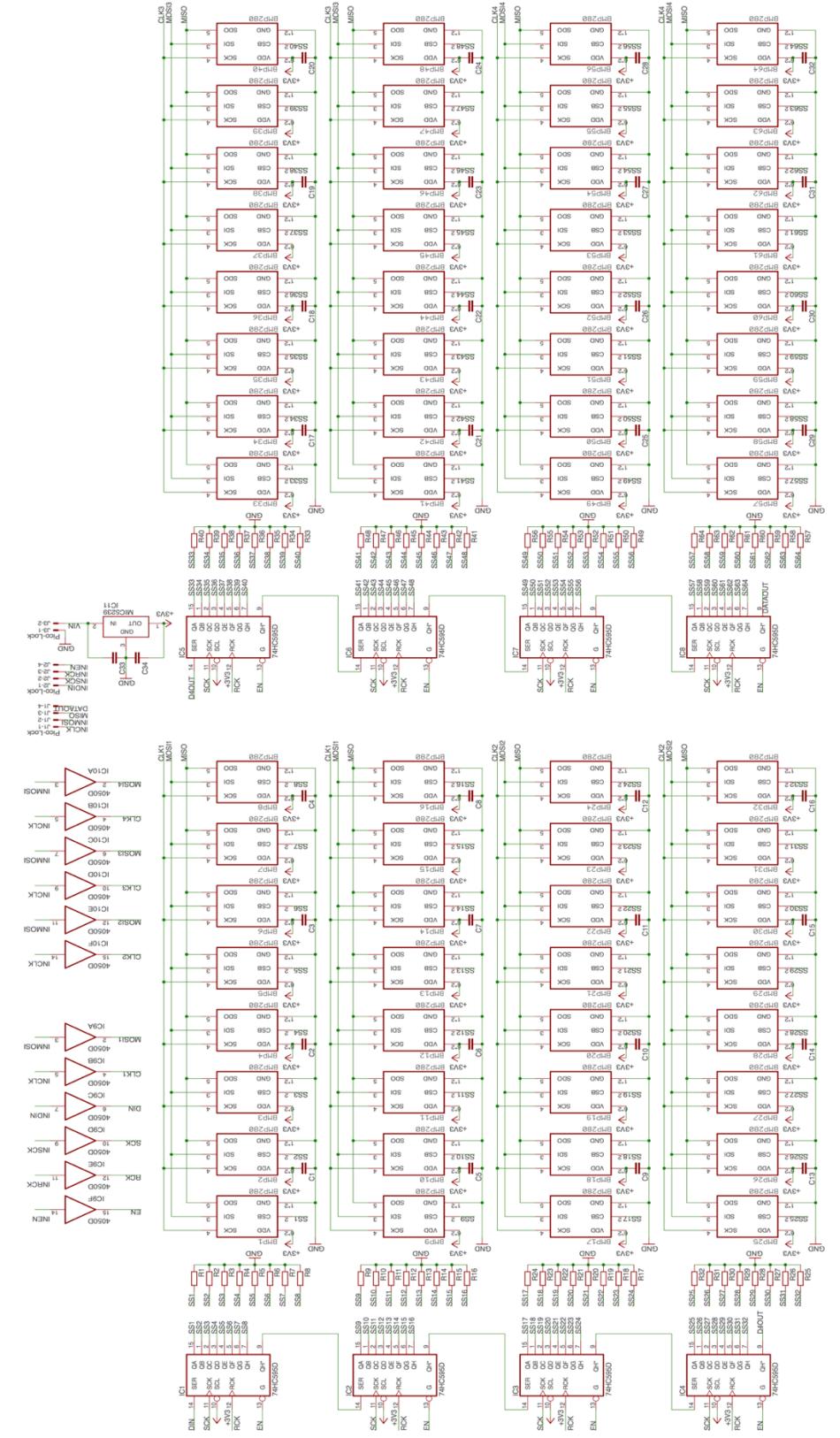


Figure 22 – Final electrical schematic sent to ALS to design and manufacture the Sensor Board

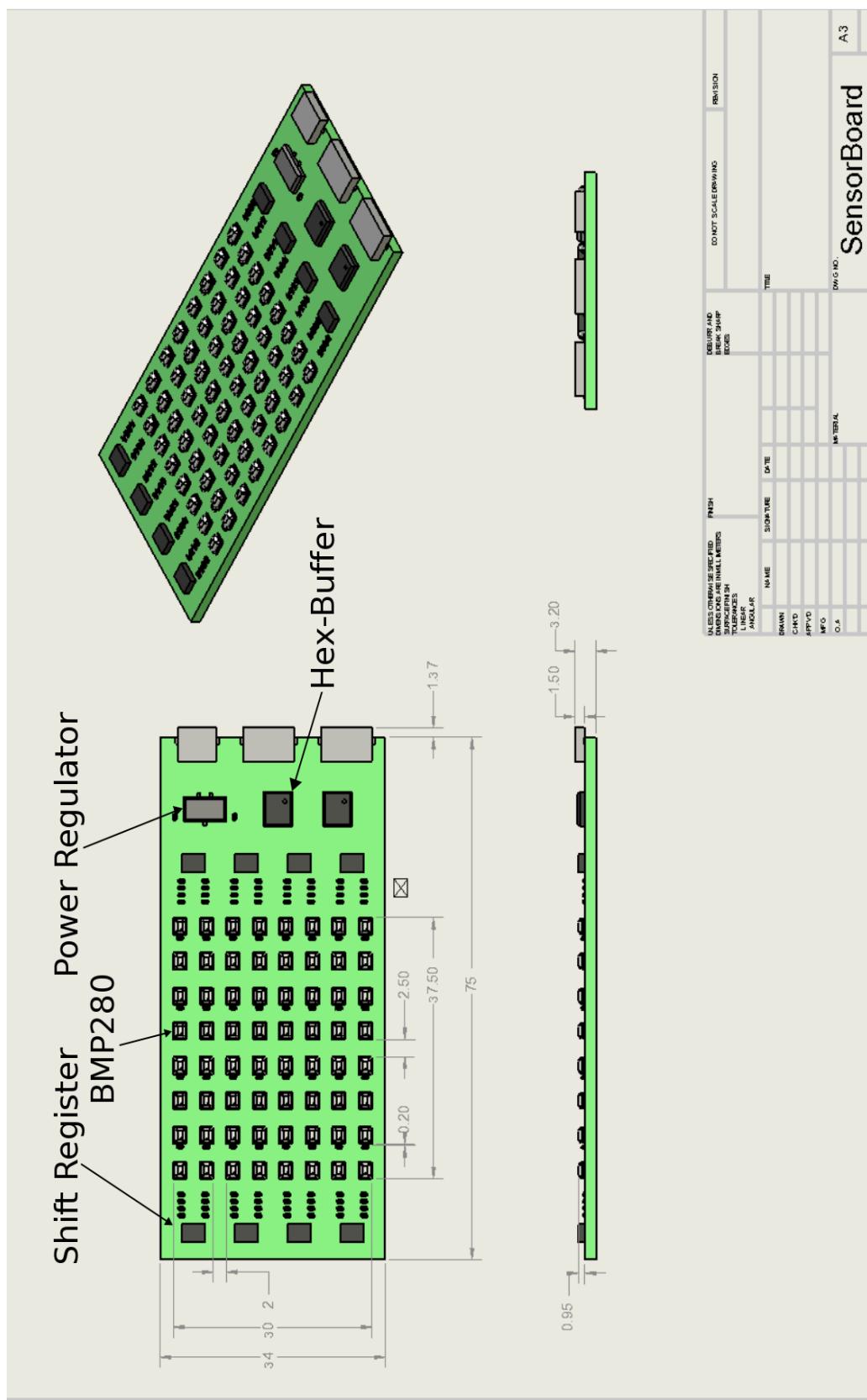


Figure 23 – Schematic of the Proposed initial component layout

2.4.Design of a box to enclose the PCB

A parametric box and gasket was designed to house the Sensor Board. These items were made parametric to ensure thickness, tolerances and sizes could be easily adjusted in case of issues with printing or the design. To be able to clamp the lid to the base, 2 channels have been created in the component so that some zip-ties can easily be used to clamp them together. To be able to get a sealed connection between a piece of pneumatic tubing and each pressure tapping. A piece of hypodermic tubing was bent by 90 degrees and glued into the holes in the lid of the box. This can be seen in Figure 24 and Figure 25.

When designed with a wall thickness of 2.5 mm the sensor box, including hypodermic tubing, meets the tolerances set by our 3rd party partner, as shown in Figure 18. If the products were 3D printed in SLS Nylon these dimensions would most likely have been adequate, however, for the GDP these components where 3D printed in PLA so the wall thicknesses were increased to 4mm to ensure the box was stiff enough to compress the gasket and get an effective seal. The final dimensions of the box used in the GDP are shown in Figure 23. And the box, sensor board, gasket and hypodermic tubing weighed 38 grams in total.

2.5.Micro-controller Hardware and Software design

2.5.1. Philosophy for distributing the data from the microcontroller

To separate the pressure measurement from any control on both the test rig and the UAV, it was decided data should be distributed from the microcontroller to other devices. The most logical way to do this was to distribute the pressure measurement data via a local area Network.

For these 2 protocols are available, TCP/IP or UDP. As more than one other device may be interested in the data at any one time the UDP protocol seemed well suited due to its ‘broadcast’ functionality. This also meant there would be less of a burden on the microcontroller as TCP/IP introduces a lot of additional overheads. To implement this, each time data was collected the microcontroller would create and send a single packet of data, as illustrated in Figure 26. The structure of each data packet was as follows.

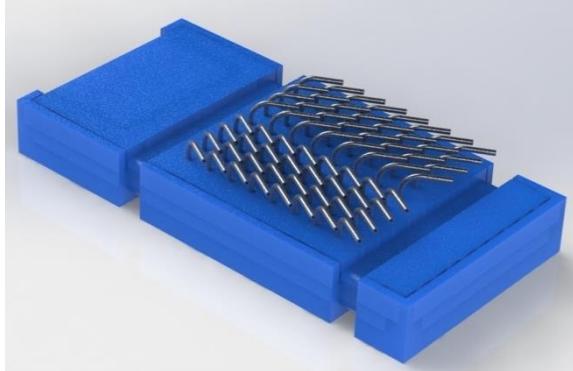


Figure 24 – A render of an assembled Sensor Box

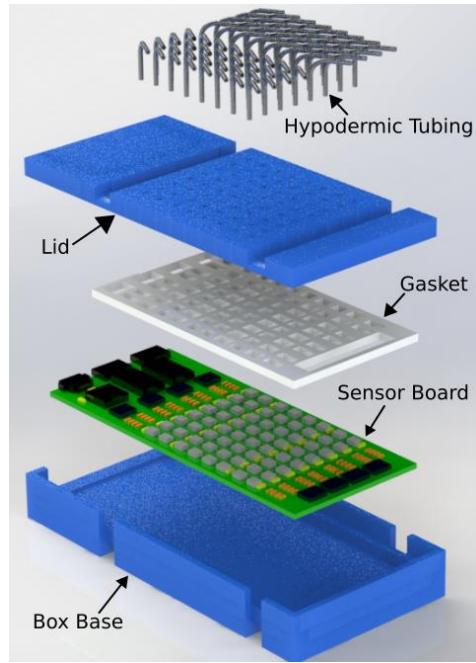


Figure 25 – A render of the exploded view of the Sensor Box Design

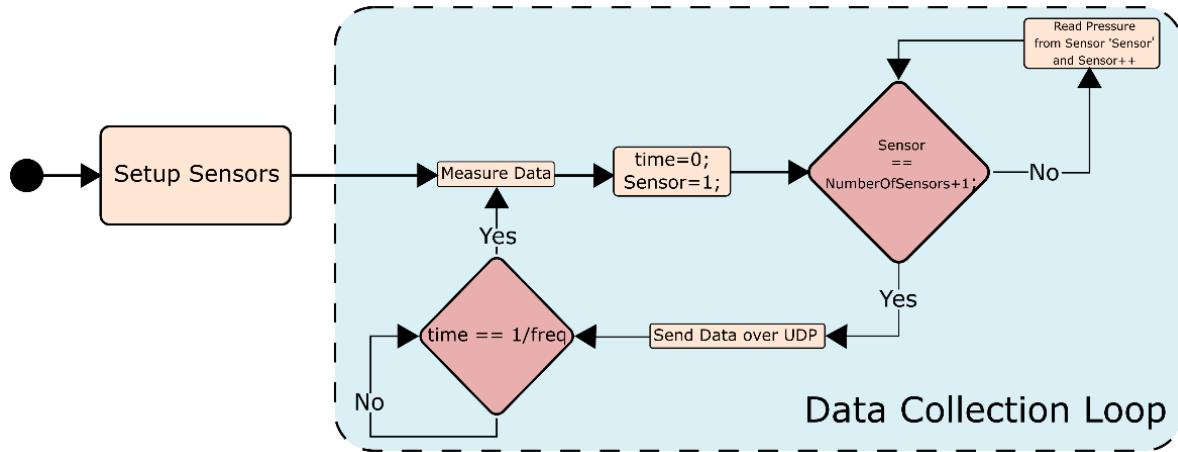


Figure 26 – A simple flowchart of the process followed in the main method on the microcontroller to collect sensor data then distribute it via UDP.

- Bytes 1-4 (int32 datatype) Packet ID – A constant number used to identify the source of the packet (if two micro controllers were sending data they could both use a different packet ID to differentiate the data).
- Bytes 5-8 (int32 datatype) Packet Number – incremented by one with each packet that is sent from the micro controller. Effectively acts as a time stamp, as if you know the data transmission frequency you can work out the relative time between packets, by dividing the difference by the frequency.
- Bytes 9-end of data (multiple double datatypes) Measured pressures – the rest of the packet is made up of the pressure sensor data.

As 4 boards were manufactured the maximum possible size of each data packet was 2056 bytes, which for a data rate of 100 Hz would need a connection speed of at least 250 kbps to operate correctly.

The UDP packet were always sent to the subnets broadcast address (typically 192.168.1.255) and port 55005, allowing the data to be collected and used by any number of other devices.

For the entirety of this project all the data was captured on separate devices using the inbuilt UDP objects in MATLAB 2017a [6]. The values were either stored in a local variable, to be saved at the end of a test, or were used to update a live plot.

2.5.2. Test-Rig micro controller design - using an Adafruit Huzzah

To create a completely wireless system the Adafruit® Huzzah was used as the initial microcontroller. The Huzzah's on-board Wi-Fi module meant it could distribute UDP packets via a local wireless network without any physical connection. Hence, with the inclusion of a small battery pack, a completely remote measurement system could be developed. This was particularly useful for the test rig, as it meant all components of the measurement system could be on the 'live' side of the load cells, eradicating any bridging issues. Due to the compact size of all the components of this system, it could conceivably be completely embedded into the wing of the test section, showing the capabilities of this kind of system.

The code used for this iteration was a re-worked version of the code used on the initial prototype and the additional WI-FI functionality came from the open-source esp8266 Arduino library [7].

When this system was used to measure one sensor board (64 sensors) a data rate of 120 Hz could be achieved.

2.5.3. UAV Telemetry Link Setup

From early in the project it was clear developing a fully closed-loop system in which pressure data would be collected and used to both, predict lift and drag and stabilise the aircraft was not feasible in the time frame of the project. Hence, the primary aim from UAV testing became to demonstrate the capabilities of the measurement system, proving it could be conceivably used for control purposes at a later date.

The primary objective was to live-stream telemetry from the pressure sensors on-board the UAV to a laptop on the ground. To do this 2 Ubiquiti® Bullet 5 *GHz* access points, programmed to be on the same network, were utilised. One would be positioned as an assess point on the ground connected to an antenna, whilst the other would be placed in the UAV, connected to an antenna. This system could sustain a 100 *Mbps* connection between the UAV and the ground station up to distances greater than a couple of kilometres, ensuring there would be an adequate connection between devices on the UAV and devices on the ground. A basic Diagram showing the final network setup used on UAV flight days can be seen in Figure 27.

2.5.4. UAV Microcontroller Selection and setup

The network created and manged by in the previous section operated in the 5.8 *GHz* spectrum, which meant it was suitable for use on a UAV as it would not interfere with the UAV's avionics, which operated in the 2.4 *GHz* spectrum. This however meant the Adafruit Huzzah was unsuitable for use on the UAV (as it's Wi-Fi module operates in the 2.4 *GHz* spectrum), hence a Raspberry Pi was used as the microcontroller in the UAV setup. To do this the code used on the Arduino needed to be ported over into a c++ function that could be executed on the Raspberry Pi. Two key libraries were used in this process:

- The WiringPi® library was used to control the GPIO pins and operate the SPI bus on the Raspberry Pi [8].

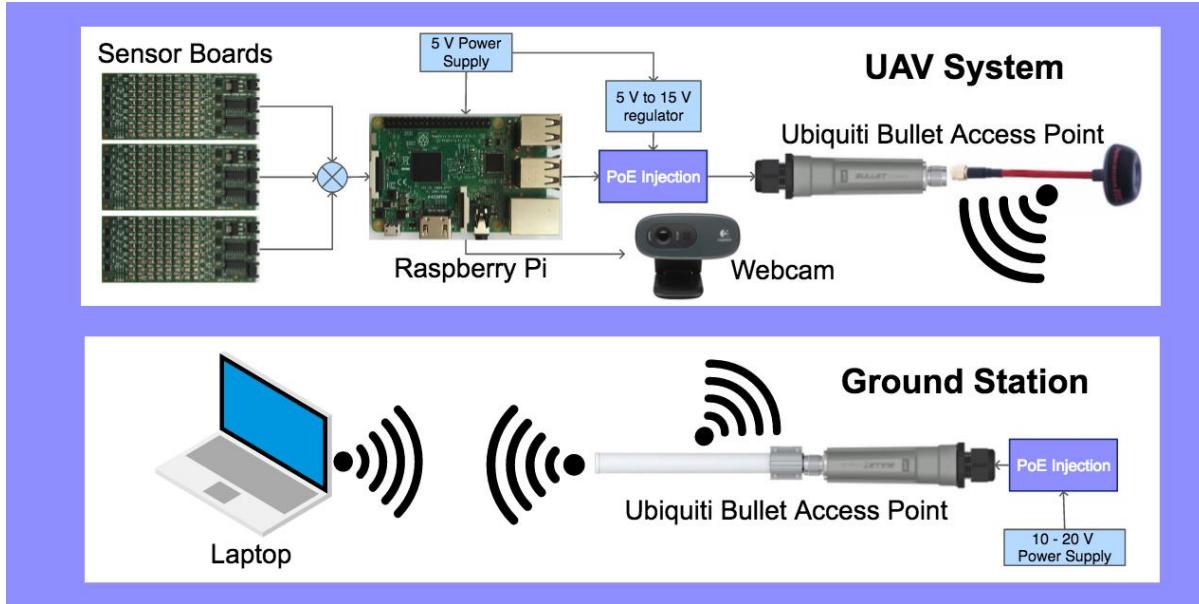


Figure 27 – A basic diagram, showing the network setup used on the flight days to collect pressure sensor data

- The Boost® c++ library was used to control timings and send UDP packets from the Raspberry Pi® [9].

When this process was complete and a sensor board was connected to a Raspberry Pi 3®, the resulting speed improvements for communicating with one sensor board, when compared to that of the Adafruit Huzzah, were striking, as shown in

Table 5.

Table 5 – A comparison of the transmission rate achievable with different numbers of sensors on the same SPI bus, using two different Master Devices

	64 Sensors	128 Sensors	192 Sensors
Adafruit Huzzah	~120 Hz	~90 Hz	~60 Hz
Raspberry Pi 3	~160 Hz	~140 Hz	~120 Hz

2.5.5. Using Multiple Sensor Boards on the Same SPI Bus

On the UAV, the aim was to use a minimum of 2 Sensor Boards (one in each wing), connecting them both to a single Raspberry Pi, which was to be located in the fuselage of the UAV. The original concept was to split the SPI bus close to the Raspberry Pi, distributing it to both sensor boards, however it quickly became apparent there was an issue with this setup as the Raspberry Pi would infrequently ‘lose connection’ with some of the sensors. By probing the system, the issue was found to be on the MISO line of the bus. This is the line used by the slaves to send information to the master device and it seemed the BMP280’s were not powerful enough to drive the entire bus effectively, when it consisted of 2 sensor boards and almost 3 metres of wiring. Two strategies were explored to resolve this issue:

1. Use the second hardware SPI bus on the raspberry Pi to control the second sensor board.
This was an effective solution, and was also a good demonstration of the capabilities of the Raspberry Pi for these situations, however it was not very scalable, i.e. you could not easily add a third board.
2. Use a multiplexer chip, controlled by a Raspberry Pi to select which MISO line to listen to. This meant each BMP280 only had to drive their portion of the MISO line, as illustrated in fig. this system was very scalable so could be easily altered to include additional Sensor boards.

Using the multiplexer chip system, 192 sensors (3 sensor boards) were tested on both the Adafruit Huzzah and Raspberry Pi 3, the Transmission speeds achieved are shown in Table 5.

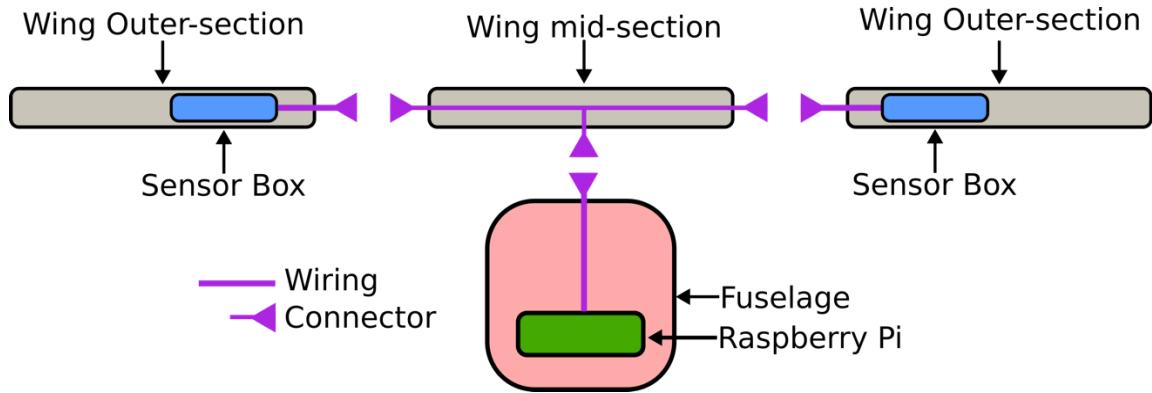


Figure 28 – A simple diagram showing how a multiplexer chip can be used to select different part of the MISO bus

2.6.Final Hardware setup for UAV for the flight tests

2.6.1. Sensor Measurement

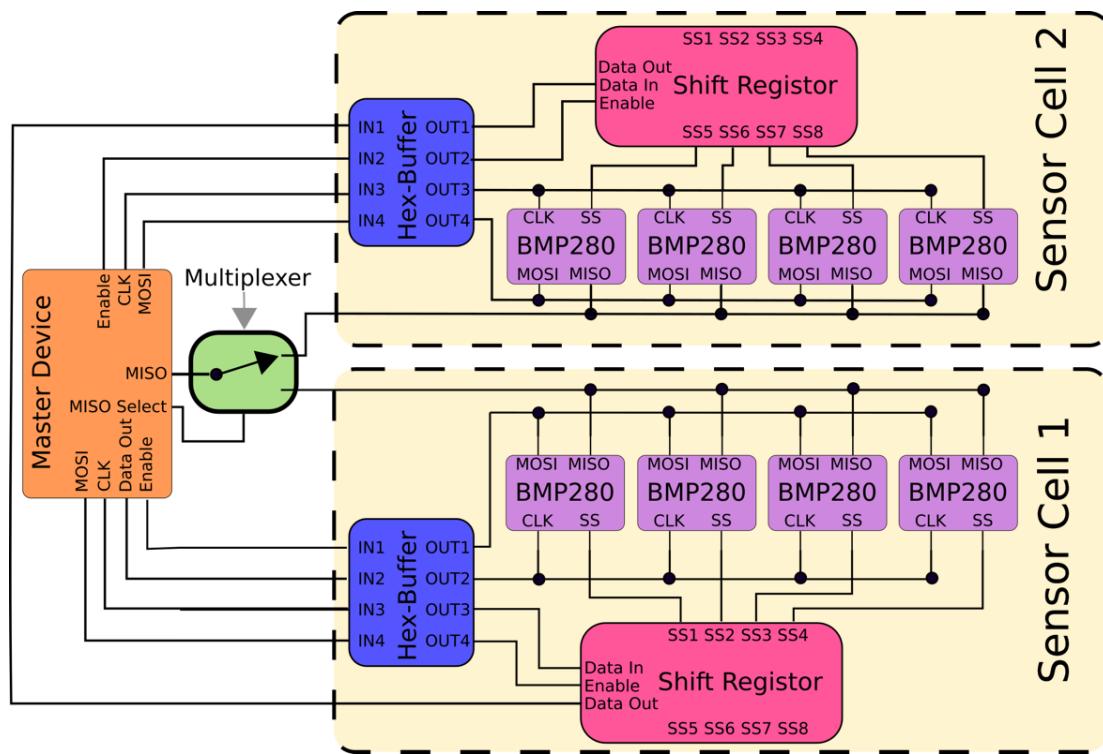


Figure 29 – A diagram showing the modularity of the sensor box wiring in the final UAV design

In the first flight test, two Sensor boards were connected (one in each wing) totalling 128 sensing elements, the system was run at 100 Hz and data was collected on a laptop on the ground. The cabling for the Sensor boxes in the wings was made in 3 sections to allow for the easy disassembly / re-assembly of the UAV, as seen in Figure 29.

For the second flight test an additional board was connected in the nose of the UAV, that was used to measure the 6 channels of a multi-hole probe protruding from the front of the UAV.

2.6.2. Webcam integration

In addition to the pressure measurements a webcam, that was connected to the Raspberry Pi®, was situated in the fuselage of the UAV. Using the MJEG-streamer application [10], a live stream of this webcam was broadcast, which could be accessed by any device connected to the local network.

3. Pressure Distribution Analysis

3.1. Wing Sizing

To correctly size the wings, the initial weight of the aircraft was estimated. An initial maximum take-off weight approximation was made using known data of the components and payload expected to be present at take-off. The assumption made is that there would be two pressure measuring sections and flaps per wing. This was as shown in the table below for a worst-case scenario:

Table 6 – Initial Weight Estimation

Number of sections per wing	2				
Component	Mass per unit (kg/unit)	Count/Total Length	Total Mass (kg)	Weight (N)	
Fuselage	1.25	1	1.25	12.2625	
Wing	0.6	2	1.2	11.772	
Motors and Propellers	0.8	2	1.6	15.696	
BMP280	0.0013	64	0.0832	0.816192	
Raspberry Pi 3	0.023	1	0.023	0.22563	
MPL3115A2	0.0012	1	0.0012	0.011772	
Carbon fibre spar	0.075	2	0.15	1.4715	
Flap control servo	0.041	8	0.328	3.21768	
Battery	0.25	1	0.25	2.4525	
Pixhawk	0.0224	1	0.0224	0.219744	
Wiring	0.05	1	0.05	0.4905	
			4.9578	48.636018	Total

This value of weight was used as an initial approximation for the wing sizing to allow for the chord and span to be selected and initial investigation and estimations to be made. An estimate for the chord and span was important to set up geometric constraints for the wing such that an aerofoil can be selected to accommodate for the pressure sensor board and spars whilst meeting the wing surface area requirements. With an aerofoil selected, simulations could then be run to test the feasibility of any methods as well as identify any risks that could arise during manufacturing.

A higher aspect ratio wing was preferred since any wing tip vortices would have less influence on a larger proportion of the surface. This would result in more predictable pressure readings across the wing, meaning sets of sensors that are further inboard would relate closer to that of an infinite

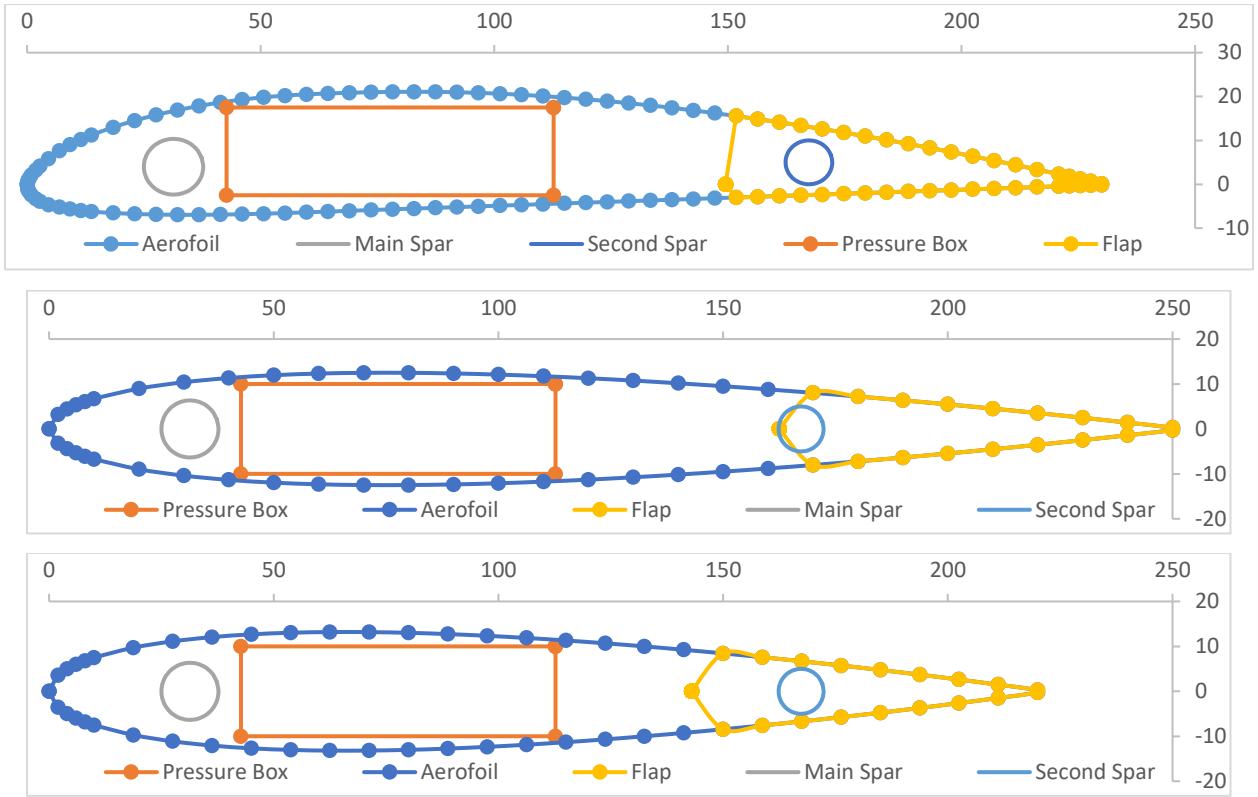
wing and better match the data obtained for a two-dimensional aerofoil in XFOIL. Since the reference UAV's tailplane the could not be easily modified, matching the wingspan to the reference wingspan would also help ensure that the UAV stays within the limits of stability.

3.1.1. Aerofoil Selection

Approximating lift from the pressure distribution over a wing should be applicable to any possible aerofoil but given the time constraints and budget, picking an aerofoil which simplifies the methods and problem would be ideal. Therefore, the aerofoil was selected to best achieve three objectives:

1. Provide a predictable pressure distribution across the chord for pressure sensor readings.
2. Match the aerofoil used for the reference fuselage to avoid structural disparities. Allow space for a spar separation of 136 mm.
3. Provide adequate lift for cruise between flight speeds of 11 m/s and 18 m/s for the given weight.

To obtain a high aspect ratio, the chord must be minimised with the constraints being the required size of the structural spars and the size of internal geometries, such as pressure boxes and servos. The initial pressure box sizes were estimated as 105 mm×55 mm×700 mm, with clearance. When orientated lengthways with the span, the chord cross-section had to house 105 mm × 55 mm.



*Figure 30 – Symmetric NACA aerofoil sizing comparisons for minimum required chord and 35% flap.
(Top to bottom: Clark-Y, NACA0010, NACA0012)*

In addition to the geometric constraints set by the internal geometries and surface area requirements, simulations were run through XFOIL for varying angles of attacks and flap angles to test each aerofoil against the objectives. The pressure coefficients across the aerofoil were recorded with a focus on the shape and peaks within the distribution along the chord as well as the linearity of change in pressure coefficient with angle of attack. A distribution with predictable peaks would be expected to reduce the number of sensors required to model or predict the size and shape of each peak. A one-to-one ratio between angle of attack and pressure coefficient for any point would further simplify calculations, which would be best achieved with a linear correlation between the two.

Initially a Clark-Y aerofoil was investigated due to the relatively flat lower surface. A flat lower surface would be expected to give predictable results with little variation along the chord, requiring only the upper surface to be fully evaluated and reducing algorithm calculations. However, from the two-dimensional aerofoil simulations, all points on the aerofoil showed a non-linear change in angle of attack with change in pressure coefficient. At smaller angles of attack, for a single pressure coefficient value, there were multiple potential angles of attack for any position on the aerofoil.

This uncertainty would reduce the accuracy of any angle of attack and lift approximations as for any given pressure coefficient value; there could be at least three possible angles of attack for a single flap angle.

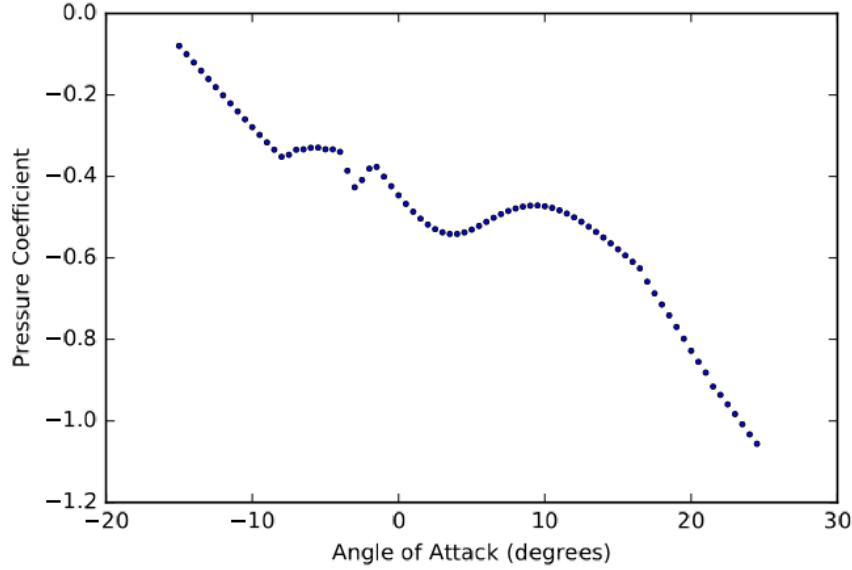


Figure 31 – Clark Y Pressure Coefficient against Angle of Attack at position (0.24, 0.08783)

The other investigated aerofoil shape was a symmetric aerofoil. Symmetry about the chord line implies that the pressure distribution for an angle of attack would be identical on the opposite surface for the same angle of attack in the opposite direction. This greatly simplifies the size of any lookup tables required and thus reduces in-flight calculation times. Four-digit NACA aerofoils are already well established aerofoils which can be easily represented and modified through the equation [11]:

$$T(x) = 10t \left[0.2969 \sqrt{\frac{x}{c}} - 0.126 \frac{x}{c} - 0.3537 \left(\frac{x}{c} \right)^2 + 0.2843 \left(\frac{x}{c} \right)^3 - 0.1015 \left(\frac{x}{c} \right)^4 \right] \quad (1)$$

where x is the position along the chord, $T(x)$ is the thickness at x , t is the maximum thickness and c is the chord length.

The primary choice of thickness to chord ratio was based on the minimum required chord to accommodate both the pressure box and the spars. A fixed fuselage was to be used with the pre-existing spar separation of 136 mm. Although meeting the criteria, the NACA0010 aerofoil was not a viable option as the minimum required chord was much greater than that of the NACA0012.

This meant a lower aspect ratio and would also require a much larger flap to keep the secondary spar, which acted as the pivot for the flap, within the bounds of the flap.

The NACA0012 was chosen as it closely matched the dimensions of the reference wing, simplifying manufacturing, whilst being able to fit the specified sensor box dimensions.

3.1.2. Final Wing Size

UAV		Heading
Mass	4.9578	Input
Weight	48.63602	Output
Wing Sizing		
Stall Speed	11	
Clmax AoA	13	
Clmax	1.42561	
Air density	1.225	
Area	0.460326	
Chord	0.22	Fuselage Width 0.2
Effective Span	2.092393	Total Span 2.292393
Aspect Ratio	9.510878	

Figure 32 – Initial Wing Size Calculations

The minimum aerofoil chord, Clmax aerofoil data from XFOIL and the MTOW estimation were used to approximate the required span. The final span, including fuselage width, was 2.29 m with an aspect ratio of 9.5 and a wing loading of 10.77 kg/m². This also closely matched the span of the reference wing with a 2.2 m wingspan. Overall, this was sufficient for safe flight to test the control system.

3.2. Methods to approximate Lift from Pressure

Multiple methods were available to approximate the lift and angle of attack from the pressure sensor readings across the wing. A method could be dependent on the type of aerofoil used or independent of it.

An aerofoil dependent method would use known data from the aerofoil to approximate the lift and angle of attack by referring to this pre-existing data. This could be geometric data or data obtained from tests or simulations, such as pressure distributions. An aerofoil independent method would focus primarily on the pressure distribution itself and calculate the lift.

The primary goals for any method are to provide an accurate approximation for the lift across the wings and to compute the lift from the pressure quicker than readings from the sensors are measured to prevent bottlenecking the system and to enable for the highest frequency gusts to be damped.

The lift across a wing correlates directly to the angle of attack and gusts can be approximated as an induced angle of attack in addition to the physical angle of attack. Approximating the angle of attack is also another viable method of predicting the lift as the lift can then be estimated from the combination of approximate angle of attack and flap angle.

3.2.1. Method 1: Lookup Table

A lookup table can be used to approximate lift from a set of pressure sensors. For each position on the chord, a pressure coefficient will correspond to a set of flap angle and angle of attack, which in turn corresponds with a lift value. A combination of multiple sensors can be analysed together to produce an approximate average angle of attack and lift.

However, multiple sets of angles of attack and flap angles both correspond to a single pressure value. The number of possible solutions increases further as the resolution of flap angles in the lookup table increases. To rectify this problem for a low resolution of flap angles, data from multiple pressures sensors can be contrasted to choose the best match across all sensors. As the flap angle resolution increases, a very large number of sensors would be required, increasing cost

and computational complexity whilst requiring additional physical space, making this an unfeasible solution for a small UAV.

An alternative solution would be to measure the flap angle as an input from the servos and store a separate lookup table for each flap. The number of files required and stored in memory would increase since multiple smaller lookup tables would be needed to represent each flap angle, but this would also decrease the computational effort in-flight as only a single smaller lookup table will be required for each search. As well as this, knowledge of the flap angle reduces the number of feasible solutions and thus the number of pressure sensors required to consolidate for a best solution.

This method would first interpolate through each lookup table to obtain and approximate angle of attack. From there, a mode of all rounded values of angle of attack would be found and the average of angle from attack for each sensor closest to this mode would be used. This angle of attack, alongside the flap angle, corresponds to lift coefficient value which can be interpolated for in a separate lookup table.

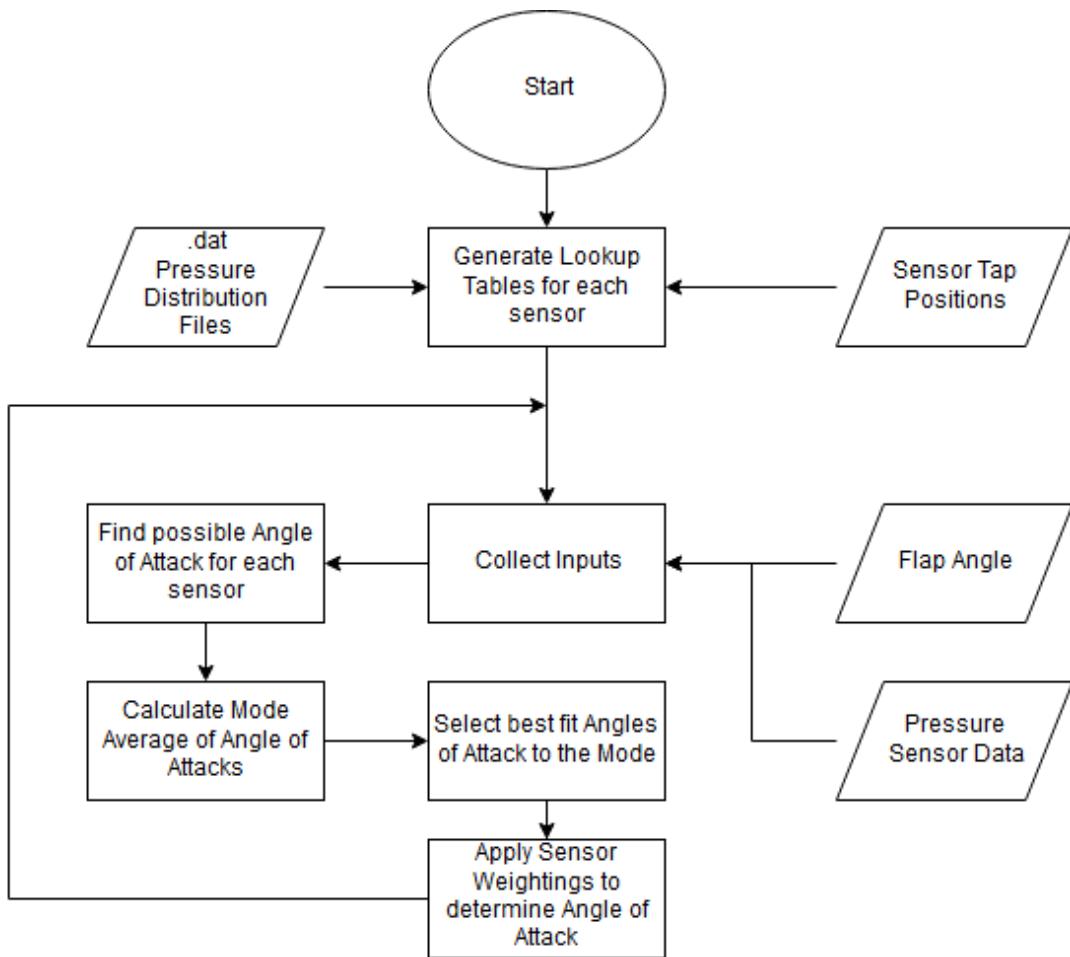


Figure 33 – Flow Chart for Lookup Table Method

3.2.2. Method 2: Integration

Another method of determining lift would be direct integration of a set of pressure readings on both the upper and lower surfaces. Then the lift would be the difference between the total lower pressure and the total upper pressure.

An advantage of using this method is that the lift calculated would be a direct physical representation of the lift across the wing due to the change in pressure, as opposed to an estimate made by the lookup table. This would also provide an aerofoil independent lift value as no previous data would be required to construct a model or lookup table.

This method would require more sensors, particularly around the leading edge, to analyse with enough information for an accurate reading. Another disadvantage would be that for an accurate

reading across the entire wing, pressure readings would also have to be made across the flap. Since the flap is a moving part, tubing from pressure taps to pressure sensors could be stretched, crimped or pulled as the flap moves. The flap is also situated at the trailing edge of the aerofoil, where it is thinnest, and there would be minimal space for tubing to be installed.

A workaround for this would be to measure the pressure between flaps. Although this would not measure the pressure across the flap itself, the pressure on a fixed wing section between flaps would be influenced by the flap angle.

The angle of attack is required to fully resolve the lift force in the correct upwards direction. For small angles of attack this will have minimal effect but for larger angles of attack the method would become increasingly inaccurate. However, using this method in conjunction with the angle of attack approximated from the lookup table method would help to alleviate these inaccuracies by providing an angle to resolve the upwards force from.

Finally, a faulty sensor would heavily skew the readings, especially with a lower sensor density. Even if the faulty sensor was identified and the readings from that sensor were disregarded, it could result in high inaccuracies between the predicted lift value and the actual lift value due to the induced lower sensor density. If the faulty sensor was essential or in a location where the pressure is expected to fluctuate highly, the error in the predicted lift could be high enough to cause dangerous behaviour as the wing attempts to maintain constant lift.

3.2.3. Method 3: Curve Fitting

A third method of approximating lift from the pressure distribution is to use a curve fitting technique. A key position on the surface of the wing is the location of the leading-edge stagnation point in which the local velocity is zero. This can typically be identified as a large peak in pressure near to the leading edge.

Since the stagnation point varies with angle of attack, it can be assumed that the position and magnitude of the stagnation point also corresponds to a single angle of attack. By fitting a curve

around sets of data near the leading edge, the location and magnitude of the peak can be approximated and a value for the angle of attack and thus the lift can be calculated.

As seen in Figure 34, there is a direct correlation between the position of the stagnation point and the angle of attack. For the wing profile considered in this report, the stagnation point is within a range of 0–X mm from the leading edge of the wing.

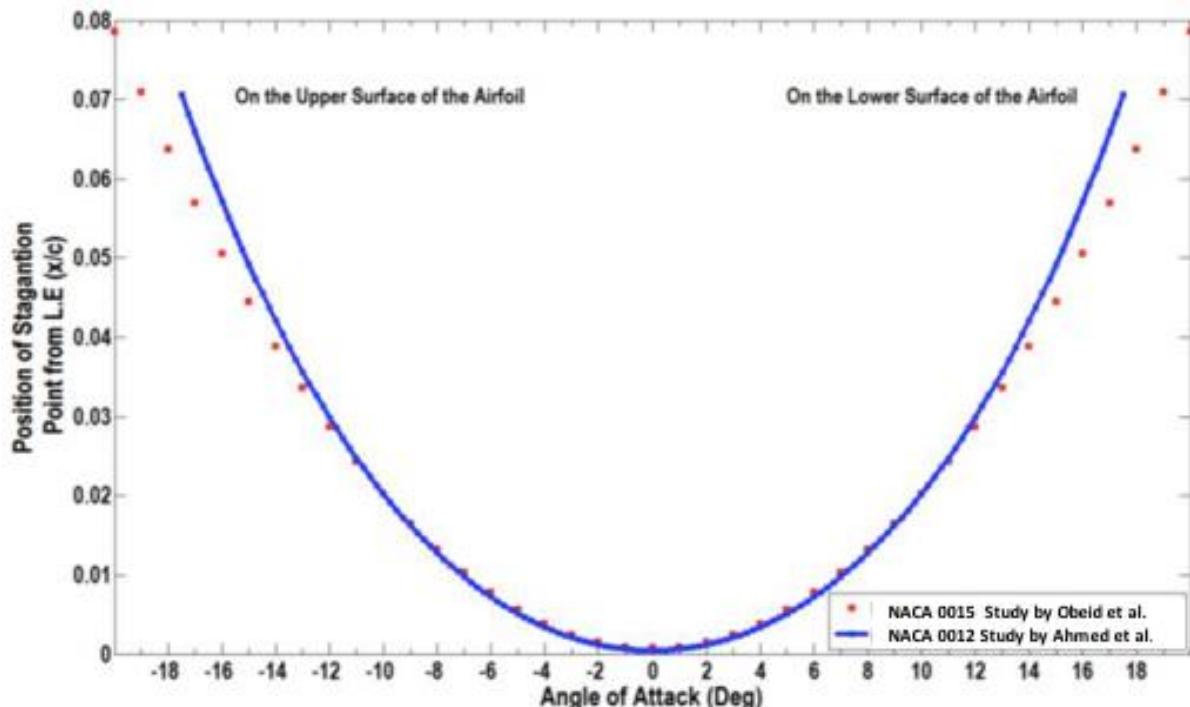


Figure 34 - Variation of stagnation position with Angle of Attack [12] [13].

Pressure sensor tapings cannot be as finely separated as the sampling points in the XFOIL simulations shown in Figure 35. represents a more achievable pressure taping distribution near the leading edge. Thus, accurate estimation of the stagnation point requires the use of curve-fitting algorithms. However, as seen in Figure 37, Figure 38 and Figure 39, no model could accurately estimate the coefficient of pressure near the stagnation point of the graph over a range of angle of attacks.

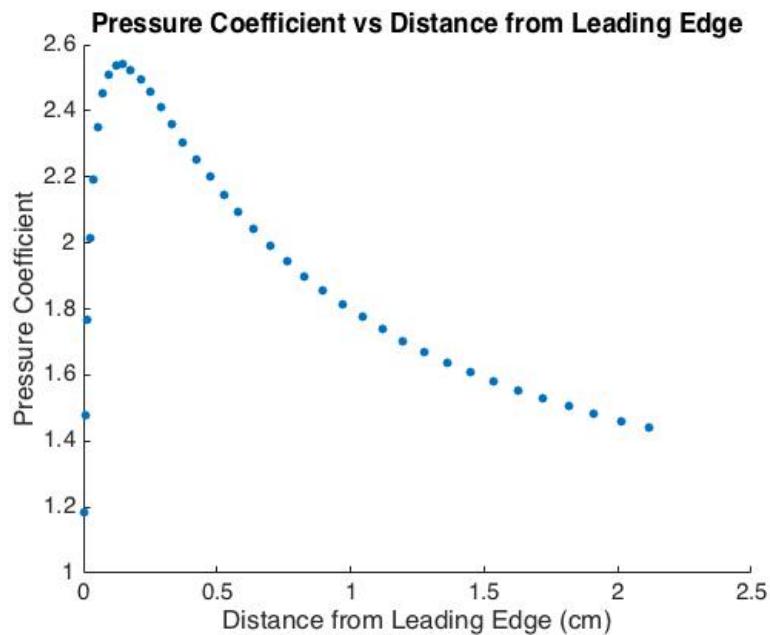


Figure 35 - Pressure Coefficients at sampling points within 2.5 cm of the leading edge. Higher Sampling point density

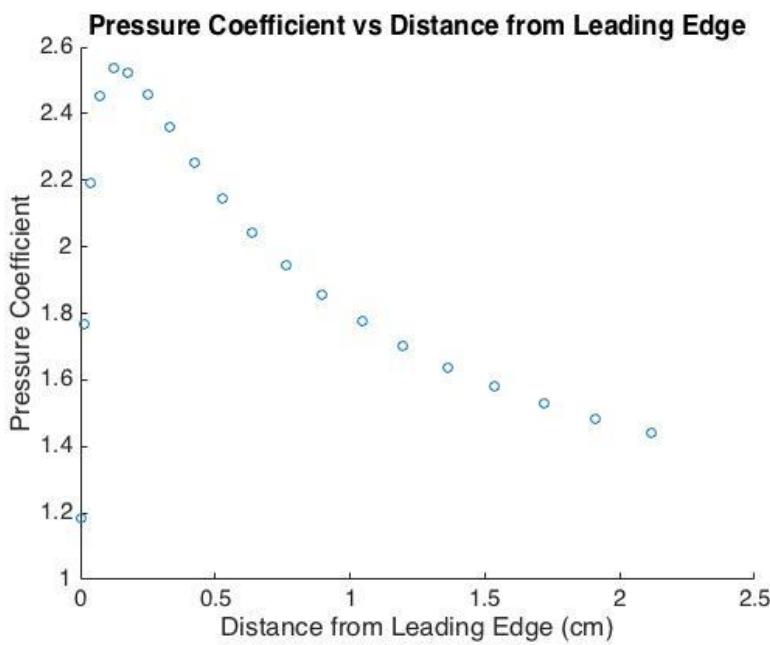


Figure 36 - Pressure Coefficients at sampling points within 2.5 cm of the leading edge. Lower Sampling Point Density.

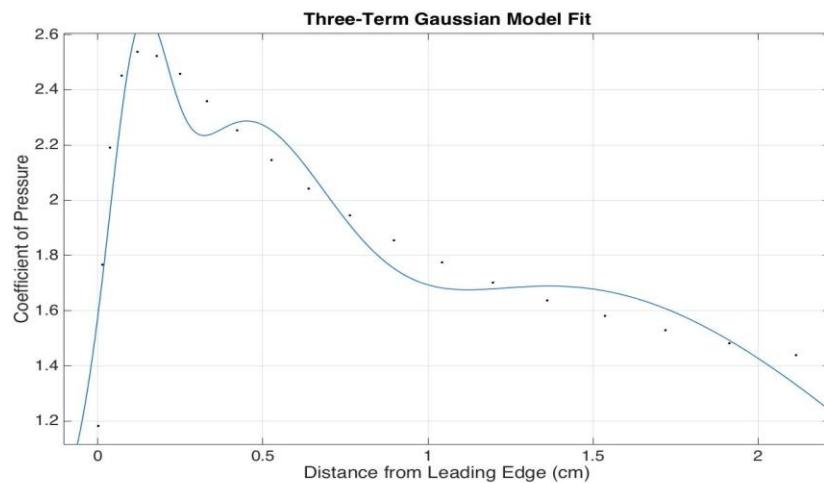


Figure 37 - Fit Obtained by using a three term Gaussian Model.

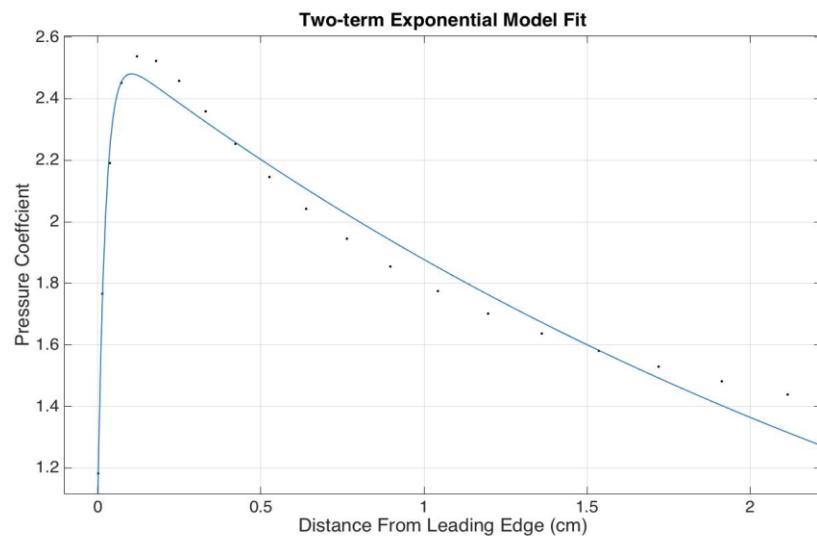


Figure 38 - Fit Obtained by using a two-term exponential model

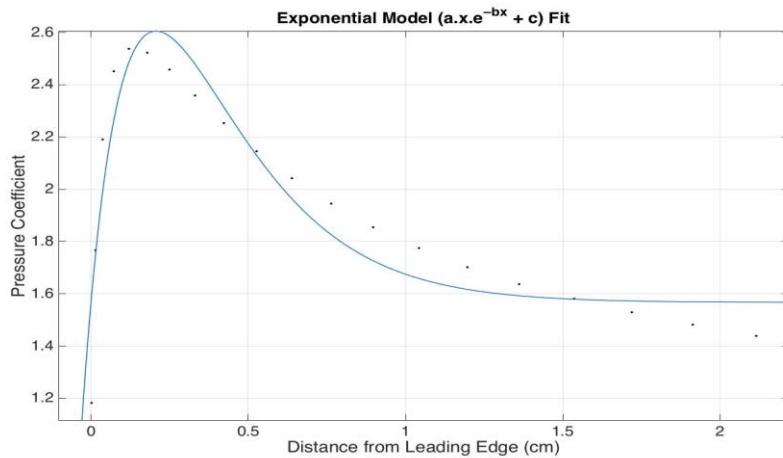


Figure 39 - Fit obtained by using a custom exponential model for fitting.

A smoothing spline fit approach was ideal for this problem as it combined the benefits of curve-fitting and interpolation to allow the spline to accurately fit sparse data while interpolating to accurately estimate the position of the stagnation point as seen in Figure 40.

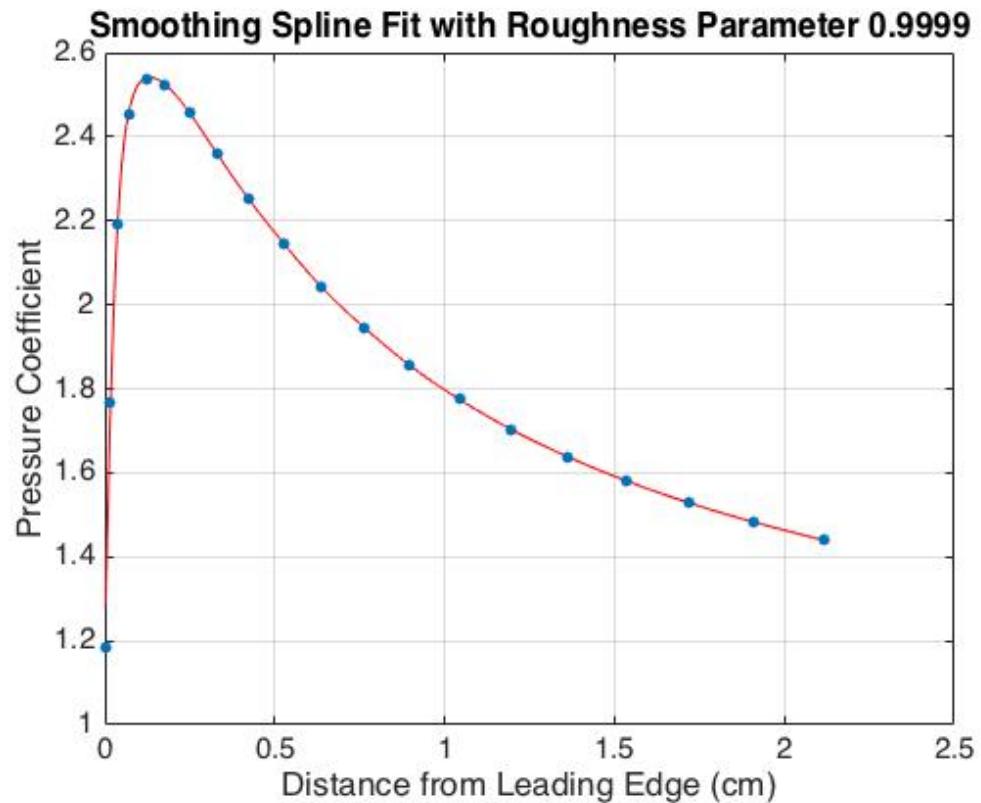


Figure 40 - Fit obtained by using smoothing-spline fit.

3.2.3.1.1. Smoothing with Cubic Splines

Considering the spacing between the pressure tapings and the presence of noise in the pressure sensor data, an interpolating spline will not accurately represent the variation of the pressure readings along the length of the aerofoil. Therefore, to accurately estimate the position of the stagnation point it would be ideal for the fitted spline to depart from the data points [14].

The fitted function (f)

$$y_i = f(x_i) + \varepsilon_i \quad (2)$$

where:

y_i represents the pressure reading at the position x_i from the leading edge of the aerofoil. ε_i is a sequence of independently distributed random variables with variance of σ_i^2 (standard deviation of i reading). The smoothing spline functions S_i is constructed such that it minimises the value of L .

$$L = \lambda \sum_{i=0}^n \left(\frac{y_i - S_i}{\sigma_i} \right)^2 + (1 - \lambda) \int_{x_0}^{x_n} \{S''(x)\}^2 dx \quad (3)$$

where $\lambda \in [0,1]$ is the parameter which determines the extent to which the spline S fits the data. A lower value of λ places the emphasis on a smoother spline.

The cubic spline function in parametric terms is expressed as:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad (4)$$

first derivative S'_i :

$$S'_i(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \quad (5)$$

second derivative S''_i :

$$S''_i(x) = 6a_i(x - x_i) + 2b_i \quad (6)$$

Thus, considering the piecewise nature of spline function, the integral term on the RHS of equation (3), can be written as:

$$\int_{x_0}^{x_n} \{S''(x)\}^2 dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} \{S''(x)\}^2 dx \quad (7)$$

From equation (5), the second derivative of S is a linear function which varies from $2b_i$ at x_i to $2b_{i+1}$ at x_{i+1} .

$$\int_{x_i}^{x_{i+1}} \{S''(x)\}^2 dx = \frac{4h_i}{3} (b_i^2 + b_i b_{i+1} + b_{i+1}^2) \quad (8)$$

$h_i = x_{i+1} - x_i$. Substituting equation (7) and equation (8) in equation (3):

$$L = \lambda \sum_{i=0}^n \left(\frac{y_i - d_i}{\sigma_i} \right)^2 + (1 - \lambda) \sum_{i=0}^{n-1} \frac{4h_i}{3} (b_i^2 + b_i b_{i+1} + b_{i+1}^2) \quad (9)$$

Conditions for constructing the smoothing spline:

1. The spline is a natural spline and thus $S''(x_0) = 2b_0 = 0$ and $S''(x_n) = 2b_n = 0$.
2. The piecewise function S_{i-1} and S_i have the same value at x_i .
Thus, from equation (4):

$$a_{i-1} h_{i-1}^3 + b_{i-1} h_{i-1}^2 + c_{i-1} h_{i-1} + d_{i-1} = d_i \quad (10)$$

3. From the first order continuity – $S'_{i-1}(x_i) = S'_i(x_i)$:

$$3a_{i-1} h_{i-1}^2 + 2b_{i-1} h_{i-1} + c_{i-1} = c_i \quad (11)$$

4. From the second order continuity – $S''_{i-1}(x_i) = S''_i(x_i)$:

$$6a_{i-1} h_{i-1} + 2b_{i-1} = 2b_i \quad (12)$$

From equations (10), (11) and (12), the value of a_i and c_i can be represented in terms of b_i and d_i .

Using the same set of equations, Pollock represented the relationship between b_i and d_i in matrix form:

$$\begin{bmatrix} p_1 & h_1 & 0 & \cdots & 0 & 0 \\ h_1 & p_2 & h_2 & \cdots & 0 & 0 \\ 0 & h_2 & p_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p_{n-2} & h_{n-2} \\ 0 & 0 & 0 & \cdots & h_{n-2} & p_{n-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix} = \begin{bmatrix} r_0 & f_1 & r_1 & 0 & \cdots & 0 & 0 \\ 0 & r_1 & f_2 & r_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & r_{n-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & f_{n-1} & r_{n-1} \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \quad (13)$$

where:

$$p_i = 2(h_{i-1} + h_i), \quad r_i = \frac{3}{h_i}, \quad f_i = -\left(\frac{3}{h_{i-1}} + \frac{3}{h_i}\right) = -(r_{i-1} + r_i)$$

Equation (13) written in matrix notation

$$Rb = Q'd \quad (14)$$

And rewriting the objective equation (9) in terms of b and equating the partial derivative with respect to b to 0 ultimately gives:

$$(\mu Q'\Sigma Q + R)b = Q'y \quad (15)$$

where:

$$\mu = \frac{2(1-\lambda)}{3\lambda} \text{ and } \Sigma \text{ is a diagonal matrix with the diagonal elements } = \{\sigma_0, \sigma_1, \dots, \sigma_n\}$$

Thus, the vector b is estimated and used to calculate the rest of the parameters. $(\mu Q'\Sigma Q + R)$ is a symmetric matrix with only 5 diagonal bands. Using an LDL matrix decomposition method based on pseudocode provided by Pollock [15], The vector b can be calculated using the matlab script ‘calcB.m’ shown below in Figure 41 (See Appendix F for implementation).

```

function [b] = calcB(n,u,v,w,q)
%% Inputs
% n: number of samples
% u: vector representing the main diagonal
% v: vector representing the second diagonal symmetric about u
% w: vector representing the third diagonal symmetric about u
% q: y values (RHS of equation)
%% LDL factorization of diagonals
v(2) = v(2)/u(2);
w(2) = w(2)/u(2);

for j = 3:1:n-1
    u(j) = u(j) - (u(j-2) * w(j-2) * w(j-2)) - (u(j-1) * v(j-1) * v(j-1));
    v(j) = (v(j) - (u(j-1) * v(j-1) * w(j-1)))/u(j);
    w(j) = w(j)/u(j);
end
u(n) = u(n) - (u(n-2) * w(n-2) * w(n-2)) - (u(n-1) * v(n-1) * v(n-1));

%% Forward substitution
%relevant values between 2 to n. position 1 is padding
for j = 3:1:n
    q(j) = q(j) - (v(j-1) * q(j-1)) - (w(j-2)*q(j-2));
end
for j = 2:1:n
    q(j) = q(j)/u(j);
end

%% Back substitution
% At n row : b(n) = q(n)
% position 1 is padding.
% hence b is assinged from n to 2. b(size) and b(1) = 0
for j = n-1:-1:2
    q(j) = q(j) - (v(j) * q(j+1)) - (w(j)*q(j+2));
end
b = q;
end

```

Figure 41 - Matlab script calcB used to calculate the vector b.

3.3.Feedback

With the approximate lift established, the UAV must oppose any deviations in lift from the desired lift value. For each combination of angle of attack and lift, there is a corresponding pressure distribution and overall lift value. Since the angle of attack cannot be directly influenced mechanically, the flaps must deflect to change the pressure distribution and the lift to the desired value.

A lookup table was used to store the angle of attack, flap and corresponding lift values. Then, a search was conducted through this lookup table using the estimated angle of attack and required lift value to determine the required flap angle for to maintain the desired lift. The difference between the required flap angle and the currently measured flap angle resulted in the direction and magnitude in which the flap had to move.

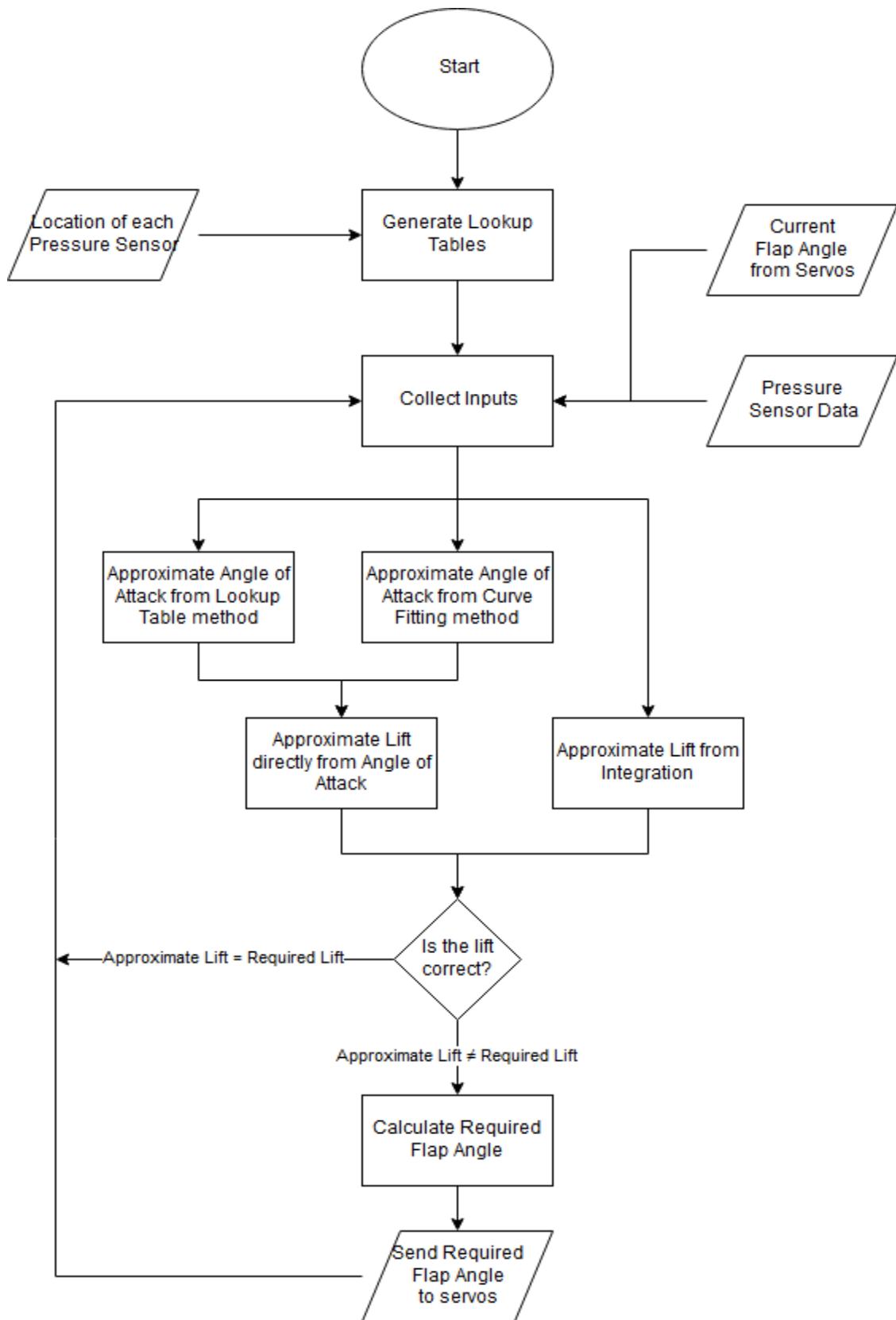


Figure 42 – Flowchart of Control Loop using Pressure Sensors and Flap Angle as inputs.

3.4. Pressure Sensor Quantity and Locations

To accurately approximate the lift coefficient from pressure sensors, the quantity and location had to be determined. To take advantage of the symmetrical nature of the aerofoil, pressure taps were positioned on the upper and lower side of the aerofoil symmetric about the chord line.

3.4.1. Chordwise- location

The first constraints for this problem were set by the geometric properties of the aircraft and the size of the pressure taps. The pressure taps had to fit within the physical boundaries of the aerofoil and any adjacent sensors could not be so close that they would overlap or cause local interference with adjacent taps. As mentioned previously, considerations had to be made for the size, motion and location of the flap as this could cause further interference with the readings or damage to the tap-sensor interface.

A secondary object to be considered was the number of taps available. More taps would require more pressure sensors which in turn would require more computational power and result in a potentially slower read rate or less accuracy for the same read rate. This would further limit the maximum gust frequency that could be detected and corrected for or the overall accuracy of the system.

A key consideration to be made was the pressure tap density across the aerofoil. As shown below in Figure 43, there is a large peak near the leading edge which translates across the chord with changing angle of attack. Due to the small width of the peak and the high amplitude, a higher density of pressure sensors would be required near the leading edge to correctly capture the characteristics of it. Whereas, between 20% chord and the flap hinge, the shape is flatter and easier to capture with fewer taps. This is particularly important when determining the lift from integration, as the error increases as the shape deviates from the actual solution.

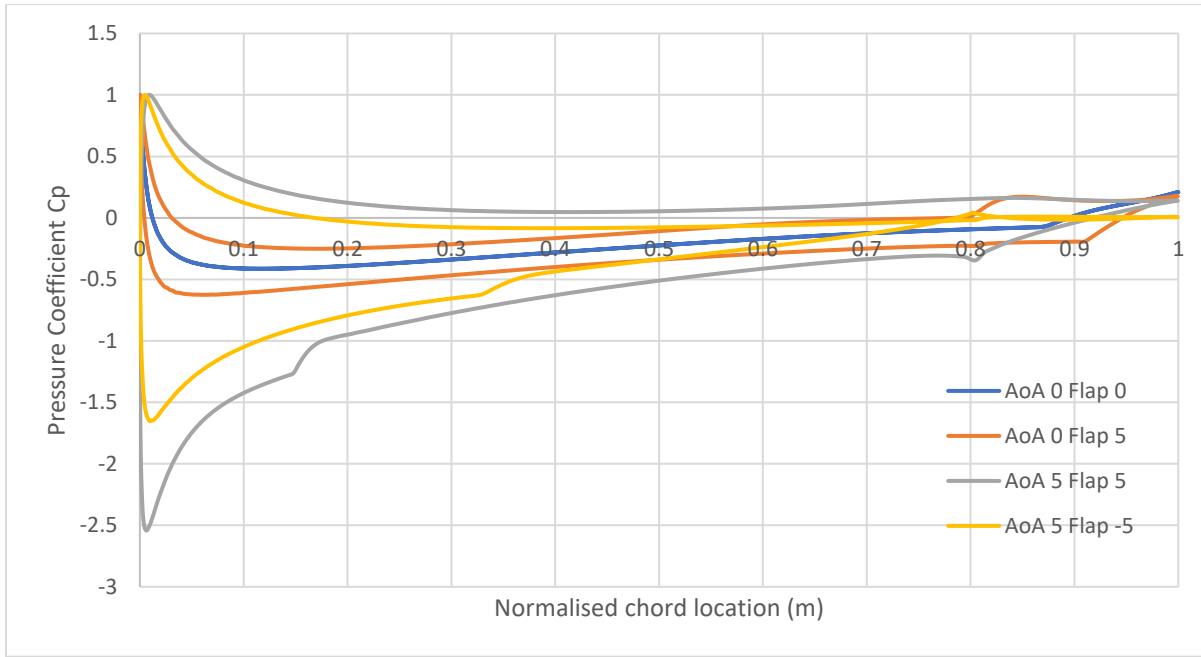


Figure 43 – Pressure distribution across NACA0012 aerofoil for $Re = 271810$ and flap = 20% chord

To approximate the lift using a lookup table, predictability of the relationship between sensor readings and overall lift is required. The first key component was the average number of solutions for any given pressure coefficient value. The lower the number of solutions, the less computational time is required to identify and select the best solution. In addition to this, there is a lower chance of an incorrect solution being selected due to a reduced number of possible solutions.

The second key component was the change in the pressure coefficient for each change in angle of attack at a given flap angle. A higher difference between pressure coefficients means a higher range of accuracy for any changes in pressure across the wing. Any noise and error from the sensors themselves would also have a less significant impact on the final result for locations on the wing where there is a greater pressure difference between angles of attack as their magnitudes would be comparatively smaller.

XFOIL was used to simulate the pressure data across a NACA0012 two-dimensional aerofoil section between flap angles of -20 *degrees* and 20 *degrees* and angle of attacks of -15 *degrees* and 15 *degrees* with a step of 1 *degrees* for both. The average change in pressure coefficient with angle of attack was averaged across each flap angle and compared with the average number of solutions of angle of attack for all pressure coefficients across the range.

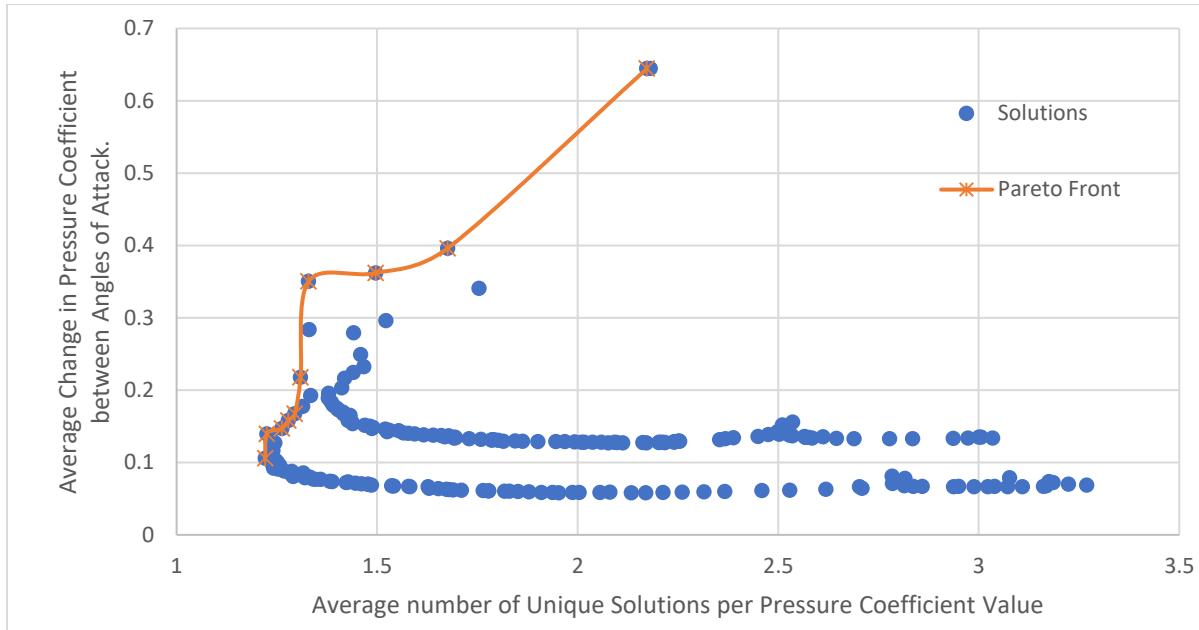


Figure 44 – Comparison of Change in Pressure Coefficient and Average Number of Solutions across a set of angles of attack and flap angles for each coordinate on the aerofoil. Ideal solution lies at top-left.

Table 7 – Pareto Optimal Solutions for Pressure Coefficient Difference against Average Number of Solutions

x/c	Average Number Of Solutions	Average Difference between pressure coefficients
0	2.1725	0.644382758
0.01	1.675833333	0.396160679
0.02	1.496666667	0.361984551
0.03	1.329166667	0.350387724
0.04	1.309166667	0.217702901
0.07	1.295	0.167463740
0.08	1.278333333	0.157978312
0.09	1.2625	0.146883104
0.1	1.225	0.139547307
0.17	1.220833333	0.105903674

Positions around the leading edge had a higher number of possible solutions but with a larger average difference between readings. Further back from the leading edge, the average number of solutions reduced as readings became more linear. However, the change in angle of attack with pressure coefficient also reduced. The Pareto solutions were within 3% and 17% of the chord. The worst solutions were those across the flap.

The lowest average number of solutions was at 17% chord, whereas it was expected to decrease further along the chord until close to the flap hinge, where flap movement should have a greater influence. Upon further investigation, the primary source of this disparity was due small separation bubbles moving across the surface with changing angles of attack, reducing the pressure in local areas. This separation bubble had a greater influence further down the chord due to the smaller change in pressure coefficient with angle of attack, resulting in more angle of attack solutions for pressure coefficients lying in the range induced by separation.

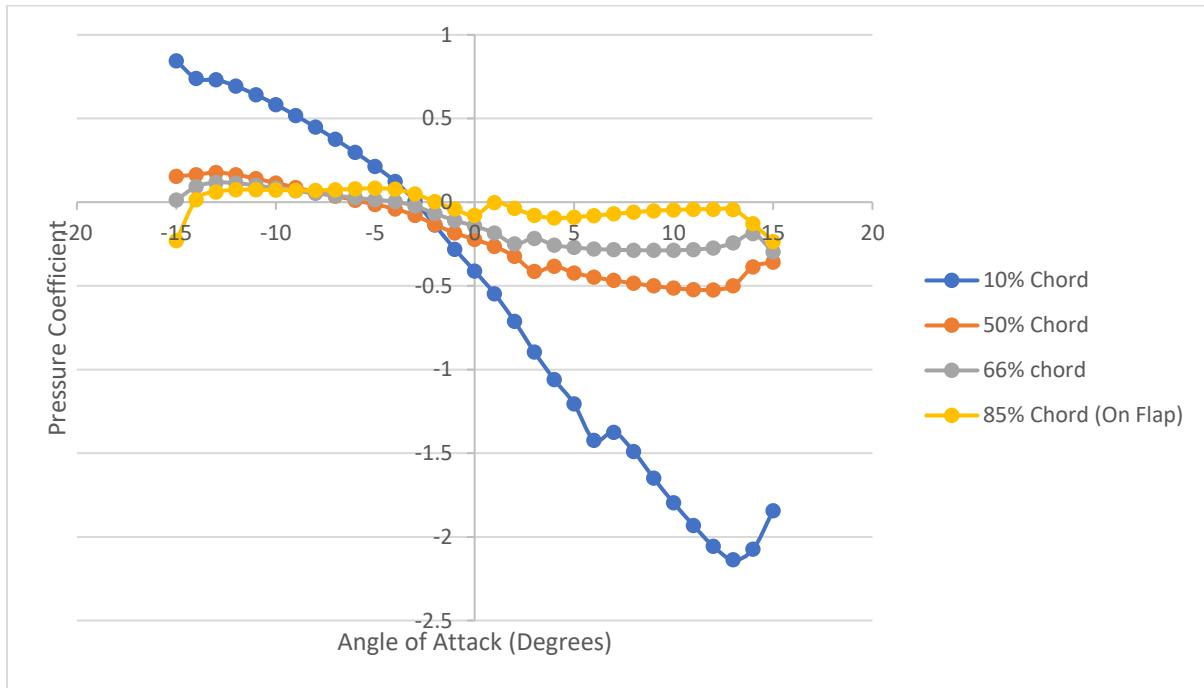


Figure 45 – Change in Angle of Attack with Pressure Coefficient for Various points across NACA0012. Small pressure coefficient spikes can be seen between 0 and 5 degrees for 10%, 50% and 66% chord and between 5 and 10 degrees for 10% chord.

Excluding the effects due to the separation bubble resulted in a lower average number of solutions, as originally expected. However, the presence of the separation bubble implies that multiple positions between 20% chord and the flap hinge should be evaluated to identify and adjust for these changes in local pressure due to separation.

From the investigation, the best solution would be to have a high density of sensors on and near the leading edge to take advantage of the large change in pressure coefficient with angle of attack and minimise the effects of noise and error from the pressure sensors. In addition to this, a spread of sensors between 10% chord and the flap hinge should also be used to take advantage of the linearity between pressure coefficient and angle of attack to adjust and validate the angle of attack

approximated by the leading-edge sensors. Readings across the flap were the furthest solutions from optimal and should be avoided or weighted very low.

To determine the number of sensors required for the integration method, the error for the approximated Cl across a different range of sensors was compared to the Cl provided by XFOIL, which is the equivalent of 504 integration points per surface. Due to the peaks at the leading edge and flap: the areas around the leading edge, mid-section of the aerofoil and flap were investigated separately. The leading edge was represented as 0% to 10% of the chord, the mid-section as 10% to 80% of the chord and the flap as 80% to 100% of the chord.

To evaluate each possible position directly alongside the number of sensors would be an unfeasible process due to the large number of sensors and their locations and the time that it would take to evaluate. The total number of combinations would be:

$$\text{Combinations} = \sum_{n=1}^{504} \frac{504!}{504!(504 - n)!}$$

Where n is the number of sensors for evaluation.

To adjust for this, the sensors were positioned equidistant to each other between the upper and lower bounds for each segment. From there, exact positions can be optimised. Then all flap angle and angle of attack combinations were tested and the average error was recorded.

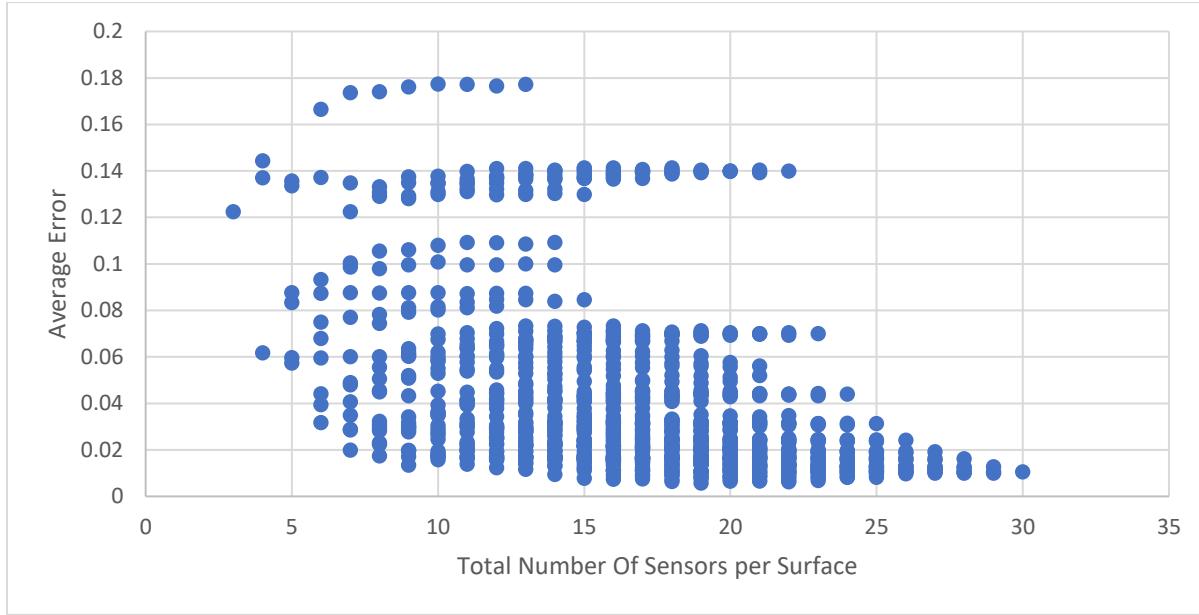


Figure 46 – Average Error between approximated case and ideal case against Total Number of Sensors.

As expected, as the number of sensors increased near to the leading edge, the error decreased up until a minimum error for 10 sensors. The lowest error for the cases recorded was for 19 total sensors, showing that the physical location of the individual sensors is equally as important as the density of sensors.

The initial solutions for best sensor positions for minimum error were at:

x/c	0.00	0.01	0.02	0.03	0.04	0.06	0.07	0.08	0.09	0.10
x	0.00	2.44	4.89	7.33	9.78	12.22	14.67	17.11	19.56	22.00
x/c	0.20	0.45	0.70	0.80	0.84	0.88	0.92	0.96	0.99	
x	44.00	99.00	154.00	176.00	184.80	193.60	202.40	211.20	220.00	

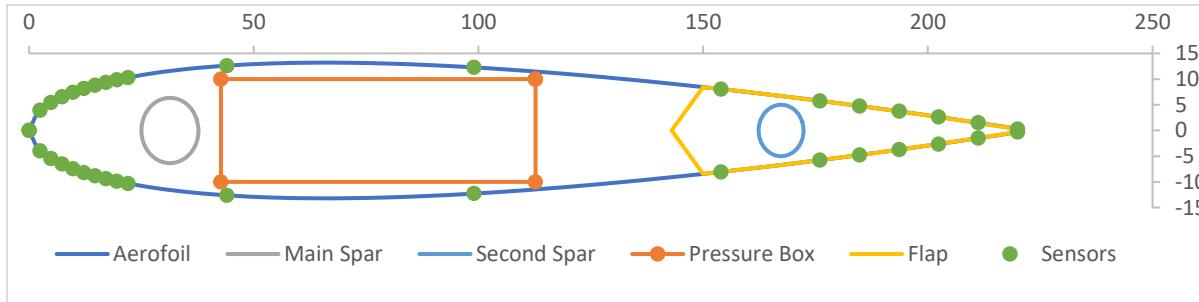


Figure 47 – Initial Pressure Tap Positions for Optimal Integration across a 2D NACA0012 aerofoil

Although this is the ideal solution, the bounds of the geometric and manufacturing constraints made it impractical. Pressure taps are unable to be positioned near the trailing edge due to slender geometry. This limits the space available for both the taps themselves and the tubing which

connects them to each pressure sensor. As well as this, the rear spar occupies a significant amount of space, leaving very little space for this tubing to pass by. The solution would be a compromise to fit the highest number of taps in the space provided without compromising the structure of the flap.

Another problem was that the density of sensors at the leading edge was too high. As with the trailing edge, there must be sufficient space to pass tubing from the taps to the sensors past the spar. For taps of 1.5 mm diameter, there was less than 1 mm between each tap meaning any manufacturing defects would have a much greater significance and an increased chance of interaction between two different taps. The only remedy would be to reduce the density of sensors, with the priority of retaining the taps in positions with the most significant contribution.

The modifications for the integration method sensors were made to fit within geometric constraints. In addition to this, the solutions for the optimum position and number of sensors for each method were combined. This resulted in tap positions of:

x/c	0.00	0.02	0.03	0.05	0.06	0.08	0.10	0.12	0.15	0.17
x	0.00	4.40	7.50	11.00	14.00	18.00	22.00	26.00	32.00	38.00
x/c	0.20	0.22	0.25	0.27	0.30	0.35	0.40	0.45	0.50	0.55
x	44.00	49.00	55.00	60.00	66.00	77.00	88.00	98.00	110.00	120.00
x/c	0.60	0.67	0.70	0.73	0.78	0.80	0.83			
x	132.00	147.00	154.00	160.00	172.00	177.00	182.00			

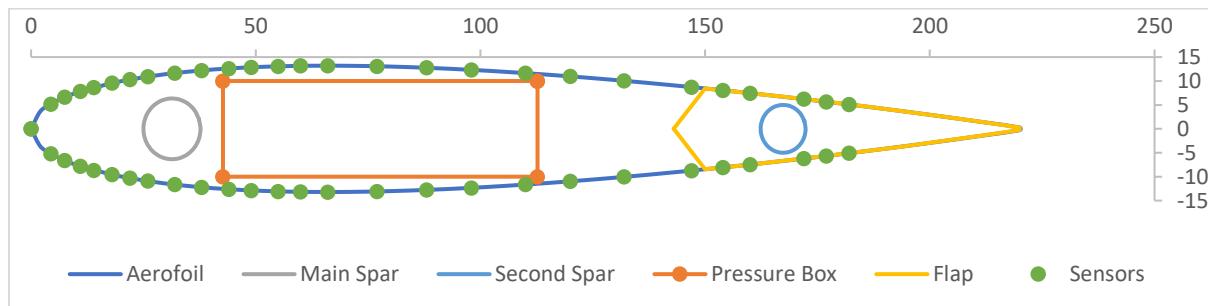


Figure 48 – Final Pressure Tap Positions

4. Calculating Lift from Pressure

4.1. Method 1 – Lookup Table

4.1.1. Generating the Lookup Table

To generate the data for a table, the desired set of flap angles and angles of attack were iterated through and simulated in XFOIL. For each set of flap angles and angle of attacks, the angle of attack was saved alongside a table of x-coordinates, y-coordinates and pressure coefficient in a file in the form of:

<Angle index> <Flap index> <Aerofoil Name>alpha<angle of attack>flap<flap angle>.dat

The angle index and flap indexes are unique indexes provided for easy manual file sorting and analysis when viewing the files from a file browser where the indexes are:

$$index_{\alpha} = \frac{(\alpha - \alpha_{\min})}{\alpha_{step}}, index_{\gamma} = \frac{(\gamma - \gamma_{\min})}{\gamma_{step}} \quad (16)$$

Using these files, a lookup table was generated for each pressure sensor corresponding to a position normalised with the chord on the aerofoil. Every file was iterated through and the angle of attack was stored in a one-dimensional array alongside an array with matching pressure coefficients for the chosen position.

Since the flap angle will be given as an input during runtime, sets of data were grouped by flap angle as opposed to angle of attack. By grouping the data this way, only the adjacent indexes corresponding to the measured flap angle need to be accessed, providing visual clarity and allowing for a simple incremental iteration. For example: the indexes would be as follows:

The lookup data is stored in a one-dimensional array. If organised by angle of attack, the index of any angle of attack and flap angle combination would be:

$$index_{\alpha,\gamma} = \frac{(\alpha - \alpha_{\min})}{\alpha_{step}} \frac{(\gamma_{\max} - \gamma_{\min})}{\gamma_{step}} + \frac{(\gamma - \gamma_{\min})}{\gamma_{step}} \quad (17)$$

To compare all angles of attack for a fixed gamma, it would require a step of:

$$step = \frac{(\gamma - \gamma_{\min})}{\gamma_{step}} \quad (18)$$

Whereas, by organising by flap angle the index of any angle of attack and flap angle combination would be:

$$index_{\gamma,\alpha} = \frac{(\gamma - \gamma_{min})}{\gamma_{step}} \frac{(\alpha_{max} - \alpha_{min})}{\alpha_{step}} + \frac{(\alpha - \alpha_{min})}{\alpha_{step}} \quad (19)$$

In this case, to iterate across all angles of attack for a given flap angle, the step would only be a simple increment as there is a linear increase for each angle of attack increase.

4.1.2. Estimating Angle of Attack from Sensor Readings

Since there can be multiple solutions for any given sensor, a way of consolidating between all sensors is required. One option would be to check across every available combination of angles of attack and choosing the combination with the smallest variation. However, this means that total number of combinations is:

$$\textbf{Total Combinations} = s_1^{i_1} \times s_2^{i_2} \times \dots \times s_n^{i_n} \quad (20)$$

For a worst-case scenario of all sensors having 3 feasible solutions, this results in $53^3 = 148877$ possible combinations. This would be computationally demanding to analyse. The alternative solution used was to calculate the mode of all recorded solutions, then selecting the combination with values closest to the mode. This greatly reduced the complexity as each set of solutions from each sensor will only need to be accessed and evaluated twice: once to calculate the mode and again to find the best fit to the mode.

To calculate the mode, all angles of attacks were rounded to an integer value first. For the most linear sensors at the mid-chord, the only situations in which it would be expected to have multiple angle of attack predictions would be near separation and near stall. Therefore, even with local pressure jumps around certain positions, the correct approximate angle of attack should still be the value closest to the mode, making this a viable option.

During stall, there is a large pressure drop across the whole aerofoil. This means any sensors which include a set of angles of attack that lie in the pressure region between the maximum pressure and the reduced pressure due to stall will also read the stall angle as a solution, as well as the actual angle of attack. Since this large reduction in pressure affects most positions across the surface, the lookup table search will be likely to also return a false stall reading across a large range of angles of attack. Therefore, without a secondary method of identifying stall, angles close to and above stall must be rejected.

Overall, the average duration of this method was 0.0015 *seconds*, or 750 Hz, to compute across 10 iterations. This is a factor of 6.25 times quicker than the maximum read rate of the pressure sensors making this a viable method in terms of computational time.

4.1.3. Individual Sensor Weightings

A carefully chosen weighting for all sensors would further increase the accuracy of the predicted angle of attack, especially in the presence of noise and fluctuations. From the initial, individual position testing, it was expected that positions closer to the mid-chord would be more consistent than those at the leading and trailing edges and thus require a higher weighting.

Sensors on the flap had the worst properties of all sensors and would be expected to have very low weightings in comparison. Sensors near the leading edge would be expected to have a lower weighting than those on the mid-chord due to the higher number of possible solutions but high enough to still have a significant contribution.

All angles of attack for all flap angles were iterated through and the error difference between the actual angle of attack and the predicted angle of attack was the objective function where:

$$\epsilon = \text{Predicted } \alpha - \text{Actual } \alpha \quad (21)$$

And:

$$\text{Predicted } \alpha = (\sum_{i=1}^n w_i)^{-1} \sum_{i=1}^n w_i \alpha_i \quad (22)$$

where ϵ is the error difference, i is the sensor index, n is the number of sensors, α is angle of attack, α_i is best fit angle of attack for a sensor and w_i is the weighting factor for a sensor. The weighting factors for each sensor were varied to minimise the average error difference across all flap angles and angles of attack.

Since there were many variables with many possible weightings and an additional random element due to noise, a simple local optimisation method would not be possible. An initial optimisation approximation was taken through a genetic optimiser with a population size of 40 and 1000 total iterations with bounds of 0 and 1 for the weighting of each position. From there a local Sequential Least Squares Programming (SLSQP) optimisation built into the Python SciPy library was used.

Noise and fluctuations were simulated by applying a random distribution with magnitudes of up to plus and minus the maximum absolute error of the sensors. This distribution used a consistent seed such that the same random distribution was applied to each set of weights for consistency when optimising. The random distribution was then summed with the exact pressure coefficients corresponding to each angle of attack and flap angle combination from XFOIL. Without this random distribution, the overall error would be zero as all positions would measure the ideal pressure coefficient for a given angle of attack and after consolidation the approximated angle of attack would be an exact match.

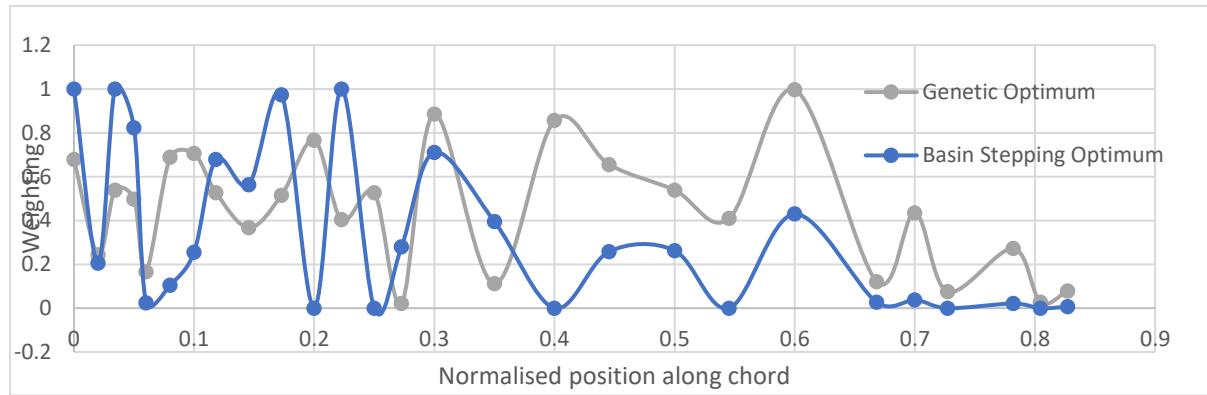


Figure 49 – Weighting of sensor taps across chord.

The best result from the genetic algorithm global search is as shown above with a minimum average error of 0.187. Further to this, a basin stepping minimisation algorithm using SLSQP gave a result of a minimum average of 0.151. As expected, as the weighting of the flaps decreased, the error difference also decreased.

However, the sensors on and near to the leading edge had an unexpectedly higher weighting than that at the mid-chord. The reason for this was due to the consolidation of the other sensors in conjunction with the large change in angle of attack with pressure difference. The large change in angle of attack with pressure difference made the effects of the random contribution to the pressure coefficients less significant. When all sensors were consolidated to find the mode, the correct angle of attack could be found, eliminating any errors caused by the non-linearity of the leading-edge positions.

Overall, it is beneficial to use a weighting distribution across the sensors. To optimise this distribution is a time-consuming process and would have to be repeated for any changes made to the system, such as the addition of more sensors or a different aerofoil.

Since the weighting distribution did not follow an obvious pattern, it's assumed that it would be required to calculate and optimise the distribution for each of these changes. With knowledge of the distribution calculated above, the optimisation process can be sped up by setting an initial distribution with a high weighting across the leading edge which decreases towards zero weighting across the flap.

4.1.4. Upper and Lower Surface Weighted Average

An addition factor to consider is the influence of the flap on the pressure upstream of it. The flap has a greater influence on the surface of the aerofoil in the direction that it is deflected in, compared to the side opposite this direction. This could cause more inaccuracies in the predicted angle of attack and lift. Evaluating the upper and lower surfaces independently and taking a weight average of the two with some weighting factor could further reduce any error due to flap deflection.

To test for this the flap was deflected in the direction towards the lower surface only and the angles of attack were predicted on the upper and lower surfaces with a weighted average of the two being taken. Random noise was overlaid over the pressure coefficients at each angle of attack to simulate imperfect conditions. The average error difference was taken was taken over 100 readings where the error difference is the same as for the individual sensor weightings and:

$$\text{Predicted } \alpha = w\alpha_{upper} + (w - 1)\alpha_{lower} \quad (23)$$

Where w is the upper to lower surface prediction weighted average ratio.

The weighted average ratio was optimised to minimise the error. Due to the random nature of the problem due to the overlaid noise, a statistical optimisation approach was taken. The average and standard deviation was calculated for 100 iterations for each weighting value between 0 and 1 at a resolution of 0.01.

The weighting factors for the individual sensors were the same for both the upper and lower surfaces. Thus, the effects of a changing upper to lower weighting ratio was independent of the

results of the individual sensor weightings. An arbitrary set of weighting factors was used and kept consistent for each input. This allowed optimisation of the upper and lower weightings to be performed in parallel with the optimisation of individual sensor weightings, saving time.

The averages were plotted against the standard deviations and the Pareto optimal solutions were found.

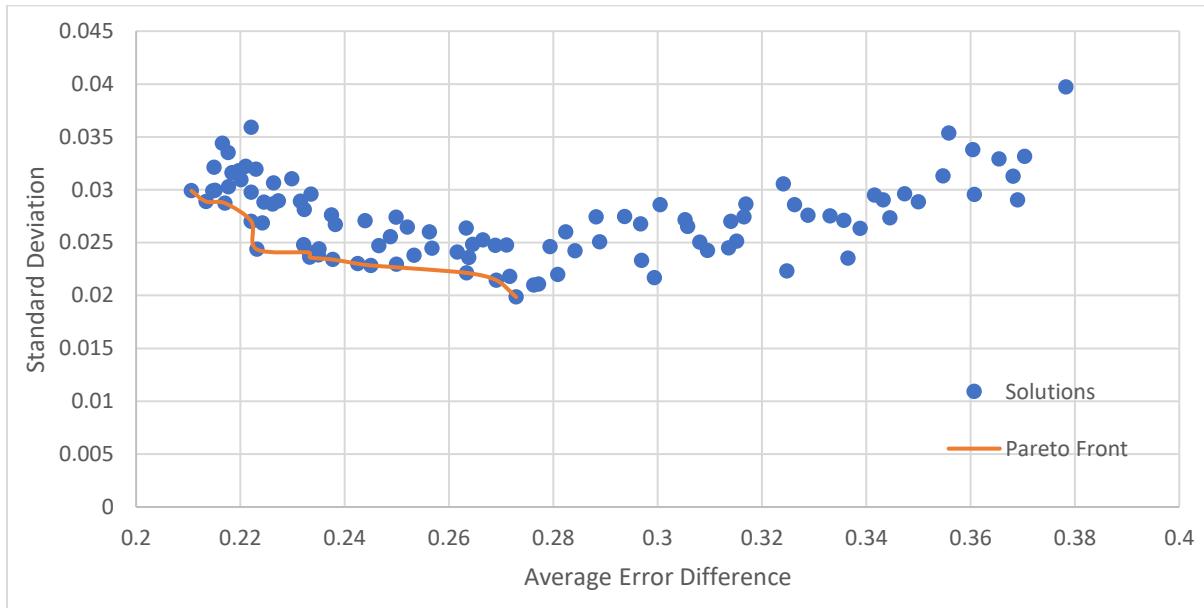


Figure 50 – Average Error against Standard Deviation for the Weighting Factor over 100 iterations.

The weighting factor chosen was the one which gave the closest solution for the average and standard deviation to the ideal solution. This was at an upper to lower weighted average ratio of 0.77.

4.2. Method 2 – Integration

For the selected sensors, the only portion of the aerofoil without any coverage is the trailing edge. To integrate across all sensors would not include the last 17% of the chord, which would result in a significant underestimation of the total lift. The pressure should not be expected to increase any further towards the trailing edge but adjustments should be made to ensure the most consistent result.

One way to predict the pressure distribution across the remainder of the flap is to apply the Kutta condition which is stated as follows: “A body with a sharp trailing edge in motion through a fluid

creates about itself a circulation of sufficient strength to hold the rear stagnation point at the trailing edge of finite angle to make the flow along the trailing edge bisector angle smooth.” [16]. From there it can be assumed that both the upper and lower surfaces would converge at the same pressure coefficient value at the trailing edge. This is further backed up by simulations made through XFOIL in which the flow converges at the trailing edge as expected:

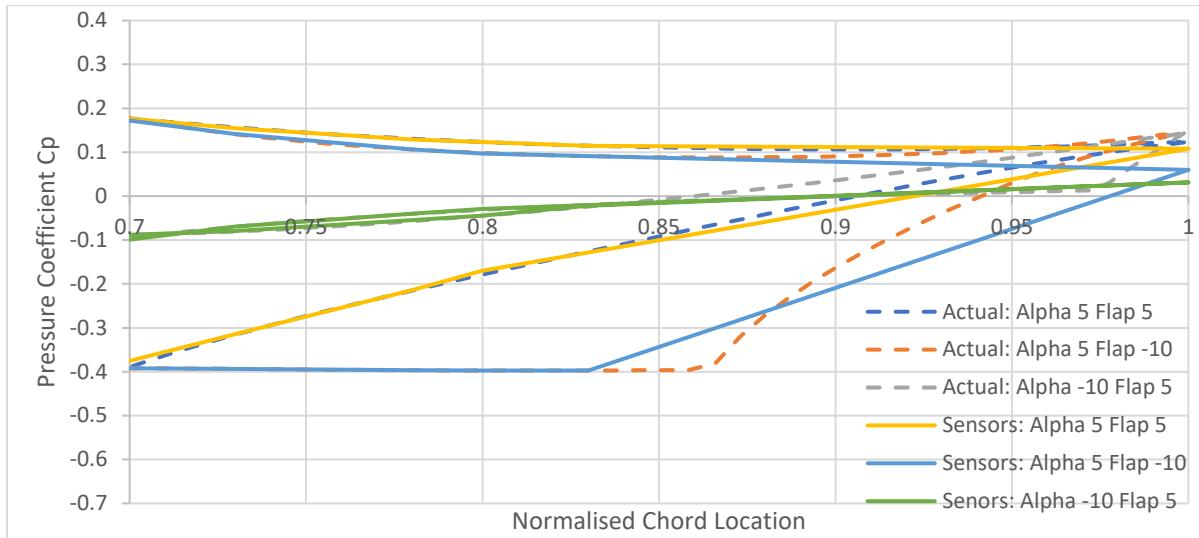


Figure 51 – Pressure Distribution across the flap on a NACA0012 aerofoil. Comparison of approximation of distribution from sensors compared to actual distribution.

From Figure 51, the approximation of the distribution from the sensors is an underestimate in all cases for the trailing edge pressure coefficient. Since the final sensor is at 83% chord, any changes in pressure coefficients beyond this cannot be found. Therefore, this method may have a significant error. For the three cases above the errors were:

Table 8 – Comparison of approximation using a limited set of sensors compared to a set of all possible sensors.

Case	Full Set Integration	Sensor Approximation	Difference
Alpha 5 Flap 5	0.097587371	0.08910665	-0.008480721
Alpha 5 Flap -10	-0.136077669	-0.12676855	0.009309119
Alpha -10 Flap 5	0.007433735	-0.000416	-0.007849735

This error is significant across the flap itself, leading to up to 100% error in the case of an angle of attack of *-10 degrees* and a flap angle of *5 degrees*. However, in comparison to the lift coefficient across the entire wing, it is much smaller at 0.76% of the overall lift coefficient of -0.9226. Overall, the error is quite small as shown in the figure below:

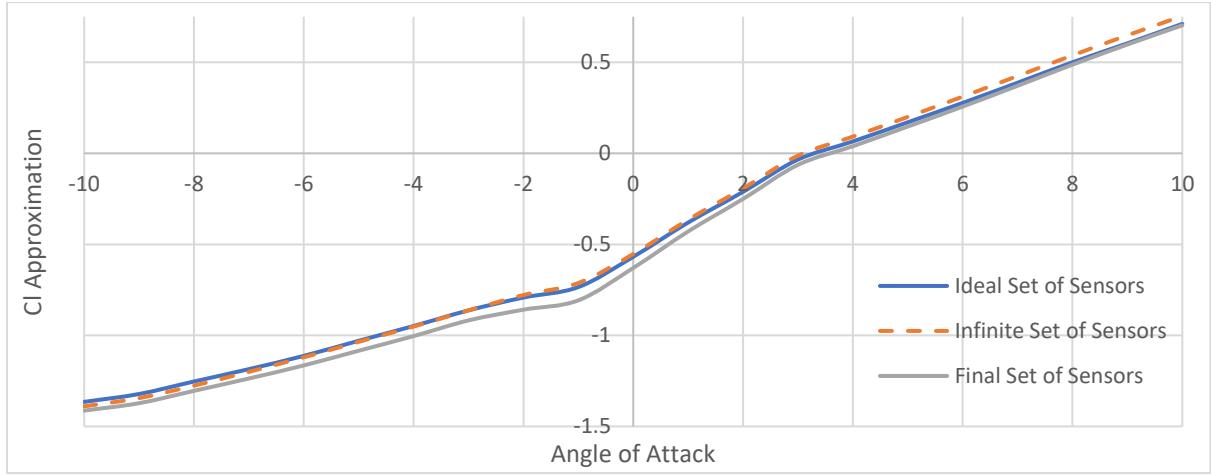


Figure 52 – Comparison of different sets of sensors for Lift Coefficient against Angle of Attack for a Flap Angle of -12 degrees.

Overall, this method could be computed faster than the maximum accuracy of the timer available at 1 millisecond for all iterations. It could then be assumed that the worst-case speed that the method could be computed at was 0.0005 *seconds*, or 2000 *Hz*. This is a factor of 16.67 times quicker than the maximum read rate of the pressure sensors meaning this is also a viable method in terms of computational speed.

4.2.1. Approximation of Total Lift across the Wing

The lift approximated from the methods above are only two-dimensional slices of the overall wing and do not represent the lift over the whole wing. The approximation of the total lift is dependent on the span-wise position of the sensor board. Assuming an elliptical lift distribution, each sensor board can be approximated as proportion of the total lift across the wing. This can then be multiplied by a factor for an equivalent constant Cl across the wing and the total lift then approximated.

Multiple sections could be used to better approximate the total span-wise distribution and the full lift over the span. Each individual section would be adjusted by a factor accounting for the difference between the measured lift and the overall lift. The average of these approximated lifts would be the overall approximate lift over the wing. The total approximate lift would simply be the summation of the approximate lift over both wings.

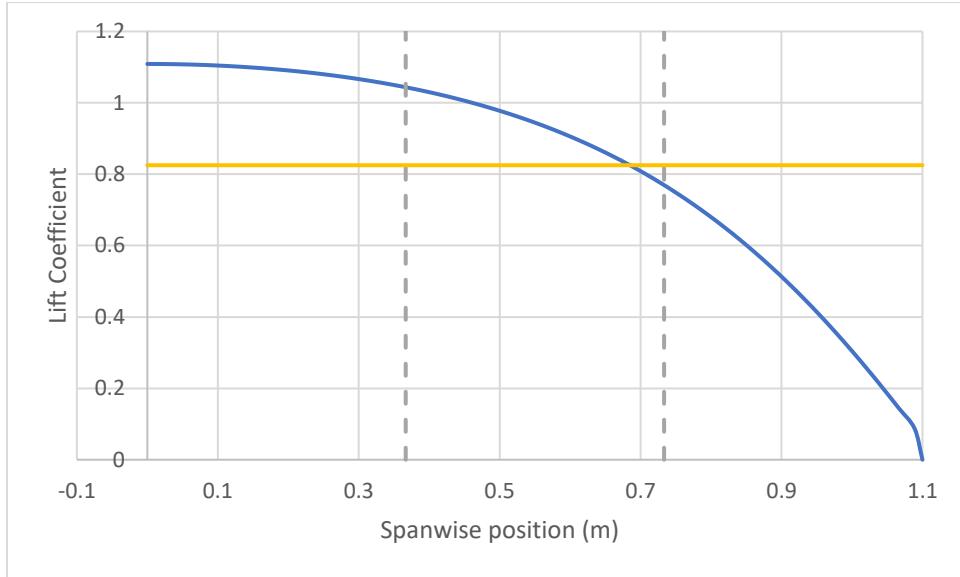


Figure 53 – Elliptical Lift Distribution compared to an equivalent constant Cl across semi-span with Sensor Positions highlighted. Where the blue curve is the elliptical lift distribution.

4.2.2. Maintaining Lift

It would be assumed that to maintain the desired lift, each wing would produce half of the required total lift. Then, the flaps can be deflected to either increase or decrease the lift over each wing section and maintain the total desired lift. With a high enough read rate from the sensors and lift approximation, only the direction of the flaps would be enough as the flaps could continuously deflect.

An improved system used a lookup table of lift coefficient against angle of attack and flap angle to approximate the required flap angle for maximum lift. As stated above, this provides both magnitude as well as direction, which allows for the option of increased control over the servo speed, meaning less chance of overshooting the required lift and quicker damping of gusts.

The lift coefficients provided by the lookup table are taken from XFOIL simulations for a two-dimensional aerofoil section. On the wing, the flaps only occupy a proportion of the total span, which must be accommodated for when investigating the required deflection such that:

$$Cl = Cl_{\gamma_0}(1 - r) + Cl_{\gamma}r \quad (24)$$

Where r is the ratio of the span occupied by the flap, Cl_{γ} is the lift coefficient at the flap angle of γ and Cl_{γ_0} is the lift coefficient at zero flap. The required angle can then be sent to the flap controllers and the flap can be adjusted.

It has been shown that it would be possible to obtain a value of the lift over a wing from the pressure distribution across the wing and that various methods have the capability of doing so within reasonable degrees of accuracy. With knowledge of the lift across the wing, the required flap angle can be identified and adjusted to maintain constant lift.

4.3.Revised Weight

After the manufacturing and procurement of components, the maximum take-off weight was revised to confirm that the initial approximation was correct and within reasonable limits for take-off. The weight of the final UAV was lower than the initial approximation by 400 g.

Table 9 – Revised Maximum Take Off Weight

Component	Mass per unit/metre	Count/Total Length	Total Mass	Weight	
Wing Left	0.598	1	0.598	5.86638	
Wing Right	0.6	1	0.6	5.886	
Wing Servos	0.011	12	0.132	1.29492	
4Max Motors	0.127	2	0.254	2.49174	
Propeller	0.03	2	0.06	0.5886	
Pressure Sensor Tubing	0.02	5	0.1	0.981	
Pitot Probe	0.03	1	0.03	0.2943	
Pressure Board	0.045	5	0.225	2.20725	
Pixhawk and Component Support Board	0.312	1	0.312	3.06072	
Webcam	0.022	1	0.022	0.21582	
Raspberry Pi	0.023	1	0.023	0.22563	
Undercarriage	0.5	1	0.5	4.905	
Fuselage Servos	0.011	2	0.022	0.21582	
Fuselage	0.75	1	0.75	7.3575	
Wiring	0.125	1	0.125	1.22625	
Auxilliary Battery	0.232	1	0.232	2.27592	
Lithium Polymer Battery	0.545	1	0.545	5.34645	
			4.53	44.4393	Total

5. Test wing design and manufacture

A small wing section was to be developed to be able to be fitted to a test rig for testing and development of the wing technology. This would then be scaled up for the full size final wing.

5.1. Section methodology

The wing was designed with a sectional methodology; this would mean getting inside the wing to change delicate components like the sensor boxes would be easier. However, this meant that each section would not be able to be glued to each other or to the structural spars. For the test wing, there would be seven different sections, four made from foam, one 3D printed and also laser cut plywood endplates.

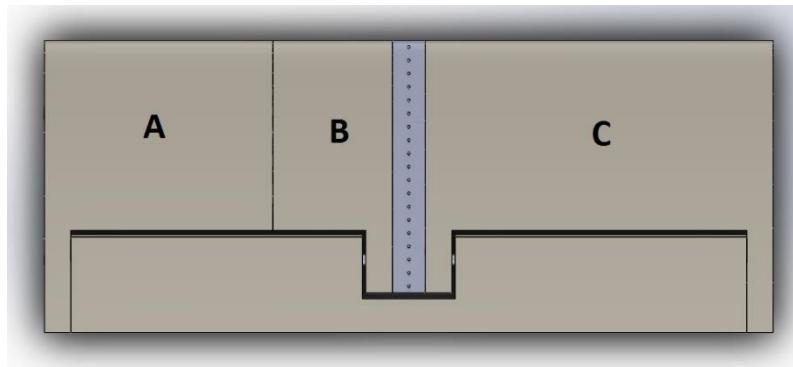


Figure 54- Test wing assembly

Two different cross sectional designs were used, the two outer sections, A and C (Figure 56), were the same with only a small cut-out section for servo wires and the wires of the sensor box. With section B (Figure 55) having a much larger cut out for the sensor box to fit in. Since the design includes one large flap along the entire wing section there is a cut out in the middle to go around the pressure tapping section.

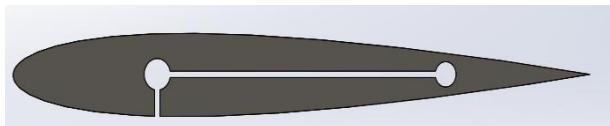


Figure 56- Cross-section of wing-part A &C

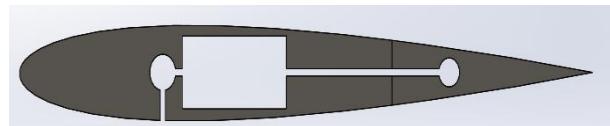


Figure 55- Cross-section of wing-part C

5.2.Spar Analysis

At the point of test wing design the expected weight of the eventual full UAV was unknown so spar diameters were arbitrarily chosen, a two-spar design was used as this seemed to be a likely configuration that would be used as it allows a greater reduction in deflection while having smaller diameter spars, something that would be important due to the lower thickness of the wing. Having two spars also allows the rear spar, dependent on placement along the chord line, to act as the hinge point for the flap.

5.3.Flap Actuation Mechanism

The mechanism for moving the flaps is simple, a push rod is connected to the servo arm using a z bend to keep it in place and a clevis glued to the other end. This clevis is then attached to a control horn which has been secured to the control surface. When the servo arm moves, the control surface also moves rotating around the rear spar.

The flap mechanism and the servos which drive them are on the underside of the wing for a few reasons, firstly for the flaps it means there's a pull force being exhibited on the control surface since the airflow is trying to push the control surface back to a neutral position. This is better for the control rod as rods will deform a lot less in tension as compared to compression, and won't be compressing the foam control surface. Secondly although the servos will be mounted in recesses in the wing the mechanism will still affect the airflow around it, these affects are more prevalent on the upper surface of the wing since the airflow over this surface is faster.



Figure 57- Flap Actuation Mechanism

5.4.Servo Selection

An online calculator [17] was used to calculate the torque required from the servo. After inputting the appropriate control surface size and max deflection angle for the servo arm and control surface, all at an expected maximum speed of 18 m/s .

5.5.Pressure Tapping Section Design

A way to gather a large amount of pressure readings to build up a good quality pressure profile around the aerofoil was needed, the solution was to create a complex piece which had a multitude of channels running through it that could then be connected to tubing which could in turn be connected to a pressure sensor. This design evolved through the iterations detailed below.

For the first iteration of a pressure tapping section (Figure 58) a solid piece 25 mm across with 67 tapping's was drawn up in SolidWorks and then printed.

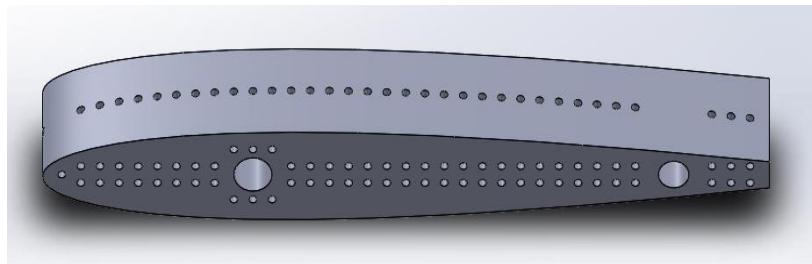


Figure 58 - Isometric view of pressure tapping piece first iteration

After this the second iteration was designed to be half hollow so as to allow the gluing of pneumatic tubing to the tapping channels while not protruding from the piece (Figure 59), this also made the piece lighter. Also, the tapping on the upper and lower surfaces were staggered so that the separation bubbles formed around each one would not interfere with the next tapping. The first iteration had tapping's in essentially 'random' x positions, now the key x positions gathered from XFOIL data were used and then more tapping's put in around these, with a larger number near the leading edge.

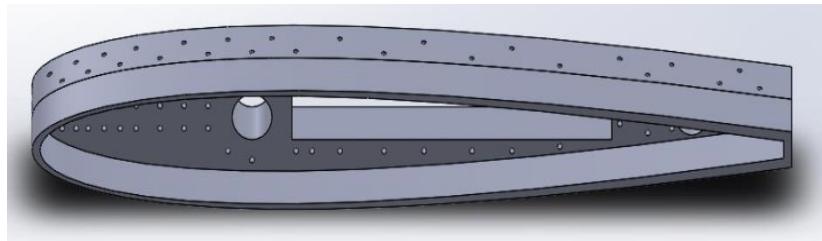


Figure 59 - Isometric view of pressure tapping section second iteration

From this solid spar supports were added (Figure 60) which then required some tapping channels to be re-routed through the spar support. (Figure 61)

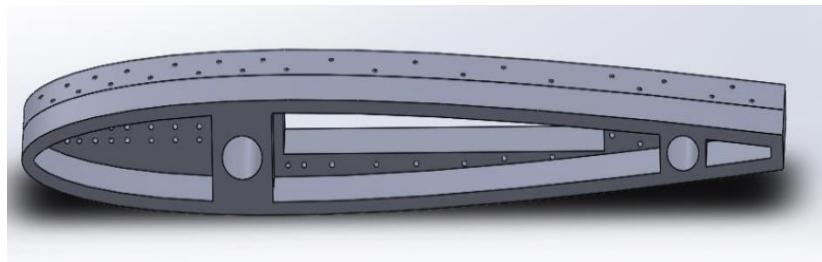


Figure 60 - Isometric view of pressure tapping section third iteration.

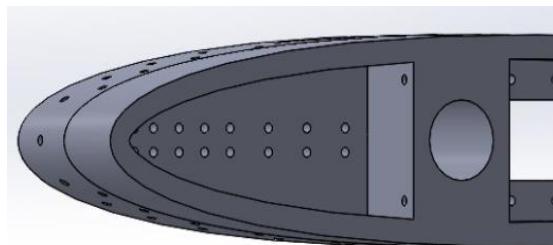


Figure 61 - View showing re-routing of pressure channels through spar supports.

5.6.Manufacture

The manufacture and assembly process for the test wing is detailed below:

1. Cut side profile shapes from Dow 3000A foam using hot wire cutter, this includes an added rounded trailing edge.
2. Measure and then cut flap sections out using a knife, and cutting the rounded trailing edge off.
3. Mark and cut servo recesses out of the foam up until the wiring slot, this slot gave a flat surface to which the servos could be mounted.
4. Glue Velcro to both underside of servo and inside servo recess.
5. The different sections are slid onto the spars and the flap mechanism constructed.

6. Final Wing Development and Manufacture

Originally the test wing developed would be treated as a wing section and simply multiplied and connected to form the wings for the UAV however the UAV came with a functional middle section that would be better than one constructed by ourselves, so the wing needed to be adapted to these changes.

6.1. Spar Sizing

Ansys was used to run simulations to enable the calculations of spar deflections of the wings while in flight. The simulations were run on only the spars themselves as the non-linear properties of the foam used as the other wing construction material would give inaccurate results, in addition to this the properties of the foam needed were unable to be found. As the total desired wing span to give sufficient lift was 2.2 m (without including fuselage) that left 1.1 m for each wing, and as we were going to use the middle wing section that came with Bormatec explorer which measured 0.3 m outboard of the fuselage it left a wing section that needed to be 0.8 m long. So thus, this was the length of the spar analysed in Ansys, the total expected weight of the aircraft was calculated to be 4.5 kg was then applied and then a load factor of 3 was used to calculate the maximum force the wing would be subjected to. The properties for carbon fibre were collected from the easy composites website [18] with the unknowns that were still needed collected from the online source [19].

To validate the simulations run in Ansys simple hand calculations were performed on a single spar to check that the outputted results were close to those found by us. This was done by using the deflection equation:

$$v = \frac{W l^3}{8 E I} \quad (25)$$

Where v is the maximum deflection, W is the total load applied, l is the length of the beam, E is the modulus of elasticity of the material (in this case carbon fibre) and I is the moment of inertia due to the shape of the beam. The moment of inertia of the beam was calculated using equation below:

$$I = \frac{\pi (D^4 - d^4)}{64} \quad (26)$$

Where D is the outer diameter of the tube and d is the inner diameter of the tube.

Table 10 - Deflection as shown by Ansys and by hand calculations

Type	Outer Diameter(mm)	Inner Diameter(mm)	Ansys Deflection(mm)	Hand Calculation (mm)
Tube	10	8	208.3	208.2

After comparing the two values deflection in Table 10 which have a difference of 1 mm, the Ansys simulations were deemed reliable.

This was then run with various inner and outer diameter values widely available on the market and with solid rods. Due to our wing being thin it became apparent that using two spars would allow us to use a smaller diameter spar but still achieve deflection within the correct limits. Lower deflection is better as it will mean that if unexpected force values are encountered the wing will remain structurally sound. The results of this analysis along with weight of each selection is tabulated below.

Table 11 - Deflections of a one spar system

Type	Outer Diameter(mm)	Inner Diameter(mm)	Deflection(mm)	Weight(g)
Rod	6	n/a	332.3	34
Rod	8	n/a	105.1	60
Tube	10	8	208.3	40
Rod	10	n/a	43.1	94
Tube	12.7	10	77	60
Rod	12	n/a	20.8	136
Tube	15.5	12.7	28.8	72

Table 12 - Deflections of a two-spar system

LE Spar Diameter(mm)	TE Spar Diameter(mm)	LE Spar deflection(mm)	TE Spar Deflection(mm)	Weight(g)
10	8	115.7	195.3	68
10	10	104.2	104.2	80
12.7	8	47.1	170	88
12.7	10	43	91.8	100
12.7	12.7	38.4	38.4	120

A leading-edge spar diameter of 12.7 mm and a trailing edge spar of 10 mm were chosen as they exhibited under 100 mm of deflection for the maximum load case and were well away from failure. Also, due to the thin nature of the aerofoil and the mounting angle of wing, spars any bigger would not have been able to fit into the wing.

In the end roll wrapped carbon fibre tubes were selected over pultruded carbon fibre tubes since roll wrapped tubes have greater strength in directions other than directly along their length, which is much more suited to a small aircraft spar as these properties allow greater resistance to crush,

fractures and torsional forces, especially so as the analysis treated the spar as a simple bending beam. This type of carbon fibre tube also comes with an added purchasing cost.

6.2.Connection of Constructed Wings to Fuselage

The Bormatec explorer aircraft wing came in three sections, two outboard wing parts (one for each wing) and a middle section that the outboard wings slid onto. This middle section contained glued in screw threads that allowed it to be screwed into the fuselage, securing it in place. Metal engine mounts were also part of the middle wing section, these features made it an important task to be able to keep this middle section while finding a way to attach our own outboard wings.

Firstly, the zero-lift angle of the original wing needed to be found so that the rest of the UAV platform would be at the original intended angle of attack when no lift was being generated. This information could not be found so the outline of the aerofoil was traced out on paper, this tracing was then photographed and uploaded into Engauge software, this software allows points to be plotted on the image and after inputting three known values for co-ordinates (from measuring the physical trace) outputs co-ordinates for any number of points on the curves. These aerofoil co-ordinates were then put into Xfoil and the zero-lift angle calculated (Figure 62), our manufactured outboard sections would be mounted at this angle due to the fact our aerofoil is symmetrical it generates zero lift at zero angle of attack.

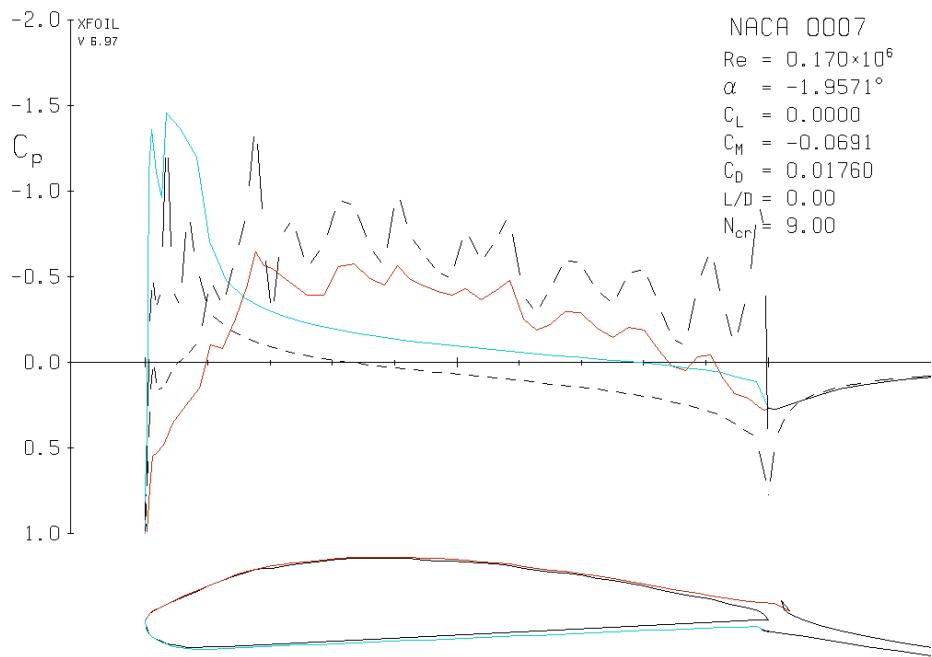


Figure 62 - Xfoil analysis of traced wing section

As the middle section chord length was 240 mm the chord length of our wing sections was lengthened by 20 mm to match.

Next a secure way to connect the outboard wings to the middle wing sections was needed. As two spars protruded from the middle section, each with a diameter of 6 mm, and from the spar analysis the spars chosen had larger internal diameters than this so could slide on top. To enable this to be possible the spars had to be 136 mm apart, to line up the leading edges and also to have an acceptable flap length the forward spar was moved forward from the quarter chord position. This movement of the spars while using the rear spar as the flap hinge point resulted in a flap length of 87 mm which is 36.2% of chord, only 1.2 % away from the original intended value. A plastic piece that slots together and then wire threaded through – as shown in Figure 63, Figure 64 and Figure 65 – to hold in place was repurposed to hold the wings onto the spars when in flight. To add extra safety a circular clamp was also placed on top of the rear spar which when tightened added extra frictional force making it harder for the wing to slide off.

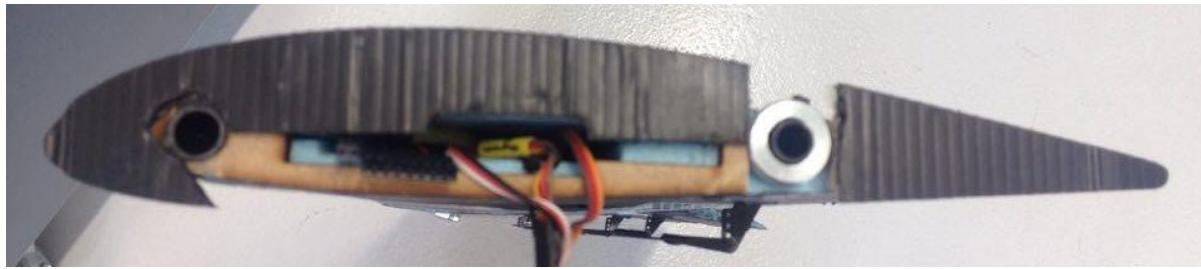


Figure 63 - Side view of outboard wing connection



Figure 64 - Side view of middle wing connection

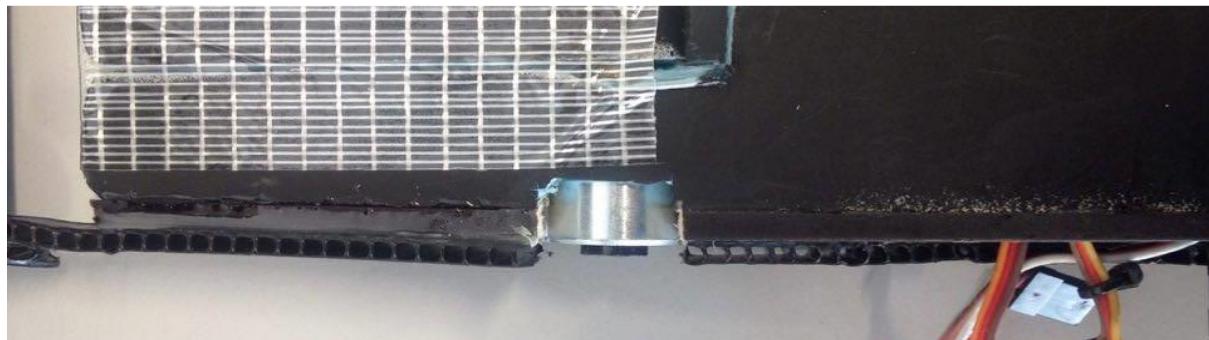


Figure 65 - Plan view of outboard wing connection

6.3. Changes to Pressure Tapping Section

With the chord length being lengthened and wing being mounted at an angle the pressure tapping section needed some alterations, firstly the x positions of the tapping's were changed to reflect the longer chord, however each position matched the original section in regard to their x-position in percentage of chord. Secondly the mounting at an angle of attack made it impossible for some of the tapping's to be routed round the new spar positions so were removed. Table 13 details the x-position of each tapping position.

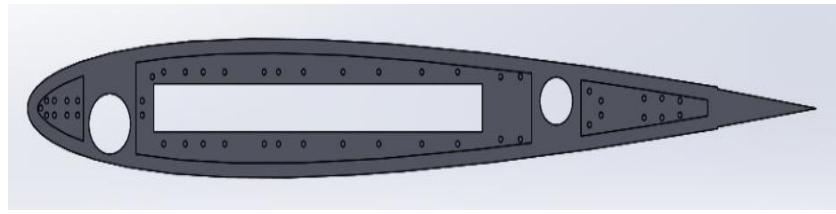


Figure 66 - Side view of the final pressure taping section

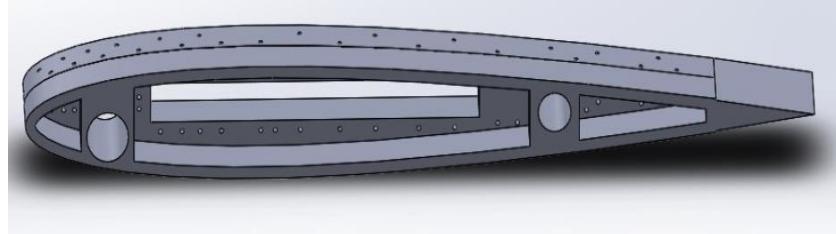


Figure 67 - Isometric view of the final pressure taping section

Table 13 - X-position and percent of chord length of each tapping

Tapping	1	2	3	4	5	6	7	8	9	10	11	12	13
X position(mm)	LE	4.4	7.5	11	14	18	22	26	32	38	44	49	55
% Chord		1.8	3.1	4.6	5.8	7.5	9.2	10.8	13.3	15.8	18.3	20.4	22.9
Tapping	15	16	17	18	19	20	21	22	23	24	25	26	27
X position(mm)	66	77	88	98	110	120	132	147	154	160	172	177	182
% Chord	27.5	32.1	36.7	40.8	45.8	50.0	55.0	61.3	64.2	66.7	71.7	73.8	75.8

6.4. Sectional Layout

In this full wing 11 sections are used, 7 made from foam, 2 3D printed and 2 laser-cut from plywood. Of the sections made from foam 4 are wing sections and 3 are flaps. In the original design, parts A and B would have been swapped with section E and A and B would have been reversed. However due to the print quality of section G it which left it mostly unusable we swapped the pieces around to allow for less wiring for the sensor box and also easier access to it. As can be seen in the figures below section A (Figure 69), has a larger wiring section cut out which would have enabled the tubes from section G to be connected to the sensor box. Section B (Figure 70) has a large cut out so that the sensor box can fit snuggly inside, and sections E and H (Figure 71) only have small wiring cut outs. The ply end plate next to H has no cut outs in while the end plate next to section A has a large cut out for wiring connections.

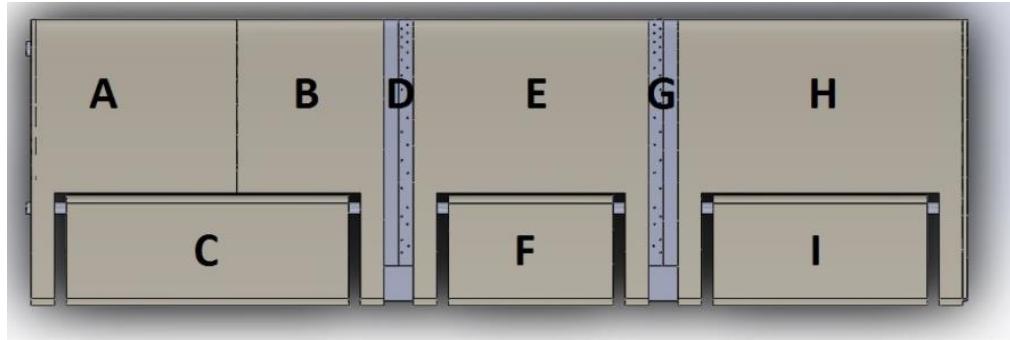


Figure 68 - Plan view of final wing assembly

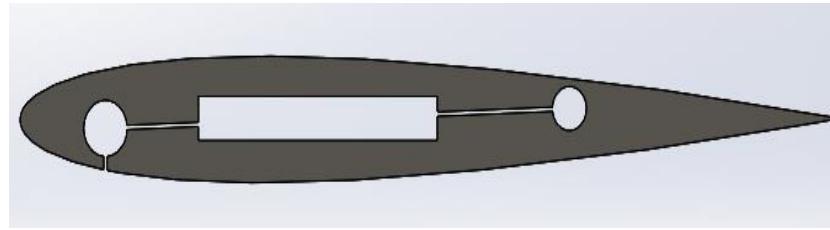


Figure 69 - Side view of section A

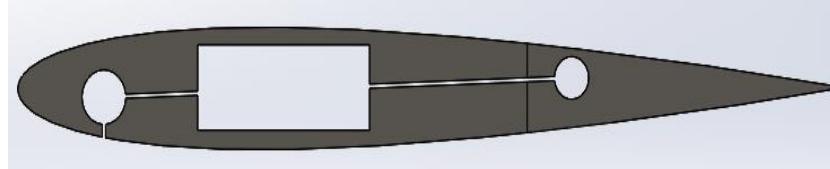


Figure 70 - Side view of section B

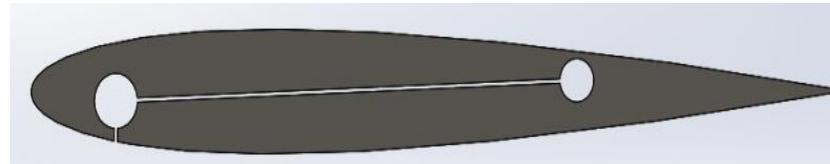


Figure 71 - Side view of section E & H

6.5.Servo Calculations

The same online calculator [17], as used for the test wing, was used to calculate the torque requirements of the servos for the various flaps. The expected top speed of the aircraft was inputted as 18 ms^{-1} , a safety factor of 50% was then applied to this torque value giving a needed torque of 0.677 kg cm^{-1} , as the servo would also need to be able to deal with gusts up to an expected 10ms^{-1} which would need a maximum torque value of 1.6 kg cm^{-1} (including the 50% safety factor). So, a tower pro SG90 digital micro servo was chosen with a stall torque of 1.8 kg cm^{-1} . Their lightweight and very affordable price additional key reasons for their selection.

6.6.Wing Manufacture

The manufacturing process for the final UAV wing was like the process for the test wing however changes have been implemented from lessons learned during the manufacture of the test wing. The steps are detailed below:

1. 2D side profile sections were cut out of correct length foam, Dow floormate 3000A, including conventions such as adding a circle to the trailing edge to not have any sharp edges and channels for the wire to enter and cut out the internal structure.
2. Next the flap slots were marked onto the foam and then cut out using a craft knife, and the rounded trailing edge sections were cut to give the sharp trailing edge of an aerofoil.
3. Due to the nature in which the flap slots were cut, there was an uneven finish from bits of foam flaking off and not straight cutting lines. Coarse and then fine sandpaper was used to correct this, giving a much better finish than that which was achieved on the test wing. Sand paper attached to small diameter doweling was then used to sand the spar holes in the foam to make pushing the spars through easier, however a tight fit was still achieved.
4. Servo recesses were cut next, due to the wiring slot in the design, cutting up to this point gave a perfectly flat surface for the servos to be mounted.
5. To give a smoother surface finish, one layer of papier Mache was applied using plain white printing paper and a mixture of PVA glue and water which was applied to the foam using a paint brush and then all the wrinkles smoothed out using a plastic scraper. The paper was deliberately laid on over the edges so next that was cut away leaving the paper Mache flush with the edges of the foam.
6. Strong gripping Velcro was then glued, using epoxy resin, to the bottom surface of the servo and to the flat surface of the servo recess. It was important to use epoxy and not any other glue as solvent based glues melt the foam away.
7. The top of the servos were then paper Mached over to leave only the servo arms and their respective channels clear.
8. Black spray paint was then applied to the dry papier Mache surface.
9. Each piece was then slid onto the spars with servo extension cables used where appropriate with each servo wire labelled. With also the sensor box in place inside the wing.

10. A laser cut end plate was then clamped, using a 10 mm rod clamp and a hose clip on the 12.7 mm spar, onto the outboard section of the wing.
11. Another laser cut end plate, this time with internal cut outs for wiring, was then glued using epoxy to the edge of the inner most foam section.
12. The cardboard section cut off the supplied UAV wing is glued to the inside ply endplate, making sure that the slots will line up with the piece on the aircraft.
13. Using a hack saw a block was cut out of the inner ply endplate and a small amount of foam to allow for a recess for another 10 mm clamp so that the inner edge of the outer wing and the outer section of the middle wing would be able to sit together flush.
14. Flap mechanism was constructed next, this involved gluing the control horns to the flaps and then cutting and bending the push rods to the correct length and gluing clevises to one end of the push rod, so that the flaps would be level when the servos were at 90 *degrees*.

7. Test Rig Development

7.1. Main Purpose and Specification

The test rig is one of the most vital components in the wind tunnel test. The design of a test rig is based on the testing requirement and it must have the capability to provide accurate results. In this section, the specifications of the test rig, based on the wind tunnel test requirement, will be discussed.

To prove that the concept of the pressure measurement system can accurately predict the aerodynamic forces acting on the wings during flight, static wind tunnel testing was required. This made sure that the pressure measurement system would work as intended on a test wing model before applying it on an actual UAV.

The basic concept of the pressure measurement system verification is to compare the lift force which is evaluated from the pressure distribution data to the lift force that is directly measured using the test rig during wind tunnel testing. However, it is known that the wing is subjected to both lift and drag forces during the flight. Hence the test rig had to be designed in a way that is capable to effectively decouple the aerodynamic forces that act on the test wing section into separate lift and drag components and measure them accurately during the testing process.

Based on the test wing model dimension and the known maximum lift and drag coefficient for a NACA 0012 aerofoil from the simulations among the interested angle of attacks and wind speed, the maximum lift and drag force was expected to be in the region of 40 N and 3.5 N respectively. However, it is always a best practice to leave a safety margin for the maximum loads. With the safety factor of 1.5, the new maximum lift and drag forces were 60 N and 5.25 N respectively. Meaning that the test rig should be able to measure up to these maximum possible loads accurately. To make sure the aerodynamic forces are measured as accurately as possible, besides measurement device calibration, the test rig's structural components and the mechanism system are required to be rigid so that the forces that act on the wing are transmitted directly to the load measurement transducer and measured accurately.

7.2.Load Measurement Device

During the test, the decoupled aerodynamic forces are transmitted to the load measurement devices for measurement. There are different load transducers available to measure the load in a system. However, it is vital to compare the advantages and disadvantages for each transducer in the aspect of load limits, response time, dimension, cost, and the ease of integration into the system and test rig design.

7.2.1. Strain Gauge

A popular method of measuring load is to use strain gauges. The electrical resistance of foil in the strain gauge varies proportionally to the amount of strain it experienced. By placing strain gauges onto the surface of the test rig support structure, the amount of strain that is experienced by the support structure in a direction due to the applied load would be directly transferred to the strain gauge. The strain experienced by the foil causes an increase in electrical resistance, and hence the output voltage obtained through signal conditioning can then be correlate to the magnitude of the load.

An example of using strain gauges to measure aerodynamic forces can be found in Timothy Craig Witushynsky's paper [20] where several strain gauges are applied onto an L-shape supporting structure and produce signals in terms of voltage in response to the strain caused by the aerodynamic forces.

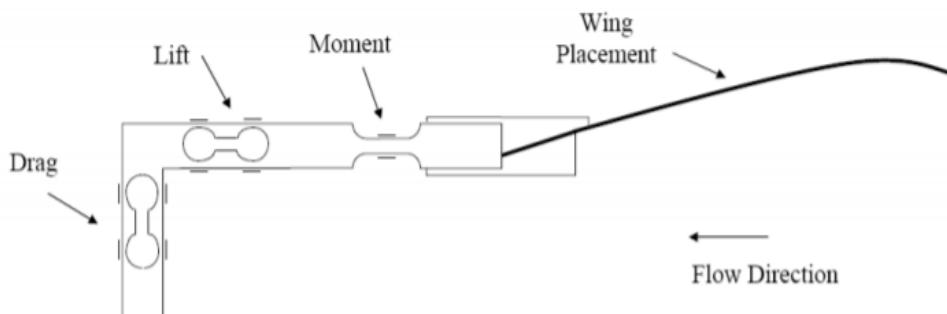


Figure 72 - Aerodynamic Force Measurement System and Strain Gauges Placement [20]

Using strain gauges as the load measurement device would provide flexibility while designing the test rig. Strain gauges are available in various sensitivity and loading ranges, indicating that specific strain gauges can be used based on the experimental requirements. Compared to other load measuring devices, strain gauges are relatively cheap. However, strain gauges are orientation sensitive due to the nature of the foil alignment. They must be placed in a specific orientation on the structure to provide the best functionality, which could be quite difficult to achieve without help of an experienced technician. If so, more time would be required to do additional calibrations on both the individual strain gauges and the whole system.

7.2.2. Single Axis Load Cell

To resolve the problem of using strain gauges as a load measurement device as discussed earlier, single axis strain gauge load cells where series of strain gauges are placed on a piece of precise machined block of material can be used. Load cells are normally available with specific designs, geometries, dimensions, and load measurement directions. This means that the test rig design would need to consider load cells as part of the supporting structure based on the nature of the load cells. Furthermore, as these stain gauge load cells are pre-manufactured, the quality and performance would be expected to be relatively consistent, so many of them could be used in a system without worrying about the position of the strain gauges. Single axis strain gauge load cell's high accuracy, sensitivity and wide range of load measurement mean that they are commonly used in weight measuring scales, from household kitchen scales to industrial heavy duty weighing machines.

Single axis load cells were used in designing External Six Component Strain Gauge Balance and were described in the paper by Samardžić et al. [21]. This balance consists of six single axis load cells which are capable of measuring lift force, drag force, side force, pitching moment, rolling moment and yawing moment [21].

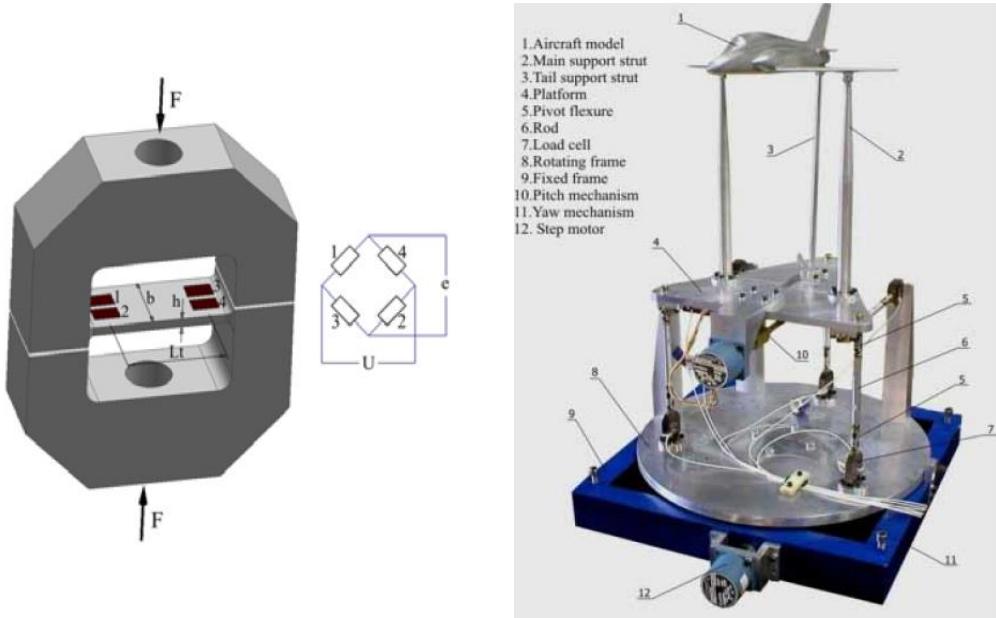


Figure 73 - Single-force load cell (Left) and force measurement system by Samardžić et al. (Right) (External Six-Component Strain Gauge Balance for Low Speed Wind Tunnels [21]

7.2.3. Force-Torque Load Cell

Force-Torque load cells are an advanced transducer which can measure and resolve the forces and torque in the x, y and z axes. This kind of device is highly utilised in the robotics industry and medical research due to its high accuracy and ability to resolve the complex forces and torques into the x, y and z axes without occupying a big space.

To use a force-torque load cell for force decoupling, it needs to be part of the supporting structure located in between the test wing model and the rigid support structure. This allows the force-torque load cell to measure and decouple the aerodynamic loads experienced by the test model before they are transmitted to the support structure. The lift forces, drag forces, and the pitching moments acting on the wing could be found easily using this device by assigning each axis to a specific force acting direction. In this case, the test rig design for wind tunnel testing can be significantly simplified with the use of a force-torque load cell. Using this load cell could save a lot of time during the test rig design stage, it is however, much more expensive compared to other load measurement devices.

After considering the advantages and disadvantages of each load measurement device, single load cells are considered the most cost and time effective option for the wind tunnel test rig. Single load cells are much cheaper than the force-torque load cell and require less calibration compared to strain gauges. Although load cells are much bulkier than strain gauges, it would not be a problem provided that the test rig is designed in a way such that they are outside of the flow generated by the wind tunnel.

7.3. Test Rig Design

7.3.1. *Selection of single axis load cell*

There are several types of single axis load cell available, the most common load cells including shear beam load cells, compression load cells and S-beam load cells. Table 14 shows the description for each type of load cells:

Table 14 - Common load cell types [22]

Load Cells	Description
Bending Beam	<ul style="list-style-type: none"> - A straight box of material loaded at one end while the other end fixed. - Strain gauges bonded on the upper and lower section at points of maximum strain. - Low capacity application - Must be loaded correctly to obtain consistent results.
Compression	<ul style="list-style-type: none"> - A material block to be loaded at one point by compression. - Commonly in integral button design.
S-Beam	<ul style="list-style-type: none"> - “S” shape block material - Suitable use in both tension and compression load measurement.



Figure 74 - (Left to right) Bending Beam load cell, compression load cell and S-Beam load cell [22]

Considering load cells are to be integrated into the test rig design, the compression load cell structure without any fixture screw holes is not suitable to be part of the test rig structure in this case. As mentioned in section 7.1, the maximum lift and drag forces with a safety factor of 1.5 are 60 N and 5.25 N respectively. These loads are considered quite small and can be easily found available within the load measuring range of both bending beam load cell and S-beam load cell.

Further comparison was made between the bending beam load cells and S-beam load cells to determine the suitability of each load cell for this design project. First, the loading condition. In this case, the test rig will only be used for static wind tunnel testing. Meaning the loads acting on the test wing model are expected to be constant in the span-wise direction, except for during the transition of changing the angle of attack. However, the lift component on the test wing will only transition into the opposite direction when the wing changes from positive angle of attack to negative angle of attack. Although the magnitude of lift acting on a symmetrical aerofoil would be expected to be the same for the same angle of angle but in the opposite direction, the aerofoil shape of the final manufactured product might turn out not be perfect symmetry. This makes testing for negative angle of attacks to be necessary as part of the validation of the pressure measurement procedure.

Since an S-beam load cell has the capability of measuring forces in both tension and compression, the opposite direction of wing loading can easily be measured during the testing. However, this is not the case when using bending beam load cells, which are normally designed for single direction loading, without additional calibration. Flipping the test wing section upside down and measuring the lift at negative angle attack can solve this problem. It makes sure the lift component acting in the same direction as the load measuring direction, with no additional calibration required.

Secondly, the cost of the load cells. Bending beam load cells are usually designed for single direction loading and they are significantly cheaper than the S-beam load cells. Considering each single axis load cell can only measure the force in one axis, a combination of at least two single axis load cells would be required to measure the lift and drag components. By considering both the time and budget available for this project, load cells were chosen as the load measurement device for the test rig which allows the wind tunnel test to be carried out on time without excess calibration while keeping the test rig within the budget available.

7.3.2. Lift Drag Decoupling Design

7.3.2.1. Initial Strut Design Concept

The test rig needs to mechanically decouple the combination of forces into lift and drag before they are measured by the load cells. The basic idea of the force measuring system is one which includes a strut which attaches one of its end to a platform which is rigidly connected to the test wing structure and the other end attaches to the load cell which is directly mounted on a rigid structure. This allows the aerodynamic forces experienced by the test wing section be transmitted through the strut and directly to the load cell for measurement.

For the lift and drag components to be decoupled, both need to have their own force measuring system that does not feel any interference from the other force. Figure 75 shows the first design of the uniaxial force transmitting strut. As can be seen from Figure 75, the strut design consists of two hinges in the perpendicular direction. A combination of these hinges allows the strut to ‘buckle’ if there is present of any side forces. This is to eliminate any side forces and prevent them from being transmitted to the load cells which would affect the direct load measurement.

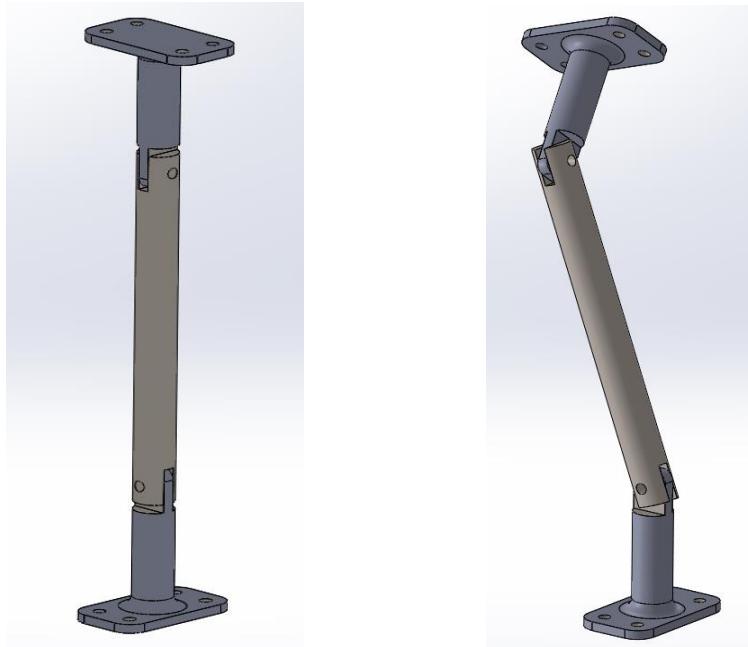


Figure 75 - First design of uniaxial force transmitting strut

However, with further study and investigation, it was found that with the bottom end of the strut fixed, the top end of the strut, which would be linked to the platform, would not remain level if the wing was exposed to side forces. This was understood due to the fact that the hinge movement is only permitted to radial motion in a single plane. If the hinges are under the load in a pure horizontal direction, the strut structure would be forced and stretched especially at the hinge location. This subsequently affects the load measurement. The other problem caused by the deformation is the friction presence between the components at the hinge. Without deformation, the friction at the hinge due to forceless contact could be reduced by using lubricant. However, the friction greatly increases due to the forced contact between components at the hinge and could not be recovered by lubricant, significantly affecting the load measurement.

Hence, this first design is not applicable for the test rig as the wing is required to stay at a fixed angle of attack during the wind tunnel testing despite the aerodynamic forces acting on the wing. One way of solving the existing problem is to add one more hinge in each of the original hinge orientations on the strut. The set of two hinges in the same orientation would allow the two ends of the strut to move in a parallel motion. This same theory can be applied to the other set of hinges. This system would then effectively decouple the vertical force from side forces without any deformation.

7.3.2.2. Improved Strut Design Concept

This strut design is based on the improved idea from previous section with some further modification. Considering for a static two-dimensional wing testing, it is expected that the system is unlikely to be exposed to side forces beside the drag. With this assumption, the force decoupling strut design was simplified from a two degrees of freedom system into one degree of freedom system as shown as in Figure 76.

This improved strut design does not have the problem of forced deformation when exposed to a pure drag force, however it can be very sensitive to the other side forces. Even a little side force could induce a significant amount of friction at the hinge which would subsequently affect the load measurement and hence the accuracy of the readings. The friction at the hinge could be reduced by applying lubricant at the hinge or using material which has self-lubricating characteristics at the hinge. However, the effectiveness of these methods can be inconsistent. In this case, a pillow bearing which can rotate smoothly despite a little side force would be the solution to this problem. Figure 77 shows the pillow bearing and the force decoupling strut system design with the pillow bearing integration.

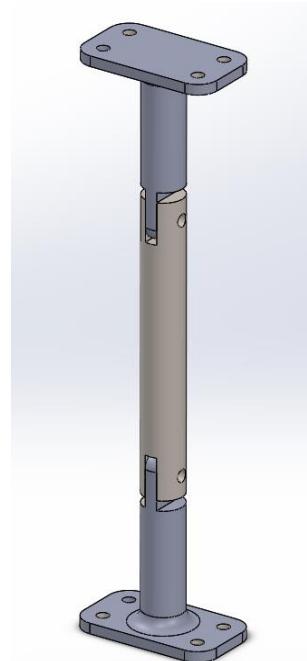


Figure 76 - Improved force decoupling strut design with normal hinges



Figure 77 - Pillow bearing (right) and the force decoupling strut design with the use of pillow bearing (left)

7.3.2.3. Material Selection

The material properties should be taken into consideration during the design stage based on the test rig requirements. Although the finalized design of the force decoupling strut can decouple the combination of forces, the accuracy of the load measurement can be affected by the stiffness of the strut material. As mentioned earlier, the idea is to transmit the aerodynamic load completely to the load cell for measurement, hence the strut should be stiff enough to do so with minimum deformation, at least for the maximum load expected to be experienced during the wind tunnel test. The selection of strut material was primarily focused on the Young's modulus which quantifies the material stiffness by through the relationship between stress and strain. In this case, a stiff material – a material with a high Young's modulus is required for the strut to effectively transmit the load to the load cell.

The most common metals available from EDMC was found to be structural steel and aluminium. Aluminium is normally softer than the steel, with a Young's modulus of around 69 GPa and around 200 GPa for steel. In this case, structural steel which has the higher Young's modulus was first chosen as the strut material. To make sure structural steel strut would be stiff enough for load transmission, finite element analysis was conducted on each strut. The test rig was considered to have three vertical struts to measure lift. The maximum possible lift with the safety factor 1.5 is 60 N , which was distributed over the three struts so each strut would experience 20 N of load. To determine strut's maximum deformation and maximum stress under the maximum possible load during the wind tunnel testing, 20 N was loaded on the top end of the strut in the direction along the strut length with the other end of the strut fixed as shown in Figure 78.

The deformation results in Figure 79 shows that the maximum deformation of the strut is about 0.01 mm . While Figure 80 shows that the maximum stress is about 3.03 MPa and the stress concentration area is located on the beam next to the bearing. The 0.01 mm maximum deformation is considered very small and could be easily considered during the load cell calibration stage. On the other hand, with the steel tensile yield strength of about 250 MPa , the maximum stress of 3.03 MPa is only about 1.2% of the tensile yield strength and it is expected not able to cause permanent deformation on the steel strut or even failure under the interest loads.

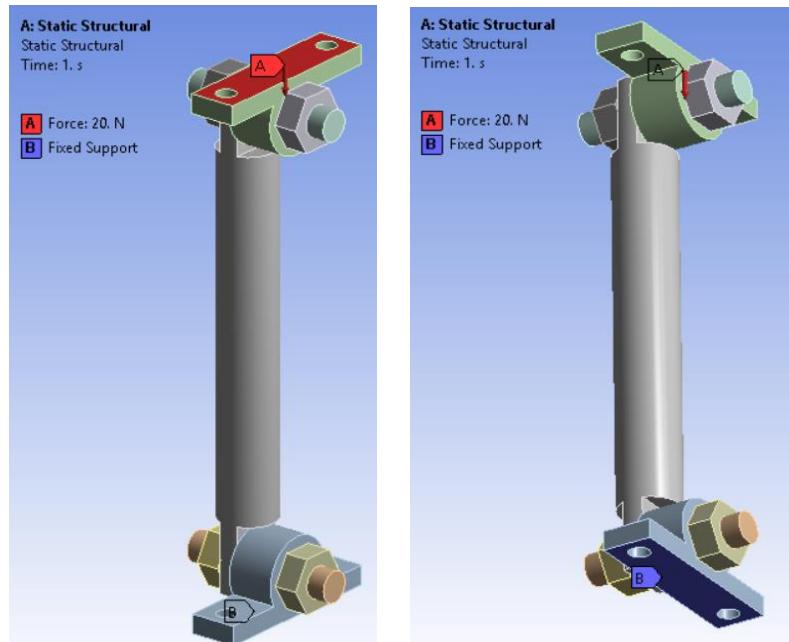


Figure 78 - Setup for the finite element analysis

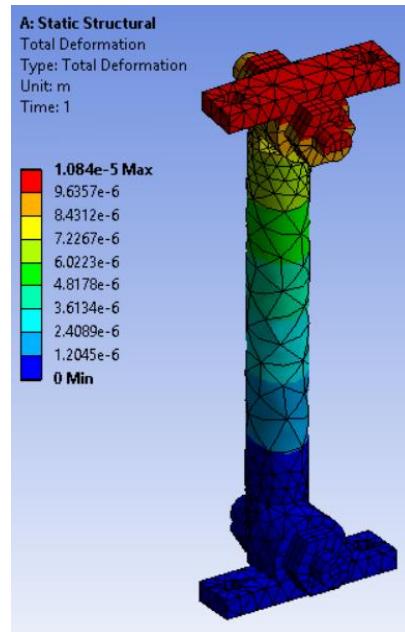


Figure 79 - Total deformation of the strut structure

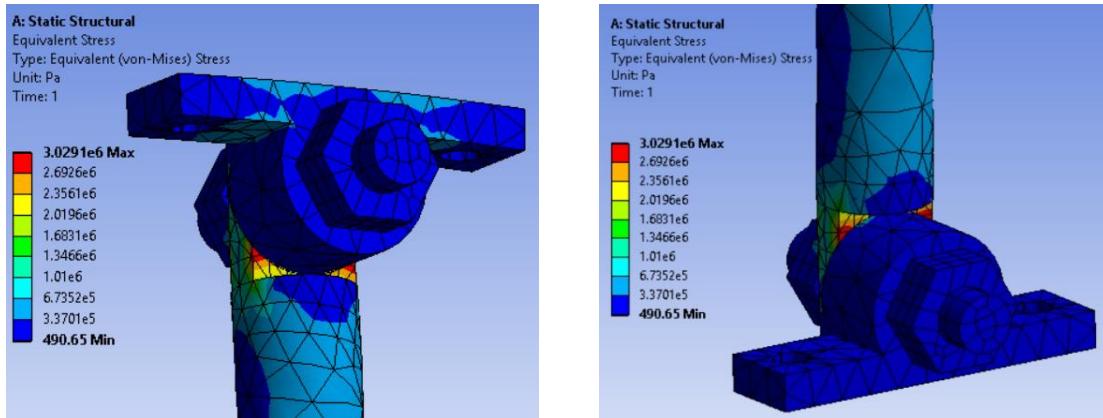


Figure 80 - Stress concentration location on the strut

7.3.2.4. Test rig force decoupling system configuration

The overall configuration for the test rig was influenced by the work of Samardžić et al. in “External Six-Component Strain Gauge Balance for Low Speed Wind Tunnels” [21], however, for this project only lift and drag had to be measured accounting for varying pitching moments. The configuration of test rig was modelled in CAD and shown in Figure 81 and Figure 82 below. It

consists of a platform which allows the test wing model to be mounted on top. The aerodynamic forces acting on the wing could then be transmitting to the platform and be decoupled by the system. The decoupling system located below the upper platform consists of three lift arms (decoupling strut placed vertically) and two drag arms (decoupling strut placed horizontally).

From Figure 82, one end of each strut is connected to the platform while the other end is connected to the load cell via a T-shape 3D printed block which helps to locate the load properly onto the load cell. There is also a small step located below the other end of the load cell to provide space for the bending beam load cell to bend when load is applied. Each arm is linked to the rigid structure for accurate load measurement. The rigid structure for the lift arm system is the base platform; for the drag arm, the rigid structure is the extended structure from the base platform which secure by an L-shape bracket. With the maximum drag with safety factor, half of 5.25 N applied onto each vertical rigid structure, the finite element analysis result shows that the maximum deformation would be only around $9\text{ }\mu\text{m}$, which again can be easily corrected for in the load cell calibration stage.

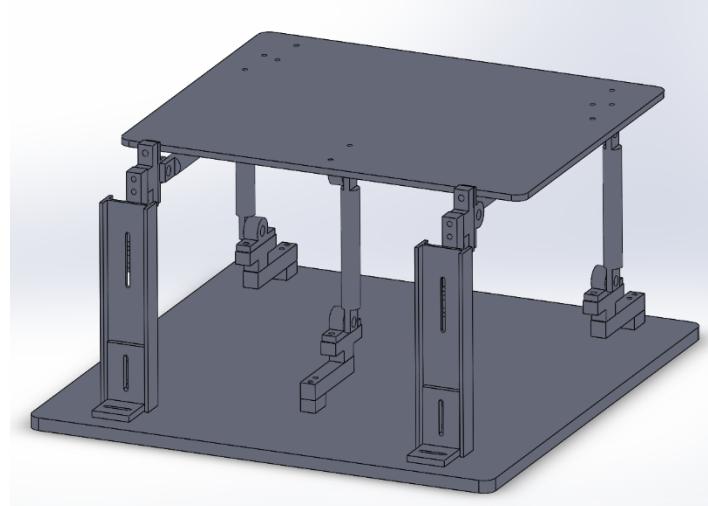


Figure 81 - Overall test rig design configuration

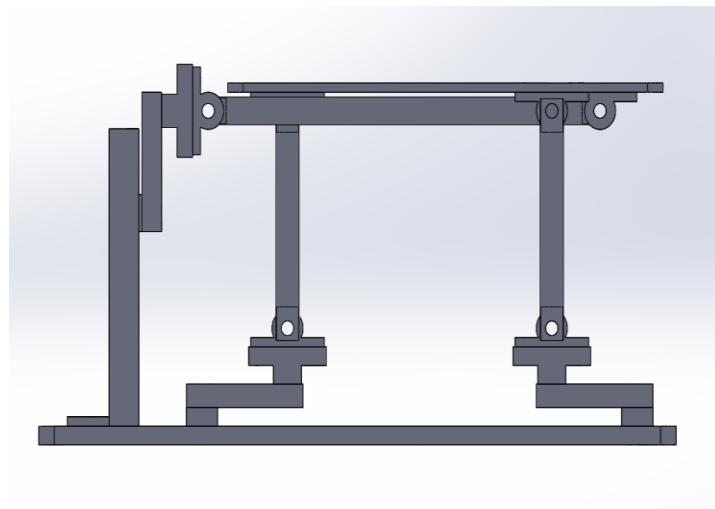


Figure 82 - Overall test rig design configuration (view from side)

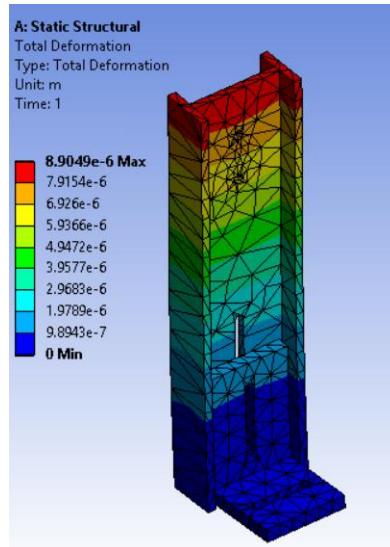


Figure 83 Total deformation for the drag arm system's rigid structure

7.3.3. Test wing section integration

The test wing section was designed to be mounted on the upper platform attached to linear rail beams. This gives the flexibility to any dimension changes on the test wing section and the wing section height; ease of removing the test wing section from the test rig; easy to construct; and it is also a time and cost-effective design compared to other designs which require the manufacture of self-design parts.

Considering the test wing needs to be able to change the angle of attack, servo motors which can give a precise angular position are used to rotate the wing. The axis of rotation was about the axis of the front carbon fibre spar in the test wing model. It was found to be difficult to connect the servo motor's gear directly to the carbon fibre, hence an extra holder was designed to be located in between the servo motor and the test wing section. To have a better picture of the angle of attack system design, it is shown in Figure 84 below, while Figure 85 shows the test wing section to be mounted on the upper platform of the test rig.

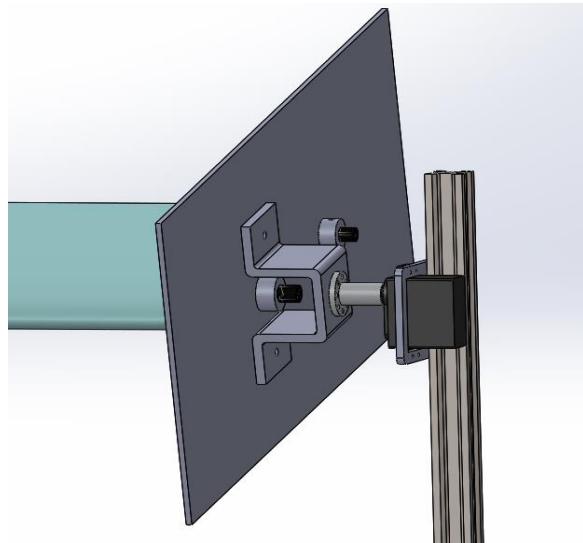


Figure 84 - Angle of attack changing system

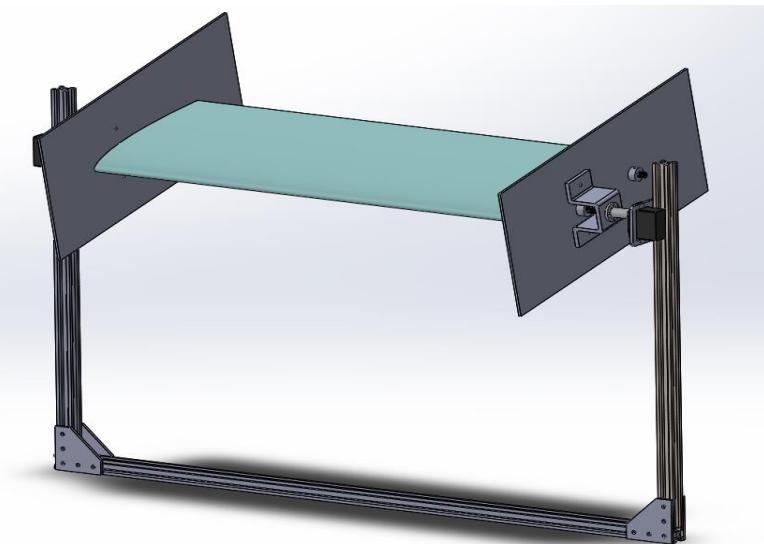


Figure 85 - Test wing mounting configuration

7.3.4. Final Test Rig Model

Full engineering drawings are shown in Appendix B.

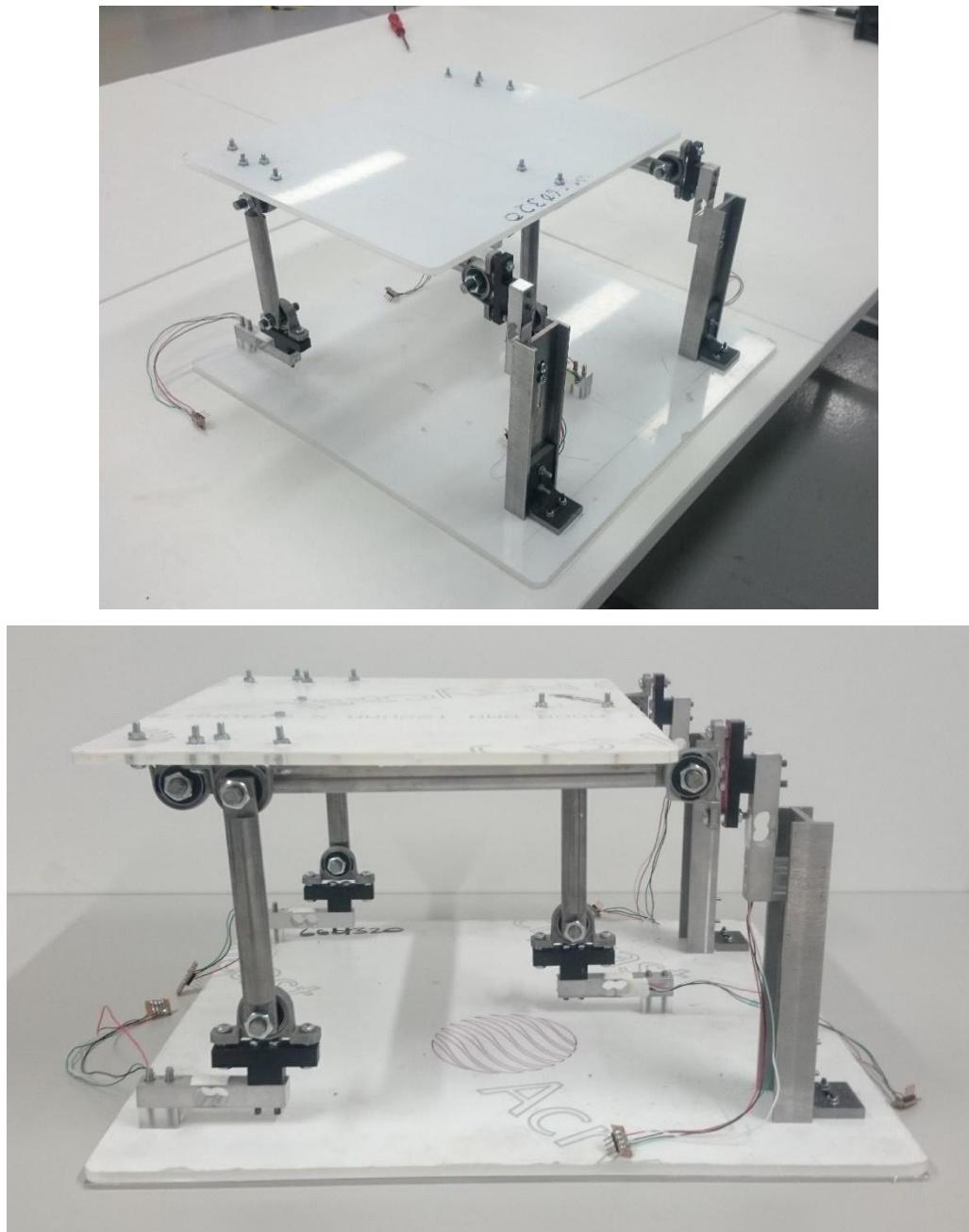


Figure 86 - Images of our final test rig

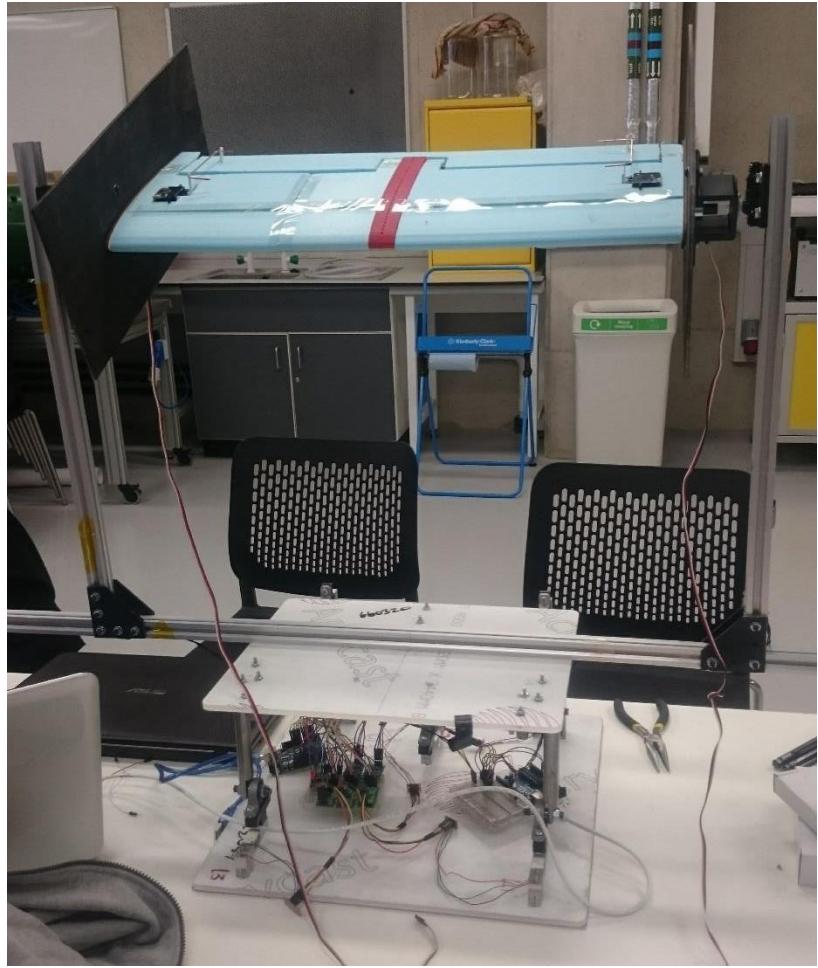


Figure 87 - Image of our final test rig with upper section and wing attached

7.4. Flow Perturbation

The third series of tests in the wind tunnel will consist of getting pressure readings while simulating the 'randomness' of wind gusts while the UAV is flying. A series of flaps were designed to give a good approximation of random wind gusts.

7.4.1. Initial Design

The wind tunnel can only give a stream of air at a constant speed. The wind speed can be changed, but at an insufficient rate to represent air gusts. Thus, an apparatus needed to be designed, that can introduce a 'gust'. Originally the idea was to include the flow perturbation flaps inside the wind tunnel, actuated by a servo on the bottom panel. Some problems that arose with this set-up were that the flaps were too small and could not produce a sufficient turbulence, the tunnel had few areas to secure the rig without it being blown away and that running wires to the servo were too

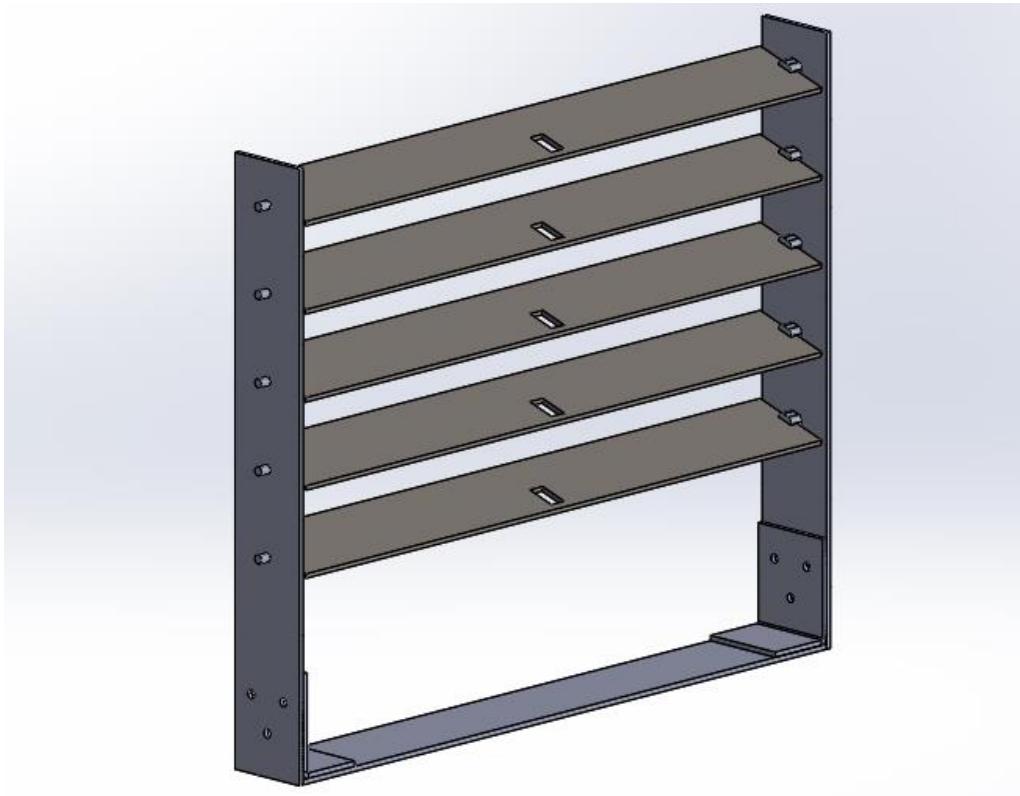


Figure 88 - Initial concept of 'gust' initiator system.

cumbersome. Thus, the idea was determined to be too difficult to implement, the initial design is included below in Figure 88.

7.4.2. Final Design

Due to the size of the test rig it was to be situated outside of the wind tunnel, thus the flaps were designed to be fitted onto the end flange of the tunnel exit. The flap system was to be a modular one, due to the presence of other students using the wind tunnel facility and designed to be easily removed and installed. The designed flaps were $150 \times 583 \text{ mm}$ made from 1 mm sheet steel, with the addition of the two plastic connectors at each end, they would comfortably span the $443 \times 606 \text{ mm}$ wind tunnel exit. Below is a SolidWorks screen capture of the flap system.

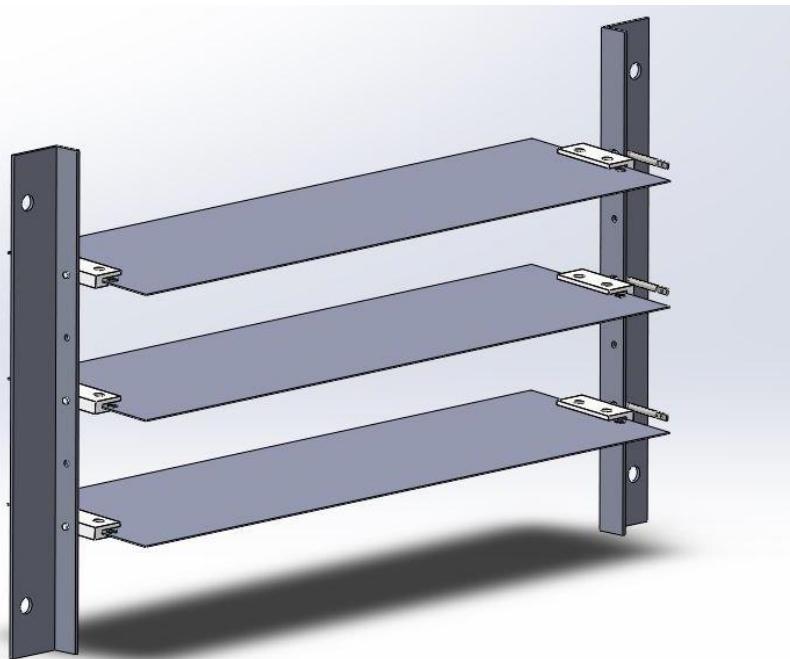


Figure 89 - Isometric view of flap system.

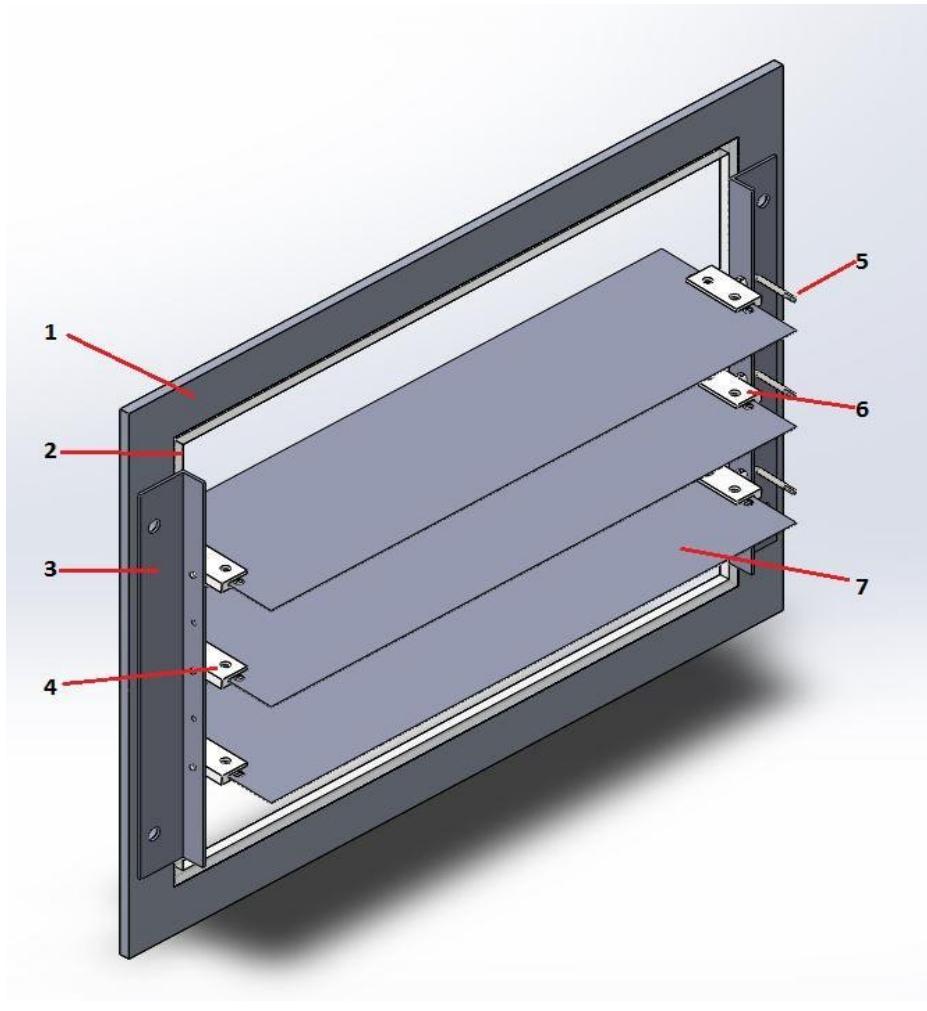


Figure 90 - Flaps mounted on flange of wind tunnel. 1- Flange, 2- Perspex end of tunnel, 3- Flap holder, 4- 3D printed connector, 5- Actuating rod, 6- 3D printed rod connector, 7- Flap.

7.4.3. Manufacturing

Most of the manufacturing was determined to be simple enough to be undertaken by the student and only the connecting rods were to be manufactured by the EDMC, due to the need for a lathe to cut out intricate parts of the geometry.

7.4.4. Finished System

After the components were assembled the flap system was mounted onto the flange. Due to previous fears that the length of the flaps (56.5 cm) was a potential problem. Their weight would cause bending and the flaps could come out of their mountings in the holder. Originally 3 mm thick sheet steel was supposed to be used, but proved infeasible due to the 2.0 kg weight of each flap potentially snapping their individual plastic connectors. Instead a thinner sheet metal was ordered of 1 mm , as the calculated weight was 665 g . When mounted the flaps did experience some bending, but not enough for them to come out of their mountings. See Appendix C for full engineering drawing.

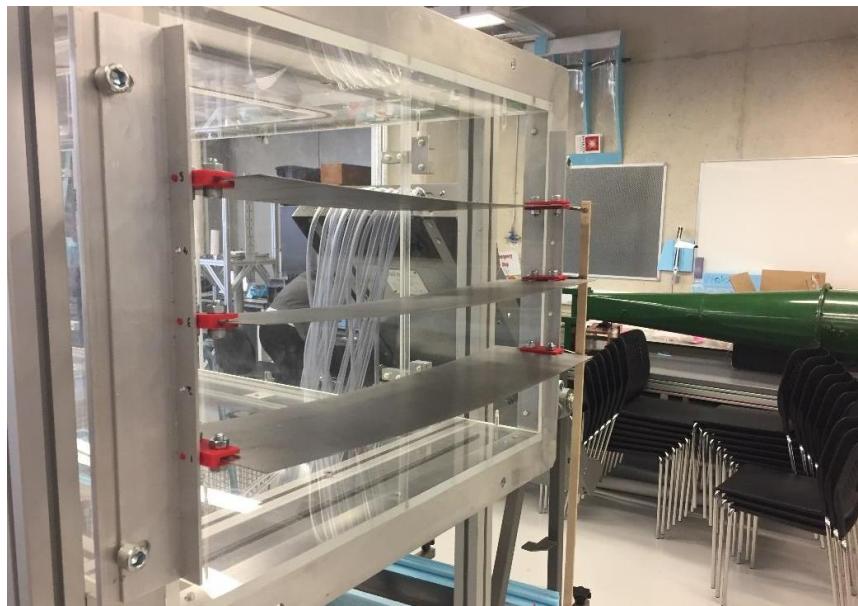


Figure 91 - Flap system mounted on end of wind tunnel.

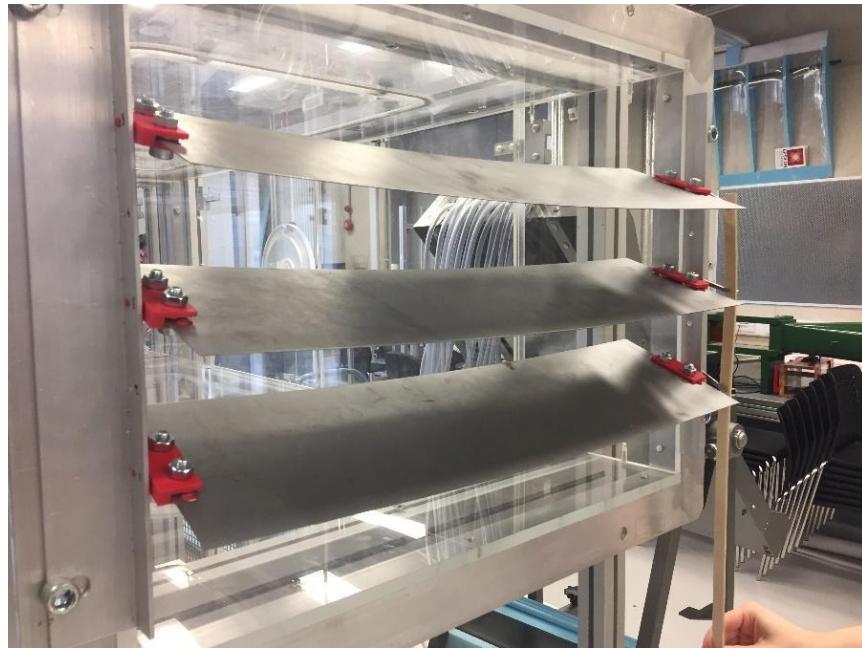


Figure 92 - Flaps actuated by a member of group.

7.4.5. Testing

Due to the wind tunnel being fully booked this final round of tests was not able to be completed for the handing in of the report. 3 days were booked for early May to do further iterations of the tests and collect more readings. These results will appear in the final presentation.

8. Test Rig Calibration

8.1. Single-axis load cell

8.1.1. Load cell specification

Material: Aluminium

Model: YZC-131

Type: Full-bridge

Range: 5 kg

Rated output: 0.1 mV/V

Input Voltage range: 5-10 V

Nonlinearity: 0.03% full-scale

Hysteresis: 0.05% full-scale

Weight: 28 g



Figure 93 - Load cell used in test rig.

8.1.2. Data acquisition

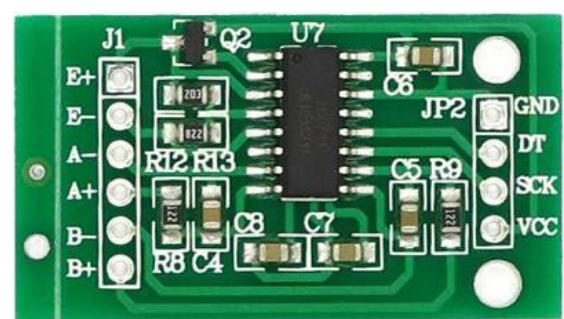
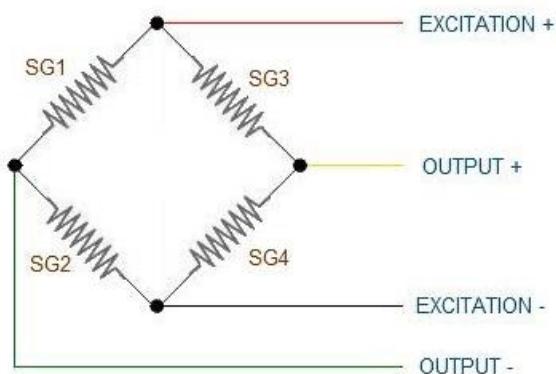


Figure 94 - a) Load cell wiring (left) [28] and b) HX711 chip (right).

To get useful information from the load cells, the output from the bridge must be amplified and converted to a digital signal. The HX711 load cell amplifier was ideal for this. It uses a clock and

a data line to interface with most microcontrollers. The added precision of a 24-bit analogue to digital converter (ADC) resulted in slower data rates (5 Hz) [23].

For higher data rates, an INA125p instrumentation amplifier was considered. The output from the amplifier is fed into the analogue input pins of the Arduino and the on-board 10-bit ADC was used to get a maximum sampling rate of 9600 Hz.

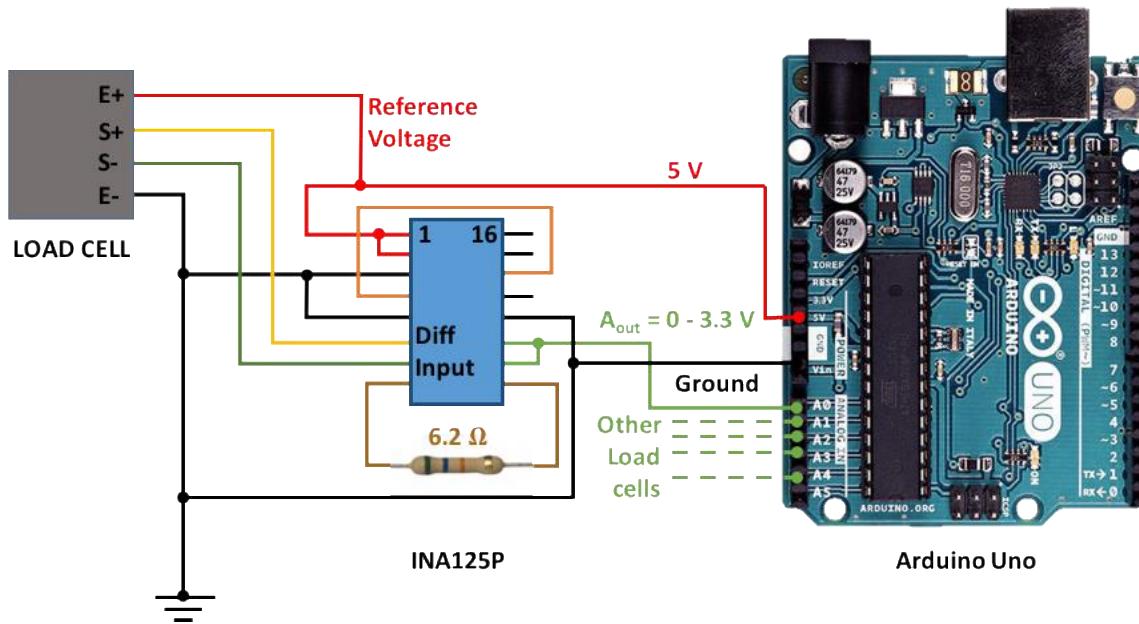


Figure 95: Schematic of data acquisition system using the INA125P instrumentation amplifier.

For this project and more specifically wind tunnel force measurement, the higher accuracy and ease of use of the HX711 chip was preferred.

8.2.Signal Processing

The load cell output primarily consists of high frequency low amplitude noise (High signal to noise ratio). Averaging the output over a fixed duration of time (5 seconds for test rig calibration) greatly mitigated this noise and was the suitable approach for test rig calibration where the loading conditions were not rapidly varying.

For the wind tunnel tests however, an exponential moving average filter was used which takes a weighted average of the last few input values.

Exponential Moving Average

$$y_t = y_{t-\Delta t} + (x_t - y_{t-\Delta t}) \cdot r \quad (27)$$

where:

y_t = Filter output at time t

x_t = Filter input (load cell output) at time t

Δt = time period between readings = $\frac{1}{\text{sample rate (Hz)}}$

r = roughness parameter ($0 < r < 1$)

For $r = 1$, equation (27) simplifies to $y_t = x_t$ where the filter input is equated to the filter output i.e. no filtering is performed. Conversely, for $r = 0$, equation (27) simplifies to $y_t = y_{t-\Delta t}$ where the filter input is neglected.

A low roughness parameter (r) results in a filter which greatly reduces noise at the cost of responsiveness. Thus, an appropriate roughness parameter was needed which considered the dynamic nature of the signal. The roughness parameter is determined

$$r = 2 \cdot \left(1 - \frac{1}{(1 + diff)} \right) \quad (28)$$

where:

$diff = c \cdot (x_t - y_{t-\Delta t})$

c is chosen such that the value of r is capped at 1.

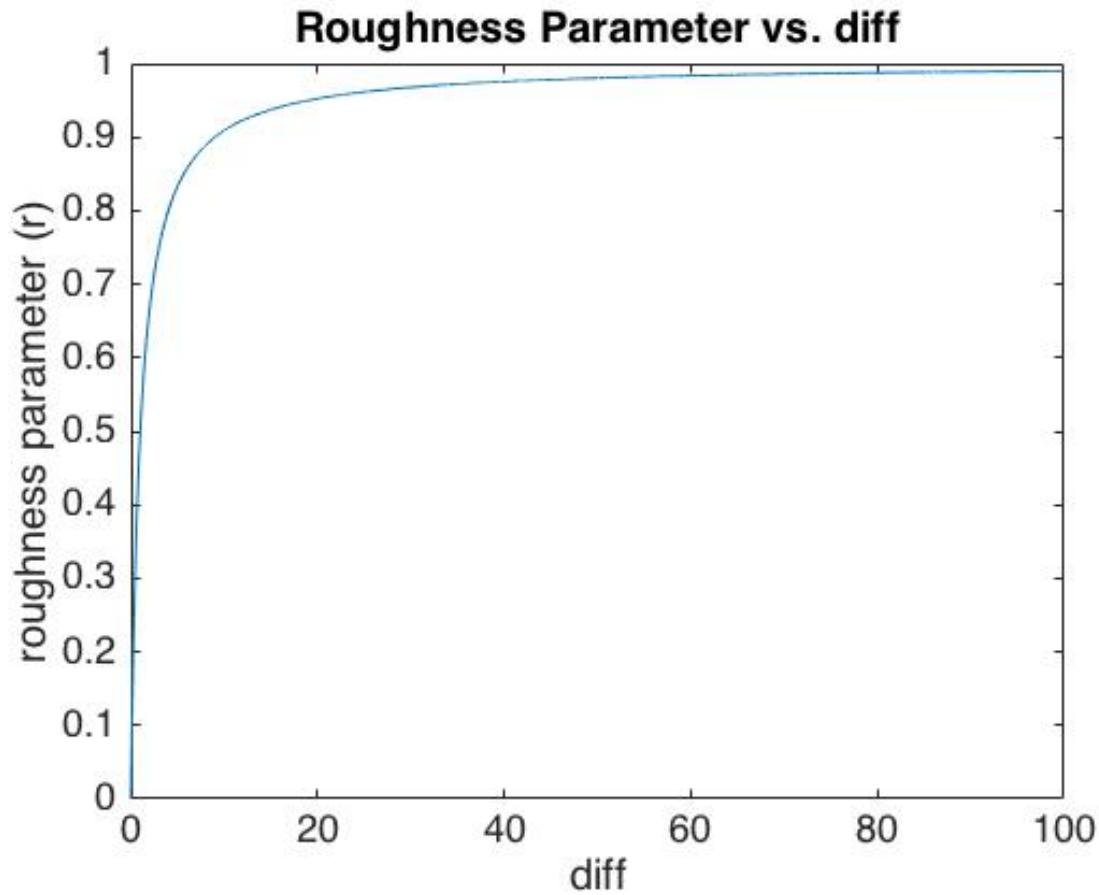


Figure 96 - Roughness parameter vs diff from equation (17).

Thus, the weighting is performed by the roughness parameter for a certain reading. Thus, there is a non-linear filter which can mitigate low magnitude noise.

However, as seen in Figure 96, the roughness parameter rapidly increases with an increase in magnitude of noise. For higher magnitude noises, the exponential moving average filter is applied in conjunction with median filtering.

When the filter encounters an input value (x_t) for which the resulting value of $diff$, passes a certain threshold, the filter determines if the input is the signal or noise using median filtering. Four subsequent values of x are recorded in a list $x_{t + \Delta t}, x_{t + 2\Delta t}, x_{t + 3\Delta t}, x_{t + 4\Delta t}$ and the median of these values is fed as input to the moving average filter. If the input was noise, it would be the last value of the ordered list. Otherwise if the input was the desired signal, it would be recorded as the median value as the signal value increases. This approach works since, by observation, it is known that high amplitude noise occurs very rarely in the output signal. See Appendix G for code.

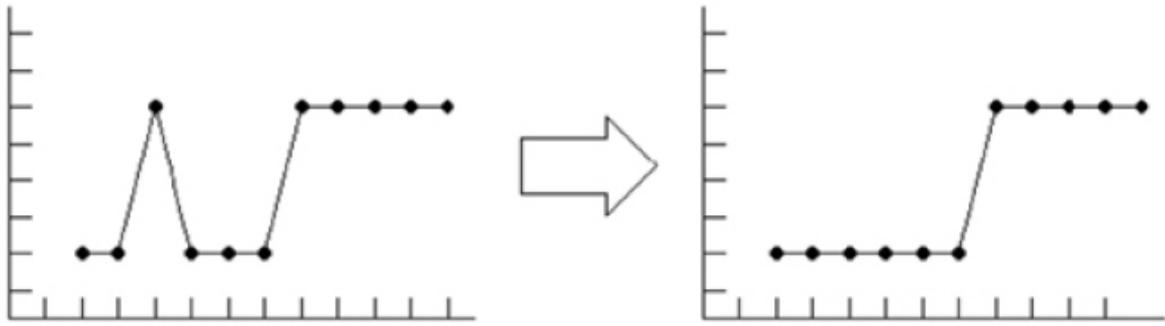


Figure 97 - A simplified representation of the application of a media filter [24]

8.3. Calibration Theory

The majority of errors induced by the load cell behaviour are approximated out using a comprehensive calibration procedure. The output of the five load cells is used to estimate the required forces (f = Lift, drag, pitching moment) using the equation

$$f_i = \sum_{j=1}^6 r_j c_{j,i} + \epsilon_i \quad (29)$$

where:

i represents the force component number ($i = 1, 2, 3$)

j represents the transducer number ($j = 1, 2, \dots, 6$)

r_j is the j th transducer's reading.

$c_{j,i}$ is the calibration coefficient for the j th transducer for the i th force component,

ϵ_i is a random variable which indicates the measurement error. The formulation involves 6 transducers although there are only 5 load cells because the first transducer ($j = 1$) is a dummy transducer (r_1 is always equal to 1) used to set the offset of the test rig – the tare weight.

The calibration coefficients ($c_{j,i}$) are estimated by the calibration procedure. Equation (29), is a first order approximation, hence the value of the calibration's coefficients are not necessarily constant. Thus, the calibration procedure must be carried out in loading conditions which are identical or very similar to the actual wind tunnel testing conditions [25].

The calibration coefficients are found to minimise the sum of squared residuals (s_i) – A weighted least squares method based on the work of M. Ferreira [26].

$$s_i = \sum_{p=1}^N w_p \cdot (y_{p,i} - \hat{y}_{p,i})^2 \quad (30)$$

where:

p is the sample number with a total of N samples.

w_p is the weight assigned to the p^{th} sample.

$y_{p,i}$ is the load cell reading for the p^{th} sample and i force component

$\hat{y}_{p,i}$ is the calibration estimate of the output for the p^{th} sample and i force component.

Equating the partial derivative of each s_i with respect to each coefficient gives 6 linear equations and 6 unknowns for each i force component. In matrix form equation (29) is written as

$$\vec{Y} = R\vec{C} + \vec{\epsilon} \quad (31)$$

taking the differential, and accounting for the weighting matrix W gives:

$$\nabla_{\vec{C}} \left[(\vec{Y} - R\vec{C})^T W (\vec{Y} - R\vec{C}) \right] = \vec{0} \quad (32)$$

Further simplification gives:

$$\hat{C} = (R^T W R)^{-1} R^T W Y \quad (33)$$

Where R is the $N \times 6$ matrix of load cell readings with the p^{th} row representing the readings for the p^{th} loading combination/sample;

\hat{C} is the matrix of calibration estimates of the coefficients [26].

The weighting matrix is calculated as

$$W = (V_w + DV_R D^T)^{-1} \quad (34)$$

The primary purpose of the weighting scheme is to emphasize reliable data while estimating calibration coefficients as even the measurement process used in the calibration has is susceptible to error. The two sources of error considered while choosing the weighting matrix were the error in the mass of the reference loads used for calibrations (V_w) and the uncertainty induced by the load cell properties and error in bridge readings of the load cells (V_R).

V_w is a diagonal matrix of size $N \times N$ where the diagonal elements were estimated by first taking V_w as an identity matrix and estimating the standard deviation for each force component i i.e.

$$C_I = (R^T I R)^{-1} R^T I Y \quad (35)$$

$$\sigma_i^2 = \frac{1}{N-6} \sum_{p=1}^N (y_{p,i} - \hat{y}_{p,i})^2 \quad (36)$$

$$W_{p,p} = 0.7 \cdot \sigma_i^2 \quad (37)$$

The V_R is measured $6N \times 6N$ matrix estimated by taking the covariance of the readings over multiple calibration processes. This was done in Matlab using the cov() function. The D matrix is a $N \times 6N$ matrix whose elements represent the sensitivity coefficients of taking partial derivatives of equation (29). The value of these elements are calculated iteratively as shown in the MatLab function – calc_CW.m below (See Appendix G for smoothingSpline.m script).

```

function [ C_mat, std_dev ] = calc_CW(R, Y, C_prev, Vr)
    %% Solve using previous iteration data
    N = size(R,1);
    I = eye(N);
    D = zeros(N, 6 * N);
    err = Y - (R * C_prev);
    std_dev = sqrt([err(:,1)'*err(:,1)+1  err(:,2)'*err(:,2) ...
                    err(:,3)'*err(:,3)]./(N - 6));
    C_mat = zeros(6, 3);
    for force = 1:1:3
        for p = 1:1:N
            for j = 1:1:6
                D(p,((j - 1)*N) + p) = C_prev(j,force);
            end
        end
        %estimate Vw using previous std_dev
        Vw = (I * 0.7 * std_dev(force));
        W = inv(Vw + (D * Vr * D'));
        C_mat(:,force) = (R' * W * R)\(R' * W * Y(:, force));
        error_temp = Y(:,force) - (R * C_mat(:,force));
        %calculate new std_dev for each 'force'
        std_dev(force) = sqrt(error_temp' * error_temp);
    end
end

```

Figure 98 - calc_CW.m

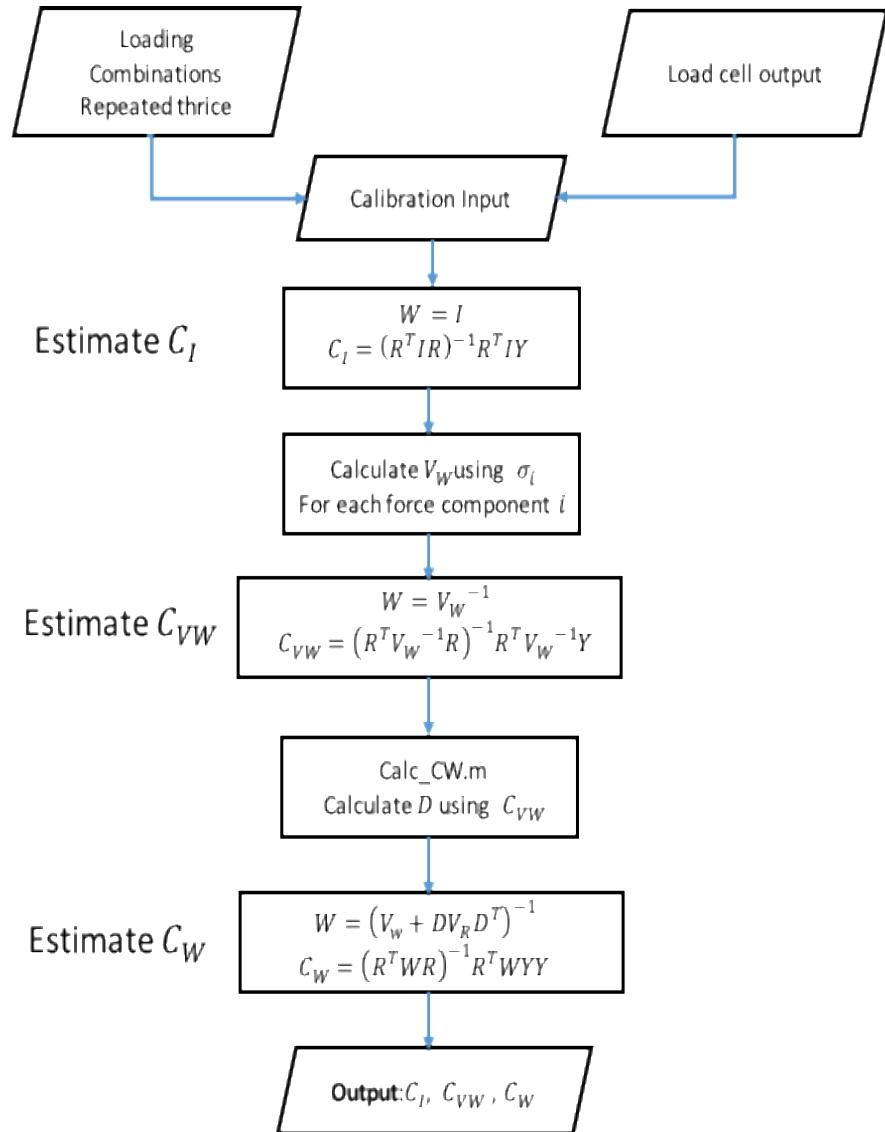


Figure 99 - Theoretical workflow of the calibration process [26].

8.3.1. Data Acquisition for Calibration

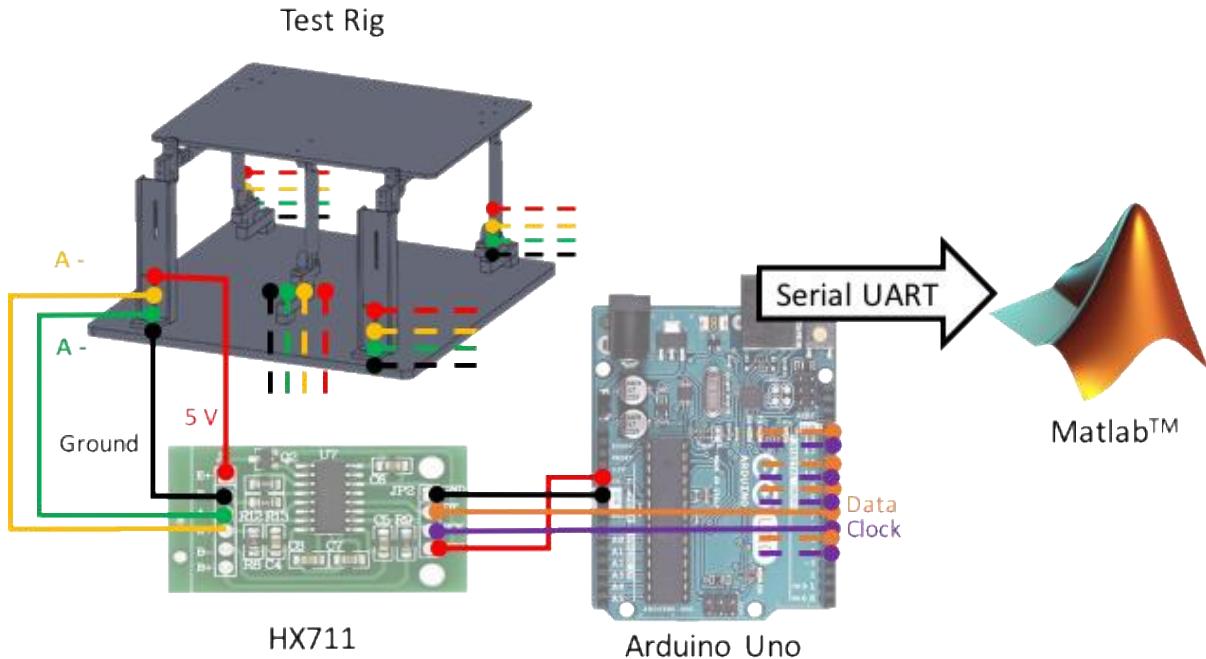


Figure 100 - Data flow during calibration

The data acquisition process is similar for both wind tunnel testing and calibration. Each load cell reading was averaged over a time period of 6 seconds and load cell data was transmitted sequential from the Arduino to MatLab over serial.

8.4. Calibration Procedure

The test rig was calibrated with combinations of the following lift, drag and pitching moments. (See Appendix G for full table of permutations)

Lift (in N): 0, 19.62, 39.24, 58.86

Drag (in N): 0, 1.962, 6.867, 9.81

Pitching (in Nm): -0.5, 0, 0.5

These forces were chosen based on estimates of forces acting on the wing section during testing and are applied using a pulley mechanism as shown in Figure 101 and Figure 102.

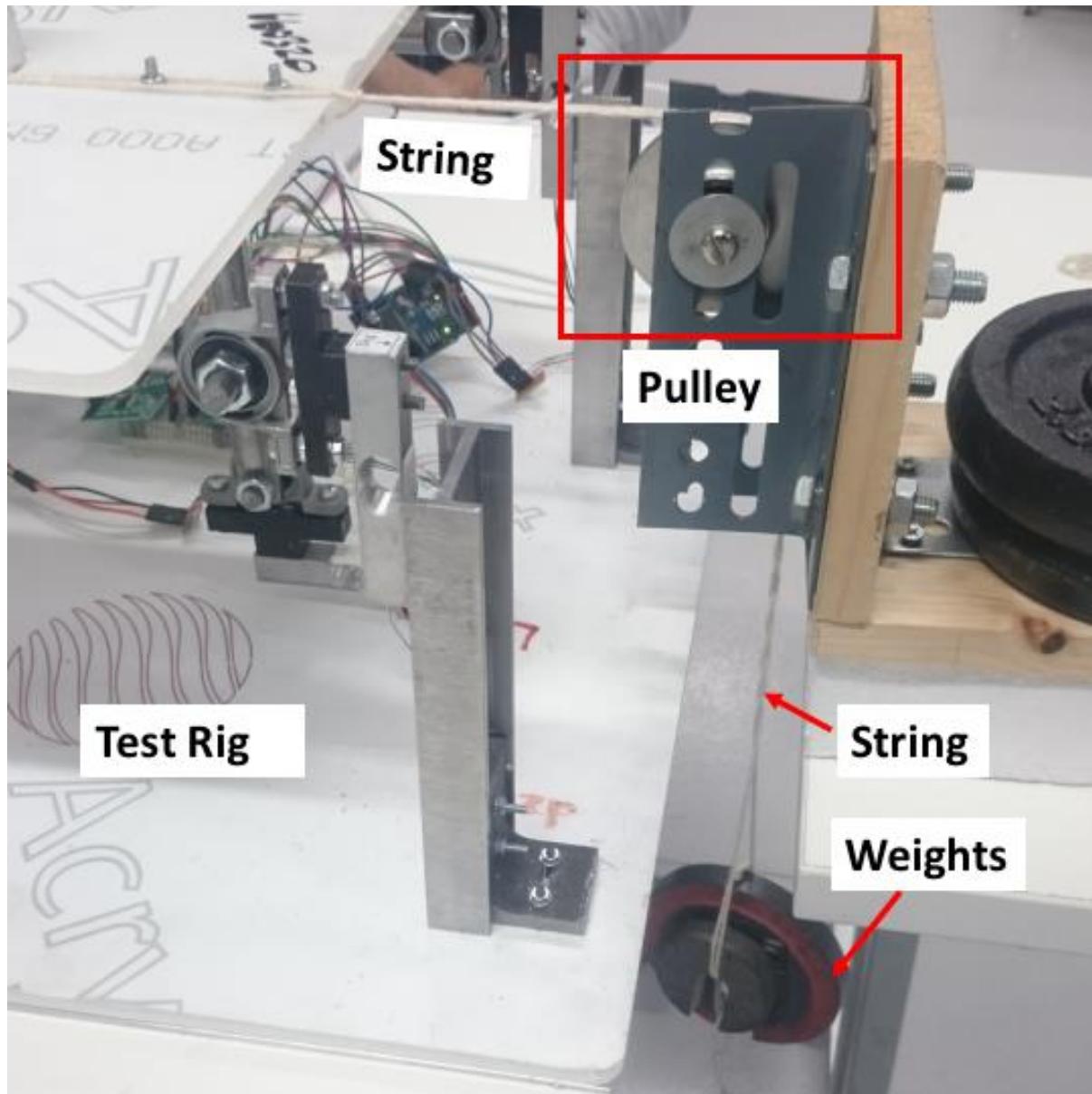


Figure 101 - Side view of pulley mechanism for drag load application.

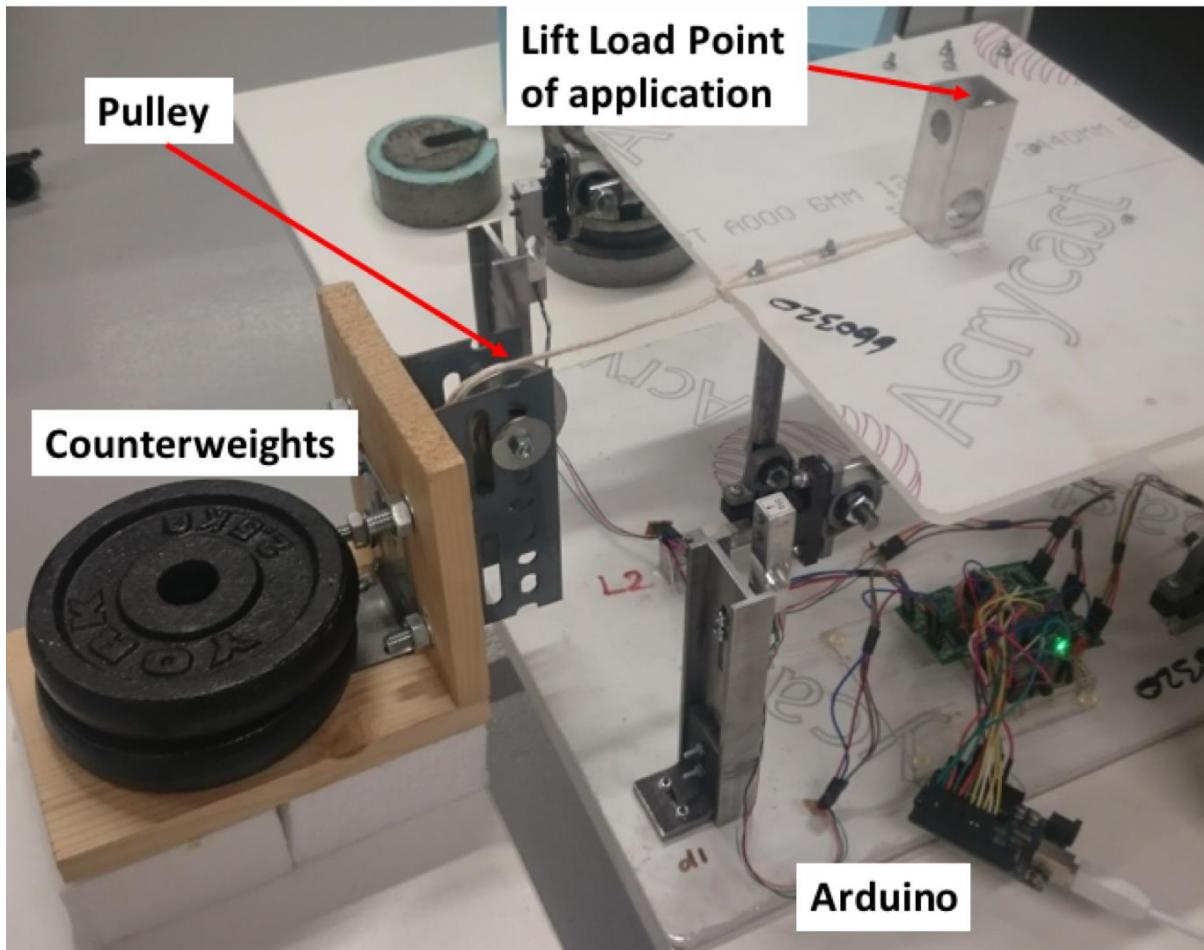


Figure 102 - Top view of mechanism for applying lift and drag forces

The required loading condition is displayed in the MatLab command window, following which users have to place the prescribed weights. To apply a drag force, weights are hung on the string shown in Figure 101. The wing section is mounted upside down on the test rig, hence the positive lift axis is in the downward direction. During the calibration, weights are placed on the top surface of the test rig at predefined points thus producing a pitching moment as well. Once the weights are placed, the user toggles a switch to begin data acquisition from the load cells and subsequent transfer from the Arduino.

8.5. Calibration Results

8.5.1. C_I Calibration results

The following results were obtained by carrying out the calibration without any weighting as described in the sections above.

Table 15 - C_I Coefficient Matrix for measuring the load applied in grams

Force Transducer \ Force	Lift Component	Drag Component	Pitching moment component
Offset	-1322.67	-1906.94	-542.346
Lift 1	-0.00243	-0.00006	-0.00350
Lift 2	-0.00232	-0.094e-5	0.00201
Lift 3	-0.00218	0.129e-5	-29.8e-5
Drag 1	-0.687e-5	-0.00233	76.71e-5
Drag 2	0.000278	-0.00218	-0.00254

Table 16 - Total root-mean-square error (in N) using C_I matrix for estimating the 40 loading conditions used for calibration

Force component	Lift	Drag	Pitching Moment
Total RMS Error (in N)	0.116	0.073	0.958

8.5.2. C_{vw} Calibration results

The C_{vw} matrix was constructed using the C_I matrix as shown in Figure 99. The weighting scheme has no impact on the results.

Table 17 - CvW Coefficient Matrix for measuring the load applied in grams

Force Transducer	Lift Component	Drag Component	Pitching moment component
Offset	-1322.67	-1906.94	-542.346
Lift 1	-0.00243	-0.00006	-0.00350
Lift 2	-0.00232	-0.094e-5	0.00201
Lift 3	-0.00218	0.129e-5	-29.8e-5
Drag 1	-0.687e-5	-0.00233	76.71e-5
Drag 2	0.000278	-0.00218	-0.00254

Table 18 - Total root-mean-square error (in N) using CvW matrix for estimating the 40 loading conditions used for calibration

Force component	Lift	Drag	Pitching Moment
Total RMS Error (in N)	0.116	0.073	0.958

8.5.3. Cw1 Calibration results

The Cw1 matrix was constructed using the CvW matrix as shown in Figure 99.

Table 19 - Cw1 Coefficient Matrix for measuring the load applied in grams

Force Transducer	Lift Component	Drag Component	Pitching moment component
Offset	-1324.61	-1923.884	-697.4948
Lift 1	-0.00243	-5.7748e-5	-0.003202
Lift 2	-0.00231	0.026e-7	0.002022
Lift 3	-0.00218	-0.520e-5	-72.897e-5
Drag 1	-10.58e-5	-0.00231	83.019e-5
Drag 2	33.95e-5	-0.002249	-0.00259

Table 20 - Total root-mean-square error (in N) using C_{W1} matrix for estimating the 40 loading conditions used for calibration

Force component	Lift	Drag	Pitching Moment
Total RMS Error (in N)	0.686	0.458	6.176

The increase in error is expected as the weighting scheme has minimised the influence of the acquired data for certain loading conditions. Co-variances of measurements for pitching moment were particularly high (due to a combination of load placement error, friction in the test rig, etc.) resulting in higher root-mean-square for the pitching moment. This indicates that the system cannot be relied upon for accurately measuring pitching moment which is acceptable.

Using the C_{W1} matrix, lift forces of 9.81 N and 29.43 N were estimated with an accuracy of 1% and drag forces of 2.45 N and 7.848 N were estimated with an accuracy of 3%.

9. UAV Test Bed Development

9.1. Fuselage construction

This section details the steps undertaken in the construction of the Bormatec explorer fuselage. The assembly came with the following items:

- EPP foam UAV fuselage top section with pre-glued of fluted polypropylene board to act as a hinge for the 3 top sections.
- EPP foam UAV fuselage bottom section.
- 2 EPP foam nose cone sections.
- Landing gear frame, wheels and preassembled front wheel servo.
- 6 cut fluted polypropylene boards with 6 cut grass trimmer lines for the closing of UAV.
- EPP foam left and right wings that were not used.
- Centre wing section, where left and right wings are attached.
- Horizontal stabiliser, with two vertical stabilisers.
- 4 plastic screws for the attachment of centre wing.
- Bag of servo connecting arms and control horns.



Figure 103 - Before building, the group members constructing the fuselage set out all parts to visualise the building process.

Below are the steps undertaken in the construction of the fuselage:

1. Top and bottom sections of the fuselage were glued together using epoxy. All subsequent gluing was completed with epoxy.
2. Gluing of all 6 fluted polypropylene board connectors to the sides of the fuselage. Constant pressure was applied to each to ensure a strong bond. The grass trimmer line was then threaded through the fluted holes, securing the top and bottom sections of fuselage.



Figure 104 - Close up of front cut board after Epoxy has set.

3. The horizontal stabiliser was measured to find the centre and glued into the correct place in the back of the fuselage. The two vertical stabilisers were then glued to the ends of the horizontal stabiliser and screwed into place.
4. A hole was cut into the board at the end for the elevator servo wires and the Wi-Fi antenna.
5. A recess in the underside of horizontal stabiliser was cut out for the elevator servo and it was secured into place with a metal bracket, control horn was glued onto the elevator flap

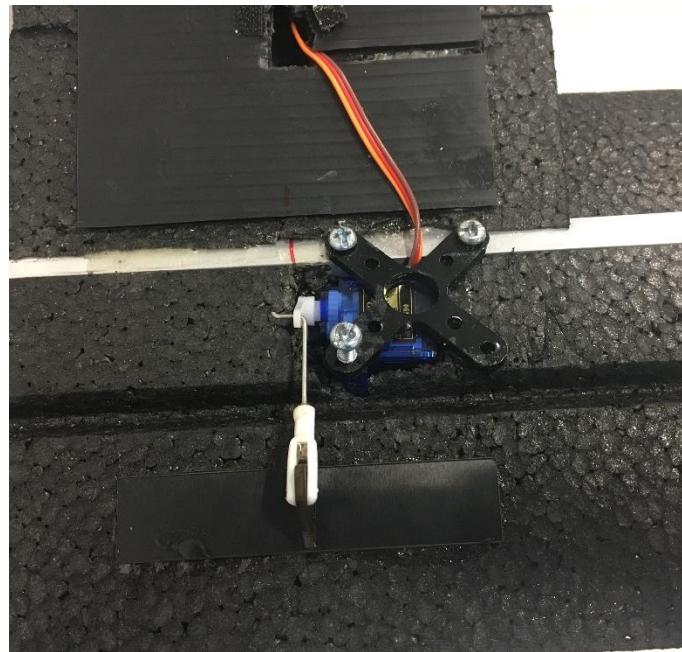


Figure 106 - Underside of horizontal stabilizer with steps 4 and 5 completed

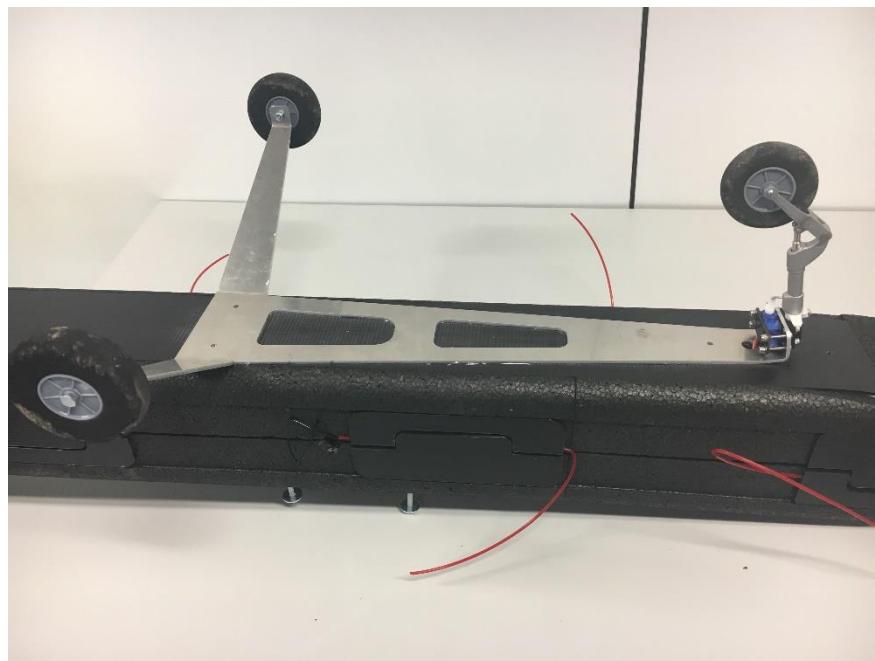


Figure 105 - Landing gear glued into place with wires of servo threaded through hole in undercarriage.

The bottom of fuselage was sanded and the landing gear was glued into place. After the epoxy set a hole was drilled through the EPP to allow easy access for the wires of the steering wheel's servo.

6. The hole on the top of the fuselage was widened to allow more space for the wires leading to the wings.



Figure 107 - Widened access for all wing wires in top of fuselage.

7. A hole was cut in the right side of the fuselage to allow for the webcam to see the ring wing flexing in flight.



Figure 108 - Hole for the webcam before it was installed.

8. The model of brushless motors did not allow for the reversal of the centre shaft, thus two channels in the centre wing were cut out behind the two motor mounting points, to clear space for the shafts.



Figure 109 - Motor shaft channel in right half of centre wing.

9. Next the centre wing was screwed into place after all connecting wires were pulled through. The wires were of the correct length to end just at the junction of the centre section and the wings.



Figure 110 - The Completed UAV fuselage before all control system and all internal wiring were installed.

9.2. Avionics purpose and requirements

Requirements for the UAV smart-wing flight system:

- Provide suitable outputs for stable flight
- Provide data logging for post-flight analysis
- Allow for stability control if we choose to implement it
- Allow for safely arming and disarming motors

Several different flight controllers were carefully considered – the two most suitable options were a Pixhawk px4 flight controller and an ‘in house’ Arduino hooked up with flight controller modules. Both systems would cost the same amount of money, however the Arduino would require more time for programming and setting up. The requirement for stability control, or different flight modes was an option that is not easily deployable on an Arduino and would require significantly more time spent on fine-tuning those parameters. Furthermore, the Pixhawk has these options ready to implement with very easy setup. Another considering factor pointing towards Pixhawk implementation was the fact that our supervisor had a spare Pixhawk set-up from a previous project that had no longer use for such a system. This allowed us to utilise the Pixhawk system into our own UAV model and saved us around £140 for a new set-up. A pulse position modulation (PPM) receiver module can be used with both systems to convert pulse width modulation (PWM) signals for use with the Pixhawk and Arduino.

9.2.1. Flight controller inventory

Items acquired from recycled projects:

- 1 x Pixhawk px4 flight controller
- 1 x GPS compass module
- 1 x power module with xt60 plugs
- 1 x buzzer
- 1 x 433 MHz radio telemetry system, comprising two receiver/transmitter modules with only one antenna supplied
- 1 x Spektrum DX8 transmitter

- 1 x 3s lithium polymer battery
- 10 x servo wires and 2 x y-cables

Items acquired from purchasing:

- 2 x 920 kV motors for propellers
- 2 x 50 A ESCs with 5 V BEC
- 1 x antenna for 433 MHz radio
- 1 x Spektrum PPM receiver module with satellite receiver
- 8 x 9 g mini servos

9.2.2. UAV mode set-up

The Bormatec fuselage comes with an elevator flap in the rear wing, and a choice to implement dual rudders on the yellow tail fins. However, the choice to fly without rudders was taken as this frees up some space on the Pixhawk servo rail and turning in flight can be achieved by solely rolling the aircraft. As for the main wings, a choice between flaps and ailerons, or flaperons were considered. The set-up for systems with flaperons included and not included are shown below.

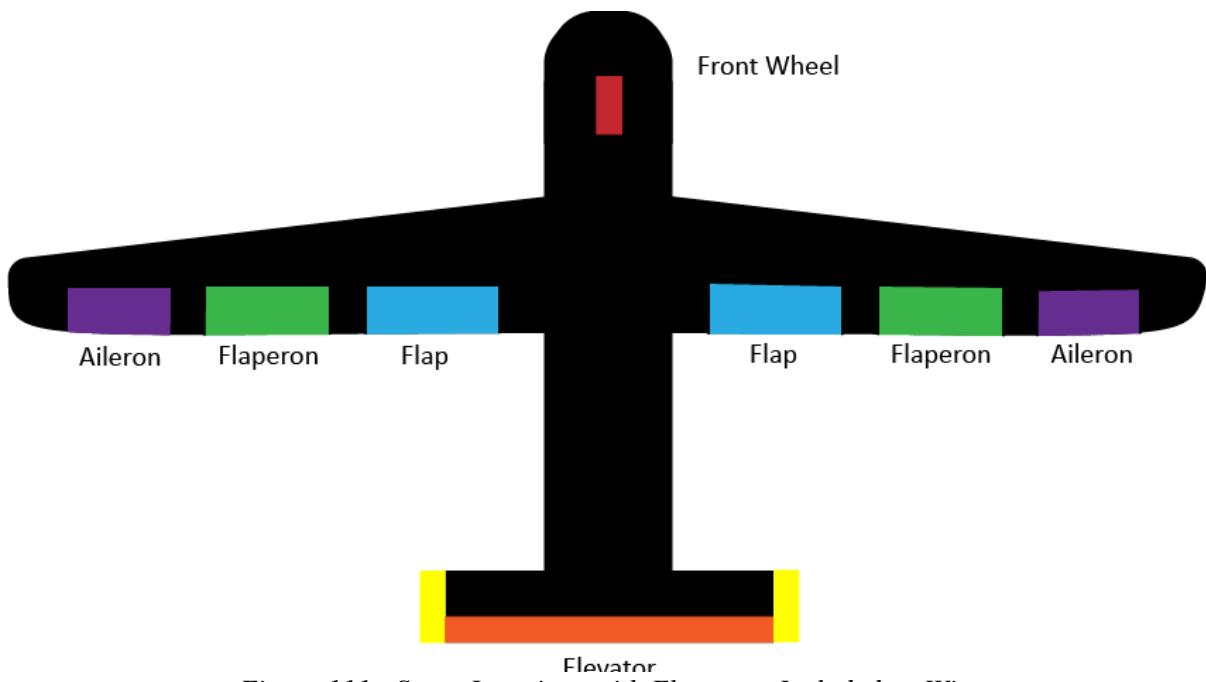


Figure 111 - Servo Locations with Flaperons Included on Wing

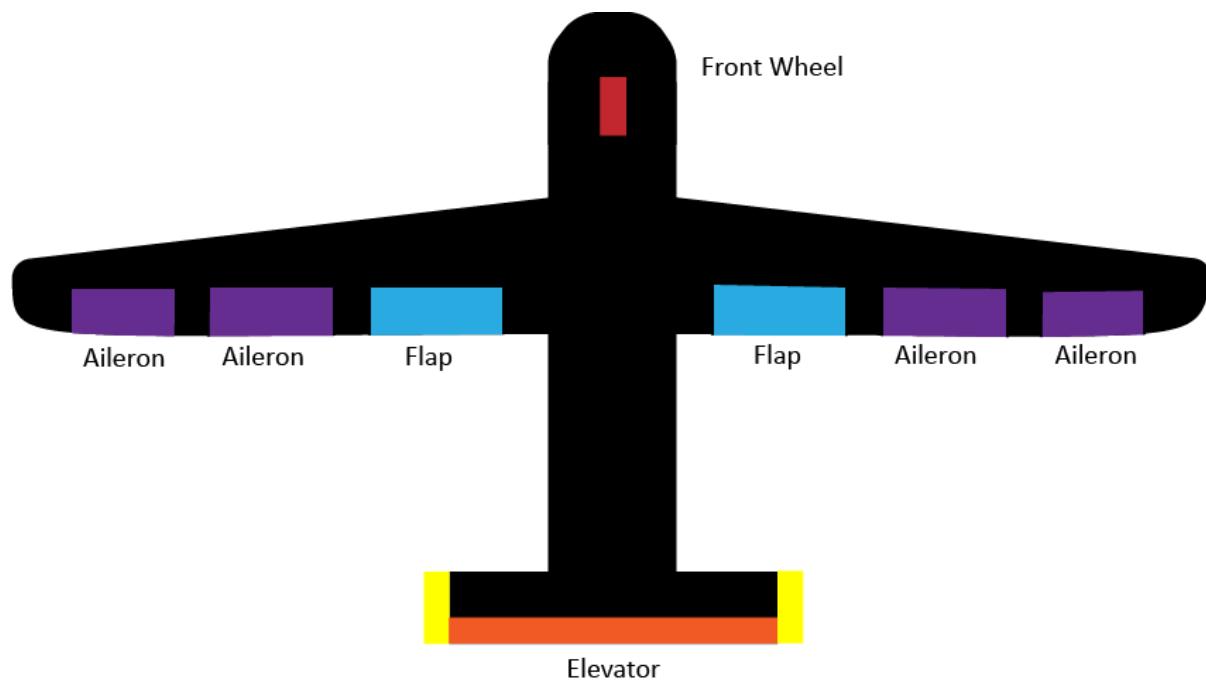


Figure 112 - Servo Locations with Dual Ailerons

Having dual ailerons would be beneficial for our system as we need to increase our roll characteristics due to the lack of a rudder. Furthermore, the use of flaperons would only be justified for a heavier aircraft which needs to implement more flaps for take-off. Finally, the fact that the servos point outwards mean that the same aileron on each wing can be programmed by one servo rail. Each servo will receive the same PWM signal and would rotate in opposite directions allowing for better control of rolling the aircraft.

9.3.Flight controller set-up

The Spektrum transmitter supplied has PPM compatibility which allows all channels from the transmitter to be fed into a single RC-in channel on the Pixhawk. This is bound to the transmitter by using a bind plug and following the bind instructions specific for the dx8 transmitter. The rc-in module can then be replaced by a lemon RX DSMX compatible satellite receiver than plugs into a special port on the Pixhawk. This also accepts the PPM signals and has greater range than the main RC receiver. This is then handled by the Pixhawk's on board 168 MHz arm cortex-m4f core processor which can control up to 14 servos via the servo rail. Distribution of these channels is programmed via the mission planner software.

9.3.1. Physical set-up

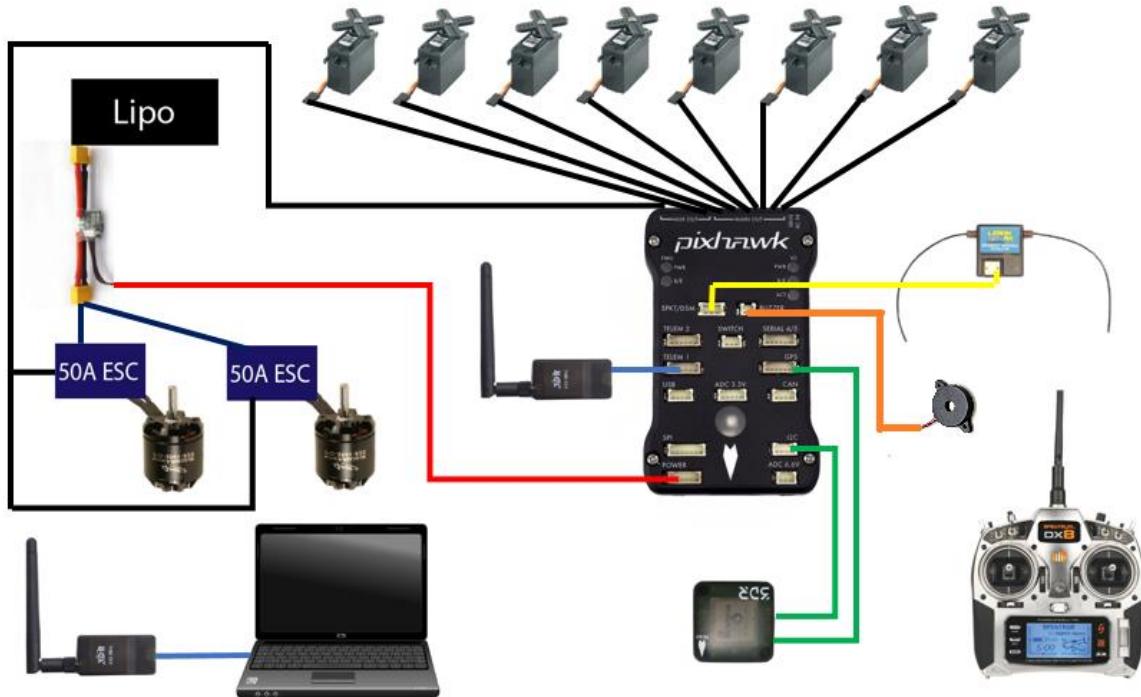


Figure 113 - Depiction of Pixhawk and Components Wiring Schematic

The Pixhawk is powered via a power module that draws a steady and safe current from the lithium polymer battery. The battery is also connected to two 50 A electronic speed controllers which provide PWM signals for the motors. The speed controllers are connected to the Pixhawk servo output rail via a y-cable as they should receive the same input. The Pixhawk has a GPS sensor connected to it via the GPS port and I²C port and the satellite is connected via the DSM port. The two-way radio link is set up by plugging one module into the telemetry 1 port on the Pixhawk and the other into a laptop via micro-USB. Eight servos are distributed in the servo output rail; elevator, front wheel, four ailerons and two flaps. A buzzer is also connected to communicate certain configurations with tones to the user.

The Pixhawk does not have enough power from the power module to supply the servos with 5 V each and so the servo rail must be powered externally. One way to achieve this is to source ESCs that are equipped with a battery eliminator circuit (BEC). This takes current from the esc source, in this case the battery, and shares this current with the servos via the servo rail. As there are two ESCs, each with its own BEC, only one of the 5 V regulated supply lines needs to be attached to

the servo rail. If both are connected a power surge could cause some servos to overheat. It is recommended to remove one of the esc 5 V wires when they are y-cabled together.

Figure 114 depicts the servo output rail on the Pixhawk and Table 21 contains information regarding the channel and the role each servo plays. Each channel can be programmed in the mission planner software to perform a different task. However, channels 1 to 4 are pre-defined for ailerons, elevator, throttle and rudder and are usually implemented in this manner.

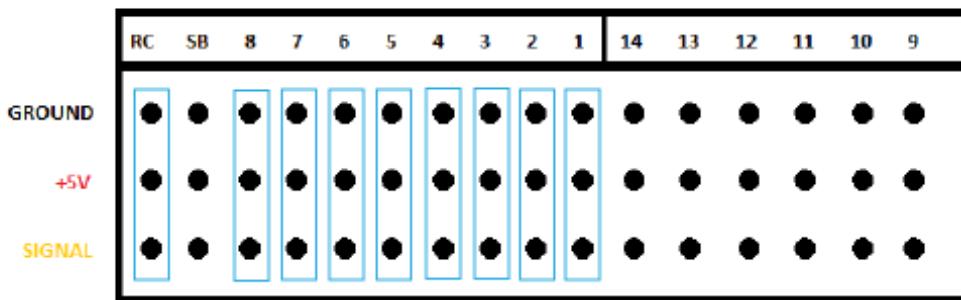


Figure 114 - Pixhawk Servo Output Rail - With Indication of Useful Pins

Table 21 - table to show channel number on Pixhawk and intended purpose

Channel number	Intended purpose
RC	Used to set-up receiver/transmitter connection. Removed and replaced by satellite module plugged into the satellite port in the Pixhawk.
1	Left ailerons, two servos y-cabled
2	Elevator
3	Two ESCs and motor combinations, y-cabled as they should be controlled on one channel for safety reasons
4	Front wheel as the system does not have rudders
5	Left flap
6	Right inner aileron
7	Right flap
8	Right outer aileron

A manufacturing error meant that the two servos for both ailerons on the right wing pointed away from each other. This meant they could no longer be y-cabled as they would move counter-clockwise to each other. Furthermore, both flap servos point away from each other meaning it would produce a roll effect rather than the intended flap action. Luckily, there are enough servo outputs on the Pixhawk to account for these errors.

The fuselage has a hollowed middle section to house the electronics and battery. A rack was designed and laser cut to allow each component to be cable tied down, secure from motion and neat in design. Figure 115 and Figure 116 show the design and layout of the board – a smaller board is cut and placed on top to fill the hollow shape of the fuselage.



Figure 115 - Laser Cut Board to Hold Electronics Inside Fuselage

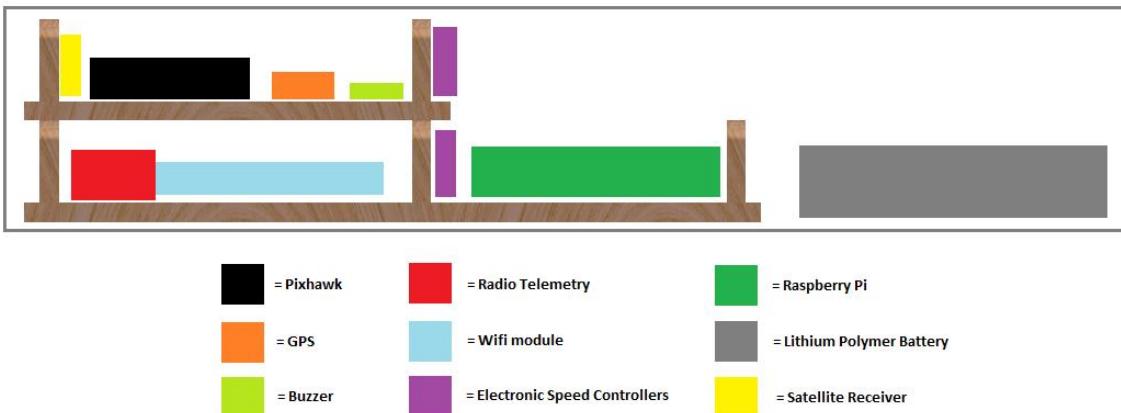


Figure 116 - Layout of Components on Development Board

The battery is placed at the end of the rack and is secured with Velcro-tape to the inside frame of the fuselage. Space was left above the raspberry pi and battery for easy access and excess wires.

9.3.2. Software set-up

The Pixhawk can be programmed using the mission planner software. Upon connecting the Pixhawk to a laptop for the first time, mission planner will upload new firmware onto the device depending on whether the system is a fixed wing plane, quadcopter, copter or land vehicle. The latest firmware version for our system is ArduPlane v3.7.1. The next step is to calibrate the mandatory hardware on board the Pixhawk. Accelerometer calibration requires placing the Pixhawk on each of its axis in turn and a level calibration to ensure starting position is correct. The compass calibration is simple as mission planner comes with a Pixhawk pre-defined compass calibration routine, which is implemented by selecting the Pixhawk/px4 button and ensuring that the checkbox “obtain declination automatically” is checked. If no GPS is acquired the compass calibration will fail – best practice is to perform this task in an open area. Adjustments can be made by changing offset values, but this is more desirable for tasks with specific waypoints or coordinate destinations.

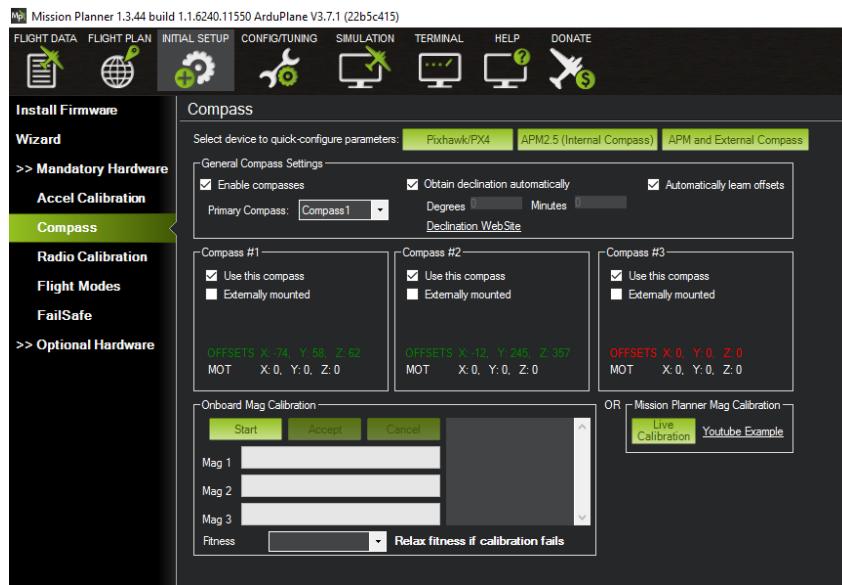


Figure 117 - Compass Calibration Screen in Mission Planner

The next calibration step is the radio calibration which first requires the receiver and transmitter to be bound to each other. This ensures that the receiver-transmitter pair talk to each other and do not interfere with other like devices. The receiver must be short circuited using a provided bind-plug which shorts the connection between ground and the signal line.

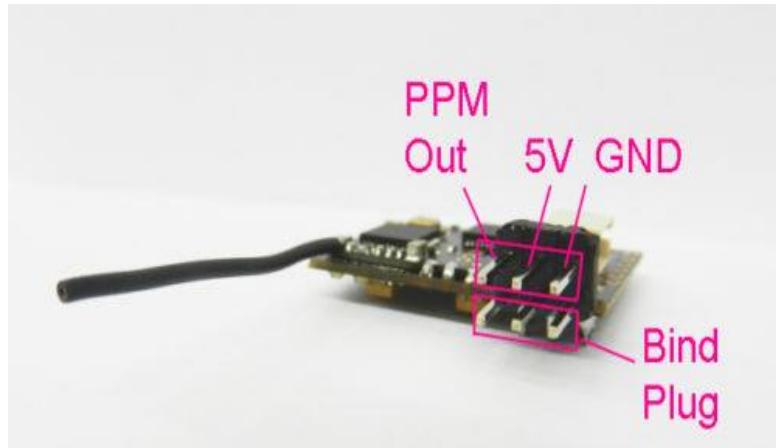


Figure 118 - RC Receiver with Indication of the Intended Bind Plug Location

Meanwhile the Spektrum dx8 transmitter must be switched on whilst holding down the “trainer” switch. Now that both devices are in bind mode, they search for each other and notify via the transmitter screen and Pixhawk buzzer that both have been paired. Radio calibration can be completed via mission planner software.

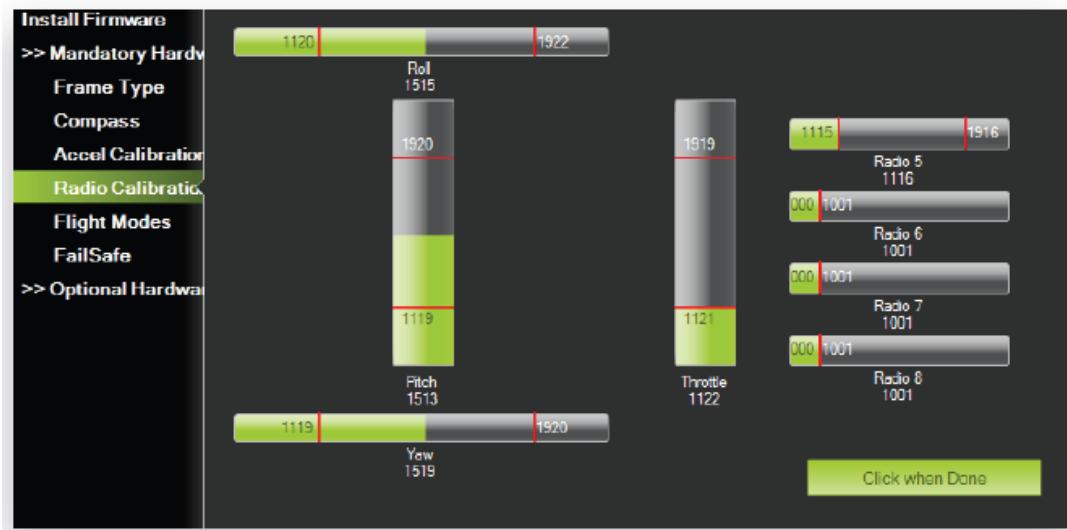


Figure 119 - Radio Calibration Screen in Mission Planner

It is important to ensure that each stick or switch on the transmitter is pairing up to its correct channel in the mission planner software. This is especially important as the transmitter typically has two modes, each with different transmitter outputs for throttle, pitch.



Figure 120 - Diagram Showing the Difference Between Mode 1 and Mode 2 on a Radio Transmitter

In our case, the transmitter was set to mode 2 with the left stick's automatic return position disabled. This enables the pilot to set a constant throttle without the stick returning to its middle position. Finally, a failsafe configuration had to be implemented to ensure safety when flying. The failsafe function cuts the throttle when radio transmission signal has been disconnected or too weak.

For the Pixhawk to know what function each servo will perform, it is necessary to update the Pixhawk parameters. Channels 1, 2, 3 and 4 are automatically set as aileron, elevator, throttle and rudder. In this case, the front wheel and rudder are interchangeable as our system does not have a rudder. Table 22 shows a list of parameters which enable the Pixhawk to know the function of each servo.

Table 22 - Pixhawk parameters relating to the intended function of each channel and descriptions

Parameter name	Value	Description
RC5_function	2	Flap
RC6_function	4	Aileron
RC7_function	2	Flap
RC8_function	4	Aileron

Other parameters include reverse functions, which reverse the direction of a servo and maximum, minimum and trim values which determine the resting position of the servo and its extreme values (RCx_reverse, RCx_max, RCx_min and RCx_trim, where x represents the channel number). For most channels, the maximum PWM value was 2000, the minimum was 1000 and the trim was 1500. Some tweaking of these values ensured all flaps on the wing were level at zero angle and that they all contributed equal roll characteristics.

10. Testing & Results

10.1. Wind Tunnel Testing

10.1.1. *Objective of Wind-tunnel Testing*

Wind tunnel experiments are essential to test for the feasibility and accuracy of the methods proposed when applied in a real-world environment. Three tests were proposed to be able to obtain enough data to verify both the accuracy of the pressure sensors and the lookup tables as well as test the response of the flaps. Using wind tunnel experiments would help prove that the system designed is both viable and safe for use during in-flight tests and applications.

10.1.2. *Lookup Table Verification*

The first experiments were set to verify that the data provided in the lookup table is correct and a close enough representation of what is expected from a three-dimensional wing. This experiment also tests the accuracy of the conversion from a simulated two-dimensional aerofoil section to a finite wing section. The procedure set for this experiment was to:

1. Mount the wing section on the calibrated test rig in the centre of the wind tunnel.
2. Take an initial baseline reading for a zero-wind speed condition.
3. Initialise the angle of attack and flap angle to zero.
4. Increase the wind speed to up to 18 m/s.
5. Rotate the flaps through angles of 0 *degrees* to 30 *degrees* in both directions for each angle of attack between 0 *degrees* and 14 *degrees* in both directions.
6. Record the lift, drag and pitching moment for each combination of flap angle and angle of attack combination.

Post-processing of the data would involve subtracting the baseline reading from each measured lift and converting this to a lift coefficient. The lift against angle of attack would be plotted and compared to the two-dimensional XFOIL simulations, adjusted to consider a three-dimensional wing through lifting-line theory.

It would be expected that the experimental lift would match the lift predicted from the simulations. If there is a systematic discrepancy between the approximated lift from XFOIL and the actual lift

for all cases, an additional factor can be applied to the lookup table data to better represent this real-world situation. If there are many independent discrepancies between the lookup table data and the experimental data, a lookup table may have to be constructed solely from the experimental data.

With the lookup table and test rig verified, the pressure sensors and pressure distribution to lift approximations could be implemented.

10.1.3. Pressure Sensor Verification

The next stage would be to verify that the data collected from the pressure sensors for a given wind speed matches the expected pressure distribution. In addition to this, testing that the lift approximated from this pressure distribution matches the lift measured from the test rig. The procedure for this stage was like the first:

1. Mount the wing section on the calibrated test rig in the centre of the wind tunnel.
2. Attach the pressure sensor controller to the test wing.
3. Take an initial baseline reading for a zero-wind speed condition.
4. Initialise the angle of attack and flap angle to zero.
5. Increase the wind speed to up to 18 m/s.
6. Rotate the flaps through angles of 0 *degrees* to 30 *degrees* for each angle of attack between 0 *degrees* and 14 *degrees* in both the positive and negative direction, approximating lift at each stage.
7. Record the lift, drag and pitching moment for each combination of flap angle and angle of attack combination. Also record the pressure reading from each sensor and the lift approximated for the given pressure readings.

Post-processing of the data would involve comparing the pressure distribution for each angle of attack and flap angle against the predicted distribution from XFOIL simulations. The lift predicted from the pressure distribution would also be compared against the lift measured by the test rig. This would firstly ensure that the pressure distributions are as expected in both amplitude and shape and that the methods to approximate lift from these pressure distributions are accurate enough for application with a feedback loop.

Provided that the approximated lift is correct to within a reasonable range of accuracy, control loop feedback with flap motion can then be tested.

10.1.4. Control Loop Feedback Verification

The final stage would be to test the implementation of feedback used to maintain constant lift. This stage is important to test for the feasibility of each method, the accuracy and response time of the methods and to identify any immediate errors or risks that could occur in-flight. The feedback was in the form of a single rotating flap across the span of the test wing. The procedure was as follows:

1. Mount the wing section on the calibrated test rig in the centre of the wind tunnel.
2. Attach pressure sensor controller to test wing.
3. Take an initial baseline reading for a zero-wind speed condition.
4. Increase the wind speed to up to 18 m/s.
5. Set the angle of attack at any known angle and measure the lift.
6. Induce flow disturbances across the wing to simulate gusts.
7. Record approximate lift and flap angle against time.
8. Set a constant lift to be maintained on the wing between -20 N and 20 N to initialise a feedback loop.
9. Wait for the flap to stabilise and verifying that the lift approximated is as expected.
10. Record approximate lift and flap angle against time.

It would be expected that constant lift is not kept entirely constant while the feedback is active but any disturbances from the ideal lift condition would be damped significantly compared to the same disturbances without feedback active.

Post-processing of the data would first involve checking that the correct constant lift is maintained without any disturbance during the times in which the feedback is active and the flap is stabilised. Further to this, the response time back to the initial lift after a disturbance is induced should be compared for the cases with the feedback on and off to verify that:

- The wing returns to the same lift condition that it was in before the disturbance was induced.
- With feedback active, there is a shorter duration between the induced disturbance and the return to the initial lift as opposed to a system without feedback.

With all three stages complete and verified, the system would be safe and accurate enough to be implemented on a UAV for flight testing.

10.1.5. Wind tunnel Testing Implementation

The wind tunnel testing procedure is illustrated in Figure 121. Although the data flow is similar to that of the calibration procedure, the objective now is to estimate unknown forces using known calibration parameters, flap angles and angles of attack.

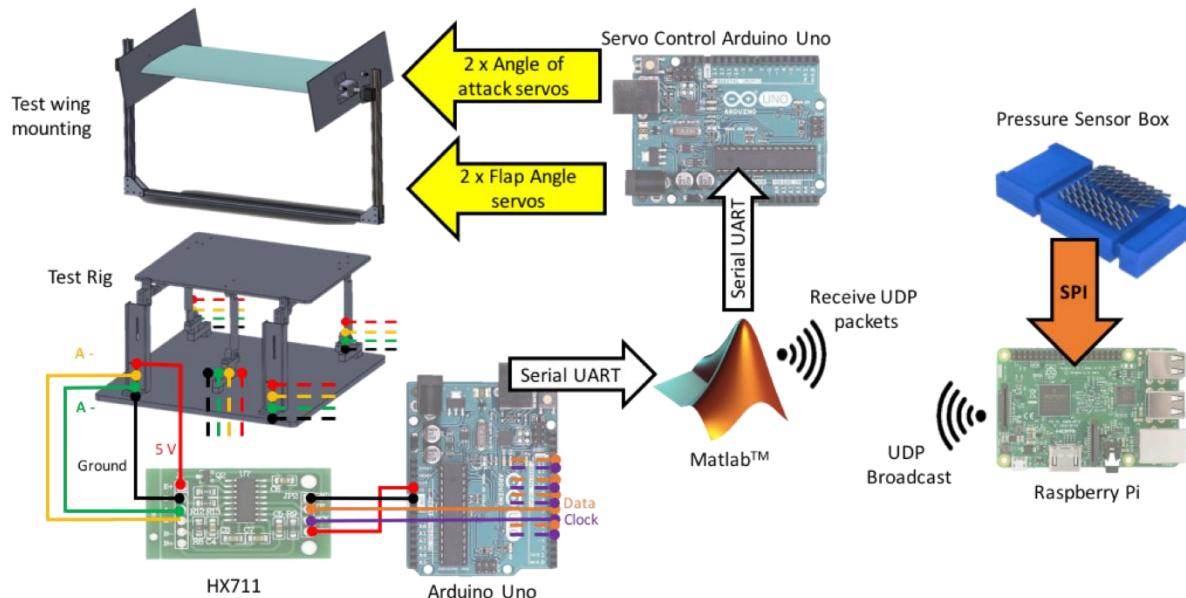


Figure 121 - Data flow during wind tunnel testing

The MatLab script `Wind_tunnel_tests.m` automates the testing procedure. It sends an instruction via serial telling the servo control Arduino the required angle of attack and flap angle. Following this it records the forces measured by the test rig and pressure data from the sensors fitted in the wing section.

The angle of attack is set to vary from 0 to 14 *degrees* in increments of 2 and then from 0 to -14 *degrees* in decrements of 2. For each angle of attack, the flap angle is varied from 0 to 30 *degrees* in increments of 3 and then from 0 to -30 *degrees* in decrements of 3. For each combination of angle of attack and flap angle, 15 samples from each test rig load cell are recorded in MatLab objects. See Appendix H for implementation.

10.2. Wind Tunnel Results

The first wind tunnel test procedure was performed with the expectations of validating the XFOIL data. However, numerous issues were encountered during the testing phase causing incomplete and unreliable data to be obtained.

The first noticeable problem during the testing was wobbling of the test rig support arms. This was later concluded to be due to flexible acrylic being used as the baseplate and the attachment points, which would bend and distort when subject to forces. This wobbling was expected to distort the results and with the knowledge of the baseplate bending, it cannot be assumed that the loads recorded were correct.

At higher angles of attack, near stall, the test wing would begin to lose structural integrity and required the tape to be reapplied. The action of reapplying the tape caused additional forces to be read and if obtained during a read cycle, would heavily distort readings. Finally, the test had to be concluded prematurely due to servo damage.

Despite the incomplete testing and errors that occurred, enough results were obtained to confirm that the procedure itself was correct and could easily be rectified with further testing. The lift read was converted into a lift coefficient value and then divided by an aspect ratio lifting-line correction factor to obtain a two-dimensional equivalent:

$$Cl_{2D} = Cl_{3D} \left(\frac{AR+2}{AR} \right) \quad (38)$$

Where Cl is the lift coefficient and AR is the aspect ratio of the test wing.

As shown in Figure 122, the lift coefficient increased with angle of attack as expected and at small angles where the errors mentioned above were minimal, gave a relatively linear result. Stall also occurred between 12 and 14 degrees which matched with the expectation of stall occurring at approximately 13 degrees. This trend confirms that both the wing and test rig procedures are operating correctly.

Although it was expected that even after correction the experimental lift coefficients would be smaller than that of the simulation, they were all a factor of 3 or more with no constant factor or trend between factors. However, given note of the errors mentioned above, this may not be the case when repeated with the errors rectified. Repeating the test should either provide a more consistent factor that could be applied to all results for the pressure to lift conversion or a set of data that matches that of the simulation data.

Altogether, no conclusions could be made from the data from this wind tunnel experiment due to the uncertainty of the results. However, the procedure was correct and the data was sufficient to show that it would be beneficial to perform further experiments after all issues had been rectified.

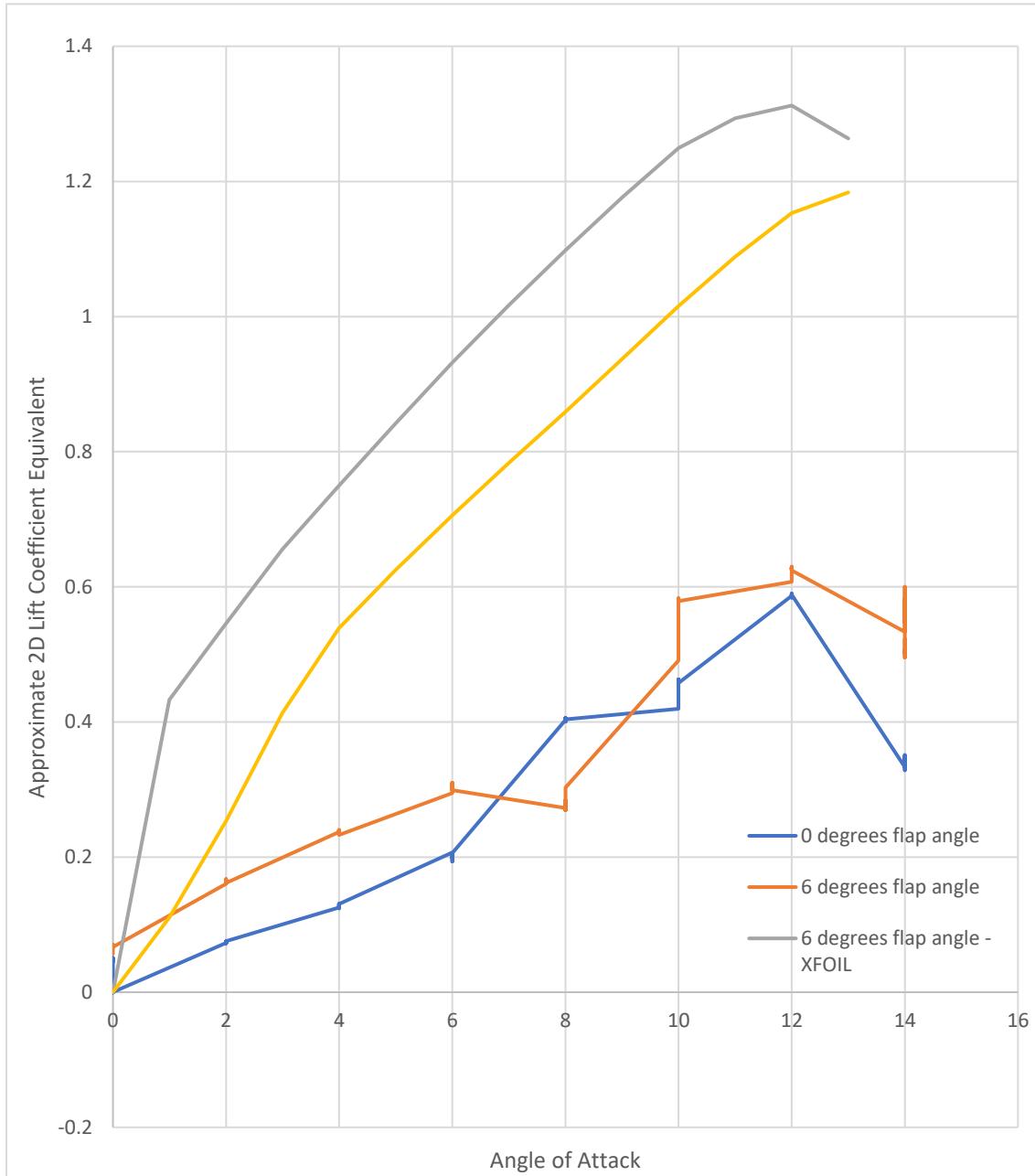


Figure 122 - Approximate two-dimensional Lift Coefficient equivalent against Angle of Attack from the wind tunnel test data.

Table 23 - Comparison of Simulation Lift Coefficient to Experimental Lift Coefficient.

Flap Angle	Angle of Attack	Simulation Lift Coefficient	Experimental Lift Coefficient
0	2	0.2537	0.073758
0	4	0.5388	0.127493
0	6	0.7059	0.202262
6	2	0.5458	0.163139
6	4	0.7499	0.236952
6	6	0.9317	0.303456

10.3. UAV Flight Tests

10.3.1. *Flight testing*

Day 1 – 21st March 2017, Paul Heckles flight centre, Draycot Aerodrome, Swindon

Difficulties arose getting the aircraft ready for the morning of the 21st. Significant efforts were made to remedy previous mistakes in manufacturing – none of which were significantly long jobs, just many small issues. Specifically, fixing the outer wings to the middle section took some time as this was an issue which would cause our system to crash if it had not been perfected. With the realisation that a second flight day was confirmed for the 23rd march, the rush to get a flying, stable model was not so great. Shortly after midday, once all manufacturing and tidying had been completed we were ready for testing. However, when the Pixhawk was connected to the battery and transmitter little or no communication conveyed. For an unknown reason, the Pixhawk and receiver seemed to be disconnected to the transmitter – no signals from the controller affected any of the servos on the aircraft. Initial thoughts included the servo power via the 5v BEC was disconnected – this was not the case as the voltage level was measured across the servo rail terminal. Another thought was the fact that many other teams had Pixhawk controllers nearby and some interference may have crossed our system causing our receiver-transmitter pair to become disconnected. A new bind procedure took place: attaching the bind plug to the receiver which is plugged back into the RC-in port on the Pixhawk servo rail; turning the transmitter off, then on again with trainer/bind button pressed. The bind was successful but the Pixhawk still did not respond to any input from the transmitter. After careful tinkering through different ideas, it was

decided to factory reset the Pixhawk controller, upload a new firmware and start the software calibration procedure again. At this point, most teams had their last flight of the day and time was against us. It was decided that we would take our system back to the lab and ensure its readiness by the morning of the 23rd.

Day 2 – 23rd March 2017, Paul Heckles flight centre, Draycot Aerodrome, Swindon

Still to this day, the unpredictable nature of the flight controller on the first flight day is unclear. However, the previous night we had the Pixhawk working and in a stable condition. By mid-morning we were assembled and had performed our safety checks:

- All flaps deflect the right way when prompted (software rectification if false: RCx_reverse parameter in mission planner).
- All flaps move the same distance when deflected. I.e. Both outer ailerons must deflect the same maximum and minimum distances to ensure roll stability, as well as both inner ailerons and both flaps.
- All flaps are trimmed and level to the trailing edge at zero position.
- Escs are calibrated so that they store the maximum and minimum throttle values for the motors.
- Both motors spin in the same direction (anti-clockwise for the specific propellers we had acquired). If false, the remedy is to swap any two of the three wires from the motor as they are three phase motors – swapping any two wires would reverse the polarity.
- Both propellers are sufficiently tight that they do not slip and reduce thrust whilst flying.
- The Pixhawk has a good GPS connection.
- The motors are disarmed naturally, to fix the propellers and to ensure safety when handling the aircraft.
- Arming the motors via the telemetry radio link to a laptop with mission planner. The software must be active and connected to the Pixhawk via a MavLink connection. Arming is achieved by selecting the “actions” tab in the “flight data” page and then selecting “arm/disarm”.
- The failsafe function shuts down the motors if there is a sudden lack of transmission to the receiver. This ensures the aircraft does not continue a flight path without input from the pilot.

- the range test between the receiver and transmitter at half power passes at a distance of 50 feet.
- Wi-fi connection is established between the raspberry pi and data recovery laptop.
- Incoming data is present on the data recovery laptop that sufficiently reflects the state of the pressure tappings.
- Video streaming from the wing-cam is live via the raspberry pi and wi-fi connection to the data recovery laptop.

10.4. UAV Flight Results

10.4.1. Flight 1 – 1:51pm 23rd March 2017

Take-off distance before the aircraft was airborne was a little longer than we would have liked. It seemed as if the aircraft wasn't moving fast enough to gain enough lift. However, once airborne at the end of the runway the aircraft performed well. Lift was sufficient to get the aircraft up to a maximum height of 49m above ground. Comments from the pilot, Paul heckles, included the trims for each flap are excellent and roll control is very accurate. Ground speed of the aircraft was also measured by the Pixhawk – the average was around 9 m/s however, with the wind the aircraft managed to reach speeds of 30 m/s.



Figure 123 - Waypoint Data Including Altitude from Flight 1 Loaded into Google Earth



Figure 125 - Waypoint Data from Flight 1 Loaded into Google Earth

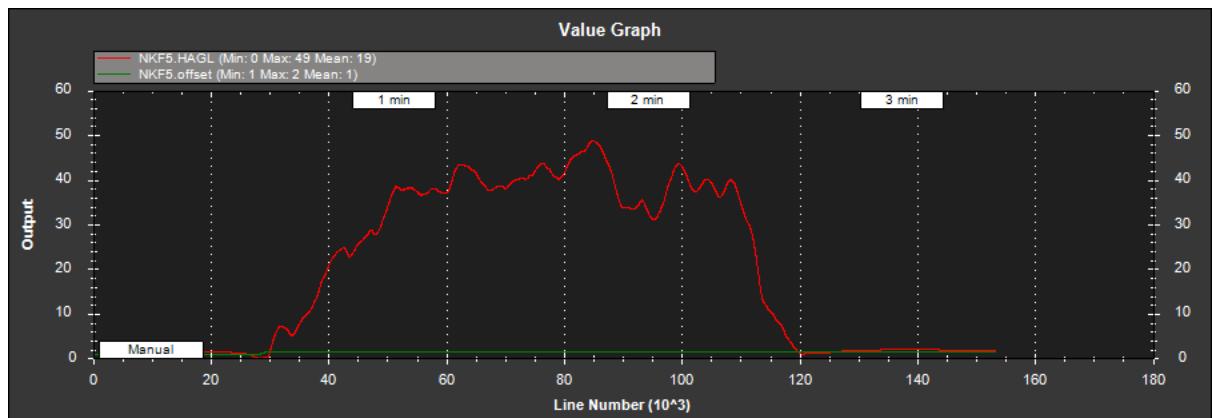


Figure 124 - Height Above Ground Versus Time

(X-axis: Time measured in seconds, Y-axis: Height Above Ground measured in meters)

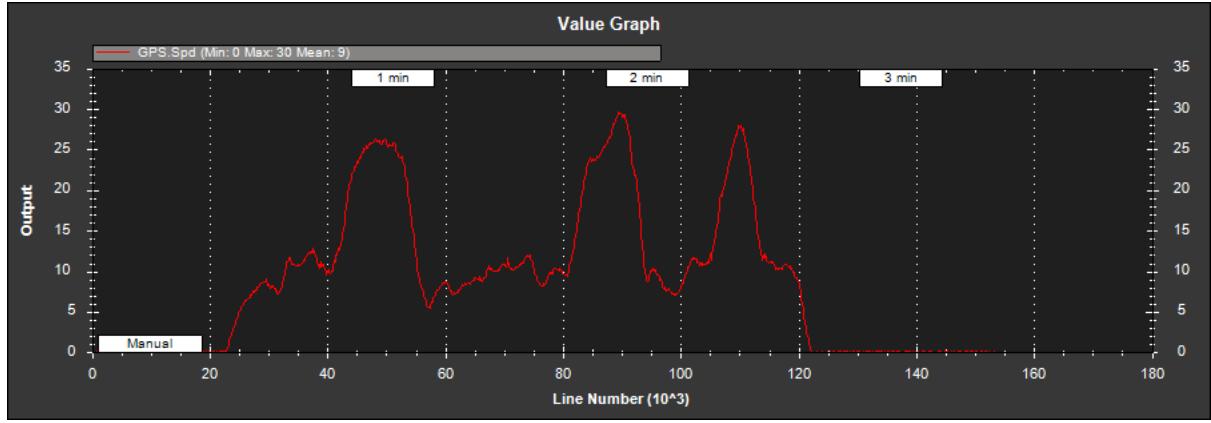


Figure 126 - Ground Speed of our Aircraft. Three Spikes Represent Flying with the Wind.

(X-axis: Time measured in seconds, Y-axis: Speed Relative to the Ground measured in meters per second)

After the flight, we put the battery on charge to ensure it has enough voltage to supply the motors with full throttle. Other steps included installing the pitot probe and yaw sensor. Wind had picked up since the last flight and concerns arose from flying a lightweight aircraft in such a strong wind. Previous flights for other teams seemed an issue with one team crashing due to strong head-wind. Further work was carried out to guarantee the wings were secure on the spars.

10.4.2. Flight 2 – 4:58pm 23rd March 2017

Initial take-off was shorter this time as the battery clearly had an improvement on the performance of the motors. Being a lightweight aircraft caused a few concerns in the early ascent of the flight as strong head winds caused the wings to wobble and appear as if they were flapping like a bird. The thin wing shape did not help in this situation as immediate concerns started to wonder if the foam would snap. However, once the plane finished its ascent it looked a lot more stable and continued flying comfortably until it landed. Notes from the pilot included how well he thought our wing was bolted onto the frame as it managed to withstand strong winds.



Figure 128 - Waypoint Data from Flight 2 Loaded into Google Earth



Figure 127 - Waypoint Data Including Altitude from Flight 2

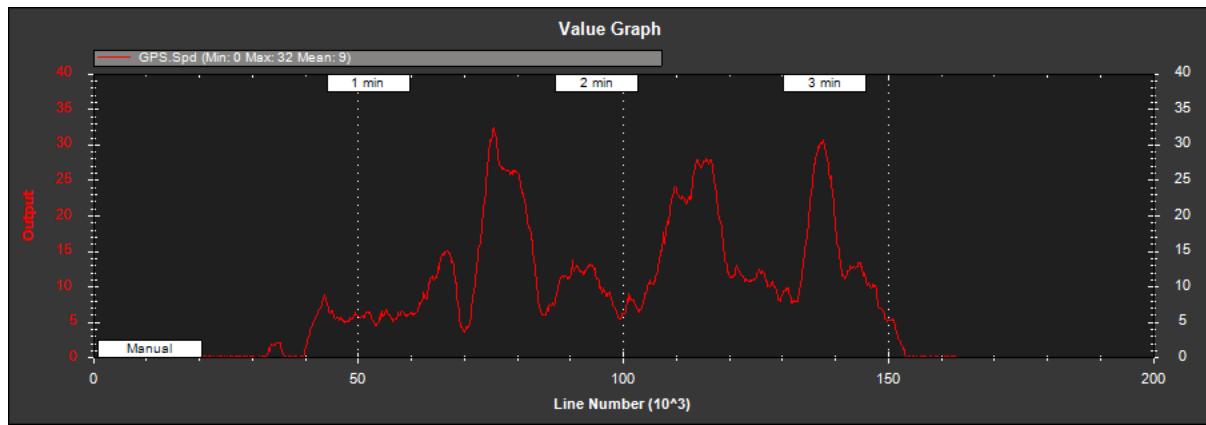


Figure 129 - Ground Speed of our Aircraft. Three Spikes Represent Flying with the Wind.

(X-axis: Time measured in seconds, Y-axis: Speed Relative to the Ground measured in meters per second)

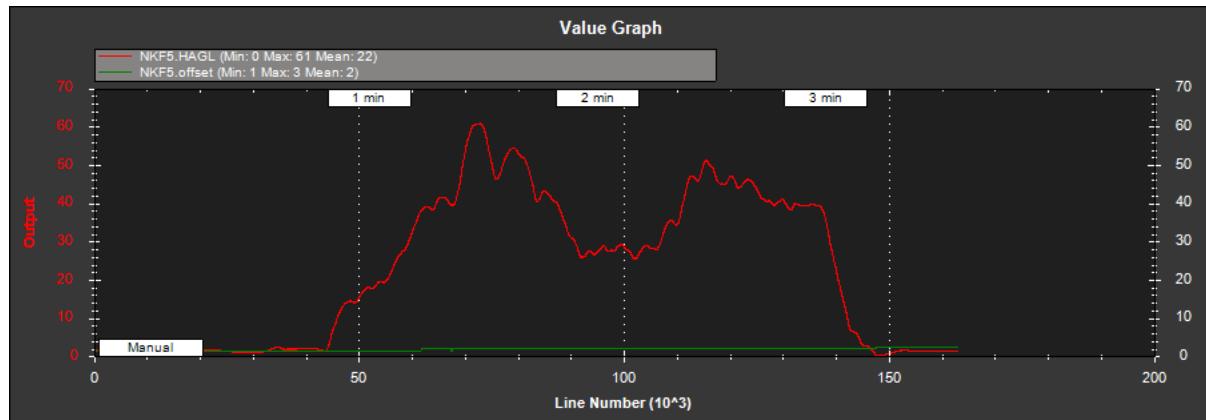


Figure 130 - Height Above Ground Versus Time

(X-axis: Time measured in seconds, Y-axis: Height Above Ground measured in meters)

11. Discussion & Conclusion

11.1. Discussion

11.1.1. Lessons from flight tests – Data Collection

Throughout the entirety of both flights there was always a network connection between the ground station and the UAV. The webcam live stream worked effectively, and data was being transmitted across the network. The Raspberry Pi 3 was also never over $\sim 40\%$ CPU usage so was easily capable of sending out all the packets. However, after the flights it was found the data recorded on the laptop from the flight tests only had 25% and 6% of the packets sent, for flight test 1 and 2 respectively. In hind sight, it was believed this was due to the time taken for MATLAB® to redraw the live plot. Hence, after the flight tests, some ground tests were completed with the same setup. These found that by completely removing the live plotting, 99% of the data packet were received, and with a 20 Hz refresh rate, 98% of the packet were received. All these tests were conducted at a range of $\sim 500\text{ m}$.

11.1.2. Lessons from the flight tests – Sensor Box Sealing

The second issue that was found on the flight day was that none of the boxes effectively sealed all 64 of their ports, with many ports showing varying amounts of coupling with their neighbours. From inspection, it was realised the gasket was stiffer than the lid, hence wasn't effectively compressing the gasket. Two design changes have since been made to alter this, firstly a softer gasket (40T) material was used. Secondly ridges were placed on the underside of the lid, as shown in Figure 131, to increase the contact area between the gasket and the lid. With these two design changes, it was found it later ground tests that the all 64 sensors could be effectively individually sealed.

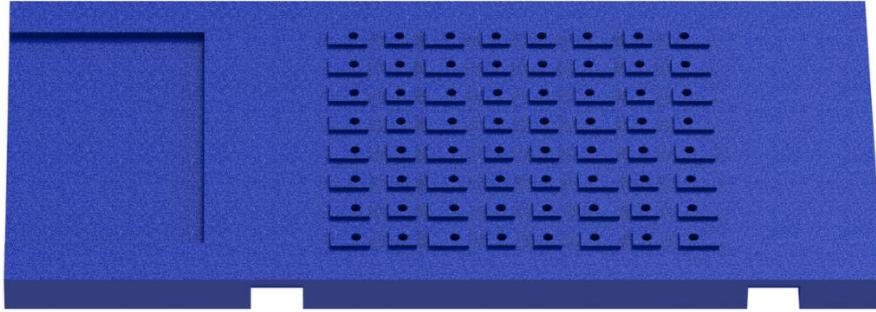


Figure 131 - A render showing the raised areas on the lower side of the lid created to increase the potential of the lid to seal each pressure sensor effectively

11.1.3. Flight test results – Static Pressure Measurements

Only 50 of the 64 ports in each wings sensor box were used for the test sections, the other 14 were left unattached, and hence were a good measure of static pressure. As all the sensors are absolute sensors they will be affected by change in static pressure, in particular how it changes with altitude. An estimate for the altitude of the aircraft (h), given a static pressure reading (P) can be calculated using equation, $h = \frac{R_0 T_0}{g M} (\ln P_0 - \ln P)$, where T_0, P_0 are standard temperature and pressure at sea level respectively, g is the gravitational constant and M is the Molar mass of air. Using this equation and taking average pressure readings across both sensor boxes for the first flight test an estimate of the aircrafts altitude during the flight can be seen in Figure 132. Also in this figure, you can see the recorded data from the barometric sensor in the Pixhawk, and the two signals have a strong correlation.

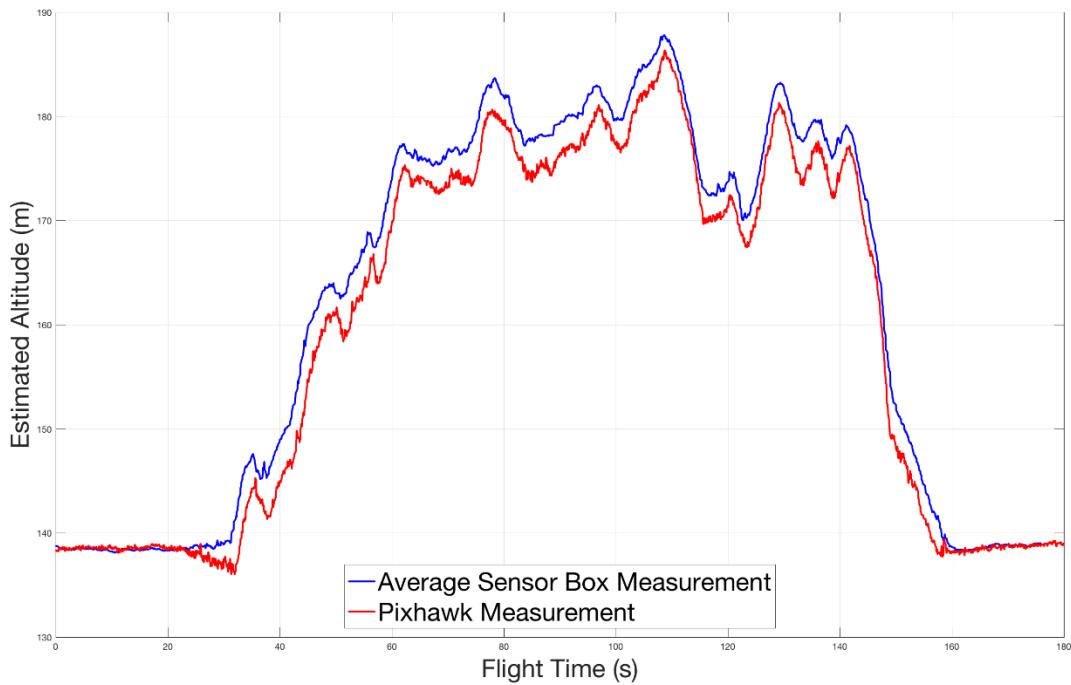


Figure 132 - A plot of the estimated altitude of the aircraft using the static pressure measured using the developed Sensor boards, compared to that measured by the Pixhawk, during the first flight test.

Such is the apparent accuracy of these measurements, that if you measure the estimated altitude of each Sensor box individually, you could estimate the roll angle of the aircraft using simple geometry (and assuming the sensor boxes are 1.5 m apart). A comparison between the estimated roll angle using the height of the sensor boxes and the roll angle measured by the inertial measurement unit in the Pixhawk can be seen in Figure 133. Although the plots do not exactly match, bearing in mind these boxes were sealed within the wing with no direct vent, there is a strong enough correlation between these two plots to show these sensors may be accurate enough to measure subtle changes such as changes in height in the region of 10's of centimetres. Due to this accuracy in future test it was realised extreme care had to be taken converting absolute pressures to relative pressures due to both effects outlined above.

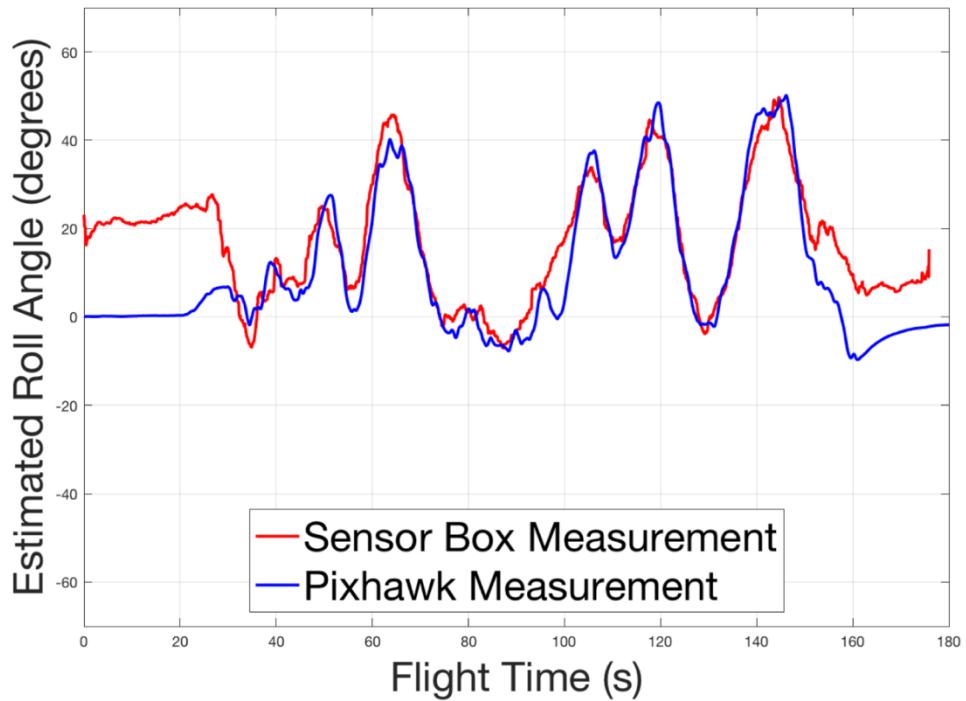


Figure 133 - A plot of the estimated roll angle of the UAV during the first flight test using 2 sources; the inertial measurement unit in the Pixhawk flight controller and a trigonometric relation between the altitude of the two sensor boxes.

11.1.4. Flight test results – Pressure Tapping Measurements

Although a few of the ports in the sensor boxes were not effectively sealed, on the day of the flight test each pressure tapping was tested to see which channel on the sensor box it corresponded to. Over 50% of the pressure tapping's did not produce a response on any channel, meaning there were some serious flaws with assembly of the wing section. In particular, the volume required to effectively root all 50 pieces of pneumatic tubing was neglected meaning many tubes were most likely squashed closed when the wing was assembled. Due to this only 10-20 channels varied significantly during the flight tests, which was not enough to be able to draw any conclusions on the shape of the pressure distribution on either wing. However, once the effects of altitude had been corrected these ~ 20 channels showed variations in the order of 200-300 pascals, meaning if the system was assembled correctly in a later test, and all the problems mentioned above are resolved it is conceivable the developed system would be capable of accurately measuring the pressure distribution around a UAV wing.

11.2. Conclusion

A pressure sensor system that is fitted inside a wing and measure the pressure distribution across it has been demonstrated. This was achieved first by choosing a cheap platform to attach a smart wing to and then designing and manufacturing a set of wings that could effectively record pressure readings at set points around the wing aerofoil. This only needed a completed control system to allow the data to be used to control lift distribution over the wing and hence dampen out effects caused by gusts, keeping the UAV on a steady flight trajectory. In addition to this, a functioning test rig which can be attached to the rear of the small wind tunnel at the Boldrewood campus has also been completed.

It has been shown that it would be possible to obtain a value of the lift over a wing from the pressure distribution across the wing and that various methods have the capability of doing so within reasonable degrees of accuracy. With knowledge of the lift across the wing, the required flap angle can be identified and adjusted to maintain constant lift. The methods provided are valid and effective across a two-dimensional aerofoil section and further approximations can be made for a three-dimensional wing.

It was shown that the application of multiple BMP280 pressure sensors were a viable method of measuring the pressure in-flight. It was possible to fit many sensors in a small area, giving potential for many future applications. Measurements from these sensors could be recorded in real time and provide an accurate result, giving similar estimations to that of the Pixhawk flight controller in terms of altitude and roll.

The wing structure has been designed to be modular and allow easy access to internal components such as the pressure sensing box. It has effective attachment mechanisms which allow easy and secure attachment to the UAV platform that was chosen and includes a 3D printed section which contains a large number of pressure tapping which connect to the pressure sensing box via pneumatic tubing.

A test rig which can successfully differentiate between lift and drag forces was developed and manufactured. It does this via decoupling the lift and drag forces using struts attached to pillow block bearings. After calibration, both forces could be measured within a good range of accuracy. The full UAV system has also been flight tested by a professional pilot which displayed that the pressure sensing system can send the pressure data to be viewed live through a Wi-Fi connection however this did also show that the large majority of this data was not successfully saved.

11.3. Recommendations and Further Work

Careful planning and following through with management tools, such as a Gantt chart, helped to make this project a success. However, some problems were encountered during parts of the manufacturing and practical experiment stage. This section will discuss the methods and designs that could be improved throughout the project to solve the encountered issues.

11.3.1. Pressure sensing data to lift

The lift was predicted from the pressure distribution across the wing with a reasonable amount of accuracy using the method outlined previously in the report. As this project is mainly to prove the design concept, it had only been applied on a 2D wing profile and wind tunnel testing. Hence, the discussed methods were valid and effective across a 2D aerofoil section – further approximation can be made for a 3D wing. However, to have a more accurate approximation of the lift over the aerofoil, the real data obtained from the wind tunnel testing on the wing section should be used to construct the lookup table. This would better represent the pressure distribution measured during the flight. Further wind tunnel testing could also be used to design and test a control system for flap actuation to test the response to gusts.

Furthermore, in this project the methods used have only been tested on a NACA0012 aerofoil section. Analysis should be run across multiple aerofoil sections to test for the feasibility on wings with different aerofoils. From there, global methods that are applicable to all or most lifting surfaces could then be established. This would further increase the scope of application of the project as it could then be applied to any surface without requiring further analysis.

11.3.2. *Test Wing Design and Manufacture*

Throughout the manufacture and build phase of the project there were problems and difficulties that were encountered which, in hind sight, could have been approached a different way.

Firstly, while in flight it was observed that the wings of the UAV seemed to flex more than expected. Although the UAV survived the flight, insufficient stiffness of the wings could have caused a crash landing and unusable data. This is most likely due to underestimation of the wind effects on the wing during the design stage. To improve the wing stiffness, carbon fibre spars with bigger diameter could be used. Using larger spars for the wing structure would enhance the wing's structural stability during flight which would allow for a safer and more confident flight during varying flight conditions.

Secondly, the servo connected to the front wheel should be upgraded for one with more torque. The landing gear was not functioning well when the UAV was put on the grass runway. This was due to a large amount of friction at the contact between the landing gear and the ground. The landing gear servo did not have sufficient torque to overcome the frictional force. Like the wing stiffness issue mentioned above, this had no impact on the outcome of the project, however, it is still an issue to flag for future work.

Thirdly, some improvements to the hot-wire cut foam sections could be made. It was found that during the assembly of the wings, the wiring slots were not big enough and it ended up being a fiddle job to feed all the wires through the wing section. An improvement would be to cut the foam sections with larger wiring slots – as long as it does not affect the shape of the aerofoil. Also, the middle section containing the sensor box should be altered to hollow out the foam in front of the leading-edge spar. This allows the pressure tubing which connects to the leading-edge pressure tapping hole to go around the leading-edge spar and connect to the sensor box, which is fitted inside the wing, without causing any blockages in the tubes.

Fourthly, improving the surface finish of the wing would be a nice addition. The surface finish of the wing sections in this project is done by using Papier Mache and spray paint. This was chosen as it is a quick, cheap solution for design purposes. After a couple of flights, it was noticed that the

corners of the wing were not stuck down perfectly and further test flights might have pulled this surface off. It could be improved by using fibreglass or Mylar to wrap around the wing sections.

11.3.3. Test rig Design and Manufacture

There are several improvements that can be made on the test rig to enhance the stability and effectiveness of the force decoupling system and the accuracy of the load measurement. It was found that the upper platform of the test rig starts to vibrate and flex significantly at the maximum test airspeed of roughly 15 m/s . It could cause a lack of accuracy in the test data as the upper plate's vibrational mode could affect the load cell measurement. To solve this problem, the platform which is made of an acrylic plate should be joined with or replaced by a stiffer material, for example a steel plate. The stiffer material would increase the upper platform's natural frequency and forcing it to be stable at the range of airspeeds used during testing. It is also recommended to carry out modal analysis as well as static structural analysis on each component of the test rig before manufacture, in order to identify and avoid any potential vibration problems that would occur.

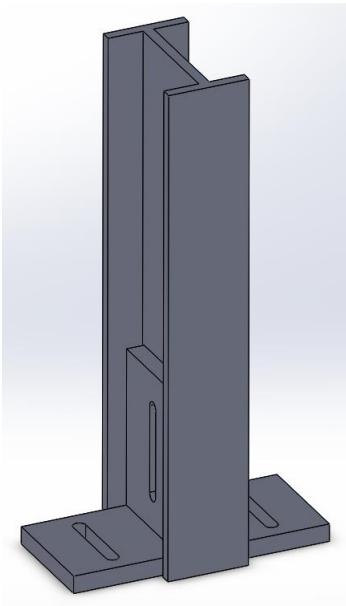


Figure 134 - Extra bracket on rigid stand for future test rig

To further stiffen the test rig, extra brackets should be fixed to the opposite side of the vertical rigid structure (see Figure 134) to further secure it on the base platform. This would increase the accuracy for drag measurement. Although the current design is sufficient to withstand the expected small amount of drag, it would not be sufficient to withstand much higher drag. Furthermore, the upper structure of the test rig has sufficient rigidity for the interested wind speed. However, it should be improved for future wind tunnel tests which focus on higher wind speed. This allows the wing structure to remain rigid even if the wing is exposed to much higher drag forces.

Bibliography

- [1] P. Bot, I. M. Viola, R. G. Flay and J.-S. Brett, “Wind-tunnel pressure measurements on model-scale rigid downwind sails,” *Ocean Engineering*, vol. 90, pp. 84-92, 2014.
- [2] D. Yeo, E. M. Atkins and L. P. Bernal, “Aerodynamic Sensing for a Fixed Wing UAS Operating at High Angles of Attack,” 2012.
- [3] H. Takahashi, K. Matsumoto and I. Shimoyama, “Differential pressure measurement of an insect wing using a MEMS sensor,” 2012.
- [4] Bosch Sensortec, “BMP280 Data Sheet,” [Online]. Available: <https://goo.gl/9okwXh>. [Accessed 6 April 2017].
- [5] Arduino, “Arduino IDE Download Page,” [Online]. Available: <https://www.arduino.cc/en/Main/Software>. [Accessed 6 April 2017].
- [6] MathWorks, Inc., *MATLAB 2017a*, Natick, MA, 2017.
- [7] I. Grokhotkov, “esp8266 Arduino Core Library,” [Online]. Available: <https://github.com/esp8266/Arduino>. [Accessed 8 April 2017].
- [8] WiringPi, “WiringPi Home Page,” [Online]. Available: <http://wiringpi.com/>. [Accessed 9 April 2017].
- [9] C. Kohlhoff, “Boost Library Home Page,” [Online]. Available: boost.org. [Accessed 9 April 2017].
- [10] L. Jackson, “MJEG Streamer repository,” [Online]. Available: <https://github.com/jacksonliam/mjpg-streamer>. [Accessed 9 April 2017].
- [11] J. Moran, *An Introduction to Theoretical and Computational Aerodynamics*, Dover Publications Inc., 1984.
- [12] S. Obeid, R. Jha and G. Ahmadi, “RANS Simulations of Aerodynamic Performance of NACA 0015 Flaped Airfoil,” 2017.
- [13] T. Ahmed, M. T. Amin, S. M. R. Islam and S. Ahmed, “Computational Study of Flow Around a NACA 0012 Wing Flapped at Different Fap Angles with Varying Mach Numbers,” *Global Journal of Research in Engineering*, vol. 13, pp. 4-16, 2013.

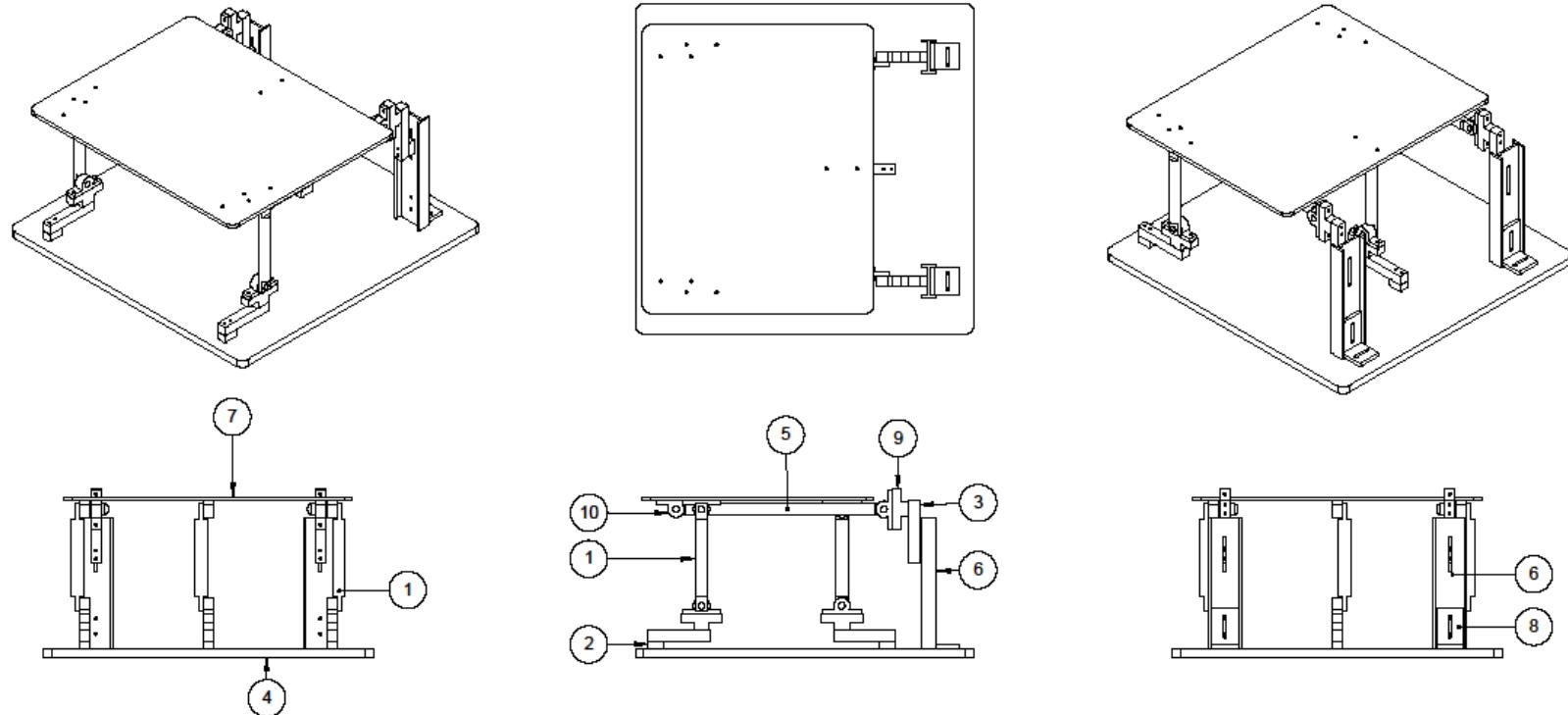
- [14] C. Reinsch, “Smoothing by Spline Functions,” *Numerische Mathematik*, vol. 10, no. 3, pp. 177-183, 1967.
- [15] D. Pollock, “Smoothing with Cubic Splines,” [Online]. Available: <http://www.physics.muni.cz/~jancely/NM/Texty/Numerika/CubicSmoothingSpline.pdf>. [Accessed 4th February 2017].
- [16] A. M. a. S. J. D. Kuethe, Foundations of aerodynamics, 2 ed., New York: Wiley, 1959.
- [17] Minnesota Scale and Giant Scale RC, “Calculate Required Servo Torque,” [Online]. Available: <http://www.mnbigbirds.com/Servo%20Torque%20Calculator.htm>.
- [18] Easy Composites, “Roll Wrapped Carbon Fibre - Easy Composites,” [Online]. Available: <http://www.easycomposites.co.uk/#!/cured-carbon-fibre-products/carbon-fibre-tube/roll-wrapped-carbon-fibre-tube>.
- [19] Performance Composites, “Mechanical Properties of Carbon Fibre Composite Materials,” July 2009. [Online]. Available: http://www.performance-composites.com/carbonfibre/mechanicalproperties_2.asp.
- [20] T. Witushynsky, “Experimental Investigation into the Aerodynamics of Small Air Vehicles with Thin Flexible Wings,” *Case Western Reserve University*, 2008.
- [21] M. Samardžić, Z. Anastasijević, D. Marinkovski, D. Ćurčić and J. Isaković, “External Six-Component Strain Gauge Balance for Low Speed Wind Tunnels,” *Scientific Technical Review*, vol. 64, no. 3, pp. 40-46, 2014.
- [22] Applied Measurements LTD., “Load Cells and Force Sensors,” [Online]. Available: <http://www.appmeas.co.uk/load-cells-and-force-sensors.html>.
- [23] sparkfun, “24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales,” [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf. [Accessed 19 04 2017].
- [24] Femtoscan Online, “Median Filter,” [Online]. Available: http://www.femtoscanonline.com/wiki/en/processing/%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%84%D0%B8%D0%BB%D1%8C%D1%82%D1%80. [Accessed 19 04 2017].

- [25] M. A. Ramaswamy, T. Srinivas and V. S. Holla, “A simple method for wind tunnel balance calibration including non-linear interaction terms,” *Proceedings of the ICIASF 1987 RECORD*, pp. 263-267, 1987.
- [26] M. Ferriera, “Design of a Six-Component External Wind Tunnel,” [Online]. Available: <https://fenix.tecnico.ulisboa.pt/downloadFile/844820067124978/Dissertation.pdf>. [Accessed 11th January 2017].
- [27] Bosch Sensortec, “Bosch Sensortec BMP280 Driver Repository,” [Online]. Available: https://github.com/BoschSensortec/BMP280_driver. [Accessed 5 April 2017].
- [28] sparkfun, “Load Cell Amplifier HX711 Breakout Hookup Guide,” [Online]. Available: <https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide>. [Accessed 19 04 2017].

Appendix A

Part description	Company	Quantity	Unit Cost (£)	Cost (£)	SubTotal	Taxes	Shipping costs	Discount	Total	Who	Date Ordered
V-Slot Linear Rail - 20x20x500mm	Technobots Ltd	2	£4.50	£9.00	£18.00	£4.81	£6.63	£3.00	£26.44	A. Hill	29/11/2016
V-Slot Linear Rail - 20x20x1000mm	Technobots Ltd	1	£9.00	£9.00							
6x CNBTR aluminium YZC-131 Kitchen scale weighing sensor load cell 5Kg	CNBTR	1	£14.46	£14.46	£14.46	£0.00	£2.60	£0.00	£17.06	R. Pasupathy	17/11/2016
2x HX711 weighing sensors AD Module load cell	FlyFun Tech.	1	£9.98	£9.98	£9.98	£0.00	£0.00	£0.00	£9.98	R. Pasupathy	17/11/2016
Replacement electronic scale 0-20Kg range weighing sensor load cell	Sourcingmap	1	£8.03	£8.03	£8.03	£0.00	£0.00	£0.00	£8.03	R. Pasupathy	17/11/2016
Collar One Piece Screw, Bore 10mm, OD 20mm, W 10mm, Steel	RS Components	2	£1.37	£2.74	£2.74	£0.55	£0.00	£0.00	£3.29	A. Hill	02/12/2016
Test Rig materials without work hours	EDMC	1	£55.00	£55.00	£55.00	£0.00	£0.00	£0.00	£55.00	Felicia	14/12/2016
Metric mounted ball bearing 8mm	S. Pawlick	10	£4.99	£49.90	£49.90	£0.00	£0.00	£0.00	£49.90	Felicia	28/11/2016
Pultruded CB tube 10mm 1m	Easy Composites	1	£9.45	£9.45	£18.05	£4.71	£5.50	£0.00	£28.26	L.Winter	16/12/2016
Pultruded CB tube 8mm 1m	Easy Composites	1	£8.60	£8.60							
Flow perturbation rig material	EDMC	1	£20.00	£20.00	£20.00	£0.00	£0.00	£0.00	£20.00	N. Tarasov	03/01/2017
UAV Fuselage	Bormatec	1	300.00 €	300.00 €	450.00 €	89.30 €	20.00 €	£84.54	£474.76	L.Winter	13/02/2017
Landing Gear	Bormatec	1	150.00 €	150.00 €							
Counter sunk Machine screws x25	AHC Ltd	1	£3.49	£3.49	£3.49	£0.00	£0.00	£0.00	£3.49	Felicia	16/02/2017
String	K's Market	1	£2.34	£2.34	£2.34	£0.00	£0.00	£0.00	£2.34	Felicia	22/02/2017
10mm Carbon fibre tube - 1.2m	Easy Composites	2	£21.85	£43.70	£86.62	£18.42	£5.50	£0.00	£110.54	L.Winter	14/03/2017
8mm Carbon fibre tube - 1.2m	Easy Composites	2	£21.46	£42.92							
2x Brushless Outrunner Motor	4Max	1	£66.98	£66.98	£66.98	£0.00	£4.99	£0.00	£71.97	A. Hill	17/03/2017
Diamond sRH771 antenna	Radioworld	1	£25.95	£25.95	£25.95	£0.00	£4.49	£0.00	£30.44	A. Hill	15/03/2017
Rx 8 channel PPM receiver	Root	1	£26.99	£26.99	£26.99	£0.00	£0.00	£0.00	£26.99	A. Hill	14/03/2017
Euro/ Pound on 13/02/2017								Total	£938.49	Budget of Project	Funds left
									£860.00		-£78.49

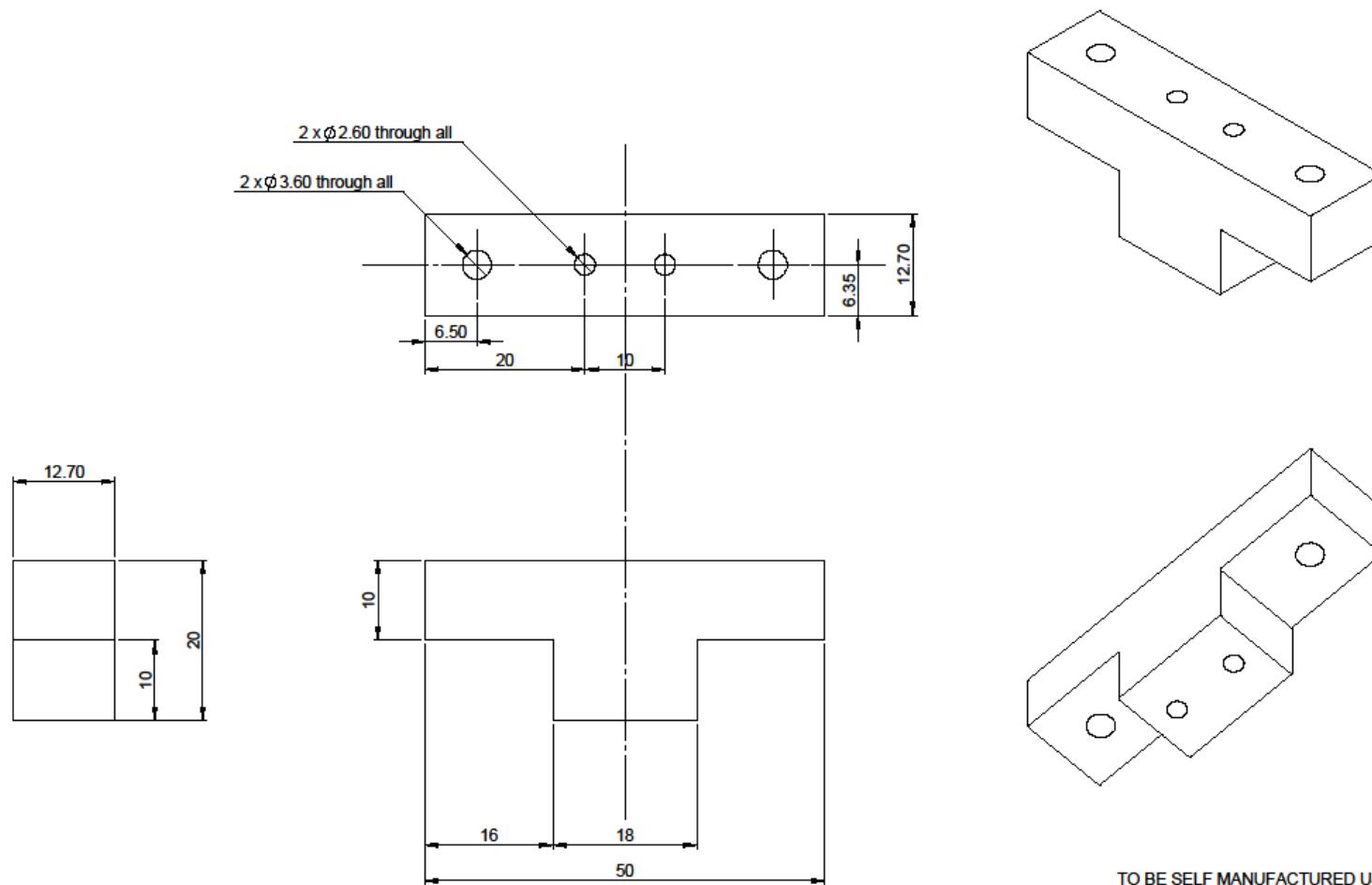
Appendix B



ITEM NO.	PARTS	DESCRIPTION	QTY.
1	Lift Arm	Request for manufacturing in EDMC	3
2	Height	Request for material only (Self-manufacturing)	5
3	Load Cell	Electronics Components	5
4	Bottom Plate	Request for material only (Self-manufacturing)	1
5	Drag Arm	Request for manufacturing in EDMC	2
6	Rigid Stand	Request for manufacturing in EDMC	2
7	Upper Plate	Request for material only (Self-manufacturing)	1
8	Bracket	Request for manufacturing in EDMC	2
9	Bearing Height	Request for material only (Self-manufacturing)	5
10	Pillow Bearing	Self-Purchase	10

SOLIDWORKS Educational Product. For Instructional Use Only

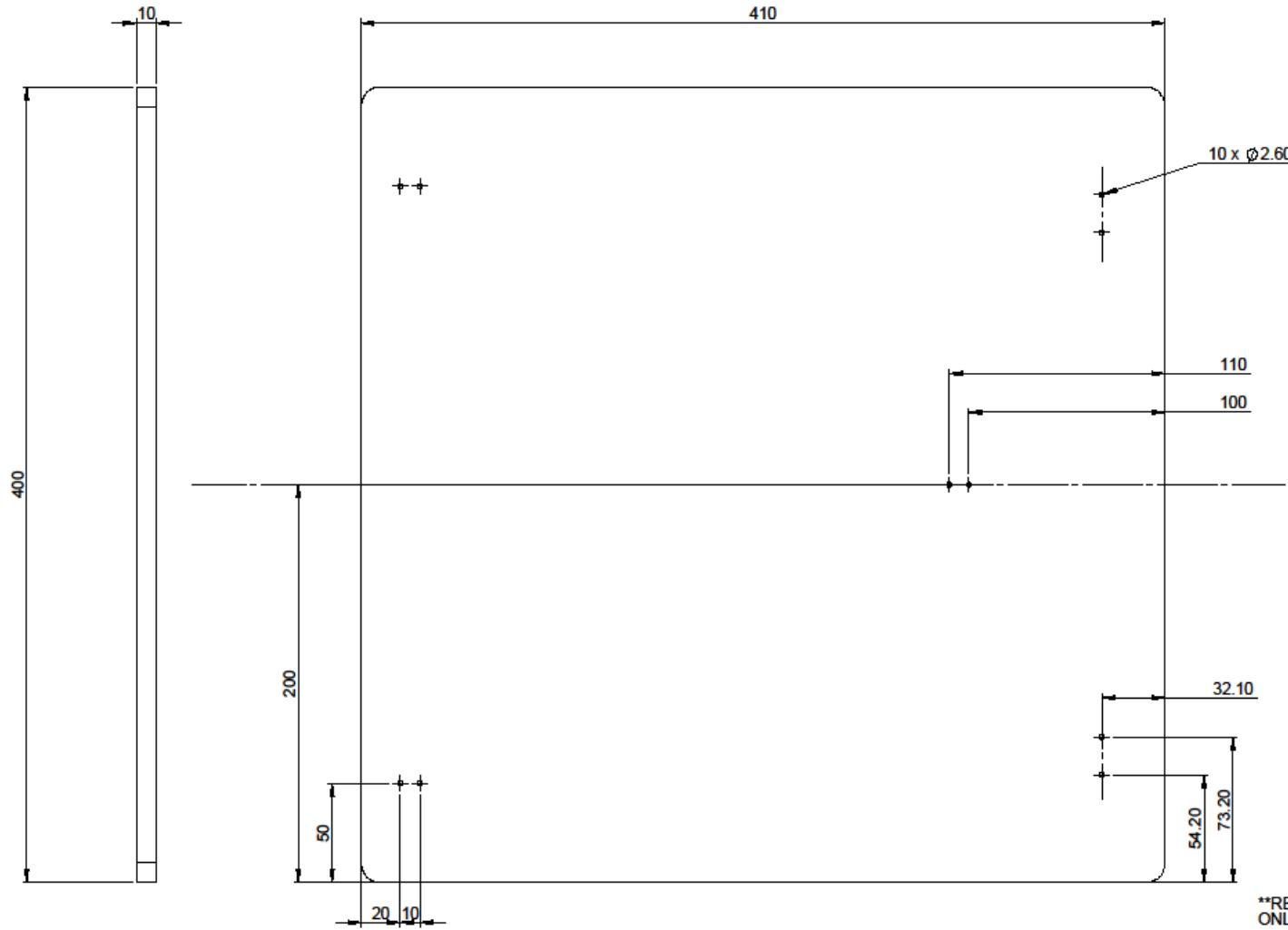
DO NOT SCALE		DRAWN BY FANG HWA TING		TOLERANCES UNLESS OTHERWISE STATED L=1±0.0mm 2.4±0.3mm 3.2±0.3mm 3.88±0.2mm		UNIVERSITY OF Southampton Faculty of Engineering and the Environment			
A3		DESIGNED BY FANG HWA TING	DATE 25 November 2016	SCALE 1:5	ANGULAR TOLERANCE 0.4° ± 0.25° ALL DIMENSIONS IN MM EXCEPT WHERE STATED				
EDMC JOB No	DEPARTMENT	PROJECT QDP 13 SMART WING UAV	SUPERVISOR DR T GLYN THOMAS	MATERIAL STEEL	SURFACE FINISH L/F ALL OVER UNLESS OTHERWISE STATED	TITLE Test Rig Assembly			
REMOVE ALL SHARP EDGES IF IN DOUBT PLEASE ASK						THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON DO NOT COPY WITHOUT WRITTEN PERMISSION.			
SHEET 1 of 1						NO OFF	ASSEMBLY NUMBER	DRAWING NUMBER	REVISION
						01	01	01	A



TO BE SELF MANUFACTURED USING 3D PRINTER

SOLIDWORKS Educational Product. For Instructional Use Only

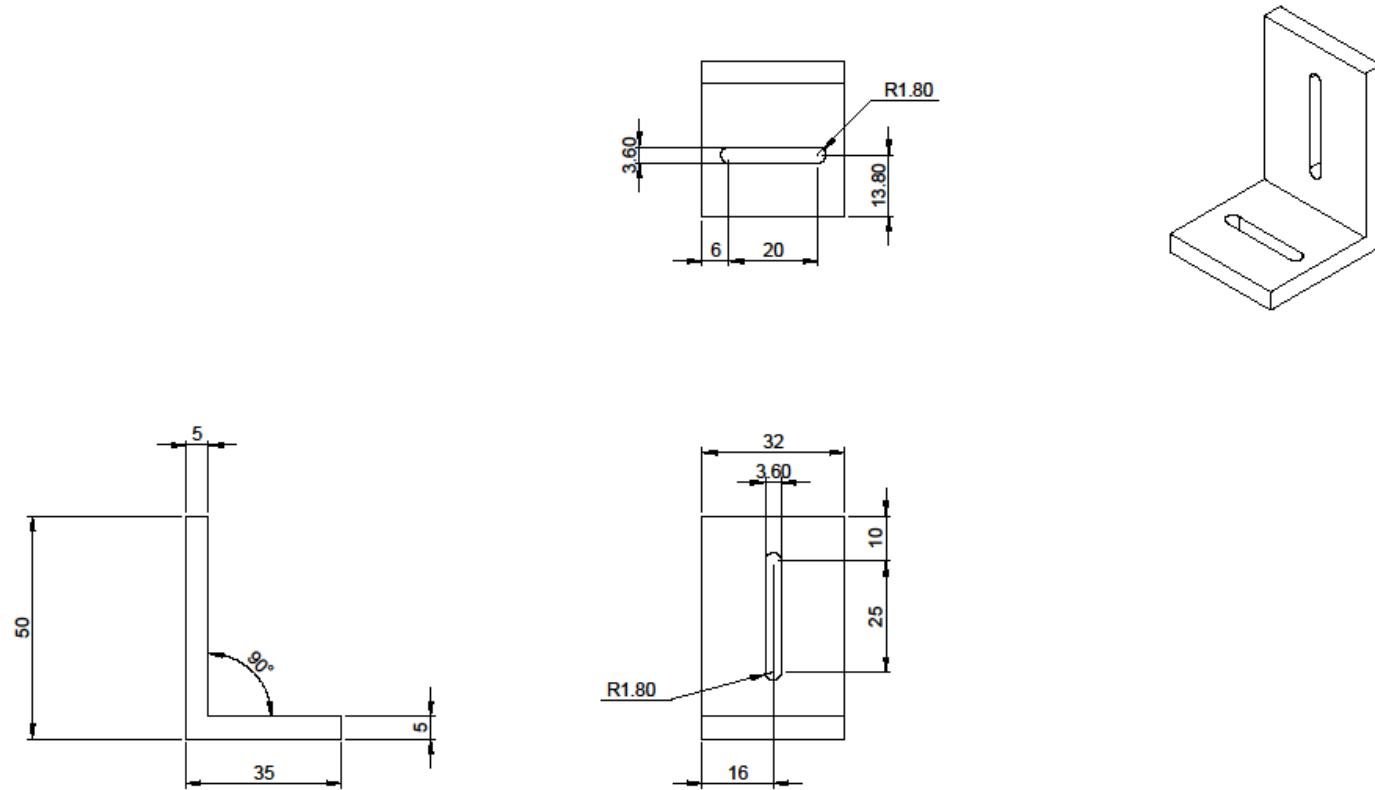
DO NOT SCALE		DRAWN BY FANG HWA TING		TOLERANCES UNLESS OTHERWISE STATED		UNLESS OTHERWISE STATED
A3		DESIGNED BY FANG HWA TING		ALL DIMENSIONS	IN MM ± 0.25mm	
EDMC JOB NO	DEPARTMENT	DATE	SCALE	ALL DIMENSIONS	IN MM ± 0.25mm	
GDP 13		26 NOVEMBER 2016	2:1	ALL OVER UNLESS STATED	IN MM ± 0.25mm	
SMART WING UAV	DR T GLYN THOMAS			SURFACE FINISH	ALL OVER UNLESS STATED	
REMOVE ALL SHARP EDGES		THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON. DO NOT COPY WITHOUT WRITTEN PERMISSION.				
IF IN DOUBT PLEASE ASK		SHEET	No OFF	ASSEMBLY NUMBER	DRAWING NUMBER	REVISION
		1 of 1	5	09		A



**REQUEST FOR MATERIAL
ONLY (SELF MANUFACTURE)

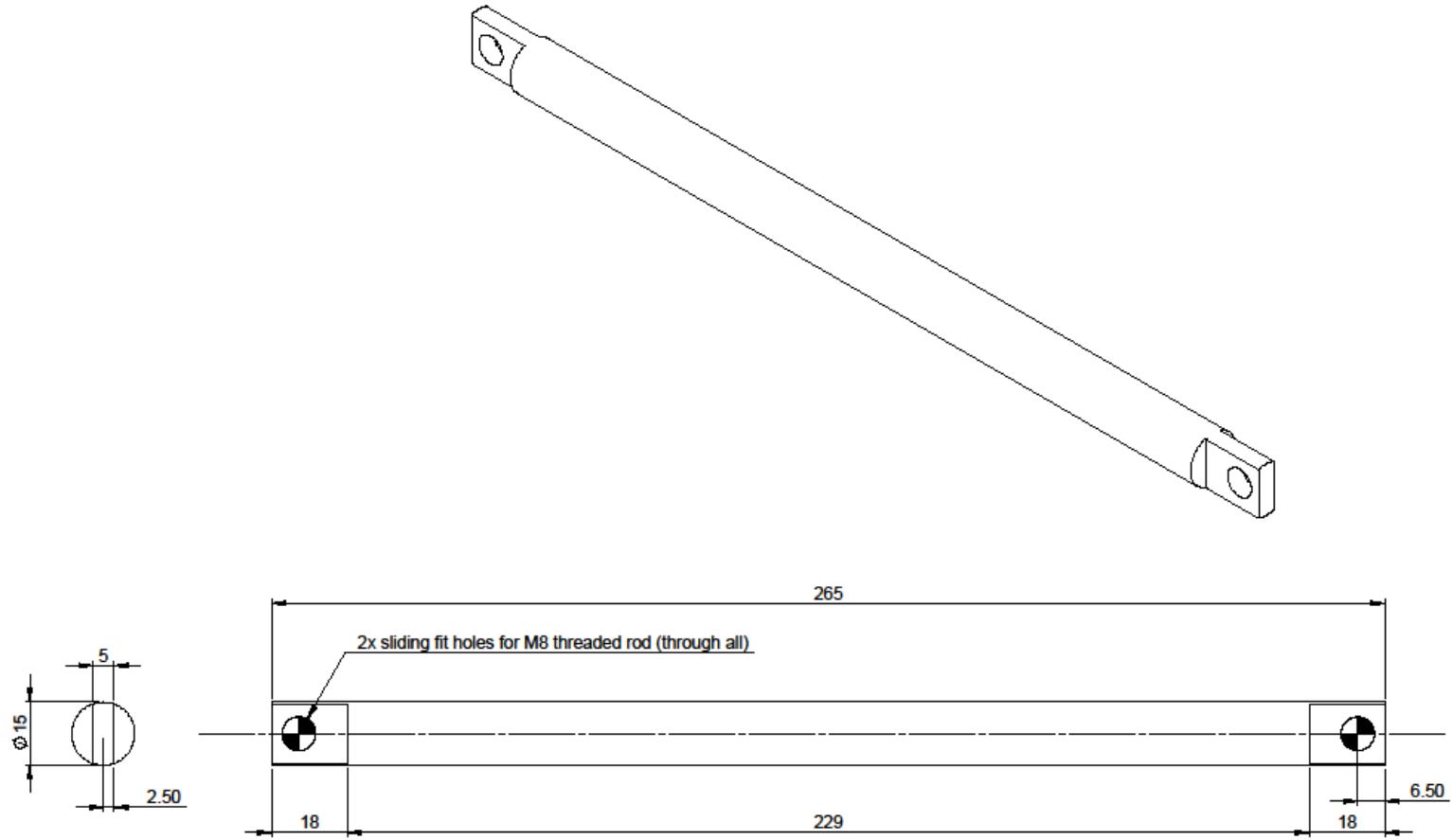
DO NOT SCALE		DRAWN BY FANG HWA TING		TOLERANCES UNLESS OTHERWISE STATED X = +/- 0.5mm Y = +/- 0.5mm Z = +/- 0.5mm		TITLE BOTTOM PLATE
A3		DESIGNED BY FANG HWA TING	DATE 25 NOVEMBER 2016	SCALE 1:2	ANGULAR DIMENSIONS X,Y,Z = +/- 0.5mm ALL DIMENSIONS IN mm UNLESS OTHERWISE STATED	
EDMC Job No	DEPARTMENT XXXX XXXX	PROJECT GDP 13 SMART WING UAV	SUPERVISOR DR T GLYN THOMAS	MATERIAL 10MM PERSPEX	TEXTURE	SURFACE FINISH 1.5° ALL OVER UNLESS OTHERWISE STATED
REMOVE ALL SHARP EDGES IF IN DOUBT PLEASE ASK	THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON DO NOT COPY WITHOUT WRITTEN PERMISSION.					

SOLIDWORKS Educational Product. For Instructional Use Only



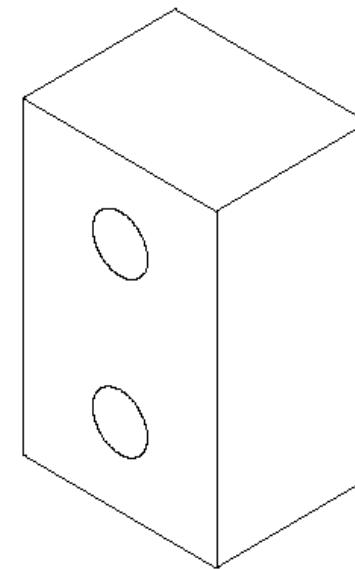
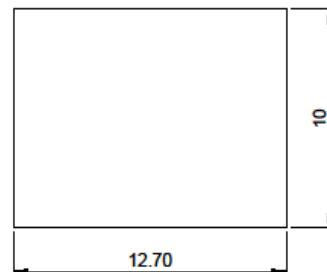
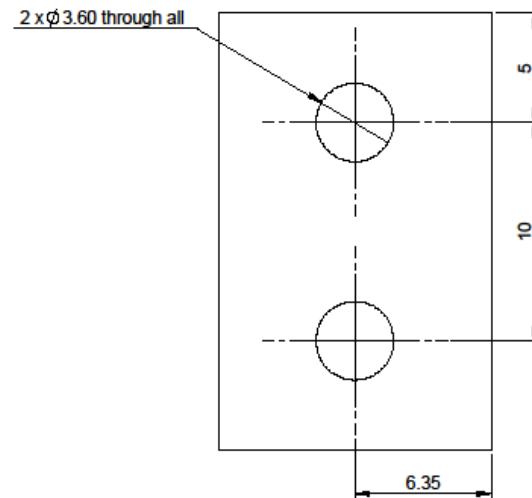
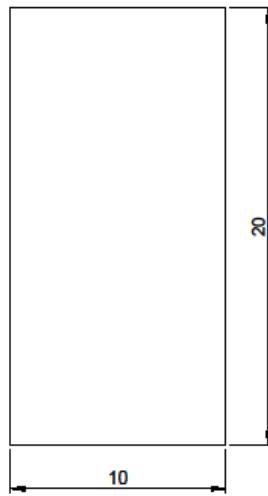
SOLIDWORKS Educational Product. For Instructional Use Only

DO NOT SCALE		DRAWN BY FANG HWA TING		TOLERANCES UNLESS OTHERWISE STATED LINEAR DIMENSIONS X ± 0.25mm Y ± 0.25mm Z ± 0.5mm		UNIVERSITY OF Southampton Faculty of Engineering and the Environment
A3		DESIGNED BY FANG HWA TING	DATE 25 NOVEMBER 2016	SCALE 1:1	ANGULAR DIMENSIONS X.X ± 0.25mm Y.Y ± 0.25mm Z.Z ± 0.5mm ALL DIMENSIONS IN mm UNLESS OTHERWISE STATED	
SDMC JOB No	DEPARTMENT					TITLE BRACKET
PROJECT GDP 13 SMART WING UAV	SUPERVISOR DR T GLYN THOMAS	MATERIAL CARBON STEEL	TEXTURE	SURFACE FINISH 1.5µ ALL OVER UNLESS OTHERWISE STATED		
REMOVE ALL SHARP EDGES	IF IN DOUBT PLEASE ASK	THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON DO NOT COPY WITHOUT WRITTEN PERMISSION.				
SHEET 1 of 1	No OFF 2	ASSEMBLY NUMBER 08	DRAWING NUMBER	REVISION A		



SOLIDWORKS Educational Product. For Instructional Use Only

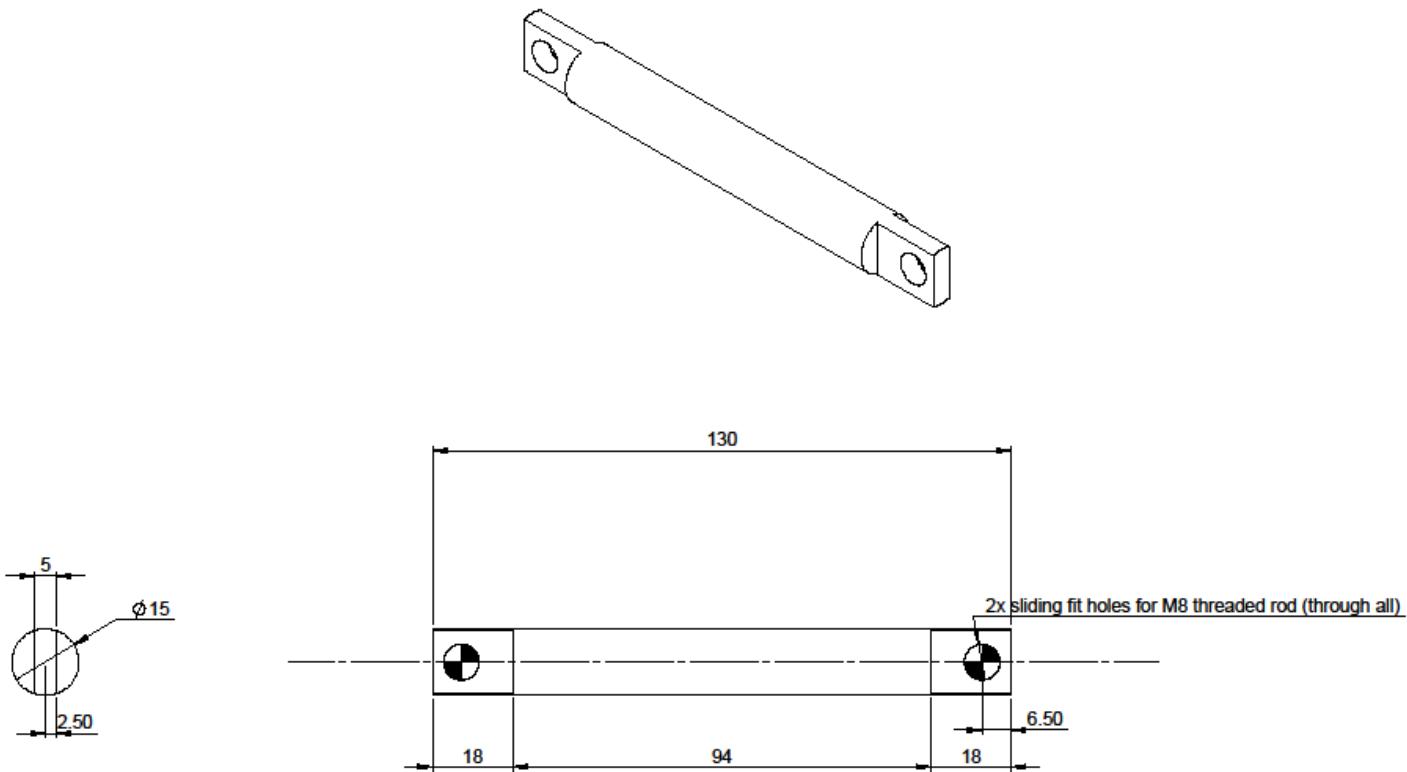
DO NOT SCALE		DRAWN BY FANG HWA TING		TOLERANCES UNLESS OTHERWISE STATED		UNIVERSITY OF Southampton	
A3		DESIGNED BY FANG HWA TING		LINEAR DIMENSIONS ± 0.5mm ∠ ± 0.5° XXX × ± 0.1mm		Faculty of Engineering and the Environment	
ECNC JOB No	DEPARTMENT	DATE	SCALE	ANGULAR DIMENSIONS ± 0.5° XXX × ± 0.2mm	ALL DIMENSIONS IN MM UNLESS OTHERWISE STATED	TITLE	DRAG ARM
GDP 13	DR T GLYN THOMAS	25 NOVEMBER 2016	1:1			SHEET	No OFF
SMART WING UAV				MATERIAL	TEXTURE	1 of 1	ASSEMBLY NUMBER
				CARBON STEEL	SURFACE FINISH	2	DRAWING NUMBER
					1.5 ALL OVER UNLESS OTHERWISE STATED		REVISION
REMOVE ALL SHARP EDGES		THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON DO NOT COPY WITHOUT WRITTEN PERMISSION.					
IF IN DOUBT PLEASE ASK							



TO BE SELF MANUFACTURED USING 3D PRINTER

SOLIDWORKS Educational Product. For Instructional Use Only

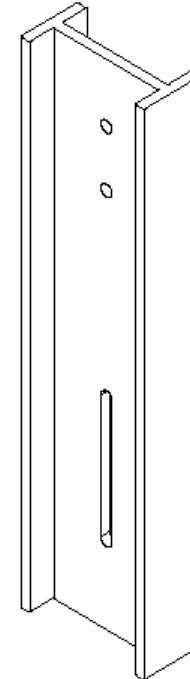
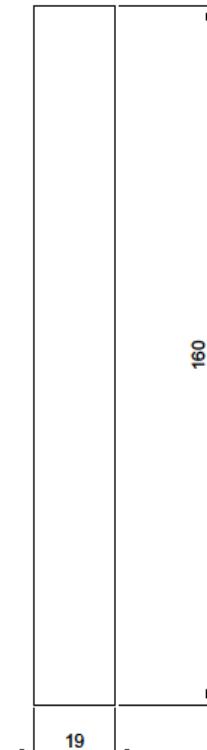
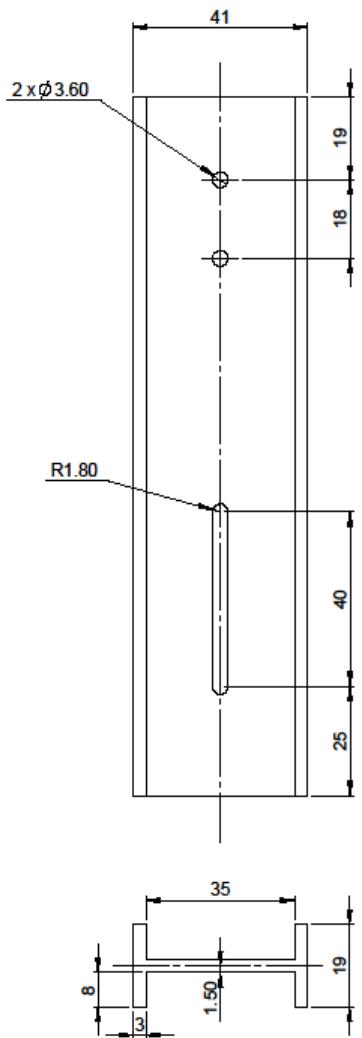
DO NOT SCALE		DRAWN BY FANG HWA TING	TOLERANCES UNLESS OTHERWISE STATED LINEAR DIMENSIONS ±0.1mm ±0.2mm ±0.3mm ±0.5mm ±0.8mm ±1.0mm ANGULAR DIMENSIONS ±0.5°mm ±1.0°mm ALL DIMENSIONS ARE IN MILLIMETERS UNLESS OTHERWISE STATED	UNIVERSITY OF Southampton Faculty of Engineering and the Environment	
A3		DESIGNED BY FANG HWA TING			
EDMC JOB No	DEPARTMENT	DATE 26 NOVEMBER 2016	SCALE 5:1		
PROJECT GDP 13 SMART WING UAV	SUPERVISOR DR T GLYN THOMAS	MATERIAL	TEXTURE	TITLE HEIGHT	
REMOVE ALL SHARP EDGES IF IN DOUBT PLEASE ASK		THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON. DO NOT COPY WITHOUT WRITTEN PERMISSION.		SHEET 1 of 1	NO OFF 5
		ASSEMBLY NUMBER 02		DRAWING NUMBER	REVISION A



SOLIDWORKS Educational Product. For Instructional Use Only

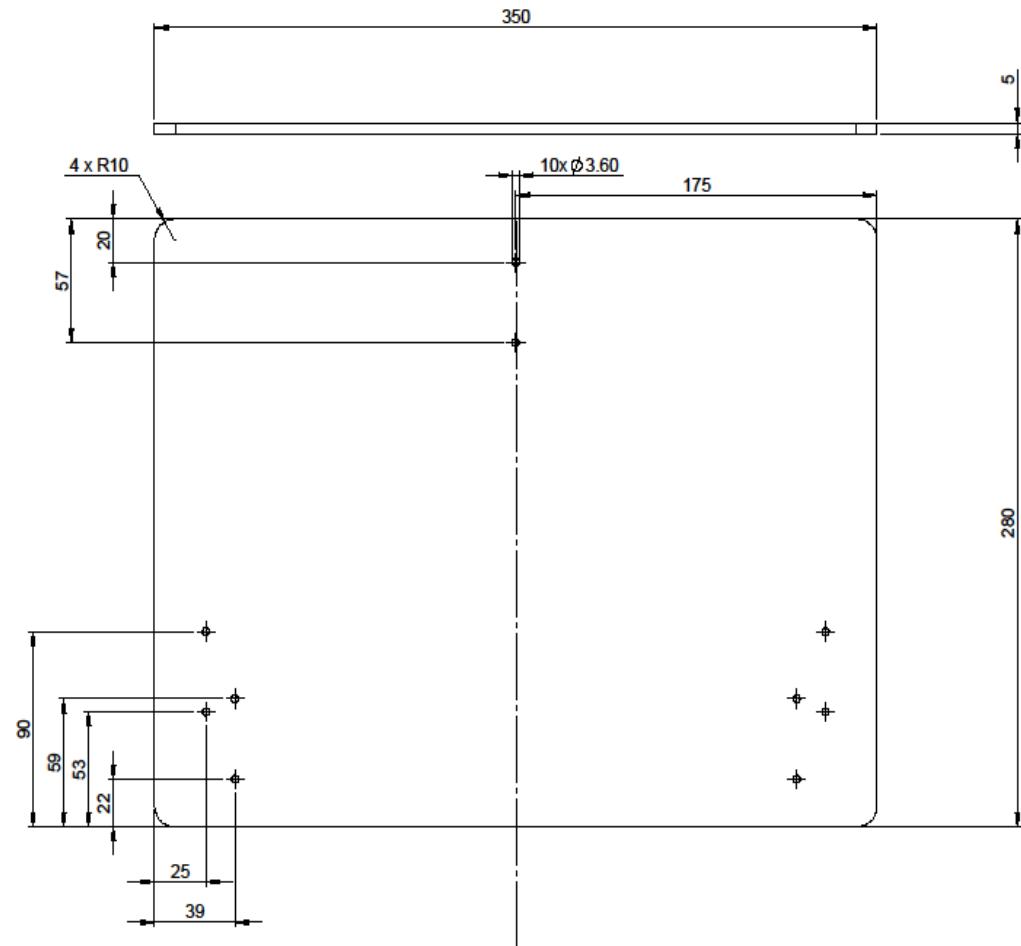
DO NOT SCALE		DRAWN BY FANG HWA TING		TOLERANCES UNLESS OTHERWISE STATED		UNLESS OTHERWISE STATED	
A3		DESIGNED BY FANG HWA TING		LINEAR DIMENSIONS		LINEAR DIMENSIONS	
ECNC JOB NO.	DEPARTMENT	DATE	SCALE	3.3 +0.0 -0.5mm	3.3 +0.0 -0.5mm	3.3 +0.0 -0.5mm	3.3 +0.0 -0.5mm
PROJECT GDP 13 SMART WING UAV	SUPERVISOR DR T GLYNN THOMAS	25 NOVEMBER 2016	1:1	Ø 4.0 ±0.5mm	Ø 4.0 ±0.5mm	Ø 4.0 ±0.5mm	Ø 4.0 ±0.5mm
REMOVE ALL SHARP EDGES IF IN DOUBT PLEASE ASK		THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON DO NOT COPY WITHOUT WRITTEN PERMISSION.					
SHEET 1 of 1		No OFF 3		ASSEMBLY NUMBER 01		DRAWING NUMBER REVISON A	

UNIVERSITY OF
Southampton
Faculty of Engineering and the Environment



SOLIDWORKS Educational Product. For Instructional Use Only

DO NOT SCALE		DRAWN BY FANG HWA TING		TOLERANCES UNLESS OTHERWISE STATED UNLESS DIMENSIONED 2.0 +/- 0.2mm 2.0 +/- 0.25mm 3.00 +/- 0.1mm		UNIVERSITY OF Southampton Faculty of Engineering and the Environment
EDMC JOB NO	DEPARTMENT	DESIGNED BY FANG HWA TING	DATE 25 NOVEMBER 2016	SCALE 1:1	ANALOGUE DIMENSIONS 2.0 +/- 0.25mm ALL OTHER DIMENSIONS UNLESS STATED	
PROJECT QDP 13 SMART WING UAV	SUPERVISOR DR T GLYN THOMAS	MATERIAL CARBON STEEL	TEXTURE	SURFACE FINISH 1.6 ALL OVER UNLESS OTHERWISE STATED	TITLE RIGID STAND	
REMOVE ALL SHARP EDGES IF IN DOUBT PLEASE ASK	The information contained in this document is the property of the University of Southampton. Do not copy without written permission.	1 of 1	No off	Assembly number 6	Drawing number A	Revision A

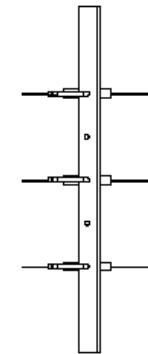
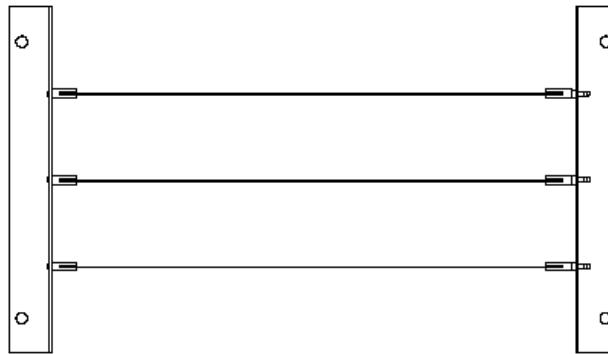
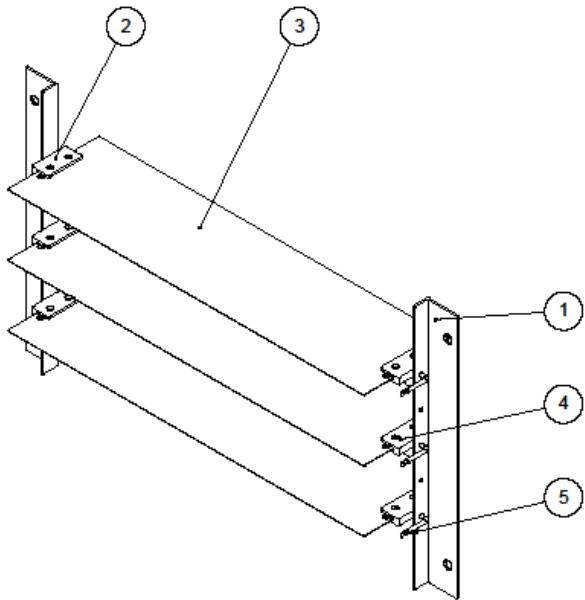


**REQUEST FOR MATERIAL
ONLY (SELF MANUFACTURE)

SOLIDWORKS Educational Product. For Instructional Use Only

DO NOT SCALE		DRAWN BY FANG HWA TING	TOLERANCES UNLESS OTHERWISE STATED		UNIVERSITY OF	
A3		DESIGNED BY FANG HWA TING	DATE 25 NOVEMBER 2015	SCALE 1:2	Southampton	
ECOM JOB NO	DEPARTMENT				Faculty of Engineering and the Environment	
PROJECT GDP 13 SMART WING UAV	SUPERVISOR DR T GLYNN THOMAS	MATERIAL 5MM PERSPEX	TEXTURE 	ANGLE TOLERANCES 2 = 0.5mm 3 = 0.5mm ALL DIMENSIONS IN mm UNLESS OTHERWISE STATED	TITLE UPPER PLATE	
REMOVE ALL SHARP EDGES IF IN DOUBT PLEASE ASK		THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON. DO NOT COPY WITHOUT WRITTEN PERMISSION.				
SHEET 1 of 1		NO OFF 1	ASSEMBLY NUMBER 07	DRAWING NUMBER A	REVISION A	

Appendix C



ITEM NO.	PART	DESCRIPTION	QTY.
1	BlindSupport	3mm unequal angle, 20x50x400mm	2
2	EndLeft	28x56x10mm block	3
3	Flap	150x583mm, 1mm sheet metal	3
4	EndRight	28x56x10mm block	3
5	Rod	Ø 5mm bent rod, L = 60mm	3

DO NOT SCALE	DRAWN BY	TOLERANCES UNLESS OTHERWISE STATED	
A3	NDT	UNLESS OTHERWISE STATED	
		X ± 0.5mm	X ± 0.5mm
		X ± 0.2mm	X ± 0.2mm
		X ± 0.1mm	X ± 0.1mm

EDM JOB No.	DEPARTMENT	DATE	SCALE	ANGULAR DIMENSIONS
N/A	Engineering and the Environment	9/1/2017	1:5	All dimensions in mm unless otherwise stated
PROJECT	SUPERVISOR	MATERIAL	TEXTURE	SURFACE FINISH
GDP 13	Glyn Thomas	-	-	1.6° ALL OVER UNLESS OTHERWISE STATED
REMOVE ALL SHARP EDGES				THE INFORMATION CONTAINED IN THIS DOCUMENT IS THE PROPERTY OF THE UNIVERSITY OF SOUTHAMPTON DO NOT COPY WITHOUT WRITTEN PERMISSION.
IF IN DOUBT PLEASE ASK				

TITLE				
Flap Assembly				
SHEET	No OFF	ASSEMBLY NUMBER	DRAWING NUMBER	REVISION
1 of 6	XX	XXX/XXX	-	A

SOLIDWORKS Educational Product. For Instructional Use Only

UNIVERSITY OF
Southampton
Faculty of Engineering and the Environment

Appendix D

Assessment																	
Title of risk assessment		FEEG6013 - Group Design Project: Design and Manufacture of a Smart Wing UAV															
Risk Acceptability				Risk Matrix		Severity					Likelihood	Overall Likelihood	Overall Severity	Residual Risk score	Any changes or extra controls?		
1-3	Risk acceptable			Probability	certainty	very low	low	medium	high	very high							
4-6	Risk to be reduced if readily possible					1	2	3	4	5							
7-14	Risk to be reduced if reasonably practicable					certainty	5	5	10	15	20						
15-25	Risk unacceptable					likely	4	4	8	12	16						
						possible	3	3	6	9	12						
						less likely	2	2	4	6	8						
						improbable	1	1	2	3	4						
ref	Task/Aspect of work	Hazard	Harm and how it could arise	Who could be affected?		Existing measures to control risk					Risk Factors						
	Test flight of the UAV	Unstabilise flight of UAV	Crash of UAV	Anyone around the UAV		Test the UAV in an open outdoor area which is far away from people.					3	3	9				
	Use of Lithium ion batteries	Overheating	Catching fire/explosion	Anyone around the batteries		Strictly follow the user manual and precaution given by the manufacturer of Lithium ion batteries.					2	5	10				
	Assembly of electronic control system.	Use of electrical equipment.	Electrical shock/burn.	Students working on the electrical systems.		Electrical equipment disconnected from power when in use. No loose wires.					1	2	2				
	Wing Tunnel testing	Wind Tunnel exhaust and noise.	Could cause injury to people caught in the wind tunnel exhaust at high airspeeds. Noise could effect hearing.	People in the vicinity of the wind tunnel.		Put a warning sign in place when wind tunnel is in use. Stand in a safe area away from the exhaust. Wear ear protection if necessary.					2	2	4				
	Simulation, programming and computational modelling.	Prolonged duration in same position.	Psychological impact, fatigue, aches.	Students using computational equipment.		Take a short break after every hour. Avoid working alone for prolonged duration. Ensure room is well lit and chair is properly adjusted.					3	1	3				
	Attaching Wing Section(s) to Test Rig	Using hand tools.	Manual usage of hand tools can cause aches and strains.	Anyone operating the hand tools.		Ensure tools are in adequate condition. Ensure the correct tools are used for each job.					2	1	2				

	Mounting Test Rig in Wind Tunnel	Using hand tools.	Manual usage of hand tools can cause aches and strains.	Anyone operating the hand tools.	Ensure tools are in adequate condition. Ensure the correct tools are used for each job.	2	1	2	
	Assembly of test rig and wing sections.	Using hand tools.	Manual usage of hand tools can cause aches and strains.	Anyone operating the hand tools.	Ensure tools are in adequate condition. Ensure the correct tools are used for each job.	2	1	2	
	Cutting material for use in test rig and wing sections.	Machinery and equipment hazards.	Crushing, trapping, cuts and bruises, amputation.	Anyone operating the machinery.	Ensure proper supervision. Avoid loose clothing and jewellery with long hair tied back. Ensure machinery is in working and set up properly for intended use. Wear safety goggles and equipment.	2	4	8	

Appendix E

Teaching week:	1	2	3	4	5	6	7	8	9	10	11	Christmas holidays	12	Ex	Ex	1	2	3	4	5	6	7	8	Easter holidays					
Week 40 Beginning Monday October 3, 2016	Wk 40	Wk 41	Wk 42	Wk 43	Wk 44	Wk 45	Wk 46	Wk 47	Wk 48	Wk 49	Wk 50	Wk 51	Wk 52	1	2	3	4	5	6	7	8	9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Wk 15	Wk 16
Task:																													
Meeting with supervisors																													
Draft a full Gantt chart																													
Define roles and distribute tasks among team members																													
Prepare presentation																													
Complete risk assessment																													
Submit risk assessment																													
Presentation at research group level to receive feedback																													
Complete form P2																													
Simulation																													
Sketchings of test rig design/mechanism																													
Ask for guideline and checklist from EDMC before starting 3D CAD modelling																													
Basic control system design																													
Decide aerofoil to be use and locations to place pressure sensors																													
3D CAD modelling for test rig																													
Discuss with supervisors the feedback received after the presentation																													
Connect sensors and motors to Arduino																													
Submit form P2																													
Purchase sensors and relevant devices																													
Design and 3D CAD modelling for UAV																													
Submit 3D CAD models to EDMC for test rig parts for manufacturing																													
Assembly of test rig																													
Experimental testing on 2D model in wind tunnel																													
Submit 3D CAD models to EDMC for UAV parts for manufacturing																													
Complete form P3																													
Submit form P3																													
Complete Peer Feedback form (1)																													
Assembly of UAV																													
Pressure Control system onto the UAV																													
Experimental testing on UAV																													
Report write-up																													
Submit final report																													
Prepare presentation																													
Design/Manufacturing members																													
Coding/System design members																													
All members																													

 * expected
 * completed
 * delayed

Appendix F

smoothingSpline.m

```
function [coeffs,j] = smoothingspline( x, y, sig, lmda)
```

Initialise Variables

```
%size of xarray
sizex = size(x,1);
mu = 2 * (1 - lmda)/(3 * lmda);
maxi1 = 1;
maxy1 = y(1);

maxi2 = 2;
maxy2 = y(2);
```

Find Max

Spline fit points with highest Cp

```
for i = 1:n+1
    if (y(i) > maxy1)
        maxy2 = maxy1;
        maxi2 = maxi1;
        maxy1 = y(i);
        maxi1 = i;
    elseif (y(i) > maxy2)
        maxy2 = y(i);
        maxi2 = i;
    end
end
if (maxi1 < maxi2)
    j = maxi1;

else
    j = maxi2;
end
```

Create and Estimate Diagonals of the B Estimate Matrix

n: number of samples

u: vector representing the main diagonal

v: vector representing the second diagonal symmetric about u

w: vector representing the third diagonal symmetric about u

q: y values (RHS of equation).

```
n = sizex - 1;

h = zeros(n,1);

u = sym(zeros(n,1));

v = sym(zeros(n-1,1));

w = sym(zeros(n-1,1));

q = sym(zeros(sizex,1));

h(1) = x(2) - x(1);

h(2) = x(3) - x(2);

%pad arrays
u(1) = 0;
v(1) = 0;
w(1) = 0;

%set b = qs first and last element to 0
q(1) = 0;
q(sizex) = 0;

%iterate from 2 to n-1
for i = 2:1:n-1
    h(i + 1) = x(i + 2) - x(i + 1);
    u(i) = (9 * sig(i-1)/(h(i-1) * h(i-1))) + (9 * sig(i) * (h(i) ...
        + h(i-1)) * (h(i) + h(i-1))/(h(i)*h(i-1)*h(i)*h(i-1))) ...
        + (9 * sig(i+1)/(h(i)*h(i)));
    u(i) = (mu * u(i)) + (2 * (x(i+1) - x(i-1)));

    v(i) = (-9 * sig(i) * (h(i) + h(i-1))/(h(i) * h(i) * h(i-1))) ...
        + (-9 * sig(i+1) * (h(i+1) + h(i))/(h(i+1) * h(i) * h(i)));
    v(i) = mu*v(i) + h(i);
```

```

w(i) = mu * 9 * sig(i+1)/(h(i) * h(i+1));
q(i) = (3 * (y(i+1) - y(i))/h(i)) - (3 * (y(i) - y(i-1))/h(i-1));
end
u(n) = (9 * sig(n-1)/(h(n-1) * h(n-1))) + (9 * sig(n) * (h(n) ...
+ h(n-1)) * (h(n) + h(n-1))/(h(n)*h(n-1)*h(n)*h(n-1))) ...
+ (9 * sig(n+1)/(h(n)*h(n)));
u(n) = (mu * u(n)) + (2 * (x(n+1) - x(n-1)));
q(n) = (3 * (y(n+1) - y(n))/h(n)) - (3 * (y(n) - y(n-1))/h(n-1));

disp('Estimating b');
b = calcB(n, u, v, w, q);

disp('calculating rest of the parameters ...')

```

Calculate the coefficients for only jth cubic

Estimate all other coefficients

```

if (j == 1)
    d = y(j) - (mu * 3 * b(2) * sig(j)/h(1));
    d2 = (-3 * b(2) * (h(1) + h(2))/(h(1)*h(2))) + (3 * b(3)/h(2));
    d2 = y(j+1) - (mu * d2 * sig(j));
    a = b(2)/(3*h(1));
    c = ((d2 - d)/h(1)) - (b(2) * h(1)/3);
elseif (j < n)
    a = (b(j + 1) - b(j))/(3 * h(j));

    d = (3 * b(j-1)/h(j-1)) + (-3 * b(j) * (h(j-1) ...
        + h(j))/(h(j-1)*h(j))) + (3 * b(j+1)/h(j));
    d = y(j) - (mu * d * sig(j));

    d2 = (3 * b(j)/h(j)) + (-3 * b(j+1) * (h(j)...
        + h(j+1))/(h(j)*h(j+1))) + (3 * b(j+2)/h(j+1));
    d2 = y(j+1) - (mu * d2 * sig(j+1));

    c = ((d2 - d)/h(j)) - ((b(j+1) + (2*b(j))) * h(j)/3);
end

coeffs = [a,b(j),c, d];
end

```

Appendix G

Send Load Cell Data to Matlab – arduinoCalib.ino

This is the calibration_send() function from arduinoCalib.ino

arduinoCalib.ino available at https://bitbucket.org/smartwinggdp/smartwing_gdp1617

```
1. /*
2.  * Synchronises matlab and arduino serial
3.  * to prevent overflow
4.  * Synchronisation done by acknowledgement chars
5.  * Arduino prints data -> matlab reads when it detects
6.  * presence of bytes in serial buffer
7.  * Acknowledgement procedure:
8.  * - Arduino waits for matlab acknowledgement
9.  * - When available -> read
10. * - If unexpected char -> light Red LED
11. *     else blink green LED twice
12. */
13. void calibration_send(){
14. //Send offset values
15. //Blink green led and red led 3 times each
16. //indicates start of calibration routine
17. blinkled(ledPin, 3, 200);
18. blinkled(redledPin, 3, 200);
19. volatile byte runflag = HIGH;
20.
21. char val = 'a';
22. Serial.println(l1.read_average(10));
23. Serial.println(l2.read_average(10));
24. Serial.println(l3.read_average(10));
25. Serial.println(d1.read_average(10));
26. Serial.println(d2.read_average(10));
27.
28. //Waits for matlab to read all data
29. // and acknowledge
30. // acknowledgement char = 'c'
31. while (Serial.available() == 0){
32. }
33. if (Serial.available() > 0){
34.     val = Serial.read();
35.     if (val == 'c'){
36.         //data received as expected
37.         //blink green twice
38.         blinkled(ledPin, 2, 300);
39.     }
40.     else {
41.         //Unexpected output -> Something went wrong
42.         digitalWrite(redledPin, HIGH);
43.     }
44. }
45. }
46. //Green led lights up to indicate ready for next stage
47. digitalWrite(ledPin, HIGH);
48. //Send data for each loading condition
```

```

49. //for (int i = 0; i < N; i++){
50.   while (runflag == HIGH){
51.     digitalWrite(ledPin, HIGH);
52.     state = digitalRead(switchPin);
53.     //Halt till load is placed
54.     while (digitalRead(switchPin) == state){
55.       }
56.     delay(3000);
57.     //Switch toggled -> load placed -> green led goes off
58.     digitalWrite(ledPin, LOW);
59.
60.     //send data
61.     Serial.println(l1.read_average(10));
62.     Serial.println(l2.read_average(10));
63.     Serial.println(l3.read_average(10));
64.     Serial.println(d1.read_average(10));
65.     Serial.println(d2.read_average(10));
66.
67.     //before proceeding check if data is received by matlab
68.     while (Serial.available() == 0){
69.       }
70.     if (Serial.available() > 0){
71.       val = Serial.read();
72.       if (val == 'g'){
73.         //blink green twice
74.         blinkled(ledPin, 2, 300);
75.       }
76.       else if (val == 'e'){
77.         runflag = LOW;
78.       }
79.       else {
80.         //Received something but it's not expected
81.         digitalWrite(redledPin, HIGH);
82.         //abort
83.       }
84.     }
85.   }
86.   blinkled(redledPin, 3, 300);
87.   blinkled(ledPin, 3, 300);
88. }
```

Receive Load Cell Data from Arduino – dataAcquisitionCalibration.m

Run this Matlab Script to begin calibration.

Establish Serial Communication with Arduino

Run this script to get data from load cells via Arduino

```
% Set comPort to whichever port Arduino is connected to
comPort = '/dev/tty.usbmodem1421';
[Arduino, flag] = setupSerial(comPort);
```

Initialise Parameters

```
% Loading Conditions
calibrationPerms
L_loc = Lift_location;
Y = Test_Cases;
N = size(Y, 1);

%unit row and 6 columns, one for each tranducer
R = zeros(N, 6);
%set unit row
R(:, 1) = 1;
offsets = zeros(1, 5);

%find offsets for the data collected
disp('getting offsets for transducers');
offsets(1,1) = readSerial(Arduino, '%d');
offsets(1,2) = readSerial(Arduino, '%d');
offsets(1,3) = readSerial(Arduino, '%d');
offsets(1,4) = readSerial(Arduino, '%d');
offsets(1,5) = readSerial(Arduino, '%d');
%indicate to arduino that data has been received
fprintf(Arduino, '%c', 'c');
```

Estimate R matrix

```
for p = 1:1:N
    %display which loading condition
    disp(sprintf('Loading condition %d ; Lift = %f, Drag = %f , pitching moment = %f,
    Lift_location = %f cm', p , Y(p,:), L_loc(p)));
    R(p,2) = readSerial(Arduino, '%d');
    R(p,3) = readSerial(Arduino, '%d');
    R(p,4) = readSerial(Arduino, '%d');
    R(p,5) = readSerial(Arduino, '%d');
    R(p,6) = readSerial(Arduino, '%d');
    if (p == N) %indicate to arduino that last data has been sent
        fprintf(Arduino, '%c', 'e');
    else %indicate to arduino that data has been received
        fprintf(Arduino, '%c', 'g');
    end
end
```

Save Data and Cleanup

```
save('calib_data100317_3.mat', 'R', 'Y', 'offsets');

fclose(Arduino);
delete(Arduino);
```

Functions Called

available at https://bitbucket.org/smartwinggdp/smartwing_gdp1617

- setupSerial.m is used to set up serial communication with an Arduino
- readSerial.m is used to read Serial data from a connected Arduino
- calibrationPerms.m generates a matrix of mass of the weights which have to be placed and where they have to be placed.

Loading Conditions

The following combinations of loading conditions were applied

Loading Condition	Lift Weight (grams)	Drag Weight (grams)	Lift load application Point (cm)	Resulting Pitching Moment milli Nm
1	36	0	0	0
2	36	200	0	0
3	36	700	0	0
4	36	1000	0	0
5	2036	0	-2.5034	-500
6	2036	0	0	0
7	2036	0	2.5034	500
8	2036	200	-2.5034	-500
9	2036	200	0	0
10	2036	200	2.5034	500
11	2036	700	-2.5034	-500
12	2036	700	0	0
13	2036	700	2.5034	500
14	2036	1000	-2.5034	-500
15	2036	1000	0	0
16	2036	1000	2.5034	500
17	4036	0	-1.2628	-500
18	4036	0	0	0

19	4036	0	1.2628	500
20	4036	200	-1.2628	-500
21	4036	200	0	0
22	4036	200	1.2628	500
23	4036	700	-1.2628	-500
24	4036	700	0	0
25	4036	700	1.2628	500
26	4036	1000	-1.2628	-500
27	4036	1000	0	0
28	4036	1000	1.2628	500
29	6036	0	-0.84441	-500
30	6036	0	0	0
31	6036	0	0.84441	500
32	6036	200	-0.84441	-500
33	6036	200	0	0
34	6036	200	0.84441	500
35	6036	700	-0.84441	-500
36	6036	700	0	0
37	6036	700	0.84441	500
38	6036	1000	-0.84441	-500
39	6036	1000	0	0
40	6036	1000	0.84441	500

Perform Calibration using Acquired Data – calibration_proc.m

Perform calibration to get coefficient matrices CI, CVw, CW1 using three sets of calibration data acquired from the test rig (calib_data100317_*.mat)

Full script available at https://bitbucket.org/smartwinggdp/smartwing_gdp1617

Output struct of calibration matrices and associated standard deviations

Author: Roshan Pasupathy

Based on: 'Design of a Six-component External Wing tunnel balance' - M. Ferreira

URL: <https://fenix.tecnico.ulisboa.pt/downloadFile/844820067124978/Dissertation.pdf> last accessed [March 4th 2017]

Load file which contains calibration parameters

```
Refdat = load('calib_data100317_1.mat');
C_struct = struct;
R = Refdat.R;
Y = Refdat.Y;
N = size(R,1);
I = eye(N);
```

Take I as Weighting matrix, calculate CI

```
CI = calc_CI(R, I, Y);

error = Y - (R*CI);

std_dev = calc_std_dev(error);

value={'No weighting';
       'Calib Matrix';
       CI,
       'Error';
       std_dev;
       };
C_struct.CI = value;
disp(sprintf('Lift error: %2.5f , Drag error: %2.5f , Moment error: %2.5f', std_dev));
CIstd_dev = std_dev;
```

Weighting matrix = Vw inverse. Calculate CVw

```
Vwl = calc_Vw(std_dev(1));

Vwd = calc_Vw(std_dev(2)); %eye(N)
Vwpm = calc_Vw(std_dev(3)); %eye(N)
```

```

CVw = zeros(6,3);
CVw(:,1) = calc_C(R, inv(Vwl), Y(:,1));
CVw(:,2) = calc_C(R, inv(Vwd), Y(:,2));
CVw(:,3) = calc_C(R, inv(Vwpm), Y(:,3));

error = Y - (R * CVw);
std_dev = calc_std_dev(error);

value={'Vw inverse weighting';
'Calib Matrix';
CVw;
'Error';
std_dev;
};

disp(sprintf('CVw Lift error: %2.5f , Drag error: %2.5f , Moment error: %2.5f', std_dev));
C_struct.CVw = value;

```

Calculate CW1

```

disp(sprintf('To proceed further, >3 calibrations of %d loading conditions are required',
N));
runflag = input('Proceed? y or n?', 's');
i = 0;
if (runflag == 'y')||(runflag == 'Y')
    %make three columns of 6N data,
    % Data 1  Data 2  Data 3
    % trans11  trans11  trans11
    % ...      ...      ...
    % trans1N  trans1N  trans1N
    % ...      ...      ...
    % trans6N  trans6N  trans6N
read1 = load('calib_data100317_1.mat');
read2 = load('calib_data100317_2.mat');
read3 = load('calib_data100317_3.mat');

%check if loading conditions were the same
if (any(any((read1.Y ~= read2.Y) + (read2.Y ~= read3.Y))))
    disp('Loading conditions do not match for calib data');
else
    Rhist = [];
    Rhist = [Rhist read1.R(:)];
    %Rhist = [Rhist read2.R(:)];
    Rhist = [Rhist read3.R(:)];
    Vr = cov(Rhist');
    C_mati_1 = CVw;
    while ((runflag == 'y')||(runflag == 'Y'))
        i = i + 1; % new iteration
        [C_mati, std_dev] = calc_CW(R, Y, C_mati_1, Vr);
        field = sprintf('CW%d', i);

```

```

        value{1} = sprintf('Regression number %d', i);
        value{3} = C_mati;
        value{5} = std_dev;
        C_struct.(field) = value;
        disp(sprintf('Lift error for %s : %2.5f , Drag error: %2.5f , Moment
error: %2.5f','CI', CIstd_dev));
        disp(sprintf('Lift error for %s : %2.5f , Drag error: %2.5f , Moment
error: %2.5f',field, std_dev));
        runflag = input('Proceed? y or n?', 's');
        C_mati_1 = C_mati;
    end
end
end

```

Display latest iteration info.

```

disp(fieldnames(C_struct));

fields = fieldnames(C_struct);

disp(i);

celldisp(C_struct.(fields{2 + i}));

end
disp('**** Available Calibrations ****');
disp(fieldnames(C_struct));

```

Appendix H

Automated calibration sections from WindTunnel.m

For full code refer to https://bitbucket.org/smartwinggdp/smartwing_gdp1617

Begin Wind tunnel test

```
% Press 'enter' to proceed
startin = input('Start?');

R = zeros(samples*flap_angles*AoAs, 6);
R(:,1) = 1;
Force_mat = zeros(samples*flap_angles*AoAs, 5);
% Record flap angles and AoAs combination
for aoa_i = 1:1:AoAs
    for fa_i = 1:1:flap_angles
        for smp = 1:1:samples
            Force_mat((aoa_i-1)*samples*flap_angles + (fa_i-1)*samples + smp,4) =
AoA_arr(aoa_i);
            Force_mat((aoa_i-1)*samples*flap_angles + (fa_i-1)*samples + smp,5) =
flap_arr(fa_i);
        end
    end
end

% Acquire data for each flap angle, Aoa combination
for aoa_i = 1:1:AoAs
    %send command to servo arduino : w<angle>
    fprintf(ServoArd, 'w%3s', int2str(AoA_arr(aoa_i)));

    for fa_i = 1:1:flap_angles
        %send command to servo arduino : f<angle>
        fprintf(ServoArd, 'f%3s', int2str(flap_arr(fa_i)));
        %tell arduino to wait
        fprintf(Arduino, '%c', 'd');
        for smp = 1:1:samples
            % Request data for sample
            fprintf(Arduino, '%c', 'r');
            current_it = (aoa_i-1)*samples*flap_angles + (fa_i-1)*samples + smp;
            R(current_it,2) = readSerial(Arduino, '%d');
            R(current_it,3) = readSerial(Arduino, '%d');
            R(current_it,4) = readSerial(Arduino, '%d');
            R(current_it,5) = readSerial(Arduino, '%d');
            R(current_it,6) = readSerial(Arduino, '%d');
            Force_mat(current_it,1:3) = R(current_it,:)* C;
            disp(sprintf('LIFT = %.6f g, DRAG = %.6f g, PM = %.6f gm, Aoa = %d, Flap
angle = %d', Force_mat(current_it,:)));
            %received all data for this sample > request new sample
        end
    end
end
```

```

    end
end

% Convert Units
Force_mat(:,1:2) = Force_mat(:,1:2) * 0.001 * 9.81;
Force_mat(:,3) = Force_mat(:,3) * 0.001;

```

Save data and Clean up

```

% Tell Test Rig Arduino to Stop
fprintf(Arduino, '%c', 'e');

% Reset Wing section to 0, 0
fprintf(ServoArd, '%s', 'f000');
pause(4);
fprintf(ServoArd, '%s', 'w000');

save('Wind_tunnel_lc_data.mat', 'R', 'R_no_wind');
save('Wind_tunnel_force_data.mat', 'Force_mat', 'F_no_wind');

fclose(ServoArd);
delete(ServoArd);
fclose(Arduino);
delete(Arduino);

```

Exponential Moving Average filter in TestRigArduino.ino

TestRigArduino.ino available at https://bitbucket.org/smartwinggdp/smartwing_gdp1617

```

1. //*****
2. /**
3.  * Flexible filtering >> updates current_value of load cell
4.  * a_pin: load cell class object
5.  * c_value: Current value
6.  * resp: responsiveness
7.  * iter_r: number of iterations
8.  * Uses median filtering for outlier rejection
9.  * - takes 5 new readings > sort > median
10. * *further work: creep recovery
11. */
12. void update_and_send(HX711& a_pin, double& c_value, double resp , int iter_r){
13.     int i;
14.     double a, rate, temp;
15.     double newval = 0;
16.     double otl_rej_arr[] = {0, 0, 0, 0, 0};
17.     newval = a_pin.read_average(iter_r);
18.     rate = calc_rate(newval, c_value, resp);
19.
20.     // check if rate > 1
21.     if (rate > 1){
22.         /* reset history*/
23.         a = 1;
24.     }

```

```

25.     else {
26.
27.         /* set routine for large changes
28.          to reject outliers */
29.         if (rate > 0.5){
30.             /*
31.               assign next five values to outlier
32.               rejection array
33.             */
34.             otl_rej_arr[0] = newval;
35.             for (i = 1; i < 5; i++){
36.                 otl_rej_arr[i] = a_pin.read_average(1); //could be iter_r also
37.             }
38.
39.             /* Sort array */
40.             arr_sort((double *) otl_rej_arr, 5);
41.             // recalculate rate
42.             rate = calc_rate(otl_rej_arr[2], c_value, resp);
43.         }
44.         a = rate;
45.     }
46.     temp = c_value; //not needed
47.     c_value = temp + ((newval - temp) * a);
48.     Serial.println(c_value);
49. }
50.
51. /* determine rate of change */
52. double calc_rate(double new_r, double curr_val, double resp){
53.     // diff = abs(new value - current value)
54.     double diff = abs(new_r - curr_val) * DIFFMULT;
55.     // y = ( 1 - 1/(1 + diff) ) * responsiveness
56.     return (1 - (1/(diff + 1))) * resp;
57. }
58.
59. void arr_sort(double* arr, int len){
60.     double temp;
61.     for (int i = 0; i < (int) (len/2) + 1; i++){
62.         for (int j = len - 1; j > i ; j--){
63.             if (arr[j - 1] > arr[j]){
64.                 //swap the lowest val down to the i_th position
65.                 temp = arr[j - 1];
66.                 arr[j - 1] = arr[j];
67.                 arr[j] = temp;
68.             }
69.         }
70.     }
71. }
72. //*****
73.

```

Servo Arduino script – ServoArduino.ino

Servo Arduino code available at https://bitbucket.org/smartwinggdp/smartwing_gdp1617

Data acquired during wind-tunnel tests:

The data obtained during the wind tunnel tests can be found at:

https://bitbucket.org/smartwinggdp/smartwing_gdp1617