POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

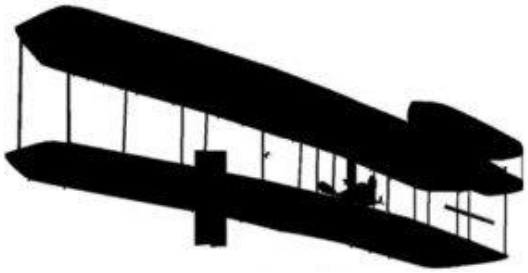
Corso di Laurea Magistrale in
Ingegneria Aeronautica

Fixed-wing UAV performance flight testing

Relatore:
Prof. Lorenzo Trainelli

Tesi di Laurea di:
Federico BUS, Matr. 799910

Anno Accademico 2014/2015



"If it wasn't for dreams,
man would still be on the ground."

- Unknown

"A chi sorride, guardando il cielo"

Table of contents

List of figures

List of tables

Sommario

Vista la recente crescita del settore dei droni e del loro utilizzo, il presente lavoro di tesi consiste nella realizzazione di un UAV sperimentale finalizzato all'esecuzione di prove di volo tipiche dei velivoli pilotati, con valenza sia didattica sia di ricerca, con potenziali applicazioni nell'ambito della qualificazione e certificazione di velivoli senza pilota. Nello specifico si è utilizzata come piattaforma volante un motoaliante elettrico radiocomandato prodotto dalla *Multiplex* controllato da un computer di bordo di derivazione *Arduino* distribuito dalla *3D Robotics* specifico per applicazioni aeromodellistiche. Alcuni sensori non presenti nella suite imbarcata originariamente, ed in particolare i rilevatori di angolo di incidenza e deriva, sono stati integrate nel sistema per completare il set di parametri necessari. Per avere un confronto con i dati sperimentali e un'idea preliminare delle forze in gioco, si è creato un semplice modello numerico del profilo dell'ala e del velivolo completo ricavando poi i dati di interesse attraverso un calcolo semplificato utilizzando i software XFoil e AVL. Varie prove sono state eseguite in galleria del vento per fornire una prima caratterizzazione sperimentale dell'aerodinamica dell'ala, per calibrare i nuovi sensori e per raccogliere quei dati come la trazione fornita dal sistema propulsivo che non è possibile determinare in volo. Pronto il sistema "hardware", si è passati allo sviluppo e validazione di alcuni modi di volo automatici, precisi e ripetibili, che riproducano delle manovre standard come quelle che un pilota addestrato eseguirebbe in una campagna di sperimentazione in volo su di un velivolo pilotato. La campagna di prove di volo si è interamente svolta al campo volo del club per aeromodelli GAT (Gruppo Aeromodellistico Trezzanese) sito in San Vito di Gaggiano (MI) per un totale di quattro giornate di voli. Una volta raccolti tutti i dati necessari dalle prove di volo stabilite si è passati alla fase di post-processing ossia all'attività di riduzione e filtraggio dei dati ed infine alla presentazione dei risultati che comprendono i grafici di CL-α, CD-α, polare parabolica, velocità di stallo in diverse configurazioni di volo, velocità di salita e di discesa ottime.

Parole chiave:

Sperimentazione in volo, prove automatiche, prestazioni aerodinamiche, UAV, ArduPilot Mega.

Abstract

Aware of the exponential growth of the drones industry and their applications in the recent years, this thesis work consists in the realization of an experimental UAV with the aim of carrying out flight tests typical of manned aircrafts, with both educational and research valence, with potential applications in the field of qualification and certification of unmanned aerial vehicles. More specifically we used as flying platform an electric powered glider model produced by *Multiplex* controlled by an on-board microcomputer derived from an *Arduino* board distributed by *3D Robotics* and described as an "open source autopilot" specific for model aircrafts. Some sensors not originally included in the system equipment, specifically angle of attack and angle of sideslip vanes, were integrated in the system to complete the set of necessary data for our tests. In order to compare the experimental results with an ideal model and to have a preliminary idea of the forces acting on the aircraft, two simplified numerical simulations were calculated for the airfoil and for the complete airplane using XFoil and AVL software. Various wind tunnel tests were performed to get a preliminary characterization of the wing aerodynamics, to calibrate the new sensors and to collect those data such as the propeller traction which are impossible to collect in-flight. Once the "hardware" system was ready, all energies were put into the development and validation of some automatic flight modes that would reproduce the standard maneuvers, precise and repeatable, that a trained pilot would perform in a flight test campaign for a manned aircraft. The whole flight test campaign was carried out at the aircraft models field of the GAT club located in San Vito di Gaggiano close to Milano (Italy) for a total of four full days of flights. Once all the necessary data were collected from the established repetitions of the flight tests, the post-processing phase started, namely the raw data reduction and filtering activity and the final results presentation comprehensive of the C_L-α, C_D-α and parabolic polar graphs, stall speed in different flight configurations, optimum glide and climb ratio.

Keywords:

Flight testing, automatic tests, aerodynamic performance, UAV, ArduPilot Mega.

1. Introduction

1.1 UAV evolution and practical utility

More and more today we hear about drones. The UAVs (Unmanned Aerial Vehicles) are, as often happens for new aerospace technologies and not only, a reality that firstly came up in the military field and then found civil applications. A huge evolution in this direction followed the miniaturization of electronics and the falling of its cost. From the *Predators* worth millions of dollars the American intelligence used to survey the sky over Iraq during the Gulf war or its border with Mexico, we are now talking about leisure small drones in fixed wing or multirotor configuration used by hobbyists worth few hundreds euros.

Figure 1.1 – The *Predator-B Reaper* used in surveying the American borders

A fixed-wing aircraft has both advantages and disadvantages in comparison with rotor-craft. Fixed-wing aircraft tend to be more forgiving in the air in the face of both piloting and technical errors, as they have natural gliding capabilities with no power. Fixed-wing aircraft also are able to carry greater payloads for longer distances on less power. There is a huge variety of fixed wing aircraft from electric battery powered small foam planes to large scale wooden replicas with multi liquid fuel engines and everything in between. As an UAV you are bound to find an aircraft that suits your needs. When precision missions are required, fixed-wing aircraft are at a disadvantage, as they must have air moving over their wings to generate lift. This means they must stay in forward motion, which means they can't hover in one spot the way a copter can and as a result cannot provide the same level of precise positioning. So for longer missions and more payload, a fixed-wing is the best choice but for keeping an instrument in one place, a copter is naturally more suited to do that.

Fixed wing or copter configuration, these machines are powerful and being already used by a variety of professionals like photographers and video makers to get amazing aerial shots.

Figure 1.2 – *DJI Inspire 1* drone equipped with gimbal stabilized HD camera

We also hear about drones spraying pesticides on the cultivated fields or sowing new plants, drones mapping natural disasters and searching for missing people, up to Amazon and its postmen drones, with the idea of flying around the cities delivering their packages. At this point one question spontaneously arises: to which extent this technology is safe and what are the laws and regulations governing its use? The speed with which we entered the "dawn of the aerial age", as stated in one quite scenic commercial spot by *3D Robotics*, and the ease with anyone nowadays is able to buy and learn to control their own personal drone has left a little confused even the regulations that should govern the use of these powerful, highly technological and controversial aircrafts.

Figure 1.3 – A helicopter configuration drone spraying pesticides

Figure 1.4 – The Amazon Prime Air delivery drone concept

The challenge for the future of commercial drones therefore is all in the possibility of reconcile the potentiality of the systems with the adoption of a legislation that regulates their use but doesn't make it complicated and/or expensive.

1.2 Thesis objectives

To be certified, all aircrafts must meet precise safety requirements and their flight characteristics should be clearly defined by standard parameters. These requirements should be met by UAV just as in the case of a new conventional aircraft, piloted by a real test pilot. The implementation of control logics that automatically govern the UAV during the flight tests is of primary importance because the pilot, in this case on the ground, through remote manual control would not be able to be precise, due to the lack of instant feedback of the behavior of the plane. In fact maneuvering from the ground, the aircraft is not always perfectly visible and its attitude is difficult to determine. On the other hand, an on-board autopilot, accessorized with all the required sensors would be able, in real-time, to govern the aircraft according to its implemented logics and specifically to execute the flight test maneuvers in a precise and repeatable way typical of a computer making it possible to verify the quality of the acquired data.

The present thesis work, carried out from the beginning of October 2014 until the end of June 2015, consists in the realization of a low-cost UAV to verify the effective potentiality of such a system along with the realization of everything that is needed to carry out a flight test campaign with the scope of characterizing some of the aircraft aerodynamic performance needed for an hypothetical certification.

The following list details the specific objectives:

- Flying platform set up – installing the autopilot and all its integrated and external sensors on the airplane.

- Numerical studies – using open source aerodynamic simulation software to model the aircraft and get a quick idea of what to expect from the experimental tests.

- Wind tunnel testing – to calibrate the air data sensors and to get a first ‘clean’ experimental result for the airplane wing aerodynamic coefficients.

- Software development – implement the new flight modes to use in the flight testing campaign and execute the code in Hardware In the Loop (HIL) simulation for validation.

- Flight testing – on field real flight performing of all the scheduled tests which are: wind determination, trimmed flights, stall, glide and climb maneuvers.

- Post processing – reducing and filtering all the collected data to obtain and present the results in the form of standard parameters and graphs.

2. The aircraft and its on-board systems

In this chapter the aircraft used in the thesis work, together with a description of its on board systems at hardware level, will be presented.

2.1 The aircraft

2.1.1 Technical specifications

The *Cularis* is an electric powered model glider produced by the German brand *Multiplex®*, famous for the introduction in the r/c world of the *Elapor*, a material very similar to propylene foam, very light but durable at the same time. Mainly made up of the above mentioned material, this model is a medium/high class glider with a 2.61 meters wing reinforced by carbon fiber spars and a 1.26 meters fuselage reinforced by PVC. The propeller is mounted in the nose in pulling configuration and the spinner automatically folds when the motor is off in order to reduce the drag during glides and ensure safe landings.

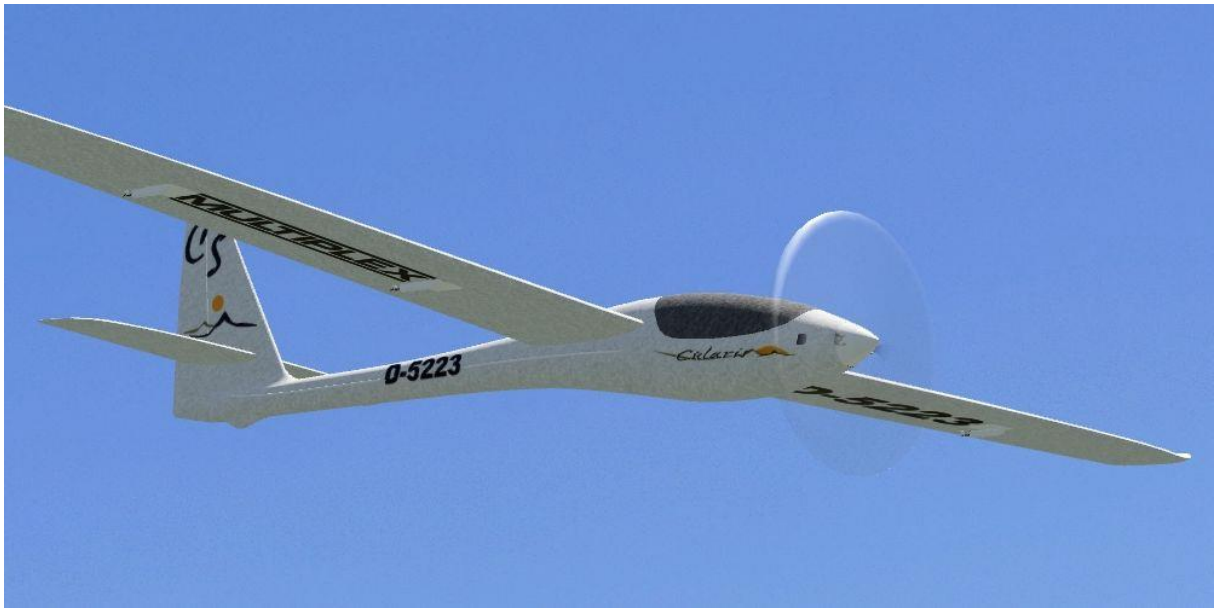


Figure 2.1 – 3D rendering of the flying platform

Mentioning the high aspect ratio and the slender body typical of a glider, this design is optimized to have a high efficiency necessary to ensure ascending flights in dynamics currents or thermals just as for the case of a real size piloted glider.

Following the main technical specifications as declared by *Multiplex* mounting the standard motorization kit *Multiplex Himax:*

Wingspan	2.61 m
Fuselage length	1.26 m
Wing area	0.436 m^2
Weight (in flight)	1.68 kg
Wing load	3.8532 kg/m^2
Aspect ratio	15.624

Table 2.1 – Flying platform technical specifications

2.1.2 Propulsive system

The propulsive system installed on the *Cularis* makes the aircraft an hybrid between a powered airplane and a pure glider with the advantages of both configuration. In fact the power available is way enough to take off without the need of another motorized airplane or a winch catapult system to get to the desired and on the other hand, the folding propeller doesn't add too much drag during unpowered glides.

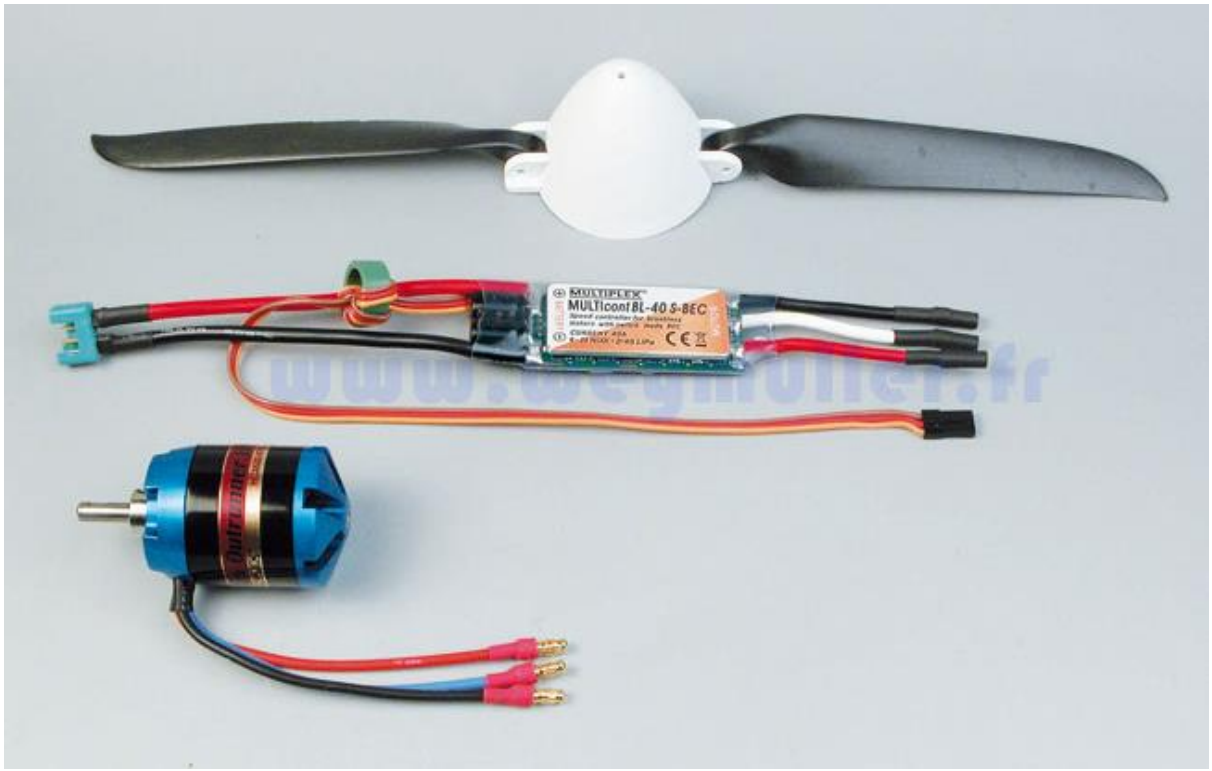


Figure 2.2 – The propulsive system of the flying platform

The propulsive system comprehends four components connected to each other:
propeller – electric motor – electronic regulator (ESC) – battery.

Propeller:

The propeller connected to the motor shaft is a twin blades fixed pitch, folding, of dimensions 12" x 6" with low warpage to ensure high performances especially at low speeds maneuvers as take off. We already mentioned the advantage of having a folding propeller in order to reduce drag during glides but this feature also becomes fundamental during landings because the model doesn't have landing gear and stops itself by friction simply contacting with the low grass of the landing field. If the propeller couldn't fold, an impact with the ground would be inevitable.

Electric motor:

The brushless motor installed on the *Cularis* can generate 400 W power and it is recommended for aerobatic models up to 1 kg and for glider models up to 4.5 kg. It has so plenty of power to rocket our model to altitude. Brushless motors have many advantages compared to the old with-brushes motors as the high ratio torque/weight and their high efficiency. Moreover these kind of electric motors are highly reliable, have a longer life cycle and do not need a dedicated cooling system. For all these reasons they became very popular in the world of airplane models.

Following the technical specifications of the *Himax C 3522-0700:*

Weight	162 g
Max power	400 W
Max angular velocity	20 000 rpm
Case diameter	35.2 mm
Length (shaft excluded)	48 mm
Shaft diameter	5 mm
Max temperature	65 °C
Rpm/Volt	700
Internal resistance	0.049 Ω
Current (optimal range)	8-29 Amp
Current (max for 15 s)	40 Amp

Table 2.2 – *Himax C 3522-0700* motor technical specifications

Electronic regulator:

The Electronic Speed Control (ESC) is an electronic circuit with the function of varying the angular speed of the motor, its sense of rotation, and in some cases can also function as a dynamic brake. This component is an independent unit, connected at one end to the battery and on the other end to the motor. A third cable comes out from the ESC which is the BEC end (Battery Eliminator Circuit) connected to the receiver that controls the throttle signal and supply power to the receiver and servos without the need of another battery.

The ESC in use on the *Cularis* is the *MULTIcont BL-40 S-BEC* produced by *Multiplex:*

Max continuous current	40 A
Working frequency	8-16 kHz (adjustable)
Voltage S-BEC	5 V
Current S-BEC	2.5 Amp
Dimensions	73 x 28 x 9 mm
Weight	43 g

Table 2.3 – Electronic Speed Controller (ESC) technical specifications

Battery:

As power source we use rechargeable Li-Poly batteries. The Li-Poly batteries are a technologic advancement of the Li-Ion batteries from which they differ for the electrolyte restrained in a solid polymer instead of in an organic solvent. The Li-Po batteries are the most used in r/c applications due to their lower weight and enhanced capacity, the most requested characteristics in this field that make them preferable over the other types of batteries despite of a higher cost.

Figure 2.3 – Li-Poly battery *Li-BATT eco 3/1-3000(M6)*

The batteries in use on the *Cularis* are two *Li-BATT eco 3/1-3000(M6)* also produced by *Multiplex.* The advantage of having two batteries at the field is that while one is being used on the airplane, the other can be recharged at the same time ensuring and almost non-stop cycle of flights. Technical specifications as follows:

Nominal tension	11.1 V
Number of elements	3 elements (3S/1P)
Capacity	3000 mAh
Continuous discharge current	Max 36 A
Dimensions	148 x 36 x 26 mm
Weight	260 g

Table 2.4 - Li-BATT eco 3/1-3000(M6) technical specifications

2.1.3 Servos

To move the control surfaces, on the *Cularis* are installed six analogic servos: four under the wings, one for each aileron (2x) and one for each flap (2x), two inside the canopy, one for the elevator and one for the rudder. Each servo, through a small integrated electronic circuit, recognize the input coming from the receiver and realize a closed loop control on the angular position output of its shaft properly powering the small electric motor to which it is connected through a reducer.

Figure 2.4 – mini analogic servo *MPX Tiny S*

The type actually used are the *Multiplex Tiny S*, a robust, reasonably priced servo which is very fast, powerful, and offers an extremely good power to weight ratio. A complete list of the servo specifications can be found in the table below:

Torque	2.6 kg/cm (4.8 V), 3 kg/cm (6.0 V)
Speed	0.11 sec/60° (4.8 V), 0.09 sec/60° (6.0 V)
Weight	16.4 g
Dimensions	29.7 x 29.5 x 11.9 mm
Motor type	3-pole
Gear type	plastic

Table 2.5 – *MPX Tiny S* technical specifications

2.1.4 Tx and Rx

The radio control system in use is composed of the digital transmitter *Futaba® 8FG Super* and of the receiver *R6208 SB* of the same brand. It is a last generation system, with the transmitter capable of managing up to 14 radio channels in the 2.4 GHz band width using the FASST technology (*Futaba Advanced Spectrum Technology)* which ensure high resistance to external interferences and permits a safe use of many of these systems also at short distances. The transmitter, originally in MODE 2, has been modified to MODE 1 to fit my habit in flying this way (throttle and ailerons on right stick, elevator and rudder on the left stick).

Figure 2.5 – *Futaba* radio transmitter and receiver

The receiver can manage up to 8 channels (e.g. ailerons, elevator, motor, rudder, flaps…) and normally handles the input signal from the transmitter and generates the correct output to the servos through PWM (Pulse Width Modulation) signals. The "width", actually the duration of the impulse, is proportional to the position of the control sticks of the transmitter maneuvered by the pilot.

2.2 The autopilot

The ArduPilot Mega system, generally identified with the acronym APM is, according to its developers definition[5], an open-source autopilot, Arduino-compatible, of professional quality and the most advanced (and one of the cheapest!) open-source autopilot IMU-based. In the intentions of its creators, it has been conceived to transform a standard r/c model in a real UAV. It can interactively be configured through a ground control station (GCS) and it's capable of flying by itself in accordance with some automatic modes. Both fixed wing and multi-copter configurations are supported by simply uploading a different firmware.

At hardware level, the APM system is based on the Arduino Mega platform, more powerful and sophisticated variant of a standard Arduino. Arduino is basically an open-source microcontroller designed to build a low cost, rapid prototyping electronic platform that can be used with relative ease also by hobbyists in a variety of contexts.

2.2.1 History

Initially developed at the Interaction Design Institute (Ivrea, Italy), Arduino integrates on a single board an *Atmel® AVR* CPU, the input/output ports, the RAM memory to save the software instructions and the serial interfaces, USB and more.

The Arduino version specifically dedicated to UAV applications is called Ardu-Pilot and through the years underwent an evolution that brought it to its actual state of the art with the last, in use version, APM 2.6. The main features and changes between the versions are listed here:

- ArduPilot – The very first and most simple variant for UAV strictly derived by the standard Arduino (attitude control and stabilization, fly-by-wire).

- APM 1 – ArduPilot-Mega first fully autonomous UAV autopilot system based on the more powerful micro-processor Atmel's ATMEGA2560. It was made by two main hardware component: MAIN board and IMU (Inertial Measurement Unit) shield.

- APM 2 – First cased ArduPilot. This "out of the box ready to use" version came with no assembling and soldering required and a global revisiting of the whole board.

- APM 2.5 – No more shield/daughterboard: magnetometer and dataflash were moved to the main board, making it easier to assemble, repair and hack. Standard connectors to protect the board from those reverse polarity/short circuit lovers.

- APM 2.6 – Same as 2.5 but this revision of the board has no onboard compass, which is designed for vehicles where the compass should be placed as far from power and motor sources as possible to avoid magnetic interference. Side entry pins.

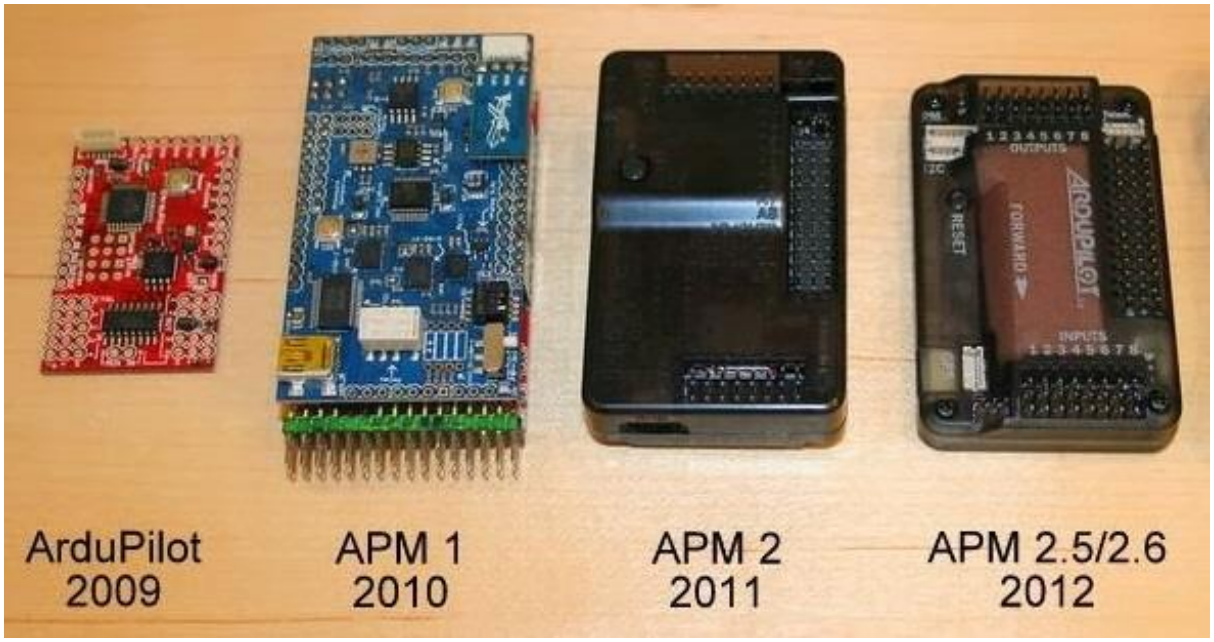


Figure 2.6 – *ArduPilot* and *ArduPilot-Mega* evolution

For more information about Arduino you can refer to the official website of the project [4].

2.2.2 The APM 2.6

The ArduPilot Mega 2.6 used as control system in this thesis has been purchased directly from the producer *3D Robotics* company located in San Diego, California. The main reason of the investment was to replace the out of date APM 1 used for the previous thesis work on the same topic with a more advanced and reliable unit.

The APM 2.6 is a complete open source autopilot system and the bestselling technology that won the prestigious *2012 Outback Challenge UAV competition*[6]. As briefly stated before, this version is ready to use, with no assembly required. It allows the user to turn any fixed, rotary wing or multirotor vehicle (even cars and boats) into a fully autonomous vehicle capable of performing programmed GPS missions with waypoints. This revision of the board has no onboard compass, which is designed for vehicles (especially multicopters and rovers) where the compass should be placed as far from power and motor sources as possible to avoid

magnetic interference (on fixed wing aircraft it's often easier to mount APM far enough away from the motors and ESCs to avoid magnetic interference, so this is not as critical, but APM 2.6 gives more flexibility in that positioning and is a good choice for them, too). This is designed to be used with the 3DR uBlox GPS with Compass, so that the GPS/Compass unit can be mounted further from noise sources than the APM itself.

Figure 2.7 – The APM 2.6

Features:

- Arduino Compatible
- Dimensions 7.5 x 4.5 cm
- Up to 8 in / 8 out radio channels
- Comes pre-soldered (with angle pins) and tested
- Power supply 5.3 V, 2.5 mA (max) from dedicated system
- Serial ports I/O for mini-USB, GPS, dataflash and telemetry
- Includes 3-axis gyro, accelerometer along with a high-performance barometer
- External Compass (Optional)
- Onboard 16 MB dataflash memory for automatic datalogging
- 16MHz Atmel's ATMEGA2560 and ATMEGA32U-2 chips for processing and usb functions respectively.
- Set includes Telemetry cable, Micro USB cable, DF13 6 Position connector for the Power Module and GPS connector cable.

Note: The Power Module is only designed to power APM, a RC receiver and APM accessories (GPS, Telemetry radio). It is not designed to power servos. The aircraft's own ESC/BEC is suited for that.

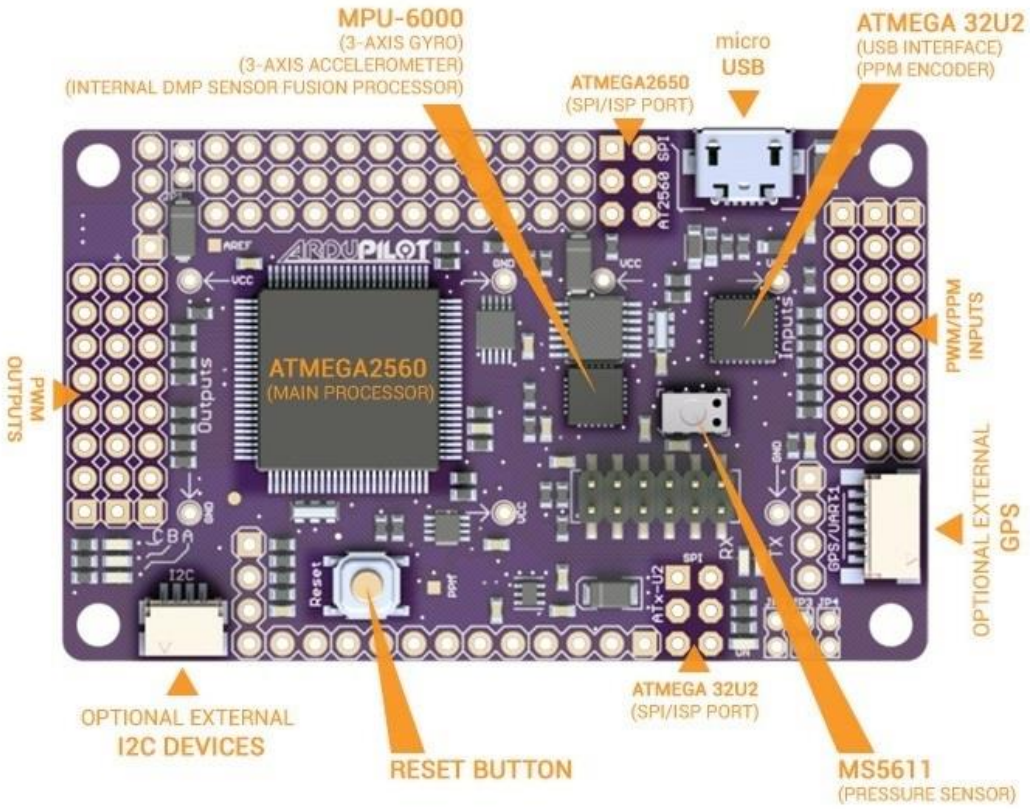


Figure 2.8 – APM 2.6 main components

2.2.3 Integrated sensors

As briefly stated before in the features section, the APM 2.6 board already integrates all the inertial measurements units and necessary sensors upgraded to the latest MEMS (Micro Electro-Mechanical Systems) technology. These sensors are incredibly small and light in weight but still very accurate and extremely reliable:

- Invensense's 6 DoF Accelerometer / Gyro MPU-6000.
- Barometric pressure sensor MS5611-01BA03, from Measurement Specialties.

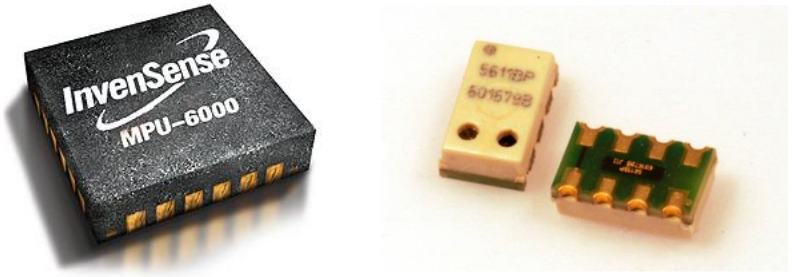


Figure 2.9 – 6 DoF Accelerometer/Gyro MPU-6000 and barometric pressure sensor MS5611-01BA03

2.2.4 External sensors

Some external sensors are needed to allow APM 2.6 autopilot to fully autonomously operate (GPS and magnetometer), others are optional but very recommended for the safety of the board and the quality of its control (voltage sensor and air data system) and the last ones are newly integrated, non-standards sensors but necessary to complete the flight testing data set.

- External GPS, uBloxNEO6 with integrated compass Honeywell's HMC5883L-TR

Figure 2.10 – External GPS and compass unit

- Integrated voltage sensor in power module

  The APM 2.6 has a dedicated connector for attaching the 3DR Power Module (PM) which is useful because it provides a stable 5.3 V and 2.25 Amp power supply to the APM which reduces the chances of a brown-out and allows monitoring of the battery's voltage and current making it possible to trigger a return-to-launch when the voltage becomes low or the total power consumed during the flight approaches the battery's capacity.

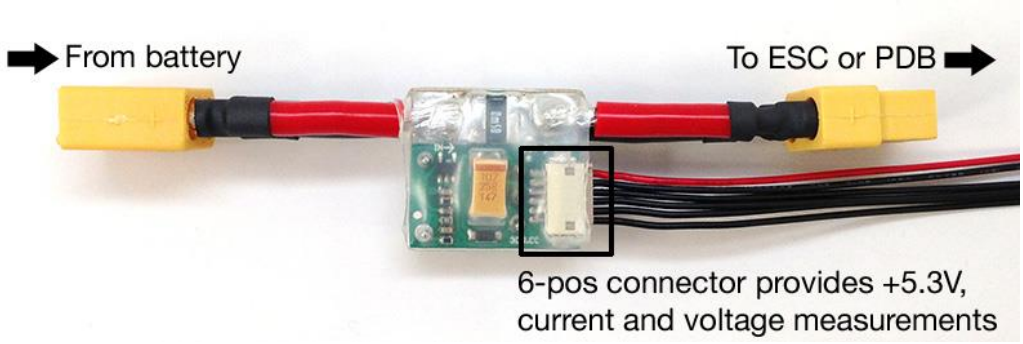


Figure 2.11 – PM with integrated voltage sensor

- Air speed sensor

APM 2.6 supports the use of an airspeed sensor, which can help in windy conditions, slow flight and autonomous landings. It is a more accurate way of determining the speed of the airplane in the moving air, of course, than using the information coming from the GPS and can be also very useful in preventing stalls. It is made of two main parts: the pitot tube with both total and static pressure ports and the piezoelectric pressure sensor element that in this case is the Freescale Semiconductor® MPXV7002.

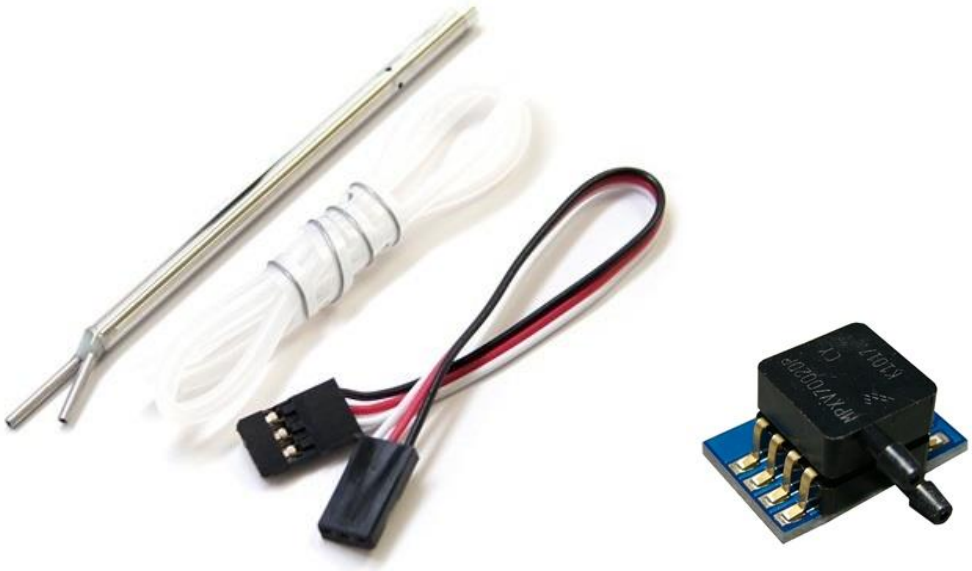


Figure 2.12 – Airspeed sensor kit

In our case the adoption of an airspeed sensor is very important because a lot of the results from the flight tests are based on the airspeed information.

- AOA and AOS vanes

Angle of attack and angle of sideslip sensors are not included in the optional hardware listed in the *3D Robotics* "official parts" page[7]. However, knowing the exact value of these two parameters in all the stages of the flight is really important in a flight test campaign.

For this reason, two small vanes, originally intended for military applications, have been integrated in the system at both hardware and software level to satisfy this requirement. The sensors are nothing more than two balanced aeronautical aluminum vanes connected to ratio-metric potentiometers. Connected directly as logical signals to the expansion ports of the APM 2.6 they get a 5 V input and respond with a variable signal proportional to the angular position of the vanes' shafts.

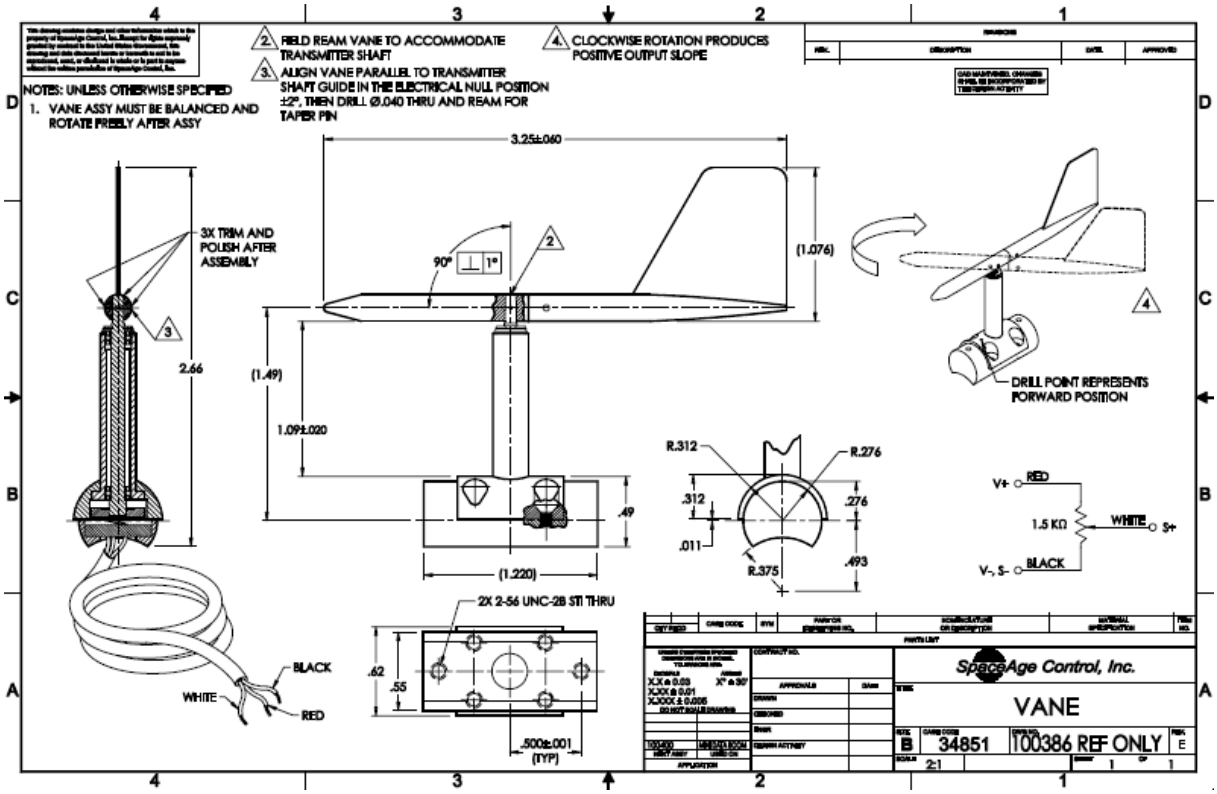


Figure 2.13 – Schematic drawing of the AOA and AOS vane sensors

Note: the sensors already came with their experimental calibrations from the producer *Space Age Control, Inc.* but needed to be recalibrated for the specific use and the particular position in which they have been installed (see section 4.3.3 and 4.3.4).

2.2.5 Telemetry

In order to obtain an inflight control and real-time monitoring of the parameters affecting the flight and all the information about it, a two way telemetry is needed. 3DR radios provide an air-to-ground data link between the autopilot and the ground station laptop or tablet.

Figure 2.14 – 3DR telemetry radios

Technical specifications

433 MHz transmission frequency (allowed in Europe)
100 mW maximum output power (adjustable)
-117 dBm receive sensitivity
2-way full-duplex communication through adaptive TDM
MAVLink protocol framing
Frequency Hopping Spread Spectrum (FHSS)
Configurable duty cycle
Error correction corrects up to 25% of bit errors
Interchangeable air and ground modules
Power Supply voltage: 3.7-6 VDC (from USB or DF13)
Transmit current: 100 mA at 30 dBm
Receive current: 25 mA
Serial interface: 3.3 V UART
Dimensions 26.7 cm x 55.5 cm x 13.3 cm (without antenna)

2.3 Final assembly

2.3.1 The “brain”

The autopilot can be thought as the brain of the UAV. It gets the info coming from the sensors just like animals and humans do through the senses, elaborates them and produces an output resulting in motion of the control surfaces to get the desired attitude. As a brain, the autopilot fits into the “head” of the UAV, which is the dedicated space in the front of the glider underneath the canopy. The *Cularis* actually is not primarily intended as an UAV platform so the space is not quite appropriate to fit a lot of electronic devices. In addition some electronics have to be in a particular position to avoid interferences with the motor magnetic field or to receive some signals from the outside (e.g. GPS or telemetry antenna). So we had to be extremely precise to fit everything inside the canopy and in the right place. The result, as shown below is a pretty “tight” but functional displacement.

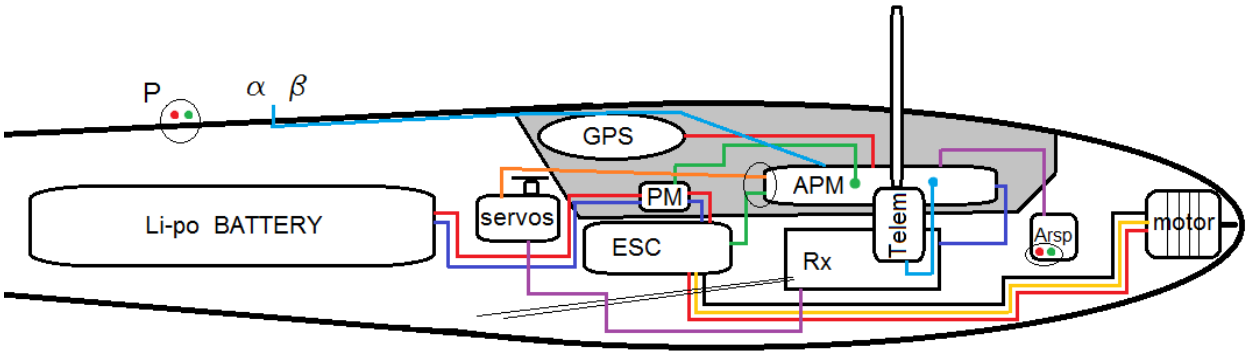


Figure 2.15 – Schematic layout of electronic equipment

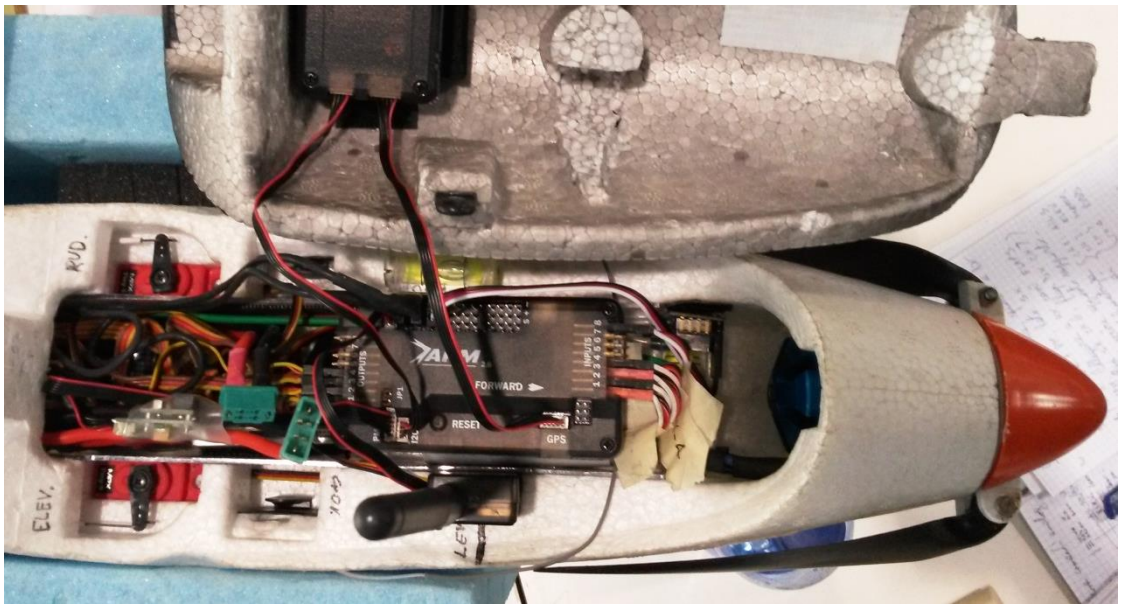


Figure 2.16 – The “brain” inside the canopy

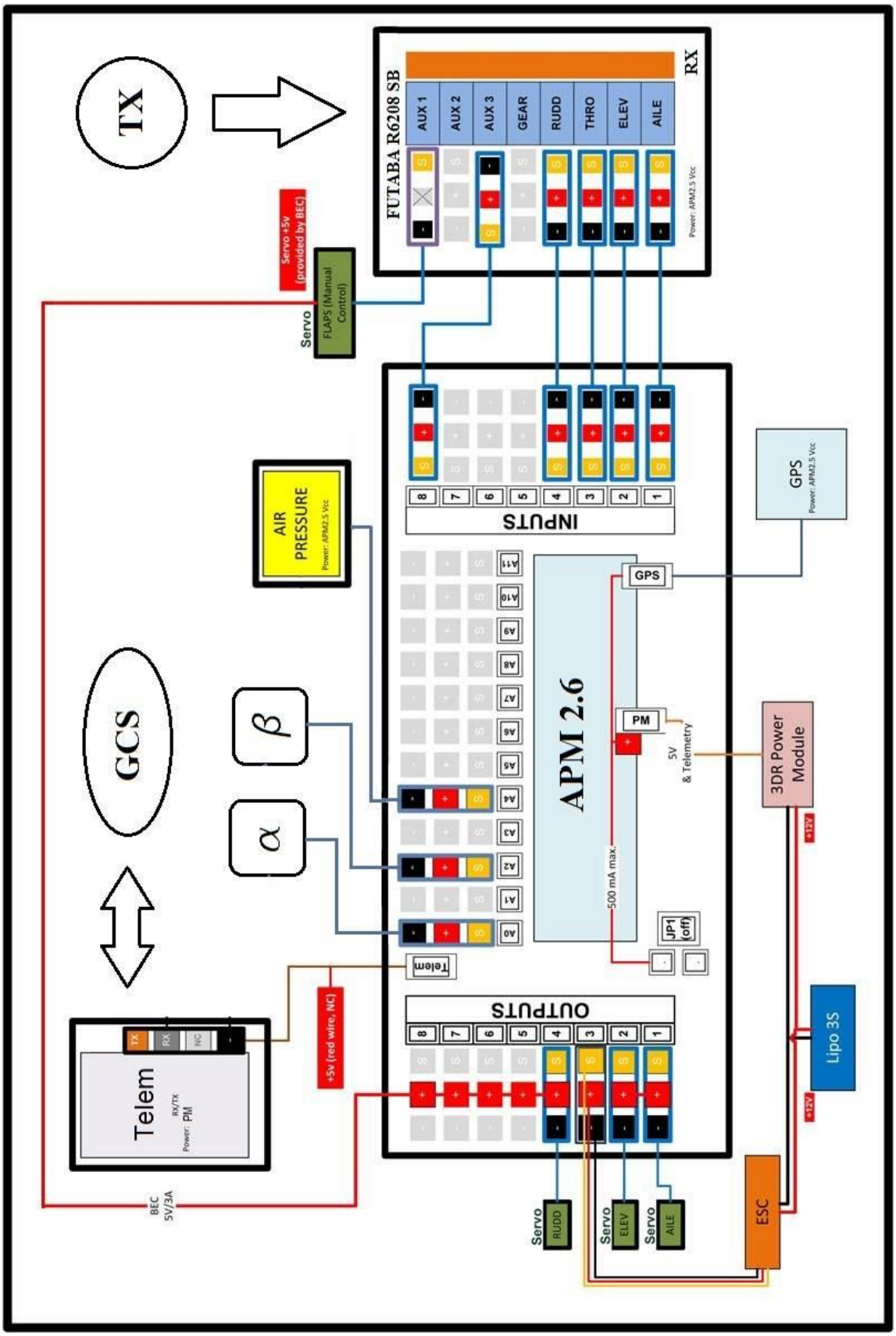


Figure 2.17 – APM and sensors electrical connections scheme

As it can be seen from the schemes above, the APM has two power inlets. One comes from the dedicated power module that gives a stable 5 V to all the sensors, the receiver and telemetry, the other is the ESC's BEC itself that only powers the rear rail of the APM to where the servos are connected. The rear rail needs more current to power the servos and it is disconnected from the rest of the electronic equipment to prevent short circuits going through the primary electronics causing burns and brown outs. Only the signal wire (of course) it is allowed to reach the rear rail from the main board. Normally all the channels of the receiver are connected to the APM. One of them is dedicated to the changing of the modes that can be controlled directly by the pilot through a switch on the transmitter. In auto mode the controller takes over completely on the servos; in semi-auto mode there's some stabilization or fly-by-wire features added to the manual control but the pilot has still some authority, in manual mode all the channels are a complete "pass-thru" and go directly from the receiver to the servos through the APM not being affected by the autopilot. About the modes we'll talk more in chapter number 5: "The control system".

2.3.2 Sub-wings supports

The air data acquisition system, composed by airspeed sensor with pitot tube and AOA and AOS vanes, had to be installed outside of the canopy in a relative undisturbed air flow area. To work properly giving plausible readings of the properties being measured, pitot tube inlets and vanes need to be placed where the air flow is as steady as possible, without major sudden variations or turbulences induced by the airplane itself. Usually a very common point where to install these devices on commercial and military jet aircrafts is in the nose. In the very front of the airplane in fact, the flow coming towards the airplane is still almost undisturbed with a velocity equal and opposite to the velocity of the airplane. In our case this position in clearly not exploitable since the *Cularis* in the front has a big propeller in pulling configuration and we should also be careful to avoid its large wake in the positioning of the sensors.

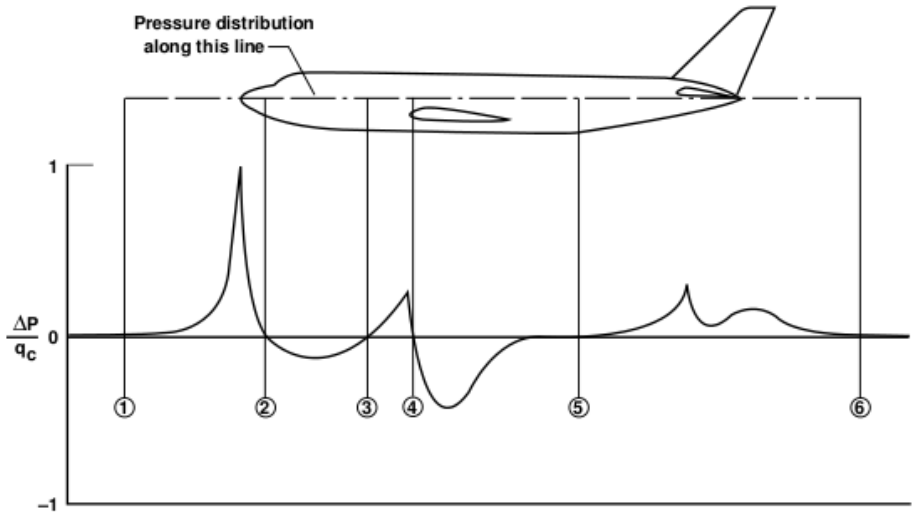


Figure 2.18 – Variation of static pressure along longitudinal axis compared to the asymptotic pressure

Different solutions have been taken into account, comprehending a long boom standing out vertically just behind the canopy among others safe-but-aerodynamic-destroyer configurations but in the end we came down to the most simple and clean solution that usually is also the best one.

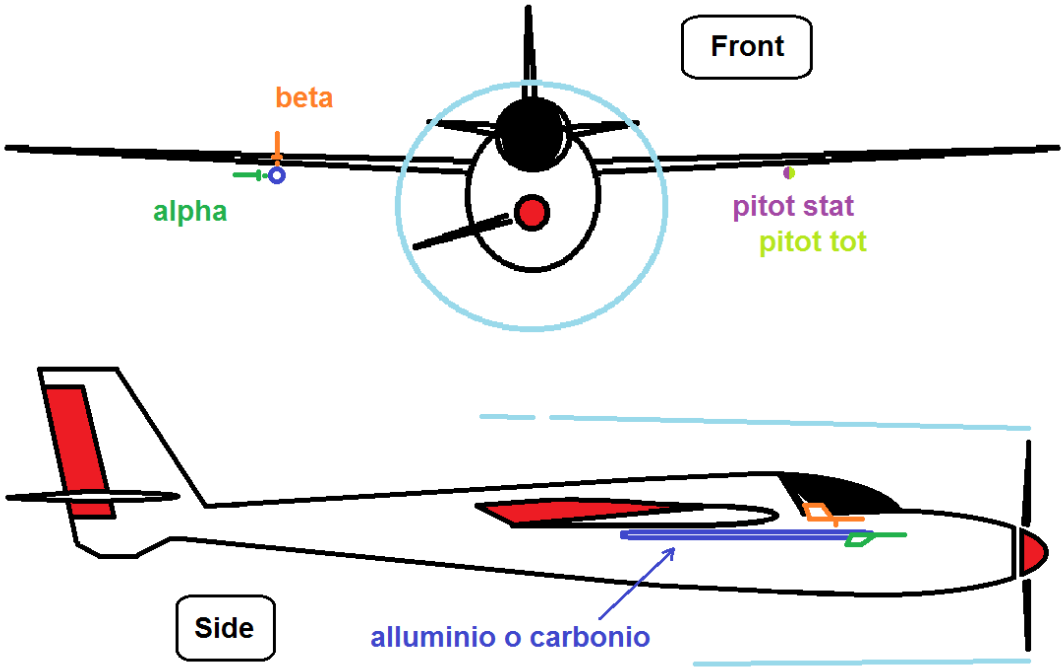


Figure 2.19 – Air data system logistic configuration

The figure above (non scaled) shows the twin sub-wings booms configuration that is the most aerodynamic and less-weight solution compared to the others and the saving in added drag also reduces the chance of induced vibrations. The booms are attached underneath the wings at about 30 cm from the fuselage, which is way out from the propeller wake but still close enough to consider negligible the change in angle coming from the flexing of the wing. On the other hand the sensors are fairly close to the terrain especially during landings and the pilot will have to be careful to land gently and flat on a short grass field in order to avoid damages.

The left boom carries the pitot tubes that integrates both total and static pressure ports, the right boom mounts the AOA and AOS vanes at 90° inclination from each other and respectively in horizontal and vertical position. One thing we had to be careful about was the alignment of the booms that had to be parallel to the longitudinal axis of the airplane rather than perpendicular to the leading edge of the wing which is tapered. Another observation is that for the alignment of the vanes we referred for the horizontal and vertical directions to the ground and not to the underneath surface of the wing which is affected by the pitch angle of the wing in relation to the longitudinal axis of the fuselage (3°) and by the dihedral angle of 1.5° which is small but still visible.

The booms are made out of a 8 mm external diameter carbon fiber tube, glued with epoxy resin to hand made balsa wood supports secured underneath the wing via double-sided strong tape. The pitot tube support had to be ballasted with a steel rod to balance the AOA and AOS sensors installation on the other wing and the tube and has been secured to the steel rod using insulating tape and small cables ties. The vanes were secured on two others aluminum perforated cylinders using very small bolts. This was intended to not perforate the carbon fiber tube which is quite fragile and we preferred to glue the aluminum cylinders to it instead. All the cables, which are the wires coming from the vanes and the two pressure tubes coming from the pitot, have been inserted inside the existing space dedicated to the cables of ailerons and flaps servos and pulled out all the way to the canopy to be connected directly to the APM and to the airspeed sensor respectively. In this way the cables, running inside the wing rather than outside, let the wing surface be as clear as possible avoiding drag increments.

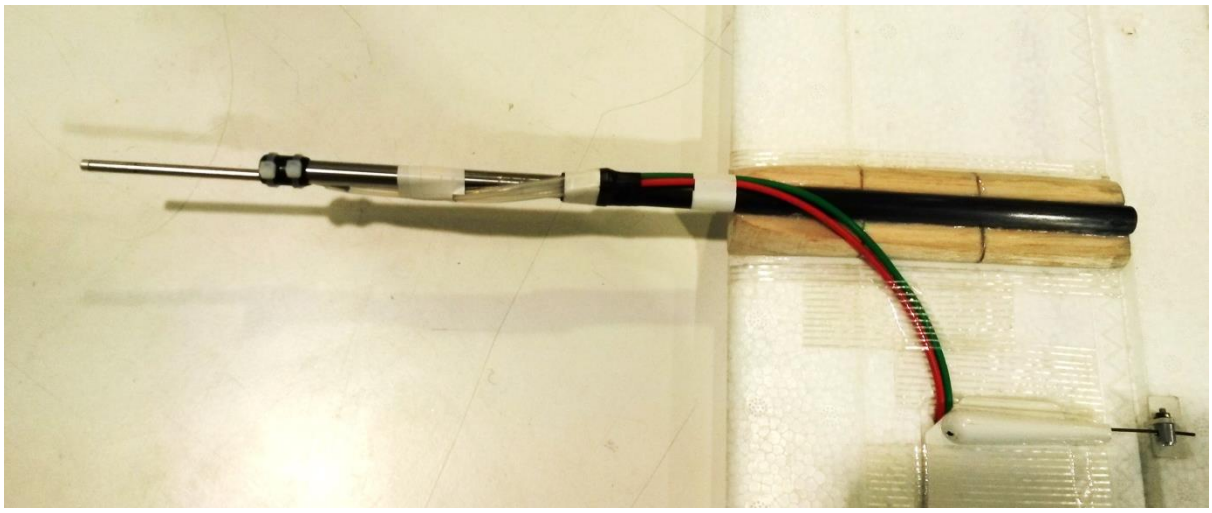


Figure 2.20 – Pitot tube sub-wing final assembly

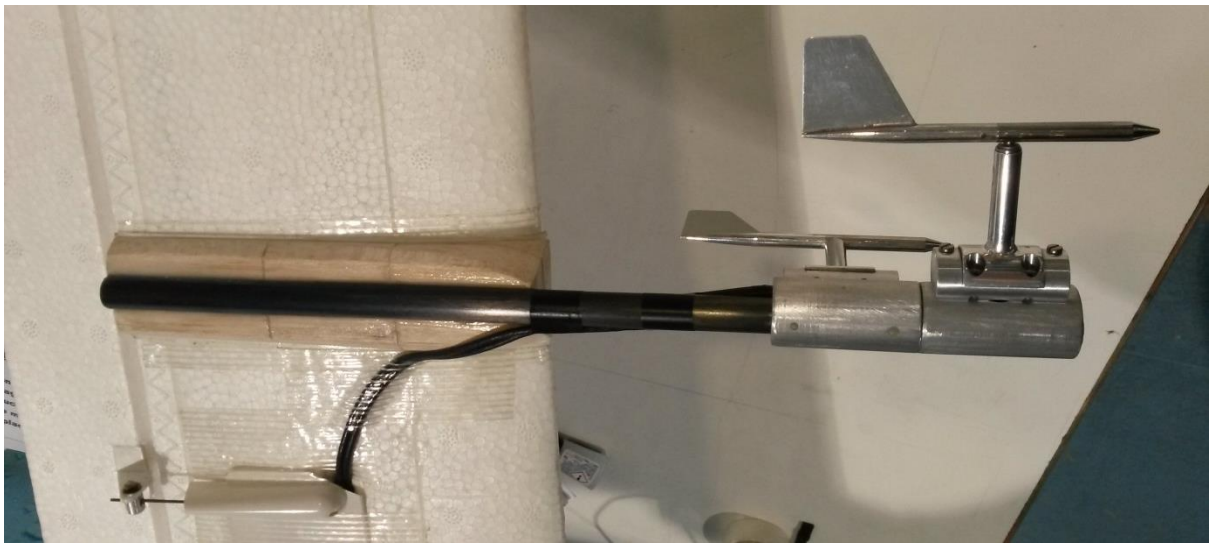


Figure 2.21 – AOA and AOS sensors final assembly

2.3.3 Center of gravity determination

Knowing the correct position of the center of gravity for an airplane is of vital importance because it affects the aircraft maneuverability and aerodynamic performances.

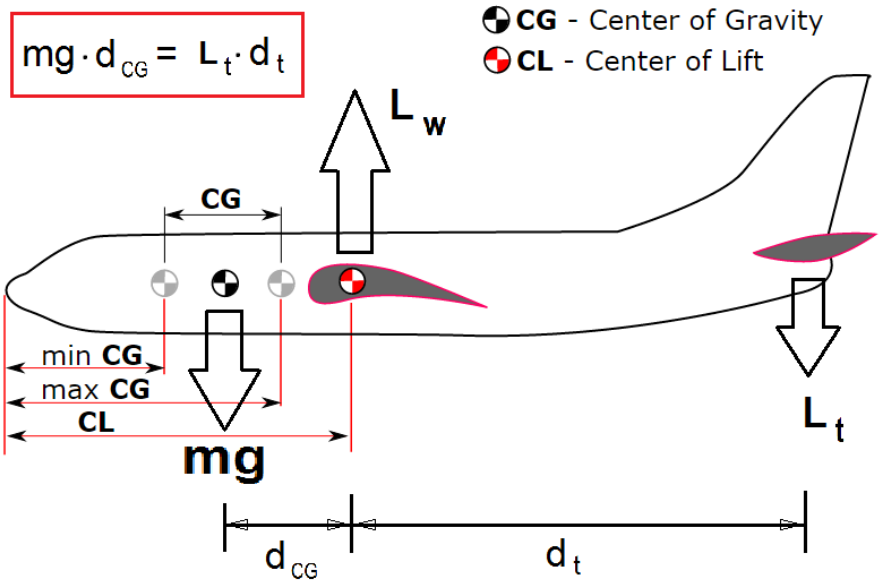


Figure 2.22 – Schematic of CG positon for longitudinal static stability for a conventional configuration airplane

Before installing all the components needed to turn the model glider into a fully autonomous UAV, a very first flight without any electronic devices other than the necessary ones for only manual control has been performed. This flight, as well as verifying the good qualities of the chosen flying platform, has been very useful to determine the correct position of the center of gravity of the plane. The right position, after some attempts and corrections, resulted to be at 62 mm from the wing leading edge close to the fuselage. This position is actually about a couple of millimeters in front of the position recommended by the producer paying a little decrement in efficiency of the overall airplane but resulting into an enhanced longitudinal stability.

Figure 2.23 – The *Cularis* during its first flight as a simple R/C model

After we installed the APM on the flying platform together with all its sensors and accessories, the center of gravity was way off its original position. This was due because of the weight that have been added in the front of the airplane, especially in the canopy and on the tips of the two booms underneath the wings. The AOA and AOS sensors together with their support, for example, weight about 90 grams and the pitot boom had to be ballasted with 70 grams more than its original 20 to balance out the vanes.

In order to restore the CG original position, the very end of the tail (were the arm of the weight force is greater) had to be ballasted with 10 and 5 grams small weights until the equilibrium was reached. The equilibrium means that the airplane stays still without pitching when only supported by two points underneath the wings and close to the fuselage at exactly 62 mm from the leading ledge.

Figure 2.24 – Final assembly of the UAV with all electronics and sensors installed

The final weight of the plane, measured with everything installed and ballasted in UAV configuration, reached 2.10 kg that is 420 grams more than its original weight of 1.68 kg also resulting in an increased wing loading of 4.8165 kg/m^2 compared with the original value of 3.8532 kg/m^2.

3. Numerical studies

As stated in the introduction, a quick numerical analysis using open source aerodynamic simulation software has been performed for completeness following the standard evolution and validation phases of a new design just as it happens for real aircrafts projects. From the results we didn't expect to collect very accurate data but at least to get a quick idea of what to expect from the experimental tests and to have a useful term of comparison.

3.1 Airfoil aerodynamics

XFOIL is an interactive program written in FORTRAN language for the design and analysis of subsonic flows around airfoils. The analysis can be run for both inviscid or viscous flows. Given the coordinates specifying the shape of a 2D airfoil, Reynold and/or Mach numbers, XFOIL can calculate the pressure distribution on the airfoil and hence the lift and drag characteristic parameters. The program was first developed by professor Mark Drela at the Massachusetts Institute of Technology in Boston (USA) as a design tool for the MIT Daedalus project in the 1980s and it was further developed in collaboration with Harold Youngren. The current version is 6.99, released in December 2013 and despite its vintage, is still widely used due to its simplicity and relative accuracy. For further reading and more info visit the official site reported in the bibliography[2].

Multiplex doesn't provide its models airfoils specification so the type of airfoil of the *Cularis* had to be find out experimentally. This has been possible reproducing the outline of the root airfoil of the wing on a piece of paper which it was then scanned and its drawing discretized with a *Matlab* function through a series of points defined by (x,y) coordinates. After we had a plot of the reconstructed profile, we have been able to measure some of its characteristic as thickness and chord measures and compare it with other similar knows profiles.

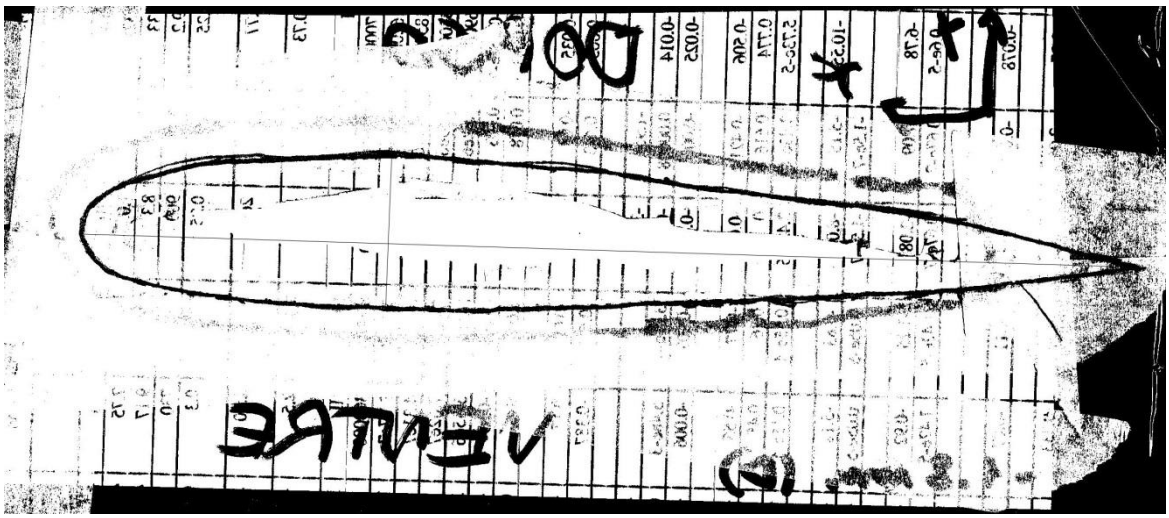


Figure 3.1 – *Cularis* airfoil outline on paper

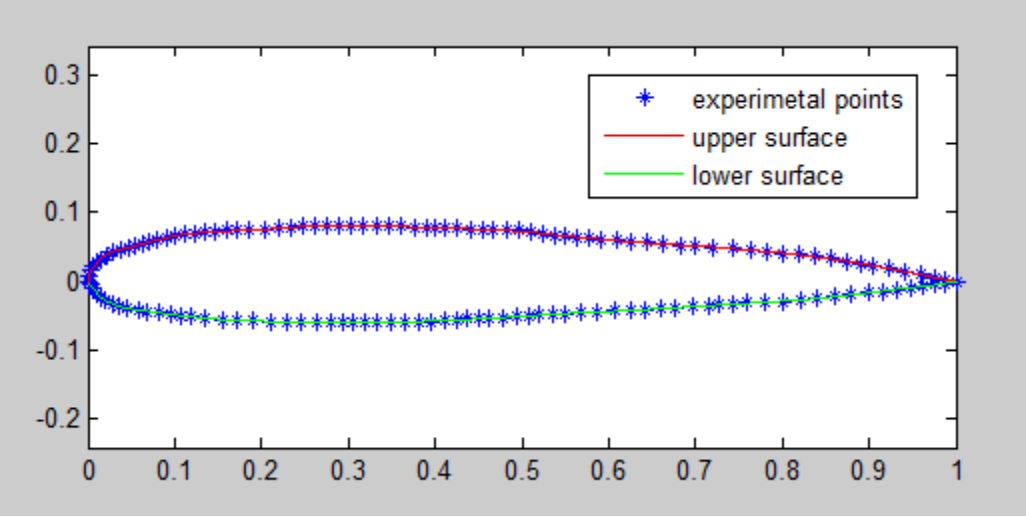


Figure 3.2 – Airfoil experimental points

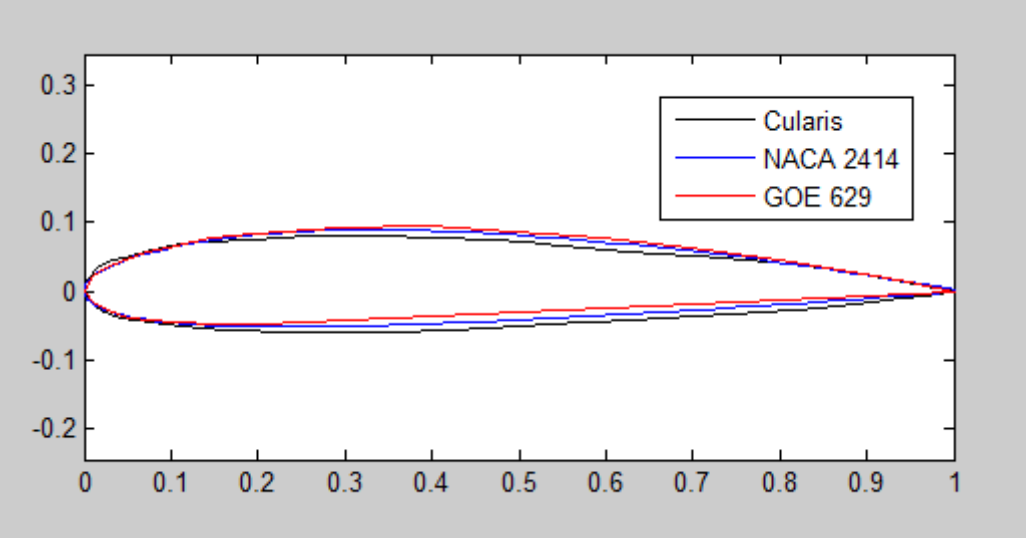


Figure 3.3 – Airfoils comparison

As it can be seen from the graphs above, the type of airfoil in use for the *Cularis* wing is very close to the four digits well known NACA 2414 airfoil. As an assumption we then will use for all our simulation this type of airfoil that is already in the software database.

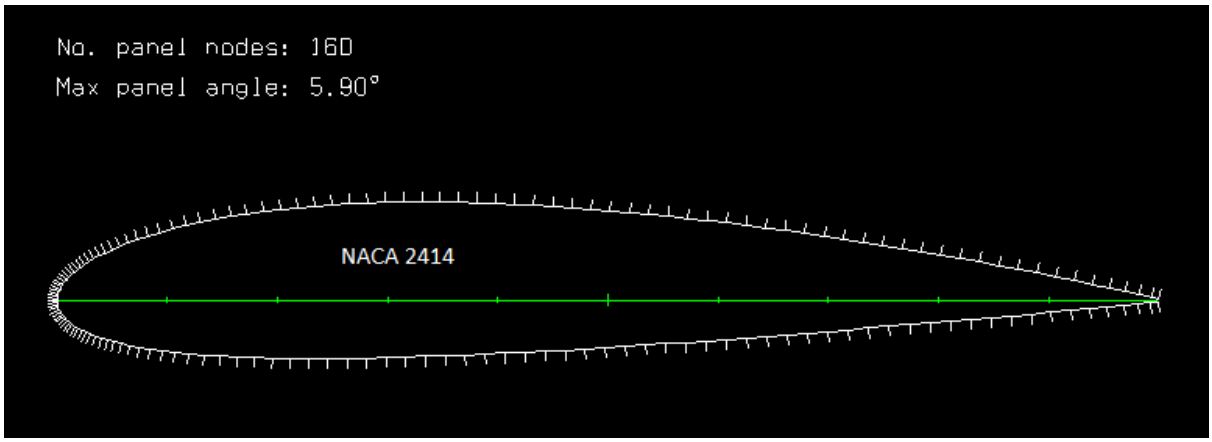


Figure 3.4 – NACA 2414 discretization in XFOIL graphic window

After we gave the software the airfoil geometry that got discretized by default into 160 panels (actually the resolution method is a panels method), we defined the correct estimation of the Reynold number specific for our application:

Air density (ρ) = 1.225 kg/m^3

Air viscosity (μ) = 18.27 x 10^{-6} Pa s

Typical airspeed (V) = 20 m/s

Airfoil chord length (L) = 0.2 m

$$Re = \frac{\rho V L}{\mu} \cong 250{,}000$$

At this point all the boundary conditions are set and we started the simulation for different angle of attack values varying from -13 degrees up to 15 degrees by increments of 1 degree. Following are the results:

XFOIL Version 6.96

Calculated polar for: NACA 2414

xtrf = 1.000 (top) 1.000 (bottom)
Mach = 0.000 Re = 0.250 e 6 Ncrit = 9.000

alpha	CL	CD	CDp	CM	Top_Xtr	Bot_Xtr
-13.000	-0.9246	0.04664	0.04163	-0.0746	1.0000	0.0418
-12.000	-0.9669	0.03798	0.03215	-0.0630	1.0000	0.0438
-11.000	-0.9520	0.03187	0.02547	-0.0515	1.0000	0.0468
-10.000	-0.9143	0.02776	0.02069	-0.0410	1.0000	0.0512
-9.000	-0.8563	0.02432	0.01706	-0.0332	1.0000	0.0570
-8.000	-0.7897	0.02166	0.01417	-0.0266	1.0000	0.0651
-6.000	-0.4947	0.01659	0.00911	-0.0441	0.9705	0.1089
-5.000	-0.3488	0.01471	0.00744	-0.0517	0.9510	0.1596
-4.000	-0.2071	0.01302	0.00610	-0.0579	0.9297	0.2474
-3.000	-0.0858	0.01152	0.00506	-0.0592	0.8995	0.3672
-2.000	0.0204	0.01014	0.00432	-0.0569	0.8588	0.5493
-1.000	0.1212	0.00933	0.00404	-0.0526	0.8083	0.7393
0.000	0.2240	0.00924	0.00407	-0.0480	0.7518	0.8805
1.000	0.3582	0.00978	0.00431	-0.0504	0.6906	0.9584
2.000	0.5364	0.01032	0.00444	-0.0639	0.6268	0.9946
3.000	0.6444	0.01069	0.00450	-0.0639	0.5718	1.0000
4.000	0.7324	0.01126	0.00486	-0.0594	0.5207	1.0000
5.000	0.8197	0.01197	0.00541	-0.0546	0.4684	1.0000
6.000	0.9056	0.01280	0.00614	-0.0495	0.4108	1.0000
7.000	0.9896	0.01388	0.00708	-0.0443	0.3427	1.0000
8.000	1.0673	0.01555	0.00849	-0.0385	0.2621	1.0000
9.000	1.1335	0.01799	0.01060	-0.0316	0.1799	1.0000
10.000	1.1809	0.02116	0.01354	-0.0223	0.1237	1.0000
11.000	1.2171	0.02545	0.01773	-0.0135	0.0933	1.0000
12.000	1.2530	0.03048	0.02292	-0.0069	0.0755	1.0000
13.000	1.2770	0.03725	0.02975	-0.0016	0.0642	1.0000
14.000	1.3028	0.04456	0.03730	0.0019	0.0562	1.0000
15.000	1.3241	0.05281	0.04584	0.0039	0.0503	1.0000

Table 3.1
Xfoil results

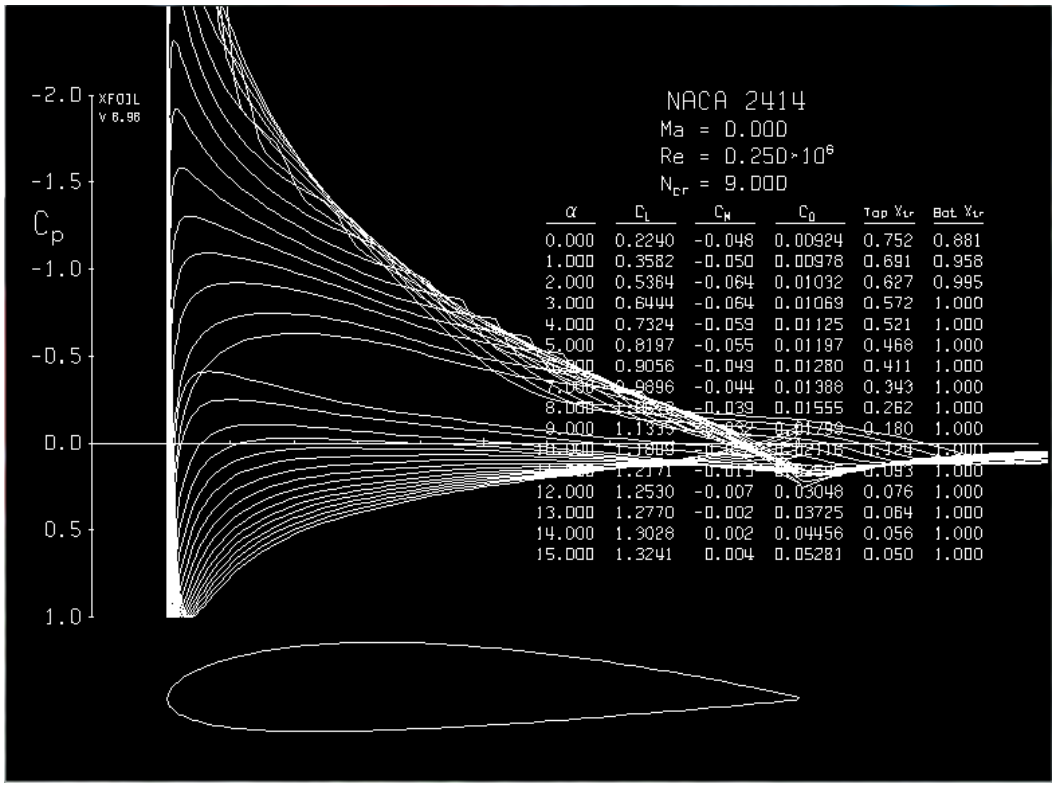


Figure 3.5 – Pressure coefficient plot as in XFOIL graphic window

Saving the results for a sweep in angle of attack into a .txt format, we could easily access the data separated into columns with Matlab and plot the typical aerodynamic curves C_L-α, C_D-α and airfoil polar (C_D- C_L):

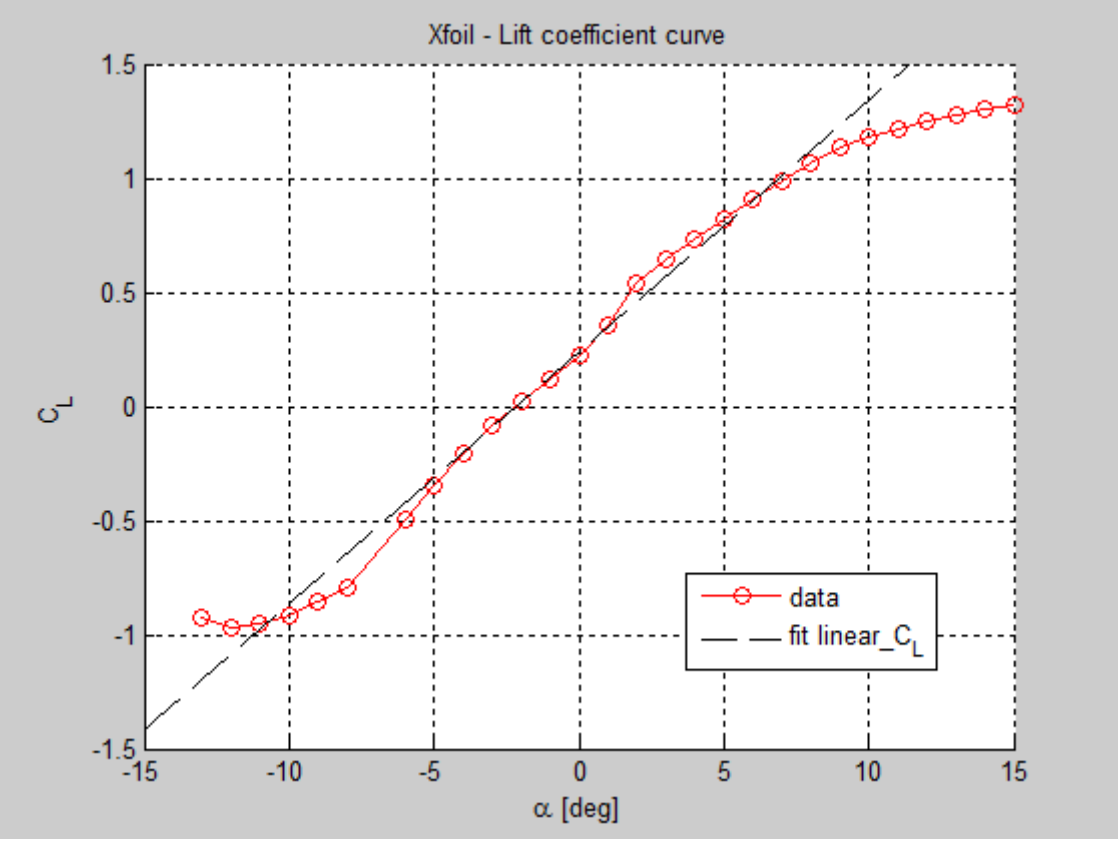


Figure 3.6 – XFOIL lift coefficient curve

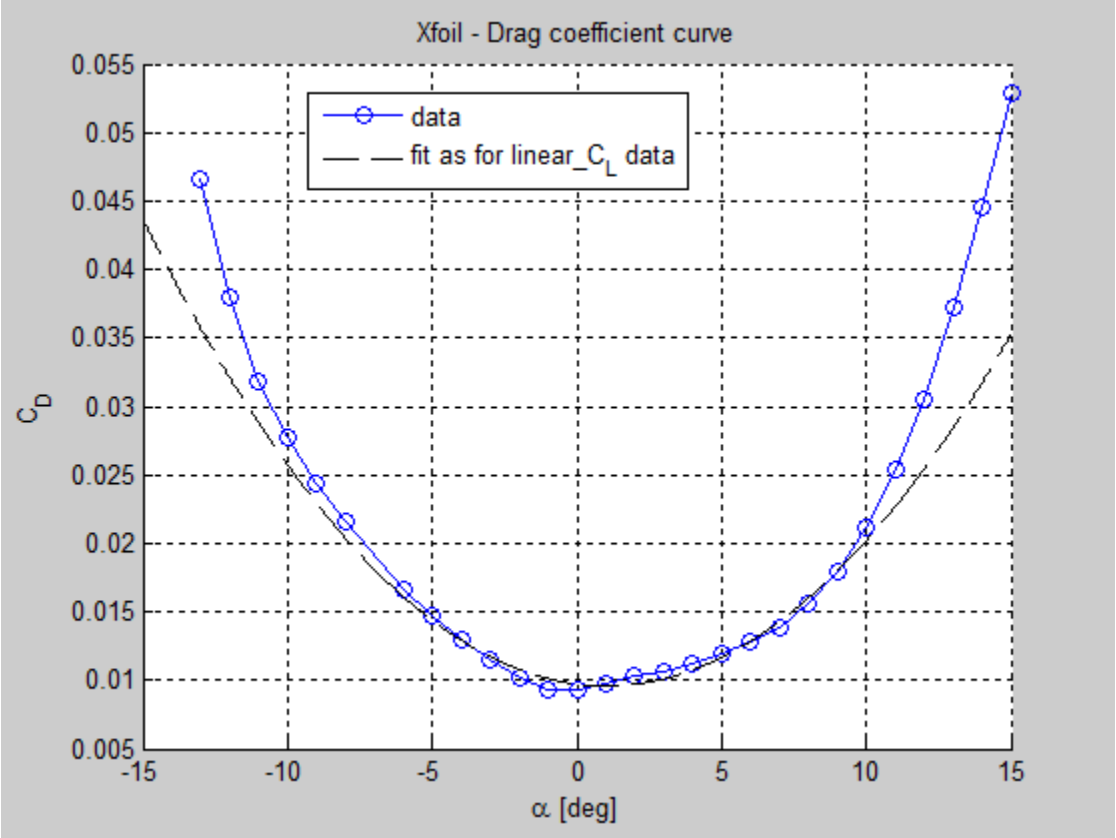


Figure 3.7 – XFOIL drag coefficient curve

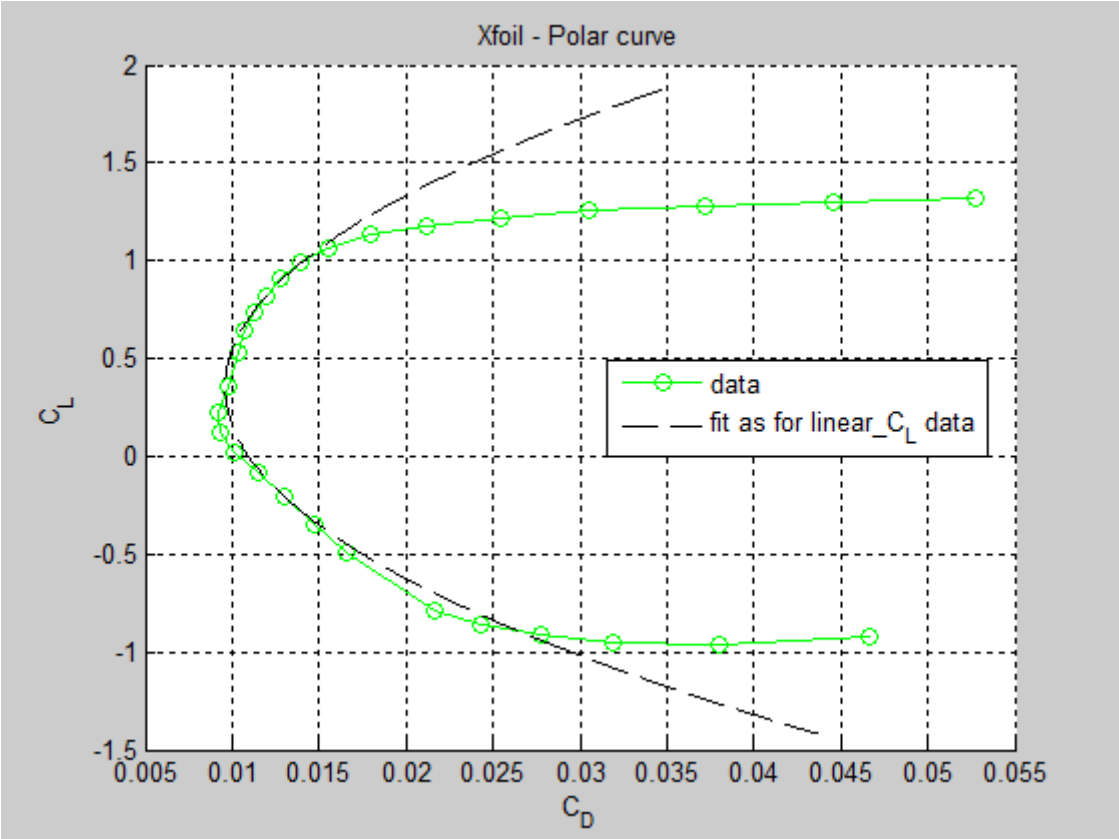


Figure 3.8 – XFOIL polar curve

3.2 Complete aircraft aerodynamics

AVL is a program for the aerodynamic and flight-dynamic analysis of rigid aircraft of arbitrary configuration. It employs an extended vortex lattice model for the lifting surfaces, together with a slender-body model for fuselages and nacelles. General nonlinear flight states can be specified. The flight dynamic analysis combines a full linearization of the aerodynamic model about any flight state, together with specified mass properties. For further reading and more info visit the official site reported in the bibliography[2].

In my opinion AVL can be a very useful tool for quickly evaluating an airplane configuration scheme. It only gives an inviscid approximation, but the data are easy and quick to calculate. Consequently, a design can be initially evaluated using AVL, but then it should be fine-tuned with a full-blown CFD code before being wind tunnel tested to ensure accuracy.

The *Cularis* geometry was created using the tools of AVL and the specifics are reported in the table below (Note: the booms with the sensors and the telemetry antenna couldn't be generated):

General	
Wing surface S	0,4366 m^2
Mean aerodynamic chord	0,1673 m
Wing span b	2,61 m
Vortex lattice - wing	
n. of swirling rings along chord line	16 (cosine distr.)
n. of swirling rings along span line	64 (cosine distr.)
pitch angle to fuselage line	3°
dihedral angle	1,5°
Vortex lattice - horizontal stabilator	
n. of swirling rings along chord line	12 (cosine distr.)
n. of swirling rings along span line	28 (cosine distr.)
pitch angle to fuselage line	1°
Vortex lattice - vertical fin	
n. of swirling rings along chord line	12 (cosine distr.)
n. of swirling rings along span line	24 (cosine distr.)
Vortex lattice - fuselage	
n. of swirling rings	28

Table 3.2 – AVL simulation geometry parameters

Once the geometry is set we could start a simulation for an angle of attack between -4 and 11 degrees:

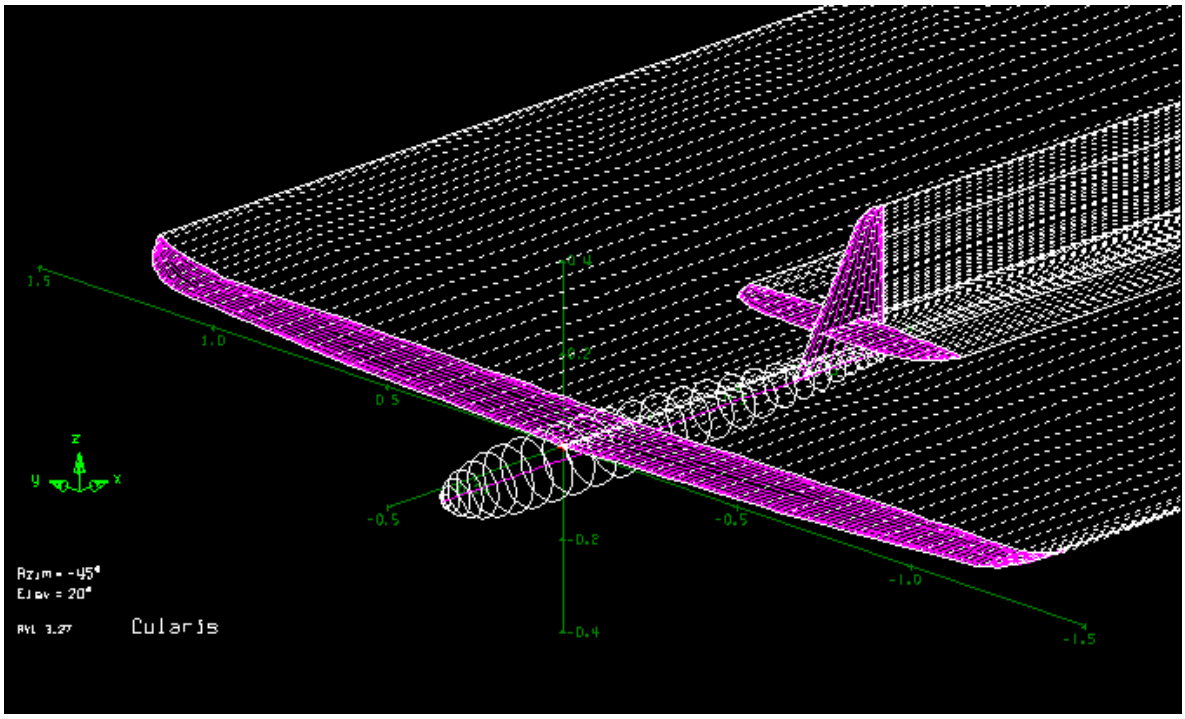


Figure 3.9 – Graphic window of AVL showing the trailing vortexes

Here we report the numerical data matrix of the results for the trimmed configuration. As it can be seen the whole column relative to the coefficient of total momentum is equal to zero. In fact the simulation respects the definition of trimmed flight that is when the pitching moment of the aircraft is zero, namely an equilibrium of momentum around its aerodynamic center.

alpha	CL	CD	Cm
----------	----------	----------	------
-3,85335	0,07000	0,01902	0
-3,06843	0,15421	0,01968	0
-2,28247	0,23842	0,02046	0
-1,49526	0,32263	0,02134	0
-0,70661	0,40684	0,02234	0
0,08368	0,49105	0,02345	0
0,87580	0,57526	0,02467	0
1,66997	0,65947	0,02600	0
2,46638	0,74368	0,02743	0
3,26527	0,82789	0,02898	0
4,06693	0,91211	0,03063	0
4,87142	0,99632	0,03238	0
5,67876	1,08050	0,03424	0
6,48967	1,16470	0,03621	0
7,30422	1,24890	0,03827	0
8,12362	1,33320	0,04044	0
8,94622	1,41740	0,04271	0
9,77325	1,50160	0,04507	0
10,60499	1,58580	0,04752	0
11,44174	1,67000	0,05007	0

Table 3.3
AVL numerical results

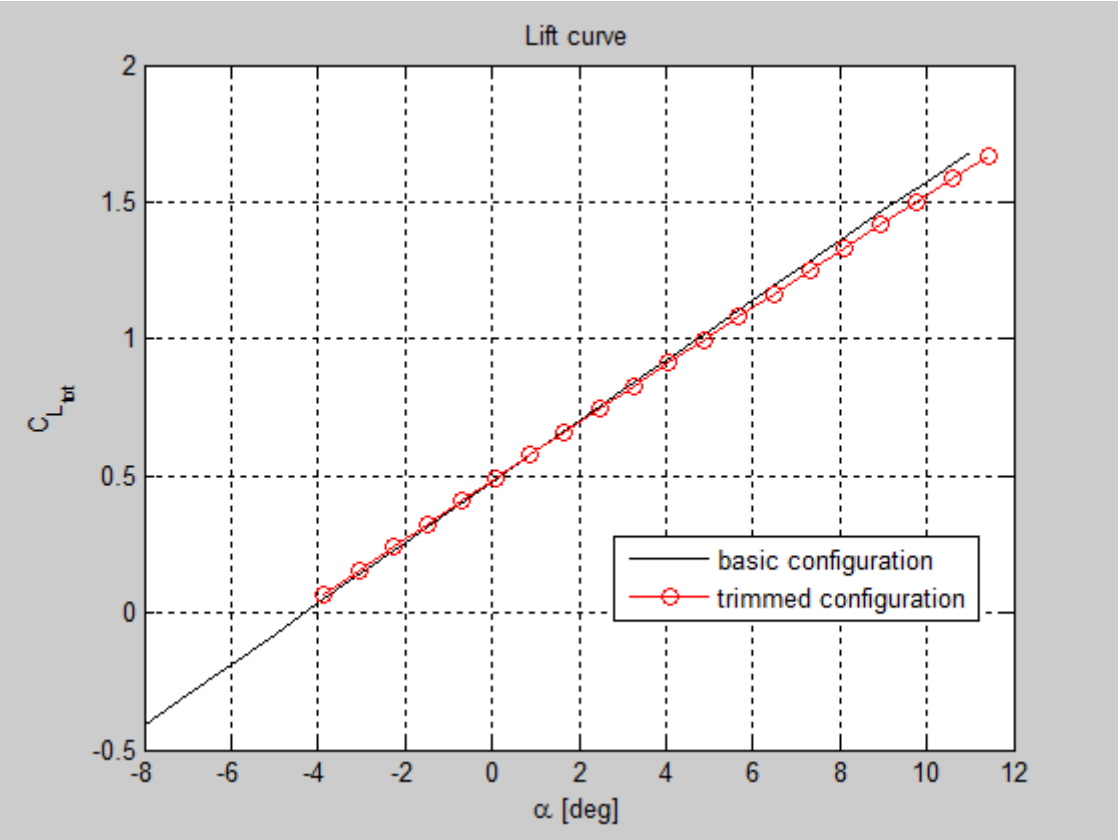


Figure 3.10 – AVL lift coefficient curve

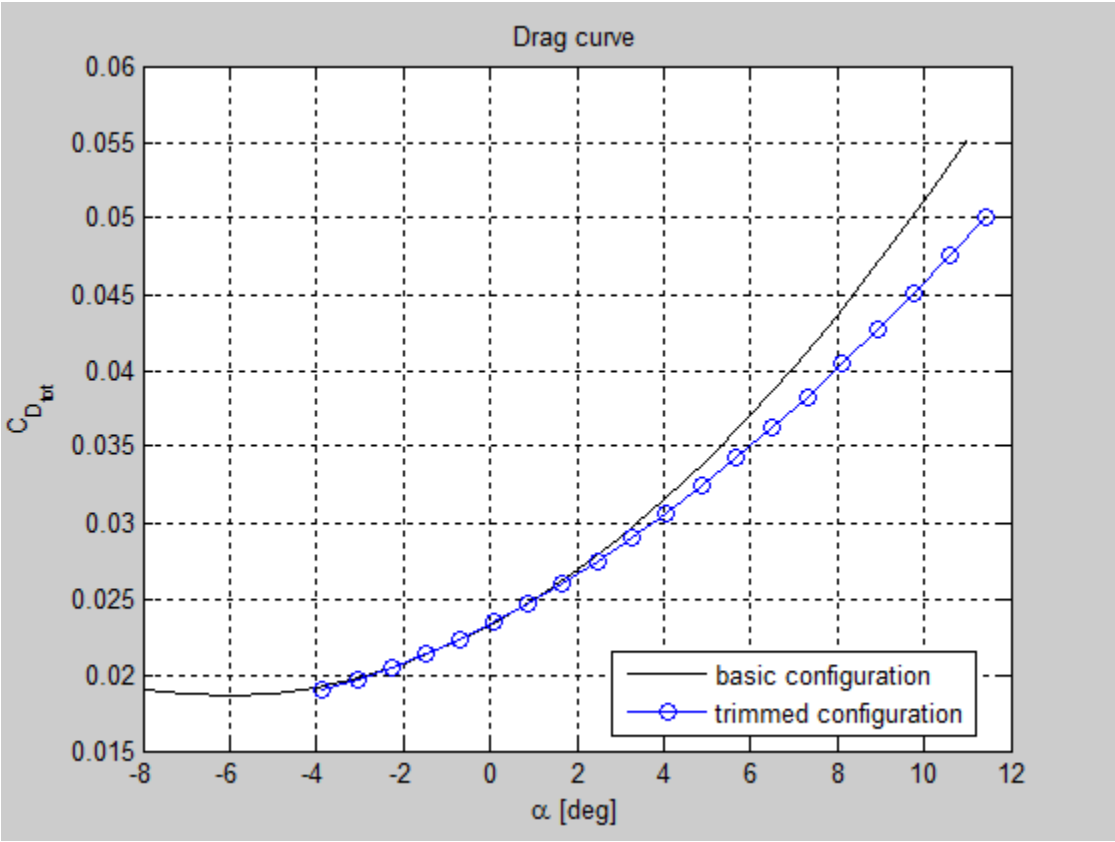


Figure 3.11 – AVL drag coefficient curve

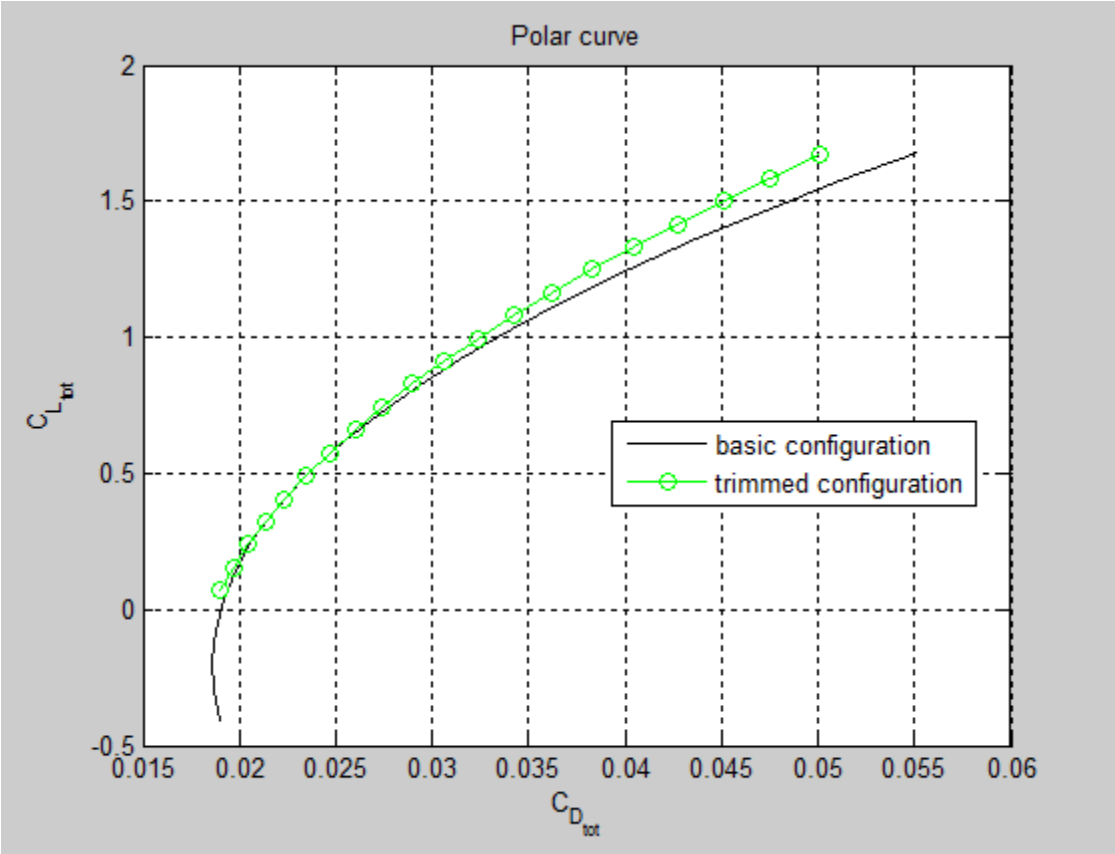


Figure 3.12 – AVL polar curve

In the three graphs above we reported both the basic configuration of the aircraft (untrimmed) and the trimmed configuration with the elevator deflected to balance out the aerodynamic momentum of the wing. Looking at the lift coefficient graph we notice that the trimmed curve is a little bit underneath the untrimmed one. This is correct because in a conventional airplane, the negative lift of the tail necessary for keeping the pitching moment equilibrium makes the overall lift decrease. Not so predictable is the case of the total drag and we notice that in this case it is less for the trimmed configuration than for the basic one improving the aircraft polar. One thing to remark however on the plausibility of the results is that the geometry model, as pointed out before, lacks of the two sub-wing booms together with their installed sensors and of the telemetry antenna sticking out from the canopy. This could lead to an underestimated resulting total drag, but sustained by the fact that the friction drag is not even taken into account by this simulation, we assume that the influence of these appendixes on the total drag coefficient can be regarded as negligible.

Of course only the trimmed results will be taken into consideration when compared to the flight testing results. The maximum aerodynamic efficiency estimated by AVL has a value of 32.

$E_{max\ (AVL)} = 32;$

4. Wind tunnel tests

4.1 Wind tunnel overview

The CIRIVE (Centro di Ricerca Ingegneria del Vento) of the Politecnico di Milano has a big wind tunnel only used by external companies (the GVPM). A smaller one directly managed by the aerospace department can be used by researchers and students for their master thesis. This last one is a subsonic flow wind tunnel in closed circuit powered by three turbines for a total of 100 kW. The test chamber is closed with a rectangular section of 1.5 m height x 1 m width and 3 m length. It is designed specifically for aeronautic applications, offering low turbulence air stream and a maximum speed of 55 m/s (~ 200 km/h). The desired true airspeed inside the test chamber is monitored by a differential pressure sensor varying the inverter frequency of the turbines to increase or decrease the speed as needed.

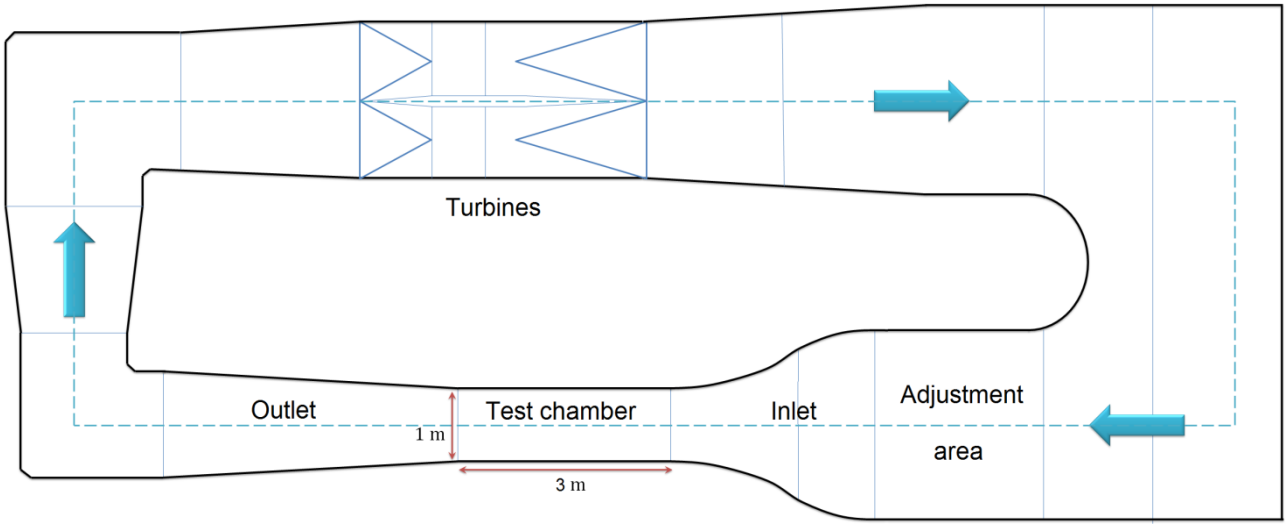


Figure 4.1 – The wind tunnel plant

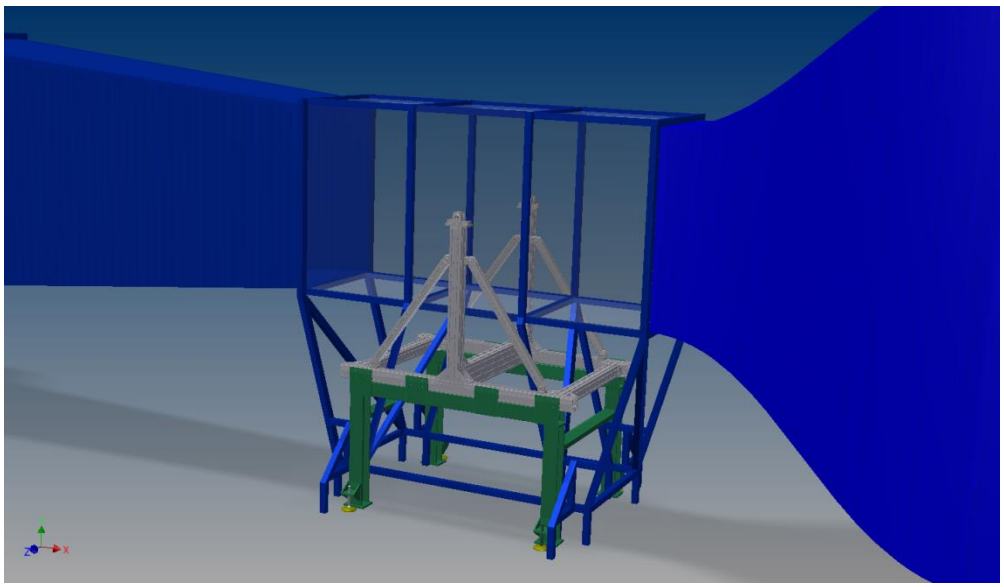


Figure 4.2 – 3D rendering of the test chamber

Figure 4.3 – Control station of the wind tunnel

4.2 Test description

4.2.1 Test objectives and set-up

The wind tunnel tests are fundamental for three reasons:

- to calibrate the air data sensors before the on field flight testing
- to get a first 'clean' experimental result for the wing aerodynamic coefficients
- to get an analytical prediction of the traction of the propeller which is a parameter not directly measurable during flight.

To be as accurate as possible in recreating the real conditions the sensors will be in during a real flight, they have been mounted (one at the time) with their original supports onto the right hand spare part semi-wing of the *Cularis.* This way, both the pitot and the AOA/S sensors will be calibrated taking already into account PEC (Positioning Error Correction), wing aerodynamic up-wash and structure flexing. Also the powering of the sensors comes from the APM analogic ports and the data acquisition system is the APM itself as it would be in real flight. While acquiring the signals coming from the sensors, another test can be performed simultaneously: the measurement of how lift and drag of the model vary while changing angle of attack by connecting the wing root to a 3-axis balance. Some errors may be taken into account, especially coming from the wall effect. In fact the semi-span of the *Cularis* is about 1.3 meters and will be mounted vertically inside the test chamber which is 1.5 m leaving out for the wing tip vortexes only 20 cm before the ceiling wall. In addition the cross section of the wind tunnel model that we are testing is quite large compared to the width of the test chamber, especially at high angles and the blockage effect can be relevant.

Figure 4.4 – The model inside the wind tunnel

As it can be seen from the picture on the left, at the base of the wing there is, fixed on the test chamber floor, a small fairing to take out from the drag measurement the contribute coming from the support of the model. The support is not touching the floor but only holds firmly the two carbon fiber spars of the wing and then it's connected itself to the pole coming out from a hole on the floor.

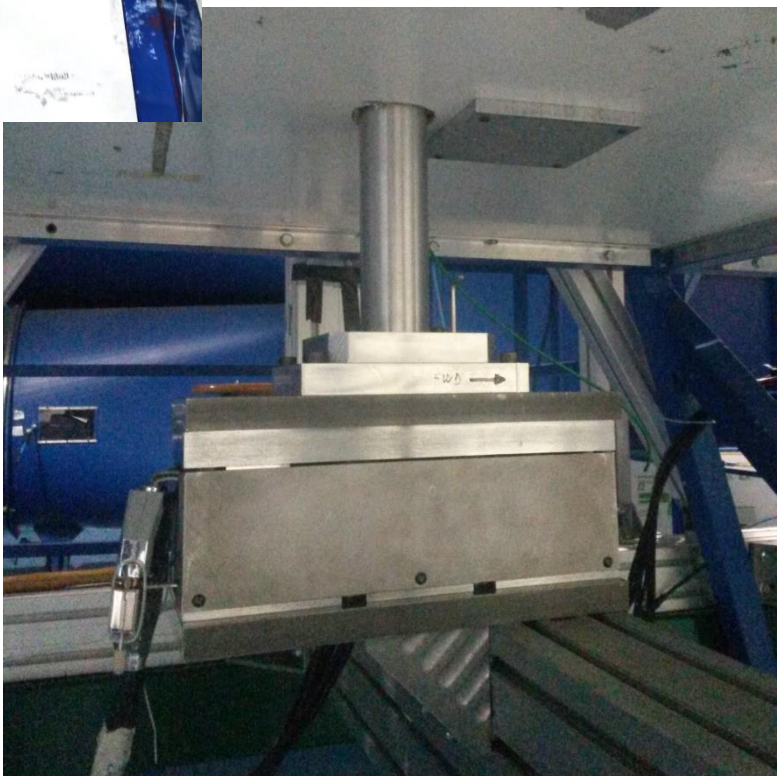


Figure 4.5 – The wing support connected to the 3-axis balance underneath the floor of the wind tunnel test chamber

Through the hole in the floor also come the cables from the sensors installed on the wing that are connected to the APM placed close to the balance (non in the picture).

In the previous thesis work by Dr. Claudio Marzo[10], another important test was performed to analytically predict the traction of the propeller at any instant varying the parameters of motor throttle % and airspeed in the test (also measurable in flight) and recording at every test point RPM value of the spinner and traction force of the whole power unit. Here is briefly reported the setup:

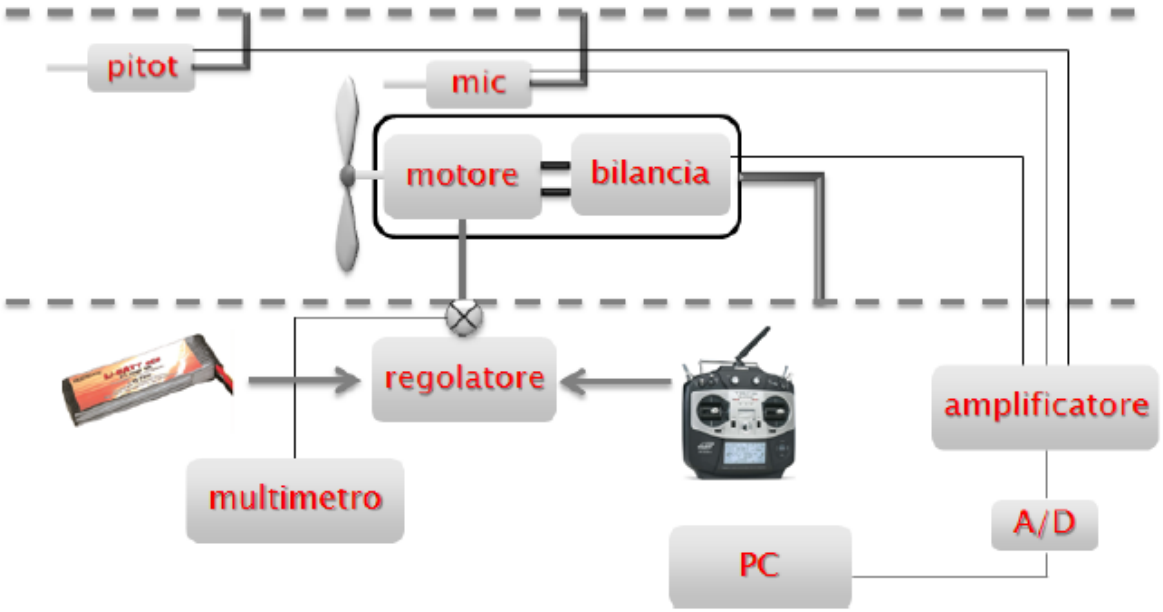


Figure 4.6 – Setup scheme for the propeller traction test

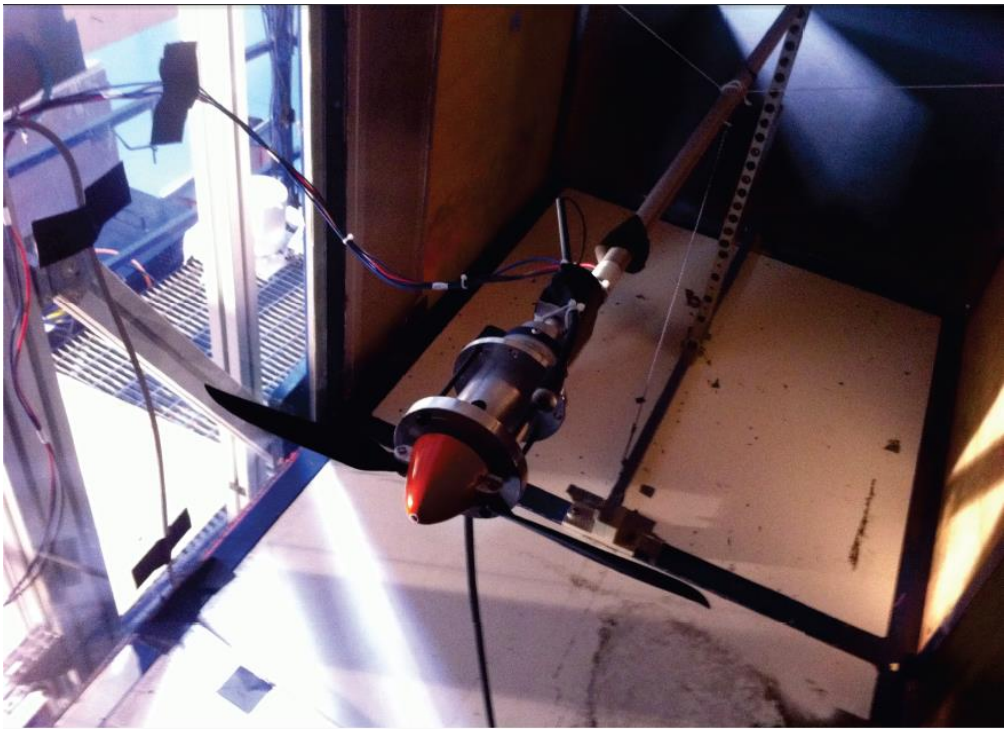


Figure 4.7 – Power unit ready inside the wind tunnel

4.2.2 Instruments

For our tests of calibration of the sensors and measurement of the wing aerodynamic parameters, we needed a support to connect the wind tunnel model to the balance. In order to be able to change angle of attack during the tests in a relative quick and easy way without measuring it every time, this support also had to rotate around its own axis. The following 3D model of the support have then been developed using *Solid Works* software, and consists of three main pieces:

1) The plate fixed on the balance. This elements has a series of small holes, arranged in two rows following a circle. Every single hole represent a position of the wing in means of angle of attack (from -20 to 20 degrees with a step of 2 degrees) and will host the place holder spine. The two slightly bigger symmetric holes are threaded and will host the screws to secure in place the moving part to this plate. In the biggest hole in the center is placed the pin around which the flange will rotate.

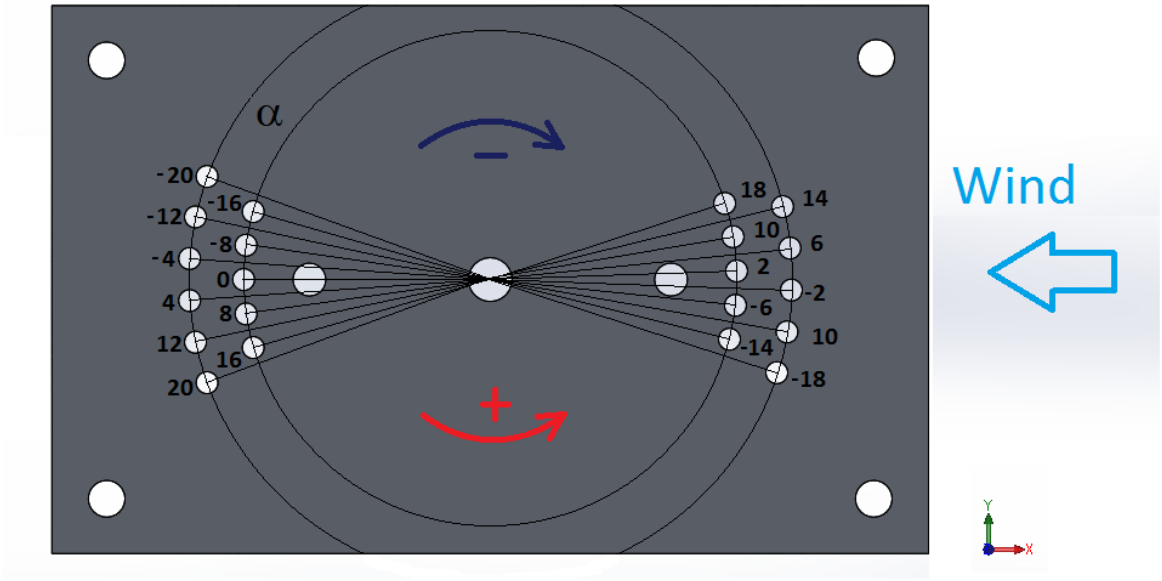


Figure 4.8 – The fixed plate on the balance with all α positions

2) The flange. This element can rotate around the center pin sliding onto the fixed plate. It is made of two pieces itself to simplify its manufacture and consists in a cylinder and a smaller plate as the base. On the base there are two semi-cylindrical slots permitting the screws to be locked to the underneath fixed plate at any of the set angle.

3) The plate that holds the wind tunnel model. This element is fixed to the other hand of the flange, on top of the cylinder. In its upper part there are two small cylinders perfectly spaced in which fit the carbon spars of the wing.

The customized support was entirely manufactured by Politecnico di Milano laboratory workshop technicians. All the elements are made of aluminum exception for the steel spines and screws. To see the executive drawings with the real dimensions, refer to Appendix B. Following, the 3D rendering of the support final assembly and the support itself after manufacturing:

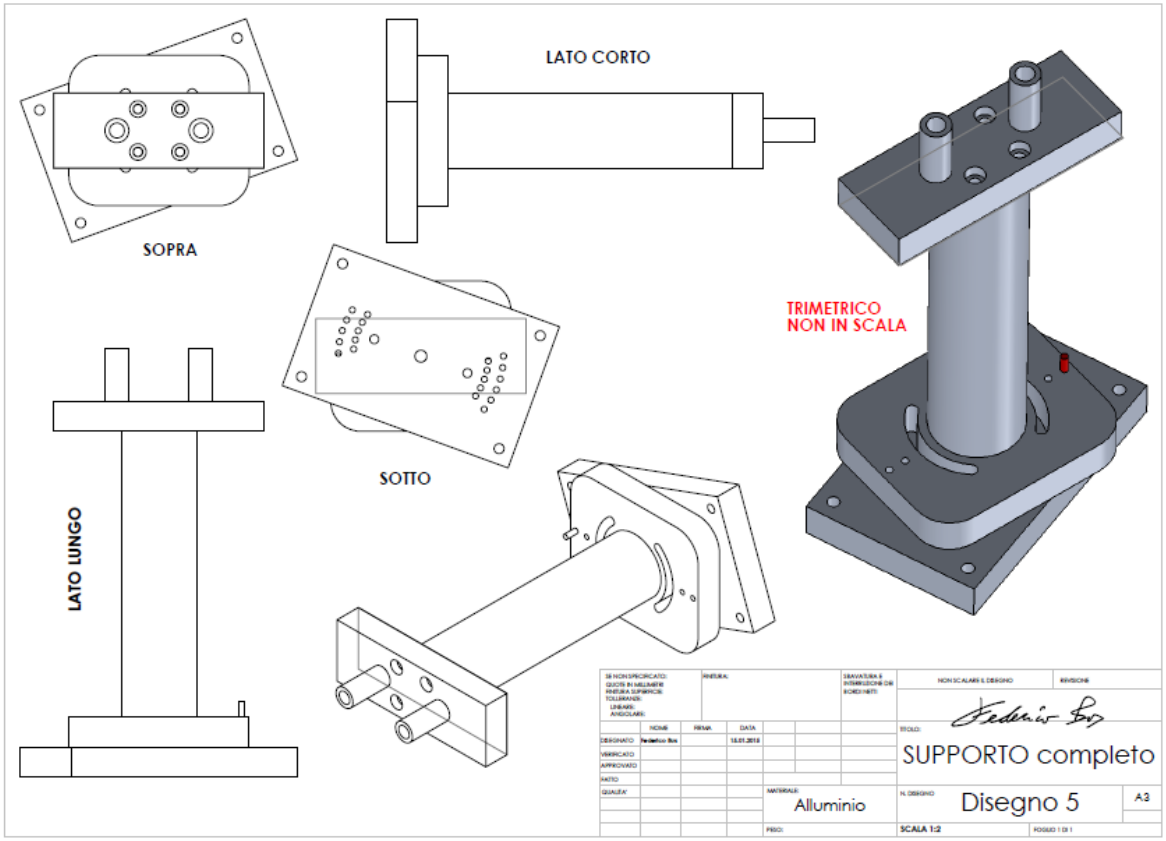


Figure 4.9 – Isometric views and trimetric 3D rendering of the assembled support

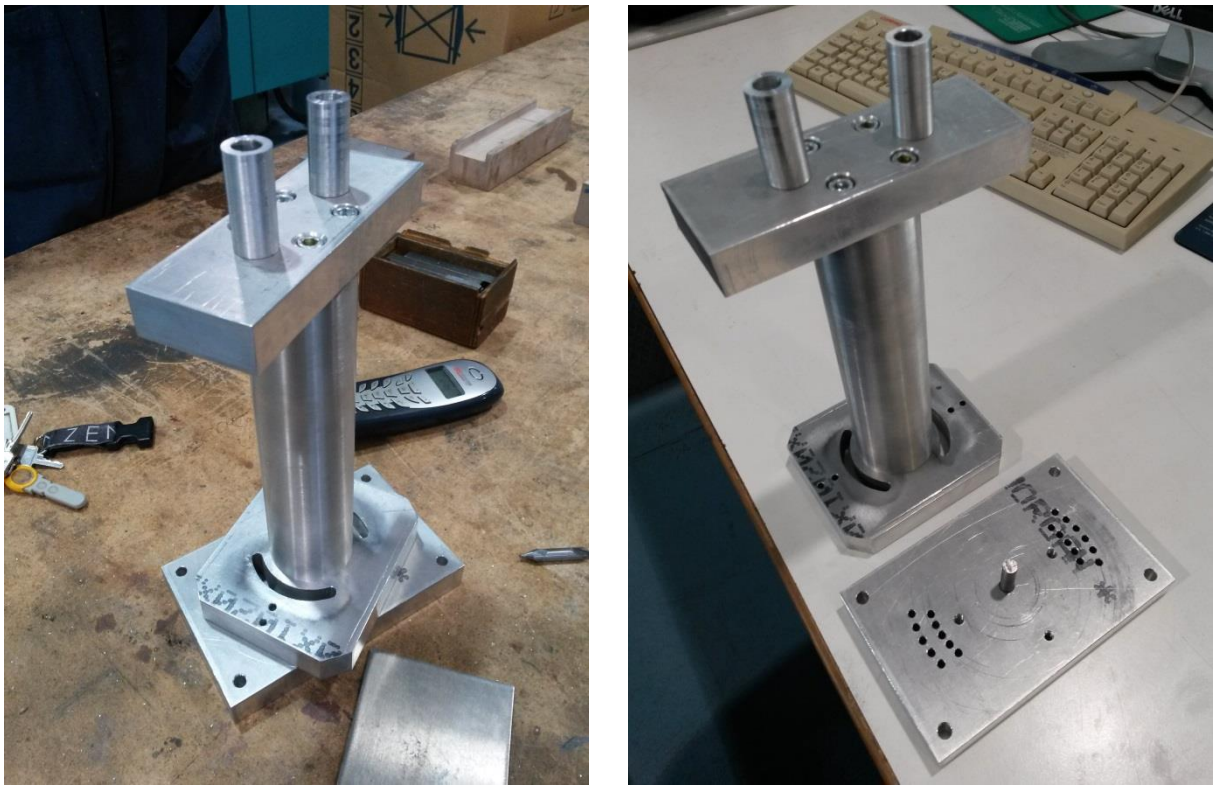


Figure 4.10 – Two different views of the finished support

To measure the two main forces (lift and drag) acting on the wind tunnel model, the *AeroLab* balance has been used. This 3-axial balance can measure forces in all three directions (x,y,z). It is a custom instrument designed by the technicians of CIRIVE and made of 7 different load cells. On the z-axis (vertical) are installed 4 cells of 250 lbf maximum each. For the later axes on the y are installed 2 cells of 250 lbf each and on the x a single one of 100 lbf. These setting result in an overall maximum measurable force of 1000 lbf (~ 4448 N) for the z-axis, 500 lbf (~ 2224 N) for the y-axis and 100 lbf (~ 444 N) for the x-axis.

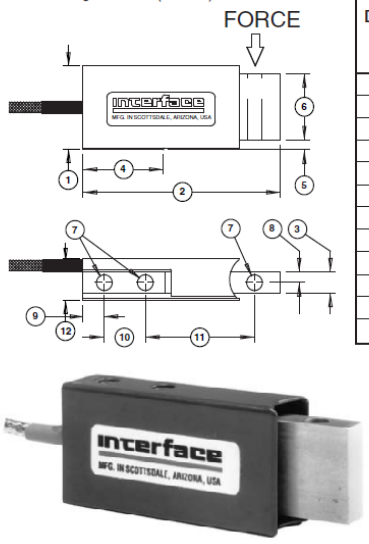
SPECIFICATIONS

ACCURACY – (MAX ERROR)
Nonlinearity–% FS ± 0.03
Hysteresis–% FS ± 0.02
Nonrepeatability–% RO ± 0.01
Creep, in 20 min.–% ± 0.025
TEMPERATURE
Compensated Range–°F 0 to 150
Operating Range–°F -65 to 200
Effect on Output–%/°F – MAX ± 0.0008
Effect on Zero–% RO/°F – MAX ± 0.0015
ELECTRICAL
Rated Output–mV/V (Nominal) 3.0
Zero Balance–% RO ± 1.0
Bridge Resistance–Ohm (Nominal) 350
Excitation Voltage – MAX 15 VDC
Insulation Resistance–Megohm 5000
MECHANICAL
Calibration Comp
Safe Overload–% CAP ± 150
Cable length–ft 5
Natural Frequency/Deflection:

lbf	Deflection (inches)	Nat. Freq. (Hertz)
75	.004	3900
100	.005	4000
150	.005	4750
250	.005	4400

STANDARD CONFIGURATION

• 5 Ft Integral Cable (MB-nn)



DIMENSIONS

See Drawing	CAPACITY (lbf) 75 inch	75 mm	100 inch	100 mm	150 inch	150 mm	250 inch	250 mm
①	1.01	25.7	1.01	25.7	1.01	25.7	1.02	25.9
②	2.38	60.5	2.38	60.5	2.38	60.5	2.38	60.5
③	0.25	6.4	0.25	6.4	0.25	6.4	0.5	12.8
④	0.97	24.6	0.97	24.6	0.97	24.6	0.97	24.6
⑤	0.14	3.6	0.13	3.3	0.1	2.5	0.12	3
⑥	0.75	19.1	0.78	19.8	0.82	20.8	0.79	20.1
⑦	0.17	4.3	0.17	4.3	0.17	4.3	0.17	4.3
⑧	0.13	3.3	0.13	3.3	0.13	3.3	0.25	6.4
⑨	0.25	6.4	0.25	6.4	0.25	6.4	0.25	6.4
⑩	0.50	12.7	0.50	12.7	0.50	12.7	0.50	12.7
⑪	1.31	33.3	1.31	33.3	1.31	33.3	1.31	33.3
⑫	0.50	12.7	0.50	12.7	0.50	12.7	0.75	19.1

Table 4.1 – Tables of specifications and dimensions of load cell

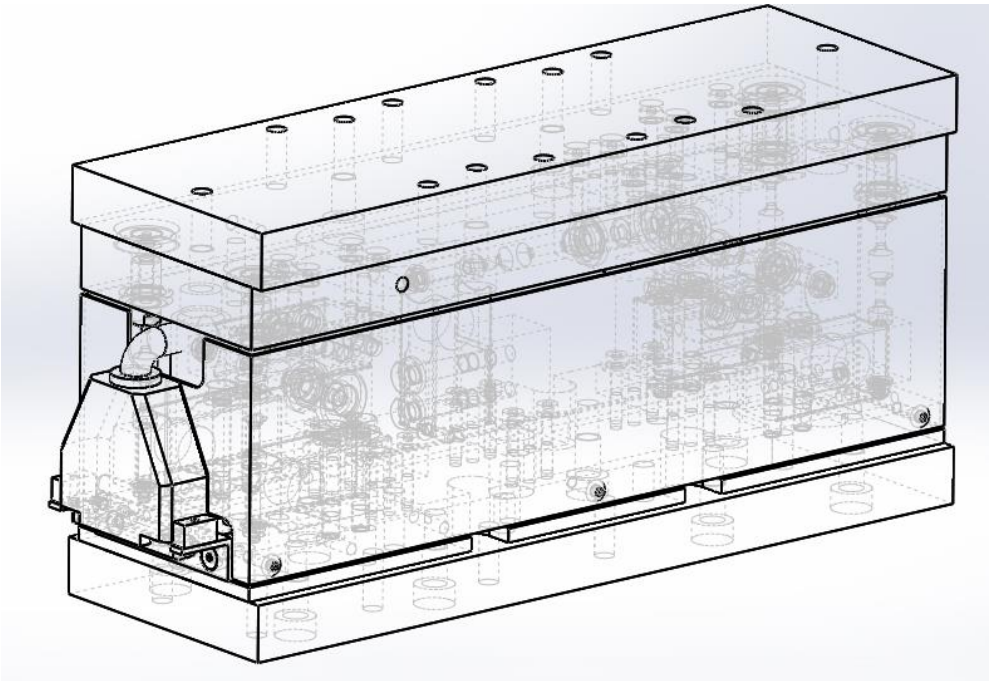


Figure 4.11 – 3D CAD model of the balance

For our purpose, we needed measurements for only two axes: on the y-axis (perpendicular to the flow direction) to measure the lift and on the x-axis (parallel to the flow direction) to measure the drag. The force on the z-axis is not important and its changes will be very small and negligible anyways. Even if the balance x and y axis are not the directions with the most available load capacity, their scale is large enough to sustain the forces of our tests (as we will see we won't go over 10 N for the drag and 25 N for the lift) and at the same time is not too large with a good instrument sensibility on those same forces.

To acquire all the data coming from the balance and from the sensors of both the wind tunnel and the model, an 18 channels system has been implemented to automatically register on a .txt file all the values of the selected data at the same time by just clicking once.

The fields of all the registered values included into the results file are:

ID, AOA [deg], Vnom [m/s], Airspeed [m/s], Din. Press. [Pa], Density [Kg/m^3], Abs. Press. [Pa], Temperature [C°], Rh [%], Pstat. Correction., F_x [N], F_y [N], F_z [N], M_x [Nm], M_y [Nm], M_z [Nm], sensor1 [V], sensor2 [V].

Figure 4.12 – Screenshot of the wind tunnel acquisition system control panel

4.2.3 Preliminary considerations

To produce enough lift force to sustain the weight of the airplane during a flight in a trimmed condition, a certain minimum airspeed threshold is necessary depending on its angle of attack. In order to have a good range of velocity for our tests to compare later on with the real flight test results, the wind tunnel airspeed had to be pushed a little beyond this threshold. To have a first estimation of the necessary airspeed, we referred to the results of the Xfoil simulation calculating the exact airspeed at which the lift of the wing will balance out the weight, given the airfoil typical coefficient of lift at a set angle of attack. One thing to be careful about is that these results will be under-estimated because the lift coefficient of an airfoil with a theoretical infinite wingspan is always higher than the correspondent lift coefficient of a finite wing. So the resulting airspeeds will be most likely less than the actual airspeeds needed for the airplane during flight testing at the same trim conditions. In addition, compared to the whole plane, here we are considering only the positive lift generated by the wing without taking into account the negative lift produced by the tail. In any cases in the wind tunnel we only tested half of the wing and the comparison is, logically, more appropriate with the results coming from the Xfoil simulation on a single airfoil than with the AVL ones for the entire airplane.

Matlab calculations using the Xfoil simulation resultant coefficients of lift:

Airfoil: NACA 2414
Wing area: 0.436 m^2
Aircraft weight: 2.10 kg (original: 1.68 kg)
Wing loading: 4.8165 kg/m^2 (original: 3.8532 kg/m^2)

Trim condition: $L = W = \frac{1}{2} \rho V^2 C_L S$

alpha:

0	3	6	9	12
1	4	7	10	13
2	5	8	11	14

Lift coefficient:

0.2240	0.6444	0.9056	1.1335	1.2530
0.3582	0.7324	0.9896	1.1809	1.2770
0.5364	0.8197	1.0673	1.2171	1.3028

Trim speed:

18.5482	10.9358	9.2248	8.2455	7.8424
14.6678	10.2578	8.8246	8.0783	7.7684
11.9862	9.6961	8.4973	7.9573	7.6911

Looking at the results, we have to remember that the wing has a fixed pitch angle relative to the reference line of the fuselage of 3 degrees. So actually when flying flat and level, the plane will start already at 3 degrees of angle of attack instead of zero. This is also what we considered when we run the Xfoil simulation and so the minimum airspeed to trim the airplane at 3 degrees angle of attack, meaning zero pitch of the fuselage, should be, according to this calculation, 10.9358 m/s. It is convenient to say here also that for the AVL simulation things were done differently, in fact the software didn't include the angle of attack of the wing relative to the fuselage line but started for the α sweep at zero angle of attack for a zero pitch angle of the whole plane. So to compare the results, to the AVL ones we should remember to add 3 degrees by default to all the AOA vectors. Talking now about the wind tunnel tests, we took as reference line the line connecting the two carbon spars which is also the chord line of the airfoil and we started from here to count the increasing of angle of attack. So these tests are directly comparable to the Xfoil ones considering the chord line of the wing parallel to the wind stream direction as zero angle of attack.

4.2.4 Test schedule

The wind tunnel test campaign is made of 4 tests:

1) Pitot calibration – this test is made of 3 runs. Each run will have a fixed angle of attack (0, 2 and 4) and the airspeed will be changed in each run from 5 m/s up to 20 m/s. The goal is to calibrate the pitot on wing in a good airspeed range accounting for all possible inherent disturbances (e.g. PEC).

2) Pitot error at high AOA – this test is also made of 3 runs but this time the airspeed will be fixed for each run (10, 12, 14 m/s) and the angle of attack will be changed in each run from a minimum of -20 degrees to a maximum of +20 degrees in steps of 2 degrees. The goal is to acquire a first polar for the wing and see if the reading of the airspeed sensor becomes inaccurate at high angles of attack due to the not parallel stream direction relative to the tube orientation.

3) AOA sensor calibration on wing – as the test number 2, this also is made of 3 runs keeping the airspeed fixed and changing the angle of attack. The goal is to acquire a second polar and to calibrate the AOA sensor installed on wing.

4) AOA and AOS sensors calibration off wing – this test is no more performed using the wing as support for the sensors. A simple L-shape rod secured on the rotating platform will support the sensors. In this test, keeping fixed the airspeed at 12 m/s and changing angles, the goal is to also calibrate the AOS sensor in means of β-sweep using the same rotating platform (the test chamber of the wind tunnel is not large enough to fit the semi-wing in horizontal position and the manufacturing of another tilting support was too long and complicated). Using the L-shape new support we were able, by simply rotating the sensors of 90 degrees, to also acquire the results for the AOA sensor off wing at the same airspeed with the idea of comparing it with its previous on-wing calibration.

4.2.5 Safety considerations

Set the schedule with all the test points (216 in total), with a maximum airspeed value of 20 m/s well higher than the 10.9358 m/s recommended as trim airspeed by Xfoil, and high angles of attack reaching 20 degrees, one last consideration had to be made: will be safe in terms of generated forces to push the semi-wing structure too far? This question is also legitimate by the added extra weight that wasn't included into the original project of the *Cularis*. Again using the coefficients of lift calculated by Xfoil for the set angle of attack and airspeeds of the wind tunnel tests, we have been able to calculate with Matlab an estimation of the forces acting onto the model structure for the most severe test points. Here the results in terms of G-force, considering that one semi-wing would take half of the total aircraft weight:

```
PITOT calibration test
@  0 deg, 20 m/s --> 1.1627 G
@  2 deg, 20 m/s --> 2.7842 G
@  4 deg, 20 m/s --> 3.8015 G

Most critical conditions
@ 15 deg, 14 m/s --> 3.3676 G
Force on semi-wing: 3.536 Kg (34.6529 N) <-- stall? Vibrations?!
@  4 deg, 20 m/s --> 3.8015 G
Force on semi-wing: 3.9916 Kg (39.1175 N) <-- MAX
>>
```

Assuming that an airplane model can go way faster in dives than at 20 m/s (72 km/h) and its wings are designed to be quite elastic to sustain forces, especially during sharp turns, most likely exceeding 4 G, we are quite confident that there won't be problems during wind tunnel tests in terms of safety. Also, in this case, the predictions for the resultant forces will be over-estimated (safe) for the same reason as before when they were under-estimated for the trim speed calculations; the coefficient of lift computed by Xfoil for a theoretically infinite wingspan are for sure higher in value than the ones for a real finite wing so the resultant lift at the same airspeed will be anyways lower in the real case than calculated. The angle at which the stall may occur may also be different and most likely at a lower angle of attack for the real model than for the simulated one. At stall angle, the lift will quickly decrease and drag will increase instead. The drag force, even for angles of attacks higher than the stall angle, will be anyways lower than the resultant lift and doesn't represent a problem.

In any case if the flexion or the vibration of the wing might be considered dangerous, the wind tunnel turbines can be stopped immediately at any point during the tests.

4.3 Test results

In this section, all the results from the data collected in two full days of tests will be presented. For each test, all the environmental data as average air density, pressure, temperature and humidity are reported. Only the graphs summarizing the results of all the runs together will be included together with a brief description for the most interesting numerical results.

4.3.1 Test 1: Pitot calibration

PROVA1
Density: 1.2042 [Kg/m^3]
Absolute Pressure: 100600.6036 [Pa]
Temperature: 17.1444 [C°]
Relative Humidity: 34.612 [%]

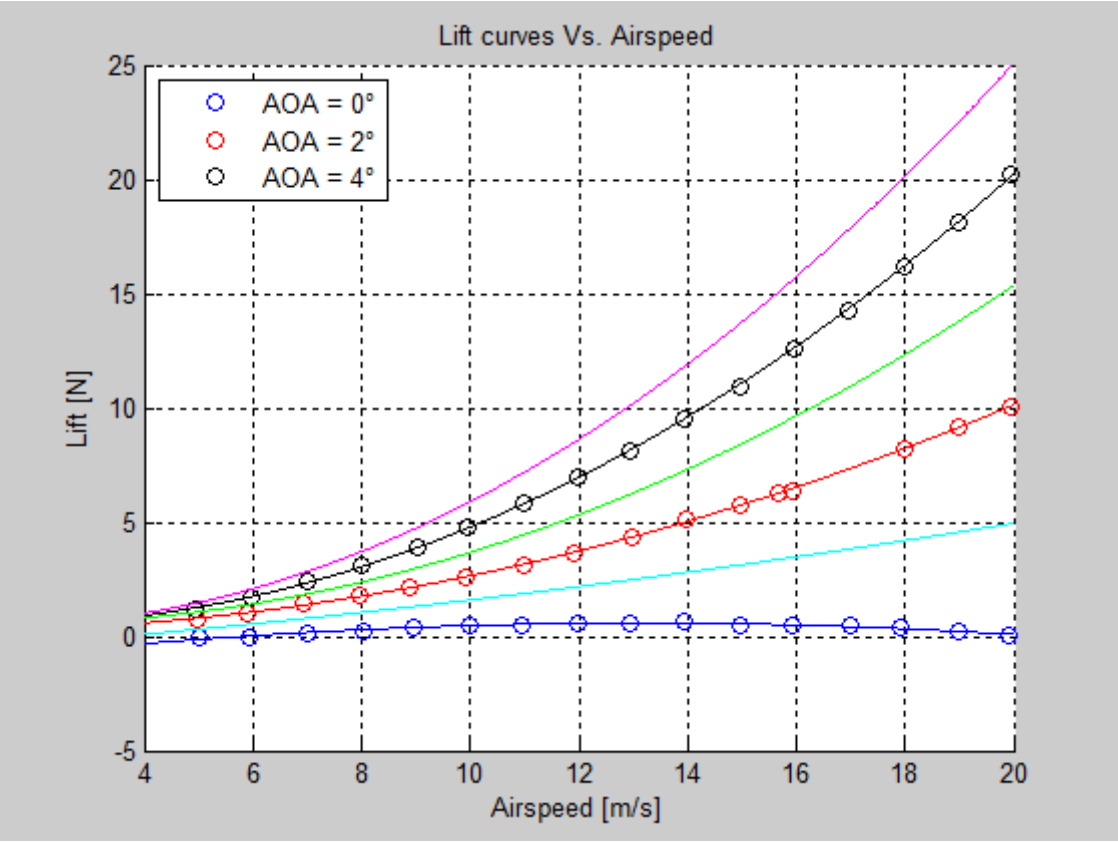


Figure 4.13 – Test 1: Lift Vs. airspeed curves

The above plot shows the raw data of lift force as a function of airspeed at different AOA. It can be observed like for a zero angle of attack the lift is close to zero for any values of airspeed; then increasing both the AOA or the wind tunnel airspeed, the lift also increases. The light blue, green and magenta lines represent the mean results calculated in between the other experimental data fitting lines for the unavailable AOA positions of, respectively, 1, 3 and 5 degrees.

Lift generated by two semi-wings (full wing):

AOA = 0 [deg], V = 10 [m/s] --> Lift = 0.93714 [N]
AOA = 0 [deg], V = 15 [m/s] --> Lift = 1.0322 [N]
AOA = 0 [deg], V = 20 [m/s] --> Lift = 0.040748 [N]

AOA = 2 [deg], V = 10 [m/s] --> Lift = 5.2336 [N]
AOA = 2 [deg], V = 15 [m/s] --> Lift = 11.489 [N]
AOA = 2 [deg], V = 20 [m/s] --> Lift = 20.1598 [N]

AOA = 4 [deg], V = 10 [m/s] --> Lift = 9.4905 [N]
AOA = 4 [deg], V = 15 [m/s] --> Lift = 21.8524 [N]
AOA = 4 [deg], V = 20 [m/s] --> Lift = 40.4193 [N]

However, to get to know at which airspeed the airplane will have to cruise in a trimmed condition, disposing of the lift trends at also 3° and 5° of angle of attack is important:

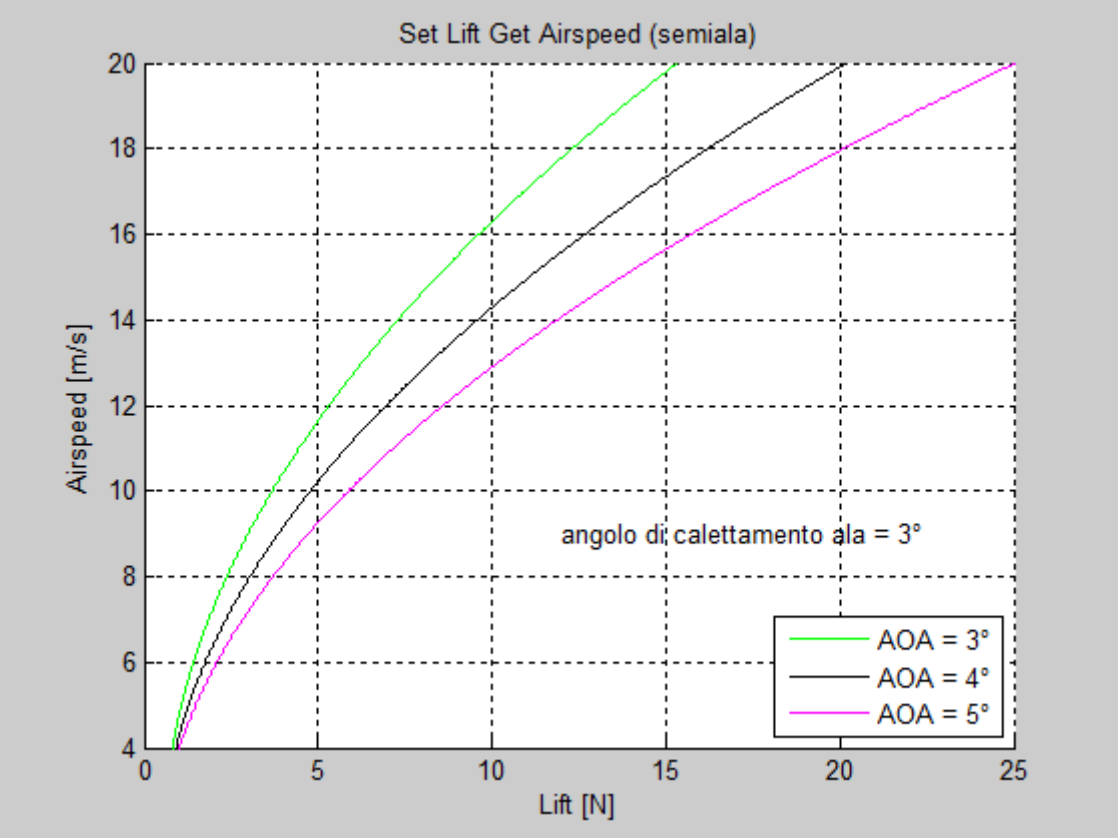


Figure 4.14 – Inverse function: Setting AOA and lift, get airspeed

SET ALPHA and LIFT, GET AIRSPEED

(Aircraft weight force: 20.58 [N])
Lift = 20.5952 [N], AOA = 5 --> V = 13.07 [m/s]
Lift = 20.5681 [N], AOA = 4 --> V = 14.47 [m/s]
Lift = 20.5817 [N], AOA = 3 (wing pitch angle vs. fuselage line) --> V = 16.51 [m/s]

As it shows the highlighted result for the speed at 3° of angle of attack at trim condition, the actual value of the wind tunnel is higher than the Xfoil prediction as expected. Likewise, also the lift force actually generated by the semi-wing in the "worst-case-scenario" is much lower than what it has been preventively calculated with the Xfoil lift coefficients. This last result of slightly more than a half of the lift that has been predicted shows how much over-estimated are the coefficients of lift calculated using Xfoil. Also because of this reason, the safety of the test have never been critical with just a normal flexion resulting in a few centimeters of displacement at the wing tip and no vibrations.

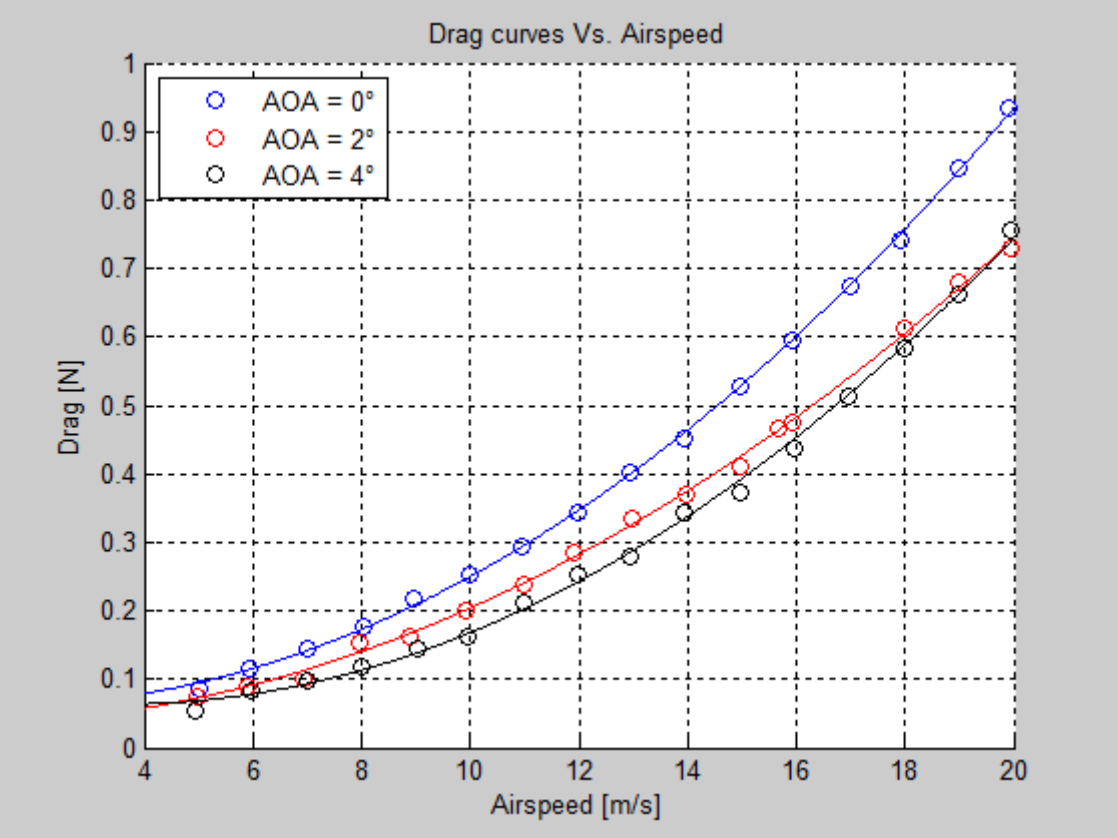


Figure 4.15 – Test 1: Drag Vs. airspeed curves

The above plot shows the raw data of drag force as a function of airspeed at the different angles of attack. As it can be observed, may actually seem abnormal, the drag decreases slightly at a bit higher angles than at zero angle of attack. This can actually be considered quite normal for an asymmetric airfoil but here the changes in the numerical values are also very small and the balance used as instrument to measure the drag in a wind tunnel test is known for not always being very reliable.

Talking now about the main goal of this first test which is the calibration of airspeed sensor, the overall graph of the results for the three runs is shown in the next figure:

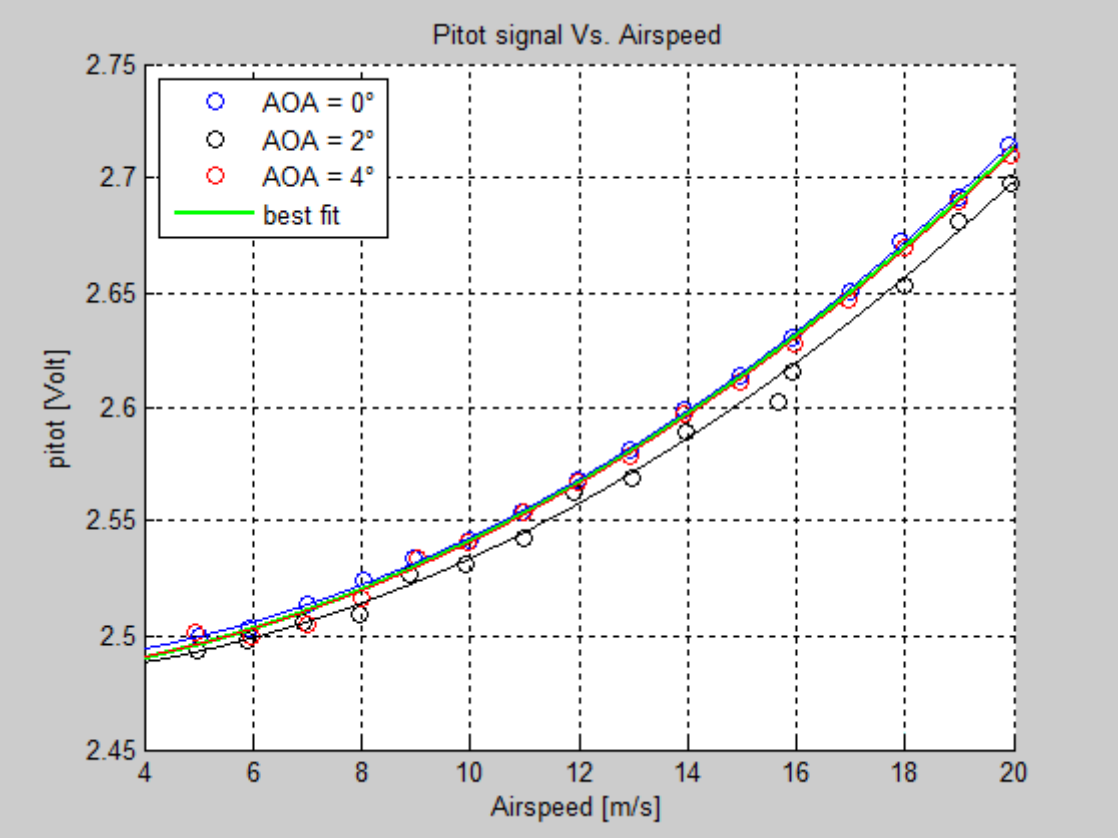


Figure 4.16 – Test 1: Pitot signal Vs airspeed

Pitot calibration fitting (x = Volt, P = Airspeed)

$P = (-159.7592)*x^2 + (897.617)*x + (-1239.5945)$

As it can be noticed from the graph above, we have very little dispersion of the data also at small airspeed values and this indicates a good sensibility of our instrument. The data also show a very regular trend and a quadratic fitting can follow almost perfectly its curve. Not relevant changes can be observed varying the angle of attack but higher values will be reached in the next test specifically intended to observe this problem, if existent.

Having now the analytic formula of the polynomial curve fitting this data reported above, in order to know the airspeed being measure by the pitot tube given the electric output of the airspeed sensor, it is sufficient to invert this function. However, because the airspeed sensor kit is a standard optional part of the APM system, the code managing it's integration was already existent and for its calibration and correct reading it has been sufficient to insert into the code the coefficient that had to be calculated until convergence as instructed by the following iterative formula:

$NEW\_RATIO=OLD\_RATIO*((AVERAGE\_AIRSPEED+ARSP\_BIAS)/AVERAGE\_ARSP)^2$

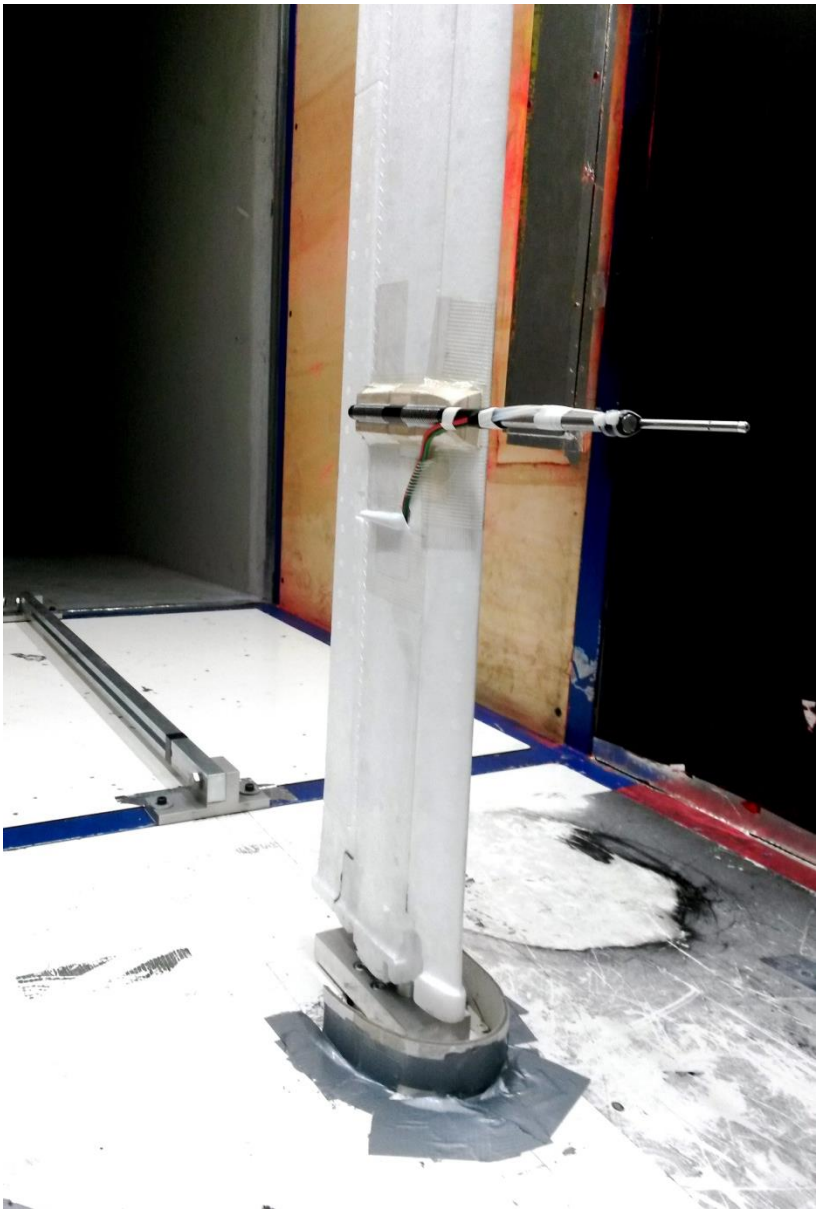


Figure 4.17 – Particular of the wind tunnel model with the airspeed sensor installed

The front tip of the pitot tube where the small openings of the pressure ports are (total pressure in the front and static pressure on the side as it can be seen), is quite far from the wing leading edge (the boom is approximately 15 cm long plus the length of the pitot tube itself) and from the fuselage (the support is installed on the wing at approximately 30 cm from the wing root). This said, we are quite confident that the measurements are not very affected by differences in pressure due to proximity to the airplane surfaces; anyways this test was primary done to account for this kind of errors (PEC – Positioning Errors Correction).

4.3.2 Test 2: Pitot error at high AOA

PROVA2
Density: 1.202 [Kg/m^3]
Absolute Pressure: 100485.5402 [Pa]
Temperature: 17.2191 [C°]
Relative Humidity: 40.856 [%]

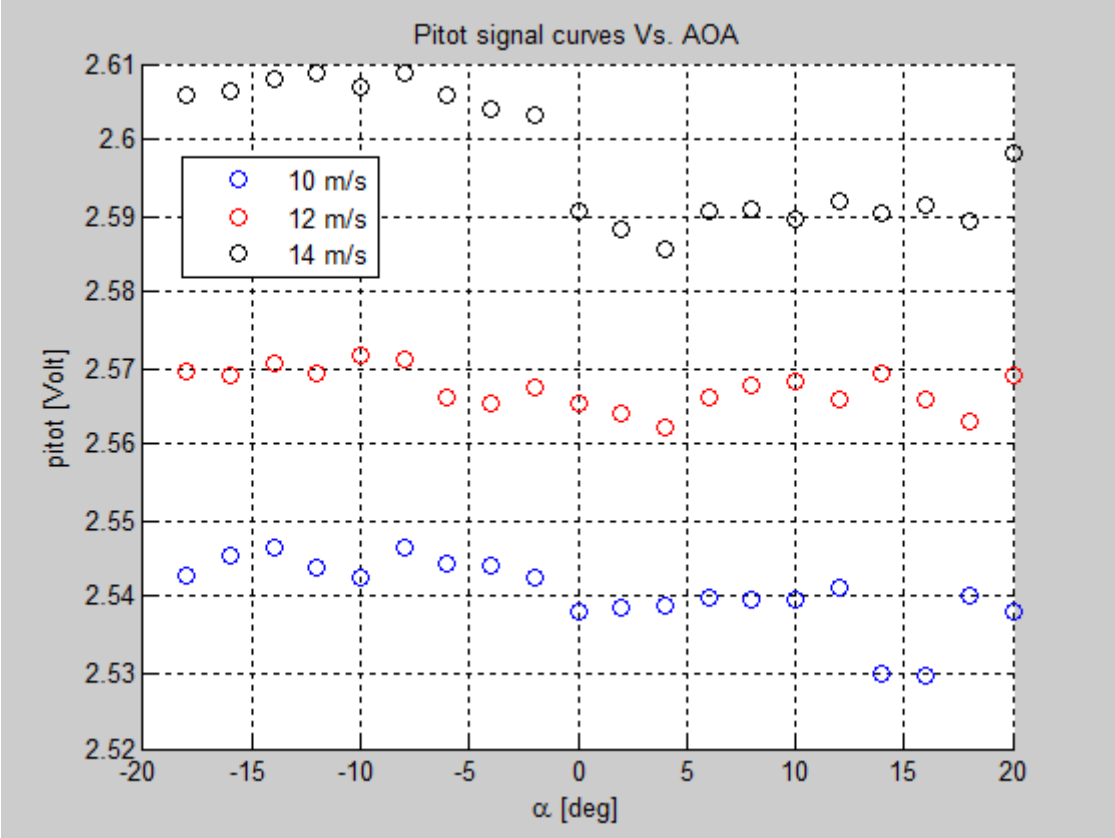


Figure 4.18 – Pitot signal Vs. AOA at different speeds

Average @ 10 m/s: 2.5405 [Volt] , delta = 0.016930
Average @ 12 m/s: 2.5672 [Volt] , delta = 0.009359
Average @ 14 m/s: 2.5980 [Volt] , delta = 0.023239

From these results it can be stated that the airspeed sensor is negligibly affected by the changes in angle of attack in the tested range from -20 to +20 degrees. In fact there isn't a clear trend in the randomly changing measurement error and its variation is anyways quite small.

Now, because all of the lift and drag forces have been measured and recorded at all angles of attack as well, it has been possible to produce a first polar typical of the wing being tested. All the overall graphs of the balance measures for the three runs are reported below:

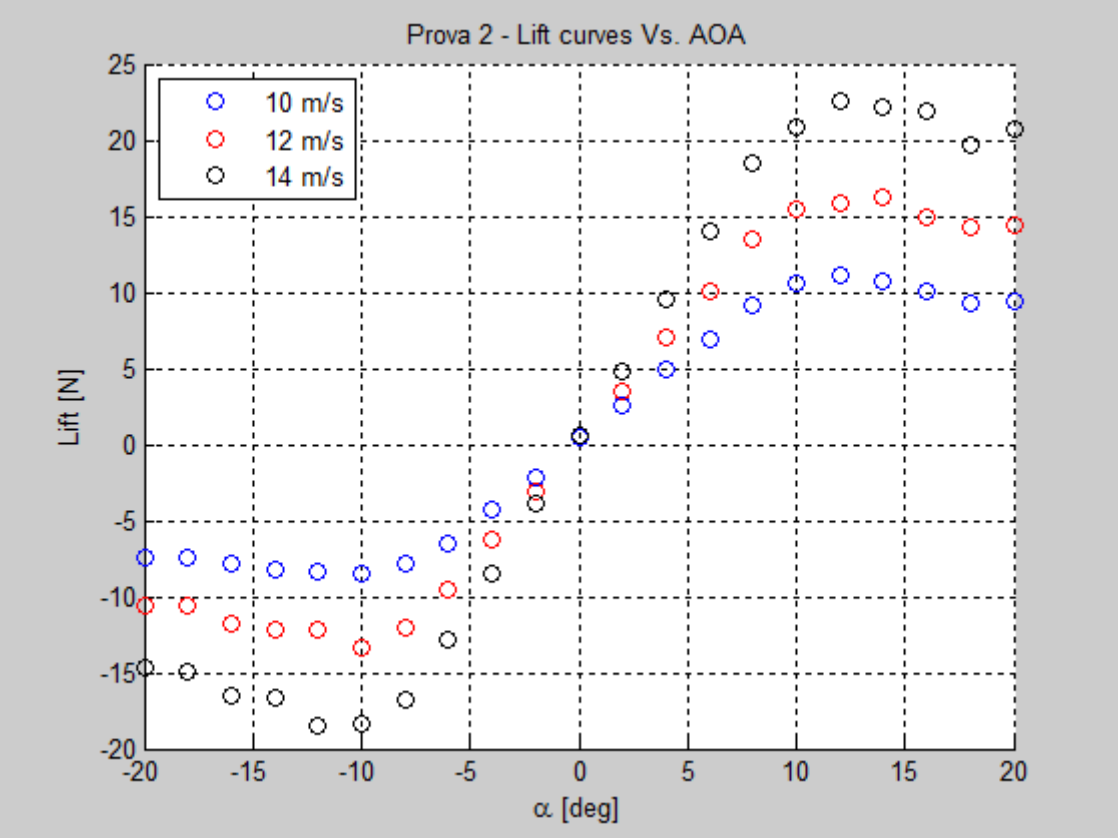


Figure 4.19 – Test 2: Lift force Vs. AOA

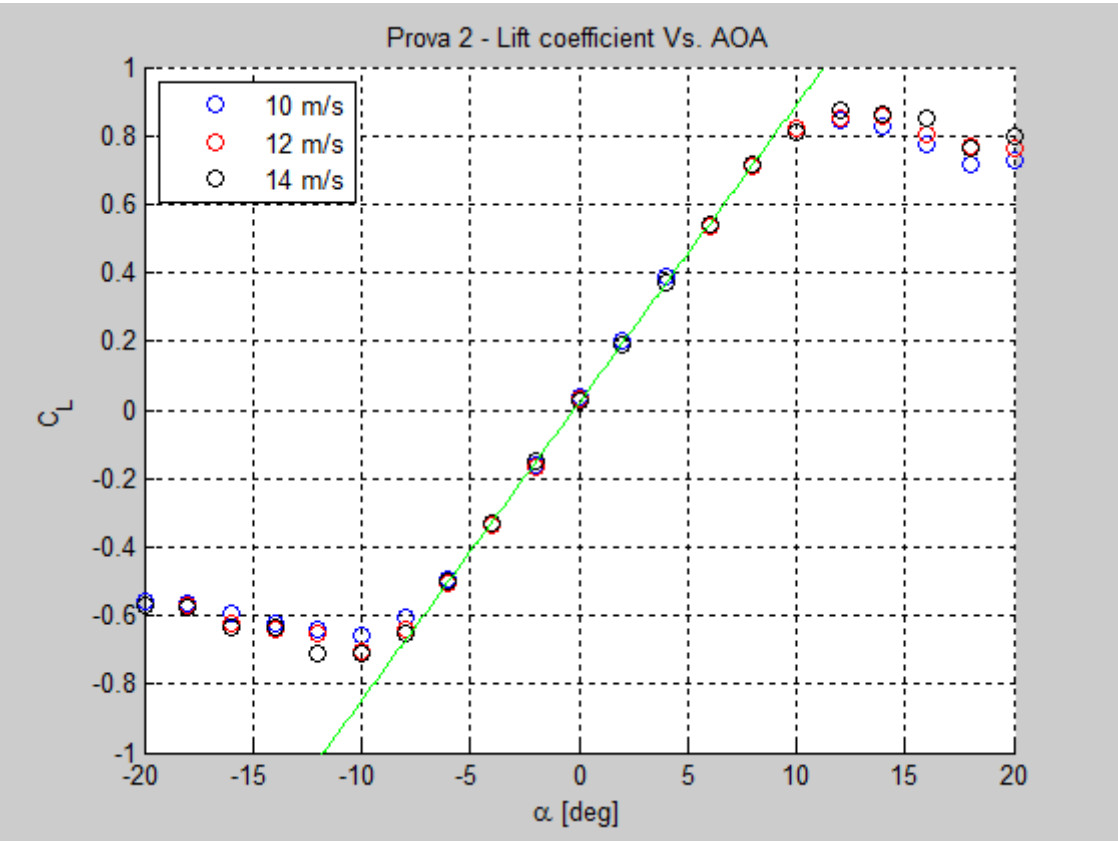


Figure 4.20 – Test 2: Lift coefficient Vs. AOA

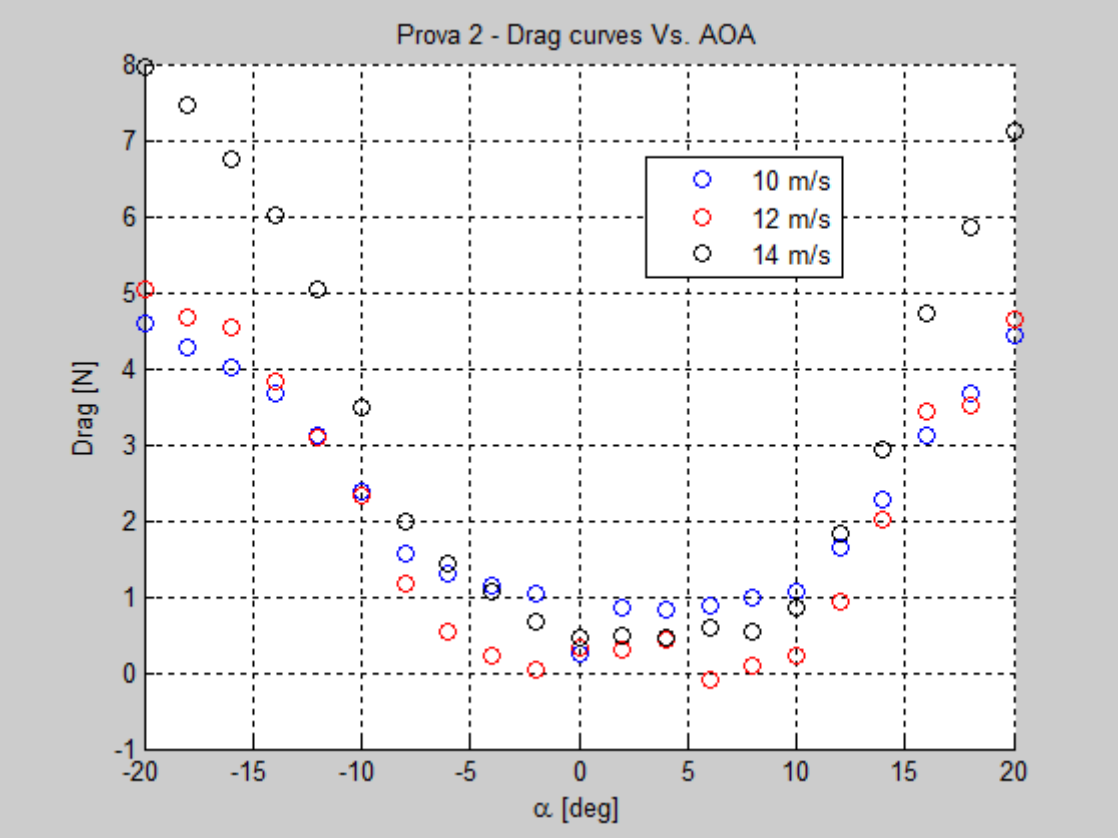


Figure 4.21 – Test 2: Drag force Vs. AOA

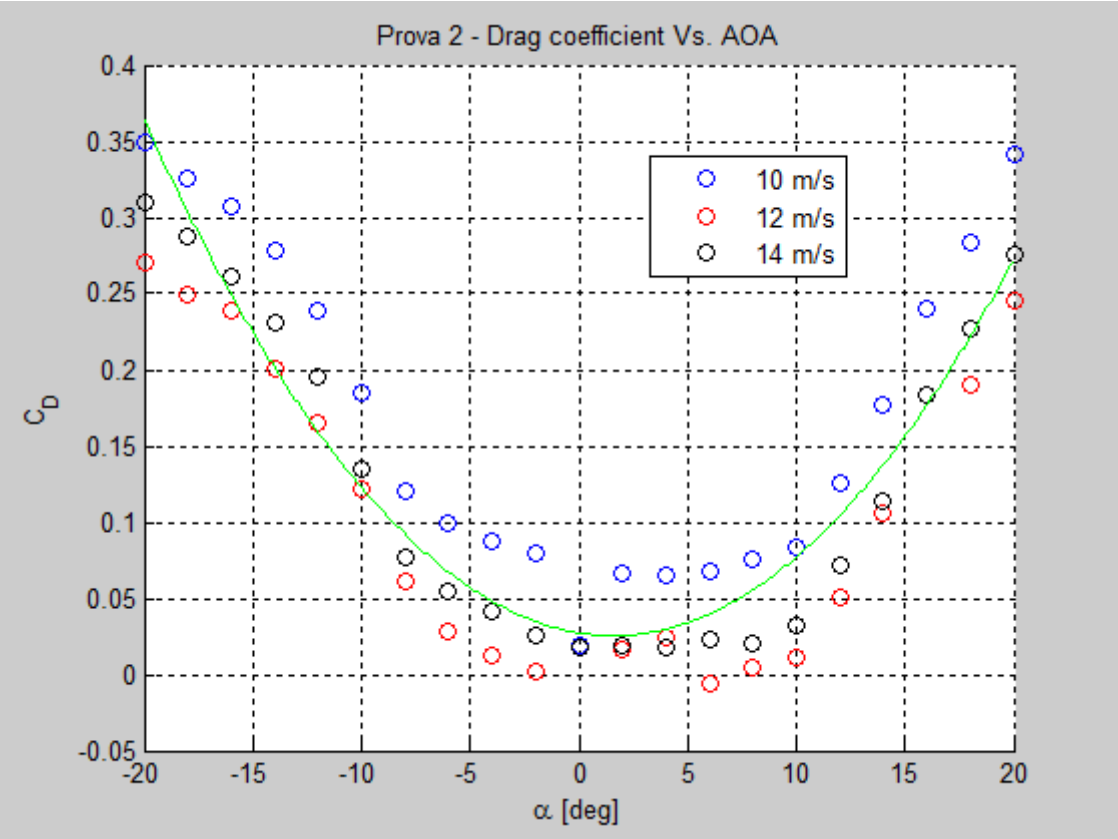


Figure 4.22 – Test 2: Drag coefficient Vs. AOA

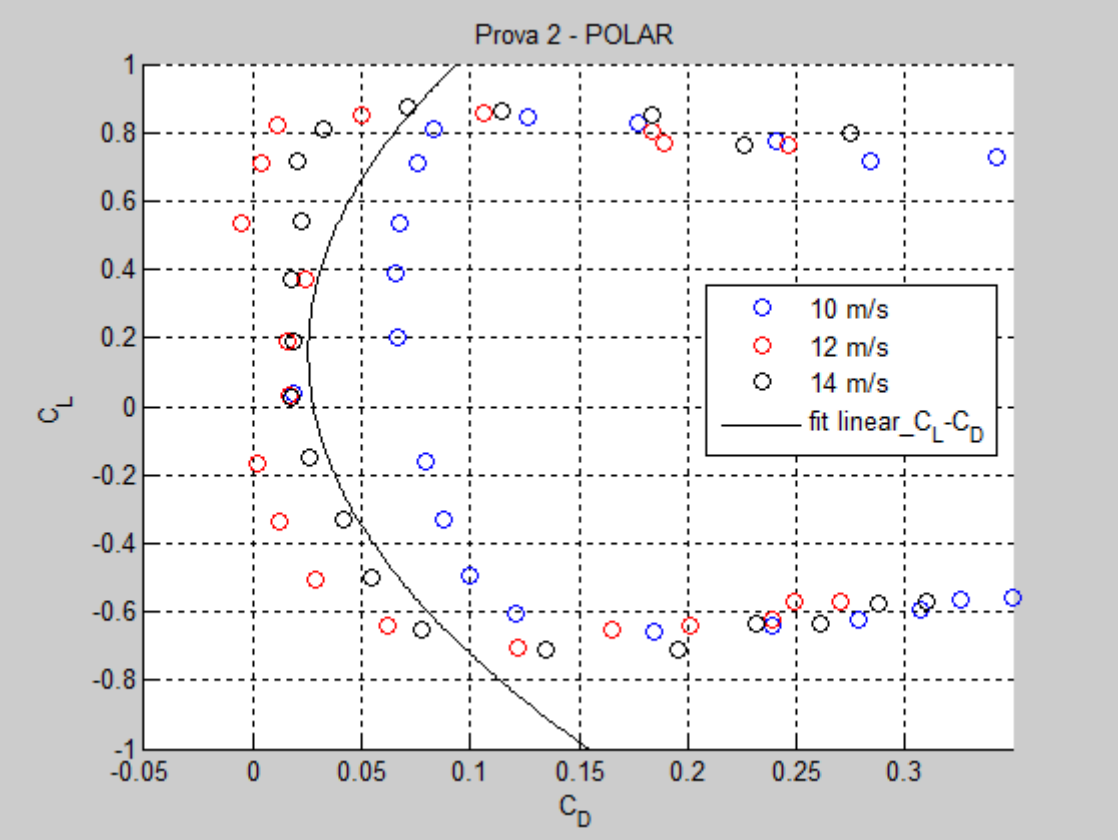


Figure 4.23 – Test 2: Aerodynamic polar

Looking at the graphs above, some comments have to be made:

The overall results for the lift and coefficient of lift are very clean and reliable with all data points lying on their line without major disturbances. For the lift force in figure 4.19, we can clearly observe how the slopes of the linear intervals increases as the airspeed increases resulting in a higher force for higher velocity. The angle at which stall occurs is very well defined at 12° angle of attack for positive angles and -10° for negative angles. In addition the three set of data for the three different velocity all crosses each other exactly at zero angle of attack. In the coefficient of lift, figure 4.20, the good quality of the recorded data is confirmed by the fact that in the linear interval (-6 to 8 degrees of angle of attack), all the points really are on the same line overlapping themselves. This also shows how the boundary layer on the wing surface was laminar at that time, only becoming turbulent close to the stall angles with the points of the three different airspeeds starting to be a little distanced from each other.

For the graphs of drag (figure 4.21) and coefficient of drag (figure 4.22), things are not quite the same. The measurement of drag for a wing in the wind tunnel using a balance is already quite difficult for many aspects influencing it like the intrinsic turbulence of the wake and wall effects that can't be avoided and usually more sophisticated methods are used. This resulted in a not very clean C_D-α curve with the three sets of data crossing each other and having relative high bias. One thing to be said in defense of the balance though is that the measures here are a lot smaller compared to the lift case especially at low angles of attack where the drag force is not higher than 1 N.

However, for our intents, the approximation of the coefficient of drag curve can be considered acceptable taking the mean of the three sets of data and a first resulting aerodynamic polar for only the aircraft wing could be traced in the linear interval of the coefficient of lift (figure 4.23).

4.3.3 Test 3: AOA sensor calibration *on wing*

PROVA3
Density: 1.2039 [Kg/m^3]
Absolute Pressure: 100411.0914 [Pa]
Temperature: 16.6477 [C°]
Relative Humidity: 36.1237 [%]

The main goal of this test is to calibrate the AOA sensor directly on the wing to account for all the kinds of disturbances that could affect its measure. However, because also in this test all the range of angle of attack had been investigated at the same three airspeeds and the lift and drag data have been recorded as well, the results of the balance measures will be presented at first in relation to the results of the previous test. As it will be seen, also for this test, the results are very close to the previous ones with just a little improvement in the drag force graph in terms of data distribution for the three velocity sets. The comments made for the previous test results are anyways valid for these graphs too. The lift and drag results of this test will then only be listed below waiting for a direct comparison in section 4.3.6:

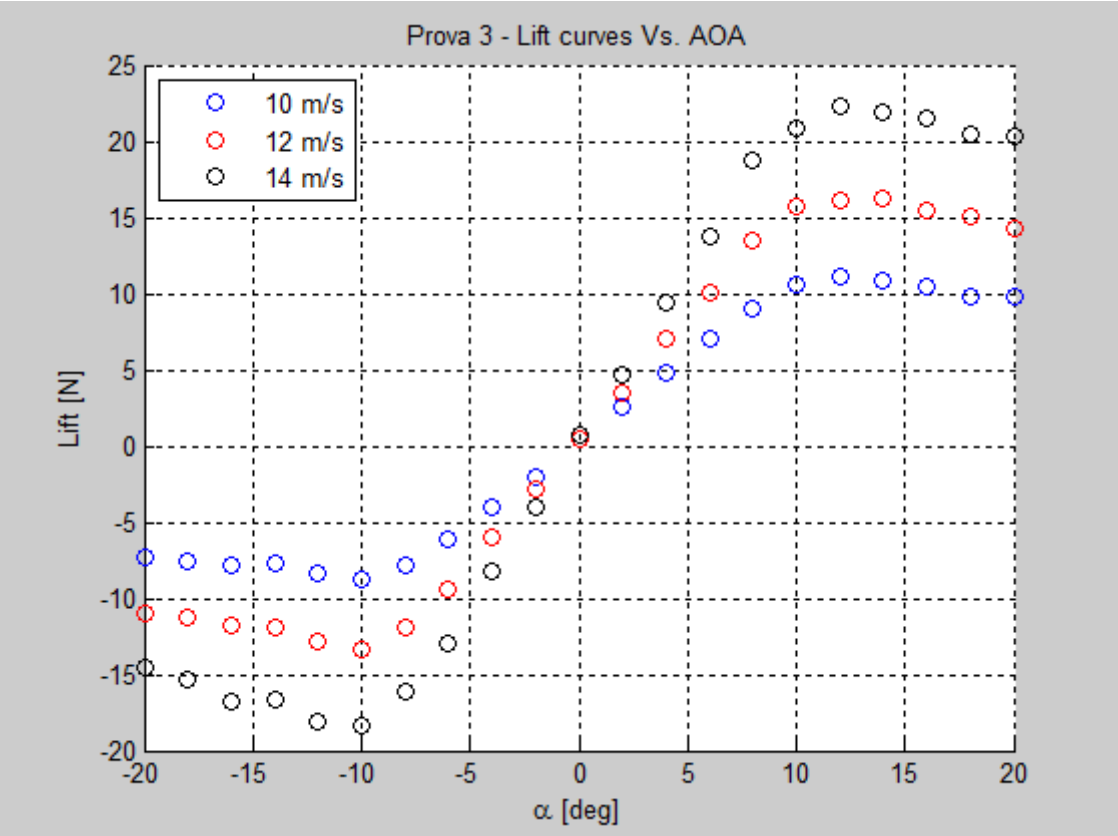


Figure 4.24 – Test 3: Lift force Vs. AOA

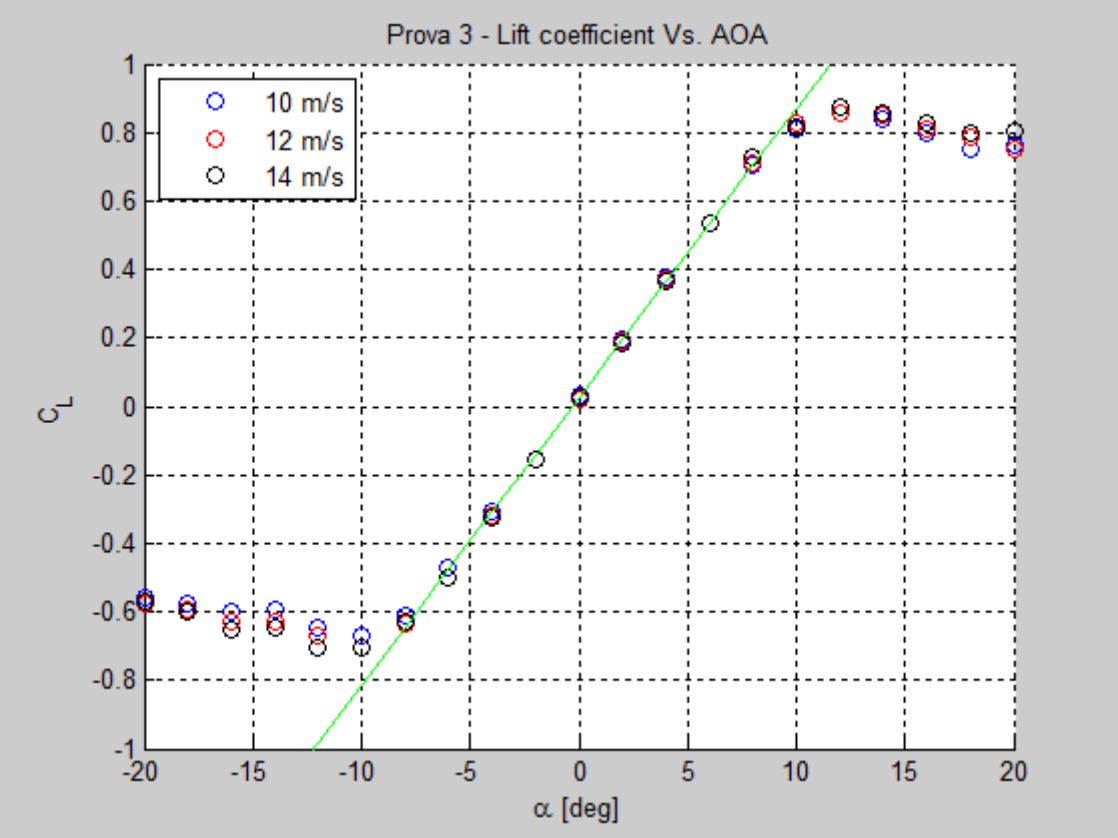


Figure 4.25 – Test 3: Lift coefficient Vs. AOA

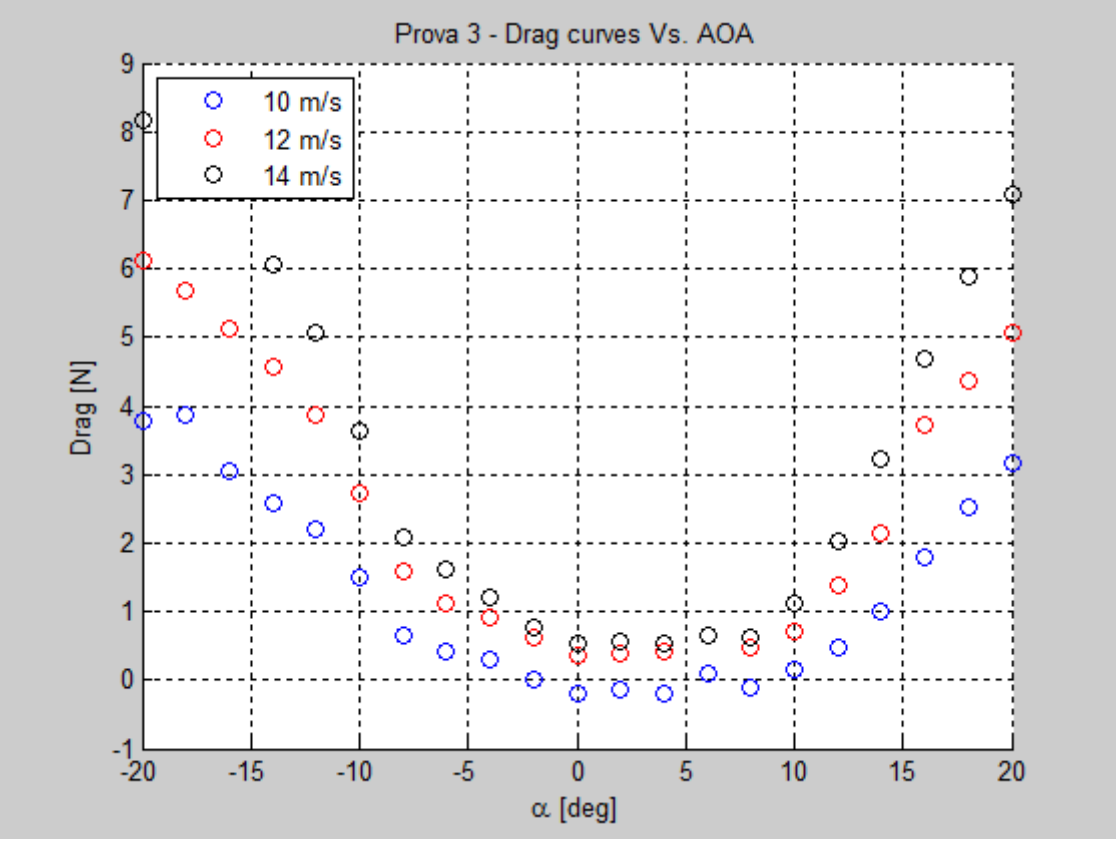


Figure 4.26 – Test 3: Drag force Vs. AOA

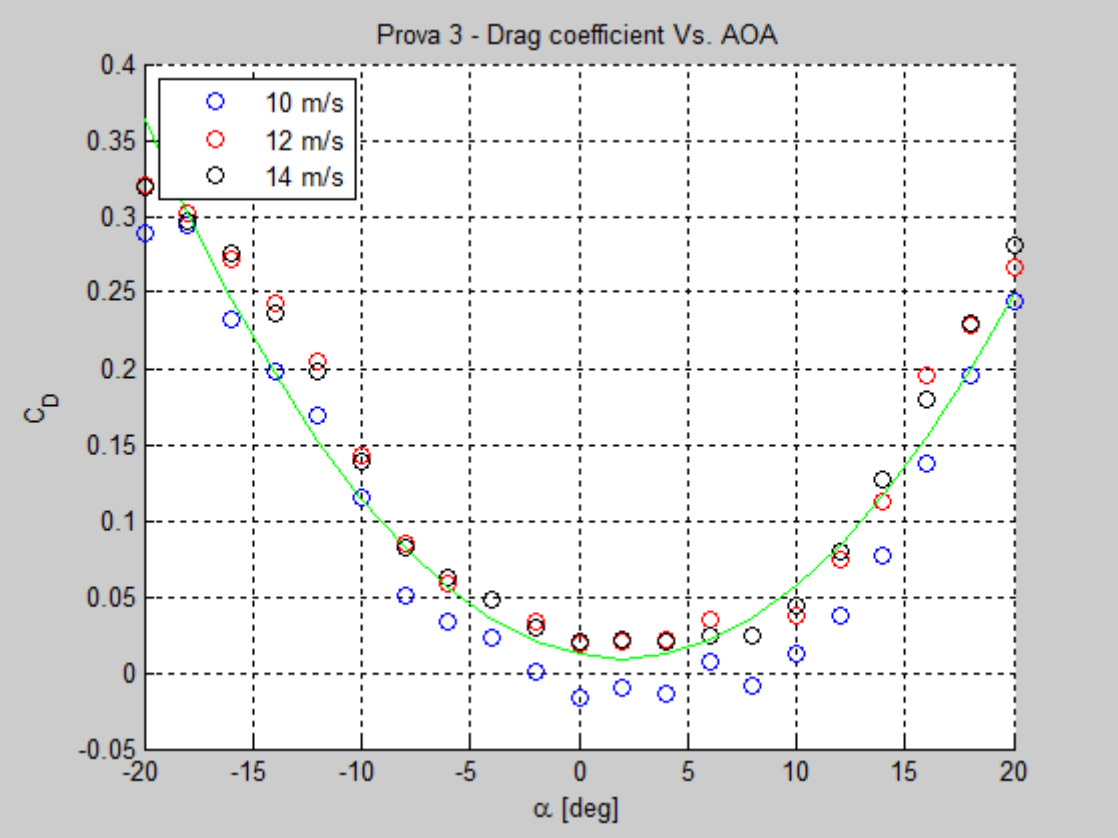


Figure 4.27 – Test 3: Drag coefficient Vs. AOA

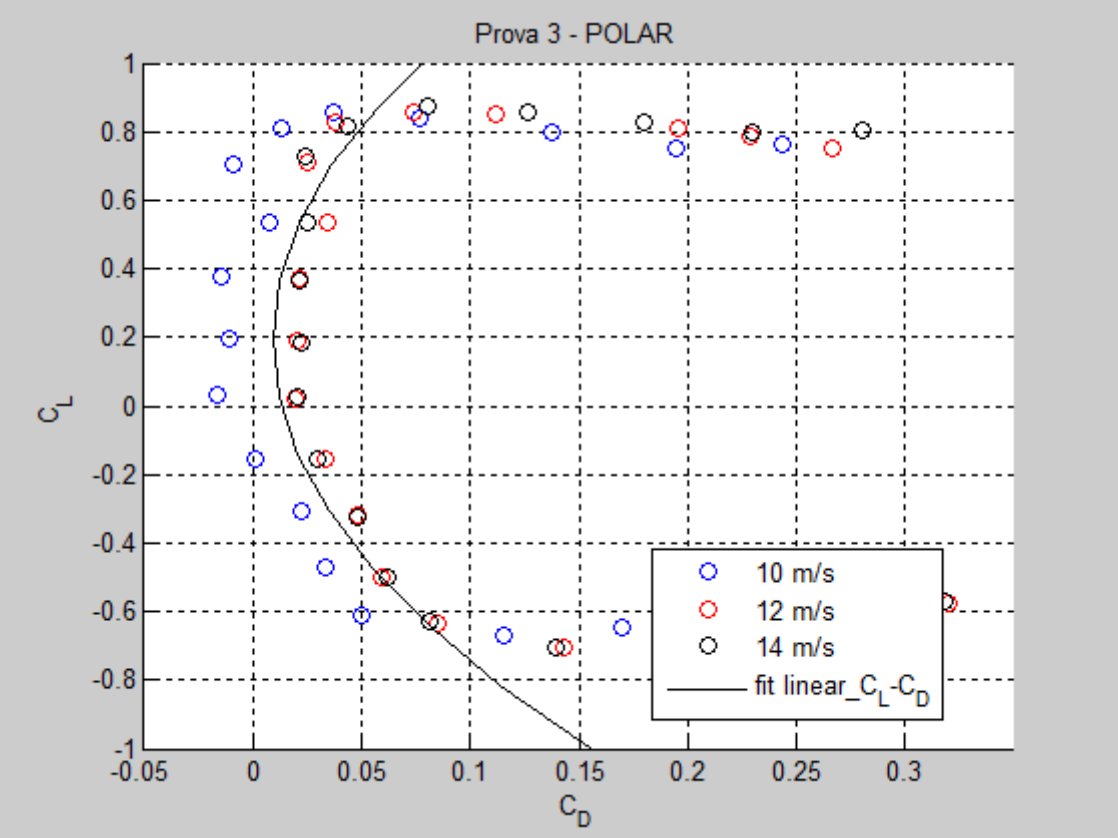


Figure 4.28 – Test 3: Aerodynamic polar

One thing we didn't say before about these results is that for the lift force we reached here its maximum at 14 m/s and 12° AOA right before stall, in spite of Xfoil prediction that stated we would have had a higher lift for the 20 m/s at 4° AOA condition instead in test 1. In fact in these two tests (number 2 and 3) we almost reached 24 N for a single semi-wing while in test number 1 we barely went over 20 N. One more remark coming directly from observing the wind tunnel model in the test chamber is that if in both conditions the wing was visually bending, when at 4° and 20 m/s it was very steady but when over 12° (stall condition) and at all airspeeds tested it was evidently shaking making this last one condition at the higher speed the "worst-case-scenario" for sure with involving inconstant high forces. However the vibration wasn't too strong and the test could continue until the maximum of 20° AOA safely also because after the stall point the lift force started to decrease as expected.

Coming now at the AOA sensor calibration, the recording of the potentiometer output voltage has been performed for all the same test points and at the same time of the lift and drag measurement. Here the results:

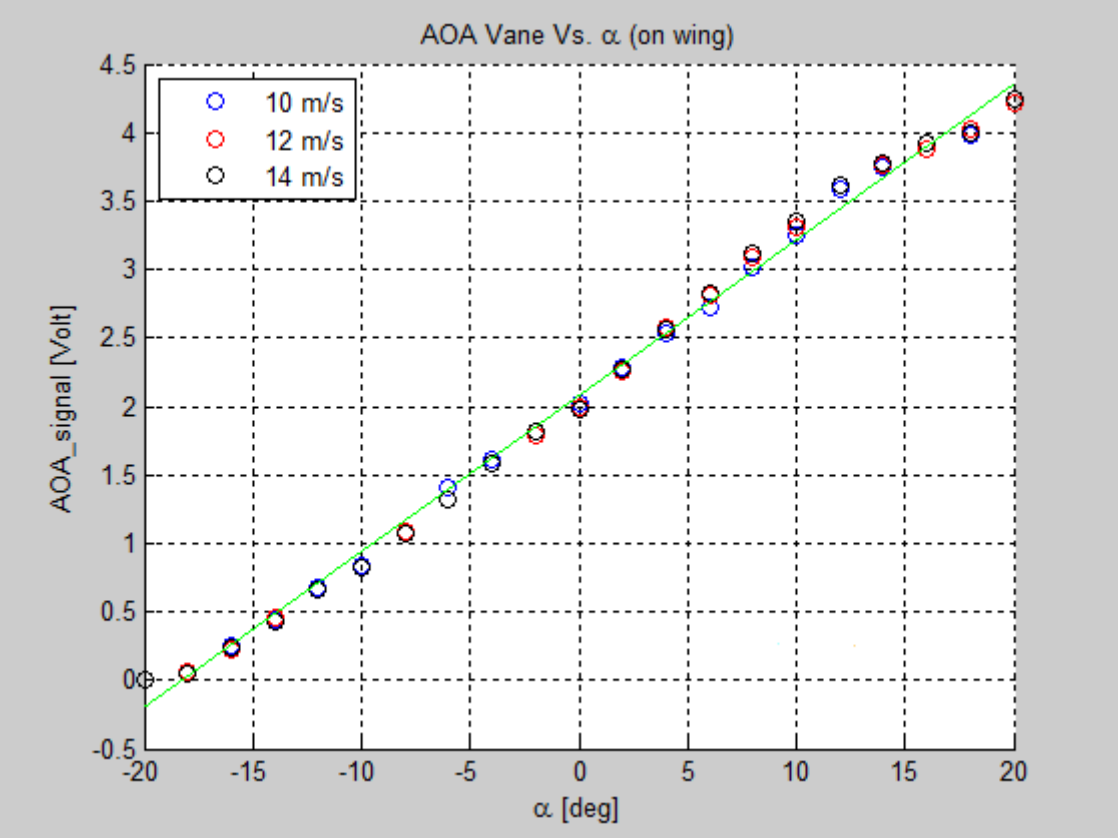


Figure 4.29 – AOA sensor signal Vs. AOA on wing

The above graph shows a good linear relation between AOA and sensor output signal with none or little disturbance confirmed by the fact that for all three runs at different airspeeds, all the data points almost overlap. To get a calibration line that allowed us to determine the angle of attack of the airplane given the sensor electrical output, we just had to invert this function:

Calibration function of vane ALPHA on wing: $y = 0.11373*x + 2.0743$

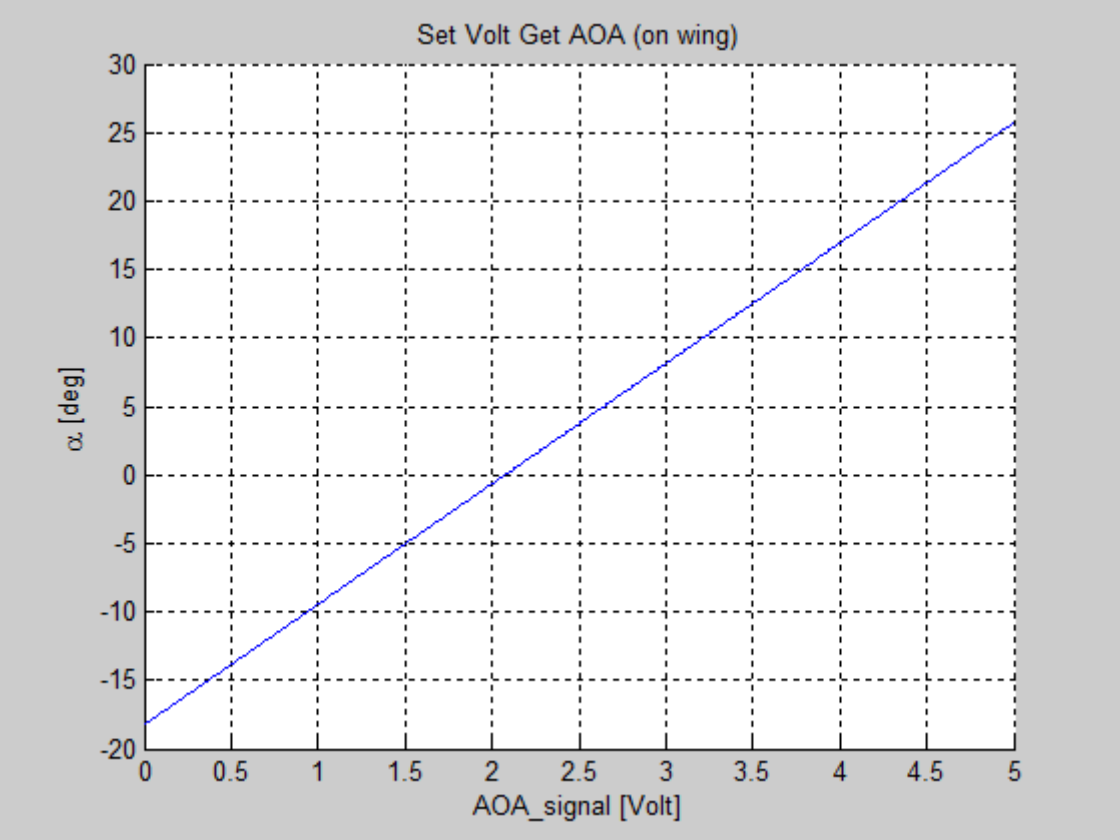


Figure 4.30 – Set-volt-get-AOA function on wing

Set Volt Get Alpha:
alpha = (volt - 2.0743)/0.11373

The above inverted relation is actually the one that will be written in the autopilot code, integrated into the newly written function that manages the acquisition of the analogic signal coming from the sensor, its real time processing and storage into the flash memory.

At the same time also the output signal of the AOS sensor has been recorded and we were hoping for a flat undisturbed constant signal indicating that the AOS vane was actually measuring a zero change of the sideslip angle exactly as in the real conditions but our weak hopes got wiped out by the results in the next figure (number 4.31). The reasons making the wind stream, and so the vane, deviate from the zero sideslip angle by changing the angle of attack, can be multiples. Surely one of these could be the mutual interaction of the two vanes that are mounted perpendicular to each other but still relatively close. Then other factors could be the bending inward or outward of the air stream getting closer to the upper and lower wing surface where the local pressure changes varying the angle of attack (see figure 4.32), the neglecting in the wing positioning on the support of the dihedral angle (that we didn't neglected in the installation of the sensors on the booms (see section 2.3.2)), the vibration of the wing especially at high angles of attack. Actually if we observe the graph carefully we can see a somehow linear trend of the data points for the 10 and 12 m/s runs. However the overall variation is quite small (0.6 V compared to the 5 V range of the potentiometer) and the linearity of the AOS signal as a function of the AOA variation is not very well defined indeed.

Also, we don't have any data to correlate the change in AOA as a function of the AOS variation on wing and so, as the AOA sensor calibration has been based only on the variation of the angle of attack, we also preferred to relate the calibration of the AOS sensor only to the variation of sideslip angle even if this test, that will be illustrated into the next section, had to be performed off wing.

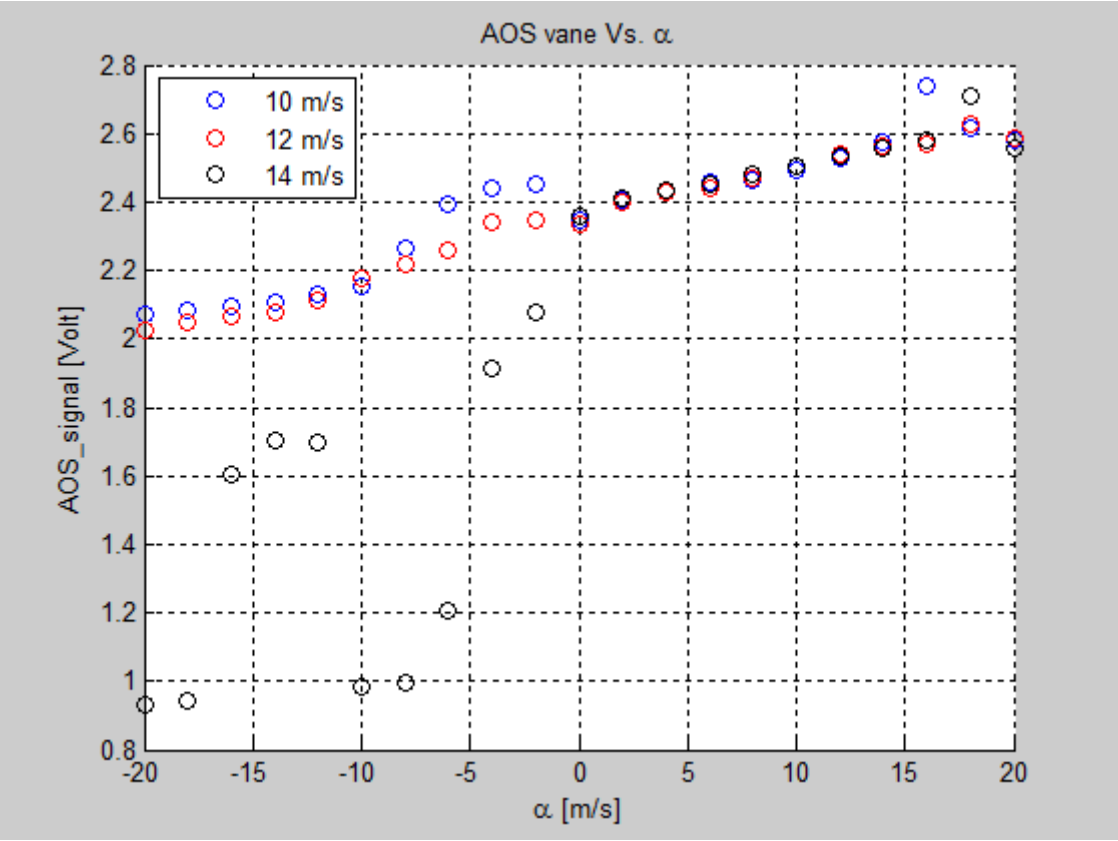


Figure 4.31 – AOS variation changing AOA on wing

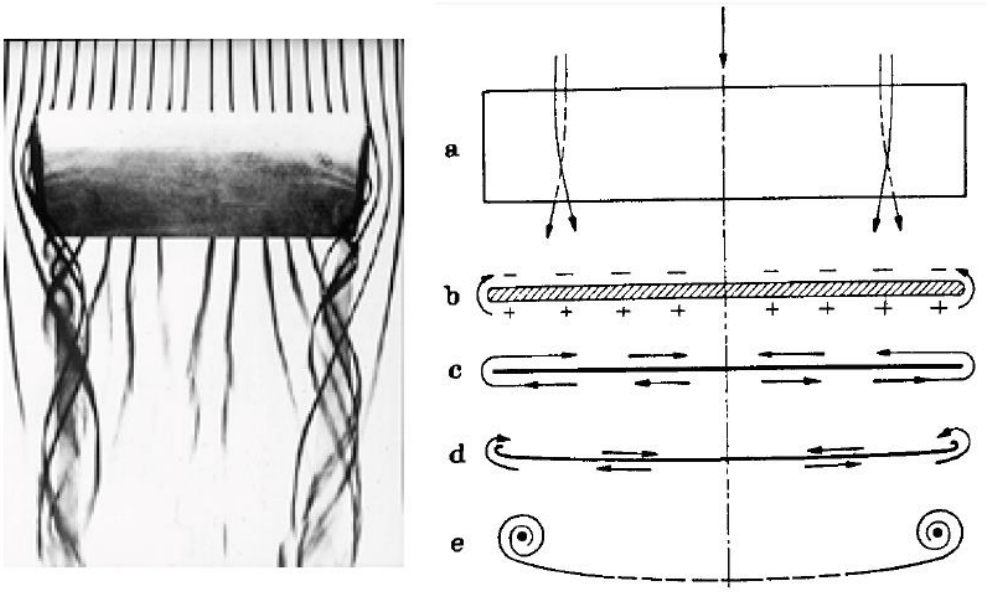


Figure 4.32 – Bending of wind stream due to the different pressure between the upper and lower wing surfaces

Figure 4.33– Particular of the wind tunnel model with the AOA and AOS sensors installed

Observation: in the picture above taken during test 3, it can be clearly seen how the AOA vane is following the wind stream direction at a high negative angle of attack. At the same time it can be noted that the AOS vane is not parallel to the direction of the boom, as expected, but its slightly pointing downwards. Because we are in a situation of highly negative α, surely there is higher pressure on the upper surface of the wing and lower pressure on the lower surface (the one facing the camera where the boom support is attached). From aerodynamic theory and referring to figure 4.32, we know that the wind stream is bended inward towards the wing root on the low pressure surface and outward towards the wing tip on the high pressure surface. Now, because the empennage of the AOS vane is closer to the leading edge upper surface, this last one might be our case with the sensor following the deviating air stream outward giving strength to one of the previous hypothetical causes affecting the measure stated above.

4.3.4 Test 4: AOA and AOS sensors calibration *off wing*

This test was done primarily to also have a calibration of the AOS sensor in function of the sideslip angle because, as we said before, the wing couldn't be fit horizontally inside the test chamber and another support allowing to swing the model in vertical position to precisely change sideslip angle couldn't be manufactured. So another very simple support was adopted, consisting in a upside-down L-shaped rod where to attach the original support with the boom and the vanes that can be oriented so with the same rotating support we could simulate either a change in angle of attack or in angle of sideslip:

Figure 4.34 – AOA and AOS sensors off wing

With this system both the tests for AOS and AOA were performed by simply rotating the boom by 90°. The resulting calibration function for the angle of sideslip sensor will be included into the autopilot software for the actual in flight measurements. As for the off wing angle of attack sensor resulting function it will only be compared, in the next section, with the previous on-wing calibration to see how big is the difference but only the "on-wing" function (most accurate) will be kept for inclusion in the APM code as declared before.

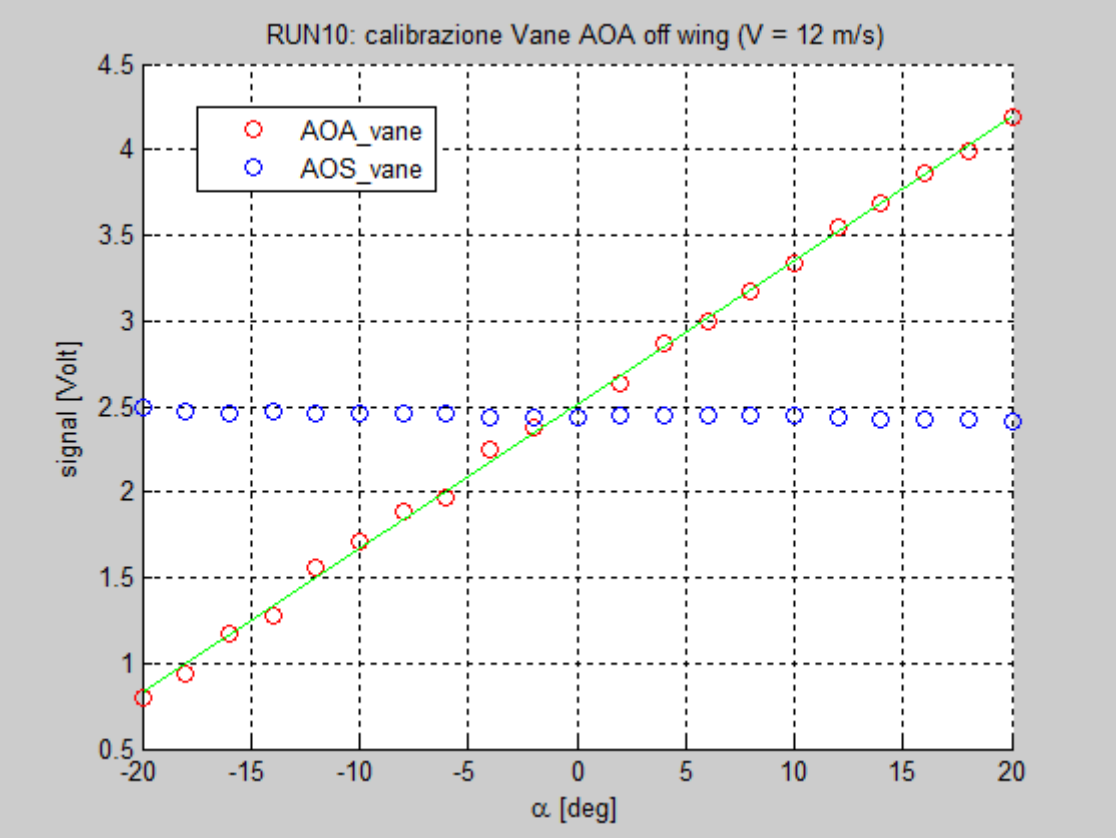


Figure 4.35 – AOA sensor calibration off wing

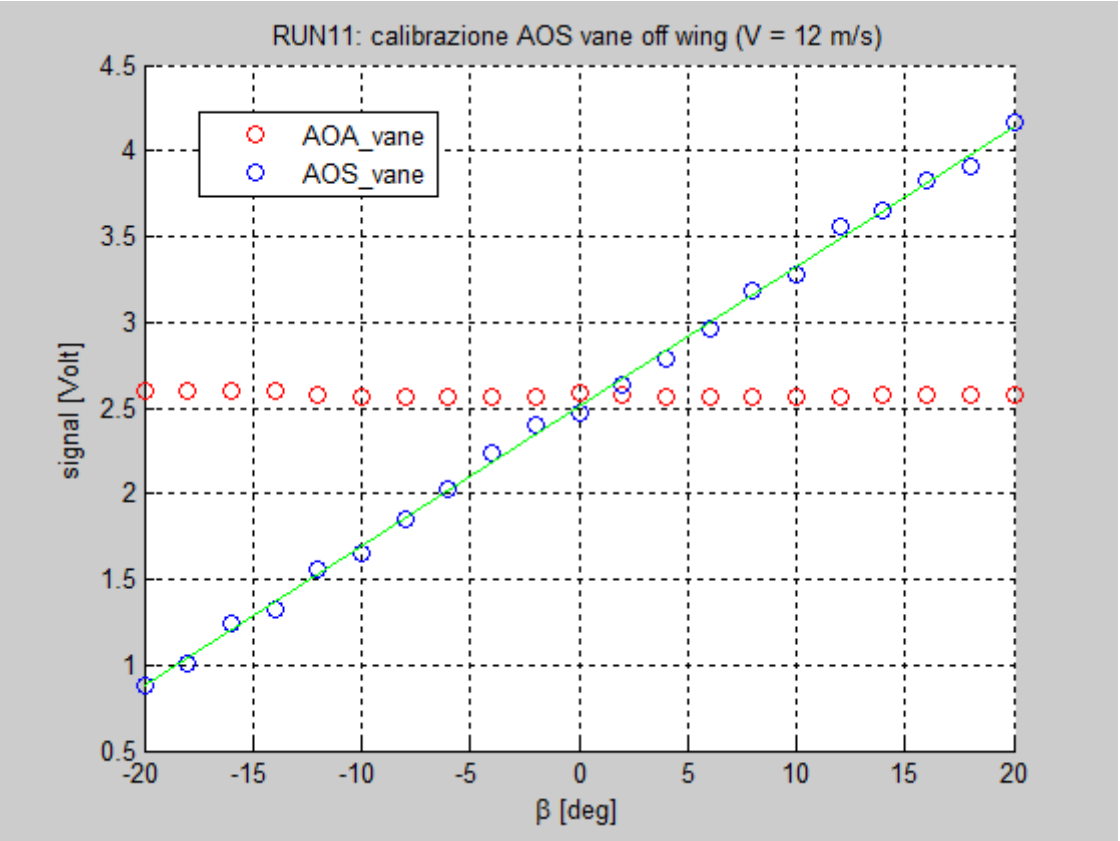


Figure 4.36 – AOS sensor calibration off wing

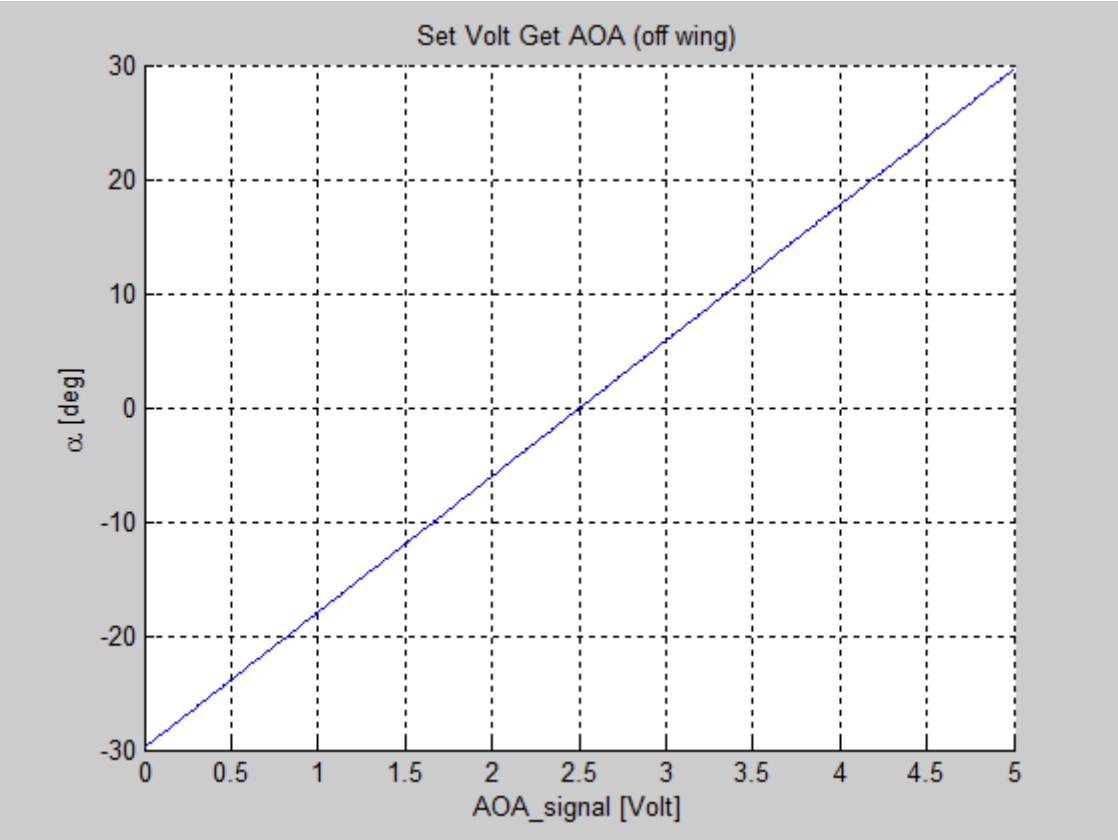


Figure 4.37 – Set-volt-get-AOA function off wing

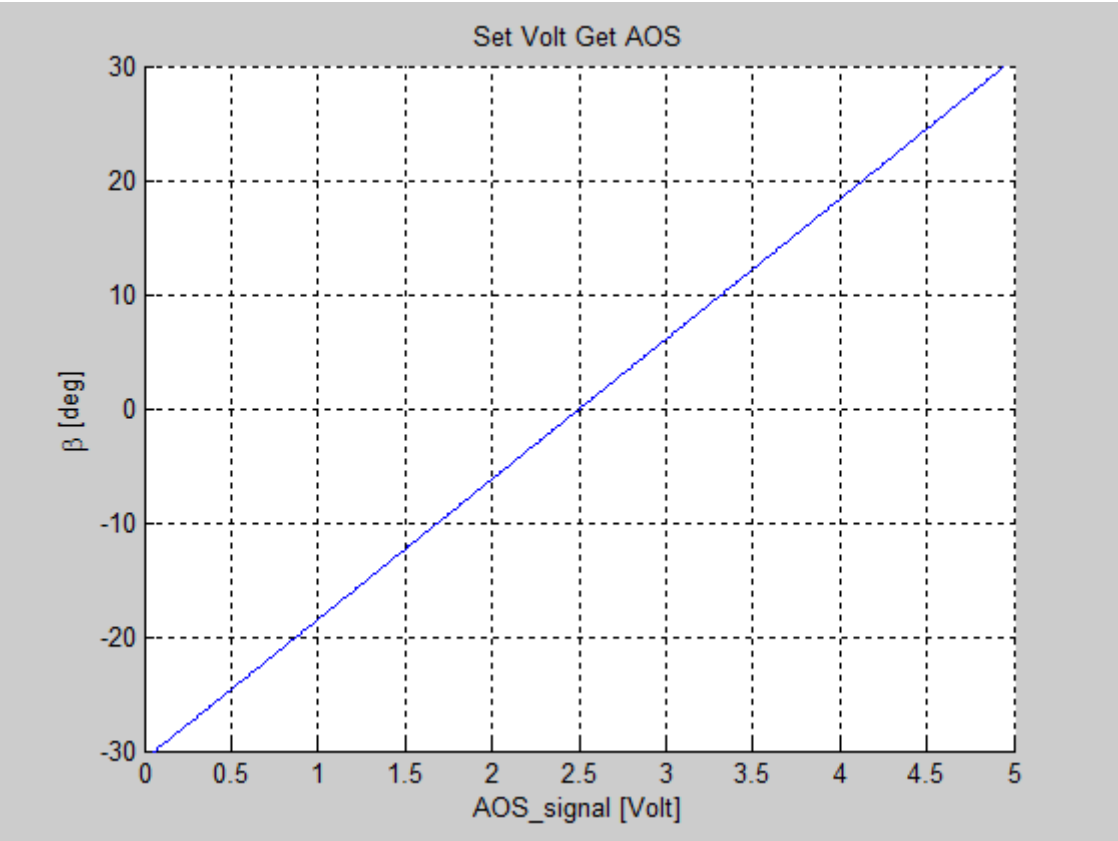


Figure 4.38 – Set-volt-get-AOS function off wing

Following the numerical results of the two tests together with the air data:

PROVA 4A
Density: 1.1968 [Kg/m^3]
Absolute Pressure: 100431.3411 [Pa]
Temperature: 18.7082 [C°]
Relative Humidity: 19.8936 [%]

Retta di calibrazione Vane ALPHA off wing:
y = 0.084096*x + 2.5078
Set Volt Get AOA:
alpha = (volt-2.5078)/0.084096 + i (i = 3° wing relative pitch angle)

Stable Vane Beta signal --> ok

PROVA 4B
Density: 1.1973 [Kg/m^3]
Absolute Pressure: 100466.5918 [Pa]
Temperature: 18.7069 [C°]
Relative Humidity: 19.9377 [%]

Retta di calibrazione Vane AOS off wing:
y = 0.081471*x + 2.5055
Set Volt Get AOS:
beta = (volt-2.5055)/0.081471

Stable AOA vane signal --> ok

Two important considerations can now be done. The first one is that unlike the case of the sensors installed on wing, while changing angle of attack now the AOS vane is not affected at all and in the same way when changing angle of sideslip neither the AOA vane is affected. This fact confirms that the proximity of the wing greatly affects the measures. The second observation is about the difference in starting position of the sensors. In fact at 0° position of the rotating platform now the sensors are also at 0° whereas when mounted on the wing at 0° position of the platform and of the wing chord line direction, the boom direction and so the AOA sensor was starting at -3° inclination accounting for the wing pitch angle in relation to the fuselage line as in the real case positioning. So to compare the AOA resulting calibrations we had to be careful to account for this difference. The solution is to simply subtract 3° to the on-wing AOA vector so the two tests will have the same starting position reference.

4.3.5 Comparison among tests

Starting with the on/off wing calibration, in this section will be presented the comparisons between tests of this same wind tunnel campaign and then how these results compare to the Xfoil simulation for lift and drag coefficient.

For the AOA sensor calibration we wanted to be precise because the angle of attack is a very important value always needed for all the flight tests performed at the field. The reason why we calibrated this sensor mounted on the original wing, is the idea to account, as precisely as possible for the conditions recreated in the wind tunnel rather than in flight, for all those uncertainties due to the vane specific position. In the following comparative graphs it will be clear that these contributes (e.g. flexure/torsion coupling of the spars, wing up-wash) are not negligible at all.

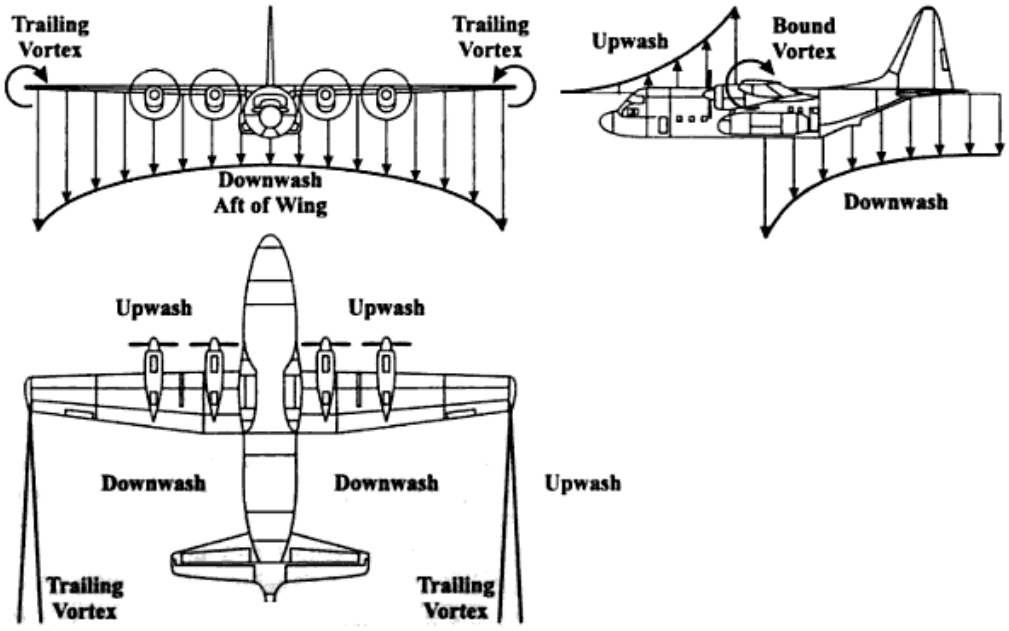


Figure 4.39 – Illustration of the up/down wash distribution and trailing vortexes for a conventional airplane

In front of the wing the undisturbed asymptotic airstream deviates upward while passed the wing is pushed downward by the bound vortex generated by the wing circulation. So in front of the wing the relative wind direction is seen at a higher angle of attack when coming close to the leading edge. This is in accordance with what the following graph shows with a higher sensor-output vs. α for the on-wing calibration than for the off wing one. In fact the AOA sensor is mounted right in front of the leading edge and at the same geometrically set angle of attack, senses a bigger variation in the airstream direction due to the wing up-wash resulting in a higher voltage output, than when is mounted on the off-wing support. In the same way also a small deformation of the wing shape under stress can result in a variation of the signal.

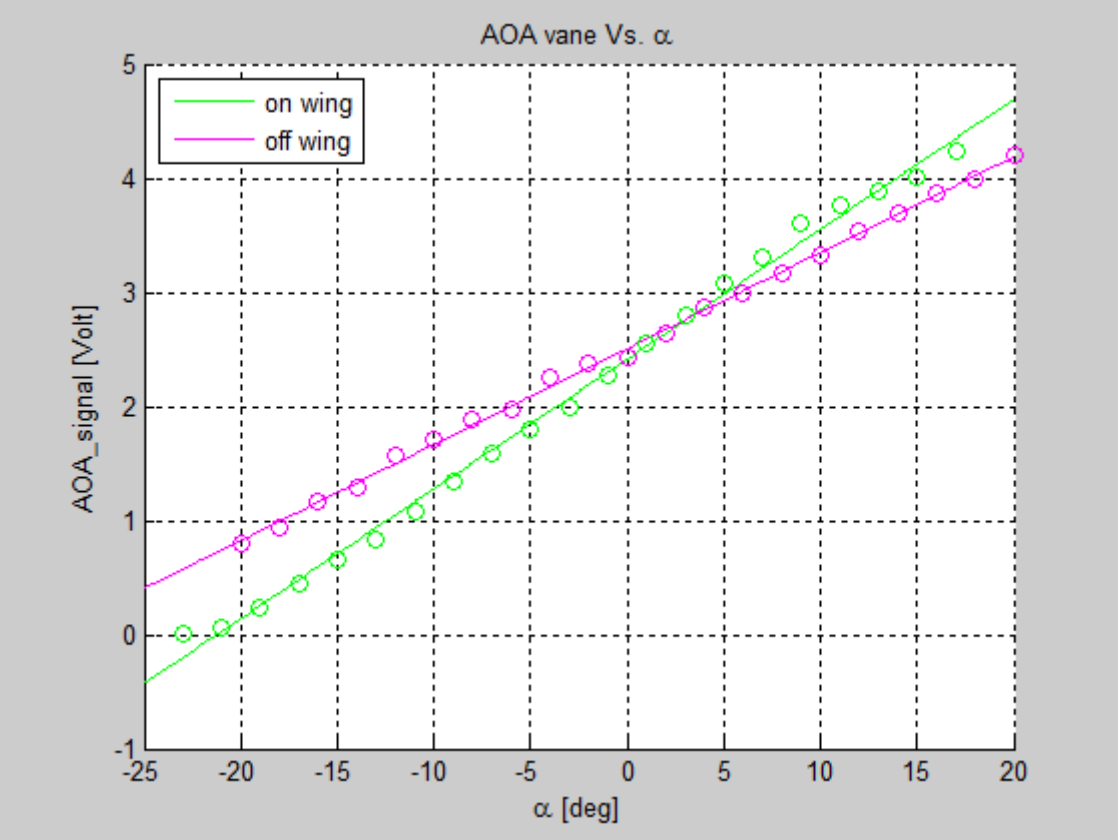


Figure 4.40 – AOA sensor calibration on/off wing comparison

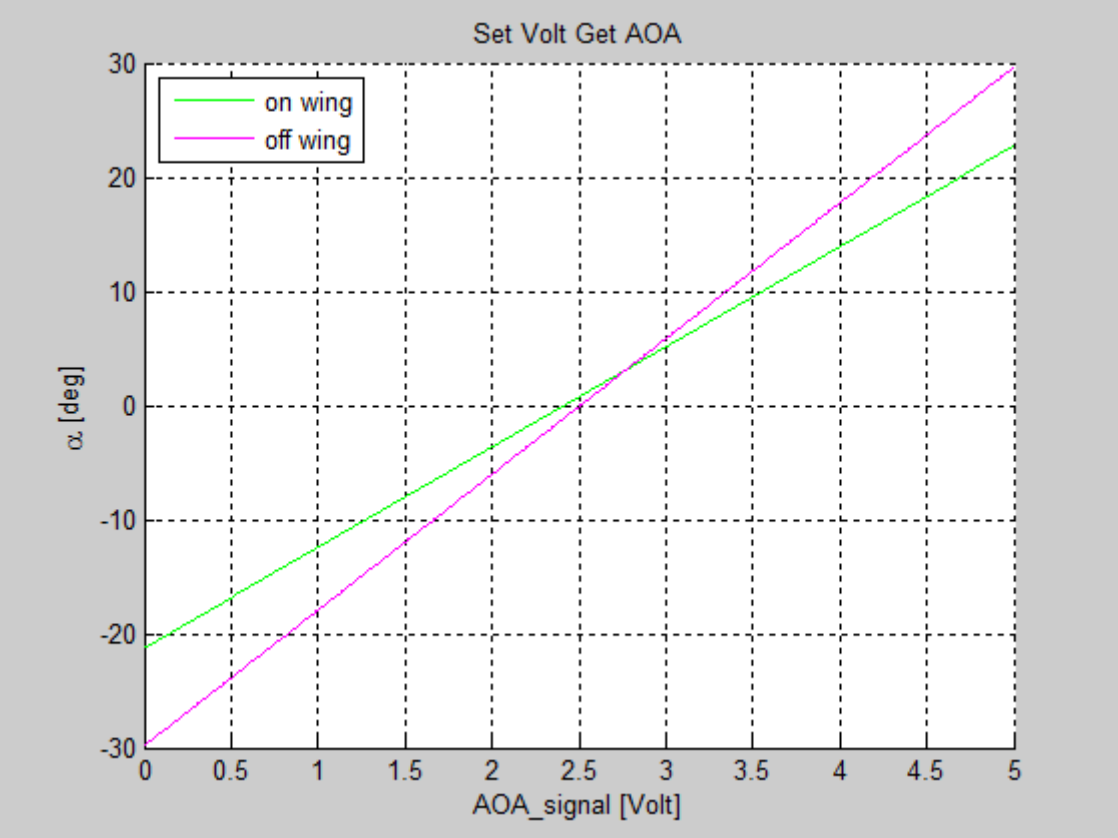


Figure 4.41 – Set-volt-get-AOA function on/off wing comparison

In tests 2 and 3 we recorded with the balance the results for the same test points creating different graphs and in the following three figures is shown the comparison between these results. As expected, because the lift measure has been very accurate and repeatable, the lines of the linear fit for the linear trend portion of the coefficient of lift curves present very little bias. The drag measure instead, as observed and explained before, was more difficult and less repeatable due to different kinds of disturbance. However, as it can be observed in figure 4.43, even if the data points are quite scattered, the resulting lines from the quadratic polynomial functions interpolating them are quite close and taking the mean between the two, the result can be considered quite accurate as well. Same thing of course for the two aerodynamic polar parabolas which are directly depending on the accuracy of the first two types of curves. The analytic results for the interpolating curves reported below give us the lift coefficient value at zero angle of attack (~ 0.0238), its linear variation (~ 0.0856) and the minimum drag coefficient value (~ 0.0196) which is obtained for an angle of attack of approximately 2°.

C_L-α linear mean coefficients: $y = 0.085584\ x + 0.023761$

C_D-α squared mean coefficients: $y = 0.00073141\ x^2 + (-0.0025726)\ x + 0.019602$

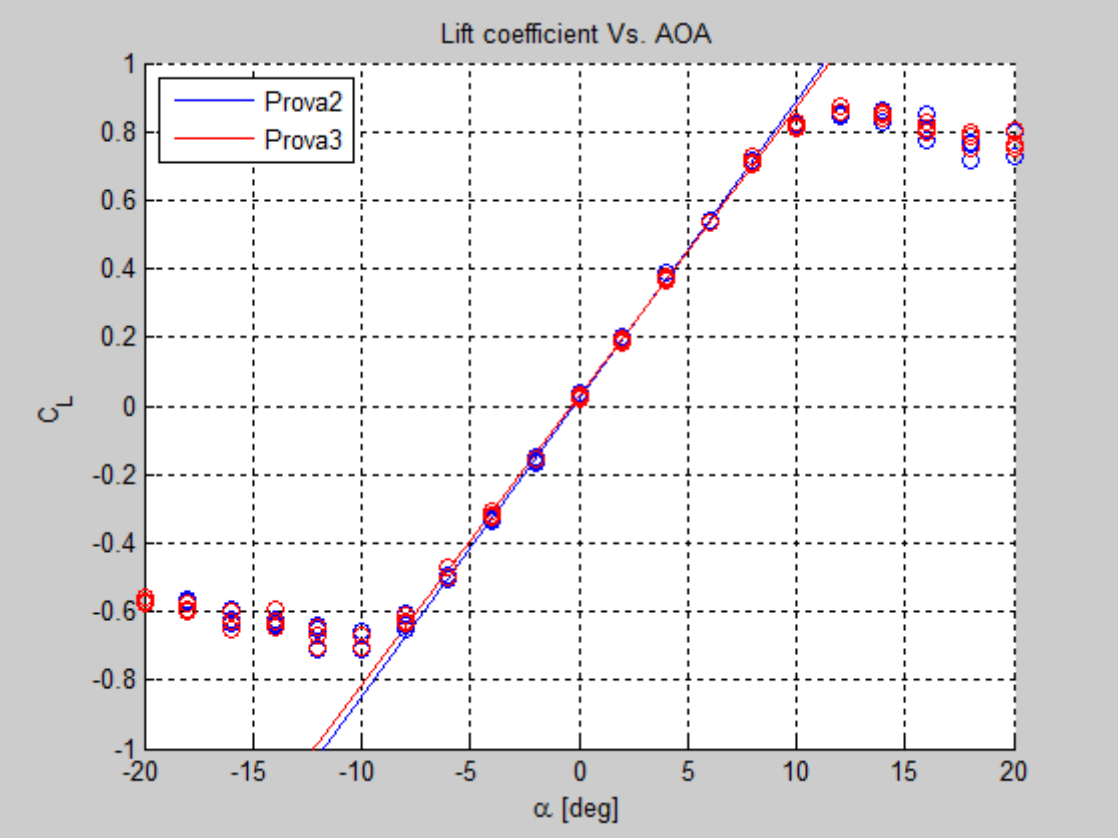


Figure 4.42 – Wind tunnel lift coefficient curves comparison

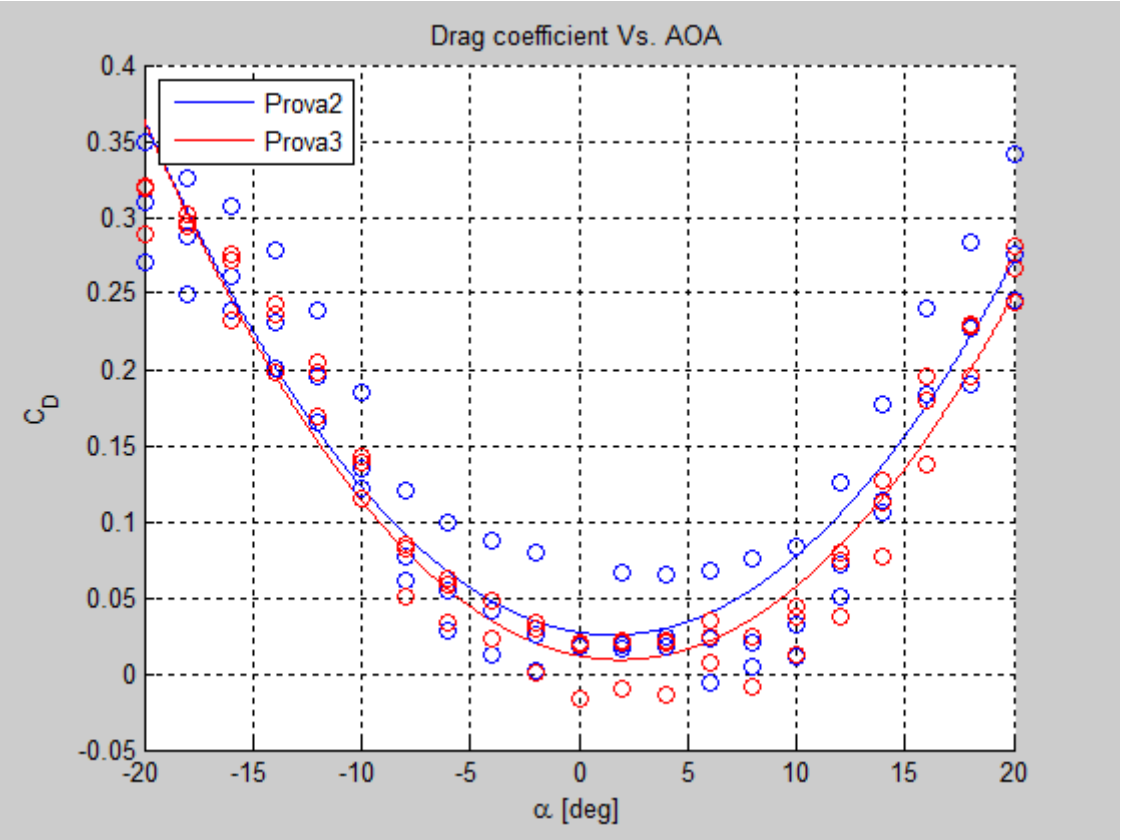


Figure 4.43 – Wind tunnel drag coefficient curves comparison

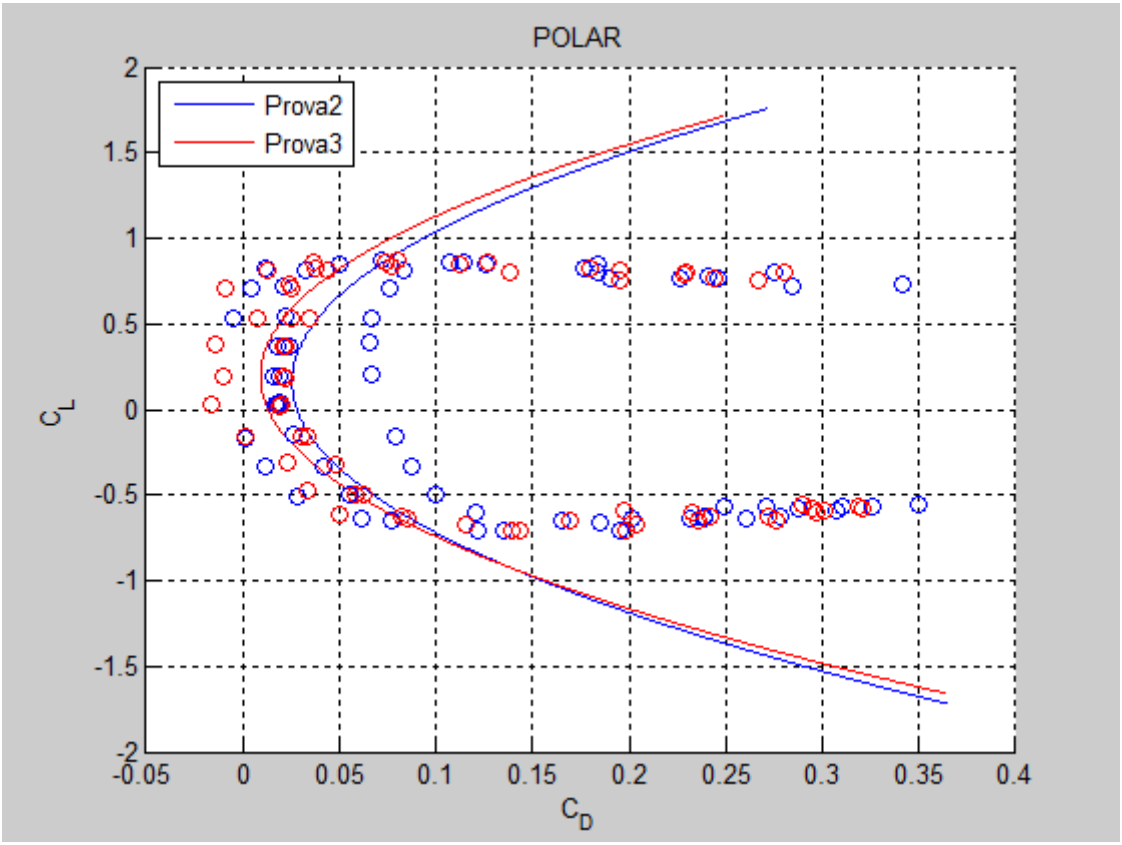


Figure 4.44 – Wind tunnel aerodynamic polars comparison

A first comparison among experimental and computational results can be done at this point plotting together the wind tunnel data along with the Xfoil simulation.

As it can be seen in the graphs below, the Xfoil coefficient of lift estimation is greater than the actual result coming from the wind tunnel while the computed drag is lower, all as expected, resulting in a better polar in terms of aerodynamic efficiency for the Xfoil simulation than for the real wind tunnel experiment.

- The lift coefficient values coming from Xfoil are higher than the wind tunnel test results because an airfoil with theoretically infinite wingspan has, by definition, no trailing vortexes resulting in a loss of lift as it happens instead for a real finite wing.

- The Xfoil drag coefficient values are significantly lower instead because the contribution coming from the induced drag is not included for the same reason as before; an airfoil with theoretically infinite airfoil has no trailing vortexes like a real finite wing which are responsible of the aerodynamic induced drag.

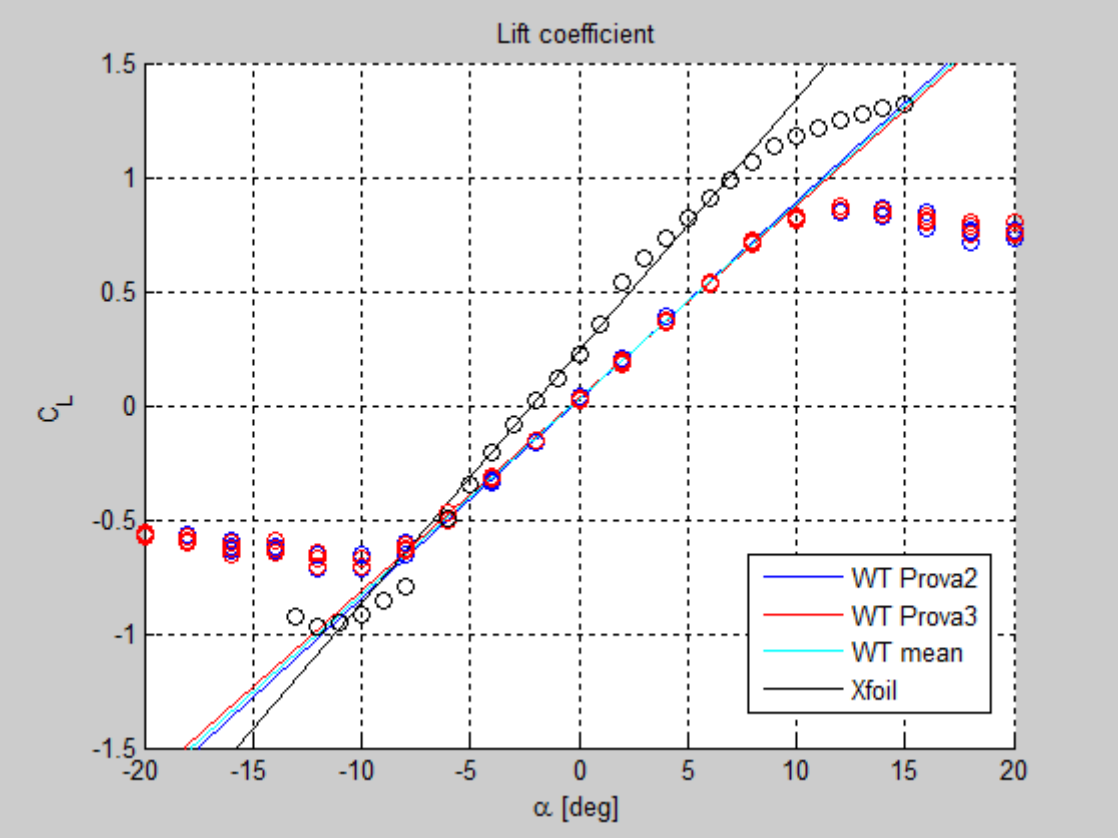


Figure 4.45 – Wind tunnel and Xfoil coefficient of lift comparison

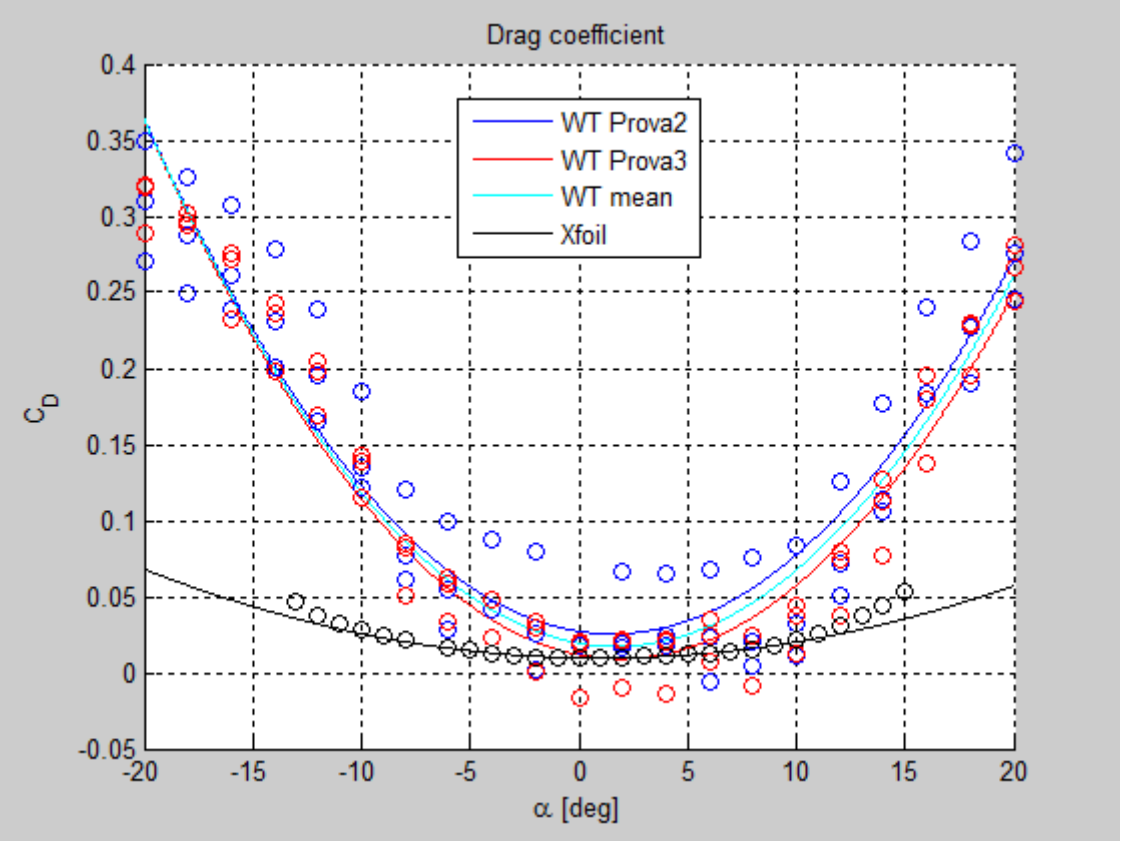


Figure 4.46 – Wind tunnel and Xfoil coefficient of drag comparison

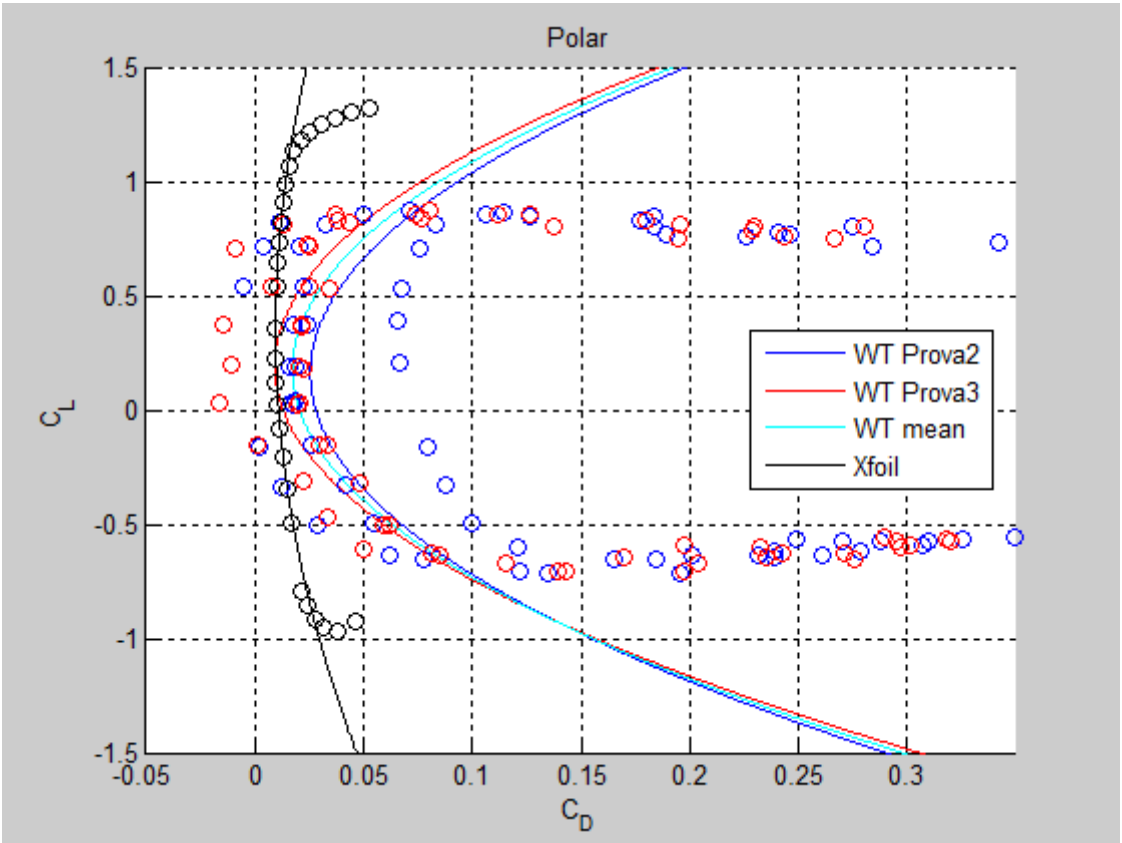


Figure 4.47 – Wind tunnel and Xfoil aerodynamic polar comparison

4.3.6 Propeller traction test

As briefly stated in the test description section, another important test has been performed in the previous thesis work[10] on the same flying platform to analytically predict the traction of the propeller at any instant varying the motor throttle % and wind tunnel airspeed (parameters also measurable in flight) and recording at every test point RPM value of the spinner and traction force of the whole power unit. These results are very important for the "speed-drag-method" to calculate the drag of the airplane during flight and be able to create the complete aerodynamic polar starting from the data acquired during multiple trimmed flight at different speeds. The full description of the method, however will be presented in section 6.5.1. Here we only report the results of the wind tunnel traction test that are useful to our work.

Each run consisted in varying at a fixed airspeed the throttle % of the motor from 20% to 100% with a 10% increment. This had to be repeated at different airspeed, firstly at 0 m/s to see the fixed point traction and then starting from 8 m/s and going up to 26 m/s with increments of 2 m/s. The load cell supporting the back of the motor measures the traction of the propeller while a microphone close to the spinner gives the information about the number of rotations per minute (RPM). See figure 4.7 for the setup.

Following an example of the typical graphs resulting from one of this runs, specifically the run at 12 m/s:

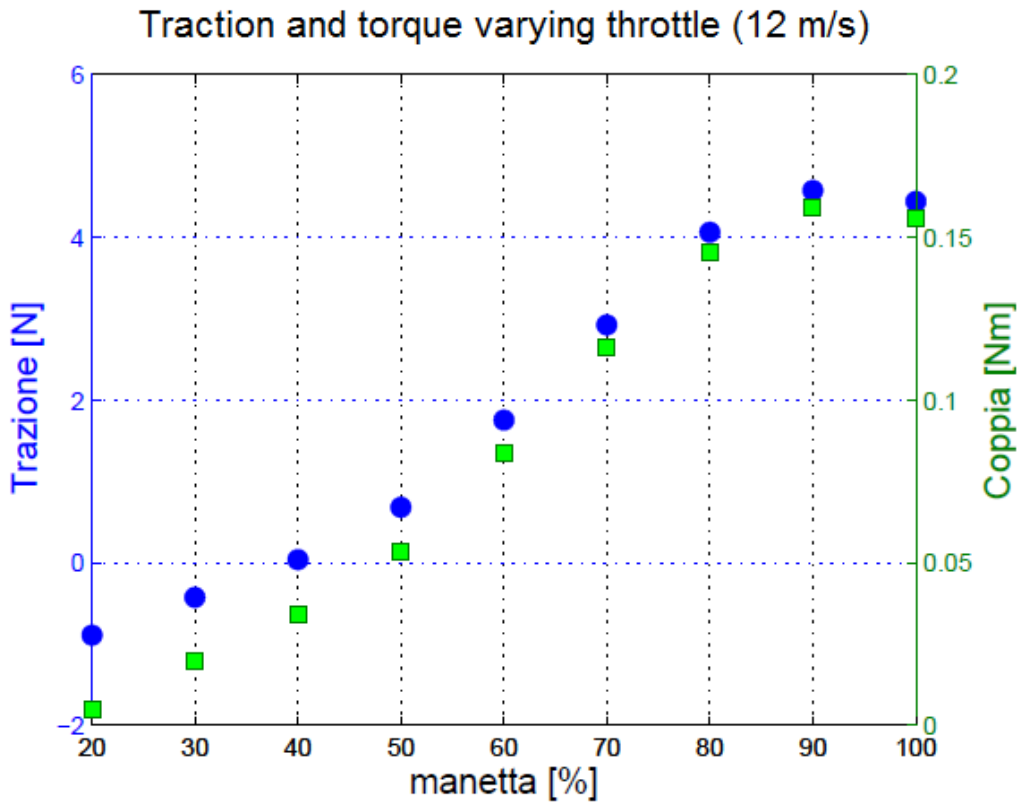


Figure 4.48 – Traction result, example run

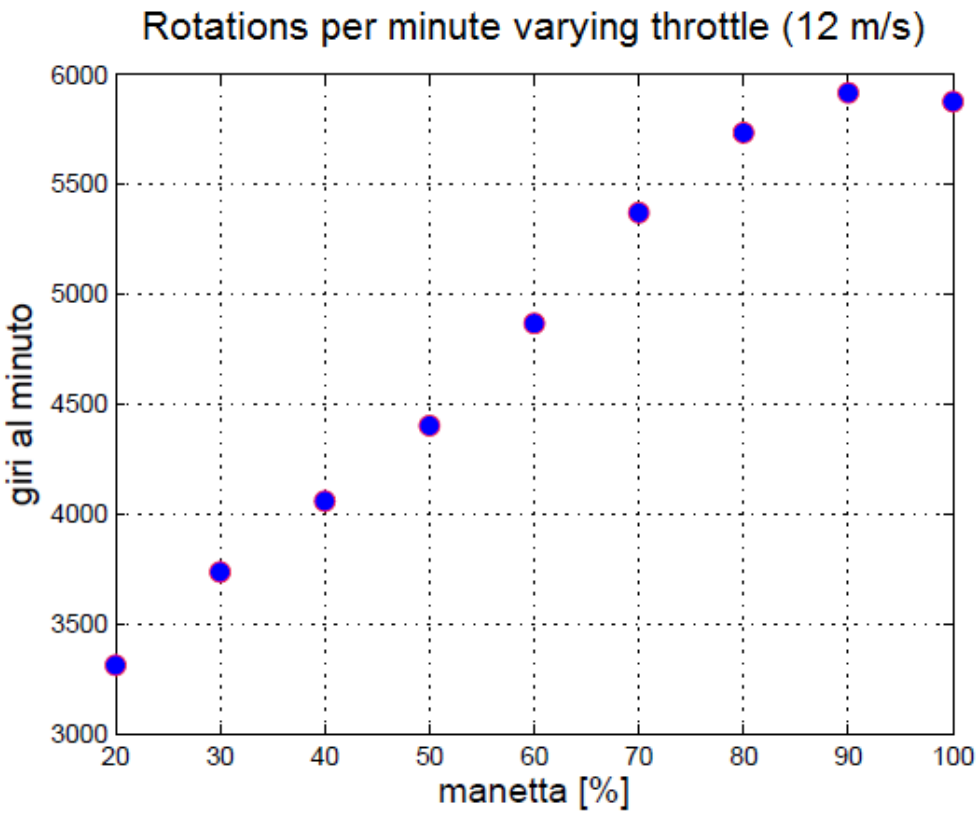


Figure 4.49 – Rotations per minute result, example run

When having the results for all the test points, it has been possible to calculate the coefficient of traction using the *Rénard* formula:

$$C_T = \frac{\mathrm{T}}{\rho\, n^2 d^4}$$

Where T is the measured traction force, ρ is the air density, the parameter n is the rotations per minute of the motor and the parameter d is the diameter of the propeller.

To have both indications ready at hand for the rotations per minute (RPM) and coefficient of traction (C_T) by simply imposing a specific value of throttle % and airspeed, which also are values directly measurable in-flight, a superficial fitting has been creating using the Matlab curve fitting tool *CFtool*[8].

$$RPM = f\,(\,thr\,\%\,,\,V)$$

$$C_T = f\,(\,thr\,\%\,,\,V)$$

The plots of the interpolant polynomials functions are reported in the two figures below:

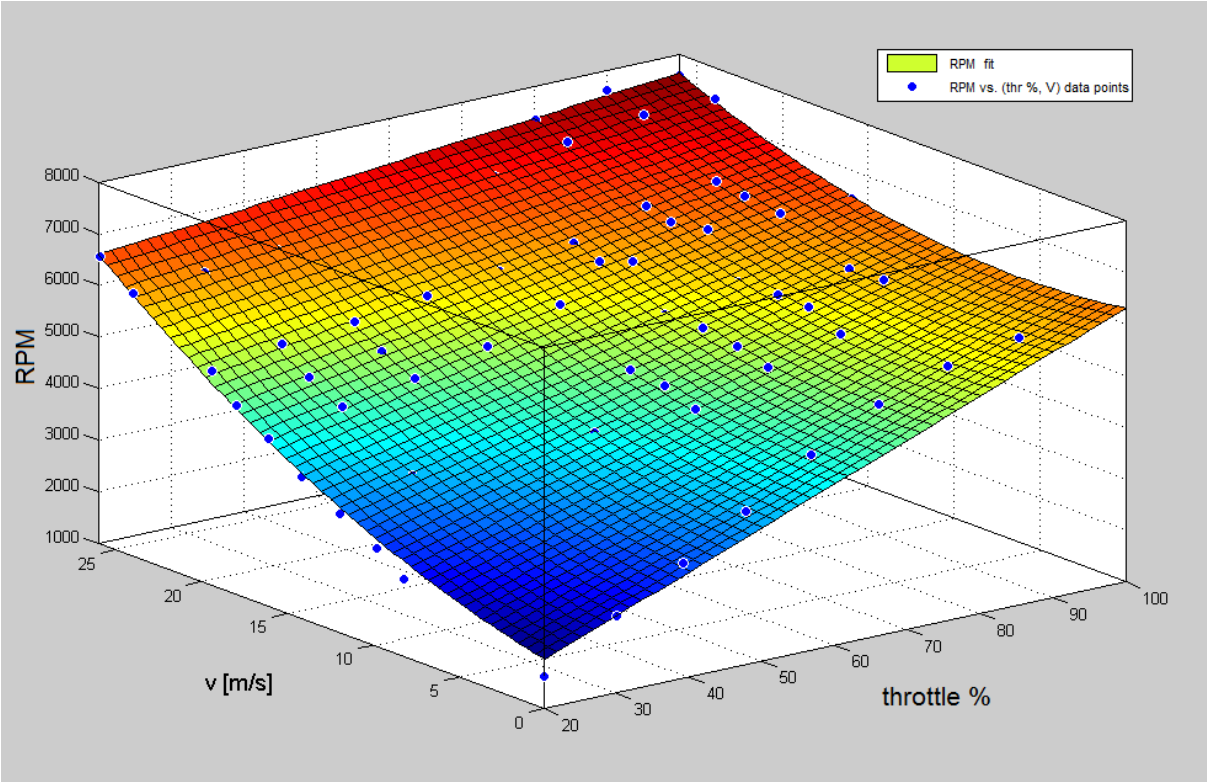


Figure 4.50 – RPM surface fitting

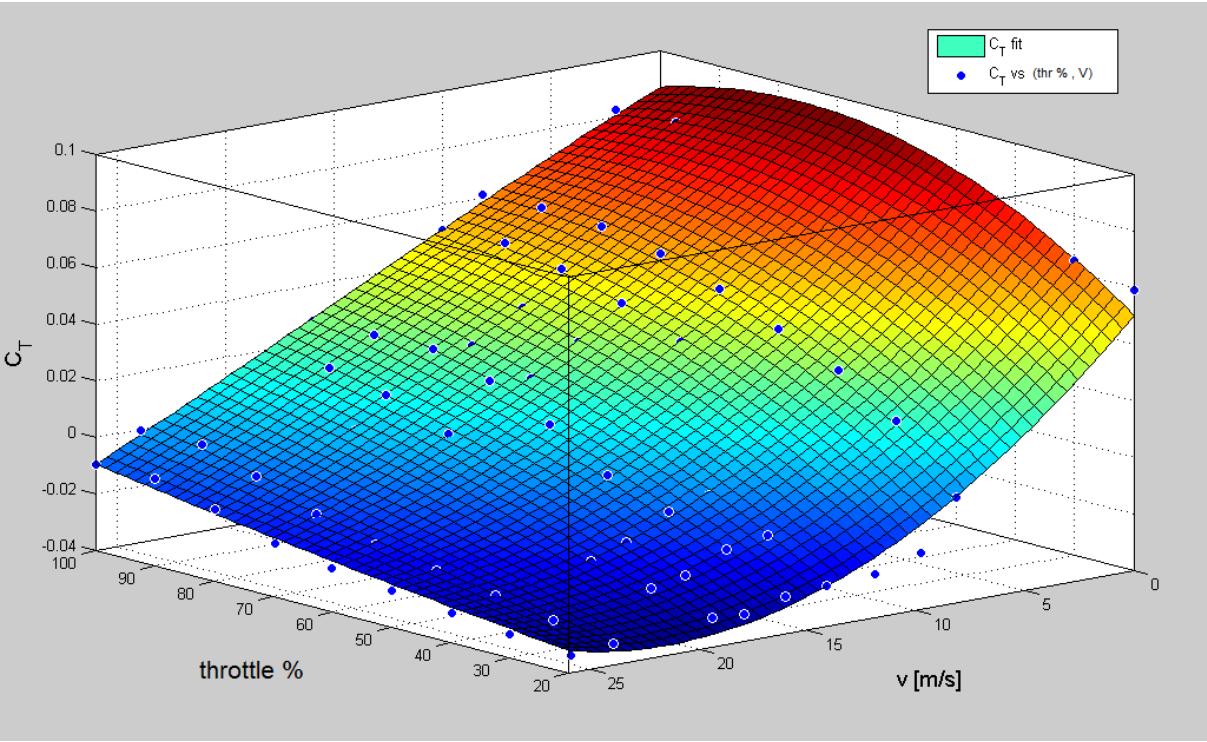


Figure 4.51 – C_T surface fitting

Following the statistic parameters indicating the goodness of the fitting:

	RPM	C_T
R^2	0.9733	0.9860
adj R^2	0.9720	0.9846

Table 4.2 – Surface fitting statistic parameters

Where R^2 is the coefficient of determination which is proportional to the variability of the analyzed data compared to the statistic model and can vary from a minimum of 0 to a maximum of 1. The *adj* R^2 is simply a variant of R^2 and is used for the regression analysis of multiple variables functions.

Also Matlab gives as an output of the fitting the analytical polynomial functions of the curves that both have the following form:

$$\mathrm{z} = p00 + \mathrm{p10} \cdot \mathrm{m} + \mathrm{p01} \cdot \mathrm{v} + \mathrm{p11} \cdot \mathrm{m} \cdot \mathrm{v} + p20 \cdot m^2 + p02 \cdot v^2 + p21 \cdot m^2 \cdot v + p12 \cdot m \cdot v^2 + p03 \cdot v^3$$

Where m is the throttle %, v is the airspeed and z the interpolant surface while the *pXX* coefficients are listed in the table below for each function.

	RPM	C_T
p00	881.5	0.01995
p10	54.26	0.002137
p01	106.4	-0.007668
p11	-1.576	-1.347e-05
p20	0	-1.392e-05
p02	3.968	-0.0002035
p21	0	5.658e-07
p12	0	-2.369e-06
p03	0	7.387e-07

Table 4.3 – Surface fitting polynomial coefficients

4.3.7 Temperature effect

The traction tests were performed at the end of June 2013 with a mean temperature inside the test chamber of 30°C. Because our flight tests were performed in the spring of 2015 at lower temperatures between 20 and 25°C, the possible changes in results due to difference in temperature for the worst case of 10°C are here discussed.

The aerodynamic forces acting on the propeller blades are influenced by the local Mach and Reynolds numbers. Because the Mach number is a function of the speed of sound which is itself a function of the temperature, we here calculate the percentage variation of the local Mach number for a change in temperature of 10°C.

Mach number: $M = \frac{V}{C}$

Speed of sound: $c = \sqrt{\gamma \cdot R \cdot T} = \sqrt{\gamma \cdot R \cdot (\vartheta + 273.15)}$

Where γ = 1.4 is the ideal diatomic gas value, R = 8.314510 J/(mol K) is the molar gas constant, T the temperature in Kelvin and θ the temperature in degrees Celsius.

So the variation of the Mach number for the same airspeed seen by the spinning propeller can be computed as follows:

$$\frac{M_{|20°C}}{M_{|30°C}} = \frac{\frac{V_{tip}}{C_{|20°C}}}{\frac{V_{tip}}{C_{|30°C}}} = \frac{C_{|30°C}}{C_{|20°C}} = \frac{\sqrt{\gamma \cdot R \cdot (30 + 273.15)}}{\sqrt{\gamma \cdot R \cdot (20 + 273.15)}} = 1.0169$$

This result shows a Mach number variation of only the 1.69 % due to a variation in temperature of 10°C and it can be neglected in comparison to the size of the inherent errors in the experimental measures.

Talking now about the Reynolds number, the temperature determines a variation of both the air density (also function of pressure) and the air viscosity:

$$Re = \frac{\rho(T,P)VL}{\mu(T)}$$

We then evaluated the change in Reynolds due to a variation of temperature at different pressures. In the figure below it can be observed that for a variation of 10°C, for a mean pressure of 1000 hPa, we have a variation of Reynolds of about 6%.

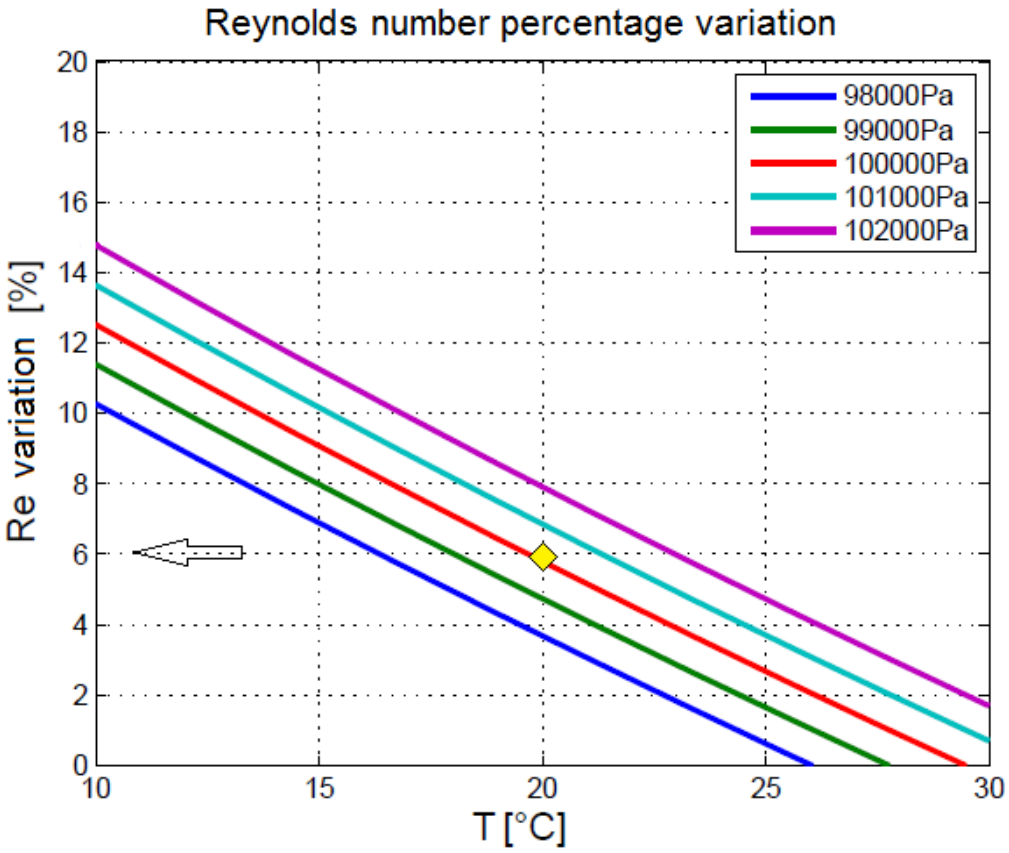


Figure 4.52 – Reynolds variation due to a difference in temperature

To evaluate the effects of the Reynolds number variation on the propeller coefficient of traction, here we base our data on the previous thesis results. These results come from a bibliography research on the propellers performances. A study conducted at the Oklahoma State University[9], presents the results of numerous traction tests in the wind tunnel, performed on propellers of reduced dimensions at low Reynolds numbers. Starting from these data, it has been possible to find a relation between the Reynolds variation and the percentage increment of the C_T coefficient.

As it can be seen from the graph in the next page, considering a Reynolds variation of 6 % as founded before consequently to a temperature difference of 10°C, the change in coefficient of traction for the spinner is about 5 %. Also in this case, this variation can be considered negligible compared to the uncertainty coming from the measurements errors.

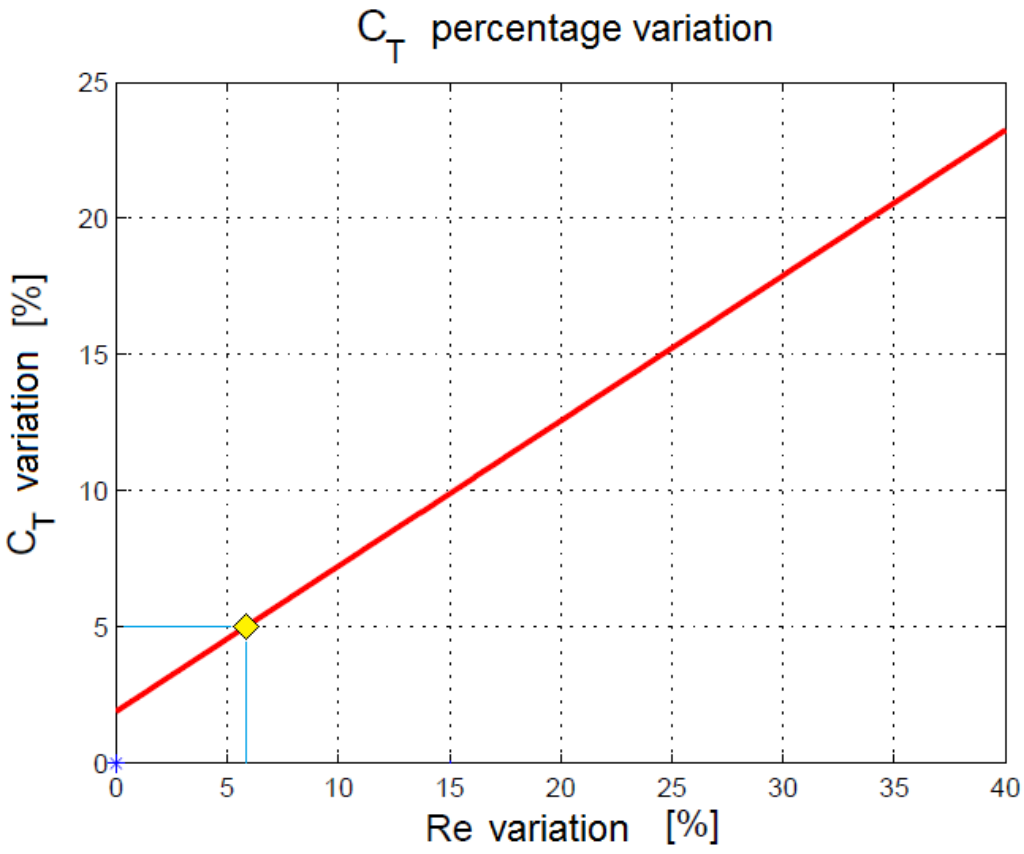


Figure 4.53 – C_T variation due to a Reynolds number variation

5. The control system

5.1 The on board software

For our scopes, the autopilot mainly has been used as an automatic control system for the execution of the flight tests and as an in-flight data recorder (FDR). The programming language used to give instructions to the system is its own language based on *Wiring* which is itself strictly derived from the widespread C++. As earlier mentioned, the APM software is based on the Integrated Development Environment (IDE) open source Arduino[4], which is a multiplatform written in Java derived from the IDE created for the programming language Processing and for the Wiring project; this platform was created to make it easier for designers and makers to realize interactive projects.

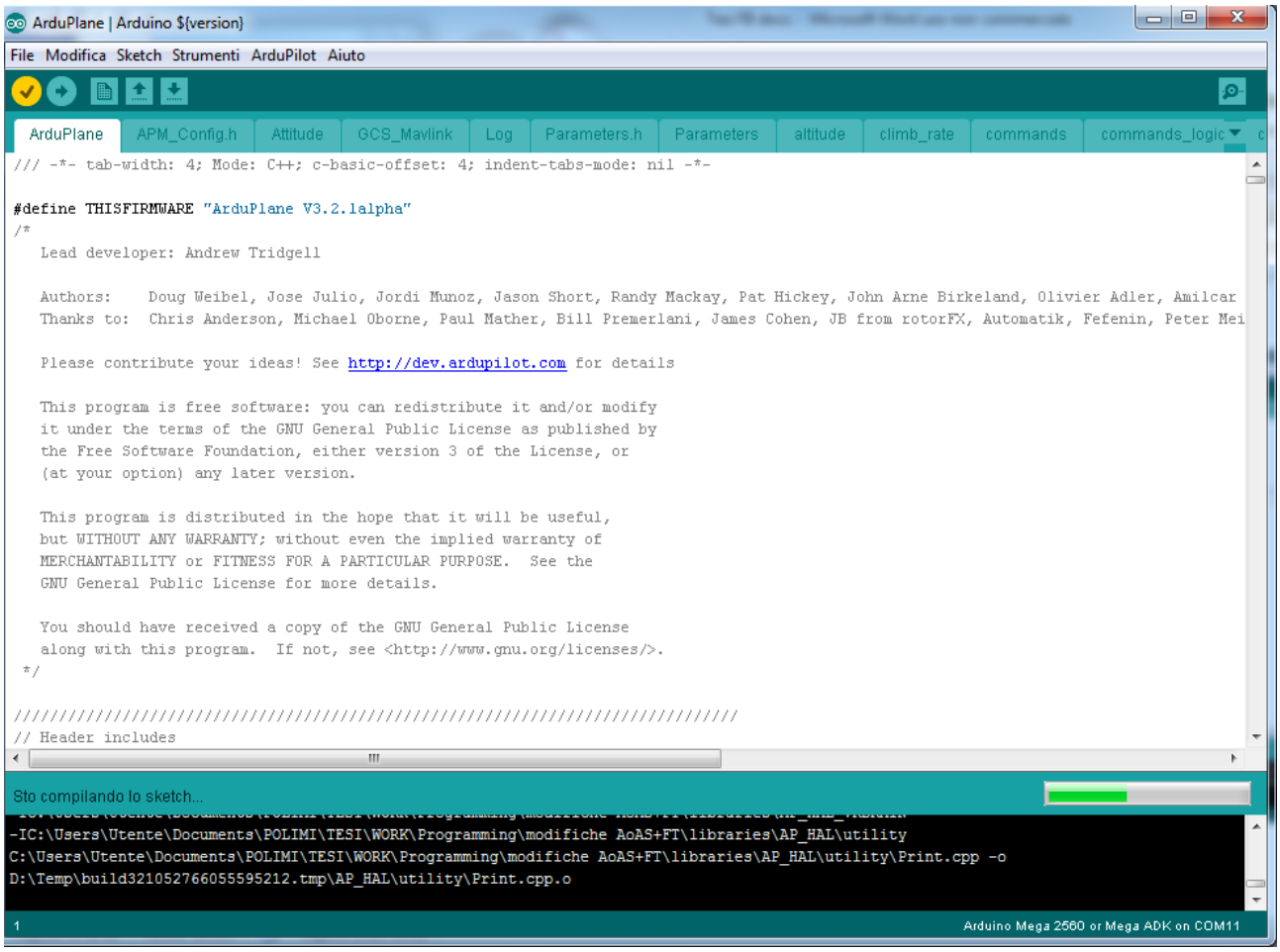


Figure 5.1 – IDE screenshot showing some codes lines while compiling

To write the source code, the IDE Arduino includes a text editor with some features as the syntax highlighting, parenthesis control and automatic indentation. From the same text editor interface, it's possible to verify and compile the code in the same action.

Talking now about the general features of the autopilot software, the Wiring programs, written in C++ and compiled by the Arduino IDE, are called "sketch" and need two main high level functions to be executed:

Setup( ) – A function called only once at the starting of the program, which defines some settings that won't be changed anymore during its execution.
Loop( ) – The function that will be repeatedly called until the powering off of the program.

The loop function in our case is subdivided into three sub-function:

Fastloop( ) – is a loop executed at a frequency of 50 Hz. In this loop are called all the functions relative to the reading of the radio signals, attitude, altitude and airspeed, the processing of the data and the sending of the output signals to the control surfaces.
Mediumloop( ) – executed at 10 Hz. It controls the communications with the GPS system and telemetry, elaborates the automatic navigation through the setting of waypoints.
Slowloop( ) – executed at 3.3 Hz frequency, in this loop are managed all the functions relative to the safety of the system and events control as the loss of radio signal e the in-flight restart.

The function managing the data logging is common to all these three loops and it has been modified to include the AOA and AOS readings in the "air data package".

At this point we can talk more specifically about the functions actually governing the airplane: the ARDUPLANE software running on the ArduPilot Mega hardware, just as released by the developers, implements some default flight modes with increasing autonomy at both stabilization and navigation levels:

- **Manual** – The controlling of the control surface movements is completely manual with deflection directly proportional to the movement of the transmitter sticks. The autopilot gets the input from the receiver and pass it on directly to the servos without any changes.

- **Stabilize** – The autopilot keeps automatically the aircraft in levelled flight (roll and pitch angles set to zero) when the transmitter sticks are centered in neutral position while the throttle channel is left completely to the pilot authority.

- **Circle** – The aircraft is kept to a constant bank angle (selectable) resulting in a continuous turn at the actual altitude but without keeping the GPS position of the trajectory center.

- **Fly by Wire A** – This is a semi-automatic mode in which the throttle is under the direct control of the pilot while the commands on ailerons end elevator channels are interpreted by the autopilot as set-points of desired roll and pitch angle in the specified range of authority.

- **Fly by Wire B** – As the FBWA but in this mode the autopilot will also try to keep altitude. A pitch command from the pilot is managed as a desired change in altitude. If elevator stick is kept centered then the autopilot will try to level the plane at the current altitude. The throttle channel is no more a direct pass-thru but a change in the throttle stick position is seen as a new airspeed target adjusting the throttle % consequently.

- **Autotune** – This mode flies in the same way as FBWA, but it does automatic tuning of roll and pitch control gains (see section 5.4).

- **Training** – This mode is like training wheels on a bicycle and is ideal for teaching students manual R/C control. If the roll is less than the set value, then the pilot has manual roll control. If the plane tries to roll past that limit then the roll will be held at that limit. The plane will not automatically roll back to level flight, but it will prevent the pilot from rolling past the limit. The same applies to pitch: the pilot has manual pitch control until the minimum and maximum limits are reached; at that point the autopilot won't allow the pitch to go past those limits. The rudder and throttle are both completely under manual control.

- **Acro** – This is a mode for advanced users that provides attitude rate change based stabilization with attitude lock. It is a good choice for people who want to push their plane harder than you can in FBWA or Stabilize mode without flying in Manual. This is the mode to use for rolls, loops and other basic aerobatic maneuvers, or if you just want an "on rails" manual flying mode.

- **Cruise** – Cruise mode is a bit like FBWB, but it has "heading lock". It is the ideal mode for longer distance FPV (Flight Per View) flight, as you can point the plane at a distant object and it will accurately track to that object, automatically controlling altitude, airspeed and heading.

- **Loiter** – In this mode the plane will circle around the point where you started the loiter, keeping the GPS position and holding altitude at the altitude that you entered loiter in. The radius of the circle can be set by changing a parameter.

- **RTL** – When in this mode the plane will return to the home point (the point where the plane first got a GPS lock) and loiter there until manual control is regained (or it runs out of fuel!). This mode is very useful to be set as default in the event of a failsafe.

- **Auto** – In full auto mode the autopilot will follow a mission (a set of GPS waypoints and other commands) set by your ground station configuration. When entering AUTO mode Plane will continue from whatever mission item it was last doing, unless you have reset the mission. The auto mode also includes the possibility of setting automatic takeoff and landing maneuvers.

- **Guided** – This mode is used when you want the aircraft to fly to a specific point on the map without setting up a mission. Most ground control stations support a "click to fly to" feature where you can click a point on the map and the aircraft will fly to that location and then loiter.

Even if some default flight modes may seem functional to the realization of a flight test, actually because of the mixing of throttle and pitch controls in the automatic modes in order to keep a desired altitude and/or airspeed, it's not possible to perform the trimmed, stall, glide and climb tests without these two controls being independent and without the possibility to be fixed. For this reason we had to implement four completely new modes, assuring the correct setting of the desired parameters during the test and the dependence of the variables on only the specific control. The description of the new flight modes will be presented in chapter number 6 as a specific flight mode has been developed for each flight test mentioned above.

However, both the default and custom flight modes are based on the two main control loops for the longitudinal and later-directional dynamics controls governing the autopilot logic. Each loop is made of a double PID (Proportional Integral Differential) control ring: the inner servo-loop controls the airplane attitude and the outer nav-loop controls its navigation.

Longitudinal dynamics:

The longitudinal dynamics is controlled by the variables of elevator deflection and throttle %. The inner stabilization ring gets as input the desired attitude angle which is compared with the current attitude angle; the resultant attitude error is the input of the regulator. The Servo_Pitch PID regulator, defined by the PTCH2SRV_P, PTCH2SRV_I, and PTCH2SRV_D gains, has the scope of reduce the error to zero through the control of elevator deflection. The outer navigation ring is based on the Alt_Nav PID regulator defined by the ALT2PTCH_P, ALT2PTCH_I and ALT2PTCH_D gains. It receive as input the altitude error, defined comparing current altitude coming from the barometric sensor with the reference altitude (set by the flight mode), and tries to bring it down to zero by calculating the attitude angle that is the input of inner ring. The throttle % variation value used as input in this loop as it influences the altitude, is normally controlled by another control ring using as variable the error coming from the desired airspeed and its actual value coming from the external airspeed sensor.

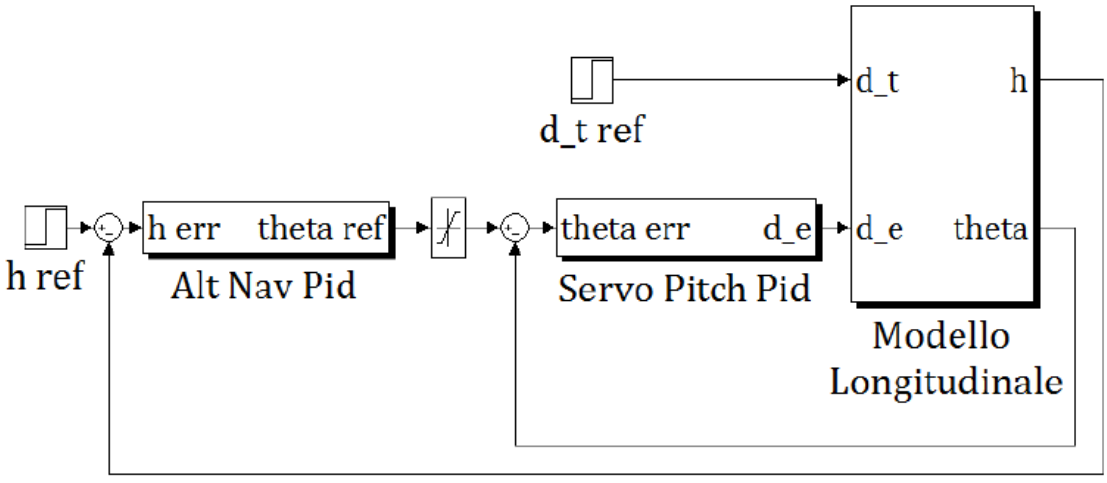


Figure 5.2 – Blocks scheme of the longitudinal control loop

Lateral - directional dynamics:

The lateral-directional dynamics is controlled by the variables of ailerons and rudder deflections. The ailerons control has an inner attitude ring based on the Servo_Roll PID regulator, determined by the RLL2SRV_P, RLL2SRV_I and RLL2SRV_D gains. The regulator gets as input the bank error calculated from the difference of desired bank angle and current attitude measured by the on-board gyros and gives as output the signal sent to the ailerons servos for the correct deflection of the control surfaces. The outer ring computes the error between the current heading and the reference heading (if any, depending on the active flight mode) and tries to bring it down to zero using the Nav_Roll PID regulator based on the HDNG2RLL_P, HDNG2RLL_I and HDNG2RLLV_D gains. In the lateral-directional dynamics is also present a parallel ring which controls the rudder output through a negative feedback on the lateral acceleration. The rudder control is also based on a crossed command for the ailerons deflection, the Aileron-Rudder Interconnect function (ARI). The scope of the ARI, based on the K gain has reported in the figure below, is to cancel out the induced yaw phenomenon* produced by the ailerons.

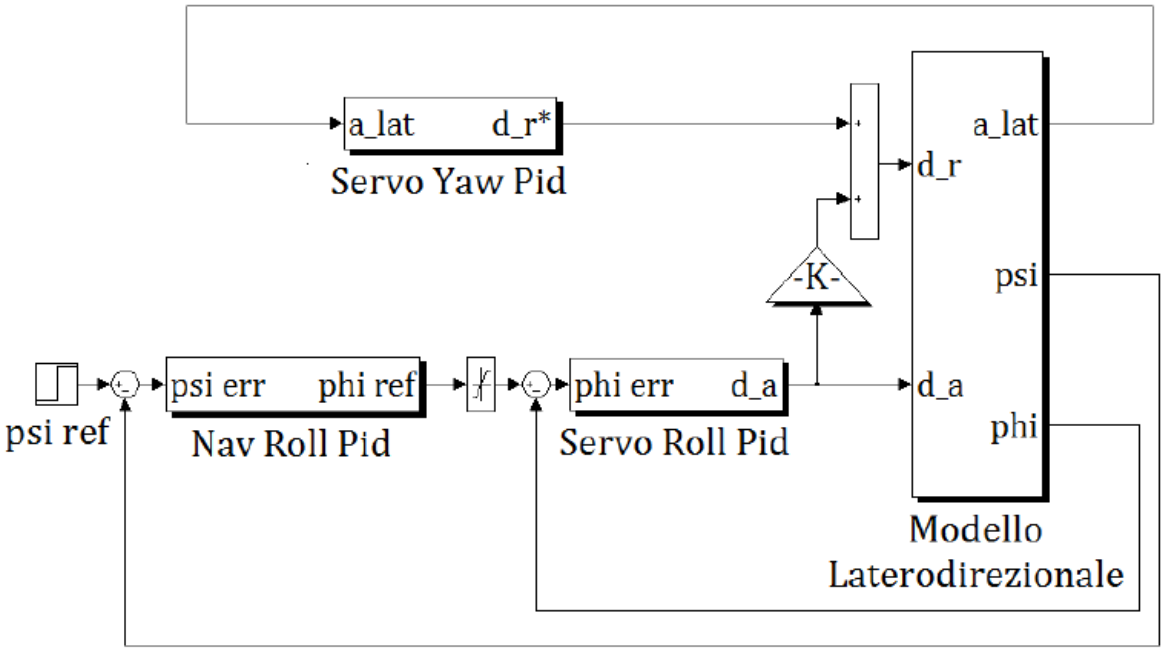


Figure 5.3 – Blocks scheme of the lateral-directional control loop

* The induced yaw phenomenon is due to the fact that when the ailerons are deflected during a turn, the aileron that lifts the external wing also produces more drag causing a yaw moment in the opposite direction of the turn.

5.2 The Ground Control Station (GCS)

The ground control station (GCS) is a very important element of the whole control system. It basically is the *Mission Planner* software latest version installed on a personal computer that can be transported on filed during the tests. The software, specifically developed to work with the APM, can be connect to the board through the micro-USB cable or through telemetry with the only difference in data baud rate (115200 bps for the cable vs. 57600 bps for the antennas). Through Mission Planner it is possible to set the desired autopilot and modes parameters changing their values without passing through the source code of the APM, connecting the system to a flight simulator as it will be discussed in the next section, define a flight path for a specific mission or monitoring in real time all the parameters recorded during flight. The real time attitude, altitude and airspeed data received from the telemetry are displayed through a graphic window showing some standard aviation instruments (e.g. artificial horizon, altimeter, vertical speed indicator..) for quick reading. The real time GPS position of the airplane is showed on a map powered by Google together with the set flight path (if any) and heading direction.

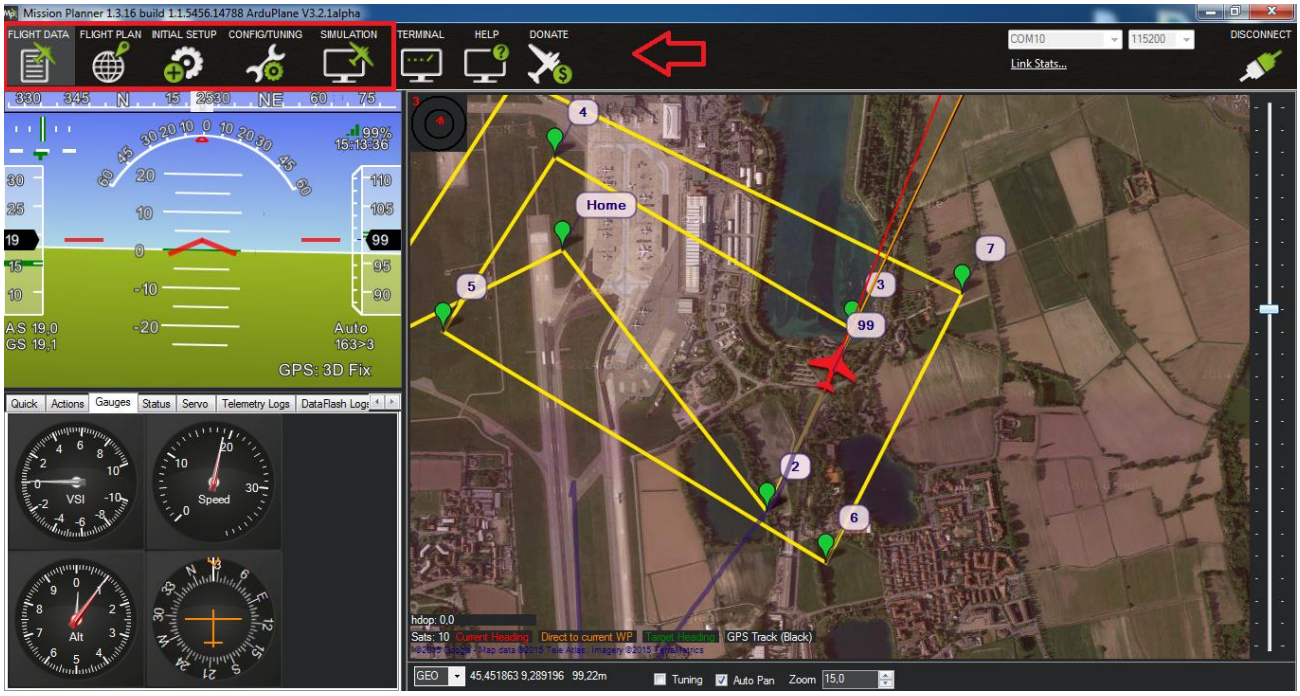


Figure 5.4 – Mission planner graphic window

In the figure above are circled the different tabs containing all the possible actions controlled by Mission Planner. In the next section we will talk more about the simulation and in section 5.4 will be discussed the configuration/tuning of the autopilot parameters in more detail. Here we focus more on the possibility to use Mission Planner as a ground control station being able to monitor all kinds of data coming from the onboard sensors through telemetry downlink in real time.

The next figure shows in more details all the information presented in the “Flight Data” tab:

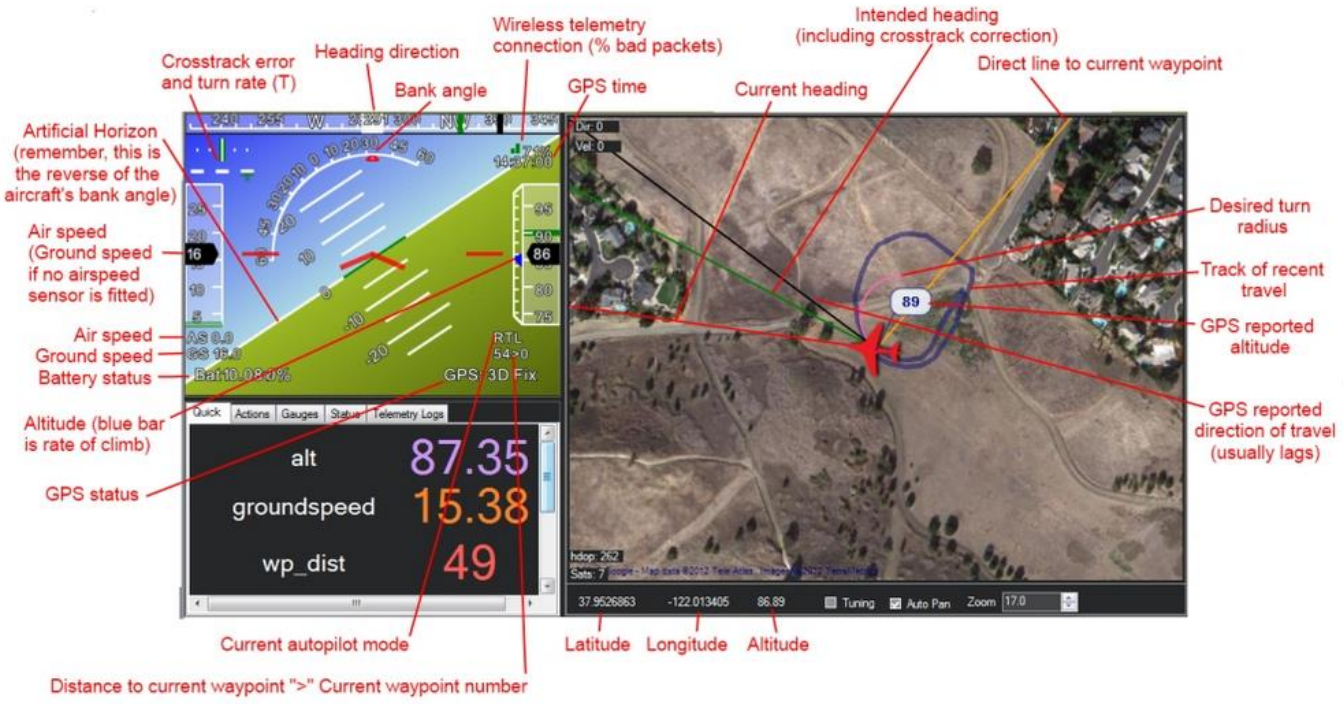


Figure 5.5 – Mission planner flight data tab informations

One last but not least important function of Mission Planner is the possibility to download the entire log of the data recorded in flight and to create a Matlab .mat file for post-processing. It also has a “Review a log” function to quickly review the time histories of the data for a first estimation. To access these functionalities one should click on the “DataFlash Logs” tab in the bar under the artificial horizon instrument as shown below:

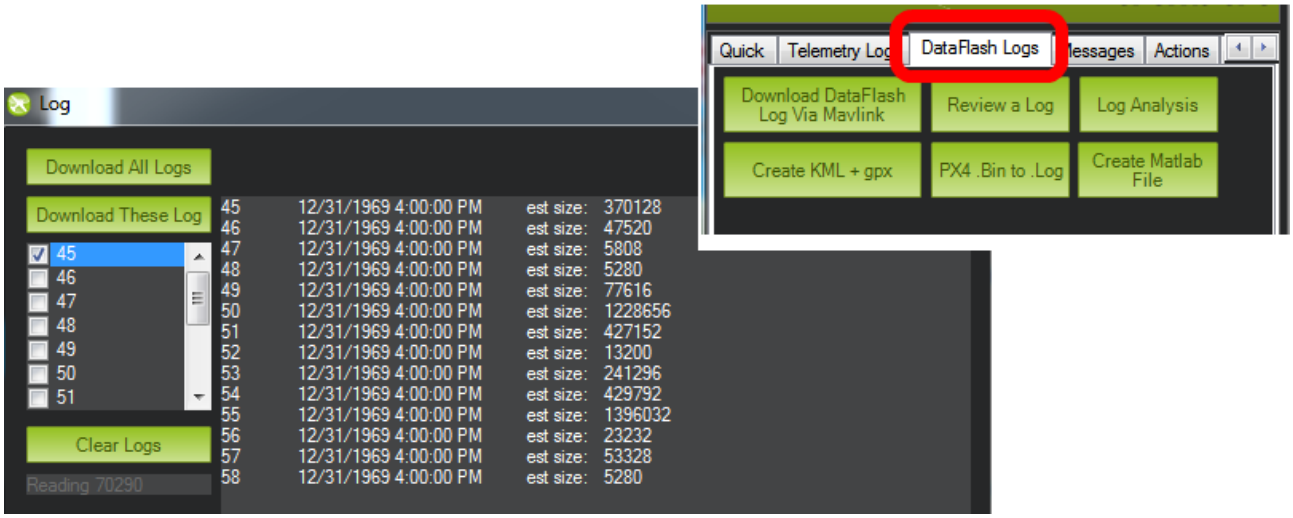


Figure5.6 – Mission planner DataFlash Logs actions list

5.3 Simulation

To verify the goodness of the results during the development of the new flight modes and because many preliminary real flights would have been too long and expensive along with a problem of safety for the aircraft itself, a first validation of the system in a simulated environment has been necessary. Specifically we used both the HIL (Hardware In the Loop) simulation, technique used for the development and testing of electronic embedded systems, and the SITL (Software In The Loop) simulation. The SITL is similar to the HIL with the difference that the executable code in this case is not running directly on the physical autopilot but also its hardware is simulated as well.

5.3.2 Hardware In The Loop

As described in figure 5.7, the autopilot software running on the onboard hardware don't receive the information from the real sensors but from a flight simulator running on another computer connected to the GCS through wlan link. To make the data transmission faster, considering the large amount of data exchange required by the simulation, the autopilot is connected to the GCS by USB cable. In this way the GCS simulation interface has the function of a bridge between the instructions coming from the autopilot and the data coming from the simulation. The normal instructions as the switching of the flight modes are given by the pilot as in the real flight through the transmitter communicating with the receiver also connected to the APM.

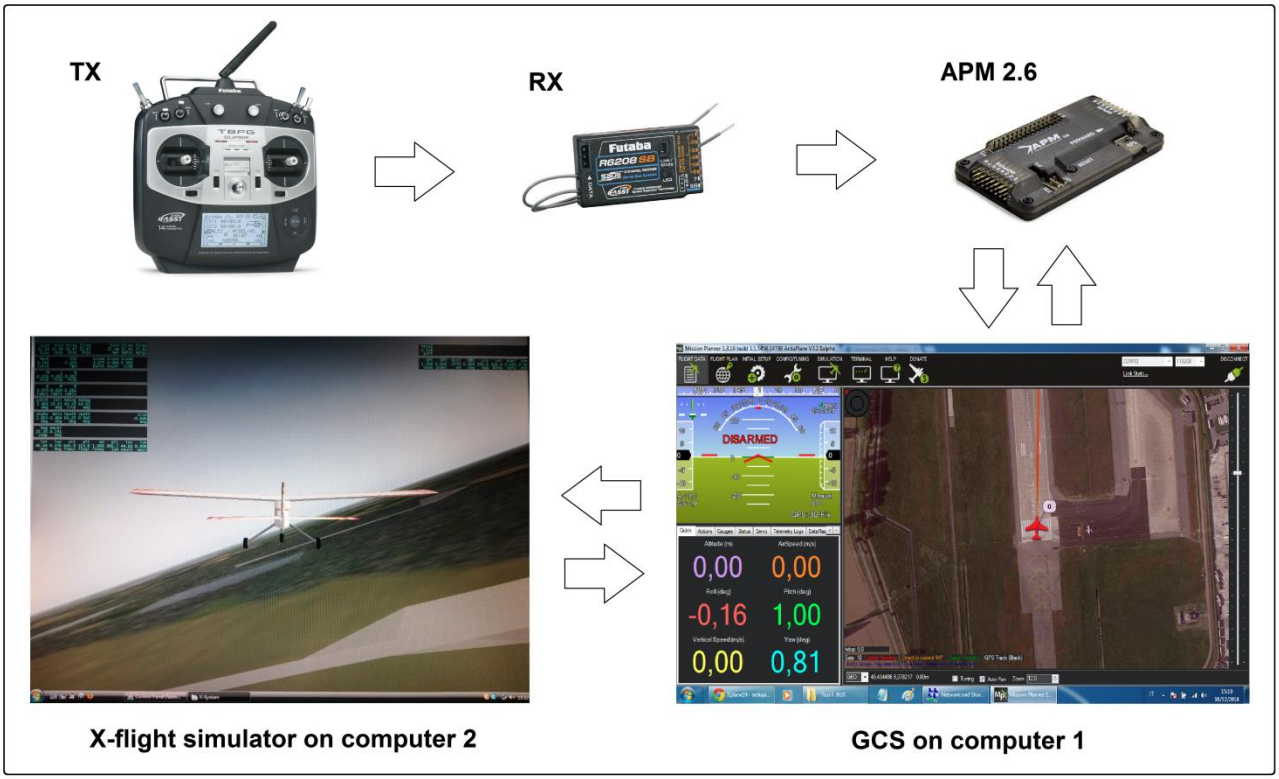


Figure 5.7 – Hardware in the loop chain

The simulation consists into the mutual interaction of the modified firmware of the autopilot with the simulated environment recreating the flight conditions in which the drone will be operating. The flight simulator software used for this scope is *Xplane 10*® which is described as the best flight simulator installable on PC. The Mission planner software has already implemented its own interface to connect to this type of flight simulator making the connection between *Xplane 10* and the GCS relatively fast and easy by only setting the right serial ports that will be communicating between the two computers. A second computer has been necessary instead of using only one PC for both software because the flight simulator needs a lot of computational power in order to correctly perform. Xplane has a selection of many different simulated aircrafts from where to choose and one of these in particular (the *Great Planes PT-60 RC* model) is similar in configuration and dimensions to our *Cularis*. This model has been choose to simulate the real aircraft flight dynamics and the general response to the autopilot inputs due to its similarity but the two airplanes are not quite the same with the *PT-60* being a trainer and the *Cularis* being a glider with many different characteristic from the wingspan to the type of motor. For this reasons, if the HIL simulation is a good instrument to verify the correct implementation of the new flight modes observing the simulated airplane response, it is not accurate in determining some specific parameters as the maximum and minimum airspeeds and the control loops gains which are different for every kind of airplane. The HIL simulation also realizes the data recording of all the simulated parameters on the autopilot flash memory just as in the real case.

Figure 5.8– The HIL simulation running on the two computers

5.3.1 Software In The Loop

The software in the loop simulation (SITL), as anticipated before, is the same as the HIL but it is capable of running the autopilot governing code without the need of a physical hardware. This is quite convenient to run a faster and lighter simulation that can be managed by one single computer and without carrying around the airplane with all its on-board systems already installed. This type of simulation, however, needs a Linux based operating system because the open source flight dynamics simulator software *JSBSym* (which is the one also powering the better known open source flight simulator *Flight Gear*) is only available on this type of operating system. On the contrary, the Mission Planner software is only supported by Windows. To run the two software on a single computer it has been necessary to create a virtual machine with the application *Oracle VM VirtualBox* that allows to run a secondary operative system as having another computer (*Ubuntu Linux* in this case) running in the host operating system which is *Windows*. This procedure may seem quite complicated but it's easier to set up than to explain following the instructions. The modified source code is then compiled and run into the simulated autopilot hardware system with the general environmental conditions and flight dynamics simulated by *JSBSym*. The resulting behavior of the UAV can be monitored on the *JSBSym* interface or on the more practical Mission Planner flight data tab which is connected to the hosted application through IP address. A very useful feature of this simulation system is that just charging the geographical coordinates of any places on earth, the UAV will perform its mission on the map of the desired place which in our case it has been set as the field of San Vito di Gaggiano where the real flight tests were performed.

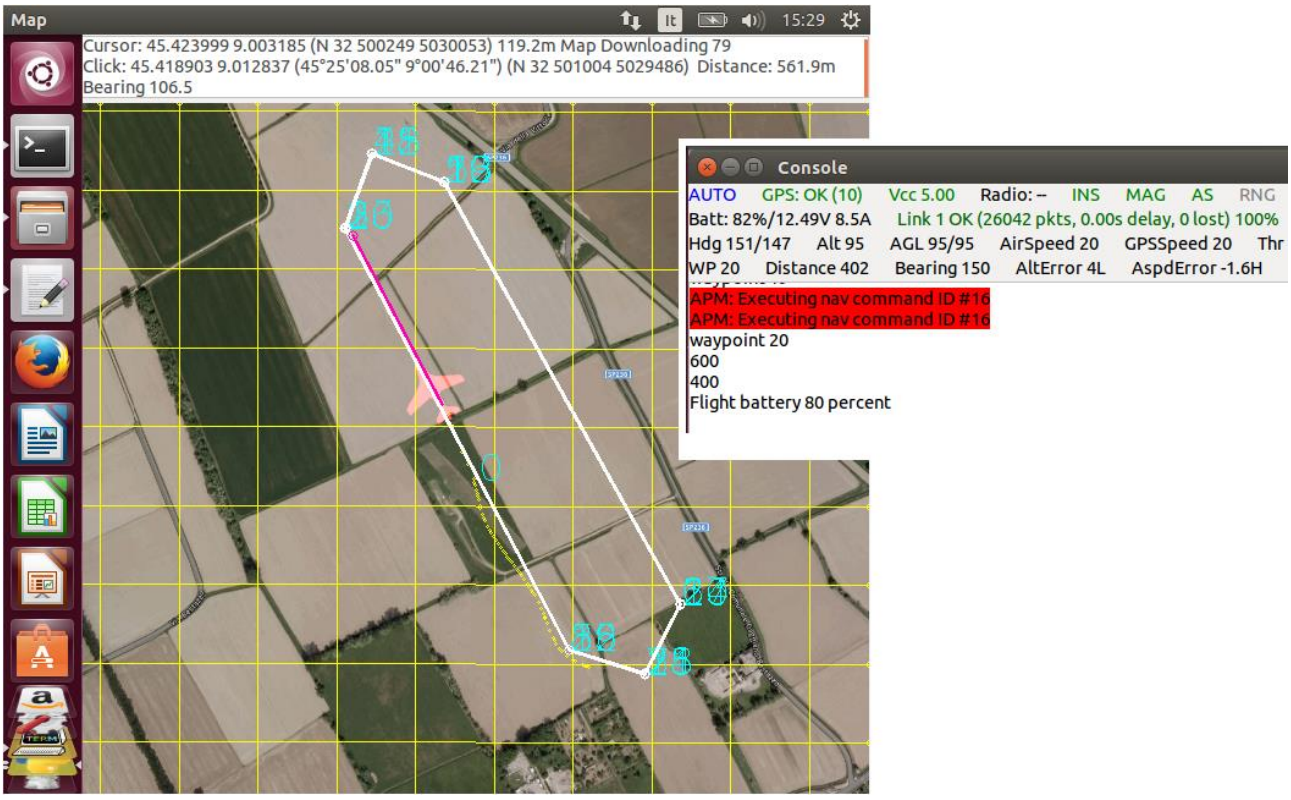


Figure 5.9 – The *JSBSim* graphic window together with the command line console

Figure 5.10 – SITL simulation run on the real flight tests local field map

This possibility of locating the simulation output on the user local field it is useful especially in the case that you have to design a mission with waypoints and you got to make sure that the actual aircraft trajectory won't be too close to any obstacles like houses or trees. In the Xplane HIL simulation this was not possible because the software is primary intended to simulate real airplanes and the default selectable geographic places are all real airports (all the HIL example figures in the previous section show the map of the Milano Linate intl. airport).

5.4 Parameters setting and gains tuning

Same required parameters are to be set in advance before the first real flight of the autopilot system, others have to be tuned during and after the first flight.

Starting with the parameters to be set before, we calibrated the APM gyros and accelerometers using the Mission Planner dedicated function in the "Initial Setup" tab. Same thing had to be done with the APM external magnetometer. For the transmitter/receiver, the characteristic frequencies for the manual mode channels and the radio PWM intervals of the 6-positions switch controlling the changing of the modes (actually is a dual position switch controlling another 3-positions switch resulting in 6 possible combinations), had to be registered in the APM database. The software asked to move all the transmitter sticks all the way from their minimum to their maximum position, to calibrate the proportional autopilot response, and to choose a flight mode for each switch position (see the next two figures).

Figure 5.11 – Radio channels calibration

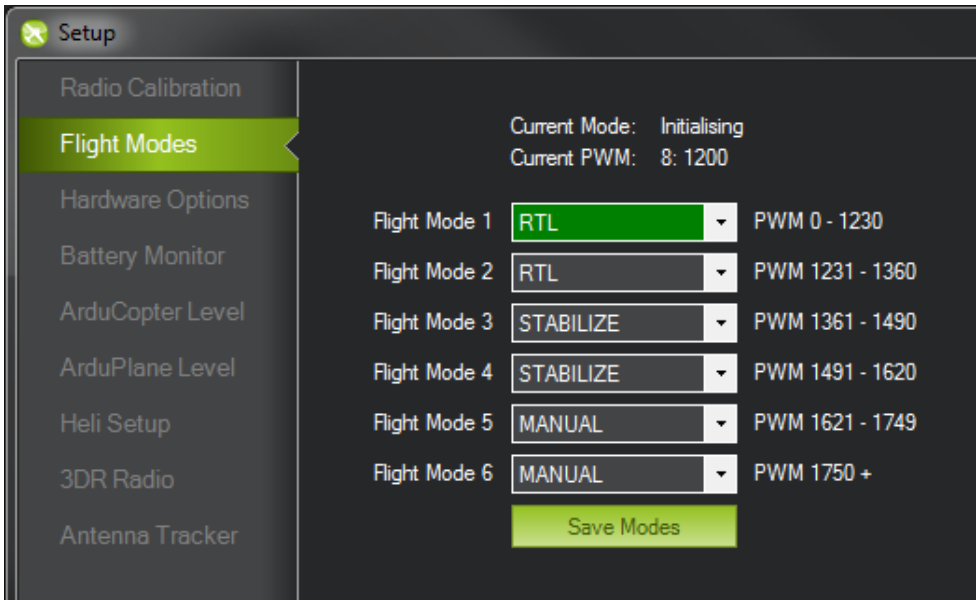


Figure 5.12 – Setting of the desired flight mode for each Tx switch position

Other parameters settable in advance comprehend:

ARMING_REQUIRE : set to 2 this parameter doesn't allow the propeller to spin at all unless the motor is expressively armed by the pilot by holding the rudder stick all the way to the left for a few seconds. (This is extremely important if you don't want to chop off your finger or anything else close to the propeller by accident).

FS_SHORT_ACTN: The action to take on a short failsafe event. A short failsafe is triggered by a loss of RC control for the default maximum time of 1.5 seconds. The code relative to the short failsafe has been hacked and a value of 2 for this parameter now results into switching from whatever mode the autopilot was in before the event to STABILIZE mode.

FS_LONG_ACTN: The action to take on a long failsafe event. When the RC signal is lost for more than default time which is 5 seconds, the autopilot will change to RTL mode when this parameter is set to 1.

ARSP_PIN: this parameter is used by the autopilot software to find the information coming from the airspeed sensor. It indicates the port number to which the analogic signal comes from. In our cases the port is the A4 indicated as number 4.

(Note: the default pins for AOA and AOS signals are hardcoded and respectively A0 and A2 ports of the analogic rail)

ARSP_RATIO: the parameter set in the wind tunnel to relate the measured differential pressure and the airspeed value (see section 4.3.1). Set to 2,0976.

ARSP_ENABLE: enables the passive acquisition of the airspeed value. Yes = 1.

ARSP_USE: use the airspeed as an active variable for navigation computations. Yes = 1.

THR_MIN: the minimum throttle % value applicable by the autopilot authority. Set to 0.

THR_MAX: the maximum throttle % value applicable by the autopilot authority. Set to 100. (Actually this parameter has been correctly set only after witnessing a problem with the trimmed flight tests with the throttle that wouldn't increase above 75% because of this parameter set to 75 by default).

LOG_BITMASK: this parameter allows the user to make a selection among all the measured data that will also be saved in the on-board flash memory. It is suggested to include all the needed data but no more than those due to the not very big storage memory (16 MB as reported in section 2.2.2). Running out of memory will result in a overwriting of the data. Each data type has a number and the resulting sum of the numbers relative to the selected data defines the LOG_BITMASK parameter set value which we set as 1270.

The following table shows all the useful data recorded with this LOG_BITMASK set value:

01 time t [ms]	08 h gps [m]	15 p [deg/s]	22 AOA [deg]
02 pitch θ [deg]	09 v gps [m/s]	16 q [deg/s]	23 AOS [deg]
03 roll ϕ [deg]	10 Latitudine [deg]	17 r [deg/s]	24 Nav roll [deg]
04 yaw ψ [deg]	11 Longitudine [deg]	18 acc x [m/s^2]	25 Nav pitch [deg]
05 v EAS [m/s]	12 Ground Crs [deg]	29 acc y [m/s^2]	26 Mag x [deg]
06 h baro [m]	13 flight mode [-]	20 acc z [m/s^2]	27 Mag y [deg]
07 t gps [_s]	14 throttle out [%]	21 Arm state [-]	28 Mag z [deg]

Table 5.1– In flight recorded data list

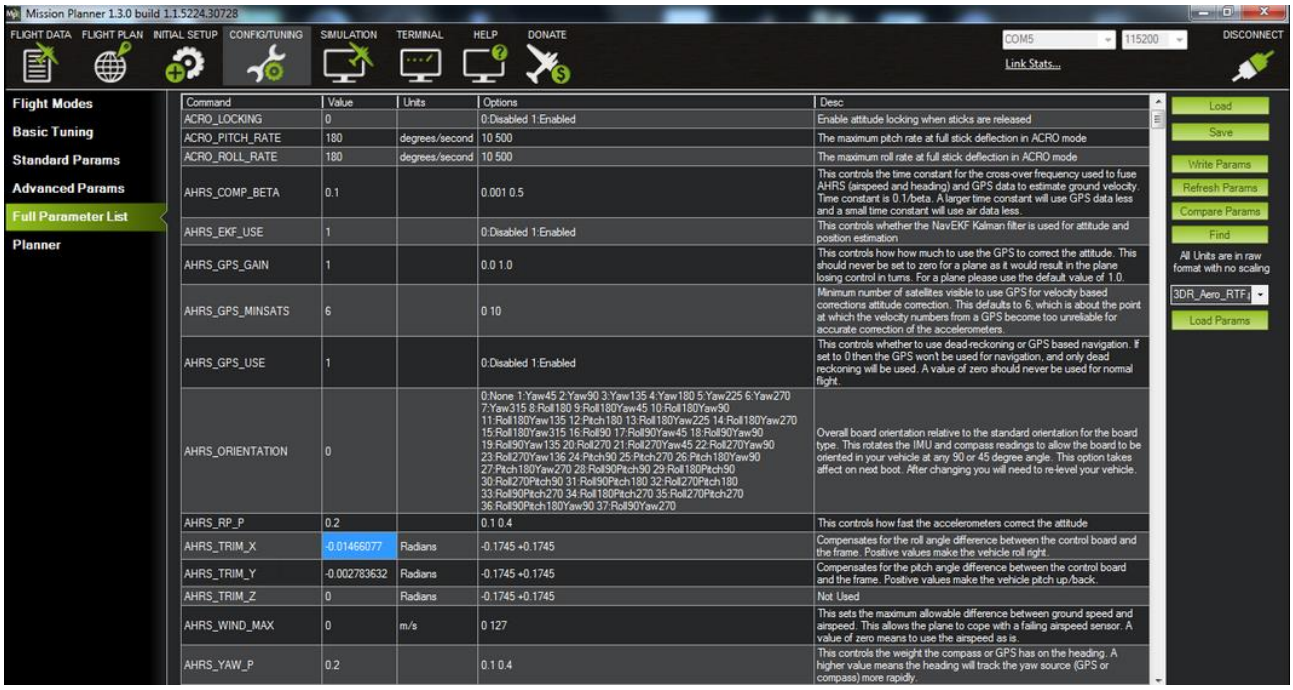


Figure 5.13 – Screenshot of the Mission Planner "Full Parameter List" tab

Other parameters had to be set necessarily after the first flight depending on the observation of the results. One preliminary flight with the whole autopilot system installed on the aircraft has been performed on field the 31st of March, 2015. Another pilot friend of mine helped me with the manual piloting of the aircraft while I was looking at the GCS. In this first test the control of the autopilot has been limited to manual, stabilize, FBWA and AUTOTUNE mode.

The first thing verified in this preliminary test has been (as for any other simple RC models) the correct center of gravity positon that resulted as perfect. With the airplane already trimmed in manual mode (the airplane flies straight and levelled with the motor off), the first parameter could be set:

TRIM_AUTO: when this option is enabled (1) and you change from MANUAL to any other mode then the APM will take the current position of the control sticks as the trim values for aileron, elevator and rudder. After you have good trim on your aircraft you can disable this option for future flights. It is better to disable this option as if the pilot is not aware or don't remember this option was on and changes from MANUAL to another mode while control inputs are not centered then the trim could be changed to a dangerously bad value.

So after trimming the airplane in manual mode and briefly switching to stabilize then back again to manual, the auto trim parameter have been set to 0 disabling it.

By setting the throttle to zero and trying to keep the pitch angle to zero until stall and then repeating the test pushing the throttle to its maximum and keeping it in level flight, three more important parameters were set observing the resulting airspeed changes through the GCS:

ARSPD_FBW_MIN: This is the minimum airspeed you want to fly at in modes where the autopilot controls the airspeed in m/s. This should be set to a value around 20% higher than the level flight stall speed for the airframe.

ARSPD_FBW_MAX: This is the maximum airspeed in m/s that you want to allow for your airframe in auto-throttle modes. You should ensure that this value is sufficiently above the ARSPD_FBW_MIN value to allow for a sufficient flight envelope to accurately control altitude using airspeed. A value at least 50% above ARSPD_FBW_MIN is recommended.

TRIM_ARSPD_CM: Airspeed in cm/s to aim for when airspeed is enabled in auto mode. It can be considered as the "cruise speed".

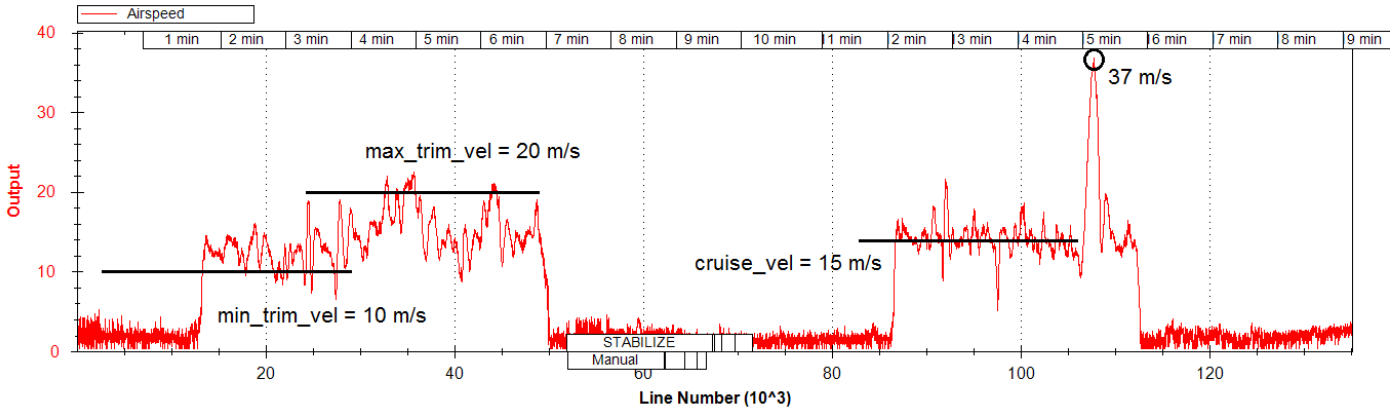


Figure 5.14 – Raw airspeed data

Referring to the above graph plotted using the "Review a Log" Mission Planner tool, after a quick evaluation the above parameters have been set accordingly leaving some safety margin (the 37 m/s peak is a dive speed not to take into account):

ARSPD_FBW_MIN = 12;
ARSPD_FBW_MAX = 22;
TRIM_ARSPD_CM = 1600;

Another observation resulting from these flights is the minimum altitude considered as safe to recover from a deep stall which is 80 meters and the maximum altitude at which the airplane attitude is still clearly visible from the ground which is about 200 meters. In terms of horizontal distance the visibility of the airplane is to be considered safe for a maximum of 400 meters from the pilot location. Remembering that one of the few rules governing the UAV flight is that the aircraft has always to be in the pilot direct line-of-sight at any time, we kept these measurements in mind for the implementation of the new flight modes.

Gains tuning using AUTOTUNE:

Getting a good set of roll/pitch tuning parameters for the aircraft is essential for stable, accurate flight in auto-modes. To help with this a specific flight mode was implemented to automatically adjust the gains depending on the type of airframe you are flying with. This system is very convenient for a fast and quite accurate tuning without manually having to change the parameters during multiple preliminary flights. After setting all the required parameters above, the specific gains used to fly in auto-modes could be set using this feature.

The AUTOTUNE mode is a flight mode that flies in the same way as FBWA, but uses changes in flight attitude input by the pilot to learn the key values for roll and pitch tuning. So the pilot uses its transmitter mode switch to switch to AUTOTUNE mode and then flies the plane for a few minutes. While flying the pilot needs to input as many sharp attitude changes as possible so that the autotune code can learn how the aircraft responds.

The AUTOTUNE_LEVEL parameter has been set to 7 in the advanced parameter screen of the ground station. The AUTOTUNE_LEVEL parameter controls how aggressive you want the tune to be. The default is level 6, which produces a medium tune, suitable for beginner to intermediate pilots. For more experienced pilot is suitable to choose level 7, which will result in a bit sharper tune (faster attitude changes). Levels above 7 are not recommended until an initial tune with a lower level has been done.

Once everything was set, we took off in manual mode and then switched to AUTOTUNE once we gained altitude. When engaging AUTOTUNE mode we had to be aware of the following points:

- the autotune system immediately setup some default values for your roll and pitch I and D gains, and your roll and pitch maximum rates. These values depend on the AUTOTUNE_LEVEL.
- the autotune system will monitor the demanded roll and pitch rates (as determined by the transmitter stick movements). When the demanded roll or pitch rate exceeds 80% of the maximum rate the autotune system uses the response of the aircraft to learn roll or pitch tuning values to quicken the response.
- every 10 seconds the autotune system saves the parameters you had 10 seconds ago. This means that if autotune causes the aircraft to become unstable, the pilot has 10 seconds to switch to another mode and recover. When switching out of AUTOTUNE mode the last saved parameters are restored.
- Starting with the default parameters for roll and pitch you may find the plane is quite sluggish when first entering AUTOTUNE. As the tune progresses this will get better.

The key to a successful autotune is to input rapid roll or pitch movements with the transmitter sticks. It should be done only one of either roll or pitch at a time, moving the stick rapidly to the maximum deflection.

For the roll gains tuning the plane will steer hard right, then hard left as you move the aileron stick. With each sudden reversal it will improve the tuning values by about 5%. The autotune needs at least 20 full stick movements to learn a reasonable tuning value. For pitch tuning the pitch transmitter stick has to be used to take the aircraft on a roller-coaster ride. Pulling back hard on the stick to pitch up, then shortly afterwards push down to pitch down. Also in this case it is preferable to continue doing this for at least 20 iterations.

As by default the initial tuning values were too low, we actually noticed that the aircraft became progressively more responsive as flying in AUTOTUNE mode. After more than 20 iterations, at a response level we were happy with, we landed the plane and connected it to the GCS for data logging. The typical ATRP files showing the progress of the autotune has been recorded in the dataflash log:

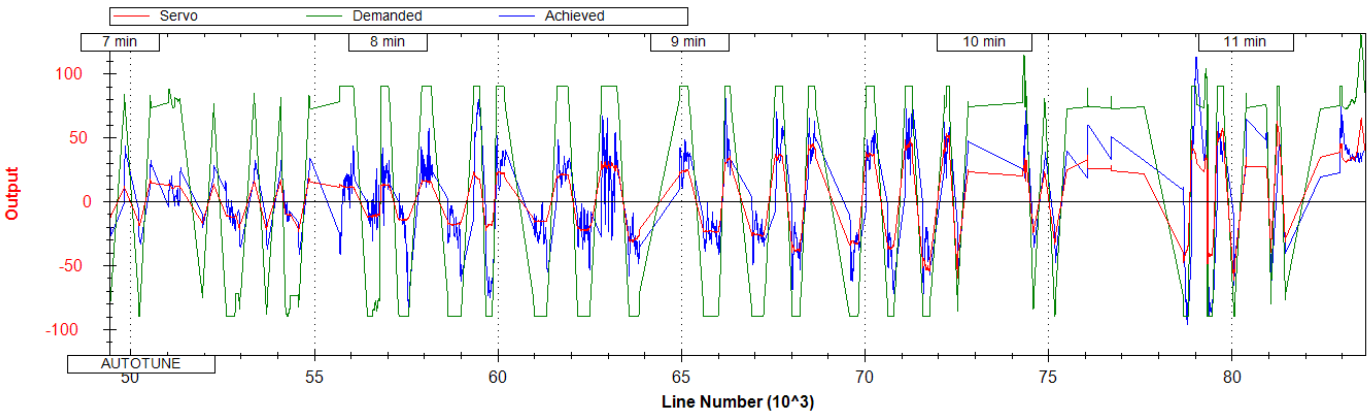


Figure 5.15 – Autotune progress for the pitch rate tuning

The ATRP.Demanded field is the demanded rate of attitude change (roll rate or pitch rate) in degrees per second. The ATRP.Achieved field is what the aircraft actually achieved in attitude change rate. As you can see in the above graph, at the start of the autotune the demanded values were much higher than the achieved, because the tuning gains were too low. As the tune progressed the demanded and achieved started to converge.

The final comparison between starting values and final values of the gains (automatically recorded by the system) can be seen in the tables below:

OLD	K_P	K_I	K_D
Roll	0,4	0,0	0,0
Pitch	0,6	0,0	0,0
Yaw	1,0	0,0	0,0

NEW	K_P	K_I	K_D
Roll	1,463	0,040	0,117
Pitch	1,078	0,060	0,086
Yaw	1,0	0,0	0,0

Table 5.2 – Tuned gains vs. default values

At this point the whole aircraft + autopilot system was finally ready to perform the flight tests in the new auto-modes.

Figure 5.16 – The *CX-15* ready to go

6. Flight testing

In this chapter will be presented all the flight tests performed, including the theory behind the design of each flight test, its preparation and the results.

6.1 Test range and implementation

The whole flight test campaign has been performed at the RC model field of San Vito di Gaggiano (Milano, Italy), property of the GAT club). In order to get the permit to operate on this field, we needed to register. The registration with the club includes an insurance for damages to third parties valid in all UE countries.

Figure 6.1 – The RC model field of San Vito di Gaggiano

Coordinates: LAT: 45.422070, LONG: 9.006032, ALT: 118 msl
Runway length: 195 m

The field is located about 20 kilometers from Politecnico di Milano and to transport the airplane together with all the other instruments needed for the flight tests, the van of the aerospace department has been reserved on the flight tests days. Depending on traffic, the time required to get to the field can vary between 30 and 40 minutes. We usually arrived at the field around 10 a.m. and we left around 4 or 5 p.m. depending on the type of flight tests that were scheduled for the day.

Together with the aircraft, the items required for the on field tests include: support for the model when on the ground (very useful to perform the level), extra batteries and charger, radio transmitter, portable computer as GCS, portable meteorological station to record the environmental data during the test.

Figure 6.2 – The UAV inside the mini-van for transport

6.2 Ground meteorological station

The portable ground meteorological station (GMS) that supported all the flight tests on the field, is a design property of the flight testing department of the university. It can measure in real time the data of absolute pressure QFE [hPa], relative humidity RH [%], temperature T [°C], wind speed and direction [m/s - deg].

More specifically, the system is based on a *Olimex E407* open source hardware board, which integrates an absolute pressure sensor *lps25H* produced by *St.Microelectronics*, temperature and relative humidity sensors installed inside the protective screen against solar radiation *SHT75* by *Sensiron*, a GPS unit *Ublox LEA4T*, mechanical anemometer for the measure of the wind speed and vane to indicated its direction. The LCD 20x4 characters screen is based on the Hitachi HD44780 controller and presents the real-time measurements on multiple pages scrollable through rotational encoder.

Even if we didn't use this last added feature for our purposes, the GMS is also capable to connect to a dedicated telemetry module through a radio access point permitting TCP/IP WIFI communications.

Figure 6.3 – The Ground Meteorological Station

Good weather conditions are needed to execute the flight tests (no strong wind or rain) and the missions had to be planned in advance following the weather forecasts. Following the logging of the four missions carried out from the end of March to the end of May 2015 along with the type of tests performed and the mean values of pressure, temperature and relative humidity registered during the day:

#	Date [dd.mm.yy]	Test type	Pressure [hPa]	Temperature [°C]	RH [%]
1	31.03.15	Preliminary parameters tuning	1007	17	46
2	21.04.15	Wind determination, Trimmed and stall	1018	21	39
3	07.05.15	Trimmed and stall	1010	23	63
4	28.05.15	Glide/Climb	1010	22	45

Table 6.1 – Missions log

6.3 Pre-flight and post-flight procedures

All the flight tests, from the initial inertial calibration of the system to the download of the flight data, have some common procedures that will briefly be described here:

- **Power on of the system and connection to the GCS**: with the model assembled and APM installed on board with all sensors and battery (disconnected!!), the Mission Planner software can be launched and the ground telemetry antenna connected to the computer via USB cable. Before connecting the battery to the APM power module, it is necessary to turn on the transmitter with the mode switch on manual and the throttle stick set to the zero position. At this point the APM can be powered on by connecting the Li-Poly battery and the connection established between the two telemetry modules by selecting the right COM port and baud rate (57600 bps). After the connection is establish, all the parameters on the GCS will start to move accordingly to the measured values by the onboard sensors.

- **Sensors calibration and parameters setup**: with the plane levelled on its dedicated support, the inertial platform has to be zeroed. Gyros and accelerometers, when the calibration button is pressed on the GCS, will consider the acquired initial data as the zero pitch and roll reference. Also the airspeed sensor has to be zeroed; this has to be done remembering to cover the pitot tube with a loose cap in order to prevent a bad zero setting caused by wind blowing in the tube (remove before flying!!). Through the control station at this point the pilot needs to set or change all those custom parameters necessary for the new flight modes definition (the detailed description of each new parameter will be included in the flight test section specific to that mode). At this point the UAV is ready to take off.

- **Arming propeller and manual take-off**: While walking to the field for take-off the pilot tries out before each flight the correct movements of all the control surfaces. At the same time arms the motor (see Arming Require parameter in section 5.4) and tries out the throttle to see if the spinner rotates freely. If all checks are positive, keeping up with one arm the model, gives full throttle and then throws the airplane straight and levelled then immediately corrects attitude with ailerons and pitch as the UAV starts to gain altitude.

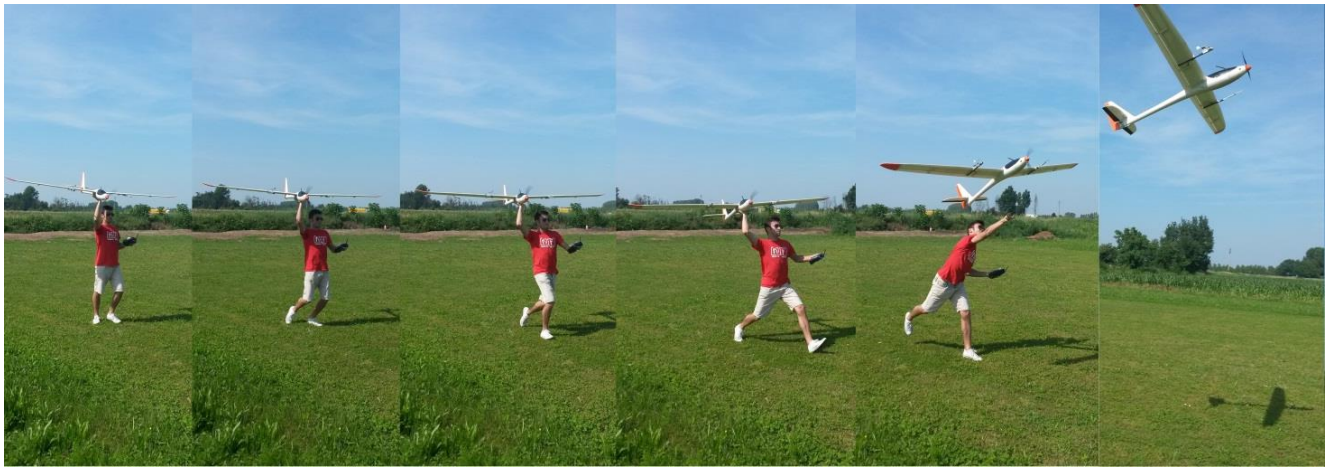


Figure 6.4 – Standard take-off maneuver

- **Execution of the test**: after the UAV has reached the safe altitude in manual mode (about 100 m), the pilot starts the actual flight test by switching through the transmitter to the desired flight test mode. At this point the full authority on the control surfaces is handed on to the autopilot that will perform one test point following that specific flight test control logic. The pilot it's not moving any of the control sticks and it's simply observing the correct execution of the test point but has to be always ready to take over if any problem arises. After a test point is considered complete (in the specific limits of time and/or distance characteristic to that test), another test point can be performed by only repositioning the UAV in manual mode and then engaging again the auto-mode.

- **Landing and data download**: When the onboard battery gets too low it's time to land. Usually the typical flight time allowed by one battery it's between 10 and 20 minutes depending on how much throttle has been used during the test. The pilot should be aware of this inconvenient keeping a bit of a margin to not end up with a drained battery in the middle of a test point. The landing is the most delicate maneuver and it's performed manually by the pilot with the aid of the flaps to reduce the landing speed. Once on the ground the first thing to do is disconnect the UAV battery, then the transmitter can be also powered off. Then the APM is connected to the GCS via USB cable (allowing a larger baud rate of 115200 bps) and the data log of the whole flight is downloaded and saved on the computer for post-processing.

Figure 6.5 – Standard landing maneuver

Safety considerations

Keep in any case a reasonable altitude during the performing of all the flight tests (minimum safe limit to recover from a deep stall in manual mode was defined as 80 meters during the preliminary first flight).

Keep the option "arming require" always on to avoid accidental engaging of the FT modes that might cause the propeller to start spinning on the ground. Only "arm" the throttle if ready to take off, in manual mode and with the stick throttle in lower position.

Always keep clear from people during take-off and landing maneuvers.

6.4 Wind estimation

This first test is the classic "square mission" used in flight testing to determine speed and direction of the wind on the field. The secondary objective of this test is to verify the fully autonomous flight capability of the autopilot tuned as in the first preliminary flight and the correct functionality of all the sensors and data recording system. The flight mode used to performed this test is the default AUTO mode.

6.4.1 Theory and design of the test

This classic flight test consists in flying four stabilized legs (800 m each) at about 90° azimuth intervals at a constant IAS and altitude (150 m). The four legs of the flight path are the sides of a square whose vertices are GPS way-points previously set through the GCS flight planner software. This mission will be repeated twice, before and after all the other tests. In the post processing, considering the data of IAS coming from the airspeed sensor and the information of ground speed (GS) and heading coming from the GPS, we can deduce the wind speed (WS) intensity and direction using the algorithm presented in the next figure. The advantage of flying a square path instead of a triangle is that from one test we can obtain four results instead of only one by using three legs at the time and mixing them in different combinations every time. This way we have a more accurate result by taking the mean of the four calculations and a feedback of how constant the wind has been during the test.

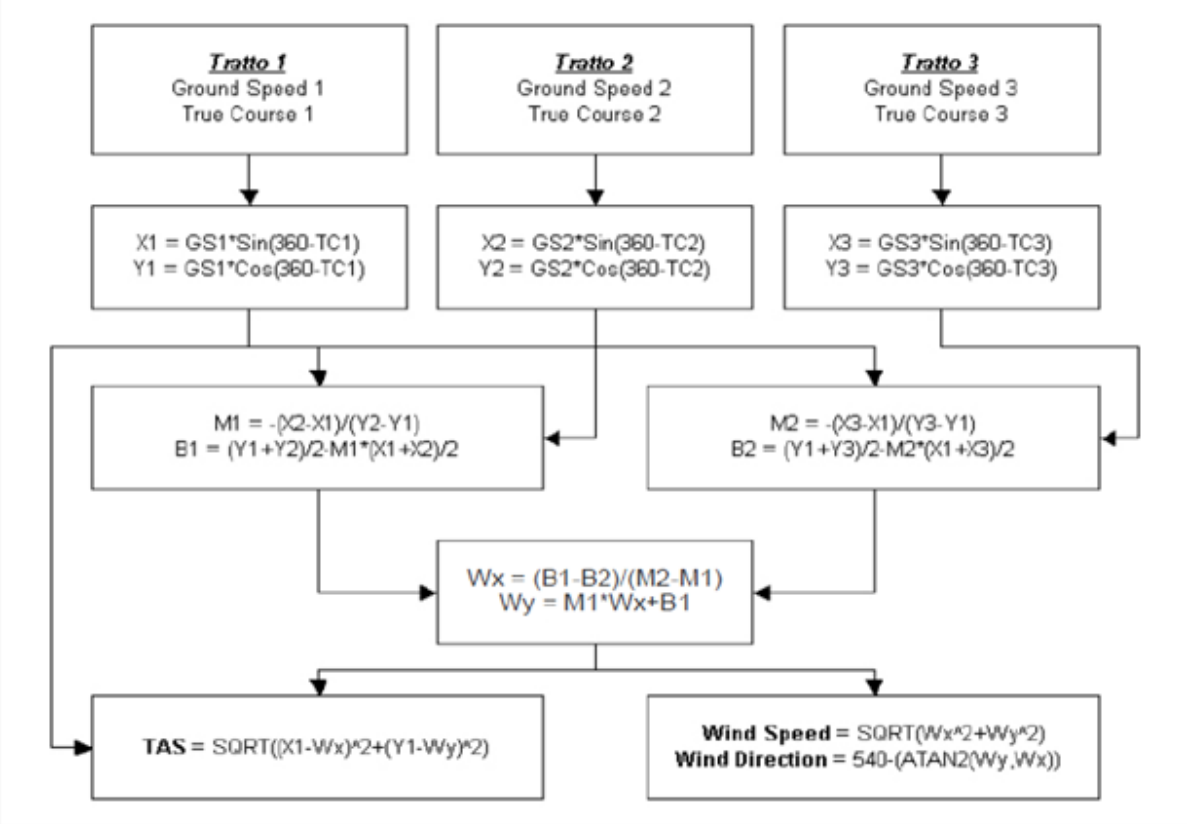


Figure 6.6 – Wind speed and direction resolution algorithm

The flight card with the waypoints the autopilot was instructed to follow in order to complete the square track mission (see the figure below) has been set through the Mission Planner "Flight Plan" tab. This mission has been firstly simulated with the SITL method.

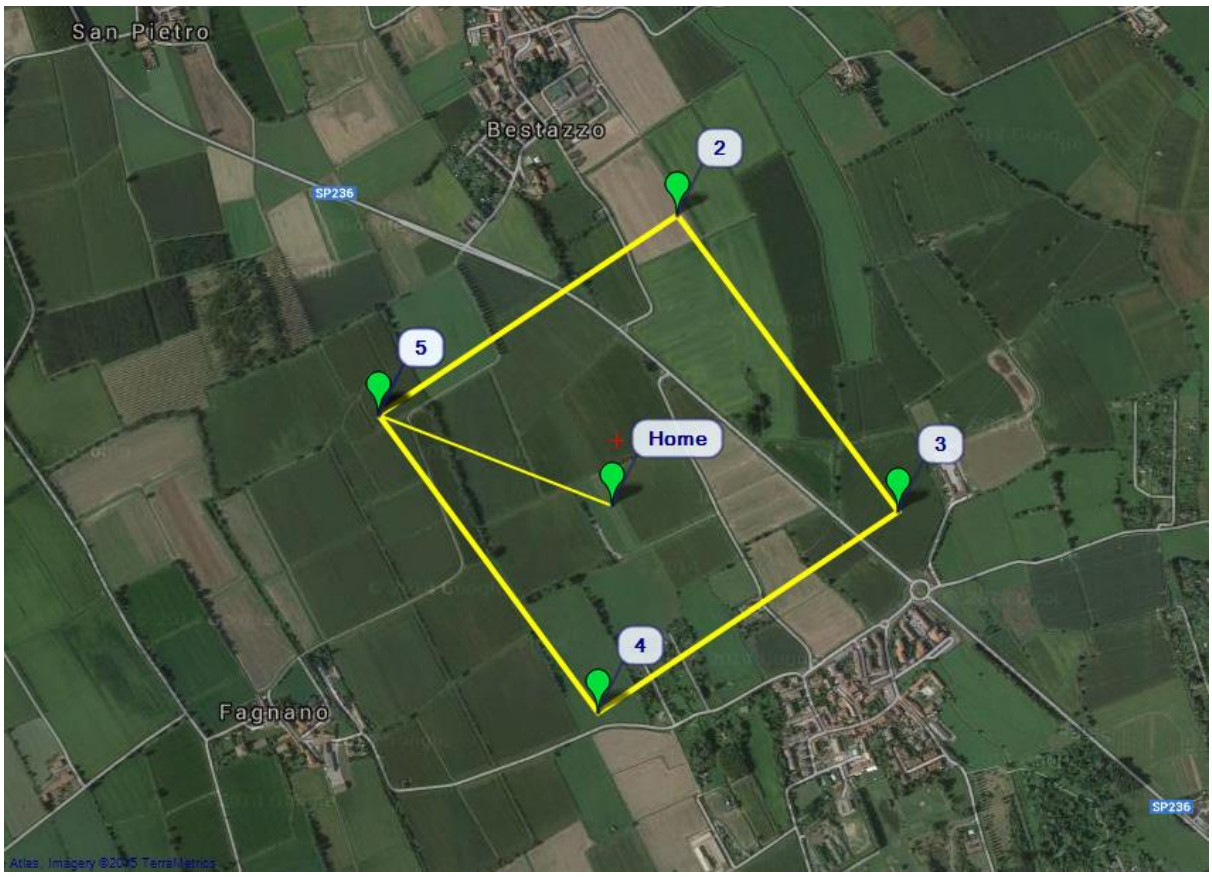


Figure 6.7 – Wind speed and direction resolution algorithm

6.4.2 Results

Following the two graphs as plotted by mission planner reporting the raw data collected from the autopilot, showing the heading of the UAV during the first square flight mission (mission A) and the difference between airspeed measured by the pitot tube and ground speed as seen by the GPS. These graphs plotted with the Mission Planner "Review a Log" tool, are actually quite useful to quickly evaluate the correct execution of the flight test and for the scale in use by default on the x-axis which refer to the line number of the stream of data recorded which is one line of data saved every 4 milliseconds. The appropriate data intervals could have been cut referring to these kind of graphs and saved as distinct matrices for further post-processing with a more professional software as Matlab.

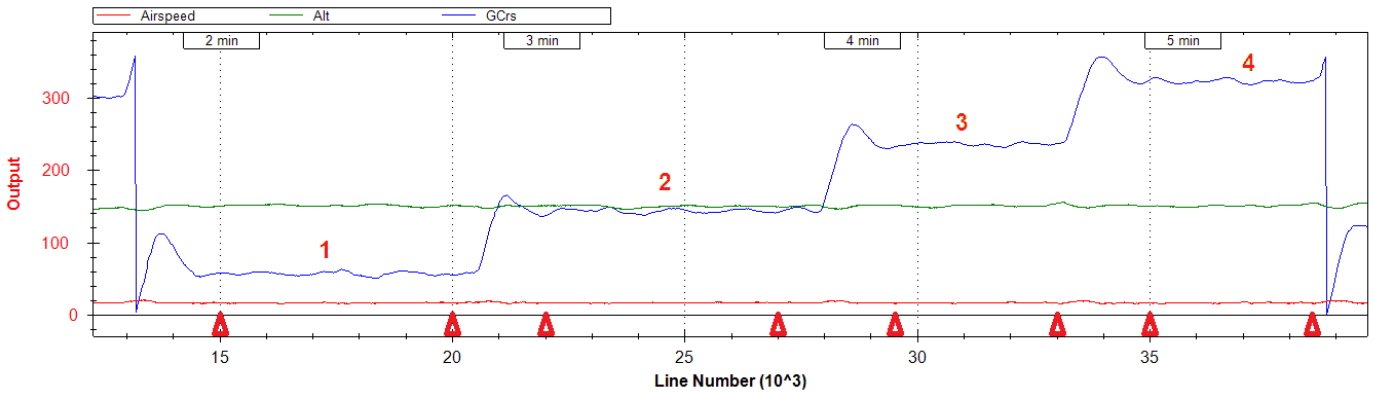


Figure 6.8 – Ground course

The graph above clearly shows the four legs of the mission that are the sides of the square path that has been flown at four constant heading differing from each other for about 90°. Only a little overshoot during the turns at the corners of the square can be blamed onto the autopilot but this 'off-track' values were not included into the intervals used to mediate the actual heading value of the leg (note the red pins on the bottom for the correct interval subdivision).

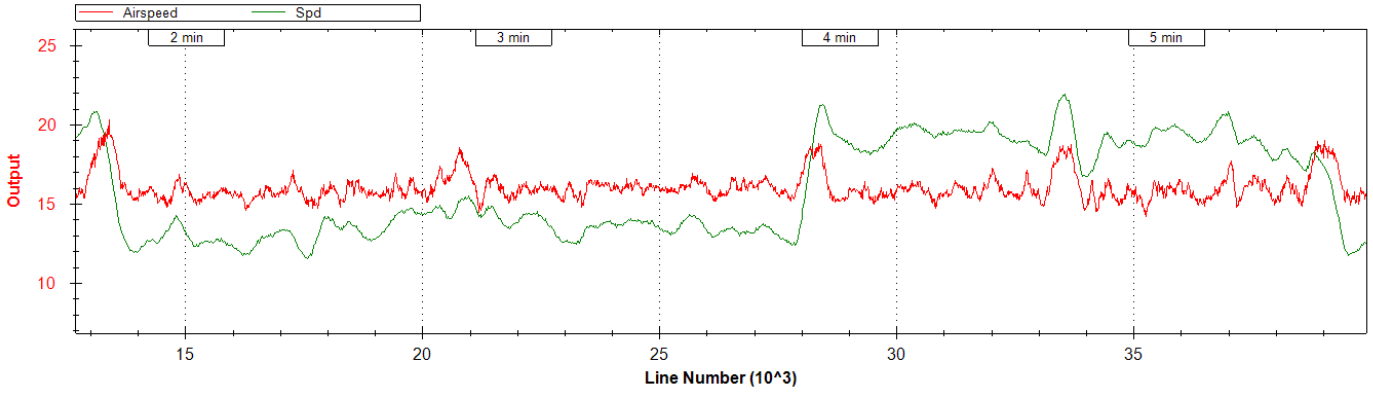


Figure 6.9 – Airspeed (red) vs. ground speed (blue)

The same intervals subdivisions have been taken to divide the velocities of the four legs and it's interesting to observe the difference between airspeed and ground speed differing from each other for wind speed in module and direction.

The quality of the data we collected (besides a small oscillation due to electrical signal disturbance not critical by taking the mean value of the interval) is confirmed by the fact that the IAS is constant during the four intervals and we have the major changes in the GS and the intersections between the twos in besides the intervals exactly when the autopilot changes the heading of the track. In the following table the means of the parameters calculated for each leg using Matlab:

TEST A	Time [s]	Altitude [m]	Delta Alt. [m]	Airspeed [m/s]	Gr. Speed [m/s]	Heading [deg]
Leg_1	43.60	150.77	5.43	15.68	13.06	56.31
Leg_2	43.45	149.66	6.09	15.96	13.53	143.21
Leg_3	26.16	150.37	6.29	15.86	19.45	236.26
Leg_4	26.06	150.00	3.56	15.80	19.28	323.37

Table 6.2 – Test A mean parameters

TEST B	Time [s]	Altitude [m]	Delta Alt. [m]	Airspeed [m/s]	Gr. Speed [m/s]	Heading [deg]
Leg_1	34.81	149.25	11.34	15.88	16.23	55.53
Leg_2	43.63	150.69	11.94	15.84	14.50	142.12
Leg_3	34.88	149.52	5.63	15.94	16.33	235.36
Leg_4	34.78	150.68	6.59	15.73	17.83	323.61

Table 6.3 – Test B mean parameters

Numerical results from the algorithm:

Algorithm Combinations	Wind Speed [m/s]	Wind Direction [deg]	True Air Speed [m/s]
A1	4.0247	176.1852	16.44
A2	4.0582	171.6617	16.65
A3	4.3053	171.5619	16.51
A4	4.3844	175.7844	16.74

Table 6.4 – Test A algorithm results

Algorithm Combinations	Wind Speed [m/s]	Wind Direction [deg]	True Air Speed [m/s]
B1	1.89	126.00	16.39
B2	1.67	133.01	16.17
B3	1.48	126.40	16.35
B4	1.68	119.19	16.17

Table 6.5 – Test B algorithm results

Final results mediate on the four combinations:

MEAN RESULTS	Wind Speed [m/s]	Wind Direction [deg]	True Air Speed [m/s]
h 10:38	4.19	173.80	16.59
h 16:26	1.68	126.15	16.27

Table 6.6 – Final mean results of the wind determination test

The final results coming from this method, are then compared with the data collected through the portable ground meteorological station during the whole day to verify their consistency:

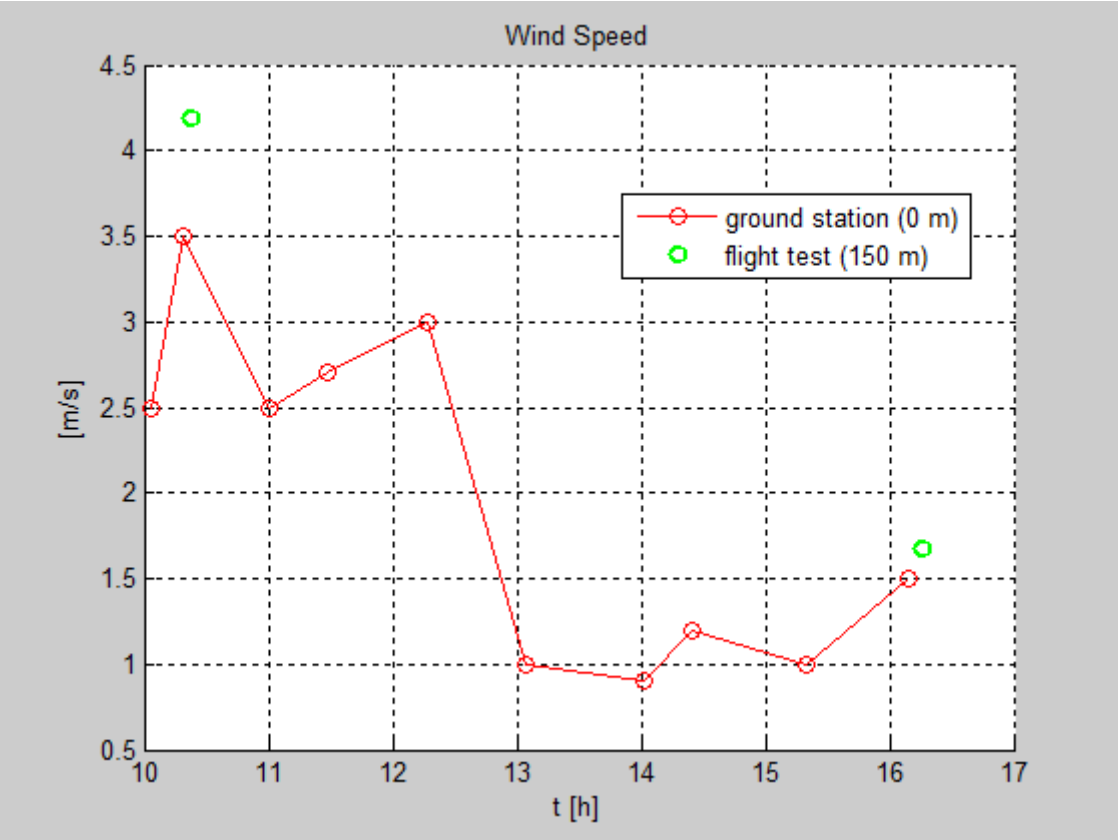


Figure 6.10 – Wind speed on ground vs. WS at flight altitude

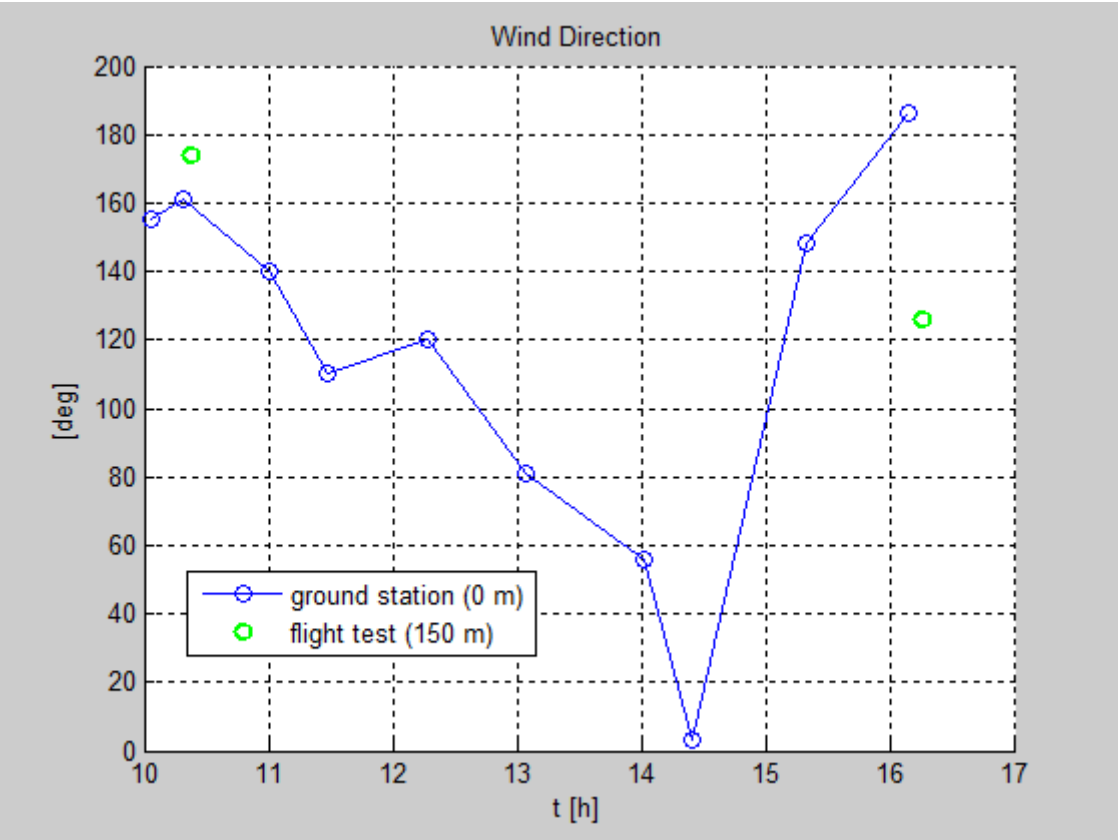


Figure 6.11 – Wind direction on ground vs. WD at flight altitude

Even if the data coming from the GMS might not perfectly give the punctual trend evolution of wind speed and direction due to the few registered test points (the annotation of these data had to be done manually with an interval of about one hour), it gives a fairly good idea of the very nonlinear variation of these two parameters during the day. However, the comparison between these data measured at the field and the two data points computed from the wind determination algorithm on the flying platform can be considered well consistent. The small bias between the two can be imputed to the difference of altitude; the wind can blow a bit stronger at 150 meters usually than on the ground where also we had obstacles close to the GMS like trees that can change the wind speed and direction reading.

As a last result, we report here the reconstruction, made from the GPS coordinates and altitude data plotted on a Google Earth map, of the path that was actually flown by the UAV autopilot in AUTO mode (green) for the "square mission A". The small overshoots also noted in the ground course graph, here are clearly visible on the corners of the "box". The yellow and blue parts are the recording of the MANUAL mode maneuvers for take-off, landing and the related circuits to gain and loose altitude:

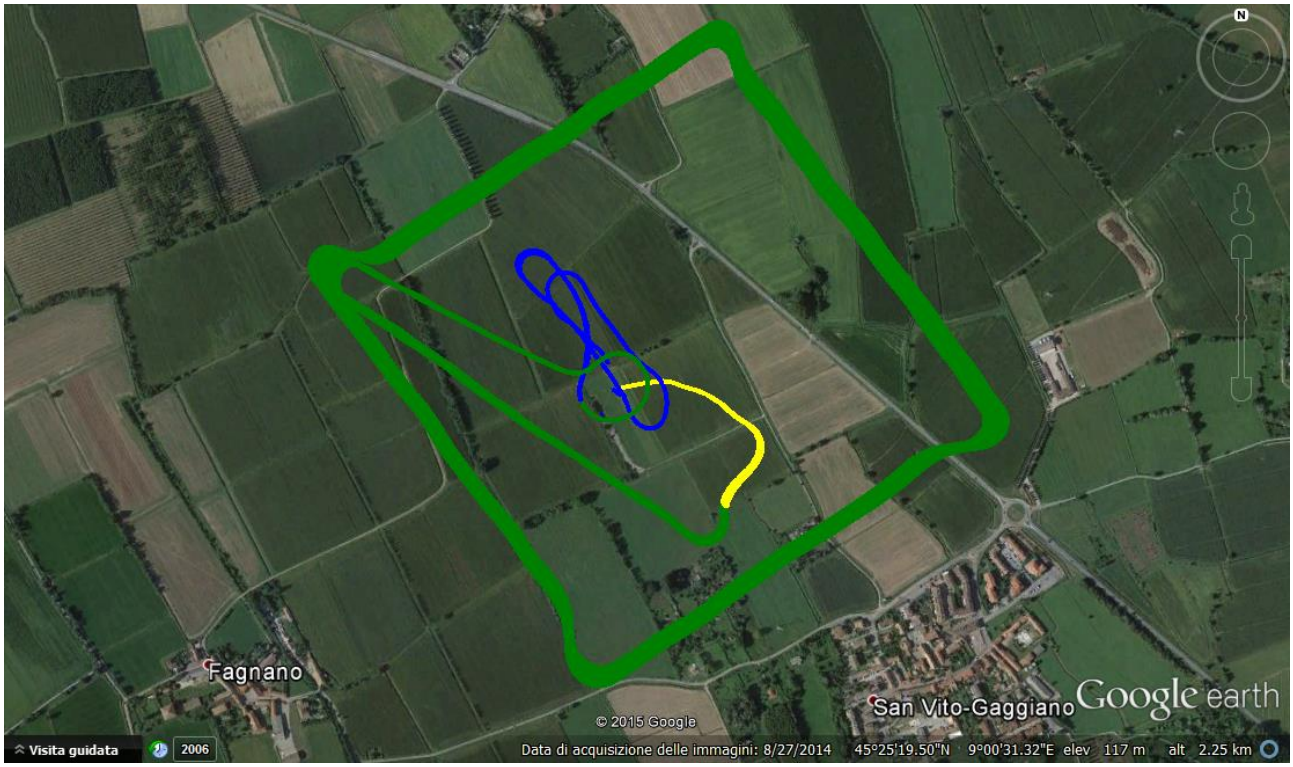


Figure 6.12 – Actual flight path flown by the APM

6.5 Trimmed flights

The performing of several flights in trim condition at different airspeeds is intended to collect the right data to produce the aerodynamic performance graphs such as C_L-alpha, C_D-alpha and the aircraft polar (C_L-C_D).

6.5.1 Speed-drag method

One way to determine the aerodynamic polar of an aircraft is the Speed-Drag method. This method is based on the fact that in an horizontal levelled uniform flight, the traction of the propeller is equal to the total drag of the airplane. Starting from the general equations governing the forces acting on an airplane along the drag and lift directions:

$$m\frac{dv}{dt} = Tcos\alpha_0 \ - D - mg\sin\gamma \tag{1}$$

$$mv\frac{d\gamma}{dt} = Tsin\alpha_0 + L - mg\,cos\gamma \tag{2}$$

Where m is the mass of the airplane, α_0 the angle of attack of the fuselage compared to the undisturbed asymptotic current direction and γ the climb angle as it can be seen in the figure below:

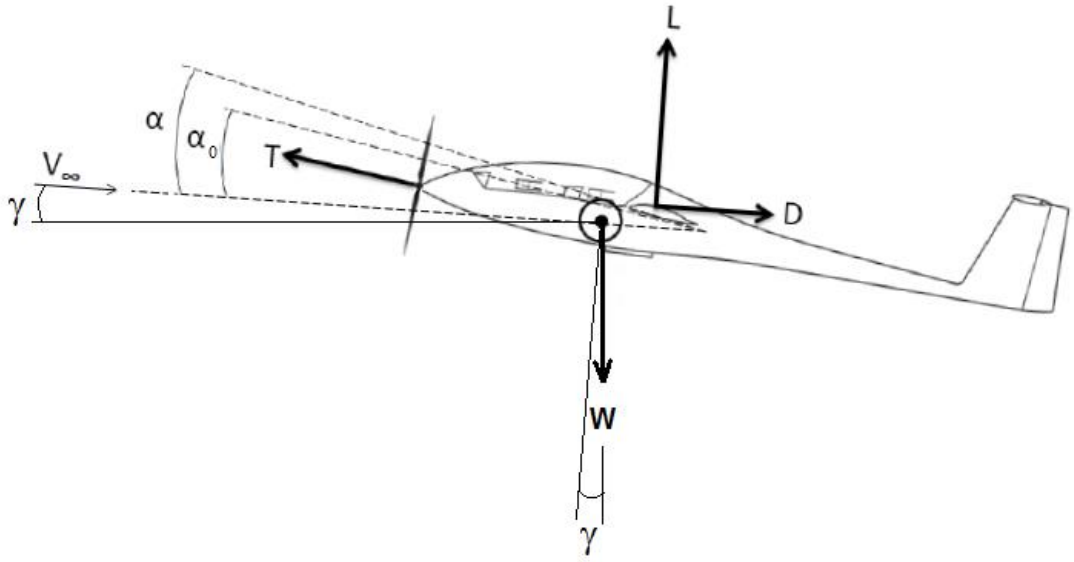


Figure 6.13 – Airplane in constant climbing attitude

Note that this is not yet an horizontal flight but the airplane is climbing following a trajectory at γ angle compared to the ground surface. The α_0 angle is the angle of attack of the fuselage which differ from the wing angle of attack by the c angle which is the difference between the chord line direction of the airfoil and the reference line of the fuselage.

$$\alpha_0 = \alpha - c$$

The pitch angle θ in this case is the sum of the α_0 and γ angles.

$$\vartheta = \alpha_0 + \gamma$$

If the flight is trimmed and uniform the accelerations are equal to zero and the equations (1) and (2) reduce to:

$$D = T cos\alpha_0 - mg \sin\gamma \quad (3)$$

$$L = mg\, cos\gamma - T sin\alpha_0 \quad (4)$$

If the flight is horizontal keeping the aircraft at the same altitude from the terrain, then the climb angle is equal to zero and the pitch is equal to the fuselage angle of attack as it can be seen in the picture below:

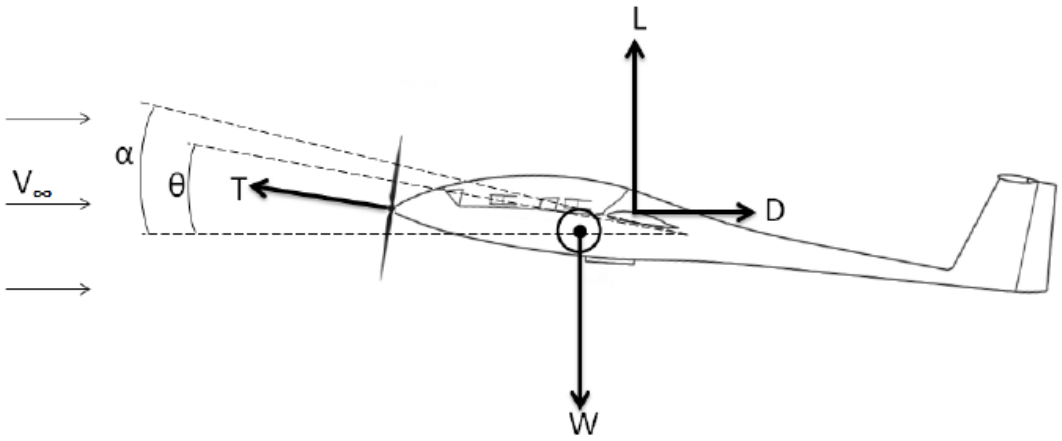


Figure 6.14 – Airplane in trimmed horizontal flight

And our equations again reduce to:

$$D = T cos\alpha_0 \quad (5)$$

$$L = mg - T sin\alpha_0 \quad (6)$$

Actually for small angles of attack, we can assume that D ≅ T and L ≅ W.
However, disponing of the indication coming from the AOA sensor (reduced by the *c* angle to refer to the fuselage reference line which is also the direction of the traction force), in the post processing code this small contribute coming from the traction vector misalignment compared to the drag force direction won't be neglected.

From the flight testing we recorded the data of a lot of different trimmed flights at the same altitude but at different constant airspeeds (at different throttle %) and angles of attack.

Then knowing all the characteristic of the aircraft as total weight, wing surface (*S*) and the propeller RPM (*n*) and coefficient of traction at different throttle % and airspeeds as determined in the wind tunnel tests, we could then calculate a couple of coefficients of lift and drag for each test point.

$$T = C_T \rho n^2 d^4$$

Inverting the *Rénard* formula (see section 4.3.7) as reported above and substituting the resulting trust force into equations (5) and (6), the two classic aerodynamic coefficients can be found:

$$C_L = \frac{L}{\frac{1}{2}\rho v^2 S}; \qquad C_D = \frac{D}{\frac{1}{2}\rho v^2 S}$$

Plotting all these couples of values for all the test points collected in the flight test and then making a quadratic fitting, it could be possible to create the experimental aerodynamic polar of the *Cularis.*

6.5.2 Trimmed mode

A specific flight mode called *TRIMMED* (number 17) has been implemented and integrated in the autopilot governing software code to assure fixed throttle percent at a constant altitude. This has been achieved by simply closing a customized navigation loop in altitude controlling the demanded pitch value as an altitude error times a K proportional gain.

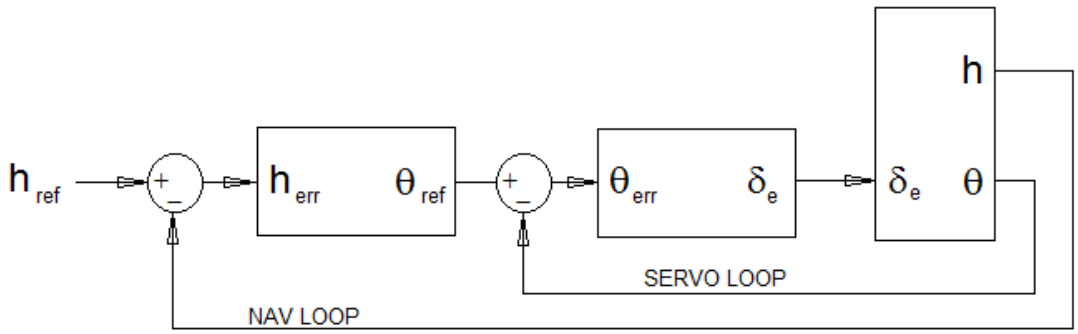


Figure 6.15 – Custom navigation loop for trimmed mode

Only the elevator will be used to control the altitude as the throttle for this mode is a fixed parameter that can only be changed manually (contrarily to what was happening for all the other default automatic modes) and that will only affect the horizontal flying speed. The wings are kept levelled by setting the roll reference value in the lateral-direction loop to be equal to zero. All the new variables we had to define (e.g. reference altitude, throttle percent and K-gain) for this mode have the possibility to be changed and adjusted through bi-directional in-flight telemetry from the GCS (see Appendix A for the actual code).

This mode guarantee the correct execution of one test point at the set throttle percent and altitude when the automatic mode *TRIMMED* is engaged through the transmitter switch. However, in order to perform more test points during one flight, the pilot has to follow some rules to bring the UAV in the correct starting position at the beginning of each new test point:

1. Climb in *MANUAL* mode until an altitude a little above the *FT_ALTITUDE* set value (120 m). This way when switching to the dedicated mode, the aircraft will take less time to adjust exactly at the desired altitude by descending at a greater ratio rather than by climbing and risking a stall right away when a low throttle percentage is being tested.

2. Position the heading against wind direction. By doing so the model, which is statically stable and very light so susceptible to wind, will naturally keep that direction fairly well with a resultant AOS close to zero throughout the whole test. We don't want to force the autopilot to keep a defined heading to avoid too large ailerons and rudder corrections that will result in a non-zero value of AOS. In any case, even if desirable for consistency with the real flight testing of manned aircrafts, the ground track doesn't have to be perfectly straight for our goals and keeping a close to zero value of AOS has been evaluated more important.

3. Switch to *TRIMMED* mode and from the point at which the transitory in altitude ends (vertical speed close to 0.0 m/s) and IAS remains still, hold the test point for about 20 to 30 seconds or until the end of safe visual contact (about 400 m from home point).

4. Switch back to *MANUAL* and go back to the starting point of the previous test segment to repeat the test point a second time or tell the person monitoring the GCS to update the *FT_THR* parameter with a different throttle percentage through telemetry while repositioning to perform another test point at a different speed.

5. Check again the requirements of altitude and heading in points (1) and (2) and start another test segment as in point (3).

6. When at the end of the required number of trimmed flight segments scheduled for one flight (before the battery charge gets too low), land the plane in *MANUAL* mode and change the battery pack and/or download the recorded data as needed.

The trimmed flight segments that have been flown present 13 different settings in terms of throttle percent: 100, 90, 85, 80, 75, 70, 65, 60, 55, 52, 50, 47 and 45 until the point when the defined throttle percent was too low to have a trimmed flight at a constant speed and altitude which resulted in a loss of altitude (non-zero but negative vertical speed) or in a stall. Setting the throttle parameter starting from 100% and then lowering it is convenient because this way more power is required at the beginning when the battery surely is full and able to provide enough power. Proceeding with the test points less power is needed.

Looking at the different settings adopted for the throttle % parameter, it can be noted that we decided to reduce the decrement step when close to the stall speed to have better and more spread out results especially for the C_L-alpha curve (e.g. 52 and 47 in between 55 and 45%). In fact, close to the minimum trust needed to sustain an horizontal flight, the airspeed and AOA values can change a lot in what is a short throttle but wide pitch attitude range.

Each flight test segment has been repeated at least twice for repeatability and accuracy considerations on the collected data in the post-processing.

6.5.3 Data reduction

One observation has to be made for the air density value and for the airspeed, which is also a function of the air density, used in the speed-drag method calculations of the C_L and C_D parameters.

The airspeed value measured through the airspeed sensor on board of the UAV, the indicated air speed (IAS), is actually an equivalent air speed (EAS) because the APM code relating the differential pressure to the speed considers as the air density its standard value at sea level and 15°C which is ρ_0 = 1.225 kg/m^3. To refer this value to the airspeed measured in the wind tunnel test (which is already corrected accounting for temperature, absolute pressure and relative humidity), we had in first place to make a data reduction in order to know the true airspeed (TAS) that was being measure during the tests using the on field air density.

$$V_{TAS} = V_{EAS}\sqrt{\frac{\rho_0}{\rho_{test}}} \tag{7}$$

The V_{EAS} can also be considered as a calibrated air speed (CAS) as it was already been corrected for instrument and position error in the wind tunnel tests. The V_{TAS} is the relative velocity between the aircraft and the surrounding air mass.

Because of the relatively low altitude of the flight tests (~ 120 m), the air density of the flight tests has been assumed equal to the air density on the ground. We calculated the humid air density at the field on the two different days when the trimmed flight tests were performed referring to the mean data of absolute pressure, temperature and relative humidity measured by the ground meteorological station and reported in table 1.4.

The density of humid air may be calculated as a mixture of ideal gases. In this case, the partial pressure of water vapor is known as the vapor pressure and the other partial pressure is the dry air. Using this method, error in the density calculation is less than 0.2% in the range of −10 °C to 50 °C [18].

The density of humid air is found by:

$$\rho_{test} = \frac{P_d}{R_d T} + \frac{P_v}{R_v T} \quad (8)$$

where,

R_d = specific gas constant for dry air, 287.058 J/(kg·K).
R_v = Specific gas constant for water vapor, 461.495 J/(kg·K)
T = Temperature (K)
P_d = partial pressure of dry air (Pa)
P_v = partial pressure of water vapor (Pa)

The partial vapor pressure of water may be calculated from the saturation vapor pressure and relative humidity. It is found by:

$$P_v = RH \cdot P_{sat}$$

where,

RH = relative humidity (%)
P_{sat} = saturation vapor pressure (hPa)

The saturation vapor pressure of water at any given temperature is the vapor pressure when relative humidity is 100%. One formula [19] used to find the saturation vapor pressure is:

$$P_{sat} = 6.1078 \cdot 10^{\frac{7.5\,T}{T+237.3}}$$

where T is in °C and the resultant P_{sat} value will be expressed in hPa.

The partial pressure of dry air is then simply found by subtracting from the observed absolute pressure the partial pressure of the water vapor, resulting in:

$$P_d = P_{abs} - P_v$$

Where also this value is expressed in hPa and should be converted in the right measure units, together with the partial vapor pressure and temperature values, before being substituted in equation (8). After getting the two air density values for the two different days of trimmed flight tests, the V_{TAS} of all the test points can be finally computed using equation (7) being careful to refer the right data to the right air density.

6.5.4 Results

Starting with showing graphically how one flight session of this test typically looks like, in the following figure are clearly visible in different colors the 4 test segments that have been performed. Each one of these segments, after post processing, will be representing one test point on the final graph.

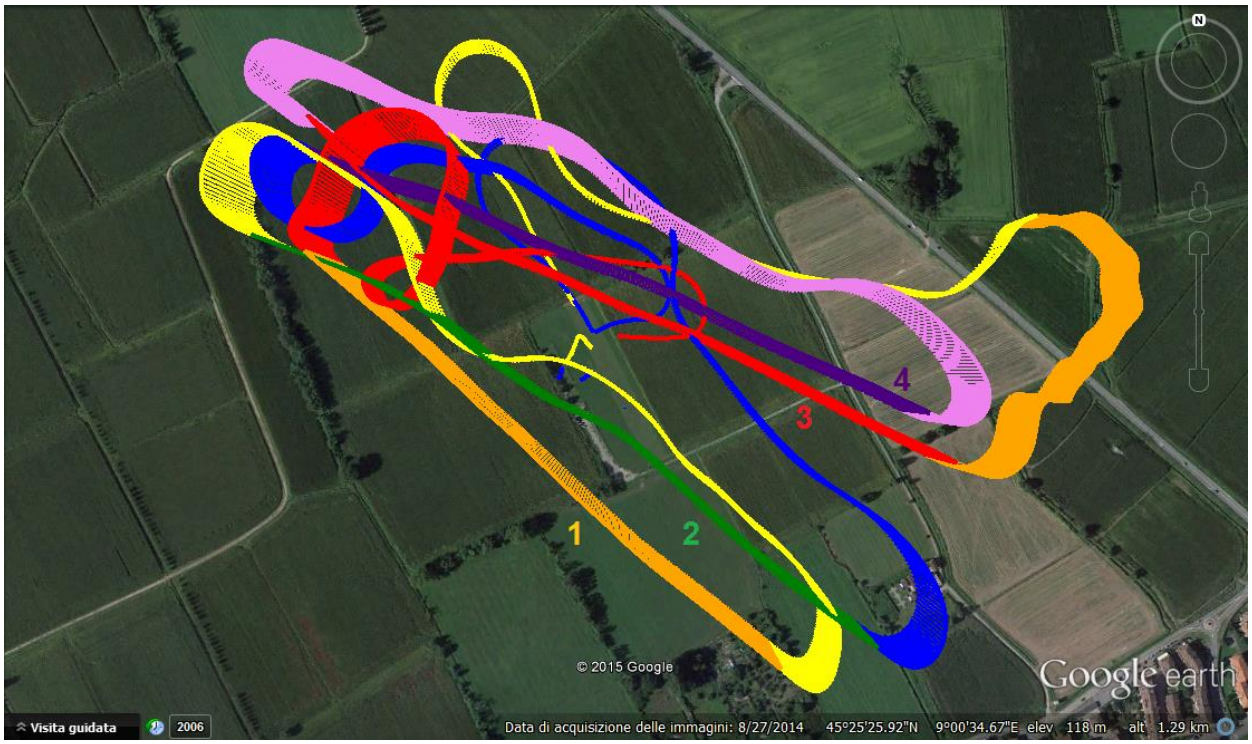


Figure 6.16 – Google Earth visualization of a typical trimmed flights session

To actually get to know the exact start and end of the test segments in order to be cut and mediate in the post processing, again the quick and easy Mission Plane review a log tool has been used. In the following figure, as an example, the two segments flown at 85 % throttle are showed. They can be recognized by the constant throttle output (in purple).

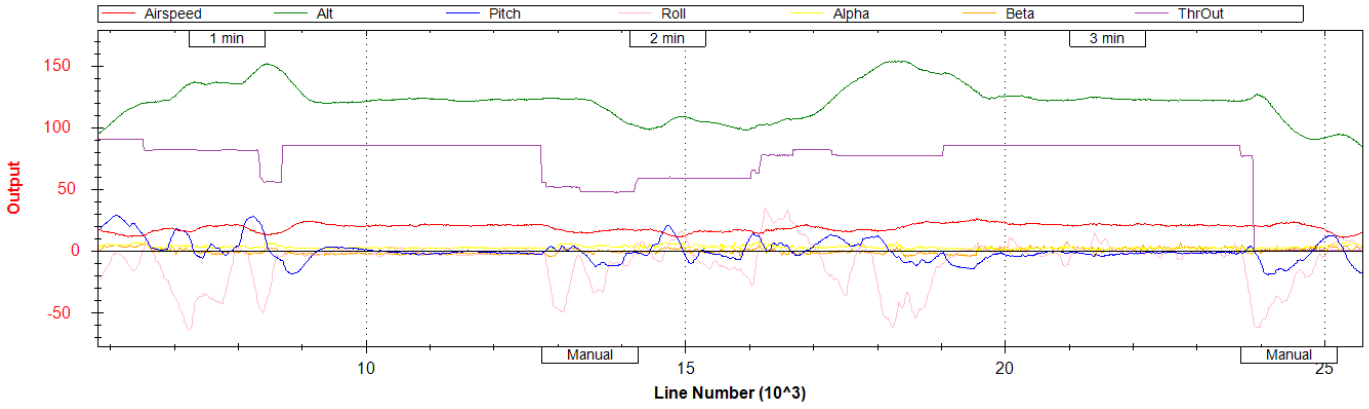


Figure 6.17 – Example of two test segments at the same throttle %

Zooming in on the first segment, the actual interval of useful data that was cut and stored for post processing is highlighted by the red lines on the sides. Inside this interval is immediate to verify how the parameters relevant to the test as altitude (green) and airspeed (red) are qualitatively constant as expected.

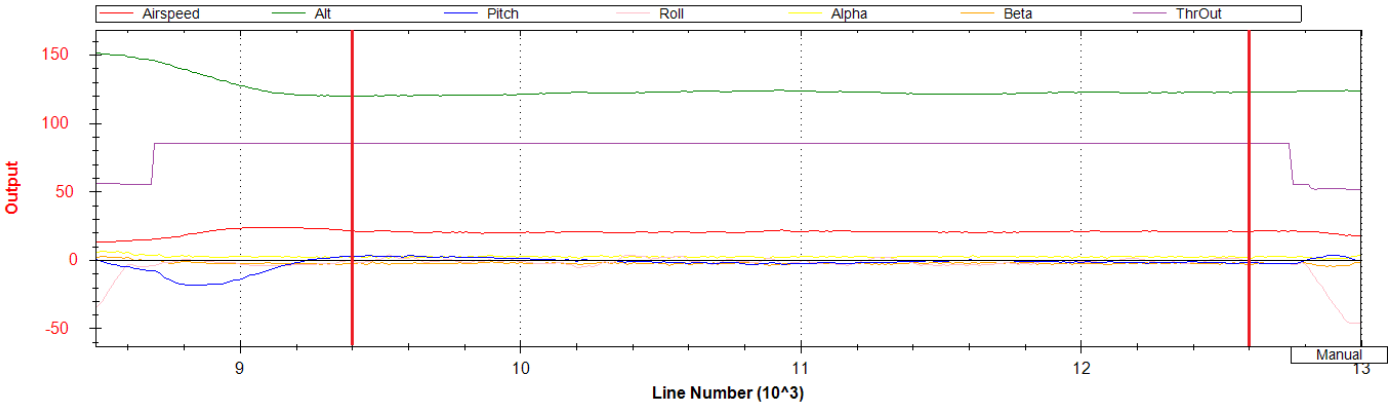


Figure 6.18 – Example of choosing the right interval of the test segment

This same procedure of preliminary selection and validation of the test segments has been repeated for all the registered data. Using Matlab we then calculated the means of all relevant parameters and performed the proper reductions to quantitatively make a second selection and validation of the results. In the following five tables are listed all the numerical results for each calculated parameter:

COLLECTED DATA after post processing:

Thr % (test)	100 A	100 B	90 A	90 B	85 A	85 B
dt [s]	13.128	21.85	21.75	20.046	24.459	26.162
h [m]	122.94	121.1048	122.0656	122.6554	122.0053	122.3293
dh [m]	4.762	5.3513	6.4566	4.49	4.1007	3.338
TAS [m/s]	22.0333	21.2323	21.0306	21.1182	20.7848	21.2576
α [deg]	1.9941	2.1189	2.2443	2.2486	2.2109	2.0352
β [deg]	-2.8602	-2.668	-2.2806	-2.5097	-2.5898	1.8879
roll [deg]	-2.8741	-1.8671	-1.7777	-1.9375	-1.3493	-1.2179
pitch [deg]	-1.8777	-0.48105	-0.94206	-2.3212	-1.2232	-2.1665
d_pitch [deg]	5.74	6.82	7.37	6.09	5.67	4.62
head [deg]	109.7921	125.8327	95.3344	107.195	131.6356	126.5758
d_head [deg]	7.81	22.75	7.47	8.05	7.86	20.76
rpm	7105.7296	7009.2281	6774.5642	6786.1122	6634.982	6700.8268
C_T	0.012171	0.015187	0.011729	0.011403	0.010369	0.0086405
T [N]	2.1154	2.5683	1.853	1.8077	1.5713	1.3355
C_D	0.01691	0.02211	0.016261	0.015731	0.014116	0.01147
C_L	0.16405	0.17669	0.17996	0.17846	0.18422	0.17612

Table 6.7 – Trimmed flights result, table 1

Thr % (test)	80 A	80 B	75 A	75 B	70 A	70 B
dt [s]	22.653	26.186	34.88	36.584	43.501	41.898
h [m]	121.813	121.6024	120.4765	120.045	118.2354	118.5193
dh [m]	5.4246	3.713	5.7301	9.9294	5.4012	5.1761
TAS [m/s]	20.378	20.3285	18.1475	17.5041	16.2988	16.4244
α [deg]	2.1955	2.26	2.743	2.9686	3.3255	3.2401
β [deg]	-2.2231	2.1567	-1.787	-1.5255	-1.0259	-1.1521
roll [deg]	-1.2403	-0.8263	-0.80172	-1.0775	-0.61287	-0.43081
pitch [deg]	-1.3956	-1.2831	0.99734	1.6847	3.0226	2.363
d_pitch [deg]	5.62	4.09	7.5	10.88	5.94	6.13
head [deg]	113.9539	115.0231	196.2387	213.6506	211.4623	212.8642
d_head [deg]	11.35	9.38	23.76	24.76	25.55	9.74
rpm	6469.0273	6461.9992	6043.651	5960.2214	5669.9148	5685.7319
C_T	0.0094517	0.0096319	0.015113	0.017603	0.019377	0.018873
T [N]	1.3616	1.3845	1.9285	2.1847	2.1762	2.1315
C_D	0.012725	0.013003	0.022395	0.02727	0.03133	0.030219
C_L	0.19162	0.19254	0.23796	0.25568	0.2947	0.29025

Table 6.8 – Trimmed flights result, table 2

Thr % (test)	65 A	60 A	60 B	55 A	55 B	52 A
dt [s]	34.907	36.609	27.894	34.907	26.186	26.164
h [m]	121.3912	118.241	118.2105	124.0526	119.644	120.206
dh [m]	4.8383	7.2846	10.6295	8.8396	4.002	6.9483
TAS [m/s]	18.1415	16.2932	16.0075	15.7734	13.6071	15.1657
α [deg]	3.0387	3.2539	3.581	3.9989	5.519	4.393
β [deg]	2.7918	-2.2806	-0.67614	-0.11207	3.4324	1.4749
roll [deg]	-1.426	-0.51921	-1.3065	-0.76974	-1.1981	-1.2709
pitch [deg]	-1.9178	2.038	1.7626	-5.9585	-1.2699	-2.3262
d_pitch [deg]	4.98	9.28	10.23	8.77	5.06	7.03
head [deg]	89.6571	176.9132	123.2009	85.8992	83.5193	133.308
d_head [deg]	54.82	18.87	33.13	50.54	65.05	44.91
rpm	5786.1611	5383.396	5343.3889	5164.0874	4868.821	4986.425
C_T	0.0091242	0.012625	0.013733	0.01085	0.019661	0.010772
T [N]	1.0515	1.2782	1.3699	0.99604	1.6044	0.92196
C_D	0.012401	0.018415	0.020445	0.015536	0.033601	0.015554
C_L	0.24155	0.29499	0.3055	0.31926	0.42789	0.34527

Table 6.9 – Trimmed flights result, table 3

Thr % (test)	52 B	50 A	50 B	47 A	47 B	47 C
dt [s]	26.16	34.807	50.643	17.441	12.227	10.424
h [m]	116.6366	119.8401	117.1053	112.4399	112.3026	112.0679
dh [m]	6.9477	13.4011	13.2961	10.3808	3.7056	5.9152
TAS [m/s]	11.9814	13.9578	13.2742	10.3495	10.4151	10.4692
α [deg]	7.2072	5.0186	5.8693	10.8599	10.0364	9.8809
β [deg]	4.3319	1.3162	2.4886	8.9115	8.1844	7.9658
roll [deg]	-0.17321	-1.371	-1.3613	-1.8167	-1.1257	-2.007
pitch [deg]	2.0803	-0.15299	1.931	6.0658	5.1163	5.9517
d_pitch [deg]	6.4	14.54	15.65	6.82	5.06	6.35
head [deg]	122.0895	121.4773	136.8683	90.0677	68.4941	119.9161
d_head [deg]	30.38	79.62	127.26	51.91	18.64	31.62
rpm	4565.562	4752.7771	4660.0475	4191.3146	4198.84	4205.0826
C_T	0.024299	0.013893	0.016764	0.027353	0.027032	0.026767
T [N]	1.7435	1.0963	1.2718	1.6541	1.6405	1.6293
C_D	0.047015	0.021509	0.027569	0.059377	0.05826	0.057282
C_L	0.55034	0.40133	0.44318	0.73401	0.7257	0.71841

Table 6.10 – Trimmed flights result, table 4

Thr % (test)	47 D	45 A	45 B	45 C	45 D	-
dt [s]	10.525	13.931	10.524	27.967	24.356	-
h [m]	112.3757	110.5699	111.6126	115.4622	116.0721	-
dh [m]	2.9457	7.4416	11.0318	4.7298	8.1521	-
TAS [m/s]	10.2768	10.9728	11.5174	11.0126	11.559	-
α [deg]	10.3248	8.4061	8.2592	8.7601	8.1094	-
β [deg]	8.5643	3.5999	3.3242	3.7775	3.1221	-
roll [deg]	-1.4987	-1.3719	-1.8325	-1.3569	-1.3408	-
pitch [deg]	4.6597	7.9253	6.5428	4.0331	2.51	-
d_pitch [deg]	3.51	8.95	9.42	8.42	14.36	-
head [deg]	104.0051	163.0859	233.5828	129.9369	158.9984	-
d_head [deg]	24.58	55.6	46.33	96.53	61.3	-
rpm	4183.0172	4190.2764	4258.2003	4195.1553	4263.4735	-
C_T	0.02771	0.022351	0.019808	0.022163	0.019617	-
T [N]	1.6691	1.371	1.2547	1.3626	1.2457	-
C_D	0.06084	0.043355	0.036024	0.042754	0.035517	-
C_L	0.74493	0.64649	0.58721	0.64159	0.58311	-

Table 6.11 – Trimmed flights result, table 5

Important notes on the experimental result:

1. All the airspeed (TAS), α, rpm, C_T, trust (T), C_D and C_L data included in the tables 6.7 to 6.11 were used in the post processing algorithms to produce the final graphs.

2. The test point 65 B is not reported because was badly conditioned and rejected. For the test points relative to 47 and 45 throttle we included 4 repetitions each instead of two to have a better statistically determination of the aerodynamics parameters close to stall condition.

3. The time intervals of the useful test segments are all above 10 seconds (minimum accepted value for considering valid a test segment); most of them are in between 20 and 30 seconds, one reaches 50 seconds. The duration is not standard and may vary according to the pilot clear or unclear viewing of the UAV during flight. The duration of the time intervals is shorter especially for the last test segments characterized by airspeeds close to stall speed, due to the difficulty of the autopilot to keep very unstable attitudes at very slow speeds easily affected by external stochastic disturbances as wind gusts resulting in a rough disturbance of the test segment.

4. The target altitude of all test segments (120 m) is everywhere respected within 10 meters. It starts off a bit higher close to the maximum throttle when the autopilot has to pitch down to win the aircraft tendency to gain altitude and logically becomes more sensibly lower at the end when the throttle becomes barely enough to sustain an horizontal flight.

5. With the throttle decreasing, the airspeed decreases as well while the AOA increases quite linearly as expected.

6. Roll is close to zero in all test segments (within 3 degrees) but always slightly negative most likely indicating the tendency of the aircraft to turn left caused by the torque of the pulling spinner. Pitch is badly conditioned most likely caused by the thermal ascending and descending currents typical of the atmospheric instability during the spring season and even stronger on a light aircraft as a model glider but the overall trend shows the tendency to increase as for the angle of attack.

7. The heading is kept quite constant when the airspeed is high (until the throttle is above 70%) with a bias around a maximum value of 20 degrees (see figure 6.16) then starts to diverge for lower airspeeds due to attitude instability and natural tendency of the airplane to roll left (see previous point).

8. The angle of sideslip (β) excursions are limited within 5 degrees in all test segments.

Following are reported the aerodynamic performance graphs C_L-alpha, C_D-alpha and the aircraft polar (C_L-C_D). The data have been interpolated using the Matlab *CFTool* and compared with the results from the wind test tunnel for the wing:

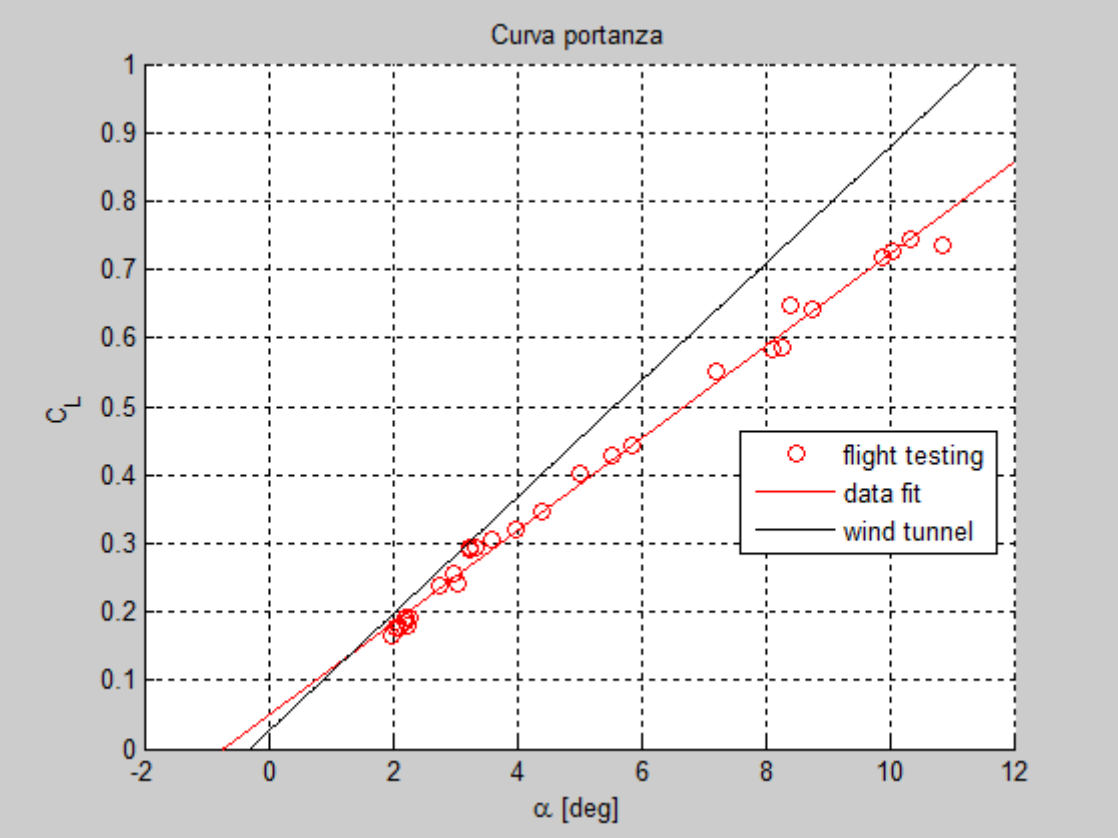


Figure 6.19 – Flight testing coefficient of lift curve

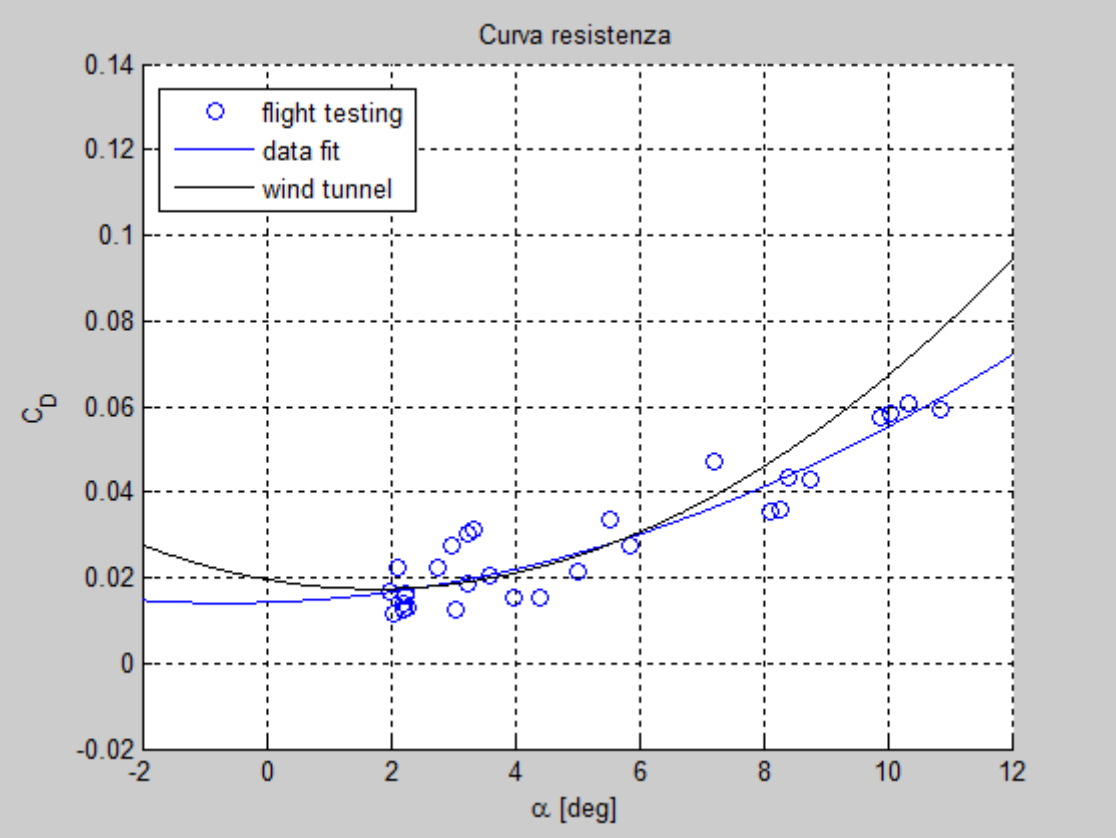


Figure 6.20 – Flight testing coefficient of drag curve

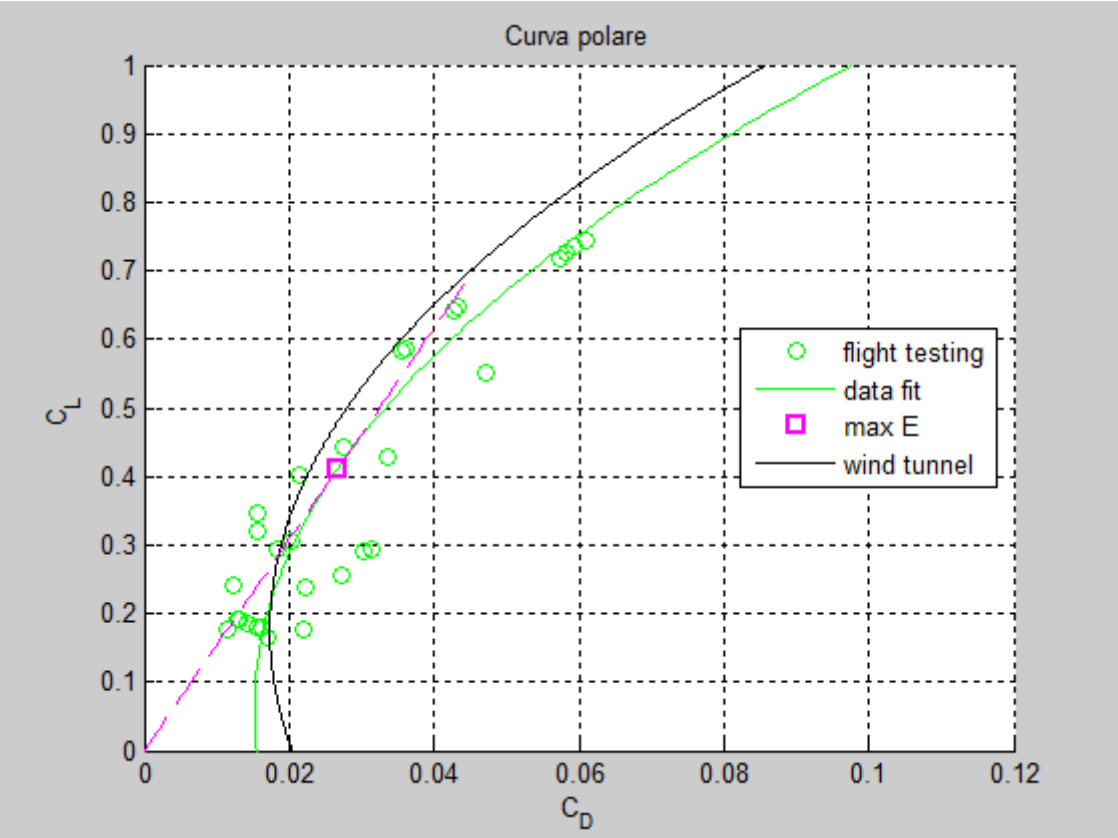


Figure 6.21 – Flight testing coefficient of drag curve

Same considerations can be made on the results at this point:

- The data points show a very good linear trend for the coefficient of lift curve with little dispersion; for the coefficient of drag and polar curves the dispersion worsen a little close to smaller angles of attack where the traction and drag values are very close but the classic parabolic trend it can be easily recognized on the overall results.

- For the coefficient of lift it was expected that the flight tests curve would have been lower than the wind tunnel tests curve. In the wind tunnel in fact only the influence of the wing has been tested while in the real flight test also the fuselage and the tail in particular gave their contribute the overall coefficient of lift of the plane.

- The coefficient of drag curve was expected to be lower for the wind tunnel tests curve than for the flight tests curve how it slightly is around 4 degrees of AOA. However, this difference is really small and going towards higher angles of attack the tendency of the wind tunnel tests curve to get higher outgrows the flight tests result. This anomaly might be attributed (as we already cited in section 4.3.2) to the difficulty of getting a precise measurement of drag using a balance, especially for small values cause by aerodynamic bodies, for the presence of disturbances like the intrinsic turbulence of the wake, the blockage effect of the size of the wing compared to the test chamber section and the wall effect of a closed test chamber that can't be avoided.

- In figure 6.21 the resultant aerodynamic polar curves are presented. The comparison shows how starting from the resultant values of C_L and C_D the overall aerodynamic efficiency is anyways higher for the wind tunnel model than for the complete airplane as is comprehensible due to the worsening presence of fuselage and tail in the real case. To this difference it can also be attributed the difference of minimum C_D which occurs for the two curves at slightly different values of C_L.

Following the numerical results in terms of efficiency for the flight tests polar curve:

Maximum efficiency = 15.3964 (Wind tunnel E_{max} = 17.2691)
C_L @ max E = 0.41
C_D @ max E = 0.02663

In order to compare the actual results with the results obtained by a previous thesis work on the same flying platform[10], another method to define the aerodynamic polar, other than simply making a quadratic fitting of the C_L and C_D data couples, has been followed.

In this method are firstly plotted on a preliminary graph the different points determined by the C_L^2-C_D couples. These points are then interpolated using a linear fit in order to define the characteristic parameters C_{D0} and k which are, respectively, the minimum drag coefficient and the slope of the linear fit:

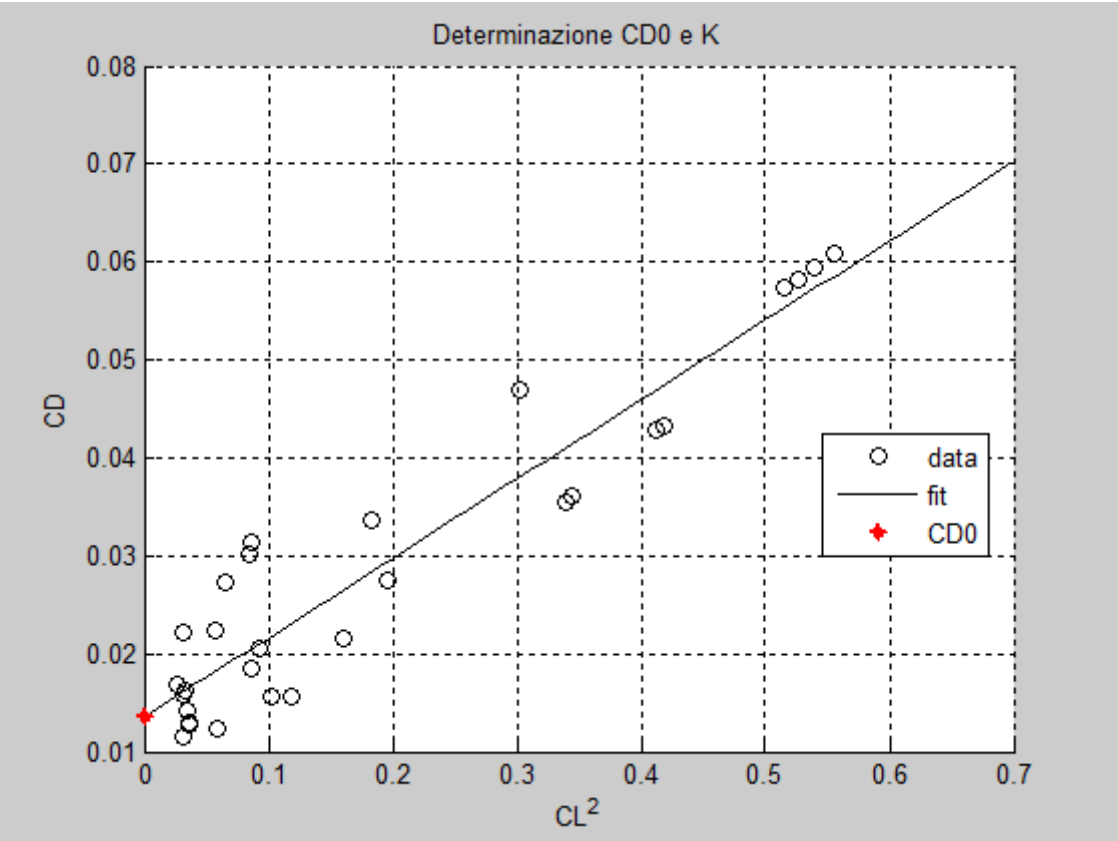


Figure 6.22 – Graph to determine C_{D0} and k

These two data will then be used to plot analytically the parabolic approximation of the polar curve by means of the equation:

$$C_D = C_{D0} + k\, C_L^2$$

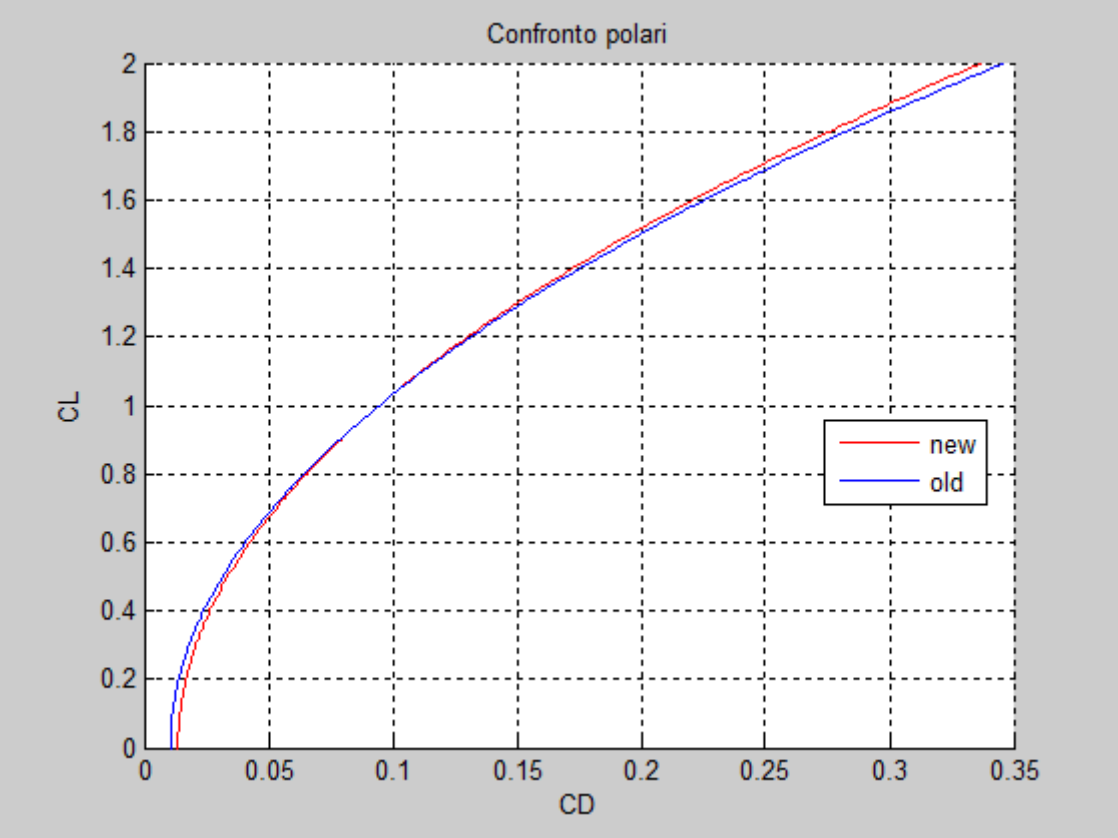


Figure 6.23 – Comparison among new and old polar curves

Project	***Method***	C_{D0}	***k***	E_{max}
Present	Speed-drag	0.0135	0.081	15.1
Previous	Speed-drag	0.0105	0.084	16.9

Table 6.12 – Comparison among coefficients relative to present and previous work

As it can be seen in the graph and table above, the two curves are very similar demonstrating the success of the test and its repeatability.

One observation can be made on the numerical value of the minimum coefficient of drag C_{D0} which for the new curve is slightly higher likely due to the two new booms installed on the sub-wings support to house the new AOA/S sensors and pitot tube that surely increased the parasite drag and also caused the efficiency to get a bit lower. The maximum efficiency values calculated for the polar in figure 6.21 and the one in figure 6.23 are slightly different due to the difference in method used to plot the data and their relative interpolation.

6.6 Stall test

In this test, a very important value for the certification of an aircraft such as the stall speed will be determined for the in-use flying platform in three different flying configurations.

6.6.1 Requirements for stall determination

The stall speed (V_S) specified in every aircraft flight manual is always the indicated air speed (IAS). This speed, in true airspeed terms, vary considerably depending upon density altitude. However, at typical civilian operating speeds, the aircraft's aerodynamic structure responds to dynamic pressure alone, and the aircraft will perform the same when at the same dynamic pressure. Since it is this same dynamic pressure that drives the airspeed indicator, an aircraft will always, for example, stall at the published indicated airspeed (for the same configuration) regardless of density altitude or true airspeed. So all the results in terms of airspeed collected by our airspeed sensor won't have to undergo any conceptual reduction (other than a simple filtering as will presented in section 6.6.3).

In addition to what have been stated above, following the guidelines for civil aircraft flight testing procedures CS-23/FAR-23[12], the tests for the stall speed determination have to undergo the following rules:

1. Starting condition of the test has to be a trimmed flight of airspeed equal to approximately $V_{TRIM} = 1.5\ V_S$ before the throttle is set to idle (zero) and the actual stall test begins.

2. The stall speed will be identified as the speed at the point where a clear and rapid change of pitch attitude and/or roll attitude will be observed with a consequential loss of governability for a few seconds

3. The deceleration during the nose up maneuver that will follow the throttle cut off to try to keep altitude, which is the stall entry rate, has to be computed by reporting the angular coefficient of the segment connecting the $1.1\ V_S$ to the final V_S value (see figure 6.24).

4. The standard entry rate value is set to -1 knot/s and the standard stall speeds together with the standard stall C_L values will be computed referring to this specific deceleration (see figure 6.25).

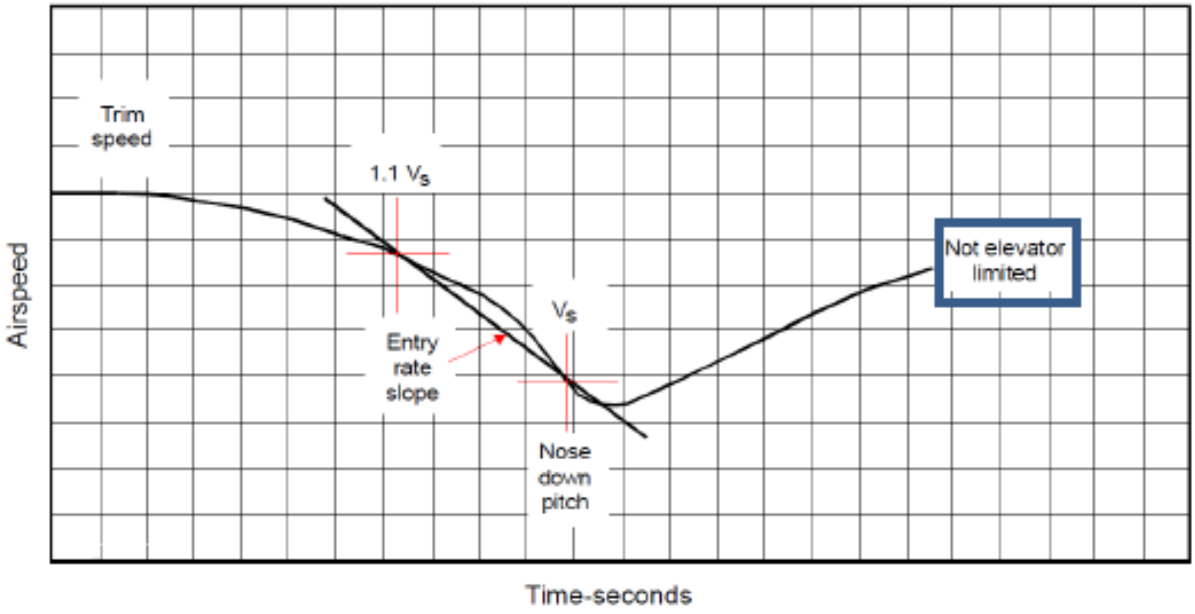


Figure 6.24 – Stall entry rate detection on the airspeed time history

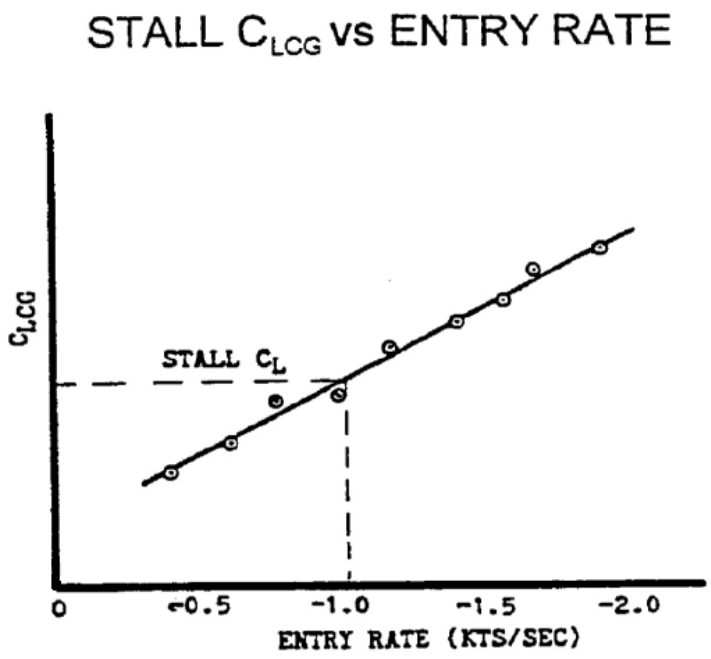


Figure 6.25 – Determination of stall C_L as a function of entry rate

6.6.2 Stall mode

To implement this mode, (mode 18 named *STALL*) we used the same code as for the trimmed mode but with the throttle percent set to zero (engine turned off, aircraft in pure glider configuration) and without the maximum pitch limitation (or better with the limitation set to the highest suggested value of 45 degrees) to actually force the airplane to stall (see Appendix A for the code).

What will happen is that even without any available power, the autopilot control logic will still try to keep altitude, but progressively it will end losing it which will cause the pitch to progressively increase together with the AOA while the airspeed will dramatically drop until the stall speed is reached.

The start of this test is the same as for the trimmed flights so follow points (1), (2) and (3) described for the *TRIMMED* mode in section 6.5.2 to take off, gain altitude, select the heading and level the plane at 1.5 the stall speed before activating the *STALL* mode (an estimation of 10 m/s has been considered as initial stall speed guess referring to the trimmed flight tests resulting in a start off trimmed condition speed of 15 m/s achievable by setting the throttle parameter to 60%) . As a safety precaution, the start altitude for the stall test has been set a bit higher (150 m) than the reference altitude set for the trimmed tests (120 m). After the stall is actually verified observing a quick change in pitch and (eventually) roll the pilot has to immediately switch in MANUAL mode to regain control of the plane by giving throttle and stabilizing its attitude.

The test has to be performed for each different flying configuration which means for cruise "clean" configuration, for "endurance" configuration and for "landing" configuration, respectively with flaps extended at three different deflection angles of 0, 8 and 16 degrees. We remember that the aircraft we are testing doesn't dispose of landing gears and with the throttle set to 0% the spinner automatically folds along the fuselage sides.

The test has to be repeated several times for a minimum 5 repetitions for each configuration in order to mediate the results and get a good estimation of the three characteristic stall speeds.

6.6.3 Results

In the following graphs are presented as an example three typical time histories, one for each flying configuration. This type of graphs represent the trends of airspeed, angle of attach, roll and pitch parameters right before (about 5 seconds), during and immediately after the stall points (2 seconds):

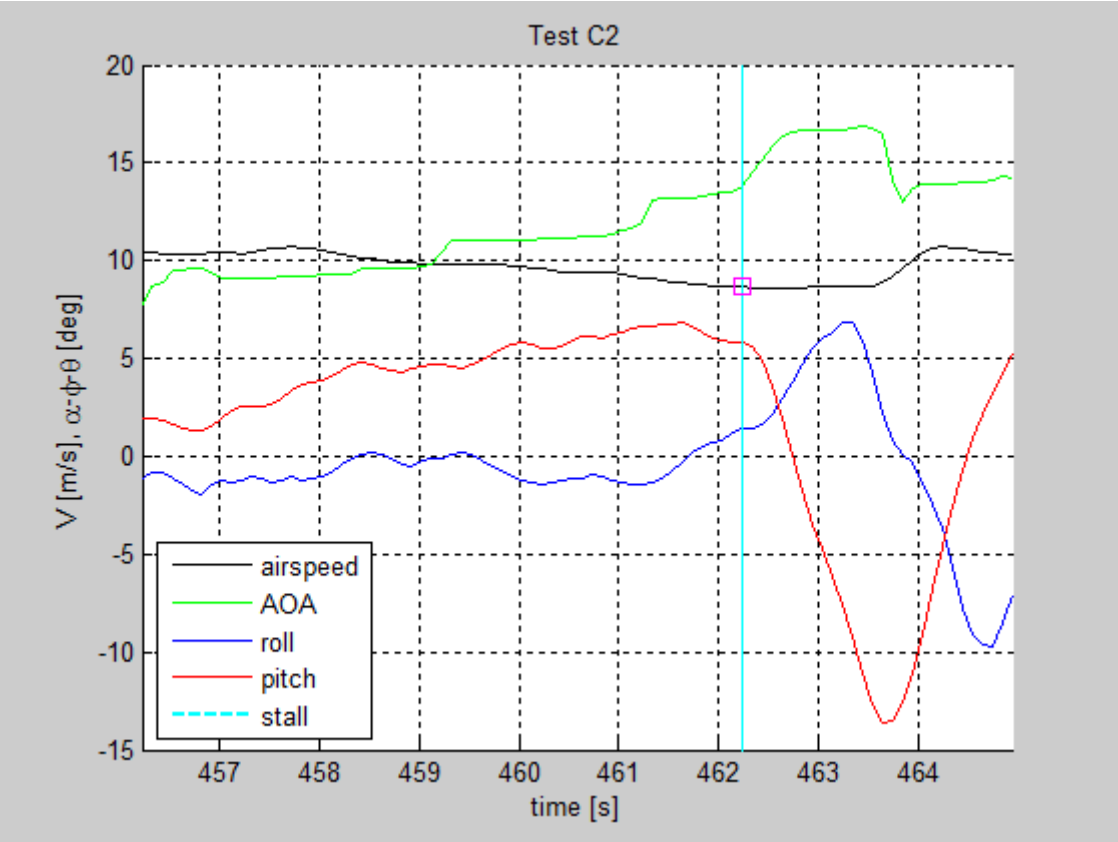


Figure 6.26 – Time history of a typical clean configuration stall

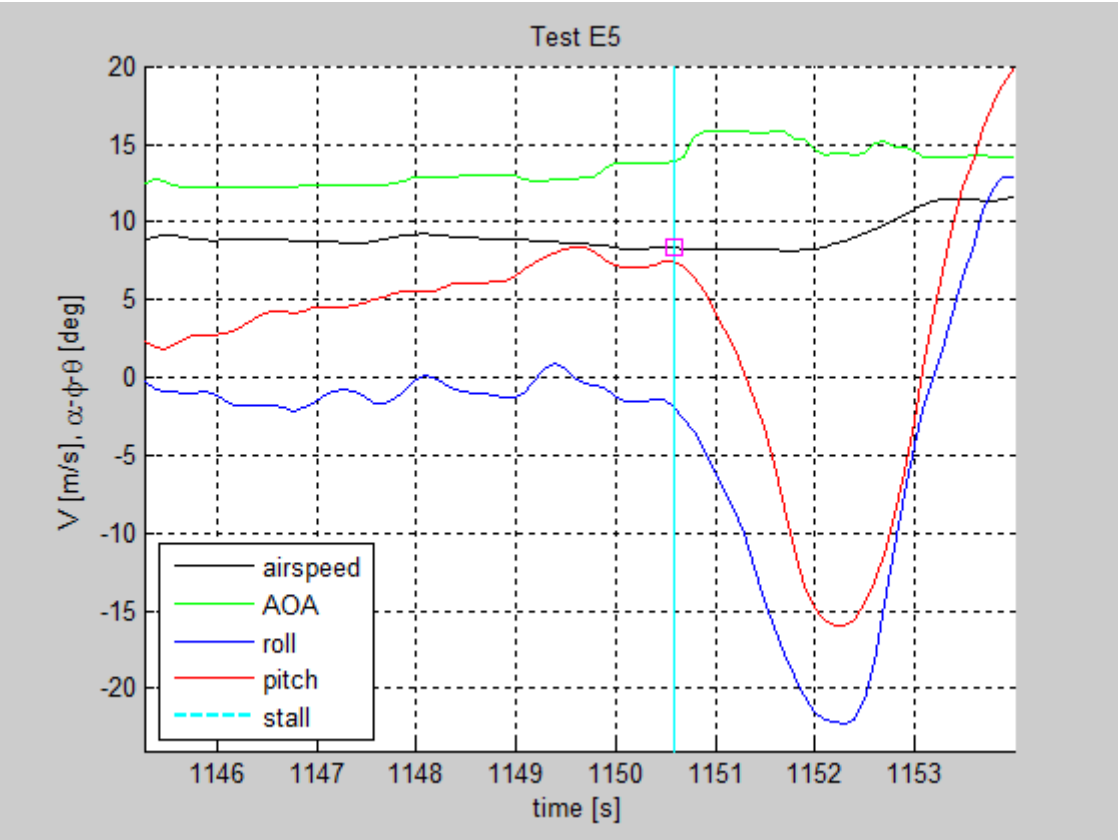


Figure 6.27 – Time history of a typical endurance configuration stall

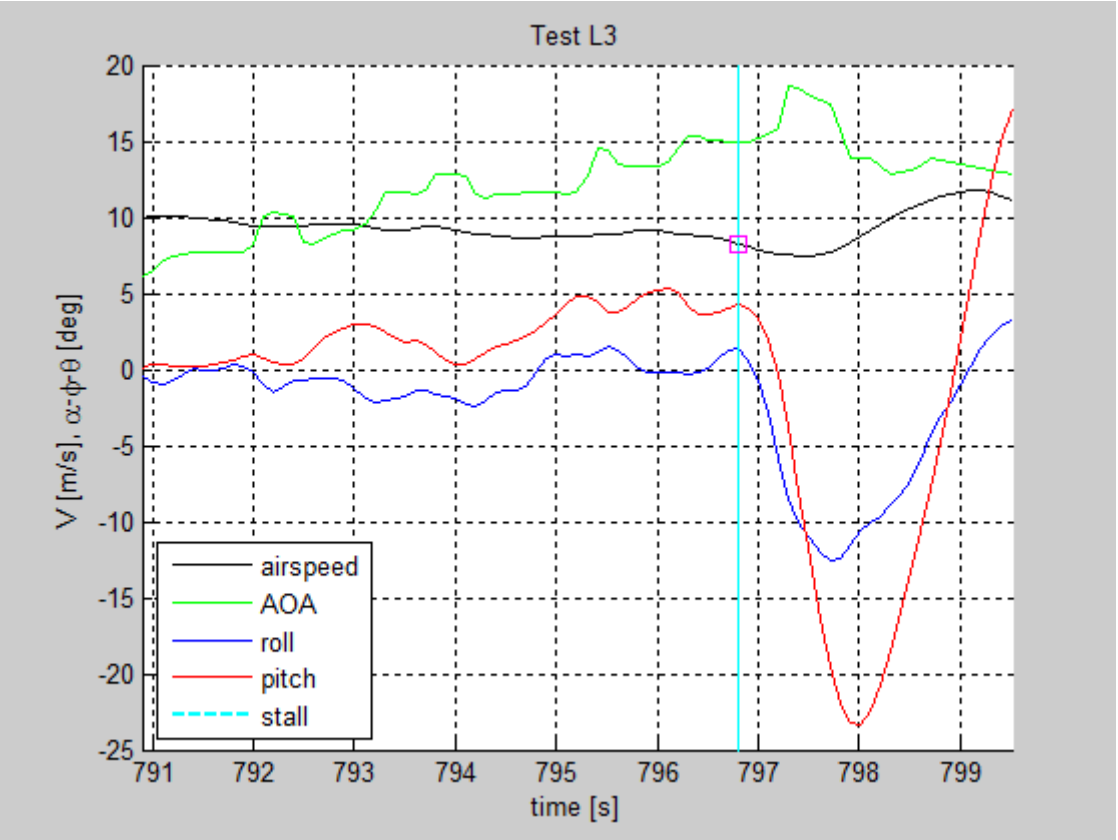


Figure 6.28 – Time history of a typical landing configuration stall

The three graphs above, all show quite similar trends:

1. the airspeed gradually decreases before the stall point, then right after reaching its minimum rapidly increases again due to nose down attitude of the airplane rapid loss of altitude.

2. The angle of attack gradually increases (not considering external disturbances as wind gusts) until and a little after the stall point due to the vane inertia.

3. The pitch angle, as the AOA, increases until a certain point then suddenly drops. Generally we refer to this point as “pitch break” and it’s the principle indicator of the exact instant at which stall occurs.

4. The average roll is close to zero (this is a requirement by the test design) until stall point. At that point the airplane, eventually, will also roll over with a wing stall indifferently to the right or to the left depending on stochastic disturbances.

5. One last observation can be made on the “quality” of the data collected which are the raw signals as registered by the autopilot: all the time histories referring to the landing configuration are more disturbed than the other two configurations probably due to the lower velocities sustaining the more “pitched up” terminal attitudes more susceptible to external disturbances in a similar way as observed before for trimmed slow flight at point 3 of section 6.5.4.

Because in this kind of test we needed a punctual value to individuate the stall speed and not an average calculated on a larger segment of the time histories as it was for the trimmed flights test, here we necessarily had to perform a filtering on the raw airspeed data affected by stochastic wind disturbances and electromagnetic noise on the signal.

The type of filter that has been used is a discrete low-pass equiripple, which belongs to the category of the FIR (Finite Input Response) filters. The signals are filtered at a frequency apt to cut the harmonics which are higher than those typical of the flight mechanics we are dealing with. The use of the function *filtfilt* available in Matlab performs a double filtering in both directions and ensures zero phase shift of the filtered signal.

Filter	***Discrete equiripple***
Category	FIR
Type	Low-pass
Passing bandwidth	1.5 Hz
Attenuated bandwidth	3 Hz
Attenuation	-60 dB

Table 6.13 – Filter specifications

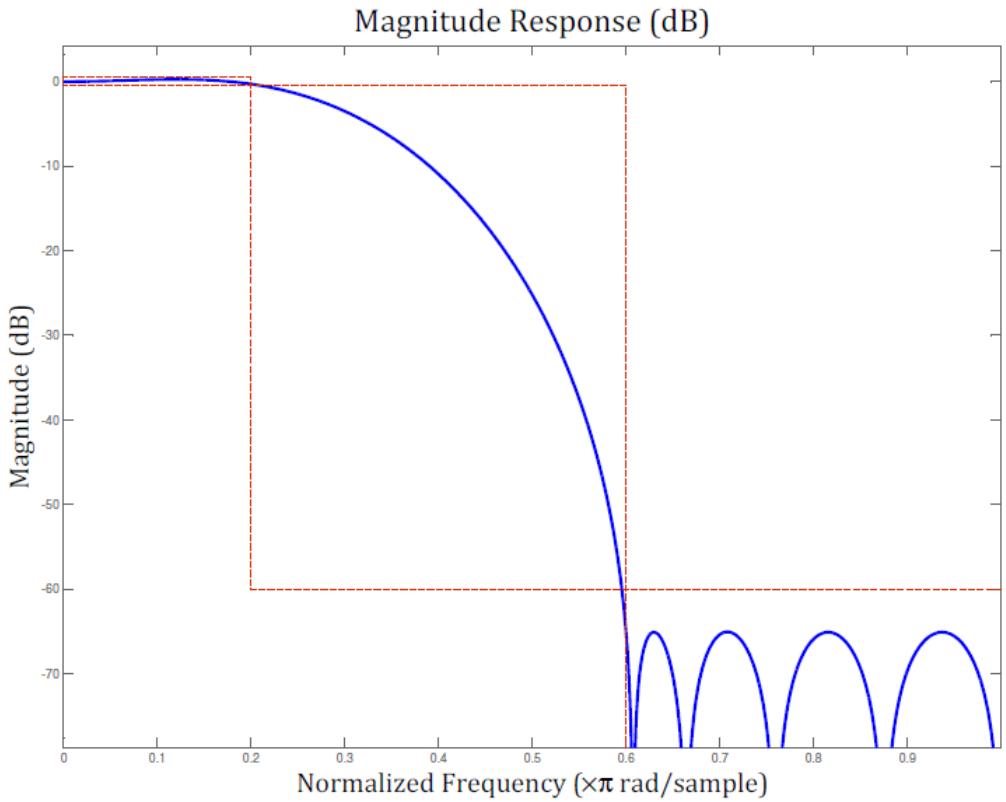


Figure 6.29 – Bode plot of the filter

After graphing the time histories for all the 15 test points (5 for each configuration), filtering the raw airspeed signal and get the correct stall speeds values, the stall entry rates has been computed as indicated in point 3 of section 6.6.1. Only the three deceleration plots relative to the three time history reported in figures 6.26, 6.27 and 6.28 will be shown here as an example but the exact same procedure has been followed for all the other test points:

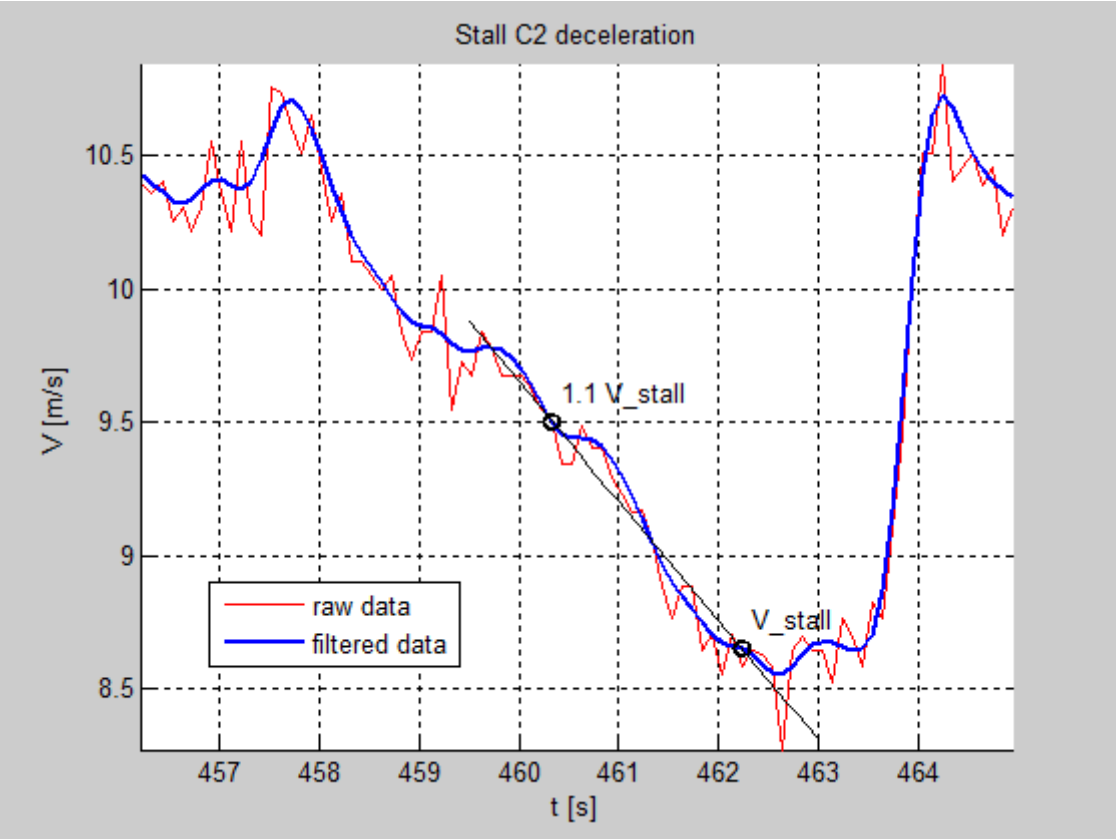


Figure 6.30 – Stall entry rate determination, clean configuration

At this much smaller scale for the airspeed on the ordinate axis than how it was plotted in the general time histories, the gradual decrement in speed before the stall point and its sudden drop briefly after it (pitch break) can be more clearly appreciated.

In all the graphs are reported both the raw data plot and the filtered one to appreciate the right signal disturbance attenuation performed by the filter especially for the electric noise affecting the piezoelectric airspeed sensor in particular al lower airspeed values.

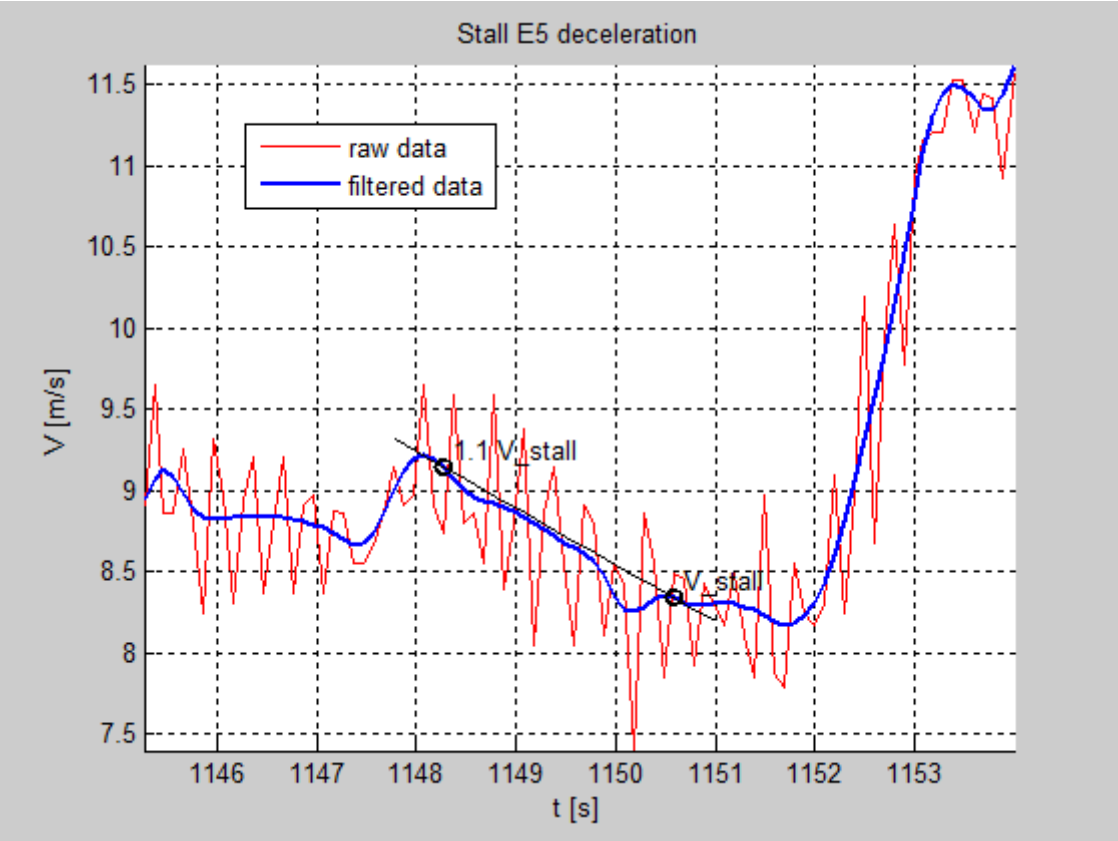


Figure 6.31 – Stall entry rate determination, endurance configuration

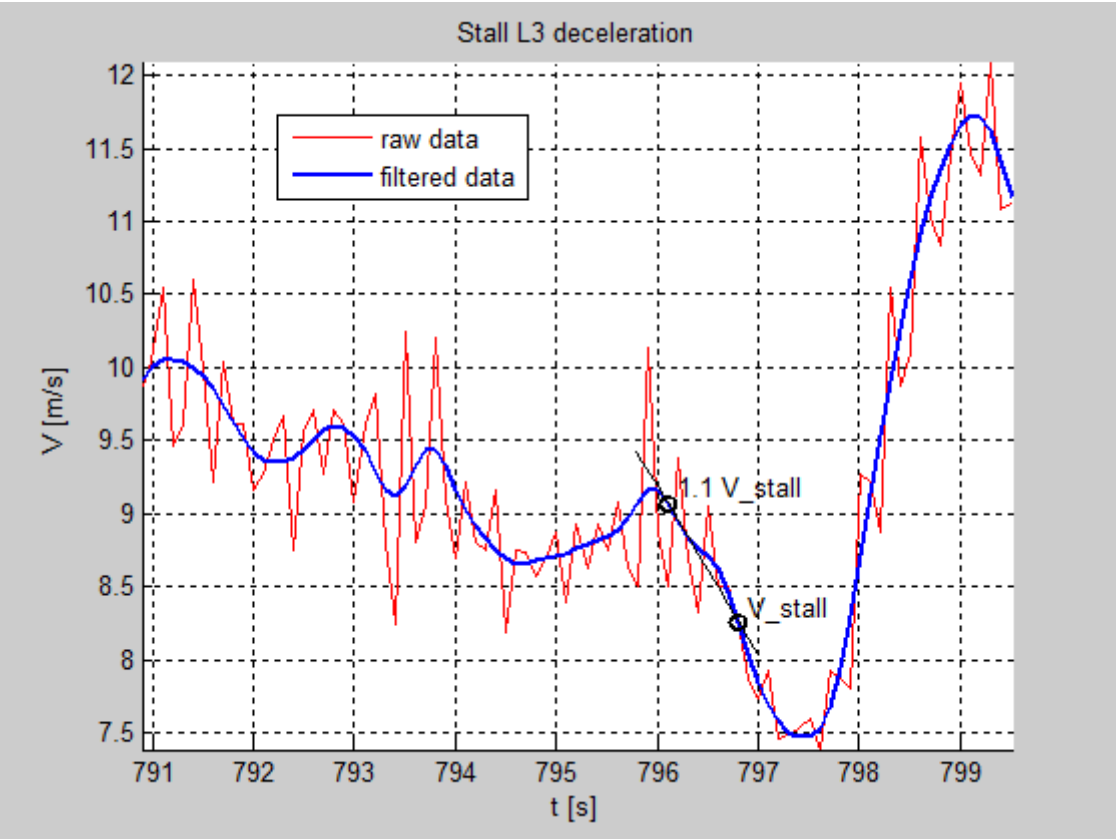


Figure 6.32 – Stall entry rate determination, landing configuration

In the following table are reported all the calculated stall speeds for each test point together with their relating entry rate accelerations (note that tests C1, C4, L1 and L2 results are not included because were badly conditioned and rejected):

Test configuration	Stall speed [m/s]	Acc x [m/s^2]
C2	8.6547	-0.44706
C3	8.9973	-0.71345
C5	8.7575	-0.5982
C6	8.8545	-0.43483
C7	8.7378	-0.7018
E1	8.4915	-0.49763
E2	8.5862	-0.32926
E3	8.7233	-0.17762
E4	8.3566	-0.2727
E5	8.3384	-0.35081
L3	8.2582	-1.1568
L4	8.3315	-0.16703
L5	8.0089	-1.0870
L6	8.3096	-0.60004
L7	7.9897	-2.9713

Table 6.14 – All stall test points results

Then, we obtained all the CL coefficients calculated by means of the well-known formula:

$$C_L = \frac{2\,W}{\rho v^2 S}$$

where W is the weight of the UAV, S the wing area and ρ the standard air density which is 1.225 kg/m3 as considered by the autopilot in the on-board processing to produce the IAS output starting from the differential pressure coming from the airspeed sensor.

As prescribed by the guidelines for civil aircraft flight testing procedures, we then plotted all the stall speeds and coefficient of lift data as functions of their relative entry rate acceleration obtaining the two graphs presented in the next two figures. As it can be observed from the pictures, for the “clean” and “endurance” configurations all the registered data were quite close in terms of standard entry rate (1 knot/sec = 0.514444444 m/s2) and the final reference values where computed by simply taking their algebraic average. Different was the case of the “landing” configuration data that, probably because it has been the most disturbed test configuration in terms of wind gusts due to the very low speeds and unstable attitude, it presented the larger variety of different entry rates. So in this case a linear fitting of the data have been executed as suggest by the guidelines for stall determination and the resultant reference values for landing stall speed and coefficient of lift also taken at the standard entry rate of 1 knot/s (the squared points on the graphs are the final resultant reference values).

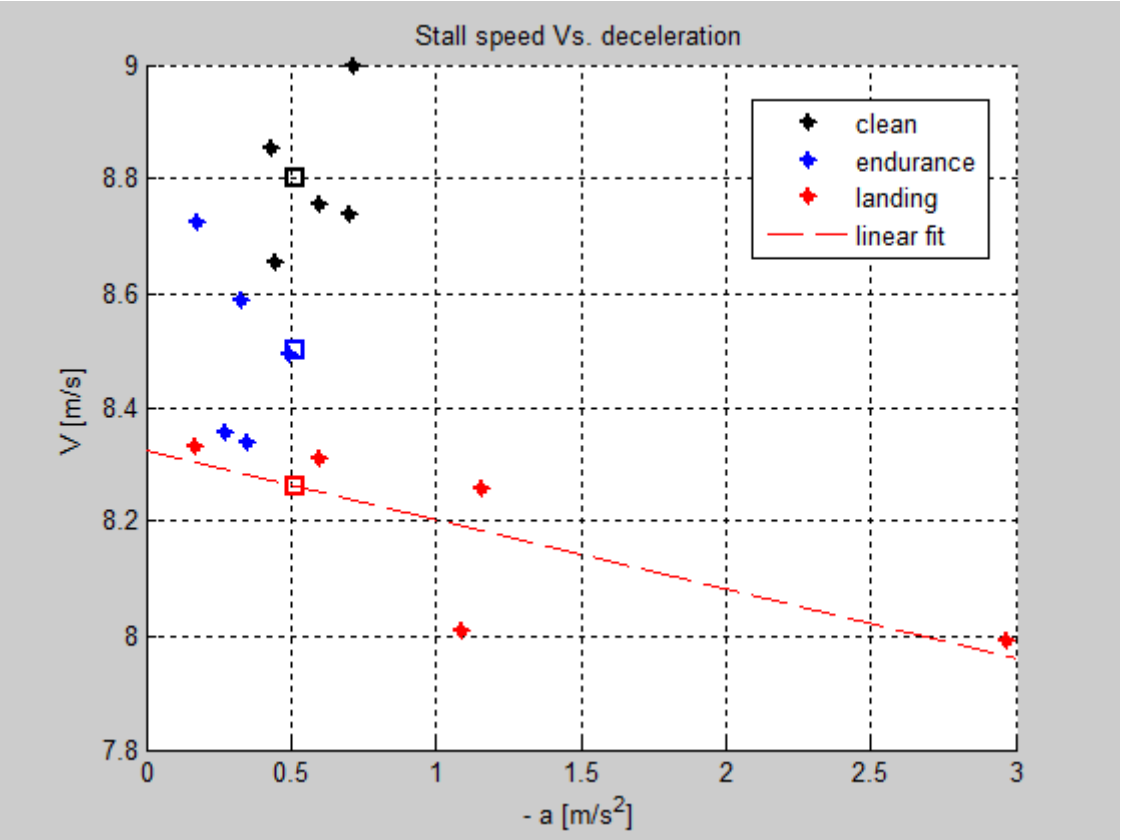


Figure 6.33 – Stall speeds determined as a function of stall entry rate

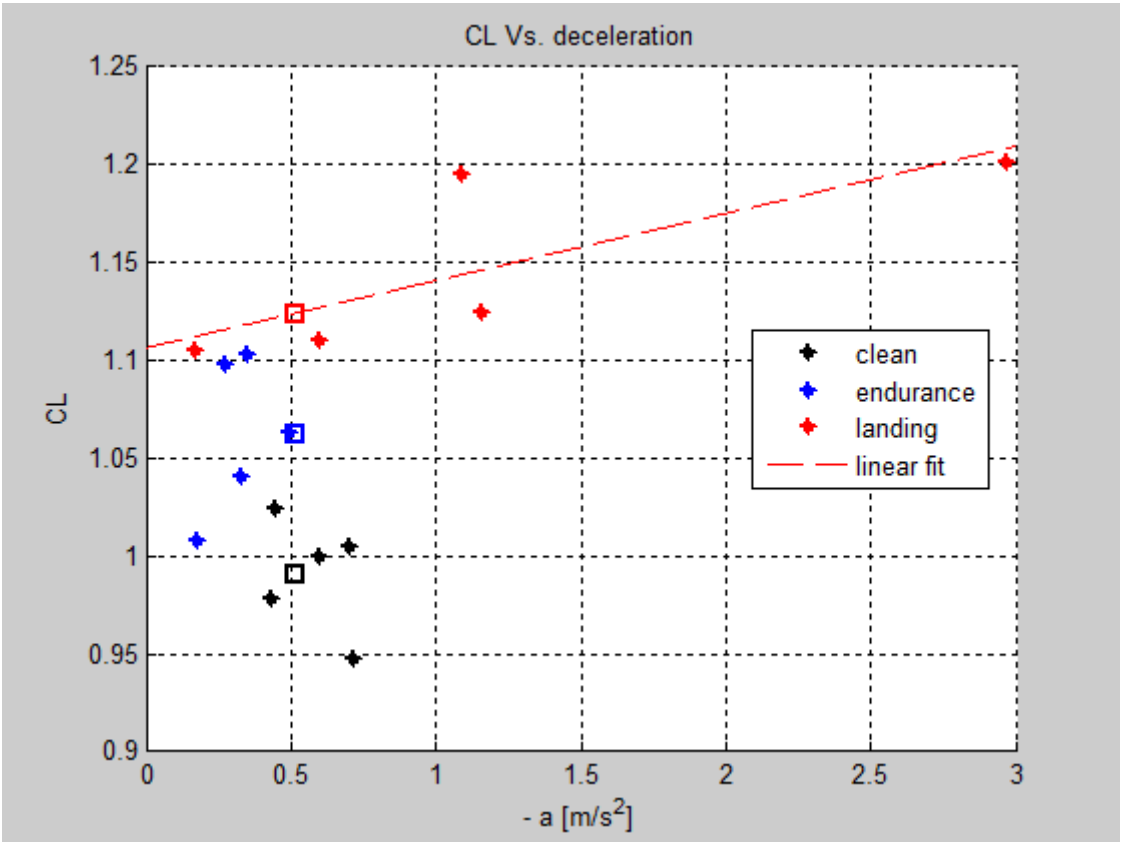


Figure 6.34 – Coefficient of lift determined as a function of stall entry rate

The final stall speeds (IAS) and coefficient of lift so defined for our UAV could then be finally reported in the table below:

Configuration	*Stall speed [m/s]*	*CL_max*
Clean (flaps 0°)	8.8004	0.99085
Endurance (flaps 8°)	8.4992	1.0627
Landing (flaps 16°)	8.2624	1.1241

Table 6.15 – Final stall test results

One last graph is presented for the stall test results showing all the stall speed test points and reference values as calculated above, divided into three groups according to their flight configuration. In this graph it can be better appreciated the tendency of the stall speed to decrease as the flaps deflection angle is increased (until a certain point) as expected.

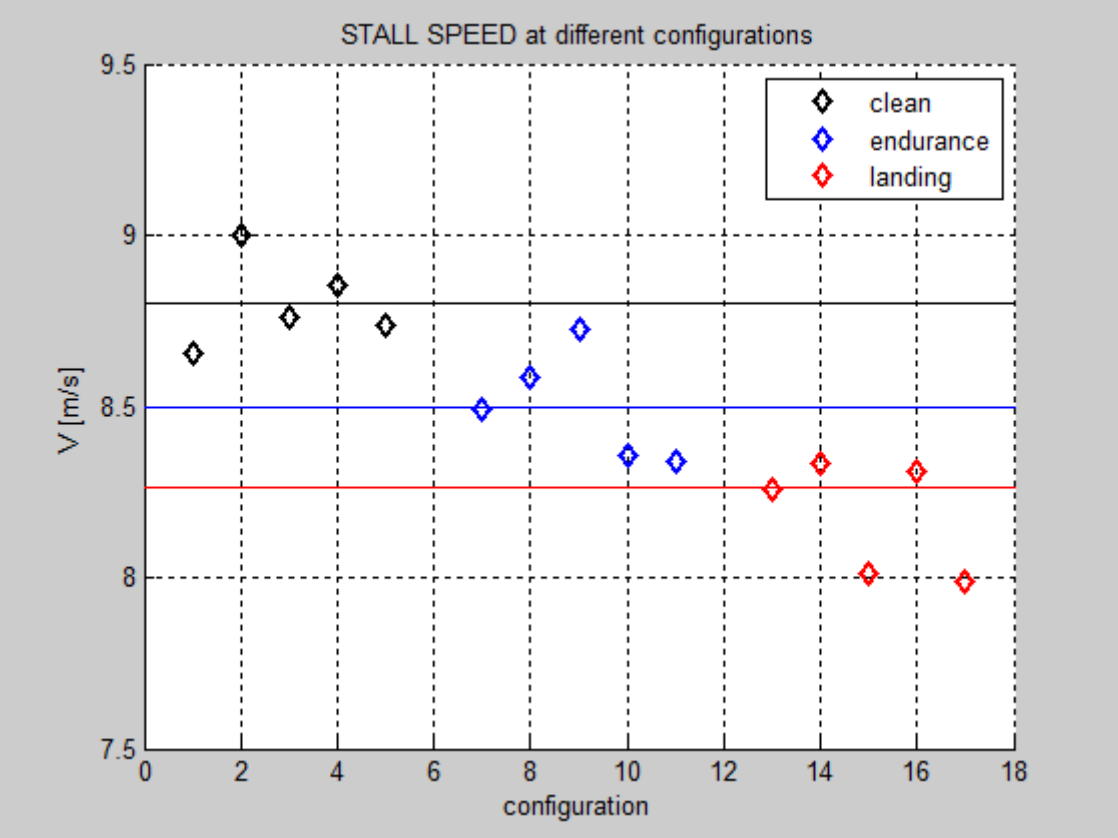


Figure 6.35 – Stall speed test points and reference values at different configurations

6.7 Glide/Climb test

The last flight test of this campaign is designed to collect the data that will be used to find the characteristic parameters defining the gliding and climbing performance of the UAV. These parameters are quite important because setting the flight at the specified speeds and attitudes suggested by the results of this test, it could be possible to optimize a glide to fly as far as possible (or as long as possible), without any power, starting from the same height before reaching the ground or to climb at full throttle gaining as much altitude as possible at the same level of energy spent.

6.7.1 Test requirements

The glide and climb flight tests have both very similar conditions:

1. One test point must be made of two legs, the first leg going in one direction (ψ) and the second in the opposite direction (ψ + 180°).

2. The directions of the two legs must be perpendicular to the wind direction (wind coming from a different side of the plane at each leg).

3. The delta in altitude for all test points should be constant (50 meters for both tests should be enough, considering the small size of the plane and its high efficiency).

In addition, the particularity of the glide test is obviously to lose altitude and of the glide test to gain altitude. All these features can be combined together to carry out both tests during a single flight. In the following figure a simple scheme of how this is intended to be achieved respecting all the conditions above is presented:

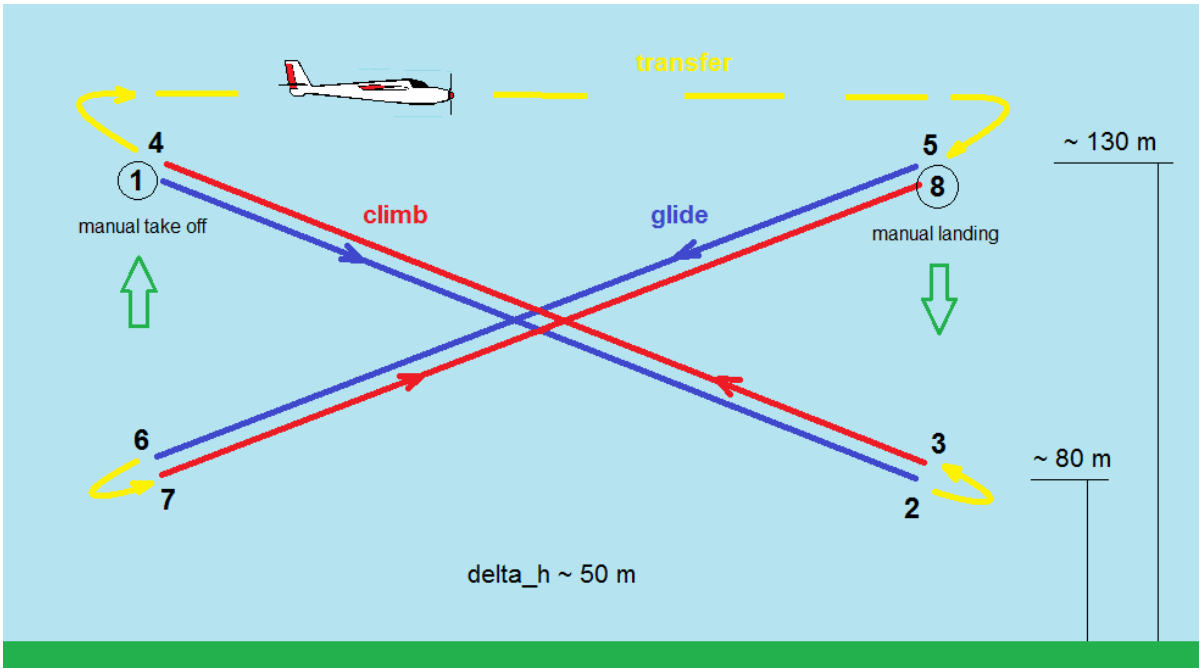


Figure 6.36 – Glide/climb tests combination

The glides/climbs maneuver was design in this way, as an "X" flight path with a change in direction every time after one test segment is completed, instead as the classic "saw tooth" flight path, primarily to never end up losing visual contact with the UAV since, especially for low glides/climbs ratios, the horizontal distance covered during one single test segment could be very large. For both types of tests we need a lot of different test segments performed at different airspeeds to be able in the post processing to define the characteristic glide and climb rate curves. The only big difference between the two tests, as we said before, is that for the glide case the throttle has to be set to 0% and the aircraft will be descending, while for the climb test the throttle has to be constantly kept at 100% and the UAV will be ascending.

6.7.2 Glide/Climb modes

GLIDE TEST DESIGN:

The glide test is carried out using the customized flight mode number 19 named *GLIDE* which is implemented constraining the throttle to 0% and the pitch angle to the desired value *FT_GLIDE_PITCH* settable through the GCS (see Appendix A for the code). In addition the flight mode also keeps the wings levelled in terms of roll angle in order to achieve a smooth and straight glide slope.

The main idea behind the design of this flight mode is that setting a fixed pitch angle, after the first few seconds of transitory stage, the flight condition will be steady and the airspeed will adjust consequently and be constant as well during the whole length of the test leg. This control on pitch rather than on the airspeed is also reasonable considering that the IMU, which collects the data from the gyros, integrated to get the attitude angles, send its data to the APM for processing at four times the speed of the air data system.

One thing to be careful about is to calibrate the level each time before a new test point paying attention that the actual level is the same as the previous level in order to provide to the auto-pilot the same zero value from where to calculate the desired attitude. Errors in this sense will result into not perfectly spaced test points but as long as the pitch angle and airspeed will be held constant, the test point will be considered valid anyways.

How to choose the right pitch angles?

Of high importance is the determination of the correct pitch angles to set to get the useful data which will be used to create the glide rate graph and to determine the important information we are looking for. From the classic flight test manuals[14] we expect to recreate a graph that looks similar to the one in the picture below:

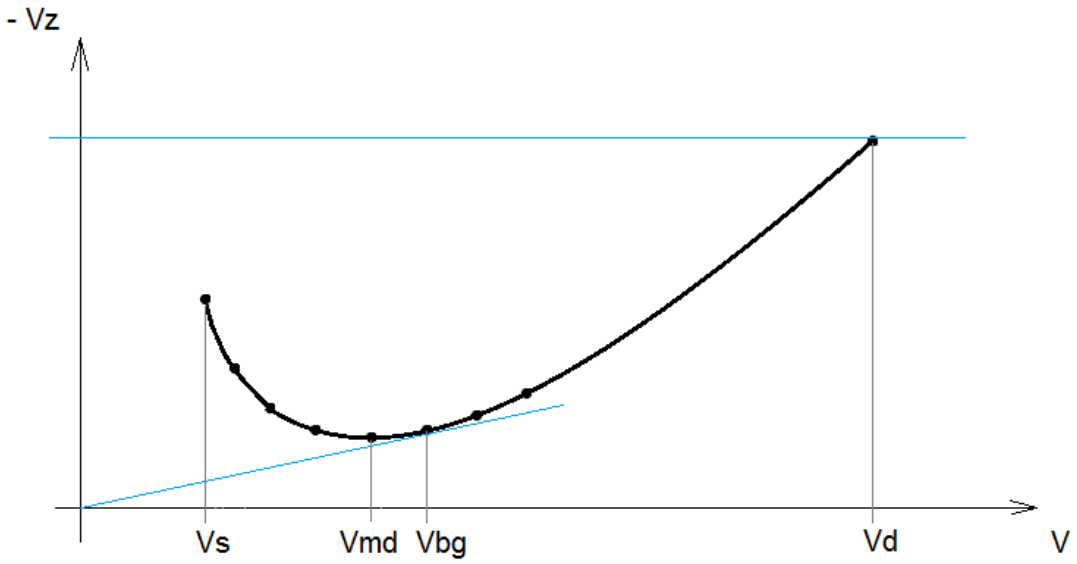


Figure 6.37 – Typical glide rate curve trend

Where for the y-axis V_Z is the vertical speed ($-V_Z$ the speed of descend) and for the x-axis V is the airspeed. In addition we have the tags of:

V_S = stall speed
V_{MD} = minimum descend speed
V_{BG} = best glide speed
V_D = dive speed

In order to correctly plot this graph, being sure to get the minimum of the function, the initial pitch angle to be set will have to be slightly positive. This way we'll start to investigate the left side of the curve and the resultant speed will be of a very low glide, if not beyond stall speed. If what we get is a stall at the first try, we'll slightly decrease the pitch angle for the next round trying to get a glide, if what we get is a constant slow glide we can try to slightly increase the pitch angle for the next round to get an even slower glide. This way, in an "iterative bisectional method", we define the pitch angle above the one we get a stall instead of a glide. After we have this value, all the following test points will be flown decreasing each time the fixed pitch angle by a small amount (e.g. 1 degree).

Going on with all the test points we will stop when we see that the glide airspeed is very high and the airplane loses altitude very fast as well. One last test at this point can be performed by setting a very large negative pitch angle (e.g. - 30 degrees) which will be useful in the post-processing to close our curve using the far right "Dive Speed" data.

IMPORTANT:
A major data density is needed around the minimum of the curve to get a better estimation of minimum descend speed and best glide speed; consider using a smaller pitch step in this area.

CLIMB TEST DESIGN:

The climb test is carried out using the dedicated flight mode number 20 named *GLIDE* which is implemented constraining the throttle to 100% and the pitch angle to the desired value *FT_CLIMB_PITCH* settable through the GCS (see Appendix A for the code). As for the glide mode, the climb mode also keeps the wings levelled in terms of roll angle in order to achieve a smooth and straight climb slope.

Just as the glide mode, but intended here for a less or steeper climb, the main idea behind the design of this flight mode is that setting a fixed pitch angle, after the first few seconds of transitory stage, the flight condition will be steady and the airspeed will adjust consequently and be constant during the whole length of the test leg.

For this flight mode, in addition to be careful of gyros calibration during pre-flight, another thing to pay attention to is to always fly with a charged battery. Start the test point with a fully charged battery and don't go beyond a safe level of charge of the battery (e.g. 20 %) because with the voltage lowering too much, a Li-Po battery won't be able to supply the correct amount of current needed to spin the propeller at the maximum range. If this happens, the test point will be incorrect because the main constrain of this test is to make sure that all the climbs at different pitch angles (and so at different speeds) have to be performed at the same value of throttle % (maximum power available).

How to choose the right pitch angles?

Also for this mode, of high importance is the determination of the correct pitch angles to get the useful data which will be used to create the climb rate graph and to determine the typical speeds we are looking for. We expect to recreate a graph that looks as something like this:

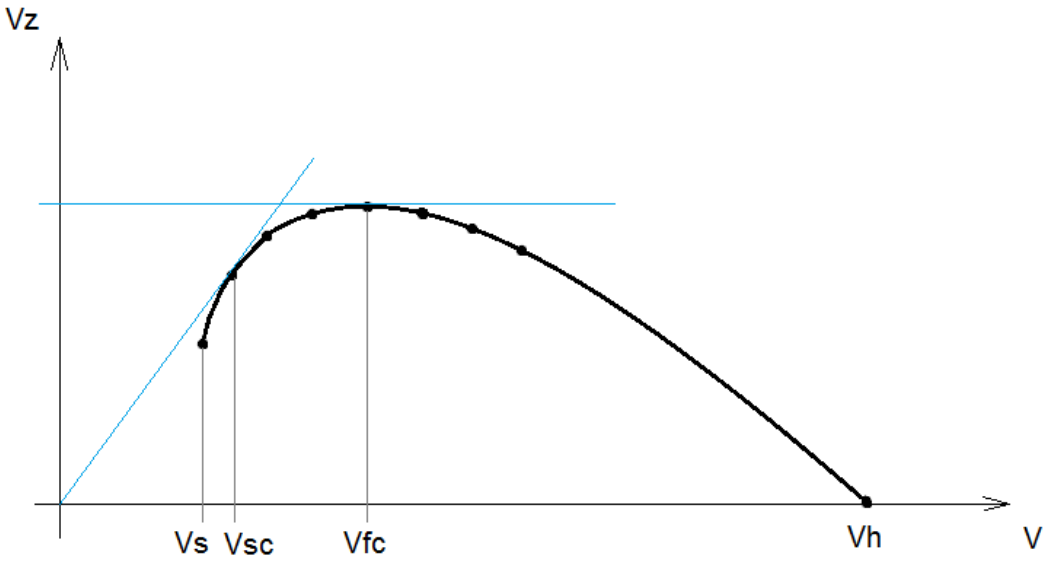


Figure 6.38 – Typical climb rate curve trend

Where for the y-axis Vz is the vertical speed (ascending speed) and for the x-axis V is the airspeed. Then in this case are indicated the tags of:

V_S = stall speed
V_{SC} = steep climb speed
V_{FC} = fast climb speed
V_H = horizontal flight speed

For the climb test, we'll start to investigate the right side of the curve. The first test point will be a levelled flight at maxim throttle (ideally at $\theta = 0°$ but this depends on the degree of accuracy of the IMU calibration in the pre-flight stage on the field). This preliminary test point will be useful in the post-processing to "close" our curve in the far right side. After this first test point, we'll gradually increase the pitch angle (e.g. 2 degrees) to start the actual climb tests and will go on with all the test points until the trust of the motor won't be enough to sustain a climb so steep.

Very important here is to get the maximum of the function so as soon as we notice that the speed of the airplane is slowing down in the climb we should reduce the delta pitch angle to a smaller amount and try to get more data in the area right before and after the fast climb speed. The last test point will be the one in which the imposed pitch angle is so large that the power available won't be enough anymore to sustain the part of the weight and the drag opposed to his traction and this will result in a stall determining the last left side point (stall speed with power).

IMPORTANT:
Be ready to switch off from the auto mode and enter into manual mode as soon as the condition of stall with power is verified to avoid the risk of crash because the airplane will be at this point at the minimum height of the maneuver (80 meters) according to the ideal flight path as pictured in figure 6.36.

6.7.3 Results

One observation to do right away observing the behavior of the UAV during flight is that the glide and climb modes, lacking of an active control on the aircraft heading, were not able at all the time to respect the condition of keeping the heading perpendicular to the wind direction. In fact as we said before, being very light and statically stable in sideslip, the heading tended to deviate more towards the wind direction. This was more obvious for the glide segments than for the climb segments so more repetitions had to be performed for the first mode. In addition, the start and final altitude of one test segment depended very much from the pilot who had the task to engage and deactivate the automatic-mode. However, alternating glides and climbs, the classic "saw tooth" trend, at least for the variation in altitude (red), could be well visualized from the GCS as pictured in the figure below:

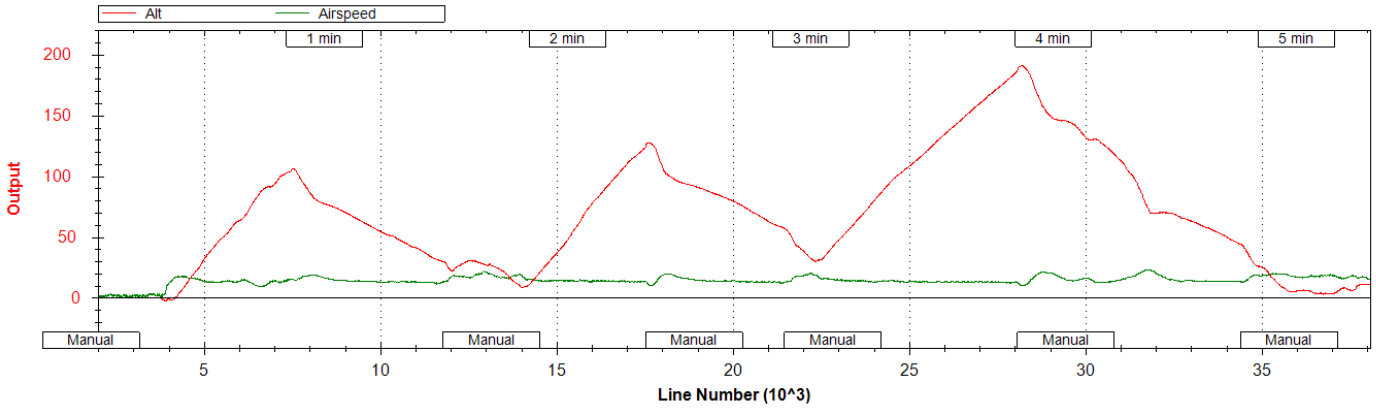


Figure 6.39 – Typical "saw-tooth" trend of the altitude during a glide/climb test session

As for the case of the trimmed flights test, all the time histories of the data registered during the different flight sessions relative to the glide/climb maneuvers have been analyzed and sub-divided into different test segments at zero or maximum throttle with constant descent/climb rate, pitch, roll, and airspeed. Following, two selection examples of a glide segment and of a climb segment with the red lines showing, once more, the beginning and ending of the useful interval of data:

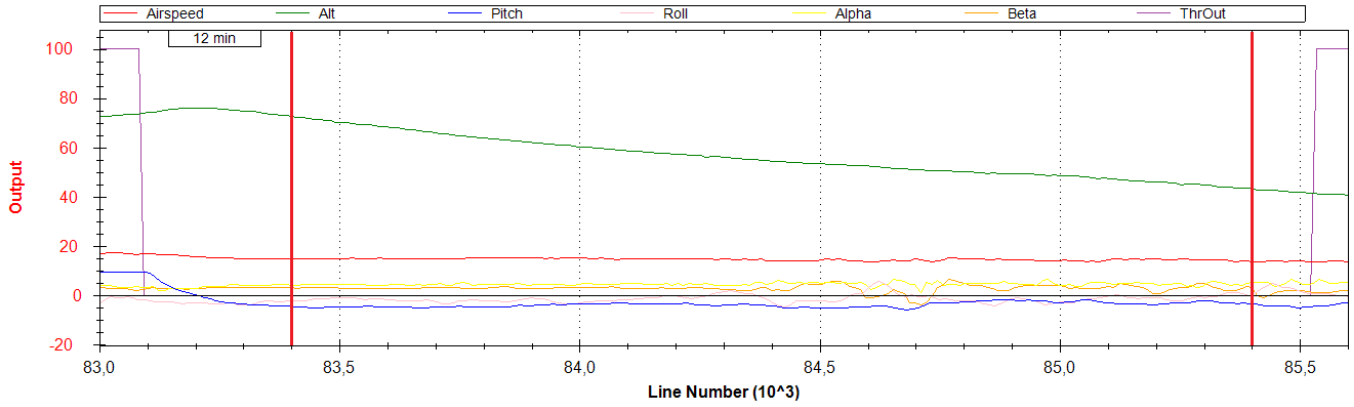


Figure 6.40 – Typical glide segment (in this case θ = -2 deg)

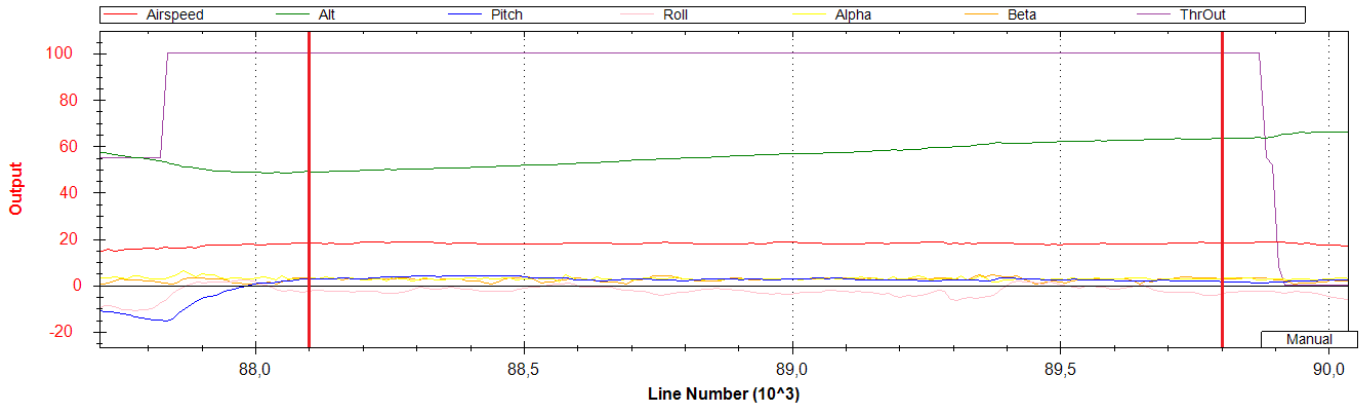


Figure 6.41 – Typical glide segment (in this case θ = +2 deg)

A lot of test segments have been performed for these tests to ensure a good precision for the estimation of the glide and climb rate characteristic curves. Precisely, 48 test segments at different pitch angles varying from 3 to -12 with a peak of -23 degrees for the dive speed have been performed using the glide mode and 34 segments instead were performed using the climb mode varying pitch angle from a minimum of 0 to a maximum of 30°. In figures 6.42 and 6.43 can be observed how keeping the throttle fixed and only varying the set pitch angle results in a direct proportional variation of the airspeed as well.

For each isolated test segment, the following parameters were calculated:

- Duration of the test segment, Δt [s]
- Altitude interval, Δh [m]
- Reference altitude, h_{ref} [m] (see figure 150)
- Airspeed, v [m/s]
- Maximum airspeed variation, dv [m/s]
- Vertical speed, v_z [m/s]
- Angle of attack, α [deg]
- Angle of sideslip, β [deg]
- Pitch angle, θ [deg]
- Maximum pitch angle variation, $d\theta$ [deg]
- Roll angle, ϕ [deg]

We won't report here, as in the case of the trimmed flights test, all the specific numerical values (which are close to 1000 numbers) but we limit to state the acceptance minimum values for the parameters above to consider a specific test segment as valid:

Parameter type	***Acceptance conditions***
Δt	> 10 s
Δh	> 10 m
dv	5 m/s (± 2.5 from average)
$d\theta$	10 deg (± 5 from average)

Table 6.16 – Acceptance conditions for glide/climb test segments

The restrictions on the duration and altitude interval of the test segments where chosen to guarantee a statistical reliability of the average parameters calculated, while the restrictions on airspeed and pitch values were imposed to reject those test segments too badly conditioned by external disturbances (e.g. wind). Imposing the above restrictions, the actual validated test segments for the glide mode reduced to 43 with 5 rejected test points and for the glide mode 31 were validated with 3 test points rejected.

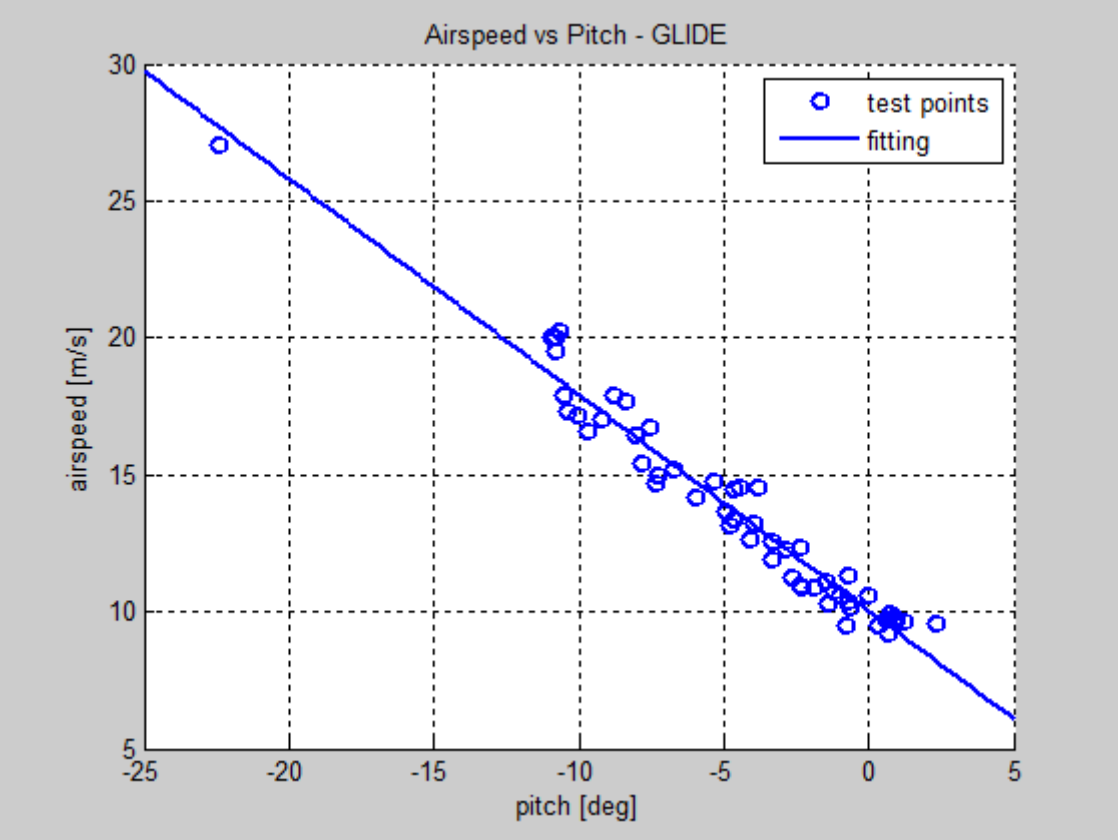


Figure 6.42 – Variation of gliding speed as a function of set pitch angle

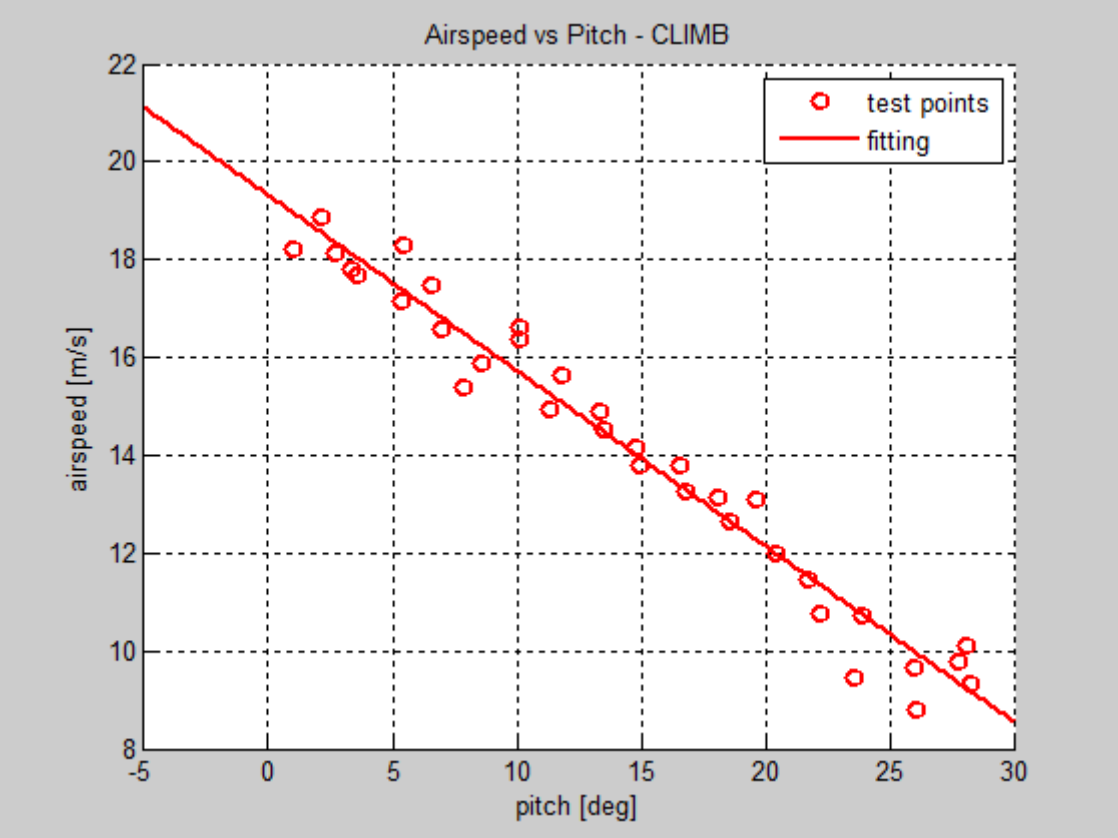


Figure 6.43 – Variation of climbing speed as a function of set pitch angle

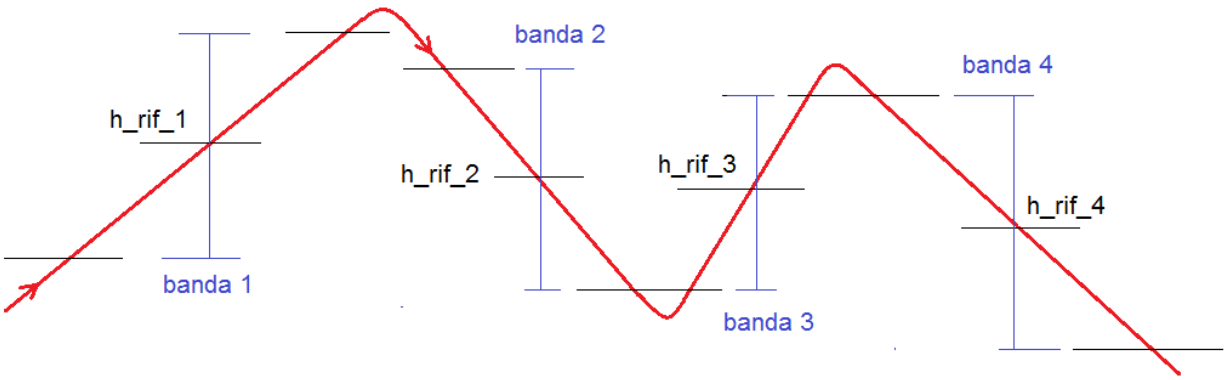


Figure 6.44 – Reference altitude as considered for each test segment

We remember that the airspeed measure by the APM system is an equivalent airspeed (EAS) already corrected for the positioning error (PEC) and instrument error.

The data of vertical speed for each test point were calculated by means of the formula:

$$V_{z_j} = \frac{\Delta h_j}{\Delta t_j}$$

where Δh_j represent the altitude interval and Δt_j the time interval for each "j-th" test point.

A data reduction had to be performed to refer the calculated apparent rates of climb to the true rates of climb using the ratio between the measured temperature and the standard temperature for each flight segment at its reference flight level. To calculate the standard temperature of each flight level, we used the standard ISA (International Standard Atmosphere) function[21] to determine the temperature profile as a function of the altitude above the mean sea level:

$$T_{Sj} = T_0 - ELR \cdot \frac{(h_g + h_{ref_j})}{1000}$$

where,

T_0 = standard ISA temperature, 15°C
ELR = Environmental Lapse Rate[20], 6.4 °C/km
h_g = ground level altitude of our field, 118 m
$h_{ref\ j}$ = reference altitude of each test segment

To calculate the measured temperature at each flight reference altitude, we referred to the temperature measured on the ground field by the portable meteorological station during the whole day, that presented a quite linear and incrementing trend as it can be seen in figure 6.45, and then on this base we again used the ISA formula to calculate the approximated test temperature at the reference altitude of each flight segment (the flight segments were supposed equally spaced in time during the whole day of flight tests):

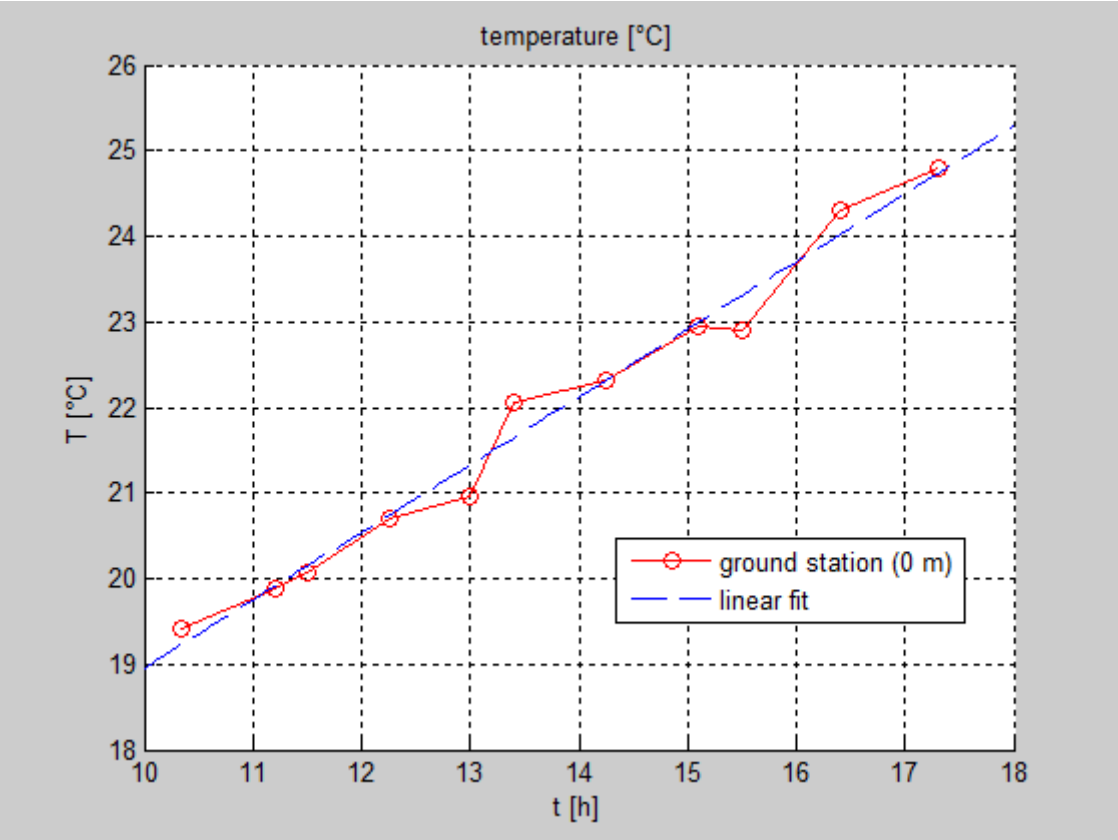


Figure 6.45 – Linear trend of the on ground temperature during the test day

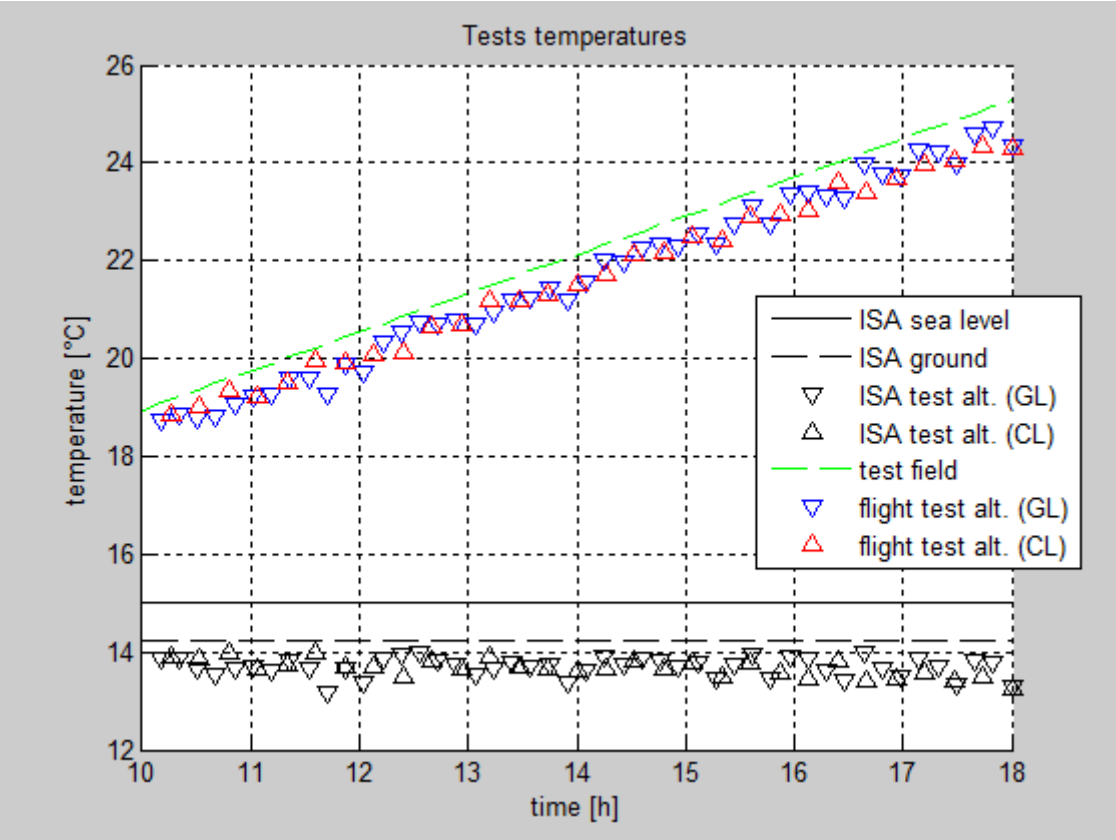


Figure 6.46 – All test and standard temperatures as calculated for each glide and climb test segment

At this point, disponing of all the actual test and standard temperatures as calculated for each glide and climb test segment plotted in figure 6.46, the reduction of the vertical speed data could have been calculated by:

$$V_{ZS_j} = \left(\frac{T_j}{T_{S_j}}\right) V_{Z_j}$$

where,

$V_{ZS\,j}$ = vertical speed in accordance with ISA temperature for each test segment
$V_{Z\,j}$ = raw vertical speed as calculated from the collected data of altitude and time intervals
$T_{S\,j}$ = standard temperature at the reference altitude value of each test segment
$T_{\,j}$ = test temperature at the reference altitude value of each test segment

In the two followings glide/climb rate plots, both the raw data and the standardized data of the vertical speeds are reported to better appreciate the gap coming from a difference in test temperature compared to standard temperature (see figure 6.45) that varied from a minimum of 5°C to a maximum of 11°C during the day of the test:

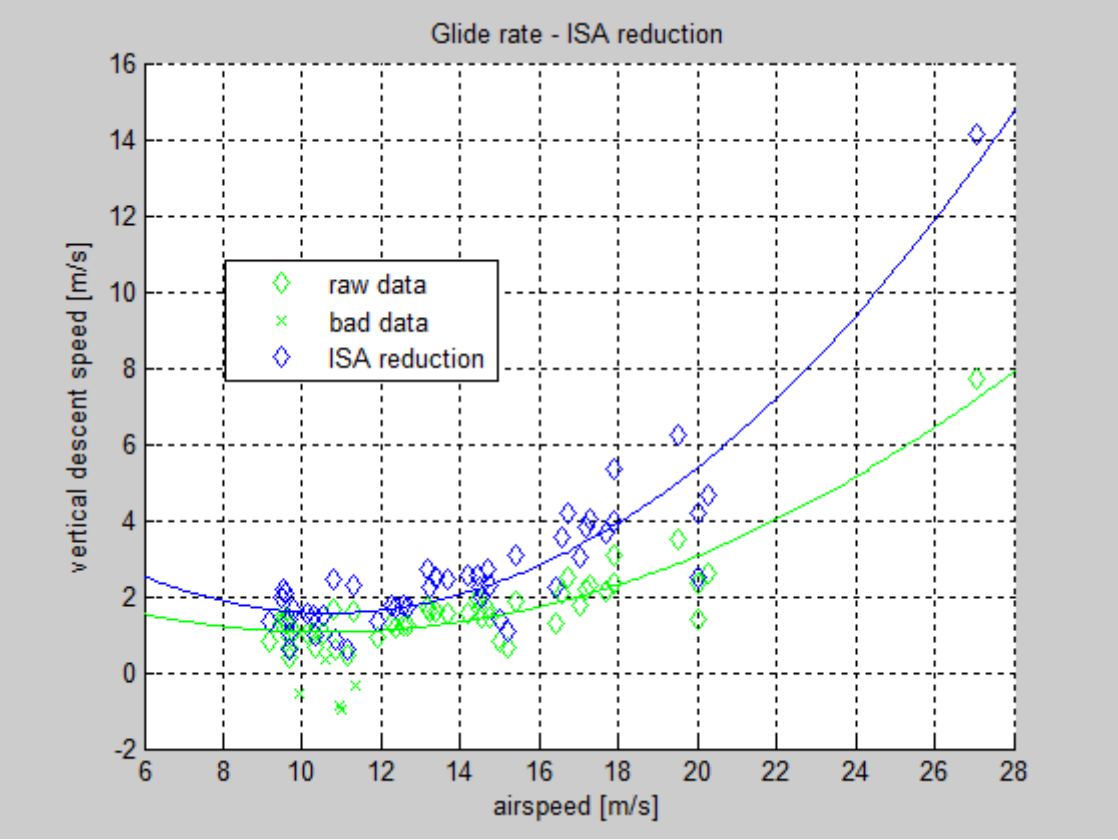


Figure 6.47 – Glide rate plot with standard and non-standard vertical speeds values

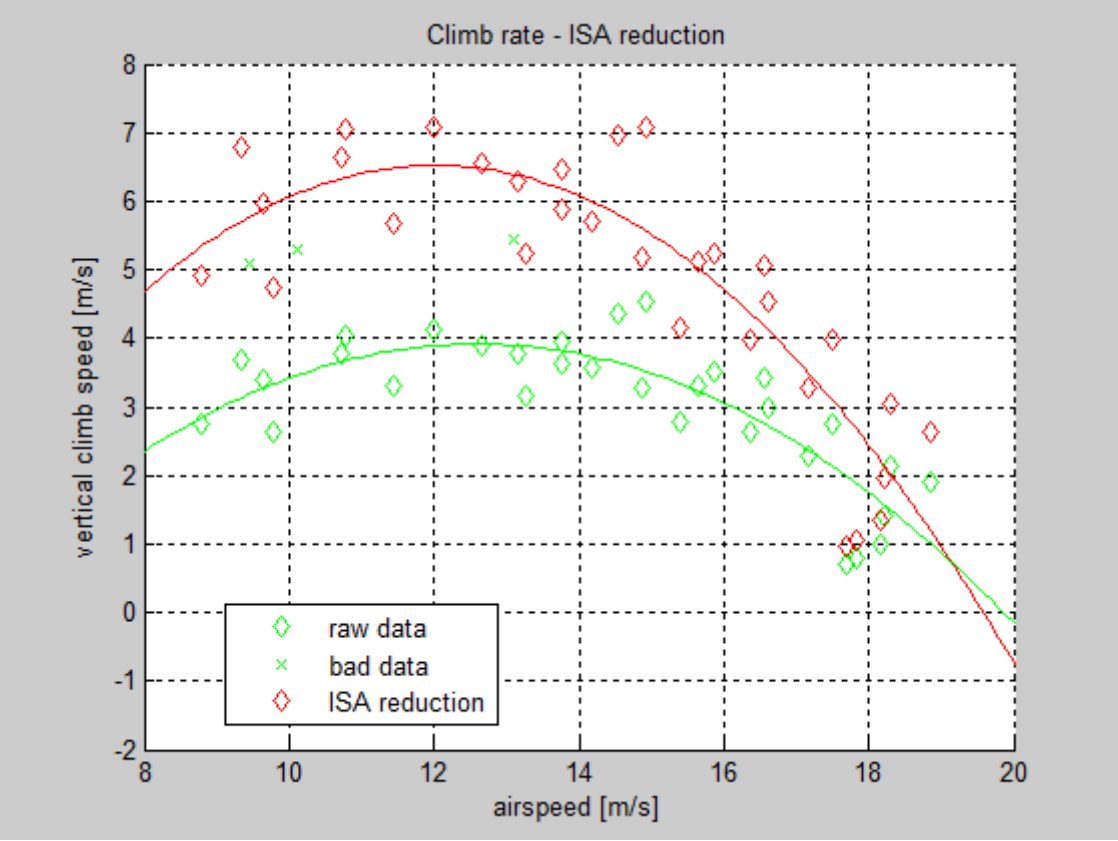


Figure 6.48 – Climb rate plot with standard and non-standard vertical speeds values

As it can be noted from the graphs above, both the standard "true" glide and climb rates increase starting from a higher test temperature than the reference standard temperature of 15°C, as expected.

From these graphs is already quite clear how our resultant curves for the glide and climb rates came out quite accurate in comparison to the typical trend of these kind of graphs, as mentioned before and showed in figures 6.37 and 6.38, in spite of some problems during the flight testing stage. Note that the "bad data" points, as reported for both the glide and the climb rate graphs, were clearly out of trend due to environmental disturbances (mainly thermal convective currents).

Now, graphing only the standardized data and their relative fitting for better visualization (see figures 6.49 and 6.50), we could also individuate the minimum and maximum points of the functions along with their maximum efficiency points in order to evaluate the important parameters of:

For the glide test: **best glide** and **minimum descent** airspeeds and descend speeds.

For the climb test: **steep climb** and **fast climb** airspeeds and climb speeds.

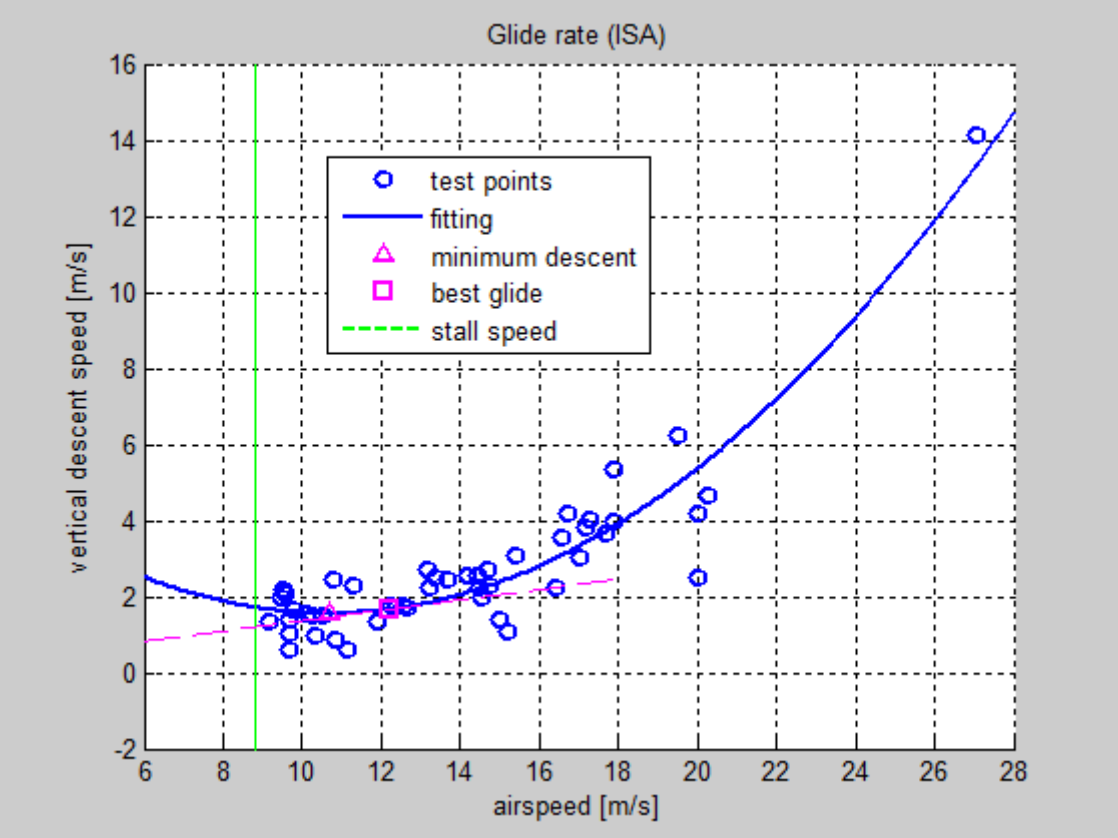


Figure 6.49 – Glide rate plot with minimum descend and best glide points visualization

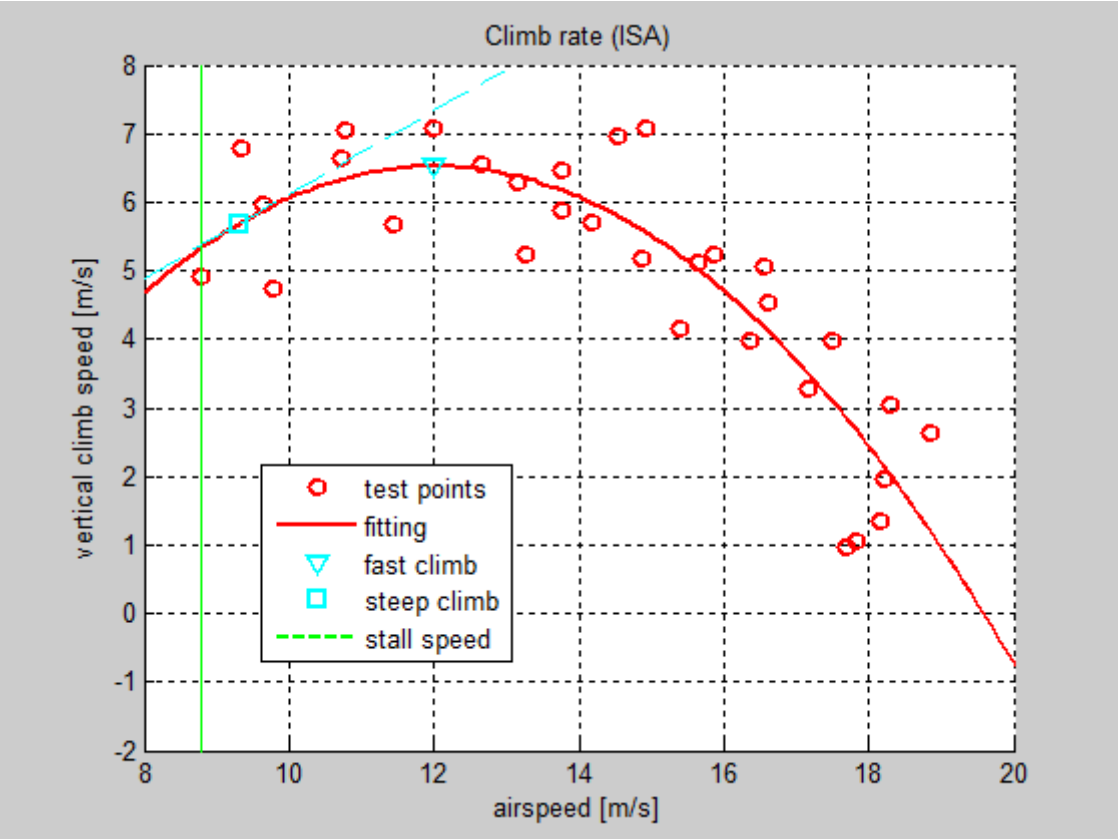


Figure 6.50 – Climb rate plot with fast climb and steep climb points visualization

The calculated numerical values of the important parameters relative to the characteristic points plotted above, are reported in the following tables:

Glide test	
minimum descent airspeed	10.7 m/s
minimum descent vertical speed	1.5605 m/s
best glide airspeed	12.2 m/s
best glide vertical speed	1.6633 m/s

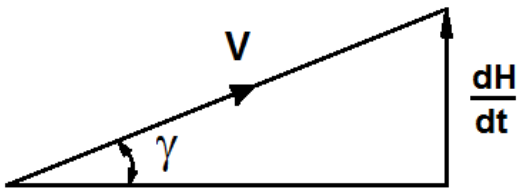
Climb test	
fast climb airspeed	12 m/s
fast climb vertical speed	6.5226 m/s
steep climb airspeed	9.3 m/s
steep climb vertical speed	5.6811 m/s

Table 6.17 – Glide and climb tests numerical results

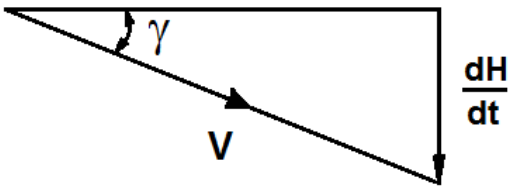
Two last graphs have been plotted to get to know the glide and climb angles at, respectively, the best glide and the steep climb conditions, calculated by:

$$\gamma_j = \sin^{-1}\left(\frac{V_{Z_j}}{v_j}\right)$$

where V_Z is the vertical glide or climb speed and v is the measured airspeed for each test segment.



γ = γ_c = climb angle



γ = γ_d = descent angle

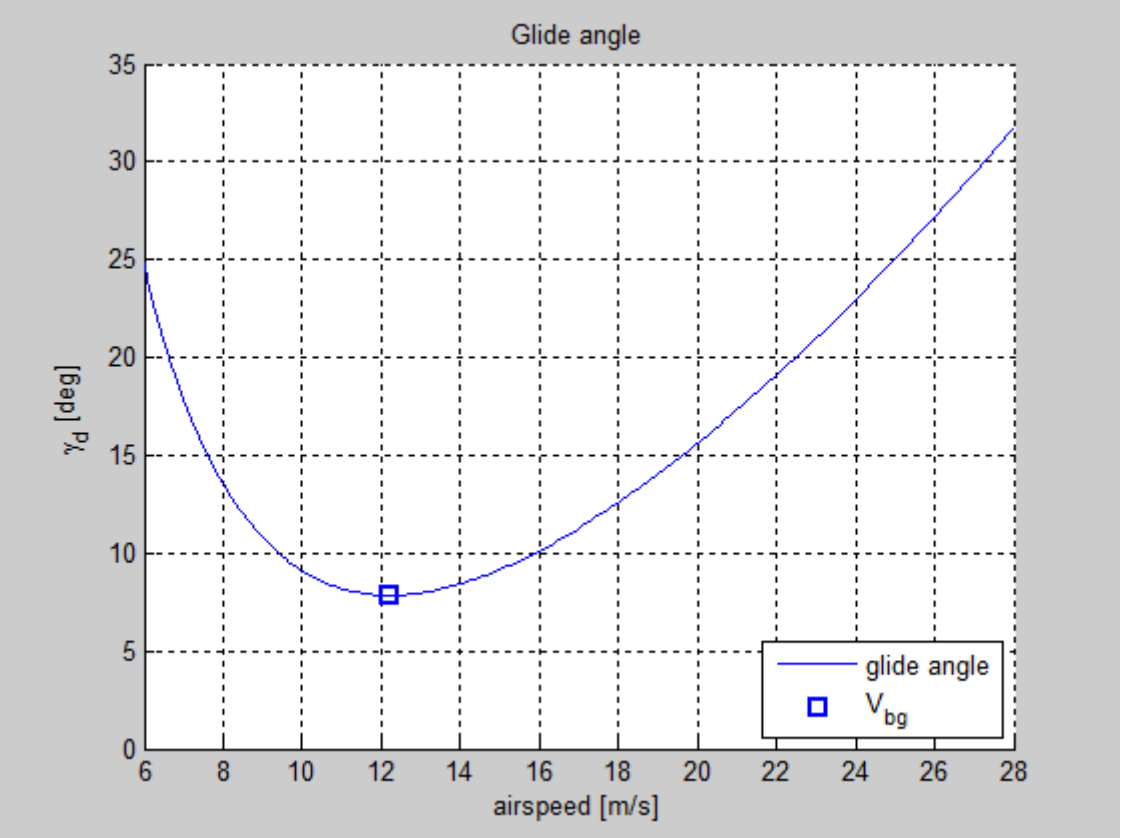


Figure 6.51 – Climb rate plot with fast climb and steep climb points visualization

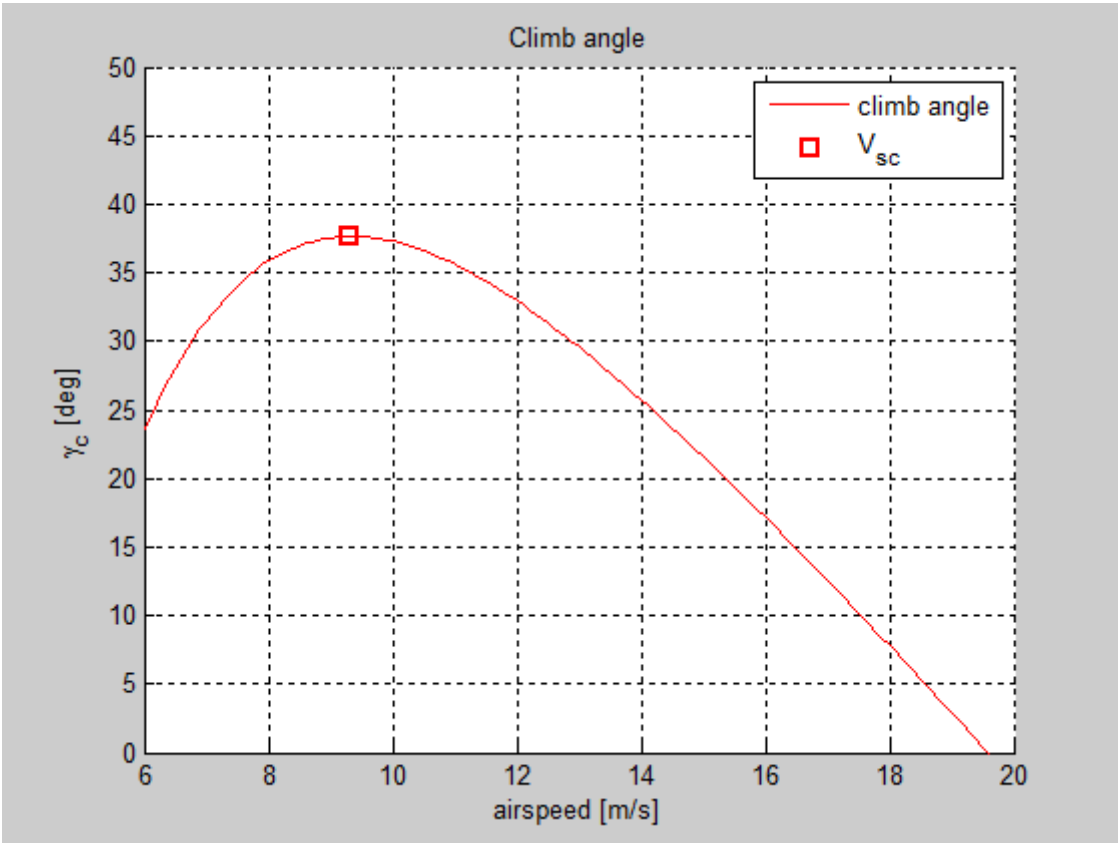


Figure 6.52 – Climb rate plot with fast climb and steep climb points visualization

The minimum and maximum points of these last two functions represent, respectively:

best glide angle (blue) = 7.8361 deg
steep climb angle (red) = 37.6528 deg

>>

These are the last results relative to the last test of the whole flight campaign which ended the 28th of May 2015.

Figure 6.53 – The last day spent at San Vito di Gaggiano test field

7. Conclusions

In this chapter a critical analysis of the thesis work is made in terms of evaluation of the results which were obtained compared to the initial objectives. This analysis raised some issues that might inspire a future development of this project.

7.1 Work evaluation

The objective of this thesis work was to perform a complete flight test campaign to define some of the specific aerodynamic parameters apt to characterize the performance of a fixed-wing UAV.

This activity has been complete in all the characteristic stages of a real flight test campaign, from the design of the automatic flight modes required to perform the tests to the post-processing of the data and the presentation of the results. In the preliminary stage of the work, in addition to the pure flight testing experimental activity, were also performed some numerical analysis and the required wind tunnel tests for the calibration of the new integrated sensors.

What has been more challenging, other than the contingencies typical of all experimental activities, is the commitment it took to become familiar with the hardware system of the UAV and especially with the code constituting the software running on it. In fact, the *ArduPlane* open source project, with many developers working on it since several years until now is already at a very high level of complexity (the whole project with the base code scripts and related libraries comes close to 700,000 lines[5]) and integrating new non-standard sensors or learning where it was that function performing that action in order to write a new flight mode that would do exactly what we intended to do resulted quite complicated.

The good thing about the APM system is that it has been proved to be very reliable in any occasions and the innovation of the *AUTOTUNE* mode brought by the new firmware version (the V3.2.1alpha we had in use on the APM 2.6 board) has been very useful to perform the new tuning of the gains after the flying platform was heavily modified by the introduction of the new AOA/S and airspeed sensors.

Especially the introduction of the AOA sensor, allowed us to present many interesting results as functions of the angle of attack which is one of the most important parameters in aerodynamics together with the airspeed. The quite good results we came down to, are the proof that the calibration has been well done together with the execution of the mixed automatic and manual flight test maneuvers for each flight test and I'm overall satisfied as both the author of this thesis and as the pilot of the UAV.

7.2 Results evaluation

The main objective of the thesis work has been accomplished and the results are in good accordance with the expectations, well characterizing the complete polar of the aircraft, its stall behavior and the glide and climb characteristic angles and velocities.

Of course the unprofessional instrumentation we used doesn't offer the same level of precision we would expect from a standard activity of flight testing on a real airplane, however, considering the ultra-small dimensions of the whole system (the APM board size is just 7.5 x 4.5 cm) necessary to fit in the UAV canopy and its very low cost, we can be nothing but satisfied of its performance.

The only big difference between a real flight testing campaign and our tests can be pointed out in the aircraft itself. Even though the *Cularis* is an excellent aircraft with a quite large wingspan to be a model glider, well made owning docile and predictable flight characteristics making it a very reliable machine, still is a small foam plane. It demonstrated, as shown by the test results, a clear sensibility to the environmental conditions (especially wind and thermal currents) mainly due to its low wing load coefficient. More in general we can say that this depends on the clear difference in ratio between its dimensions and the dimensions of a classic general aviation aircraft compared to the environment variables.

As an example, we can point out the data points rejected in the glide rate graph (see figure 6.47) which presented a "negative descent speed". In fact, as we could directly see on the field during the glide test session, and as recorded by the autopilot in the flight time history, with the throttle set to zero and the pitch close to zero as well, for some test segments our model glider was doing exactly what he was meant to do: soaring with the ascending thermal currents typical of a warm spring day in the late morning. So in the unpowered flight test where we expected the UAV to glide at a certain sinking rate, it was actually climbing instead due to external agents as reported in the time history below:

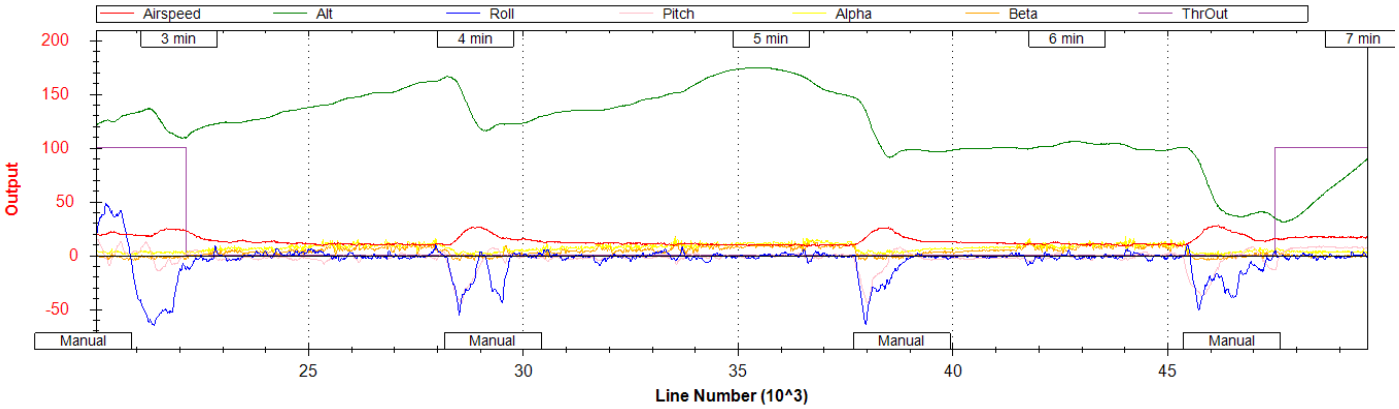


Figure 7.1 – Proof of the UAV soaring due to thermal ascending currents

In the previous figure it can be noted that when the throttle (violet) is set to zero, the airspeed (red) decreases initially and then becomes constant but the altitude (green) keeps to increase! However, in spite of this isolated case, the other many regular test points that were plotted and considered for the fitting still assured a quite good estimation of the glide rate curve.

Talking now specifically about the C_L-α, C_D-α and aerodynamic polar curves, we could make a quite interesting general comparison between the results coming from the flight test, those coming from the wind tunnel and the numerical simulation ones obtained with Xfoil and AVL. For all the comparison graphs presented in the next three figures, we have to remember that the Xfoil results as the wind tunnel ones refer to only the aircraft wing, while the AVL and flight test results refer to the complete airplane. Graphing all these curves referring to different models in a single figure might seem strange but is actually a good way to appreciate how much (or not much) the presence of fuselage and tail influences the measures/simulations among solely the main wing results comparing to other factors as ideal/viscous fluid, 2D/3D models, wind tunnel close test chamber or open air as it will be investigated:

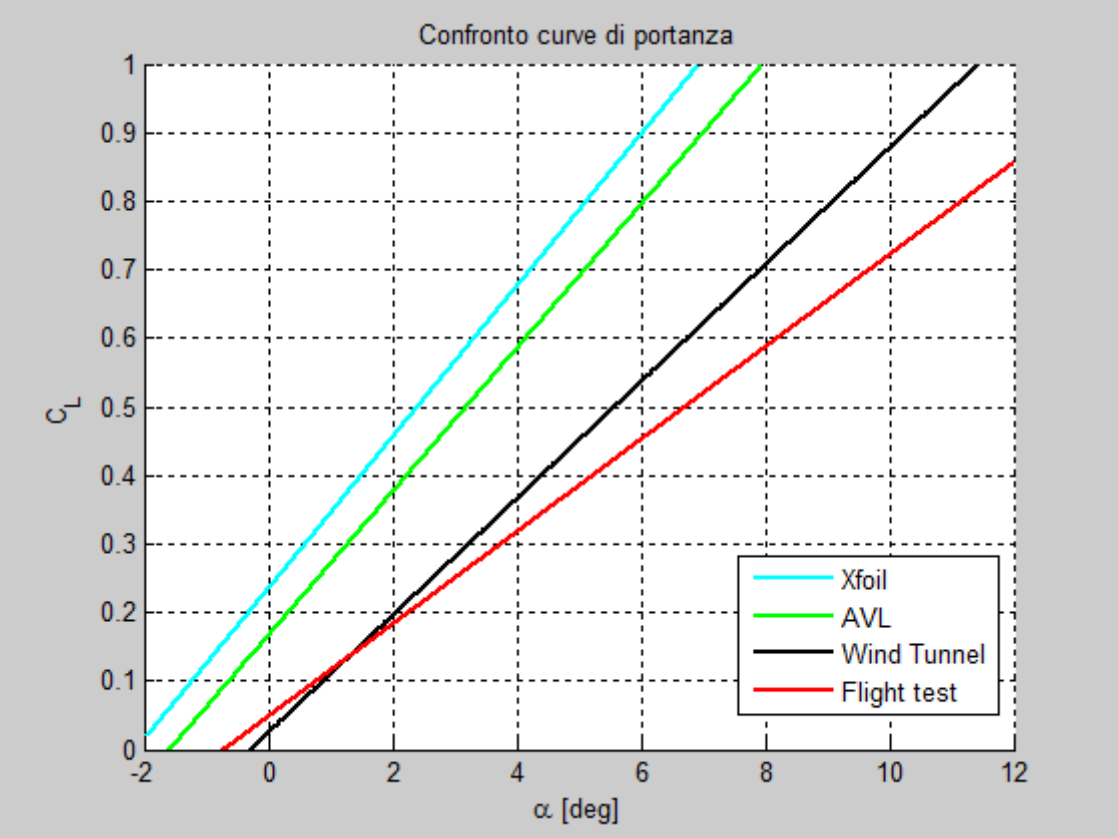


Figure 7.2 – Comparison among all coefficient of lift curves

As expected, the curves coming from the simulated environments present higher lift coefficients than the experimental ones. The Xfoil result is better than the AVL because Xfoil only simulates a 2D airfoil with theoretical infinite wingspan while AVL simulates the whole aircraft. Same thing can be said for the wind tunnel result comparing to the flight testing.

A single wing lift is not influenced by the negative lift produced by the tail in a conventional configuration aircraft. AVL is still very far from the flight test result because it only simulates an inviscid flow.

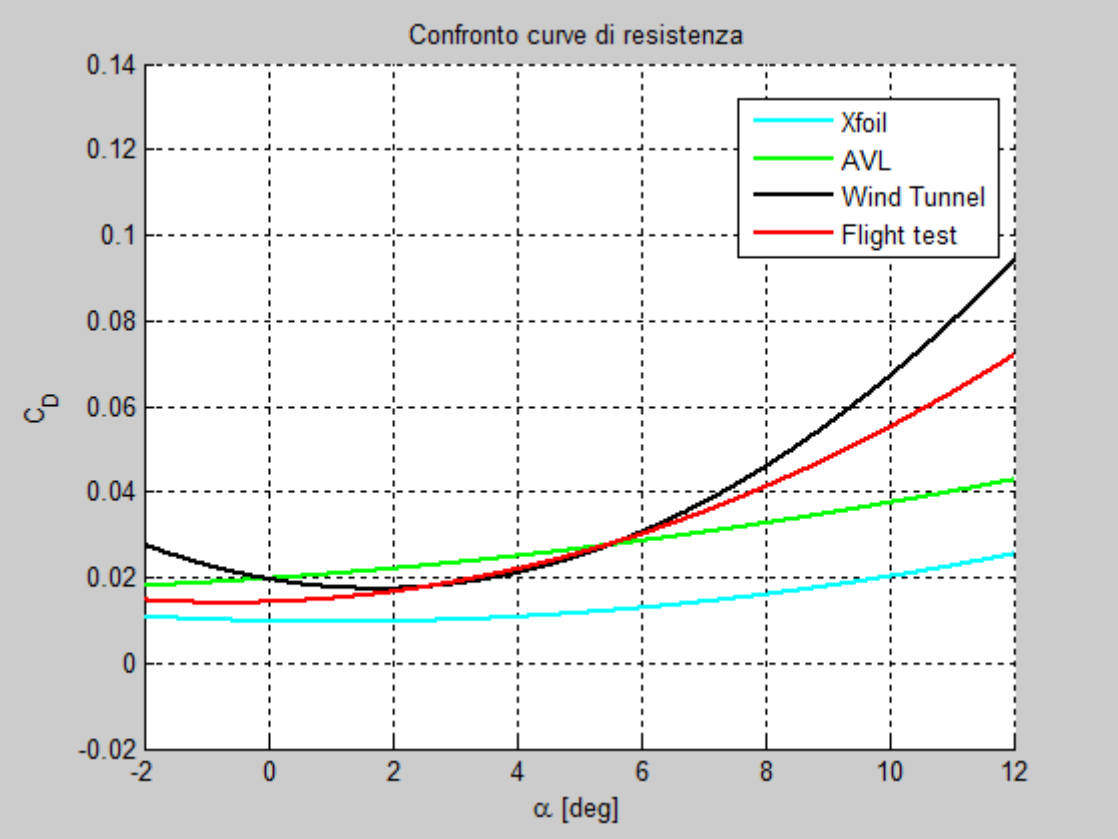


Figure 7.3 – Comparison among all coefficient of drag curves

Looking at the coefficient of drag curves in the figure above, immediately come to the eyes the shape of the parabolas which are quite "closed" for the experimental result of both the flight test and the wind tunnel while very "open" for the two relative to the numerical simulations. For the Xfoil result the low drag is due to the lack of its induced component typical of a finite wing but here again we are only considering a 2D airfoil. For the AVL result, as we said before, the simulation is based on an inviscid flow vortex lattice method and the friction component of the drag is not included. The wind tunnel coefficient of drag was expected to be lower for the single wing than for the complete airplane as in the flight test result; however the drag measurements in the wind tunnel using a balance have always been quite unreliable and the resultant coefficients were overestimated especially at high angles of attacks most likely due to effects of blockage of the wind tunnel model itself or of the test chamber walls as already explained in section 6.5.4 where this last comparison was discussed.

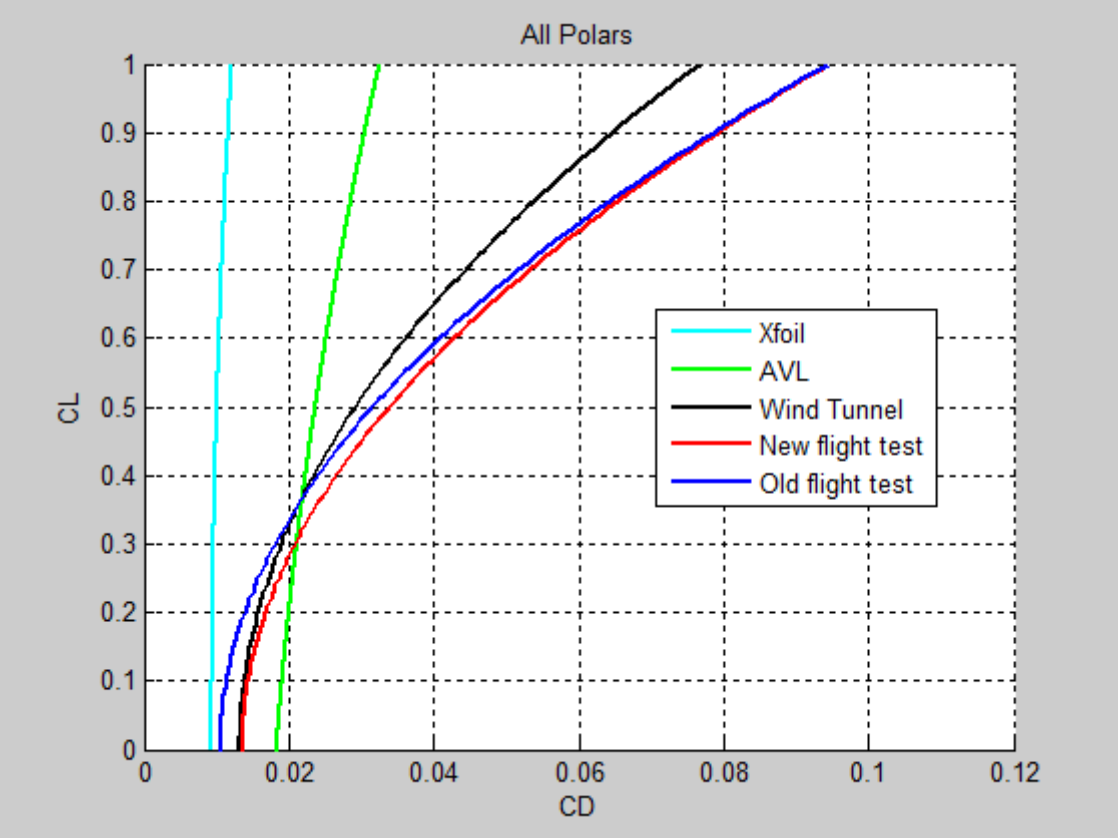


Figure 7.4 – Comparison among all aerodynamic polar curves

Due to very high coefficients of lift and very low related coefficients of drag, the aerodynamic polar curves as computed by the numerical simulations present very high maximum efficiencies (~ 32 for AVL and up to 70 for Xfoil) which are quite large also for the best designed real gliders. For the real cases, the experimental wind tunnel test overall offers as expected a higher maximum efficiency (17.5) than both the old flight test polar (16.9) and the new one (15.1). As we said already, the efficiency of the new experimental flight test polar, compared to the old one, is also paying the extra drag added by the two bigger sub-wings supports + booms + AOA/S sensors which were not yet integrated into the system.

7.3 Future developments

Concerning the flight testing activity aiming to the determination of the characteristic parameters of the aircraft, the system, as it is set up right now, already have all the necessary tools for the development of new, more sophisticated and automated procedures apt to expand the flight envelope considered in this thesis and to identify new performance standards.

However some issues pointed out in the course of this work can be improved and specifically:

- The AOA and AOS sensors have been calibrated one at the time taking into consideration only the angle of attack variation for the AOA vane and only the angle of sideslip for the AOS sensor. We didn't account for the mutual interactions existing between the two vanes at their not directly correspondent angles when mounted on wing. This was due mainly because only the AOA sensor could have been calibrated for a large interval of angles of attack being mounted on wing while for the AOS sensor, the small size of the wind tunnel test chamber and the lack of a proper support, made it impossible to also calibrate this last one while mounted on wing. By designing a new support that could also swing the model inside the test chamber simulating a sweep in angle of sideslip, and maybe also taking into account the dihedral angle we neglected, a much more fine tuning of the AOA and AOS sensors calibration could be accomplished obtaining this kind of functions for their relative output signals:

  $$\text{AOA signal [Volt]} = f(\alpha, \beta)$$
  $$\text{AOS signal [Volt]} = f(\beta, \alpha)$$

- Improve the controller governing the navigation loop we designed in "trimmed mode" for the execution of the trimmed flights test. The data collected showed a quite variable pitch output and maybe this could be corrected by introducing an integrative and/or a differential gain into the loop in addition to the existent proportional one.

- Introduce automatic heading control for those tests which require to keep a certain heading especially if not parallel to the wind direction. As we pointed out several times, the aircraft is very susceptible to the wind and always tends to point towards the wind direction due to sideslip static stability.

As a general conclusion, the most important development of this project would be the increasing of the degree of automation of the flight tests. Reducing to the minimum the actions taken by the pilot needed to accomplish a test flight session from take-off to landing, would surely improve the degree of precision and repeatability of the tests. However, the difficulties found in modifying the original autopilot source code in this sense, suggest to submit the future development of this thesis topic mainly to a graduating computer engineer maybe paired with an aeronautical engineer. Only in this way, in my opinion, the full potentiality of the APM system could be exploited.

Appendix

A – Source code for new flight modes

As showed in the HIL enumerator in the figure below, four custom flight modes were added to the default ones and specifically: TRIMMED (#17), STALL (#18), GLIDE (#19) and CLIMB (#17).

```
50  // HIL enumerations. Note that HIL_MODE_ATTITUDE and HIL_MODE_SENSORS
51  // are now the same thing, and are sensors based. The old define is
52  // kept to allow old APM_Config.h headers to keep working
53  #define HIL_MODE_DISABLED               0
54  #define HIL_MODE_SENSORS                1
55
56  enum FlightMode {
57      MANUAL        = 0,
58      CIRCLE        = 1,
59      STABILIZE     = 2,
60      TRAINING      = 3,
61      ACRO          = 4,
62      FLY_BY_WIRE_A = 5,
63      FLY_BY_WIRE_B = 6,
64      CRUISE        = 7,
65      AUTOTUNE      = 8,
66      AUTO          = 10,
67      RTL           = 11,
68      LOITER        = 12,
69      GUIDED        = 15,
70      INITIALISING  = 16,
71      TRIMMED       = 17,
72      STALL         = 18,
73      GLIDE         = 19,
74      CLIMB         = 20
75  };
```

Figure A1 – Flight modes enumerator

We report here only the C++ code lines inside the set_flight_mode( ) switch function showing the high logic implemented for the four new flight modes:

```
//*** New Flight Modes by FB

case TRIMMED:
    // we have a control loop on altitude hold achieved by changing the pitch using only elevator
    // deflection, target altitude and throttle percent are fixed
     trim_pitch = g.k_delta_e * (g.ft_altitude - relative_altitude());          // (target_alt - current_alt)
     // limitations in attitude change
     if(trim_pitch < -g.ft_pitch_min) trim_pitch = -g.ft_pitch_min;
     if(trim_pitch > g.ft_pitch_max) trim_pitch = g.ft_pitch_max;
```

```
        // autopilot inputs
        nav_roll_cd = 0;
        nav_pitch_cd = trim_pitch * 100;                    // *100 to get centi_degrees
        channel_throttle->servo_out = g.ft_thr;
        // !! Warning: WATCH OUT FOR STALL under minimum throttle !!
        break;

case STALL:
        // tries to keep altitude even if throttle is set to zero
        // this will lead to a stall when pitch angle, increasing with
        // the altitude error, will be too large along with low airspeed
        trim_pitch  = g.ft_k_delta_e * (g.ft_stall_alt - relative_altitude());          // no limitations
        // autopilot inputs
        nav_roll_cd       = 0;
        channel_throttle->servo_out = 0;          // suppress throttle
        nav_pitch_cd = trim_pitch * 100;          // *100 to get centi_degrees
        // !! Warning: activate STALL mode only after TRIMMED mode !!
        break;

case GLIDE:
        // set throttle to 0 %, keeps wings levelled and makes the plane glide
        // keeping the desired pitch angle (ft_glide_pitch)
        nav_roll_cd       = 0;
        channel_throttle->servo_out = 0;          // suppress throttle
        nav_pitch_cd = g.ft_glide_pitch * 100;          // *100 to get centi_degrees
        break;
        // !! Level the APM before take-off !!

case CLIMB:
        // set throttle to 100 %, keeps wings levelled and makes the plane climb
        // keeping the desired pitch angle (ft_climb_pitch)
        nav_roll_cd       = 0;
        channel_throttle->servo_out = 100;          // set max throttle
        nav_pitch_cd = g.ft_climb_pitch * 100;          // *100 to get centi_degrees
        break;
        // !! Level the APM before take-off !!
```

Following the definition of the new constant introduced in the code as they appear on the GCS graphic window. All of them have the possibility to be changed and adjusted through bi-directional in-flight telemetry (the last argument in brackets is the default value of the parameter):

```
// @Param: FT_ALTITUDE
// @DisplayName: FT Altitude
// @Description: The target altitude of the test
// @Units: Meters
// @Range: 100 150
// @Increment: 1
// @User: Advanced
GSCALAR(ft_altitude, "FT_ALTITUDE", 120)

// @Param: K_DELTA_E
// @DisplayName: AltHold Gain
// @Description: The gain we set for the altitude hold loop
// @Units:
// @Range: 0 1
// @Increment: 0.1
// @User: Advanced
GSCALAR(k_delta_e, "K_DELTA_E", 1.5)

// @Param: FT_THR
// @DisplayName: Fixed thr %
// @Description: Value of fixed throttle out in percent
// @Units: Meters
// @Range: 0 100
// @Increment: 10
// @User: Advanced
GSCALAR(ft_thr, "FT_THR", 60)

// @Param: FT_PITCH_MIN
// @DisplayName: Nose down limit
// @Description: Value of maximum achievable down pitch command
// @Units: Degrees
// @Range: 0 45
// @Increment: 5
// @User: Advanced
GSCALAR(ft_pitch_min, "FT_PITCH_MIN", 15)

// @Param: FT_PITCH_MAX
// @DisplayName: Nose up limit
// @Description: Value of maximum achievable up pitch command
// @Units: Degrees
// @Range: 0 45
// @Increment: 5
// @User: Advanced
GSCALAR(ft_pitch_max, "FT_PITCH_MAX", 10)
```

```
// @Param: FT_STALL_ALT
// @DisplayName: FT Stall Altitude
// @Description: The target altitude the controller will try to reach before stalling
// @Units: Meters
// @Range: 100 150
// @Increment: 1
// @User: Advanced
GSCALAR(ft_stall_alt, "FT_STALL_ALT", 150),

// @Param: FT_GLIDE_PITCH
// @DisplayName: Pitch angle GLIDE mode
// @Description: Desired pitch angle value to keep in GLIDE mode
// @Units: Degrees
// @Range: -40 10
// @Increment: 2
// @User: Advanced
GSCALAR(ft_glide_pitch, "FT_GLIDE_PITCH", -2)

// @Param: FT_CLIMB_PITCH
// @DisplayName: Pitch angle CLIMB mode
// @Description: Desired pitch angle value to keep in CLIMB mode
// @Units: Degrees
// @Range: -10 40
// @Increment: 2
// @User: Advanced
GSCALAR(ft_climb_pitch, "FT_CLIMB_PITCH", 2)

////////////////////////////////////////////////////////////////////
```

B – Wind tunnel support executive drawings

Here are reported the executive drawings created using the CAD software *Solid Works*. These drawings refer to the four main parts constituting the rotating wind tunnel model support described in section 4.2.2. Each part has been produced by specialized technicians in the aerospace lab workshop utilizing CNC machines.

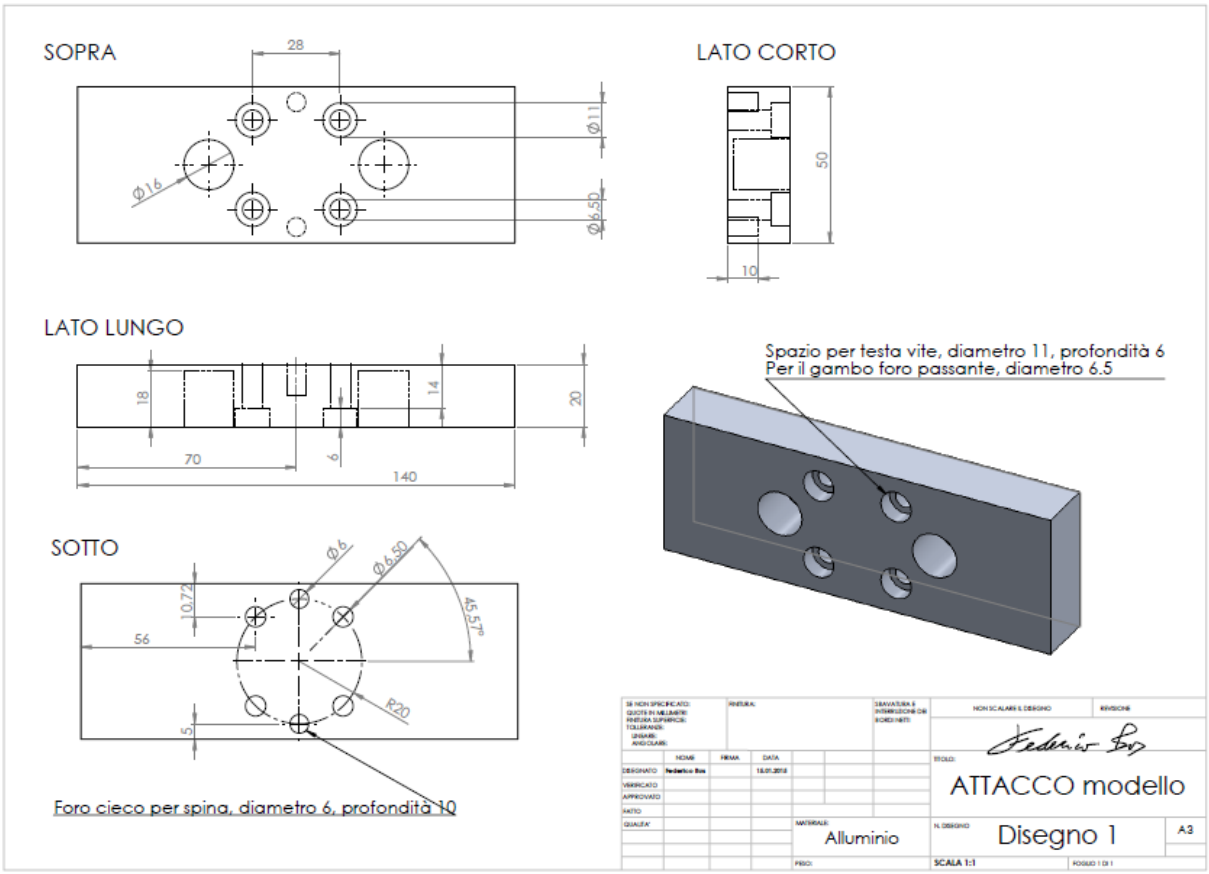


Figure B1 – Model connection plate executive drawing

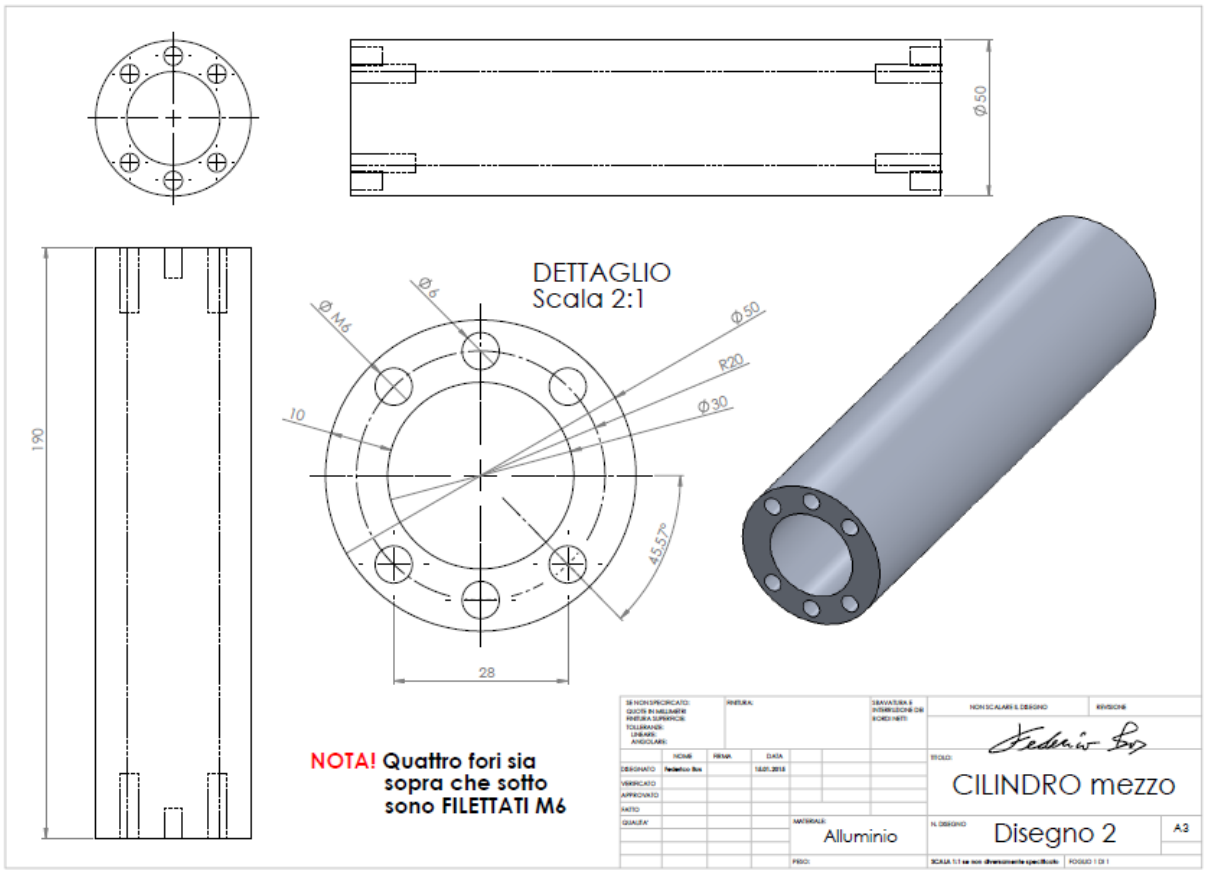


Figure B2 – Cylinder executive drawing

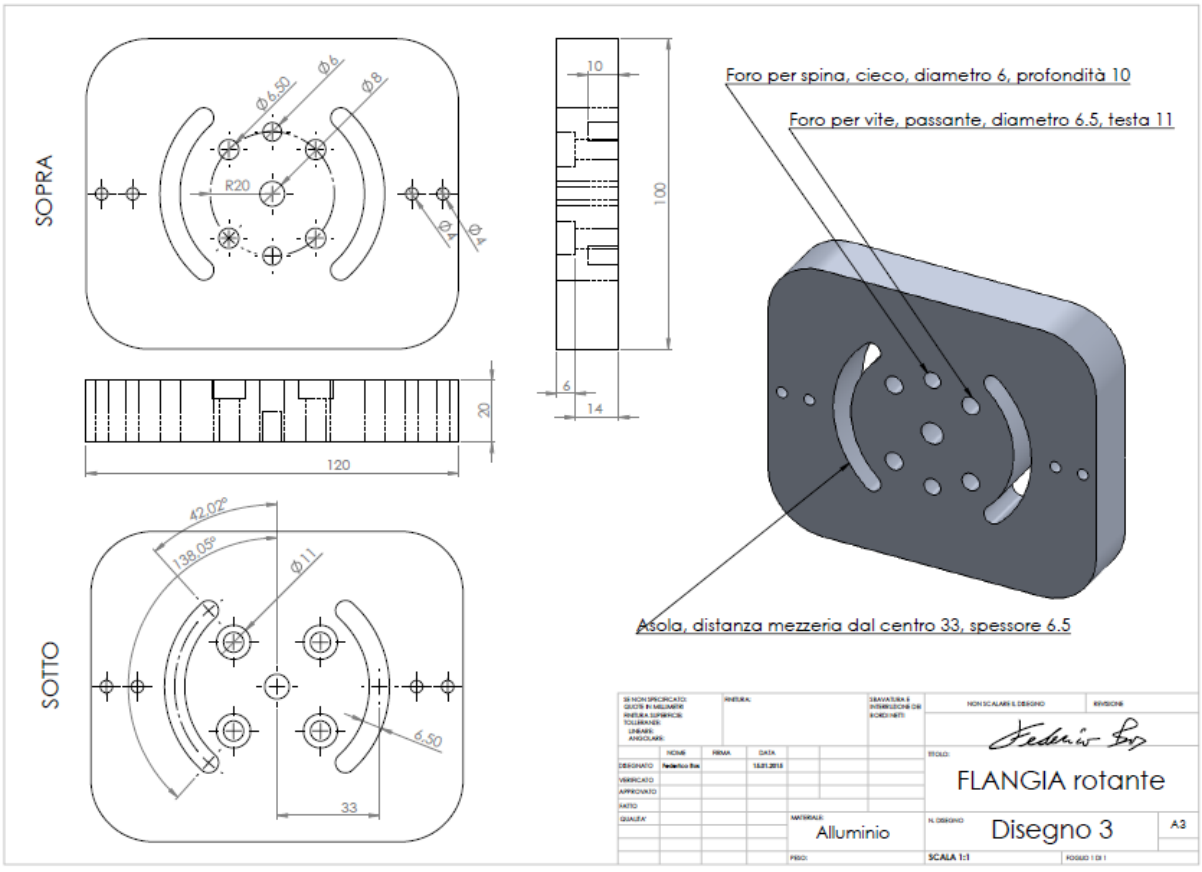


Figure B3 – Rotating flange executive drawing

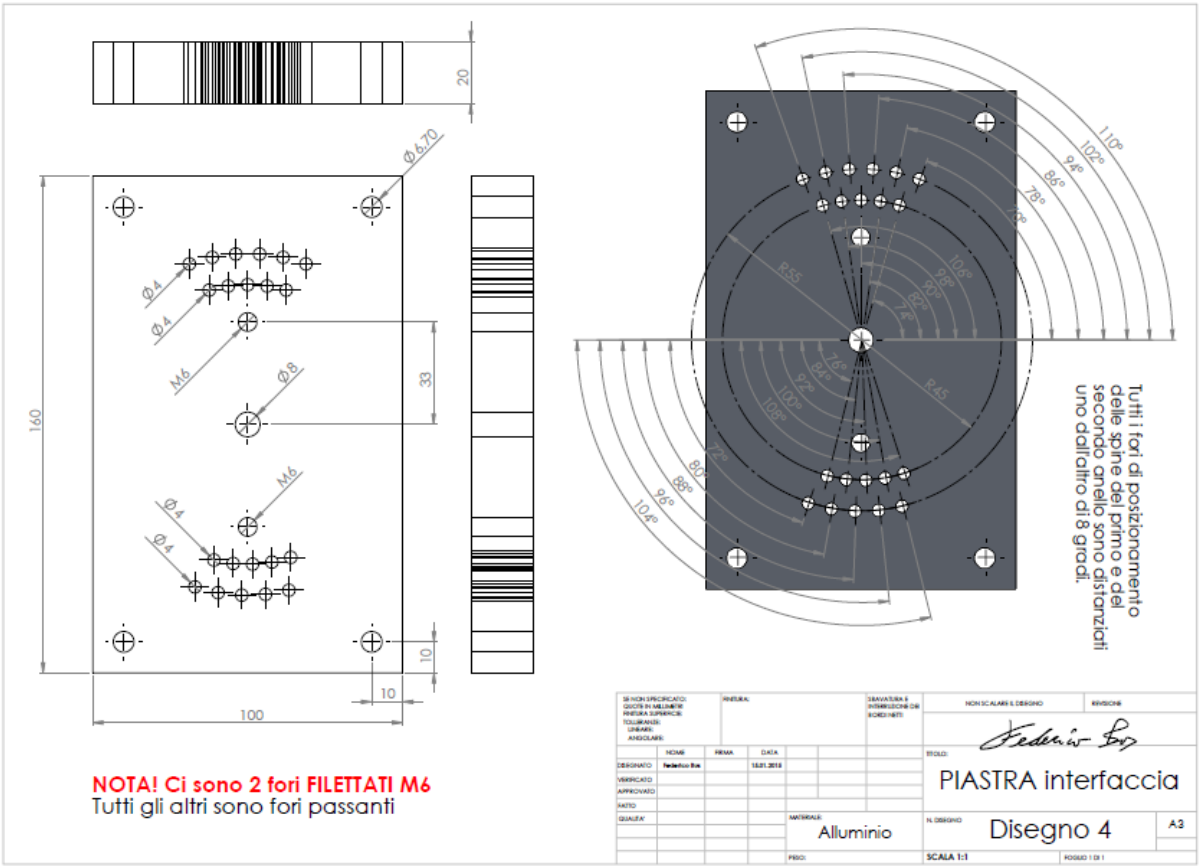


Figure B4 – Balance connection plate executive drawing

List of acronyms

AHRS	Attitude and Heading Reference System
AOA	Angle of Attack
AOS	Angle of Sideslip
APM	ArduPilot Mega
ARI	Aileron-Rudder Interconnect
AVL	Athena Vortex Lattice (software)
BEC	Battery Eliminator Circuit
CAD	Computer Aided Design
CAS	Calibrated Air Speed
CNC	Computer Numerical Control
CPU	Central Processing Unit
CX-15	*Cularis X-flight* 2015
EAS	Equivalent Air Speed
ELR	Environmental Lapse Rate
ESC	Electronic Speed Control
FDR	Flight Data Recorder
FT	Flight Test
GCS	Ground Control Station
GMS	Ground Meteorological Station
GPS	Global Positioning System
GS	Ground Speed
HIL	Hardware In the Loop
IAS	Indicated Air Speed
ISA	International Standard Atmosphere
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
MEMS	Micro Electro-Mechanical Systems
PEC	Position Error Correction
PID	Proportional Integral Derivative (regulator)
PM	Power Module
PWM	Pulse Width Modulation
RAM	Random-Access Memory
RPM	Rotations Per Minute
SITL	Software In The Loop
TAS	True Air Speed
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
WD	Wind Direction
WS	Wind Speed

List of symbols

C_D	Coefficient of drag
C_L	Coefficient of lift
C_T	Traction coefficient
°C	Degrees Celsisus
d	Diameter
D	Drag force
deg	Degrees
E	Aerodynamic efficiency
E_{max}	Maximum aerodynamic efficiency
g	Gravity constant
h	Altitude
h_{ref}	Reference altitude
K	Degrees Kelvin or proportional gain
L	Lift force
M	Mach number
m	Mass
n	Motor/propeller RPM
R^2	Coefficient of determination
adj R^2	Adjusted coefficient of determination
Re	Reynolds number
S	Wing surface
T	Traction force or Temperature
V_{TAS}	True air speed
V_{EAS}	Equivalent air speed
V_S	Stall speed
V_{MD}	Minimum descent speed
V_{BG}	Best glide speed
V_{SC}	Steep climb speed
V_D	Descent speed
V_Z	Vertical speed
V_{FC}	Fast climb speed
V_H	Horizontal flight speed
W	Weight

α	Angle of attack of the wing
α_0	Angle of attack of the fuselage reference line
β	Angle of sideslip
γ_d	Descent angle
γ_c	Climb angle

δ	Control surface deflection
Δh	Altitude interval
Δt	Time interval
θ	Pitch angle
μ	Dynamic viscosity
ρ	Air density
ρ_0	Standard air density
ϕ	Roll angle
ψ	Yaw angle

Bibliography

[1] Lorenzo Trainelli. Lessons presentations, *Flight Testing course*. Traction forces and Stall behavior. Politecnico di Milano. a.a 2014-2015.

[2] Mark Drela. Massachussetts Institute of Technology, Boston, (MA), USA. *Xfoil software* official guide. url: http://web.mit.edu/drela/Public/web/xfoil/ and *AVL software* official guide. URL: http://web.mit.edu/drela/Public/web/avl/

[3] Jewel B. Barlow, William H. Rae, Alan Pope. *Low-Speed Wind Tunnel Testing*, 3rd Edition.

[4] *Arduino project* home page. 2005. url: http://www.arduino.cc/

[5] *APM developers* reference website. url: http://dev.ardupilot.com/wiki/learning-the-ardupilot-codebase/

[6] Outback Challenge. url: https://en.wikipedia.org/wiki/UAV_Outback_Challenge

[7] 3DRobotics on-line store. url: https://store.3drobotics.com/products/

[8] Matlab *Curve Fitting Toolbox* Linear and Nonlinear Regression. URL: http://www.mathworks.it/it/help/curvefit

[9] Dustin Eli Gamble and Andrew Arena. *Automated dynamic propeller testing at low Reynolds numbers*. PhD. Thesis. Oklahoma StateUniversity, 2009.

[10] Claudio Marzo. *Sviluppo e sperimentazione di procedure di prove di volo automatiche per UAV*. Master thesis. Politecnico di Milano, 2014.

[11] W. Durham. *Aircraft Flight Dynamics and Control*. Aerospace Series. John Wiley & Sons, 2013. Cap. 12. isbn: 9781118646793. url: http://books.google.it/books?id=9fwdAAAAQBAJ.

[12] Certification Specifications for Normal, Utility, Aerobatic, and Commuter Category Aeroplanes CS-23 – Book 2: *Acceptable Means of Compliance* (AMC)

[13] Joseph Katz e Allen Plotkin. *Low-speed aerodynamics. From Wing Theory to Panel Methods.* McGraw-Hill Book Co., 1991.

[14] R.D. Kimberlin. *Flight testing of fixed-wing aircraft*. AIAA education series. American Institute of Aeronautics e Astronautics, 2003. Cap. 4 and 16.

[15] Stoliker F. *Introduction to flight test engineering*, RTO-AG 300, Vol. 14 (2005)

[16] J. Roskam. *Airplane Design: Preliminary calculation of aerodynamic, thrust and power characteristics*. Airplane Design. DAR corporation, 2000.

[17] Performance: *Flight Testing* edited by Daniel O. Dommasch.

[18] https://en.wikipedia.org/wiki/Density_of_air#Temperature_and_pressure

[19] Shelquist, R. *Algorithms*. Schlatter and Baker, 2009.

[20] Todd S. Glickman. *Glossary of Meteorology* (2nd ed.). American Meteorological Society, Boston. (June 2000) ISBN 1-878220-34-9.

[21] ISA temperature profile. url: http://home.anadolu.edu.tr/~mcavcar/common/ISAweb.pdf

[22] The History of Drone Technology, url: http://www.redorbit.com/education/reference_library/general-2/historyof/1113154560/the-history-of-drone-technology/

Acknowledgements

In primo luogo vorrei ringraziare il mio relatore di tesi, il professor Lorenzo Trainelli, che mi ha proposto di prendere in mano questo progetto facente parte del *Poli X-Flight* ossia il gruppo di lavoro che comprende tutte le attività connesse con la sperimentazione in volo del Politecnico di Milano. Suo è anche il merito, in tutti i frequenti colloqui e scambi di email in corso d'opera che ci sono stati, di essere sempre riuscito a darmi dei buoni consigli ed anche a trasmettermi un po' di sano entusiasmo.

Per la parte software devo ringraziare l'ing. Federico Rossi "guru" dei microcontrollori e piattaforme open source per il supporto relativo all'integrazione dei nuovi sensori nel sistema e progettazione delle logiche di controllo che a dir suo sono proprio una "stupidata". Per la parte relativa alle prove di galleria del vento la mia stima va all'ing. Donato Grassi che mi ha regalato non poco del suo prezioso tempo per tutto quello è stato fatto, dal primo disegno a mano libera del supporto al post-processing dei dati raccolti in galleria (oltre ad essere venuto a recuperarmi in furgone quella volta che mi hanno abbandonato al campo volo).

Ringrazio i tecnici di laboratorio Antonio e Roberto, il primo per tutte le uscite col furgone e i pomeriggi passati insieme al campo "a far volare i giocattolini" e a mangiare affettato, sempre pronto a fare una battuta e tirarmi su di morale anche nei più lunghi pomeriggi d'inverno passati al computer del laboratorio, il secondo per la parte più tecnica relativa alla realizzazione certosina del supporto di galleria del vento.

Vorrei menzionare qui anche i miei amici e colleghi del Politecnico con cui ho passato molte ore di lezione, di studio e di svago: Andrea Rigo, il Gian Alitta, la triade Mirko-Alessandro-Sebastian, Chris e Davide Berbenni che non si scompone mai a parte quando non partono i motori degli aeromodelli.

Martino! Il mio "copilota" che è venuto da Verona fino a Gaggiano in giornata per aiutarmi con il volo preliminare per la messa a punto del sistema e che stranamente non ha schiantato niente come direbbe il Roby.

Ultimi ma non meno importanti i miei familiari: genitori, sorelle e nonni che mi sono sempre stati vicini. Arianna che mi ha amato e sopportato.