This document describes the development and system integration process of the *SmartGimbal* device developed by *DigiSky Srl*, the *SmartGimbal* is a device that performs the controlling of spatial orientation and photographic parameters of a camera for aerial survey installed into the *SmartBay* platform.
This document describes the development and system integration process of the *SmartGimbal* device developed by *DigiSky Srl*, the *SmartGimbal* is a device that performs the controlling of spatial orientation and photographic parameters for a aerial survey installed into the *SmartBay* platform.
This document describes the development and system integration process of the *SmartGimbal* device developed by *DigiSky Srl*, the *SmartGimbal* is a device that performs the controlling of spatial orientation and photographic parameters for a aerial survey installed into the *SmartBay* platform.

The *SmartBay* platform is a patented invention of *DigiSky Srl* and consists in a general purpose wing pylon able to carry in flight and manage different types of aerial monitoring sensors such as thermal camera, multi spectral camera and so on.
The *SmartGimbal* system provides a structure to integrate a video camera into the *SmartBay* platform and control the pointing by the PAN and TILT motion of the camera and the optical parameters like focus and zoom.
DigiSky had already developed a prototype for this device, but it could not carry out yet for all the function exposed above and moreover it was absolutely unable to perform a mission ensuring a sufficient level of safety and reliability.

This document explains the phases which brought the *SmartGimbal* project from the prototype state to the operative product state, focusing on the technical choices and followed procedures and validation.
The final product consists of a solid metal structure able to host a camera assisted by a motion control system composed by two DC actuators with encoder and transmission chain, this structure can be installed into the *SmartBay* pylon and through it, the *SmartGimbal* and the camera can be interfaced with a control panel located into the aircraft cockpit.
The communication between the console and the *SmartGimbal* pylon is carried out with a wired system using an RS232 serial protocol, two logic boards, one for the console and one for the pylon device;
The link between the boards allows the management of the functions of the overall system providing also information about the system status for the user by a display as interface.
The operator inside the cockpit is able to control the camera orientation using a joystick while the camera parameters are controlled with buttons and knobs on a panel. The airborne camera provides as output an SDI video signal directly displayed on a monitor in order to be managed for the specific mission profile.
At the end of the development, some flight tests were conduced in order to verify the system relaiability and expected functionality.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 DigiSky company

Figure 1.1: The *DigiSky* company logo

Founded in 2007 by the astronaut and test pilot Maurizio Cheli and the engineer Paolo Pari, *DigiSky* operates in the aero photogrammetry sector and in general in the field of the advanced aerial surveying, developing all the necessary avionic devices and systems autonomously [1].

DigiSky has developed a proprietary patented avionic technology named *SmartBay*, this system is a special wing pylon able to install on a general aviation plane wing, several types of sensors providing a fast reconfiguration of the plane in order to accomplish the several mission requirements [2].
DigiSky has also equipped all own aircrafts in the fleet with a common mechanical and electrical interface system, ensuring an extremely fast plane reconfiguration according to the mission to be carried out.

[1] https://www.distrettoaerospazialepiemonte.com/azienda/digisky-s-r-l/
[2] https://www.DigiSky.it/

1.2 The SmartBay platform

The company flagship product is the *SmartBay* platform, it consists in a carbon fiber wing pylon located under the right plane wing, able to carry in flight up to 40 Kg among sensors and devices.

This pylon is equipped with a GNSS sensor and IMU that provide data about the plane trim and position, it has the capability to accommodate under three specific carbon fiber support plates.

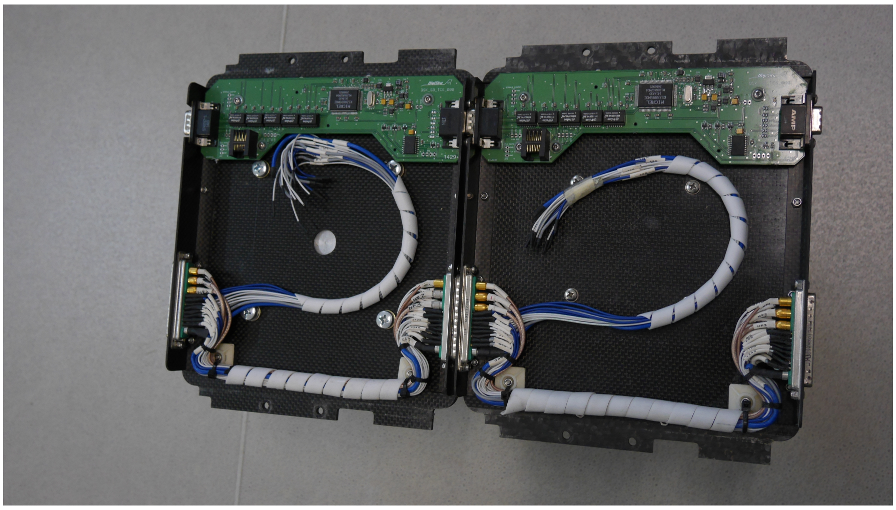

Figure 1.2: Two *SmartBay* plate stacked together

Each one of these plates has two standard connectors at the two opposite sides, allowing them to be simply "stacked" each other into the pylon. This configuration also provides an electrical connection for both power supply and data for the several sensors in order to perform a fast and practical "plug and play" solution, in these photos is possible to observe the *SmartBay* platform with and without a carbon plate installed:

Figure 1.3: The *SmartBay* with and without a plate

Figure 1.4: The *SmartBay* platform with the *SmartGimbal* device installed

Nowadays, this structure is EASA certificated for the installation on the Tecnam P92, an high wing single pistons' engine plane with a wide diffusion in the general aviation.

Figure 1.5: The P92 plane with, on the right wing, the *SmartBay* pylon

1.3 The SmartGimbal device

The *SmartGimbal* is a device conceived for aerial photoshooting allowing the operator and the pilot a quick change of the camera pointing while maintaining the flight attitude, this device is designed to host different cameras in order to extend the operative fields, selecting every time the appropriate camera for the mission requirements.
The *SmartGimbal* project also includes a cockpit control console to manage the gimbal movements and the camera parameters too.

1.4 Thesis goal

The topic of this document is the description of technical choices and procedures followed during the integration and engineering process for the *SmartGimbal*, detailing the constructive and mission requirements that drove the development of the project to the operative state.
It is important to underline the fact that the development of this project was the subject of two other university theses, which allowed the development of the hardware component and the implementation of a first control software, which laid the basis for the definitive control software developed in this thesis project.

The goal of this thesis is the conclusion of the *SmartGimbal* project, leaving open the opportunity for future improvements ensuring a usable and reliable product. To work properly this system need a solid and reliable electronic unit for the functions control and UI, a complete aerodynamic fairing and some mechanical modifications.

1.5 The starting point

The metal structure of the *SmartGimbal* was designed by *DigiSky* then the realization of the components was committed to an external company while the assembly was carried out within the company. The structure was subjected to some test and simulation to obtain the EASA certification, during these tests this system was unprovided of any type of electronics parts.

In 2021 a master degree thesis made by *Mirco Vinciguerra* [3] feed to the project a first sketch of an electronic control system implemented on a wood bench, as shown in these figures:

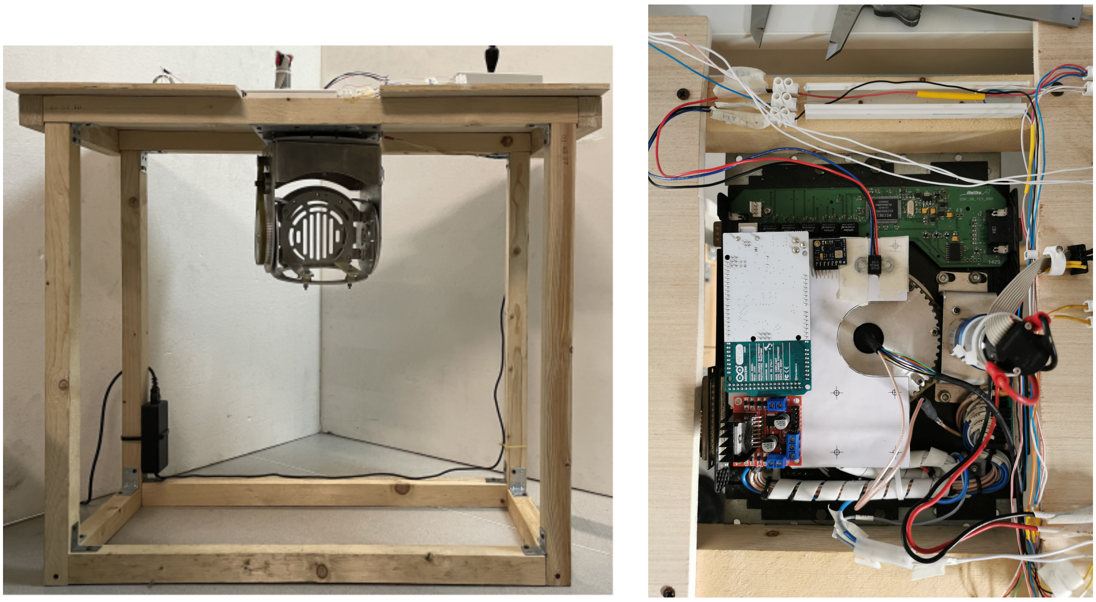

Figure 1.6: The first *SmartGimbal* prototype

[3] Development of a closed-loop control system for an airborne gimbal camera, Mirco Vinciguerra 2021

1.6 Project requirements

Over then the camera controlling, a purpose of this device is to protect the camera from the external airflow guaranteeing a safe handling in PAN and TILT, avoiding the risk of cables intertwining during the rotation;
From the electrical point of view, the system has to provide the electrical power supply and a data communication line for video and non-video data exchange between the camera and the cockpit, making the pilot or operator to be able to see the camera view.

1.7 The Tecnam P92 plane

As previously mentioned, nowadays, the *SmartBay* platform, and consequently the *Smart-Gimbal* project, is EASA certificated to be installed on a Tecnam P92 plane. This high wing plane made by the Italian company "Tecnam Costruzioni aeronautiche", is a solid training aircraft, suitable for all the flight experience levels from amateur to sports piloting. This plane is capable of carrying a pilot and a passenger, that means for *DigiSky*, a pilot and an operator working on the *SmartBay* devices.

The plane airframe is a rectangular aluminum wing and an aluminum fuselage jointed by a solid steel structure that also connect the plane body to the tricycle landing gear. The plane is equipped with a liquid and air cooled 4 cylinders Rotax 912 engine and a two blades variable step propeller fed by two wing fuel tanks with a total capacity of 70 lt.
In the following tables are reported some technical data about the P92 [4]:

Lenght	6,4 m
Wing span	8,7 m
Height	2,5 m
Wing surface	12,18 m^2
Empty weight	325 kg
Maximum take off weight	450 kg

Table 1.1: Tecnam P92 weights and dimensions

Maximum airspeed	223 km/h
VNE	250 km/h (IAS)
Stall speed	63 km/h (Full flap)
Cruise speed	160 km/h
Climb speed	5.9 m/s

Table 1.2: P92 performance

[4] https://it.wikipedia.org/wiki/Tecnam$_P$92

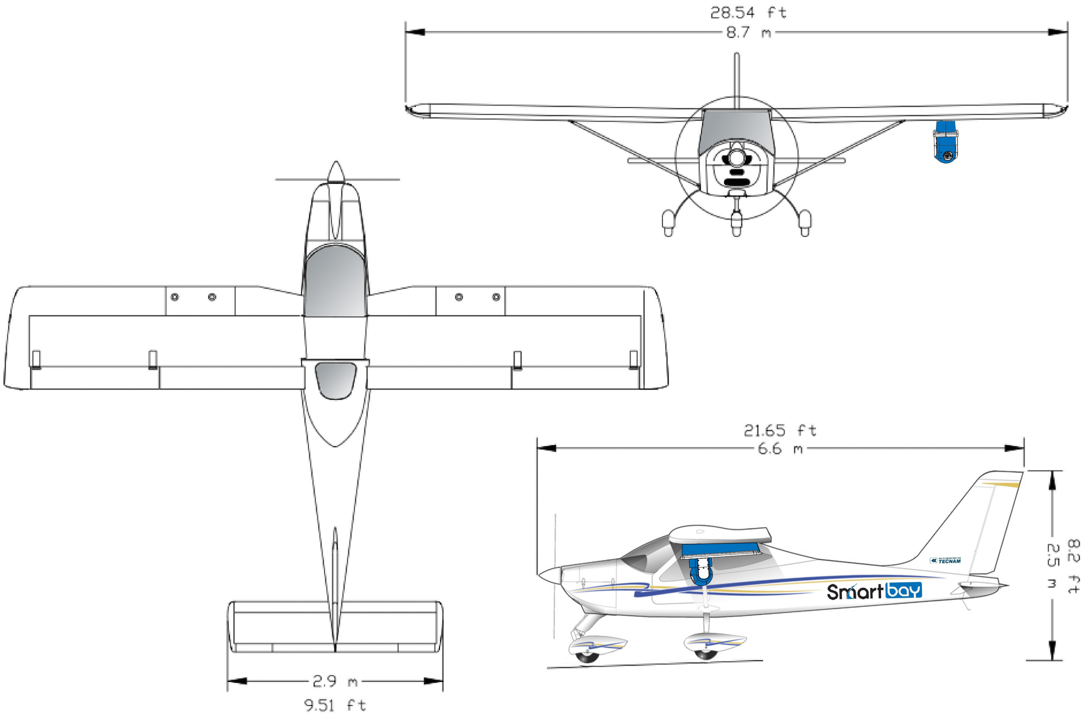

Figure 1.7: Tecnam P92 plane view

1.8 The system main architecture

The *SmartGimbal system* was conceived initially in two blocks:

The first is the core of the *SmartGimbal* that is an aluminum structure containing the camera located under the plane wing, this structure is composed by the chassis with the camera and the motion actuators realized with two DC motors with reducers, coupled with two toothed belts to the PAN and TILT rotation axes.
Speed rotation and position of the two axes PAN and TILT are realized by two quadrature encoders, one per motor, and three Hall effect sensors.

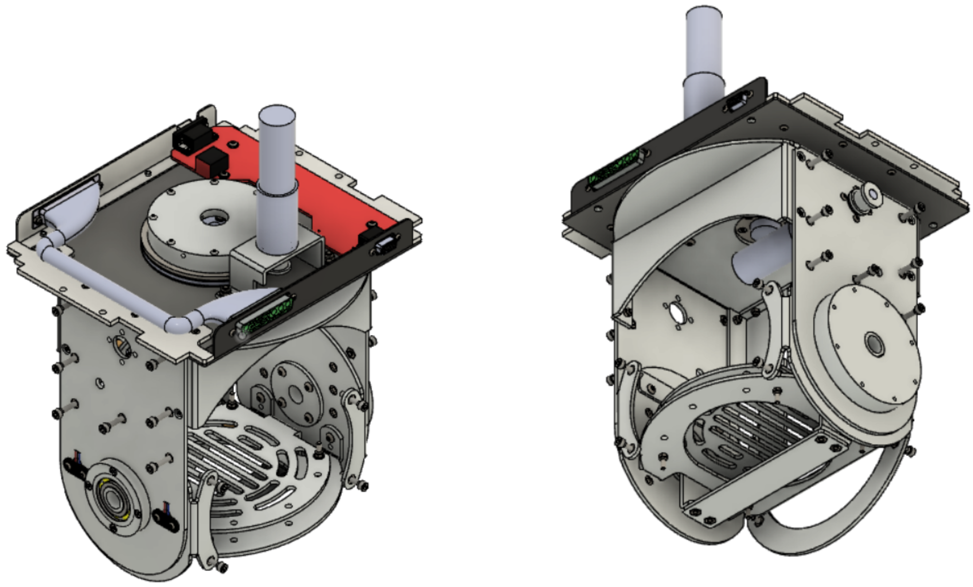

Figure 1.8: Two view of the 3D model of the metallic chassis

The second block is a command console for the operator to control remotely the camera PAN and TILT and other working parameters from the cockpit.
Being the *SmartBay* a general purpose wing pylon, it is equipped with a standardized wiring harness to which all the payloads must be adapted, so they must be compliant also with this harness to permit the connection between the two blocks.
Moreover, the *SmartGimbal* is a system designed to operate on an airplane that is an harsh environment due to the presence of EMI disturbances from the onboard NAVCOM radio and the engine spark ignition with a distance between the two block of more 4 m;

An overall view of the system components location is visible in the following picture:

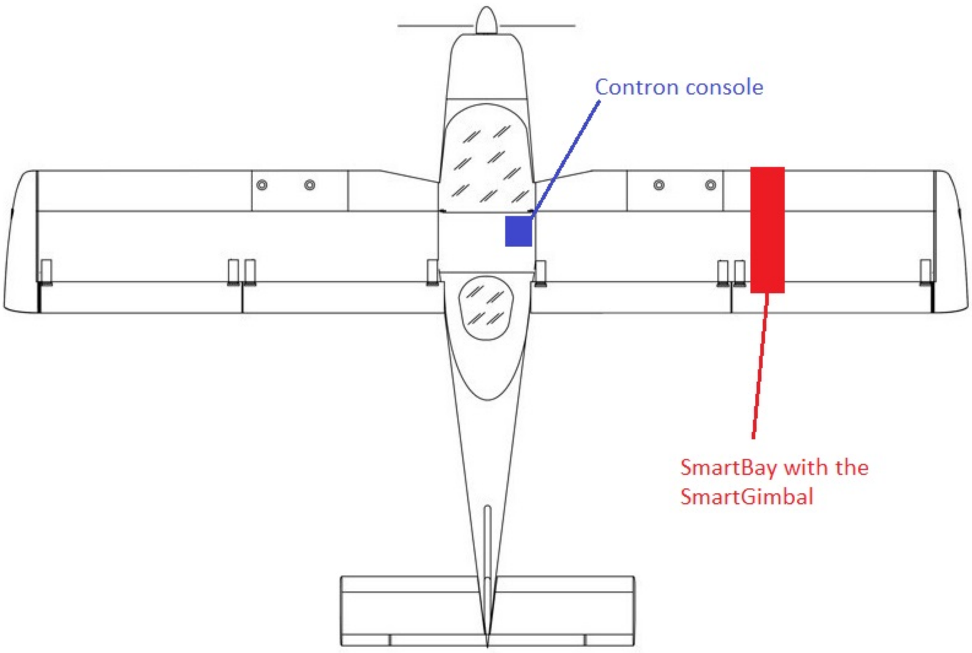

Figure 1.9: The system component's location on the plane

In this conditions to allow both data and power supply connection, an electrically robust communication protocol was chosen.
A wired serial communication line was implemented for the UART, keeping the original interface already partially realized in one of the precedent thesis job.
A lot of improvement were added in the control and communication SW layers, especially to improve the overall system robustness.
The UART interface and power supply rely on 4 GPIO lines available in the *SmartBay* platform connector.
The four lines are used as follow:

N°	**Function**
1	Electric power supply
2	Electric ground
3	UART TX
4	UART RX

Table 1.3: Gimbal and console interconnections

1.8.1 The control software development

Being still a prototype and due to the low computing feature required, the faster and practical solution was to implement the control software using an *Arduino* board, an open source and easy to use hardware platform.
Moreover Arduino offers an extremely user-friendly programming environment and can be assisted from the SW *MatLab-Simulink* developed by *MathWorks*. Using a simple "add-on" package is possible programming the board by the Simulink environment, more simply and intuitively than a standard C coder, allowing everyone to configure the system according to the own mission requirements.
At the end of the project, thanks to the good performance reached, it was decided to keep in the final design the Arduino board.
Among the several available Arduino board models, the Arduino Mega 2560 was selected. It is a valid trade-off between a large number of analog and digital I/O pin, a relatively fast processor and the possibility to work with input voltage signals up to 5 V, a feature not very common in this processor's family.
The reason to prefer a 5 V voltage range is that this is standard working voltage for an extremely wide range of electrical components (included all the ones used in this project), making the system easily interfaceable with other future system extensions [5].

[5] https://store.arduino.cc/products/arduino-mega-2560-rev3

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560, a RISC microcontroller belonging to the AVR family from Atmel [6].
This microcontroller has 54 GPIO digital pins (of which 15 can also provide PWM output), 16 analog input pins, 4 hardware UARTs ports and one I2C port;
The CPU works with a 16 MHz crystal oscillator and can be programmed using directly the onboard USB connector, further technical details about this board are reported in the following table:

Logic operating voltage	5V
Accepted power supply voltage	6-20V
Recommended supply voltage	7-12V
Digital I/O pins	54
Analog input pins	16
DC current per I/O pin	20mA
DC current for 3.3V pin	50mA
Flash memory	256 kB of which 8 KB used by bootloader
SRAM	8 kB
EEPROM	4 kB
Clock speed	16 MHz

Table 1.4: Arduino Mega 2560 technical specification

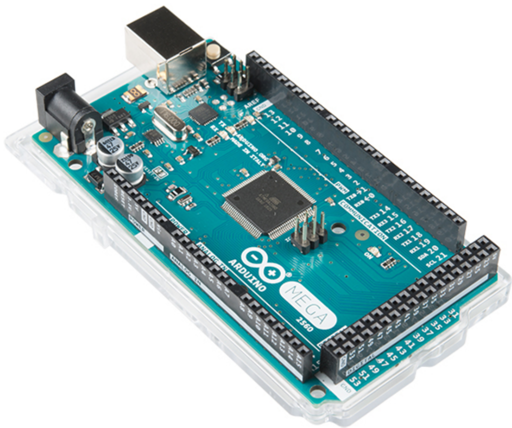

Figure 1.10: The Arduino Mega 2560 board

[6]https://it.wikipedia.org/wiki/Atmel$_AVR$

Thanks to the *Simulink Support Package for Arduino Hardware* add-on, the microcontroller can be programmed using the Simulink blocks, giving to the final program a user-friendly graphical appearance to understand the functions [7].
Moreover, a second add-on feature, the *Embedded coder*, is able to generate automatically the code and download it into the board, with a real time SW parameters tuning.

[7] https://www.mathworks.com/matlabcentral/fileexchange/40312-simulink-support-package-for-arduino-hardware

Chapter 2

The SmartGimbal pylon hardware development

The gimbal structure is the component that contains the camera protecting it from the airflow and is the most important part of the system in terms of safety, being a device that can compromise the plane structure and the flight handling qualities with its weight and location under the wing;

2.1 The supporting structure

2.1.1 Overview

It consists of an aluminum "cage" capable of accommodate cameras weighing up to 3 kg [1], jointed to a *SmartBay* carbon fiber plate like the one showed before.
This metal cage permits the TILT motion of the camera through two balls bearings that allow the rotation of the central part of the structure as shown in the following figure:

[1] Development of a closed-loop control system for an airborne gimbal camera, Mirco Vinciguerra 2021

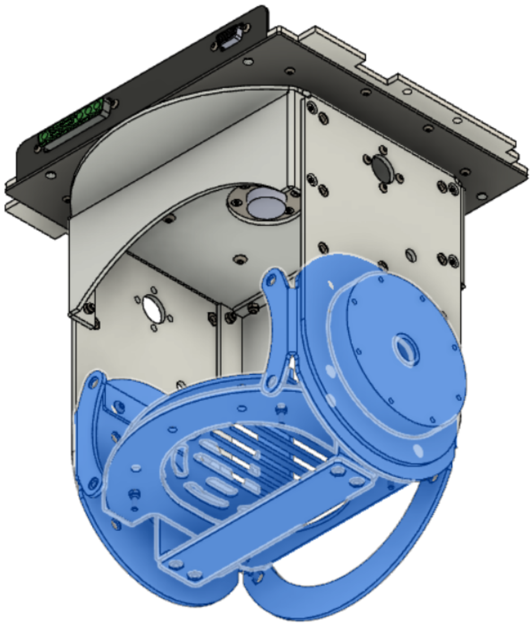

Figure 2.1: The rotation section of the cage

To accomplish the PAN movement, the entire metal cage must be rotated. To do this, the cage is jointed to the carbon fiber plate by a strong ball bearing with a central hole in order to allow the passage of the wiring harness to the camera.
Here is shown the 3D model of the mechanical coupling:

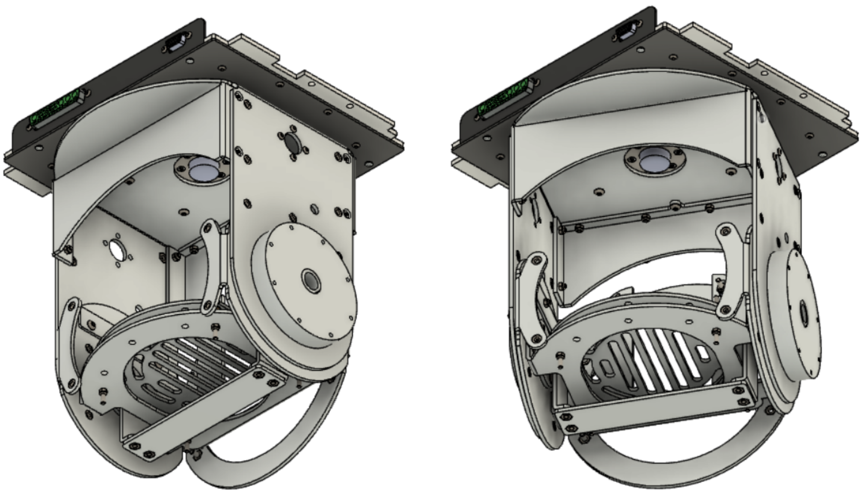

Figure 2.2: The PAN rotation joint

2.1.2 Wiring solution

The motion of the cage could lead to have problems in terms of electrical wiring.
One of the requirements is the complete rotation around the PAN axis, preferably with a continuous rotation in the same direction and the only way to satisfy this, is to create a mechanical separation of the PAN axis rotation, permitting however the electrical connection.
For this purpose special devices called *Slip Rings* are used, like the one shown in the figure below:

Figure 2.3: A slip ring device

In short this device is composed by two plate connected together by a bearing which allows the reciprocal rotation of the parts and to ensure the electrical continuity a special circular contact is used;
This is constituted by a rounded plate with several concentric tracks of conductive material (one for wire) where some brushes (installed on the other plate) can slip on keeping the electrical continuity, this picture better explains the function of this device:

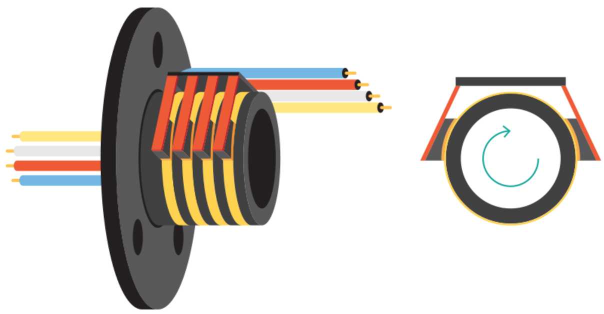

Figure 2.4: A slip ring interior

Initially, the *SmartGimbal* system was conceived to bring the video signal from the

camera directly into the cockpit through an SDI communication protocol, using a shielded coaxial cable that is mandatory to ensure the bandwidth of 270 Mbit necessary for a high definition shot.
Due to the brushes sliding, the *Slip Ring* is not suitable to reach this bandwidth necessary for a high definition signal data transfer [2];
So a special device is needed: a slip ring with 12 traces and a central coaxial cable to route the video signal.
Here is shown the chosen slip ring, a "Servotecnica SRF022-12/1RF":

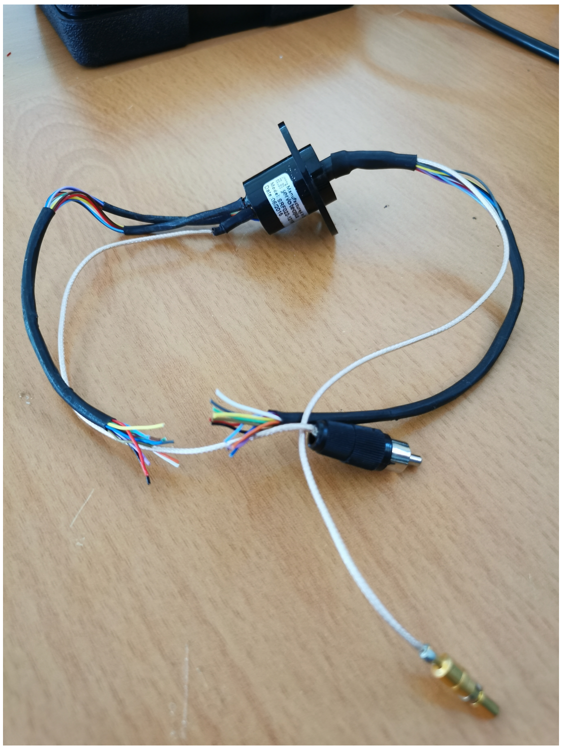

Figure 2.5: The *SmartGimbal* slip ring

[2]https://it.wikipedia.org/wiki/Serial$_{D}igital_{I}nterface$

The slip ring is an integral part of the structure and is located coaxially to the PAN bearing fixed to the carbon fiber plate as shown in this figure (highlighted in blue):

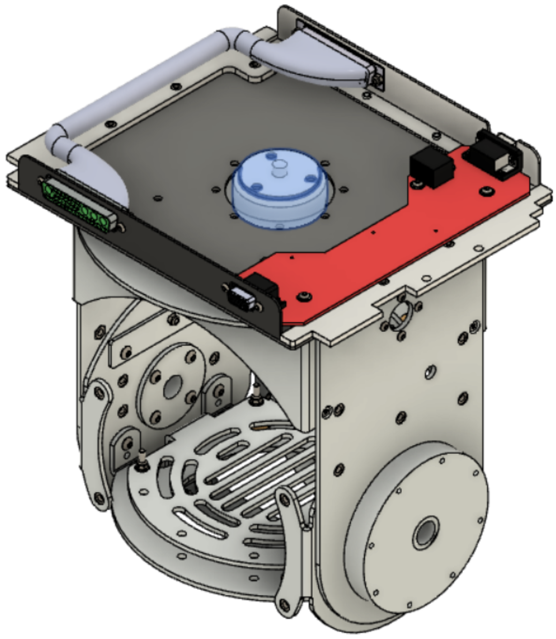

Figure 2.6: The slip ring installed on the structure

2.1.3 The motion system

To allow the movements of the structure on the two axes, two electromechanical DC rotary actuators are used;
These are actuators that can be improperly defined "modular" as them consist of several components assembled together to adapt to different needs;
Being an aerial photography device, the *SmartGimbal* project requires slow and constant movement;
Except for a minimum mechanical friction due to the balls bearings on the rotation axes, there are no other forces capable to generate significant moments.
The center of gravity of the camera is placed approximately at the point of intersection of the two axes of movement PAN and TILT, i.e. in the point such as to generate the minimum possible rotational inertia given only by the distribution of mass of the camera.
Being the angular speeds of movement and the camera masses neglectable, it is possible to consider the actuators in no load condition approximately.
Each of the two actuators is composed by a DC motor coupled to a speed reducer: a "Faulhaber 2642-012 CXR" motor with a "Faulhaber 30/1 159.1" reducer with a 159/1 ratio [3], [4].

Figure 2.7: The Faulhaber 2642-012 CXR DC motor

Nominal voltage	12 V
No load RPM	6400 RPM
Stall torque	132 mNm
Torque constant	16.9 mNm/A
Max efficiency	78%

Table 2.1: Faulhaber 2642-012 CXR data

[3] https://www.faulhaber.com/it/prodotti/serie/2642cr/
[4] https://www.faulhaber.com/en/products/series/301/

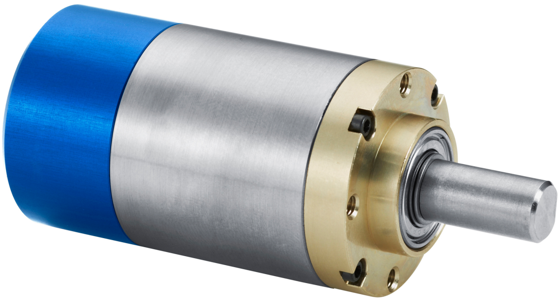

Figure 2.8: The Faulhaber reducer gearbox

Continuous torque	4.5 Nm
Peak torque	6 Nm
Maximum efficiency	60%

Table 2.2: Faulhaber 30/1 159.1 data

On the DC motor shaft there is also a quadrature encoder coupled, a "Faulhaber IE3-256L".

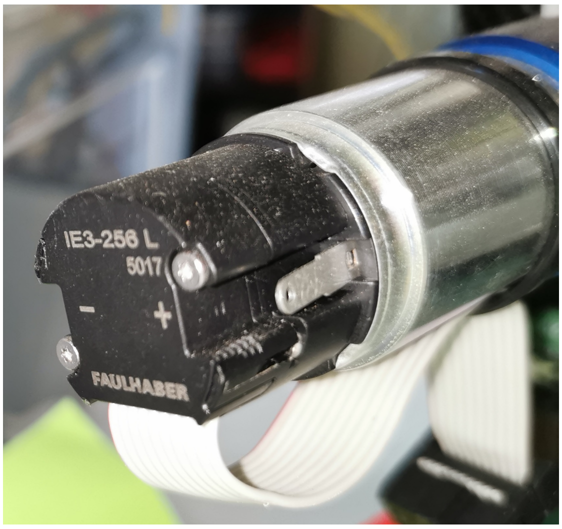

Figure 2.9: The quadrature motor encoder

Once assembled, the actuator has a tubular shape as shown in this figure:

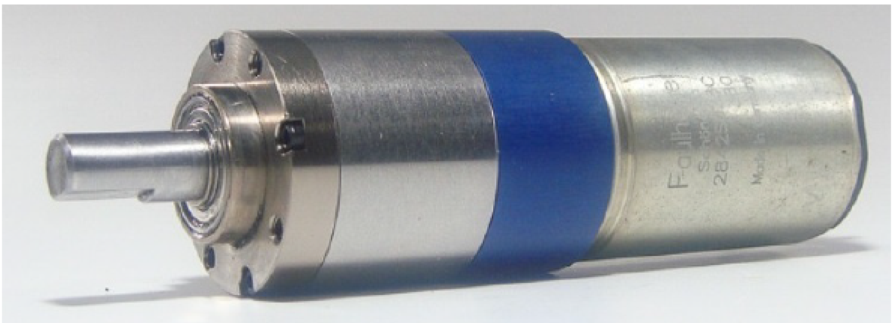

Figure 2.10: The complete actuator

The two actuators are connected to the rotating part of the structure with two wheels and toothed belts adding a further reduction of 6.25/1, reaching a total reduction ratio from the motor to the structure axes of 993.75/1.
Here are reported some data about the "final actuator" inclusive of both the reducers (the one coupled with the motor and the toothed belt) referred to the rotation of the axes:

Continuous torque	29.25 Nm
Peak torque	39 Nm
Maximum RPM	6.2
Maximum efficiency	46.8%

Table 2.3: Complete actuator chain data

2.1.4 Structure and motion system merging

TILT actuator In this picture is visible the location of the actuator inside the structure:

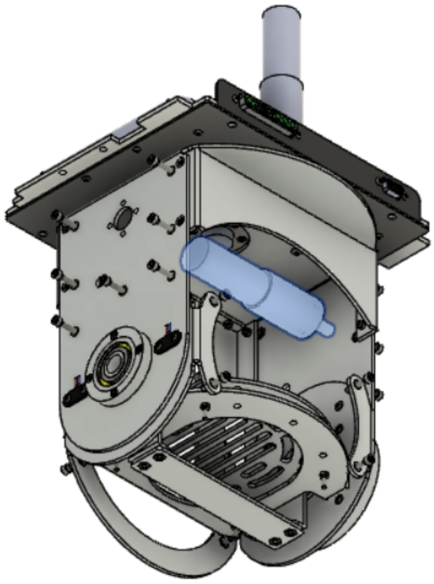

Figure 2.11: The TILT actuator inside the structure

PAN actuator The toothed belt coupling, forces the PAN actuator to have the rotation axes parallel to the PAN structure rotation axis;
Due to the wiring and circuit presence on the carbon fiber support plate of the *SmartGimbal*, the positioning of the actuator must be in front or rear of the PAN axis (respect to the plane direction), this position is also constrained by the particular shape of the *SmartBay* wing pylon:

The bottom section of the pylon has a rectangular shape in order to receive the carbon fiber plates on two special rails, highlighted in blue in the picture, while the remaining part of the pylon has an airfoil shape in order to reduce the air drag as shown in the pictures below:

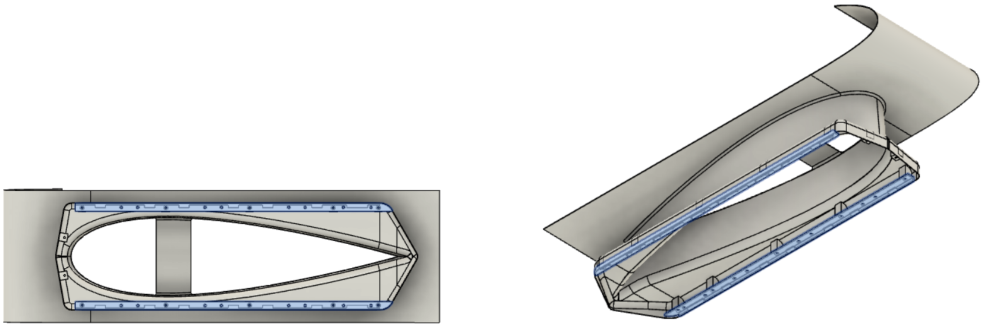

Figure 2.12: The *SmartBay* carbon wing pylon

A lateral positioning of the actuator is clearly not allowed, furthermore behind the PAN axis is located a router used for the Ethernet connection between the *SmartBay* and the cockpit so, the only allowed position for the actuator is above the PAN axis as shown in this figure:

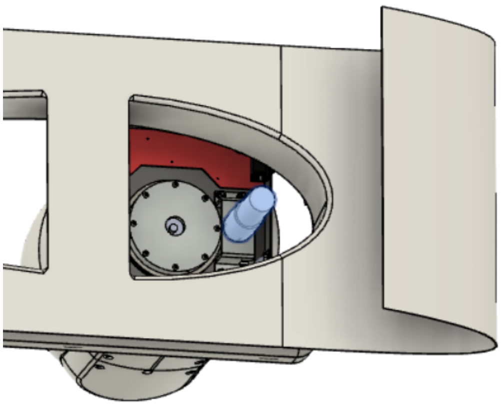

Figure 2.13: The PAN actuator positioning

The actuator is supported by an "omega-like" aluminum bracket that fix it to the carbon fiber plate.
This particular configuration of the top section of the support structure, involves also that the only available position for the *SmartGimbal* plate is in the slot closest to the wing leading edge, this in terms of flight safety and handling qualities, brings an important advantage shifting the center of gravity towards the leading edge and thus ensuring a slightly swooped attitude of the aircraft.

2.2 Electronic board and wiring development

2.2.1 Overview

As previously mentioned wing pylon structure was designed to receive and implement commands coming from the control console inside the cockpit, to accomplish this task some electronic control is needed.

In particular the required tasks for the wing pylon structure electronic system are listed here:

- Receiving and decoding of the commands coming from the cockpit console
- Controlling the structure movements, monitoring its position and speed
- Providing power supply to the camera
- Controlling the camera optical parameters according to the operator commands
- Providing feedback to the console about the structure status

Before developing any circuit and electronic system configuration is important to know what the "flying plane environment" and especially the *SmartBay* environment offer in terms of electrical interface.
The platform offers a wide range of available power supplies with two of the most common working voltage such as 5 V and 12 V.

Concerning the data signals, the *SmartBay* is equippeded with two shielded coaxial video data lines available but doesn't have a specific line for the UART communication used for this project, but it is possible to use the six GPIO lines integrated in the *SmartBay* platform wiring harness.

2.2.2 Definition of the blocks

Once the *SmartBay* available supply voltages are known, to satisfy all the requirements of the project it is useful to decompose the electric system in singles functional blocks, every one with a specific task. See the following list of all necessary functional blocks:

- Power supply
- Main controller
- Actuators controller
- UART management unit
- Camera optical parameters controller

In this figure are shown all the theoretical interconnections between the various components to better explain the system architecture:

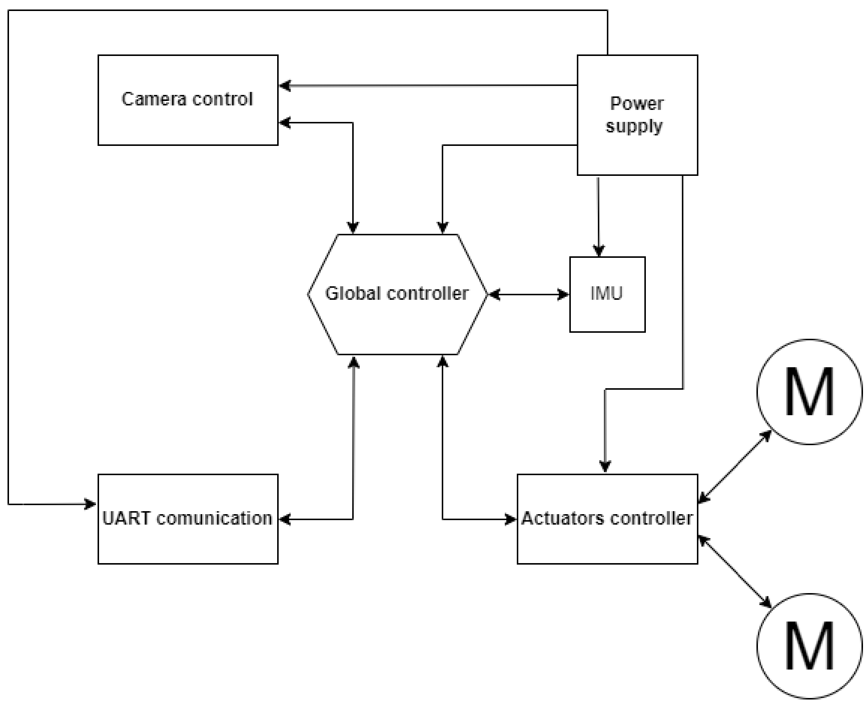

Figure 2.14: Functional scheme of the wing pylon structure electric system

2.2.3 Power supply

The power supply block provides electric power supply to the gimbal structure and to the cockpit console, this last consideration will be discussed better in the *System components merging and ground tests* chapter.
It's important to know, to design this block, the power requirements of each component of the circuit, starting from the actuators and moving to the others with lower current consumption.

Actuators absorption calculation A simple way to know the actuator current absorption, having already the actuators available, is to measure the maximum current needed to move the system simulating a friction induced by external factors. The operating voltage of the actuator is up to 12 V and having available this voltage level on the plane, this one was chosen. This value also guarantees a relatively fast motion of the structure.

To measure the maximum actuator current absorption, the simplest and most effective method was to provide the actuator a constant 12 V power supply and then trying to stop it manually.
Due to the remarkable actuator torque, it is impossible to block the actuator without break some system components, so was set a reasonable maximum absorption of 1 A before the stalling point to preserve the mechanichal integrity of the system;
The current value chosen corresponds to a moment generated on the relative axis of the structure (therefore down to all the reductions ratios) of 54 Nm.

This torque can only been reached if the movement of one of the actuators is prevented by some seizure coming from external phenomena of the system, for example, consequently to a The condition of both actuators blocked is quite rare, is more plausible to have for example a deformation of the structure, having two actuators inhibited but not completely blocked, anyway it was decided to adopt a maximum total current value for both the actuators equal to 2 A, resulting in 24 W power.

Console absorption calculation The control console does not contain power electronics and the console component that needs higher current is the LCD with an absorbtion of 0.16 A @ 5 V [5], this will be discussed better in the *SmartGimbal cockpit console hardware development* chapter.

The others components of the control console are an Arduino Mega 2560 board and some control electronics, which in total need a current of about 0.1 A.
The console operates at 12 V power supply, the reason of this choice will be explained in the *SmartGimbal cockpit console hardware development* chapter. Globally, the cockpit console needs 12 V of power supply with a current of about 0.26 A.

[5] https://forum.arduino.cc/t/lcd-current-draw/246467/3

Camera controller and others The remaining parts are digital circuits working at 5 V with a total absorption of 120 mA.

The power supply block will also have to provide electric supply to the camera, due to the fact that the embedded camera battery was removed being the batteries typically unreliable in severe temperature environment (the gimbal under the wing can pass from 40° C to the ground up to 0° C at high flight levels in short time),
The camera has a dedicated port for an external 12 V power supply.
Totally the camera and the remaining parts of the circuit in this block require two power supplies: 12 V 1 A and 5 V 0.12 A.
The power needs for the *SmartGimbal* are summarized in this table:

Voltage	**Current**
12 V	3.26 A
5 V	0.12 A

Table 2.4: Required electric power supply

2.2.4 Actuators control block

The purpose of this block is to drive the actuators accordingly to the global controller commands;
In order to perform a precise and fluid rotation of the camera, it's mandatory to have actuators able to move at variable speeds, gettable by a driving PWM command.
The used controller is unable to provide signals with enough current to directly drive the actuators, an amplifier is needed. To control a DC current motor in both rotation verses a so-called *H-bridge* circuit is used.
Following is represented how a H-bridge works:

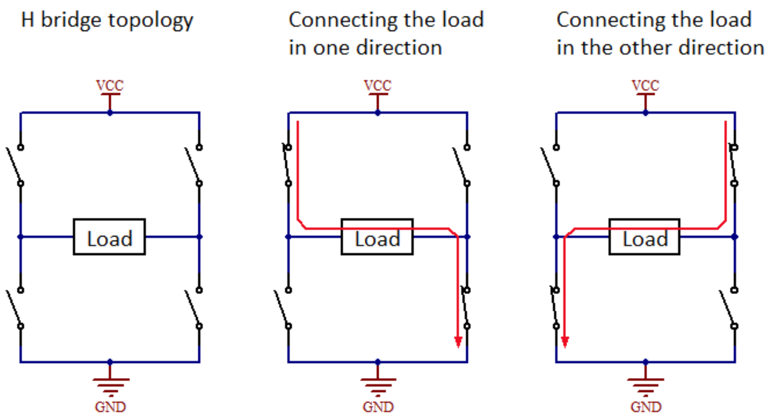

Figure 2.15: H bridge circuit

With two actuators, two of these stages with the relates control electronic are used;

This electronic control is mandatory to guarantee the correct sequence of activation of the switches to avoid "leg short circuit" or damages in case of software bug.
This configuration is very used, so integrated electronics ad hoc modules have been developed with one or more H-bridge in one device, including also the controls and protections.

In order to simplify the circuit and the assembly, it was decided to use a smart power IC integrating two H-bridges: a L298N based module was chosen, shown in the figure below:

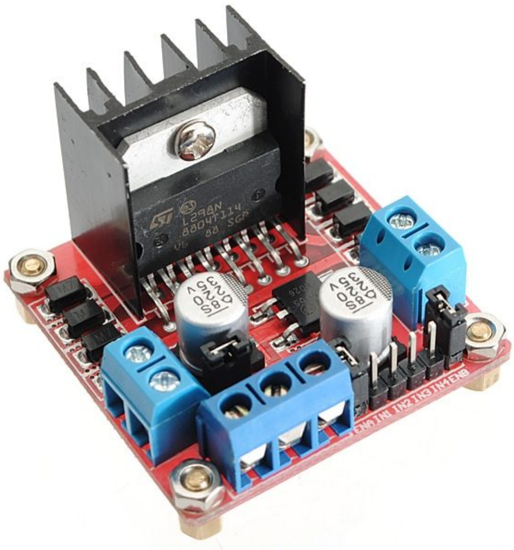

Figure 2.16: The L298N module

This module operates from 3 to 30 V and can provide to each actuator a continuous current of 2 A.
The global controller can drive this device by 2 command pins per actuator, providing the two PWM control signals for the two rotation directions, clock wise and anti-clock wise and speed.

2.2.5 UART management block

As anticipated in the chapter 1, the entire system was conceived to make all the components communicate through a UART protocol.

This method of communication between the *SmartGimbal* structure and console was already implemented in the first prototype and worked well but with some problems.
The UART protocol on the first prototype was used to allow the data exchange between two microcontrollers using as communication line an extremely short cable (less than 1 m) and functioning in a "laboratory environment", so without the presence of electromagnetic interference generated by other devices.

The complete system will work in a much more harsh environment characterized by the presence of strong electromagnetic disturbance generated by the aircraft engine, by the plane on board electronics and also by any other sensors present on board the *SmartBay*. Moreover the UART line has to pass inside the plane wing structure with the other cables, in order to reach the cockpit, crossing in this way a notable distance of about 5 m. As mentioned in the chapter 1, there are no UART dedicated cables into the *SmartBay* platform, so two GPIO lines were used (TX and RX).
Using not shielded cables and working in the condition exposed previously, is difficult to guarantee a data exchage without errors.
To improve the *signal-to-noise* ratio, it is a good approach to adopt a stronger protocol in terms of data integrity.

The choice was to improve from a TTL UART interface to a powerful one, the RS232 interface that, albeit of an old conception, is very robust and extremely diffused.
To interface the microcontroller UART with the RS232 line, are available dedicated components in the market. In order to simplify the circuit, it was decided to use a MAX232 that is a RS232-UART bidirectional converter with 5 V power supply.

Figure 2.17: The RS232-UART module

2.2.6 Camera control block

There is not much space inside the metal cage to receive all the market available cameras, only relatively compact cameras can be used in this system.

A camera already available in the *DigiSky* company sensors suite, employed for other purposes but compatible with this system, is the *BlackMagic Design Micro Cinema Camera* shown in the figure below:

Figure 2.18: The BlackMagic Design Micro Cinema Camera

This extremely compact camera was design to be used for aerial shots using drones, indeed this camera can only be controlled by an external interface.

Different communication protocols can be used to operate with this camera, one of this is the SBUS protocol developed by Futaba. This protocol was initially conceived for RC airplane models in order to get a more powerful and simple communication compared to the traditional PWM one to manage the several servomotors present in plane models. It consists (differently from the NRZ levels of the UART) in a non-inverted mono-directional serial bus working with a 100K bps of baud rate, and capable to carry on up to sixteen proportional servomotors channels and two on-off servomotors channels [6].

Some tests have been made to generate this signal directly by the microcontroller, but it was complex to manage at software level and presented quite a few problems related to the signal timing, causing this signal to not be interpreted correctly by the camera often.
Consequently, was decided to adopt a less refined but more reliable solution, being the SBUS a quite diffused communication protocol in the RC airplanes model field, there are

[6]https://github.com/uzh-rpg/rpg$_{q}uadrotor_{c}ontrol/wiki/SBUS - Protocol$

in the market a lot of preassembled modules capable to interface this protocol with the more common PWM system.

The PWM signals in the field of airplanes dynamic modeling and in general in dynamic modeling applications, is a standard used to control servomotors, ESC and so on, it consists in a 50 Hz TTL PWM signal with a positive variable duty cycle between 5 and 10% as shown in this picture:

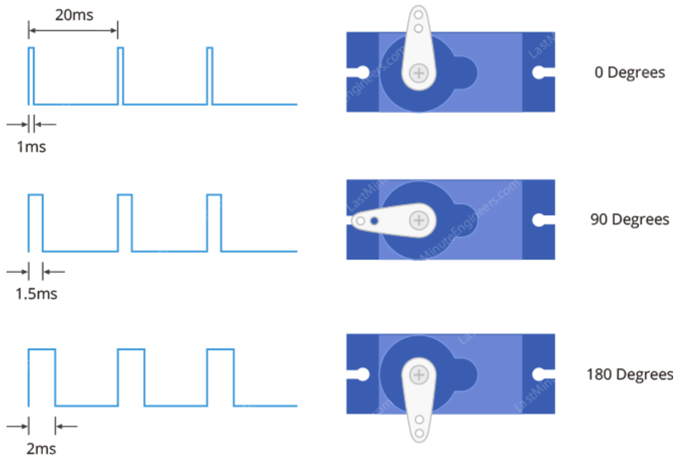

Figure 2.19: The servomotor standard control signal

This signal is easy to be managed and moreover, Arduino and Simulink, offer software libraries for use in this type of modulation.

The solution adopted consists to generate as many servo signals as are the camera functions to be controlled (differently to the SBUS, the PWM modulated system needs a servo signal for each function) Using the microcontroller and then merge and convert these PWM signals into a single SBUS signal with a PWM-SBUS converter. The generated SBUS signal will be used to control the camera.

The converter module used is shown in this picture, is a converter based on an STM32 microcontroller made by Jhemcu.

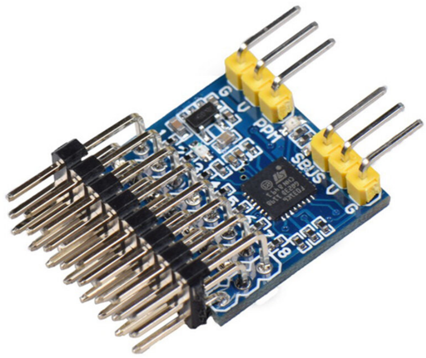

Figure 2.20: The PWM-SBUS converter

This module offers up to eight servo PWM channels as input, to control completely the camera (for the desired controls) five channels are enough.

IMU expansion

Having already available this architecture, it was decided to prepare hardware blocks for some other new functions of the *SmartGimbal*, one of these is the pointing keeping on a target with the aircraft in motion. The functionality expressed in this way, can be misleading, it is not a target pointing driven by an "image recognition process" because it would require dedicated equipment with video data processing capabilities.

The main idea of this functionality is, once the camera is correctly oriented, to maintain the orientation by counteracting the movements and the attitude change of the aircraft.

To obtain the aircraft attitude data, a **MEMS IMU** is used, in particular a Xsens MTi-1-0I-T shown in the figure:

Figure 2.21: The Xsens MTi-1-0I-T IMU

This is a high performance IMU with the following features:

Gyroscope	
Standard full range	2000 deg/s
Bandwidth (-3 dB)	255 Hz
g-sensitivity	0.001 º/s/g
Accelerometer	
Standard full range	16 g
Bandwidth (-3 dB)	324 (x, y) 262 (z) Hz
Electrical	
Power supply voltage	2.19 to 3.6 V
Power consumption	44 mW @ 3V

Table 2.5: Xsens MTi-1-0I-T specification

The *SmartBay* platform already integrates this IMU within its control electronics, its data are merged with the airspeed data coming from a pitot sensor present on the *SmartBay* pylon, into a single data signal transmitted via an RS232 interface to all the payloads connected on the same TX line, this signal is called *"NMEA fligh data"*.
In order to program and test the *SmartGimbal* system on a test bench (without the data coming from the plane), a socket capable to accommodate the IMU sensor was installed on the gimbal electronic board.

The NMEA flight data indeed are taken from a bus present on the *SmartBay* communication line that makes the flight data available for all the payloads installed on the wing pylon, these are decoded through another RS232-UART module like the one used to communicate with the cockpit console.

2.2.7 Global controller

As the name suggests, the purpose of this block is to control the wing pylon structure, taking in input the data from sensors and providing in output GUI information and actuators commands.
As mentioned in the chapter 1, the *SmartGimbal* was designed to have as controller an Arduino Mega 2560 board, this board is under-exploited here due to the relatively few inputs and outputs to manage, however this leaves possibilities for future system expansion by adding other devices or sensors.

The data in input to the controller with their relative characteristics are listed in this table:

Input	**Characteristics**
PAN quadrature encoder	2 TTL input channel
TILT quadrature encoder	2 TTL input channel
PAN motor control	2 TTL PWM output channel
TILT motor control	2 TTL PWM output channel
Camera control	5 TTL input channel
Hall sensors	3 analog input channel
Console UART	1 RX-TX UART port
NMEA flight data	1 RX UART port
Status-debug LEDs	2 TTL output channel

Table 2.6: Controller board input and characteristics

2.2.8 Electronic design

The electric schematic was designed using the **Eagle CAD** provided by **Autodesk**; The circuit is quite simple, over the connections of the various blocks, it includes the power supply protections and two "debug LEDs".
In the circuit there are two supply voltages, 5 V and 12 V both protected with two *self-resetting fuses* respectively of 0.5 A and 4 A.
The 5 V power supply provided by the *SmartBay* is a stabilized 5 V line that can be used as direct power supply of logic circuits while the 12 V voltage, coming directly from the plane engine generator, needs some precautions;
In order to protect the electronics from ESD and overvoltage a RBO-40 TVS diode is used, this component contains also a reverse voltage protection diode.
Most of the used device work with a 5 V voltage, while the IMU needs a 3.3 V feed by the internal power stage already available in the Arduino board.

The IMU communicates by a I2C protocol that works with an SCL signal and an SDA signal. To rescale the voltage range of the IMU I2C lines from 0-3.3 V to 0-5 V voltage range acceptable by Arduino, a *level shifter* is needed.

Here is shown the electrical schematic:

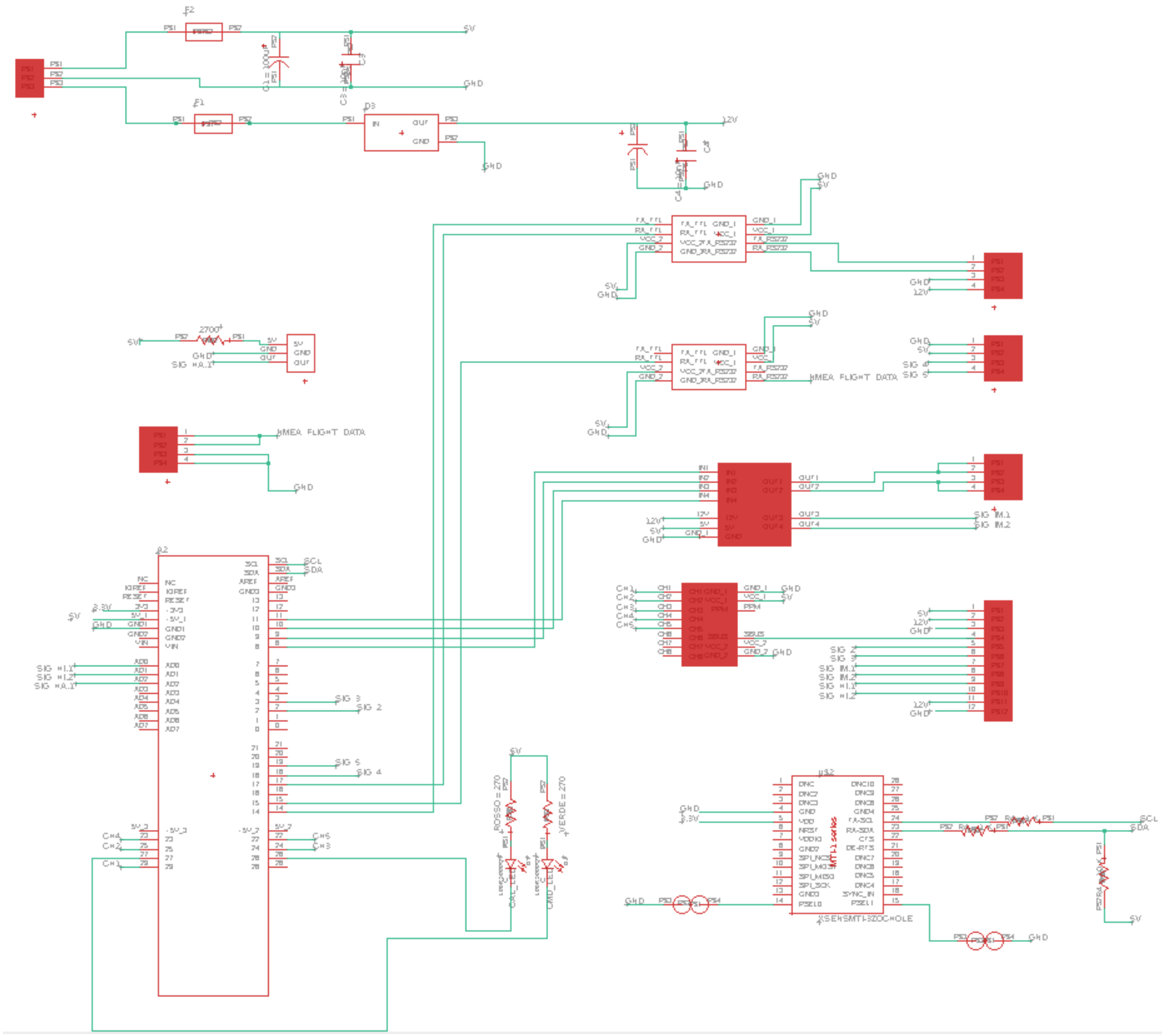

Figure 2.22: The gimbal electrical schematic

2.2.9 PCB design

All the parts were assembled on a single PCB designed with the same *Eagle* suite; The PCB is placed on top of the PAN axis main gear, to exploit it and to install a position Hall sensor, it was decided to place the PCB extremely closed to the toothed gear. The position Hall sensor, a "LITTLEFUSE 55100-3M02A", is used to identify a specific position of *SmartGimbal* during the rotation on the PAN axis by a tiny triggering magnet fixed on the gear. The sensor is mounted on PCB at few millimeters of distance from the magnet and its output is read by a digital port of the Arduino.

The PCB is designed to be a *plug and play* solution with all the peripheral devices in order to facilitate the maintenance and disassembly.
To make the most of the available space in the gimbal structure, the PCB has a horseshoe shape capable to "surround" the PAN main toothed gear as shown in this picture:

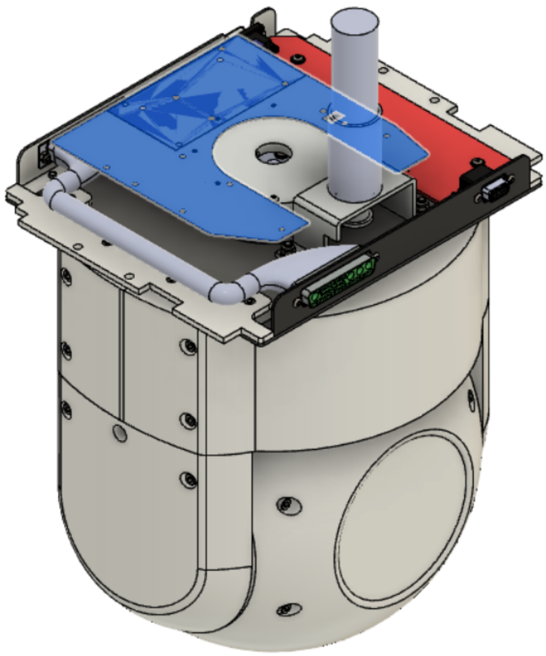

Figure 2.23: The PCB 3D model installed in the *SmartGimbal*

The PCB is a two layer PCB, it was design using the **Eagle CAD**, here are reported some pictures showing the PCB design and rendering.

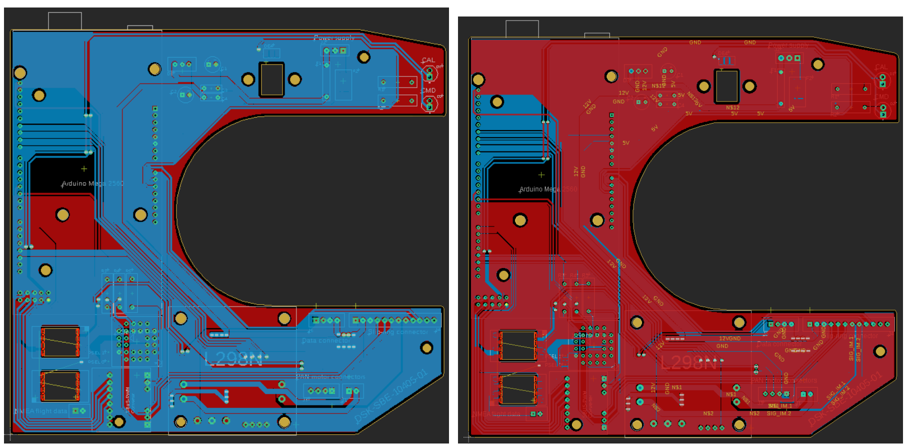

Figure 2.24: The PCB design

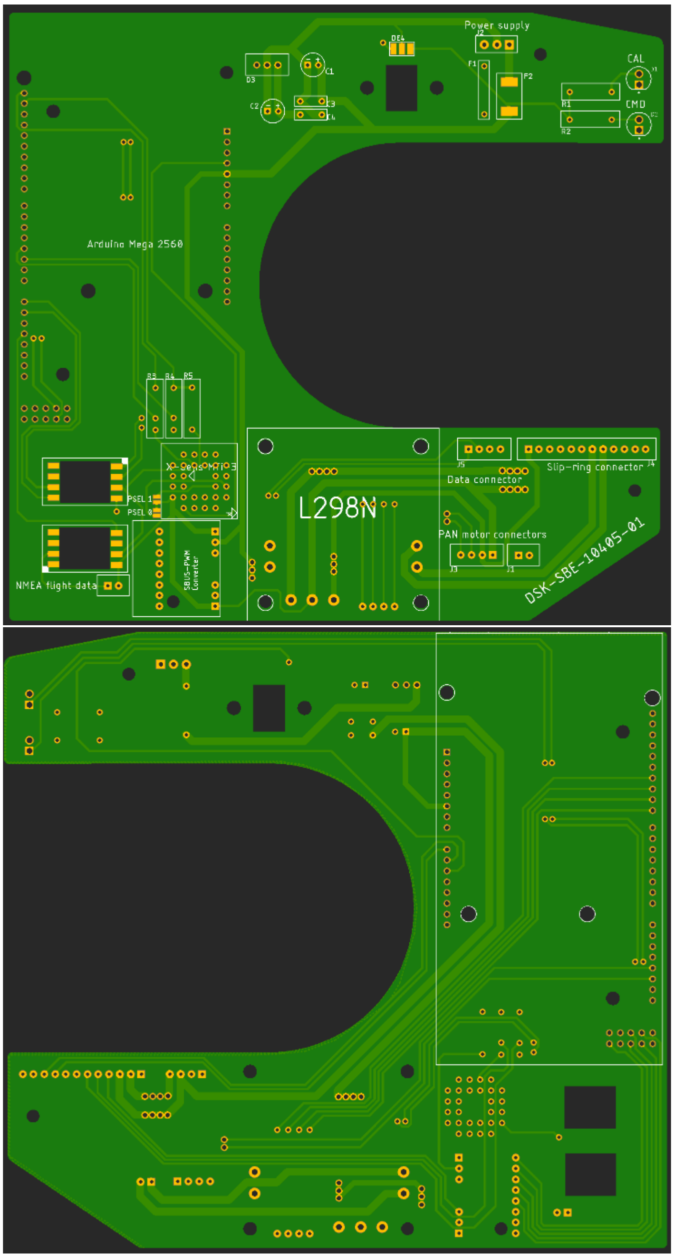

Figure 2.25: The PCB rendering

Chapter 3

SmartGimbal pylon software development

The core of the control software is located in the Arduino of the *SmartGimbal*.
All the main functions of the system are implemented in the microcontroller of gimbal structure.
The code execute the gimbal control algorithms, the camera control routines and the calibration procedure.
Moreover, in this way, the gimbal is able to work independently in case of broken connection with the console, setting in a safe condition by itself.
The entire software is developed in the Simulink environment and then flashed in the Arduino board.

3.1 Motion control laws

For the two movement axis the same laws are implemented, the only difference is due to the lenght of stroke of the two axis;
On the PAN axis is allowed a complete and continuous rotation while, with the TILT axis, only a partial rotation of 180 degree is allowed due to the wiring solution adopted.

A full rotation of the TILT axis would involve a tangle of the camera wiring resulting in possible tearing of the conductors, these figures better shown this problem:

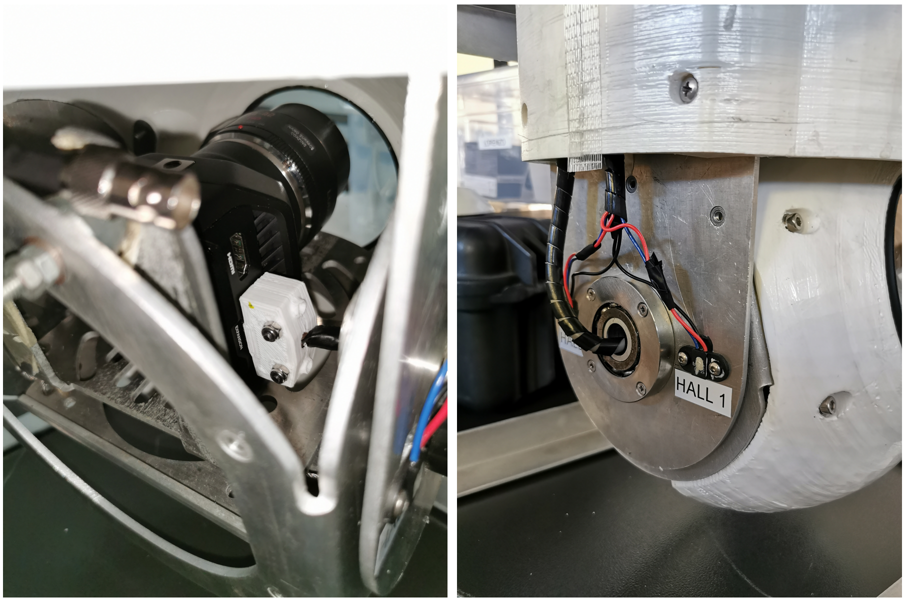

Figure 3.1: The wiring inside the *SmartGimbal* structure

In order to identify the absolute position of the two axes, three Hall sensors are used: one for the PAN axis and two for the TILT axis, corresponding to the rotation limits of the structure.
These sensors are triggered by tiny magnets installed on the rotating parts of the structure.

To measure the axis relative position and speed, the two quadrature encoders integrated in the actuators are used, the movements of the motors are speed controlled proportionally with the joystick angle of the console.
The control laws are basically a PI. It control the angular velocity, including some condition to detect and manage the end of stroke condition.

The control is divided into four procedures:

- Received command elaboration
- Angular velocity and position measuring
- Rotation limits identification
- PI logic implementation and motion control

3.1.1 Received command elaboration

The *Simulink support package for Arduino* provides an easy way to manage the several communication modalities of the Arduino board, there are predefined blocks for the definition of the data packet and the management of the transmission and receiving of serial port.
The gimbal system expects to receive a structured data packet composed as shown in this table:

Data label	**Format**	**Content**
PAN	uint8	PAN movement data
TILT	uint8	TILT movement data
Calibration	uint8	Calibration command (digital 1 or 0)
Zoom	uint8	Zoom value
Focus	uint8	Focus value
Autofocus	uint8	Autofocus command (digital 1 or 0)

Table 3.1: Gimbal received data packet structure

This decoding function is implemented by the *Protocol Decoder* block, giving in output the six values extracted.

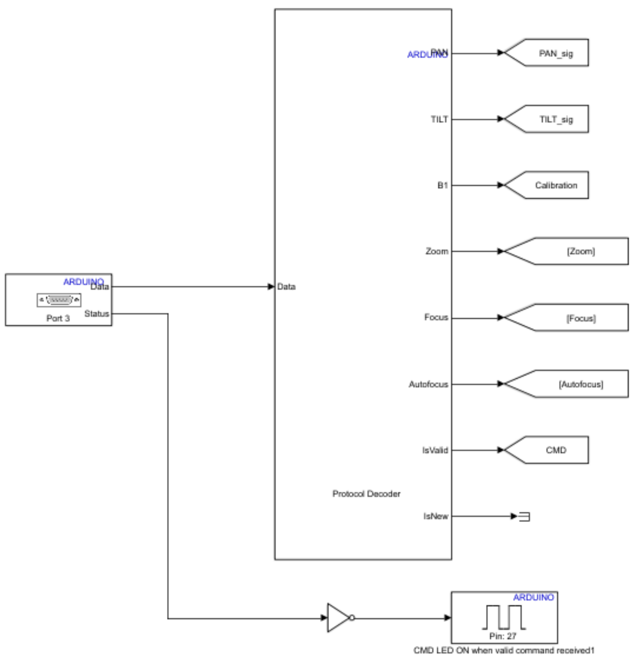

Figure 3.2: The Simulink *Protocol Decoder* blocks for the received data

The commands to control the motion are sent in an *uint8* format;
The angular velocity command is a number from 0 to 255; 128 is the middle value considered as 0, upper value means a clock wise speed instead a lower value means counter-clock-wise rotation of the gimbal. The L298N module has four PWM inputs corresponding to the clockwise and anti-clockwise direction signals for each motor section: to move a motor for example to clockwise direction, a PWM is applied on the "clockwise direction input" while the dual "anti-clockwise direction" input has to be set to zero.
So to accomplish this logic, the software rescales the received value from 0 to 255 in a PWM for speed value and a rotation verse.
The figure shows the algorithm implemented in Simulink for the PAN axis, the same is for the TILT axis.

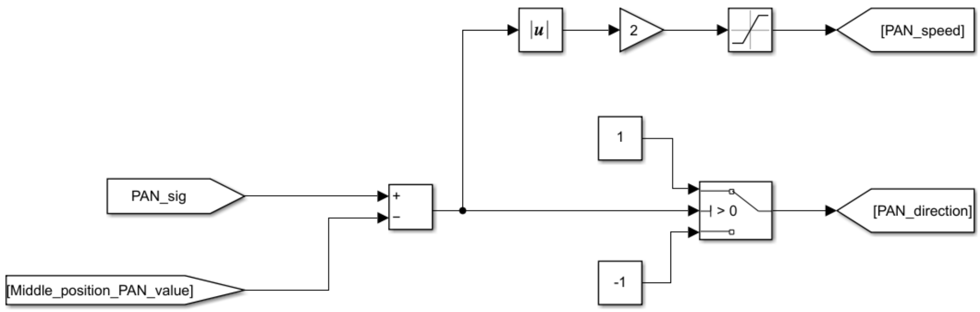

Figure 3.3: The Simulink implementation for the motion commands received from the gimbal

3.1.2 Angular velocity and position measuring

Both these measures are obtained from the encoders installed on each actuator, to read the data from the encoder the *Simulink Support Package for Arduino Hardware* offers a special and ready to use Simulink add-on block *Rensselaer Arduino Support Package Library.*
This block, called *soEncoder*, needs two input pins where the encoder is connected as parameters, it gives in output a value corresponding to an angle delta that needs to be divided for four in order to get the exact degree interval being a quadrature encoder.

Considering that the desired angle is the angle delta of the motor and not the rotation angle of the gimbal axes, this angle must be further elaborated according to the transmission ratios of the actuator and the belt transmission. This is implemented in a simple Simulink subsystem, the *soEncoder* block.
Once the angular position is obtained, a Simulink block derives this data in time obtaining the angular velocity, here is shown the code for the PAN axis, the same is made for the TILT axis:

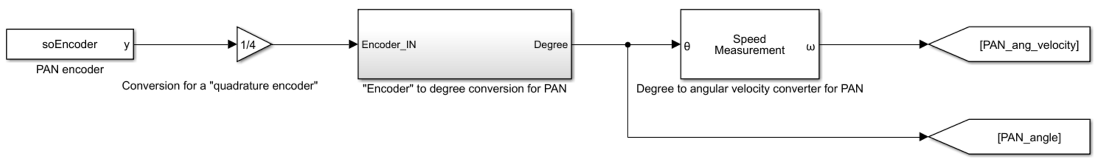

Figure 3.4: The Simulink implementation for angular velocity and position measuring

3.1.3 Rotation limits identification

To identify the rotation limits three Hall sensors are used, these sensors provide in output a low level voltage where the magnet is into the detecting interval while high level when is not triggered.

3.1.4 PI logic implementation

Being the axes movements controlled in speed, the PI controller provides in output a PWM value in order to keep the error between the target speed and the measured one as close as possible to 0.
The Simulink implementation is a bit more articulated due to the presence of not ideal conditions and electric noise, this figure shows the complete implementation of the PI logic in Simulink for the PAN axis clockwise direction:

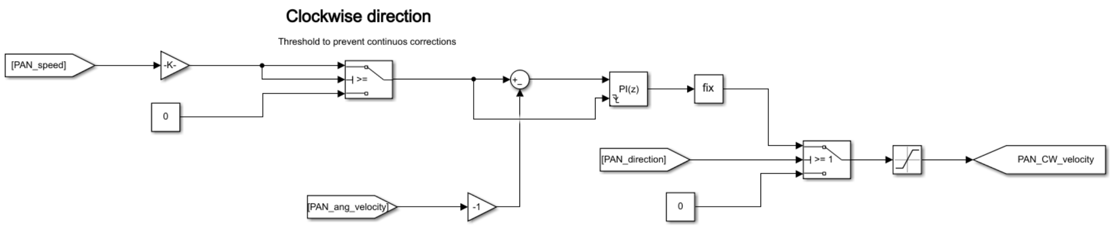

Figure 3.5: The Simulink implementation for the PI logic

The second block from left is a *Switch block*, the purpose of this block is to send an angular speed only if the input value is above a threshold in order to avoid continuous correction by PI due to disturbances, then the *Sum block* calculates the error between the target and actuated rotation speed.

The *PI block* is the core of the process, it contains the proportional and integral parameters, the input port with the "square wave" in input is the reset input, it is mandatory due to the presence of the integrative component in order to bring to 0 the error as soon as the commanded speed is set to 0;
Being a quite slow system, the absence of this reset will cause an error increment during fast command impulse sequences, causing the saturation of the PI and losing the system control.

Following the PI there is another *Switch block* that routes the PWM value to the motor driver only if that specific rotation verse is chosen, in case of opposite commanded rotation verse the PWM is taken from another equivalent PI controller, the two PI coefficients are calibrated a bit differently due to the light asymmetry of the actuators response in the two rotation verse.
Finally, *Saturation block* keeps the value in output from the PI into a safe interval.

3.1.5 Commands sending to the L298N module

Once the PWM value has been calculated it can be sent to the L298N module after checking that the system excursion limits have not been reached or that there are no other routines in progress such as the "calibration" which will be discussed later.
This control is done with a *Switch block* that, in function of the *Calibration tag* flag, allow to the PWM value to pass and manage a PWM output or ignore it giving the output management to another routine or simply setting to 0 the PWM duty cycle value.
In the PAN axis a full rotation is allowed, so the only condition to give the output control is the calibration mode, so a single *Switch block* is necessary as shown in this figure:

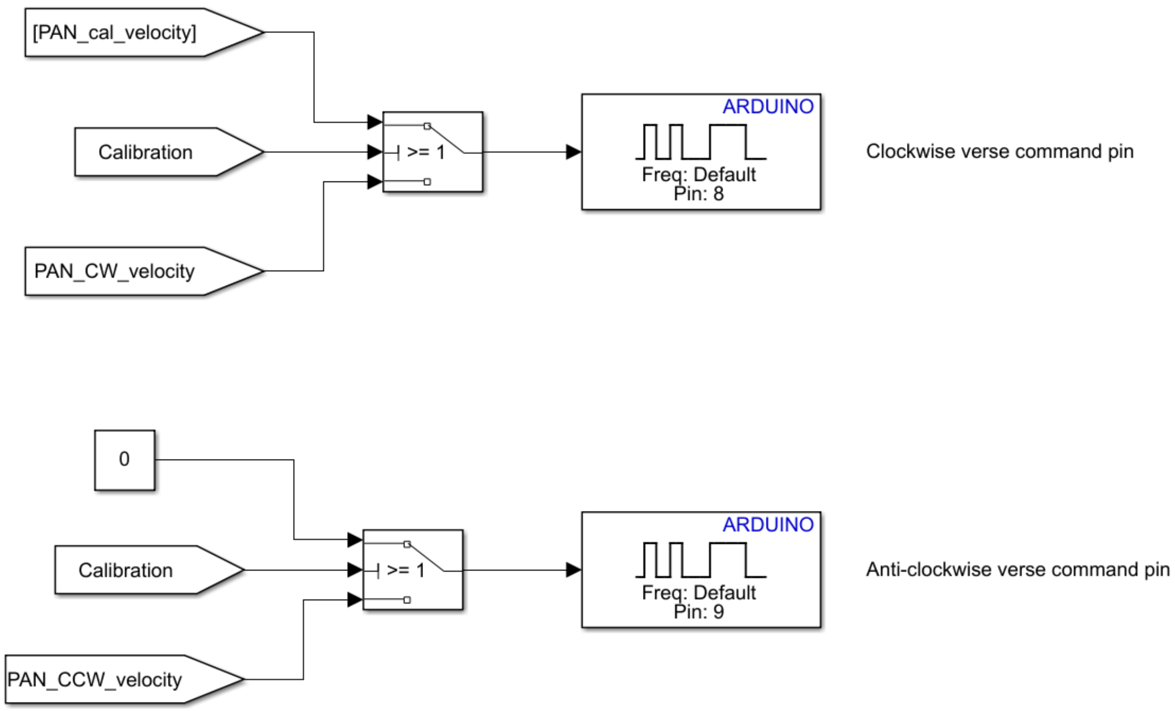

Figure 3.6: The Simulink implementation for the PAN command sending logic

In the TILT axis a further control is needed due to the presence of the upper and lower rotation limits, so another *Switch block* is placed over the first one, holding the movement in a specific direction if the limit is reached.

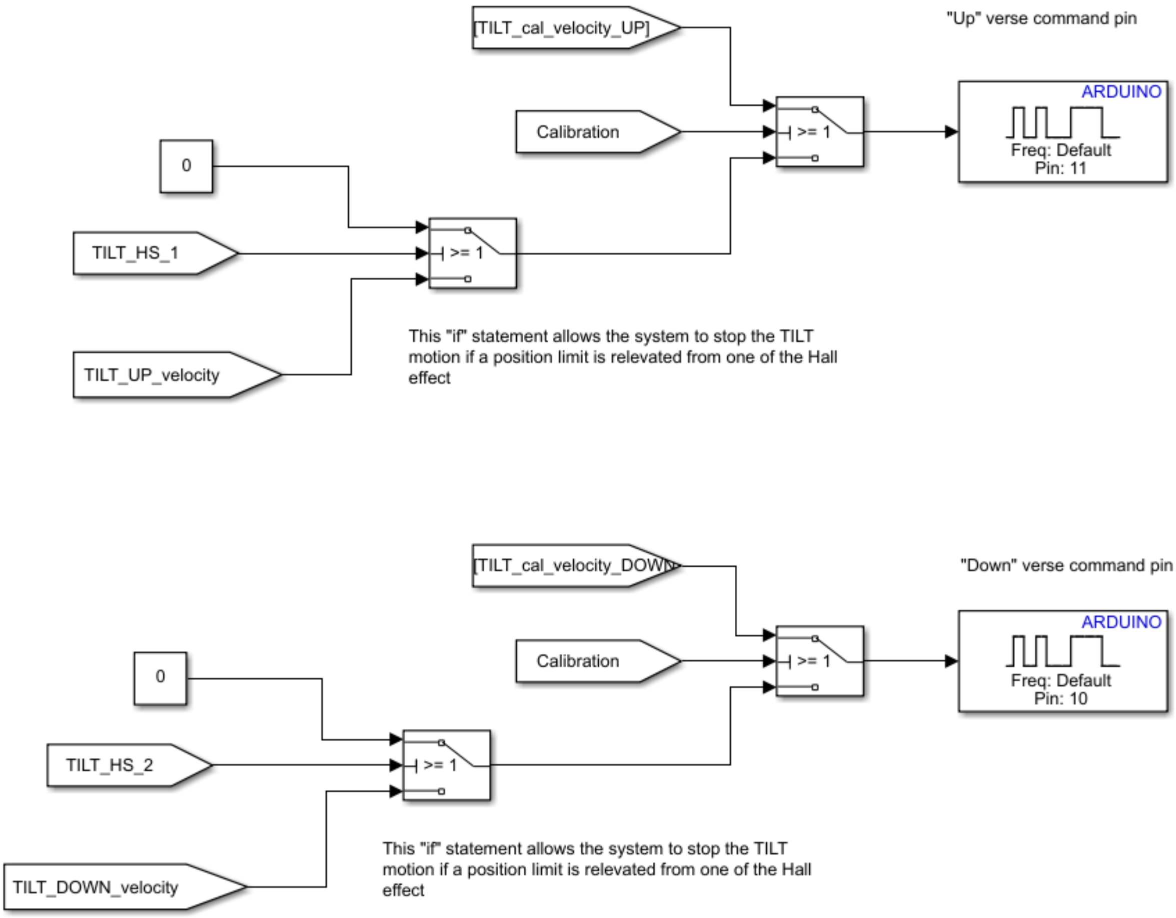

Figure 3.7: The Simulink implementation for the TILT command sending logic

3.2 Secondary functions

To work properly, the gimbal software needs some other functions that are:

- Calibration procedure
- SBUS control
- Data transmission to the console

3.2.1 Calibration procedure

With this name two functions are performed, the first is the orientation of the gimbal towards what that can be defined as the "zero" position, i.e. pointing the camera lens in a known direction;
The second is the zeroing of the angles detected by the encoders, in this way once the system has carried out the procedure, the angles sent to the operator into the cockpit will correspond to the pointing angles refered to a known orientation.
To recognize the "zero position" the calibration procedure follows two different routines, one per axis.

PAN axis calibration The purpose of this routine is to position the magnet installed on the PAN axis main gear under the PAN Hall sensor that identify the zero position for this axis.
This axis enjoys the possibility of having a continuous rotation, so the simplest way to locate the sensor is rotate the gimbal until the magnet is detected by the Hall sensor and then stop the motor.

Unfortunately it is not so simple to do due to a technological limitation of the Hall sensor, it has not a punctual detection but a detection range that for this axis is about 20°, too much to be considered as a reference point.
The sensor brings this tolerance, but fortunately the extremes of this range are positioned in two extremely precise points that can be exploited for a punctual detection, indeed is important to be sure that during the rotation the PAN axis will stop always on the same extreme.
The simplest algorithm is, after receiving the calibration command, to rotate the PAN axis always in the same direction in order to detect every time the extreme located on the same side and, once the magnet is detected, stop the rotation. In short, is an *if statements*, this algorithm presents a non-operating condition!
Following the precedent algorithm once the gimbal has received the calibration command, the system starts to rotate the PAN axis but if the magnet is located into the sensor detection range, it will immediately stop leaving the magnet in a non-precise position, something more reliable is needed.

It is necessary that the system as first operation checks if the magnet is not located under the sensor, in this case it should move the magnet away from the sensor detection range and then execute the previous algorithm, here is shown the procedure flow chart:

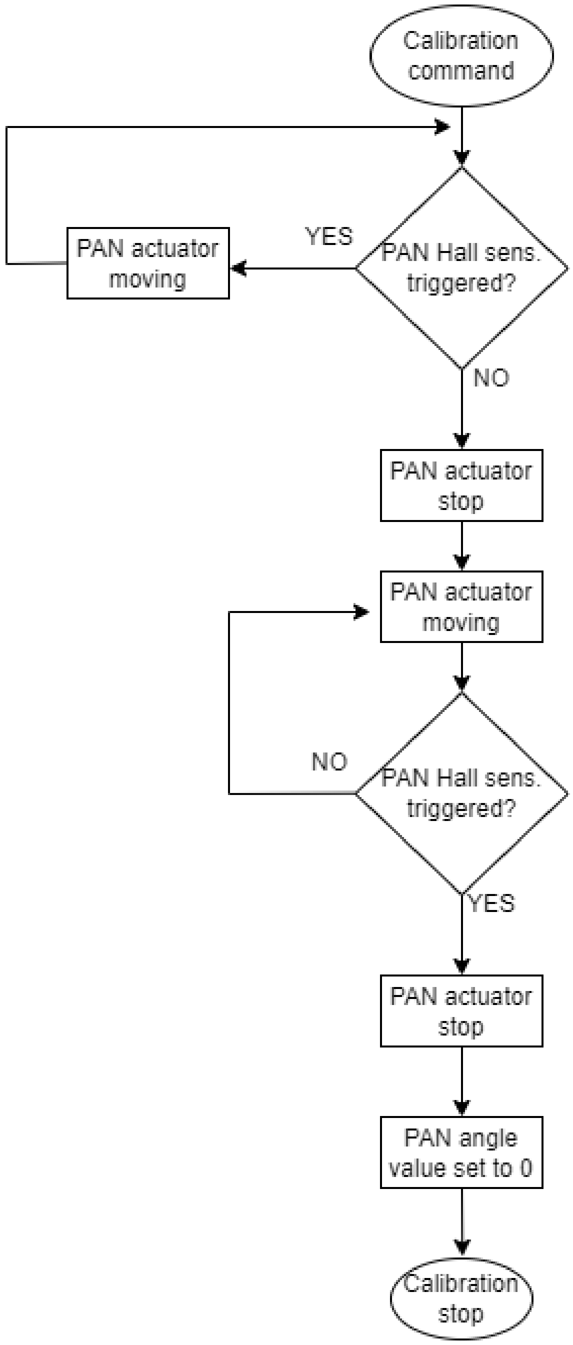

Figure 3.8: The PAN axis calibration logic

Simulink works with a visual language, uncomfortable for the implementation of procedural logics;
To implement this routine the Simulink *C function block* is a good solution.
It consists in a custom block where the user can define input, output, variables and constants correlating each one with other by means of a C code;
Once completed the block firmware writing, Simulink provides to add to the block the input and output ports to give it the appearance of a "standard block" as shown in this figure:

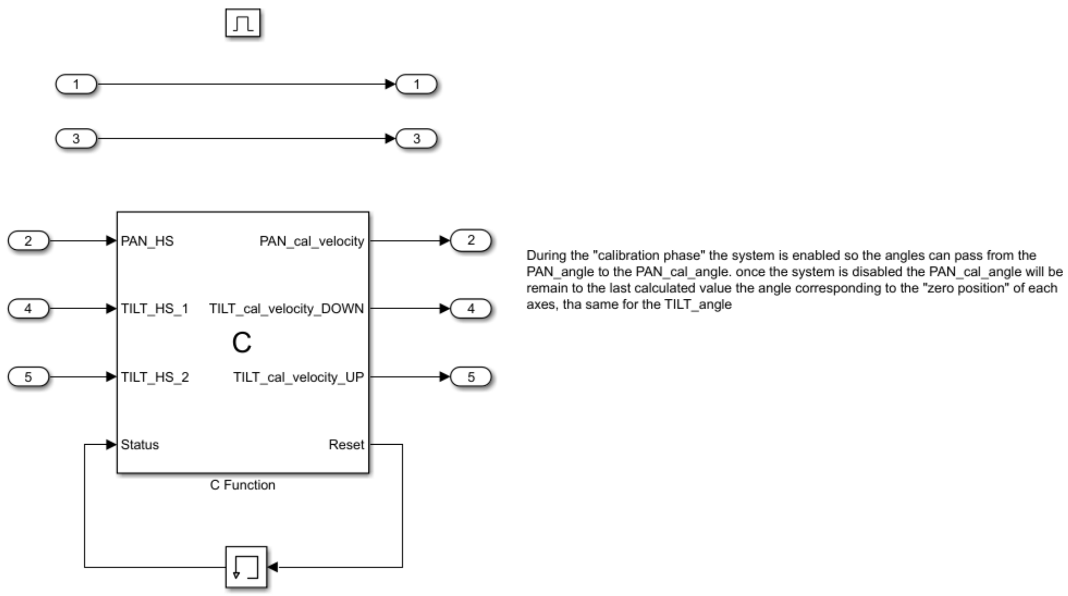

Figure 3.9: The C code for the PAN axis calibration logic

During the code execution, this block will be used like all the other standard Simulink blocks.

TILT axis calibration For the TILT axis a similar routine is followed but due to the excursions limits, a more sophisticated code is necessary.

Unlike the PAN axis here a continuous rotation is not possible because the system is designed to allow the rotation along a semicircular path with limits defined by the Hall sensors, in reality the path is not a complete semicircle due to the presence of the Hall sensor detection range as shown in this figure (the light blue color shows the TILT path while the orange circle the Hall sensors triggering interval):

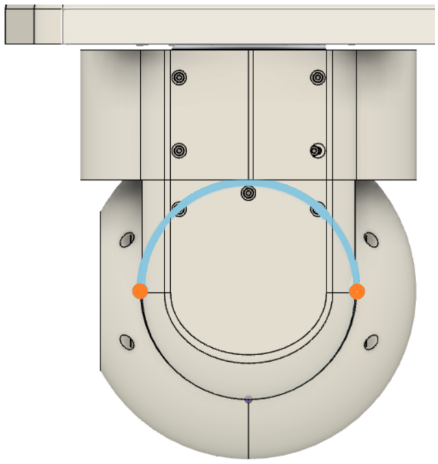

Figure 3.10: The motion path for the TILT axis

Once the gimbal has received the calibration command, the system first starts to rotate in a clockwise direction (referring to the 3.10 figure) in order to locate the "Hall 2", once the magnet has reached it, the rotation is inverted and the gimbal rotate until the "Hall 1" is triggered, this procedure ensures to trigger the "Hall 1" sensor always on the same semicircle of the detection range as explained before for the PAN axis;

Now the system can pass to the final part of the calibration procedure, pointing the camera in the same direction of view of the pilot, to do it this sequence is followed:

1. The gimbal starts to rotate in an anti-clockwise direction incrementing a timer
2. Once the "Hall 2" sensor is reached the timer is stopped taking the time employed to complete the semicircle and stopping the gimbal
3. The gimbal switch to rotate in a clockwise direction stopping after half of the measured time.

These two procedures (PAN and TILT) are carried out simultaneously once the "CALIBRATION" button is pressed on the console.

3.2.2 SBUS control

The SBUS signal to control the camera is generated by the PWM-SBUS converter module, the Arduino board provides to this module up to five servo signals in order to control:

- **Zoom** defined by a linear value
- **Focus** defined by a linear value
- **Autofocus** defined by a boolean value

The remaining two channels are left for future custom defined functions.
To create the servo signal the Simulik Arduino support package, provides a block called *Standard Servo Write* that, given an angle value as input, provides in output the corresponding PWM signal, five of these blocks are used, one per channel.

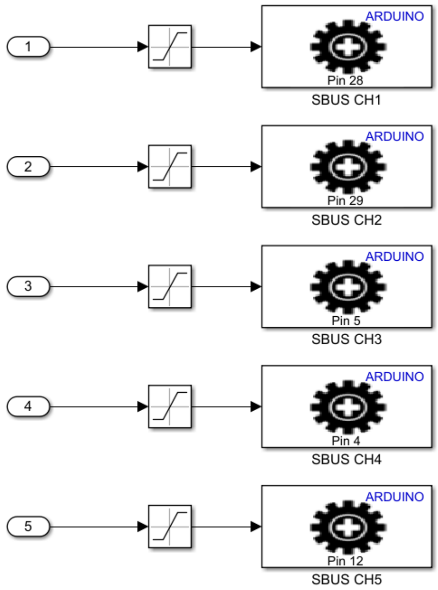

Figure 3.11: SBUS blocks implementation

3.2.3 Data transmission to the console

To provide the user information about the system status some data need to be transmitted to the console inside the cockpit, to do this the same logic of the 3.1.1 function is used.

With the *Protocol Encoder* block, the data are organized into a data structure composed as follows, then sends them to the console with the *Serial send* block that accepts as input a structured data packet.

Data label	**Format**	**Content**
PAN curr angle	double	PAN angle in [deg]
TILT curr angle	double	TILT angle in [deg]
Calibration	uint8	Calibration in progress (digital 1 or 0)

Table 3.2: Gimbal transmitted data packet structure

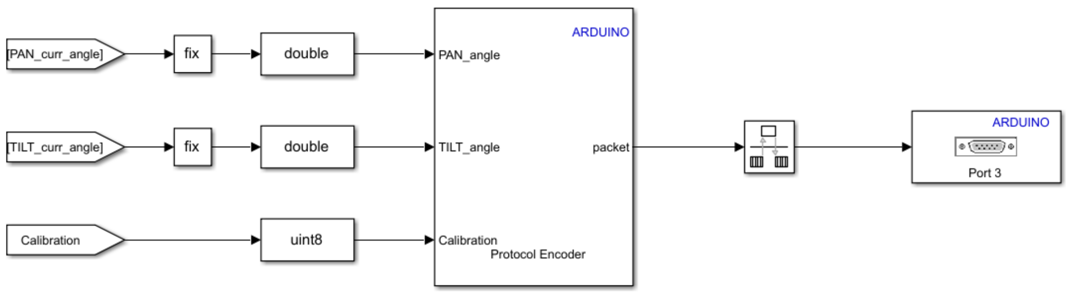

Figure 3.12: Gimbal transmitted data blocks

Chapter 4

Console hardware development

The console allows the operator to manage all the system and get information about its status, providing him data regarding the gimbal motion or similar.
The console is designed to be handled like a gaming controller, although it can also be used by the pilot as a "cockpit add on" fixing it on the cockpit structure.
It provides all the necessary commands to control the gimbal except for the preview of the current camera view, for this a dedicated display is needed;

This configuration allows different cockpit accommodation for the several parts of the system, the connection through the console to the gimbal inside the plane is made by a single four ways connector to guarantee a safe and fast setting up.

4.1 Console general conformation

Here is shown a picture of the console with every function explained:

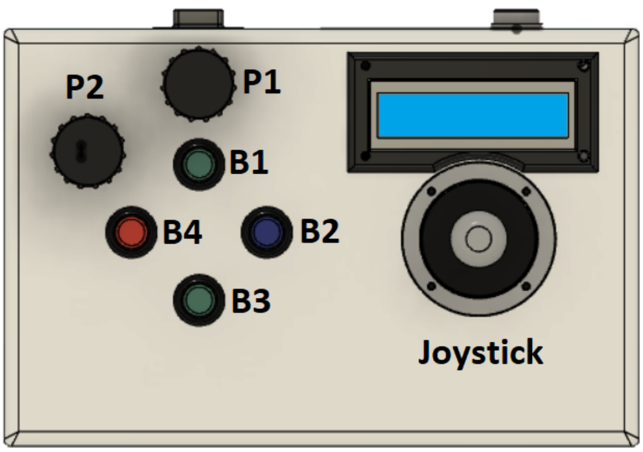

Figure 4.1: The control console

- **B1** This button enables the operator to start the calibration procedure
- **B2** This button is an "emergency button" to stop all the movement of the system
- **B3** This button allows the autofocus of the target
- **B4** Leave unused for custom function
- **P1** It controls the camera zoom
- **P2** It control the camera focus
- **Joystick** It controls the movement of the gimbal in terms of angular velocity

The console has two connectors, the first circular one connects the console with the gimbal providing power supply and a data exchange line. The second is a D-SUB 15 connector working as an expansion connector with external components, providing two TX-RX serial lines.

4.2 Electronic board

4.2.1 Electronic board general considerations

The console electronic board is very simple, it must carry out these tasks:

- Provide a 5 V power supply to the Arduino board and the display
- Provide an interface between the Arduino board with the remaining components like button, potentiometers, ecc.
- Allow the conversion between RS232 and the UART for the data exchange with the gimbal structure

4.2.2 Main Hardware description

On board microcontroller The microcontroller used in the console is the same used into the gimbal, an Arduino Mega 2560;

In this project this microcontroller is under-exploited, the reason to use it is the fact that its working voltage is 5 V and can provide up to four UART TX-RX serial line useful to interface the *SmartGimbal* system with other devices.

LCD display The display used is a very diffused module, a 16 columns for 2 rows LCD display, interface it with the Arduino board is very simple thanks to the *HD44780* driver with a 4 bit parallel data transfer already implemented in many SW libraries such as Arduino's *LiquidCrystal*.

Board power supply

The console internal circuit works with a 5 V power supply, this voltage is available into the *SmartBay* pylon but being a relatively low voltage that needs to pass inside the wing to reach the cockpit, there was the risk to receive disturbances;
In order to avoid this issue, it was decided to use for the power supply of the console a voltage of 12 V. Once in the console, this voltage must be lowered and stabilized down to 5 V, so first to dimension the supply circuit is necessary to know the current consumption of each component.

The most expensive component in this term is the LCD due to the LED back lighting that need up to 0.16 A, the Arduino board needs about 50 mA while the remaining part of the board circuit are "signal electronics" with negligible consumption, the total is around 210 mA.
In this condition a linear voltage regulator dissipates 1.47 W reducing from 12 V to 5 V, this can be managed with an adapt heatsink without looking for to a more complex and noisy solutions such as a switching power supply circuit.
Thanks to its wide diffusion ad reliability the LM7805 regulator was chosen, the complete power supply circuit is shown in this figure:

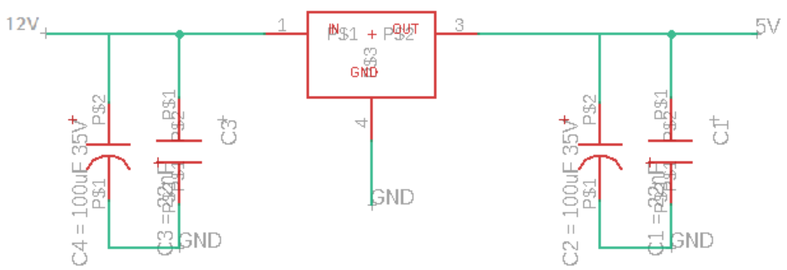

Figure 4.2: The control console power supply circuit scheme

4.2.3 Anti bounce circuit

Being a device working in an environment conditioned by a notable presence of mechanical vibrations, it is mandatory that buttons are protected from the risk of bouncing, to do this, an anti bounce circuit with a low pass filter with a 760 us constant time was used.

4.2.4 PCB design

All the previously mentioned devices have been integrated in a two layers PCB design with the *Eagle CAD*, here are shown some rendering of the board:

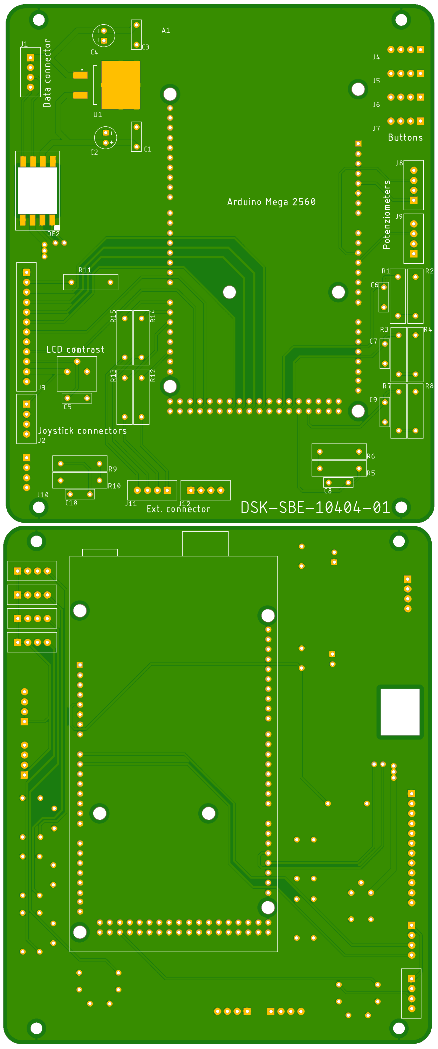

Figure 4.3: The console PCB rendering

4.3 Console structure and ergonomics

The console is a device developed in order to be handled with both hands by an operator or by the pilot with one hand with the console properly fixed on the cockpit, so it is fundamental to do some considerations about ergonomics.

The main command of the console is the joystick, so is mandatory making attention on its positioning. The main idea is to handle the console like a "gaming controller" using the joystick with one or maximum two fingers, so its best position is on the right side of the console near the edge in order to be handled with thumb and forefinger.
On the console there are also two knobs used to control the camera zoom and focus;
These controls will be use often during the mission, it is important to place them in an easily reachable position, after some "handling tests" the best position was found on the left side of the console near the corner, again to be handled with thumb and forefinger;
To improve the ergonomics also special knob was design with a special external profile shown in this figure:

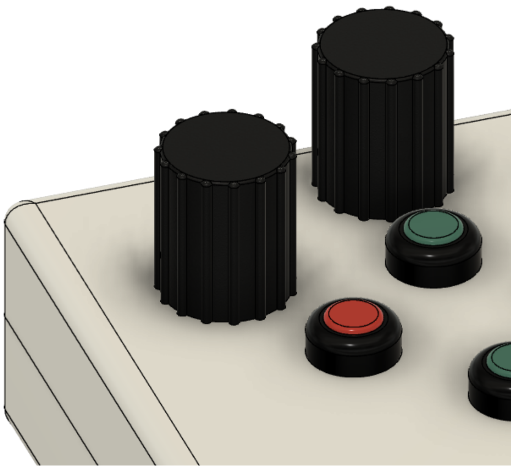

Figure 4.4: The custom console knobs

Once designed, the two knobs have been created with the 3D printing technique in ABS.

Chapter 5

Console software development

The software for the console is also developed by Simulink, the purpose of this code is to check the status of the control buttons and potentiometers and send them to the gimbal microcontrolle.

The software is divided in two blocks, the first is mainly composed by a Simulink *Protocol Encoder* block with the purpose of building the message containing the commands for the *SmartGimbal* pylon, this block also contains the Simulink routine relative to the management of external inputs.

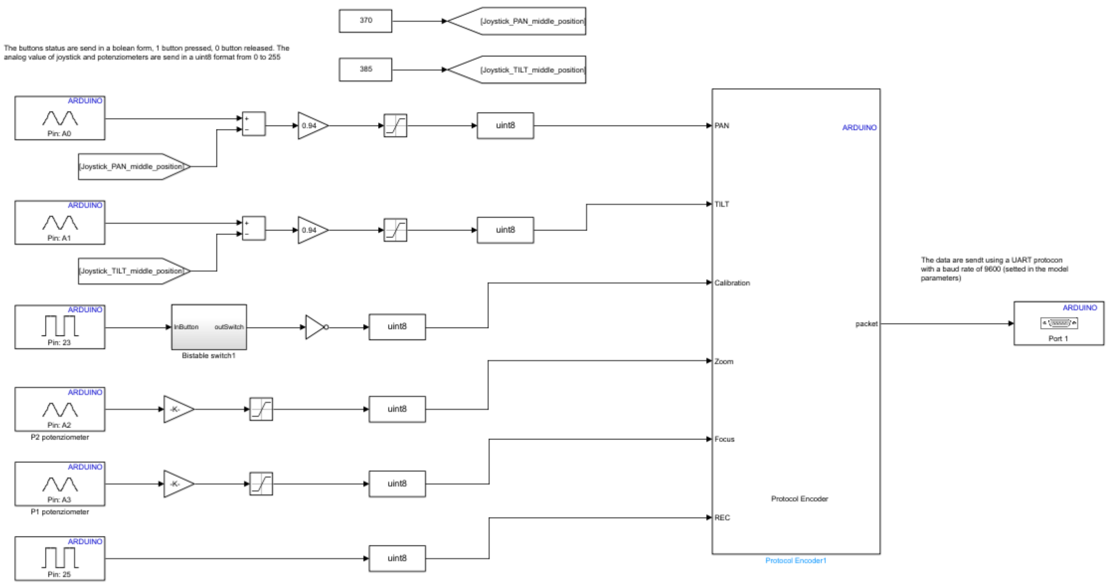

Figure 5.1: The "Protocol Decoder" software block

The second software block can be considered as the dual of the one previously exposed, it contains a *Protocol Decoder* in order to receive the PAN and TILT position angles from the *SmartGimbal* pylon.

To show the angles on the display, a Simulink block called *S function* is used. It is a further improvement of the *C function block* already explained in the chapter 2, it is basically a block able to execute custom code written in different programming languages.

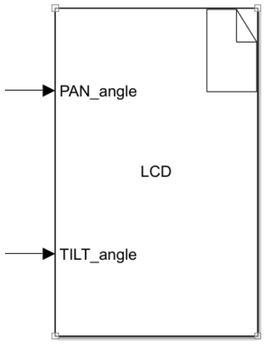

Figure 5.2: The "S function" block

The code used is the same one that would be written in C using the Arduino IDE, it is enough to include the same library used in the C Arduino code, for this purpose only the *LiquidCrystal* library is required.
This library contains all the instructions to implement the communication with the display, providing to the programmer simple commands to set the position of the cursor on the display and manage the characters strings.

```
/* Includes_BEGIN */
#include <math.h>
#include <string.h>
#include <stdlib.h>
#ifndef MATLAB_MEX_FILE
#include "LiquidCrystal.h"
#include "LiquidCrystal.cpp"

LiquidCrystal lcd(25,27, 33,35,37,39);

#endif
/* Includes_END */

/* Externs_BEGIN */
/* extern double func(double a); */
/* Externs_END */

void LCD_Start_wrapper(real_T *xD,
                       SimStruct *S)
{
/* Start_BEGIN */
/*
 * Custom Start code goes here.
 */
/* Start_END */
}
```

Figure 5.3: A section of code contained in the "S function" block

The code contained in this block was developed by *Mirco Vinciguerra* [1], the actual system does not need a code to allow the user to view the orientation angles of the gimbal, ain that the system has been designed to be used observing the video output from the display placed in the cockpit.
It was decided however to insert this block in the final console software for further system functions upgrades.

[1] Development of a closed-loop control system for an airborne gimbal camera, Mirco Vinciguerra 2021

Chapter 6

SmartGimbal prototype component production

Once terminated the design of each component it was possible to move on to the building of the individual parts, adding only some small safety improvements in order to pass the flight tests successfully and a possible operational life.

The component that had to be produced are listed below:

- Electronic boards
- Full aerodynamic fairing

6.1 Electronic boards

Once design by the *Eagle CAD* the PCB realization was commissioned to an external company then assembled with the components in *DigiSky*.

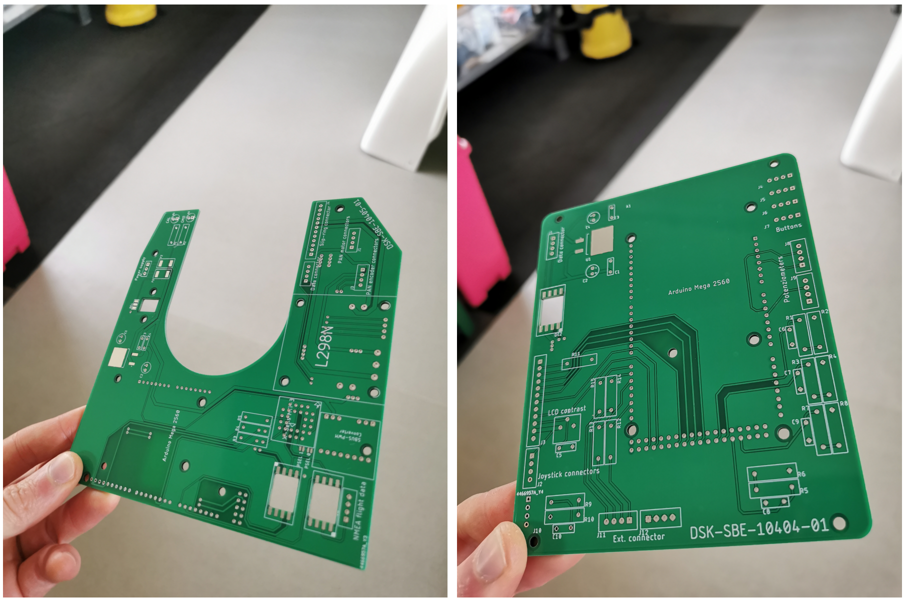

Figure 6.1: The first PCB versions

6.2 Aerodynamic fairing

The aerodynamic fairing was developed simultaneously with the gimbal supporting structure but differently to it, only one prototype for "demonstration" purposes was build.

The original project foresaw to realize these components in plastic material discarding more complex solutions such as the realization in composite of fiberglass or CFRP, this because they would not have offered any kind of advantage but only an extremely more complex and expensive production process;
The fairing is made of six components and being this a prototype and in general a device aimed at a highly specialized sector, therefore with extremely reduced production volumes, was immediately identify in the *additive manufacturing* the best building solution, this also due to the fact that *DigiSky* already got a 3D printer.

The first printing test were carried out using an ABS filament, in that the ABS is quite easy to use but the final version on the fairing was printed in PLA, that shows an excellent good appearance and a quite good robustness. Here is shown the front sphere of the aerodynamic fairing with installed the UV filter, the filter in this application is used only as a protection glass for the camera.

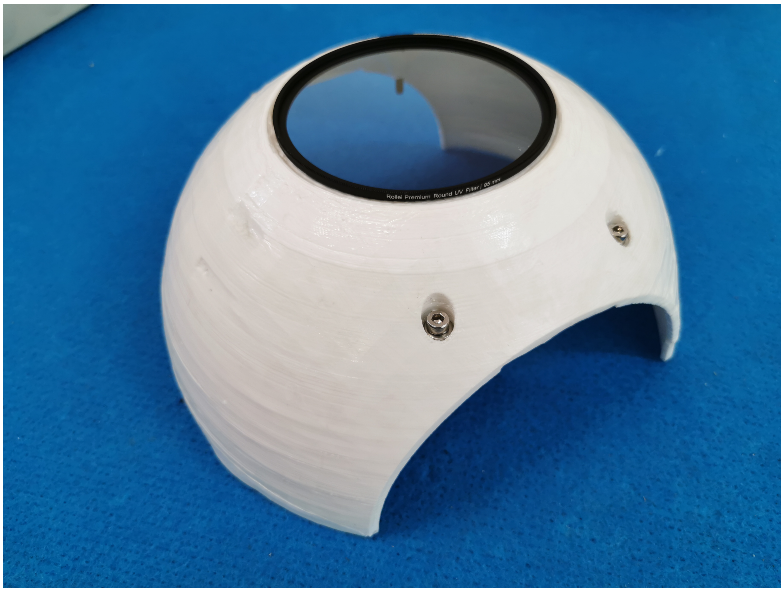

Figure 6.2: The front sphere of the aerodynamic fairing

Chapter 7

Ground tests

Once the two main components of the system have been completed and tested individually, it was possible to merge them to begin the tests of the complete system;

7.1 Laboratory tests

The first tests were conduced on a laboratory bench directly connecting the gimbal to the console and the camera to the display that would be installed in the cockpit later, in this figure is shown the test setup:

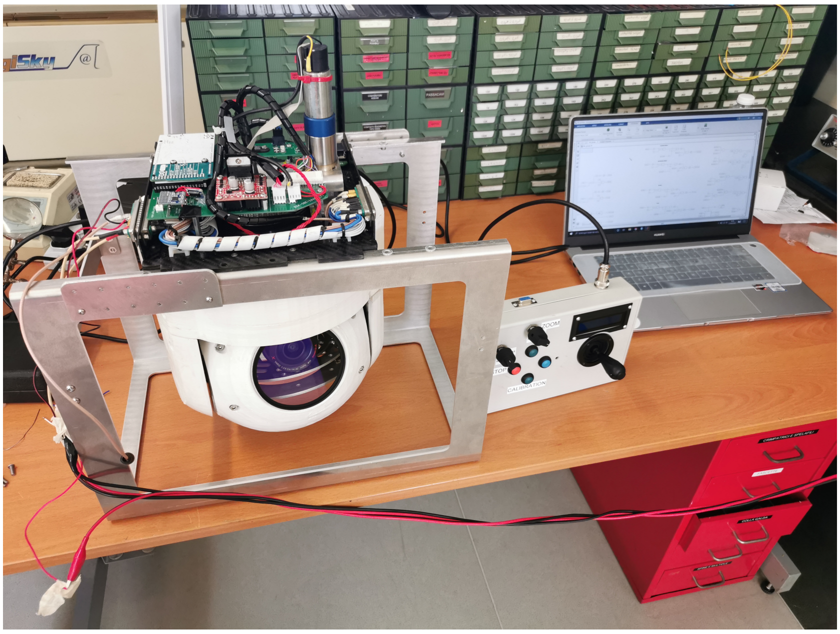

Figure 7.1: The laboratory tests setup

The purpose of these test was firstly verify the entire functionality of the system and secondary, to calibrate the PI control parameters in order to ensure smooth and precise movements of the gimbal.

7.2 Ground test on the plane

Once ensured the system safety and reliability on the test bench, it was possible to install the system on the *SmartBay*.
The *SmartGimbal* pylon installation follows the standard procedure of all the platform design to be used with *SmartBay*;

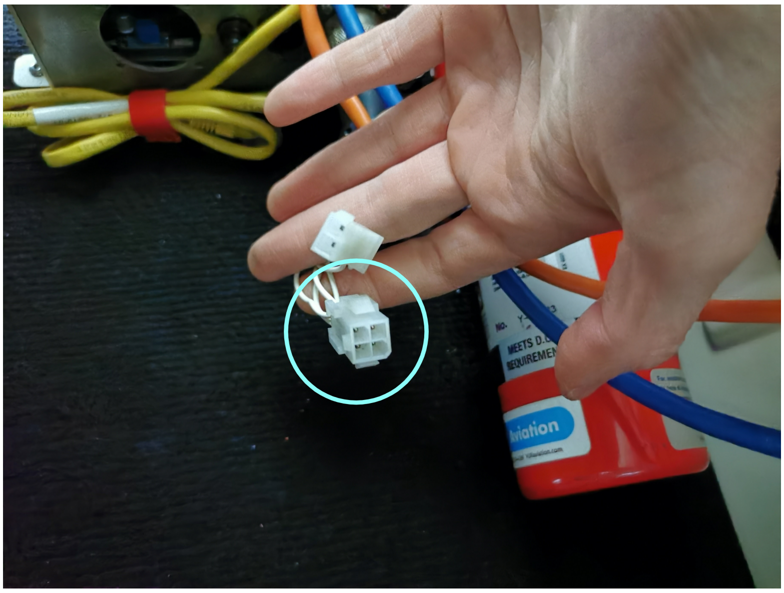

Figure 7.2: The cockpit console connector

After the console has been connected to the system the *SmartGimbal* is ready to work, to power up the system is necessary to provide power supply to the *SmartBay* pylon through the "*SmartBay* master switch" in the cockpit:

Figure 7.3: The *SmartBay* master switch

Being one of *SmartBay* payloads requiring the highest amount of energy (for a Tecnam P92 plane), to test this system on the ground with the plane engine off, it is mandatory to provide plane energy from an external source;

The plane battery does not have enough capacity and would eventually run out, here there is a figure showing the plane setup for the ground tests.

Figure 7.4: The plane setup for the ground test

7.2.1 Preliminary ground test

The first ground test has the aim to verify that the data exchange between console and gimbal works properly, once checked it was possible to pass the ground test with engine on;
This test was of fundamental importance due to the fact that on the plane the engine in motion is the stronger source of mechanical vibration and electromagnetic emission.
The test was conduced in the apron in front of the *DigiSky* hangar, once completed the checklist and turned on the engine, the first concern was the reaction of the gimbal to the engine vibrations:

Running the engine from the idle (1800 RPM) to the cruise RPM, a resonance frequency was registered around 1800 RPM with the gimbal that rocked violently.
This was the only engine speed at which gimbal resonance was recorded, otherwise the gimbal structure was not subject to any kind of vibration. This RPM value is a transition regime that will be keep for a few seconds only during the plane engine start up;
Verified the solidity of the structure, it was possible to check the motion of the gimbal.

Figure 7.5: The plane with engine ON during the ground test

7.2.2 Ground test

The previous test was a preliminary test which does not demonstrate the full functionality of the system because there wasn't objective data;
The aim of this test was to check the correct function of the PID controllers during the gimbal motion, so precise numerical data was required.
The gimbal can rotate on two axis PAN and TILT, in both cases it is possible to define an imaginary path in which travel time can be measured, the test consist in commanding one of the two axis with a velocity step input measuring the time used to complete the path;
This procedure was repeated with engine ON and engine OFF, the purpose of the test is to verify if the PID is able to control precisely the speed in different vibrations conditions.

If the PID work properly, the PAN and TILT paths travel times for a specific angular speed at different engine RPM should be almost constant.
The PAN axis path consists in a complete turn of the structure using the PAN Hall sensor and the magnet as a reference where start and stop every turn.

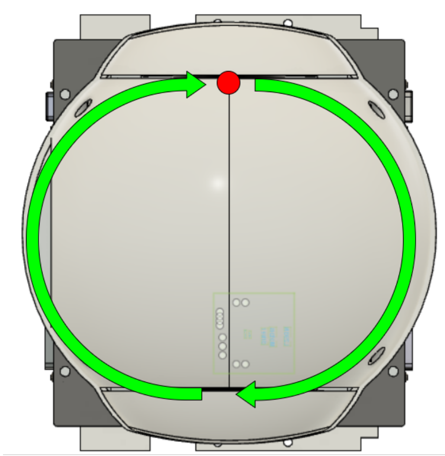

Figure 7.6: The PAN axis test path

The TILT axis is a bit more complicated, as there is no possibility of completing a full turn. The test path in this case is composed by a semicircle ran first in a clock wise verse and then back in an anti-clock wise verse, following the steps showed in these pictures:

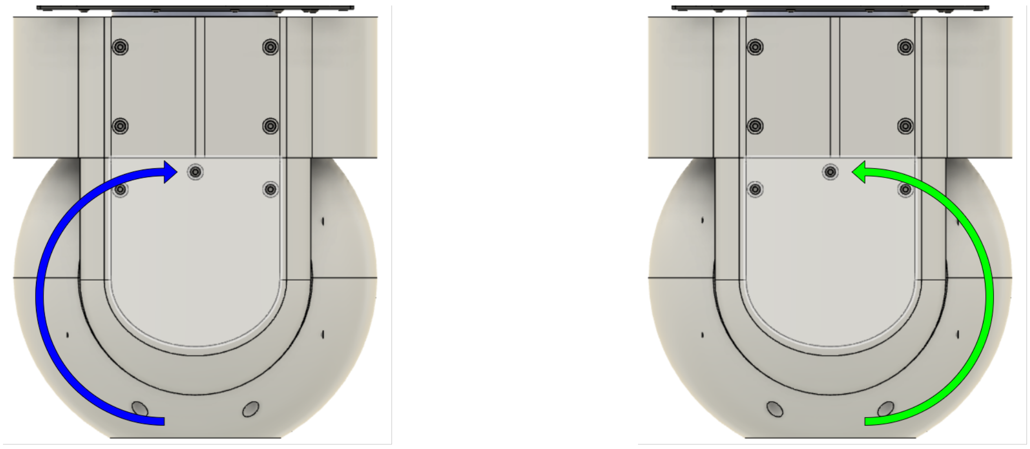

Figure 7.7: The TILT axis test path steps

This sequential test SW was implemented using the Arduino IDE due to its simplicity and avoiding the use of Simulink that needed more time.

Again, two codes (gimbal console) were necessary with the following tasks:

Gimbal SW

- To receive the commands from the console
- To implement the test procedure,
- To send to the console the measured time so that it can be recorded for post-processing by the pilot.

Console SW

- Convert the commands provided by the user and send it to the gimbal
- Show on the display the results in order to be annotated by the pilot

The most important feature of this test SW is the complete logic equivalence to the Simulink code.
To make the test consistent, it was mandatory to use the same values used in the Simulink code executing also some specific laboratory check to verify the congruence of the two SW.
The developing of the PID on the Arduino IDE was made using the Arduino library *ArduPID.h*, this library integrates everything needed to implement a PI control system also with the integrator reset like in the Simulink SW.

Here are shown two flowcharts with the steps followed during the test for the PAN and TILT axis.

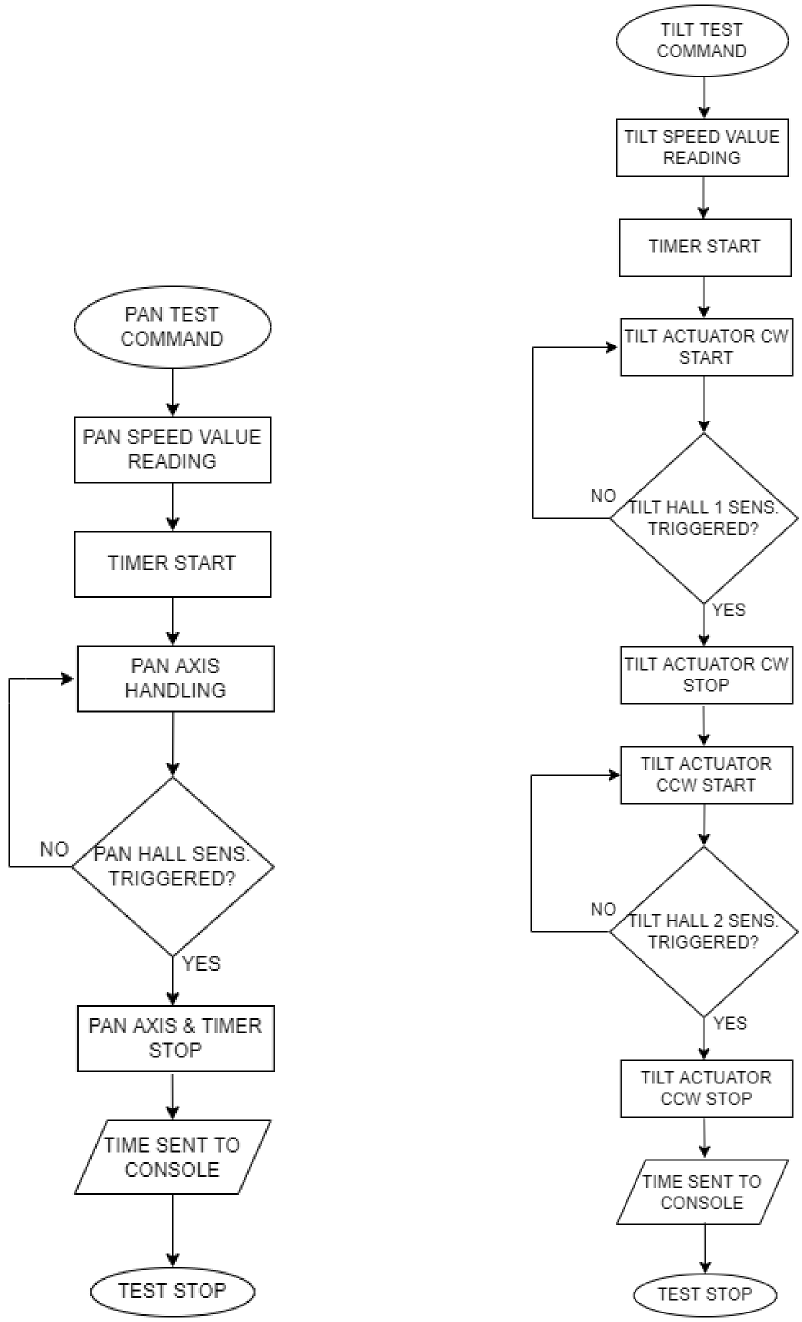

Figure 7.8: The PAN TILT axis test procedure flow charts

To perform this test the console layout has been modified also, it was necessary to reduce to the minimum all what are not "digital interaction" between the pilot and the system. During this test the only commands usable on the console were the four buttons and the potentiometers which would be used as two "switch", rotating them from the minimum to the maximum.
Here a picture of the console in the test configuration:

Figure 7.9: The console "test configuration"

Where the several commands are used to:

- **SPEED:** this knob defines the rotation velocity step magnitude during the test
- **ZOOM** this knob defines the camera zoom value
- **STOP** pressing this button the system will stop every operation and movement
- **CALIBRATION** pressing this button the system will do the calibration
- **PAN** pressing this button the system will start the test for the PAN axis
- **TILT** pressing this button the system will start the test for the PAN axis

Once one of the two axes test is completed, the measured time will appear on the display with four digit only, the firs two are the seconds while the last two the hundredths of a second. This strange time visualization was decided to minimize the workload of the pilot during the flight.

Ground test checklist For a precise test execution it was decided to write and follow a checklist in order to not leave out any possible action, the checklist foresees the execution of the same tests in the same order with engine off and engine on, here is shown the page relatively to the "engine off" test:

1. GROUND TEST, ENGINE OFF **ESTIMATED TIME: 5 min**

Procedure	Time:
Set the **SPEED** knob on **MIN**, press the **PAN** button, report the time on the display.	
Press the **STOP** button	
Press the **TILT** button and report the time on the display.	
Press the **STOP** button	
Set the **SPEED** knob on **MAX**, press the **PAN** button, report the time on the display.	
Press the **STOP** button	
Press the **TILT** button and report the time on the display.	

Note

Figure 7.10: The ground test checklist

As visible, the checklist needs the interaction with an operator only for pressing a button or read and write a number in order to reduce the operator workload.
These times alone are not consistent to evaluate the correct functioning of the system, they must be compared with the time coming from the flight test that will be discussed later.

Chapter 8

Effect of the SmartGimbal installation

In this chapter will be analyzed the CFD simulation of the aerodynamic effects of the installation of the *SmartGimbal* on the P92 plane wing providing four cases listed below:

1. Plane without the *SmartGimbal* in cruise condition with speed of 56.07 m/s
2. Plane with the *SmartGimbal* in cruise condition with speed of 56.07 m/s
3. Plane with the *SmartGimbal* in cruise condition with speed of 72.54 m/s
4. Plane with the *SmartGimbal* near stall condition with speed of 22.64 m/s

All these simulations were carried out with the following common air parameters:

- Air density of $1.15 Kg/m^3$
- Dynamic viscosity of 1.775 $\times 10^{-5} Pa * s$
- Air temperature $284.15K$

The following CFD were produced by the company *BOGGI AERONAUTICS* [1] with CAD files provided by *DigiSky* with the following model assumptions:

- The CAD model was simplified using "standard" bodies such as spheres and plate, for example the three carbon plate of the *SmartBay* were analyzed as a single plate
- The screws hole of the aerodynamic fairing were neglected
- Due to the distance between the *SmartGimbal* and the fuselage this latter was not considered into the simulations
- The transitional analysis was made only on a portion of the wing.

[1] Boggi Aeronautics, Aerodynamic and structural analysis on P92P001, 2018

In these figure is shown the computational domain of the simulations:

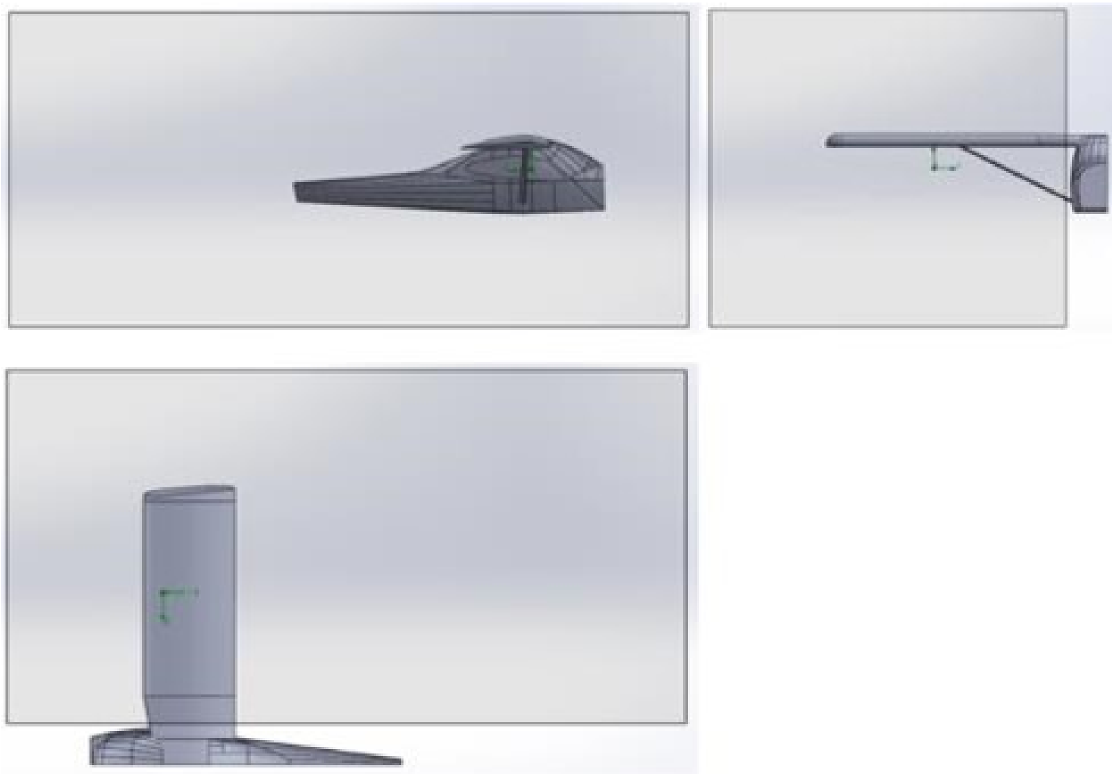

Figure 8.1: The CFD simulations computational domain

8.1 Case 1

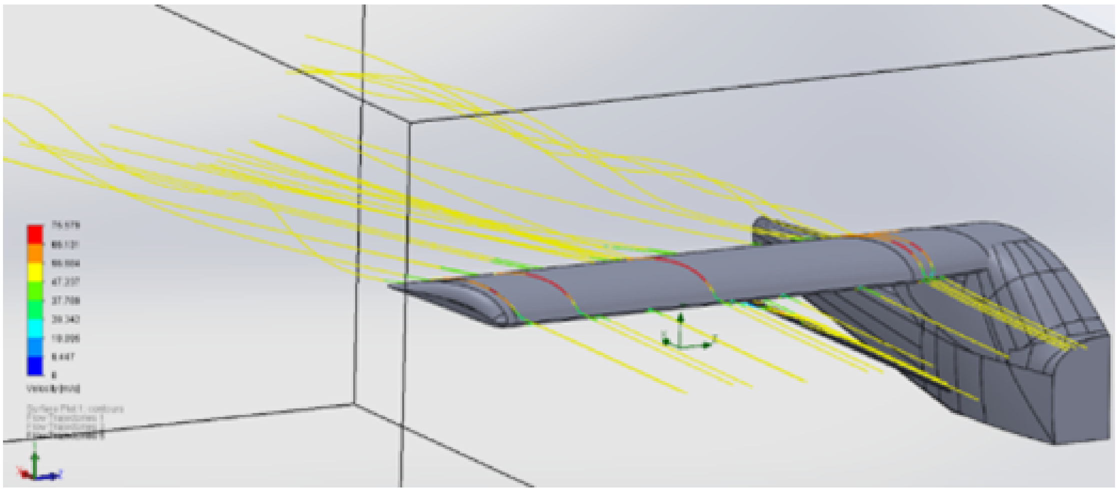

Figure 8.2: The streamline around the wing

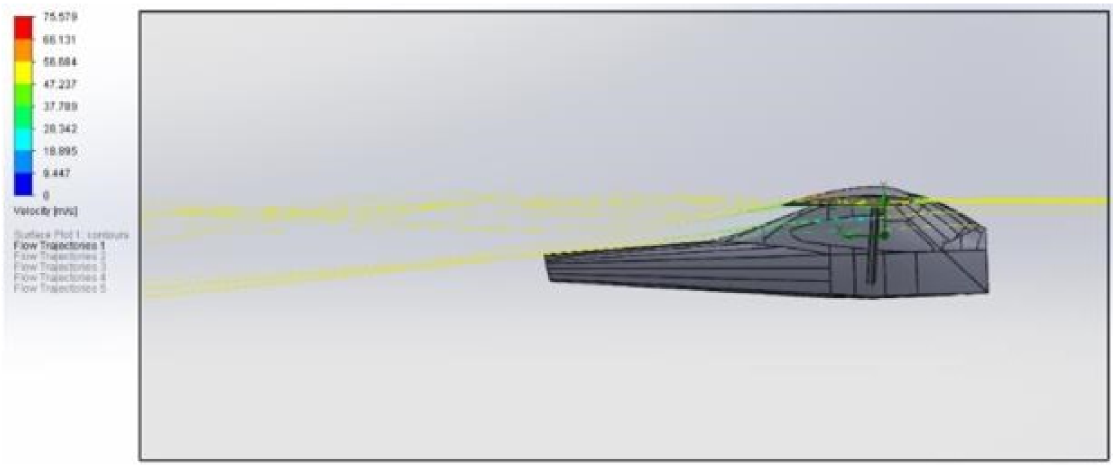

Figure 8.3: The streamline around the wing (lateral view)

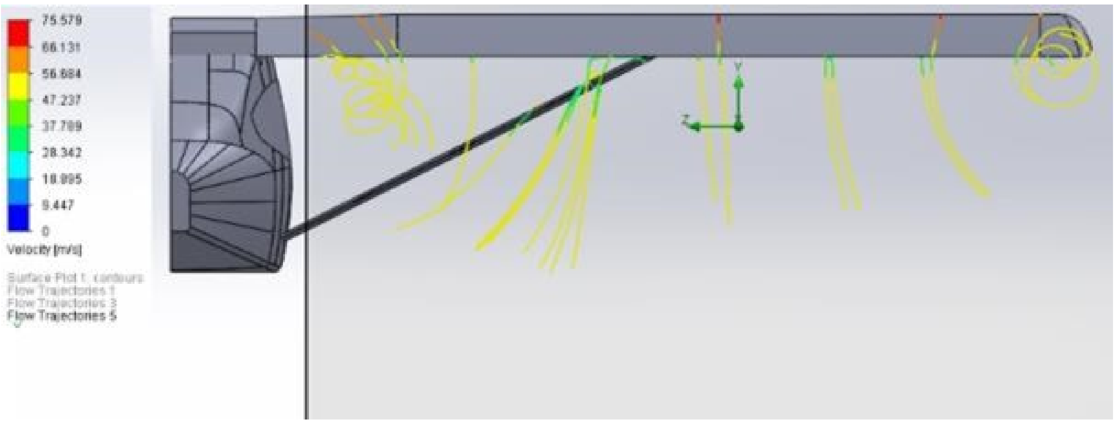

Figure 8.4: The streamline around the wing (rear view)

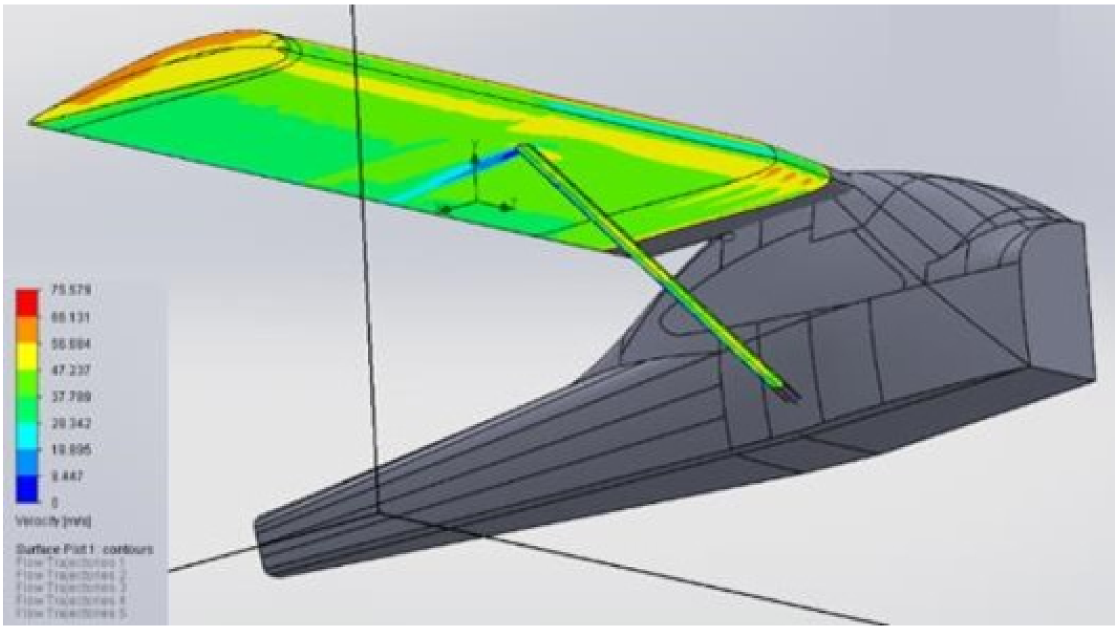

Figure 8.5: The surface speed on the belly of the wing

On the wing section of the computational domain, the following forces and moments are present. The values on the axes in the table are referred to the RS with origin in the center of gravity represented below:

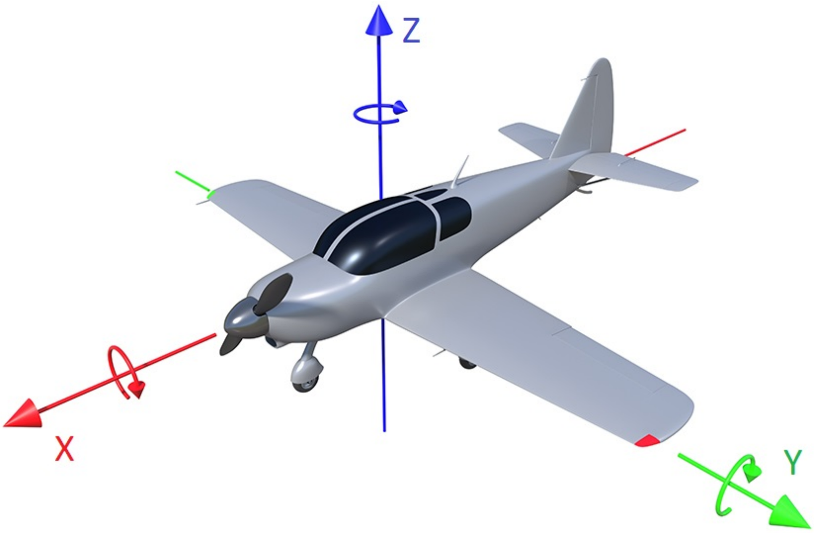

Figure 8.6: The plane reference system

Force on X axis	-400 N
Force on Y axis	10 N
Force on Z axis	3100 N
Moment on X axis	400 Nm
Moment on Y axis	-1100 Nm
Moment on Z axis	50 Nm

Table 8.1: Forces and moments on the wing obtained in the CFD **case A** simulation

8.2 Case 2

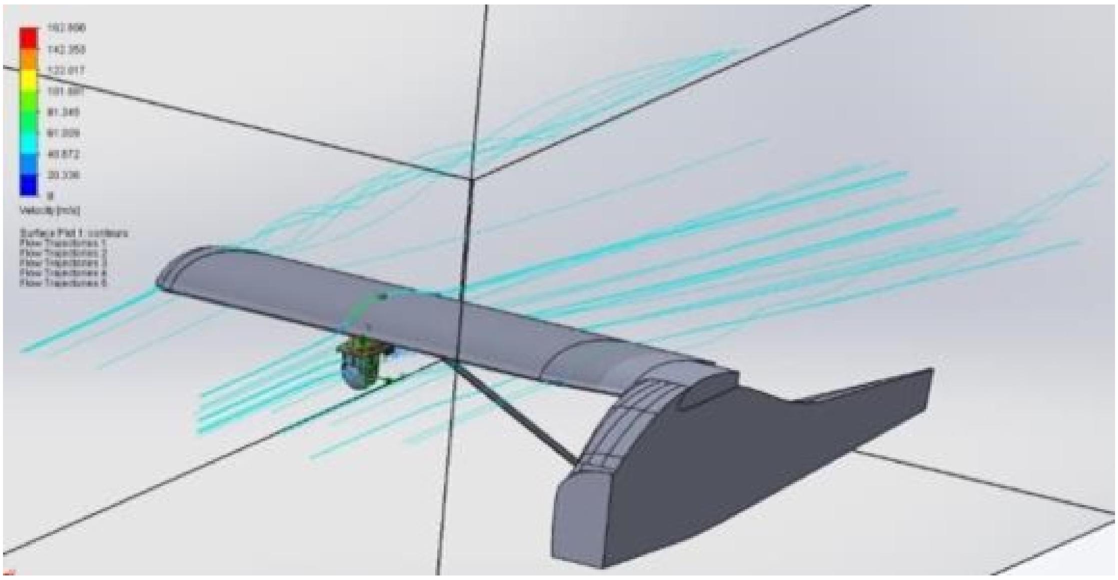

Figure 8.7: The streamline around the wing

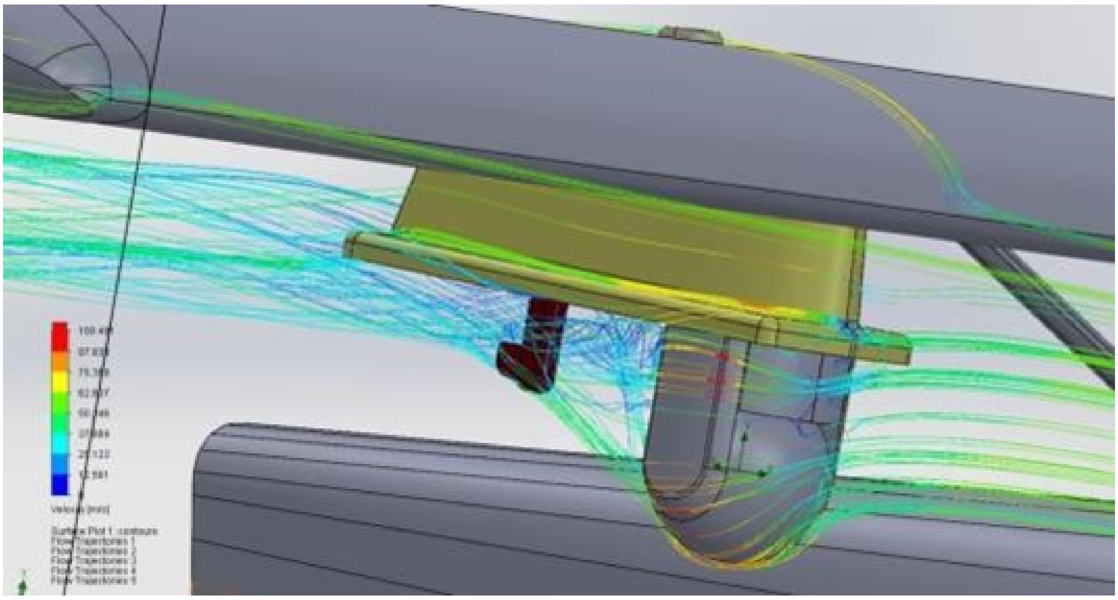

Figure 8.8: A zoom of the streamline around the *SmartGimbal*

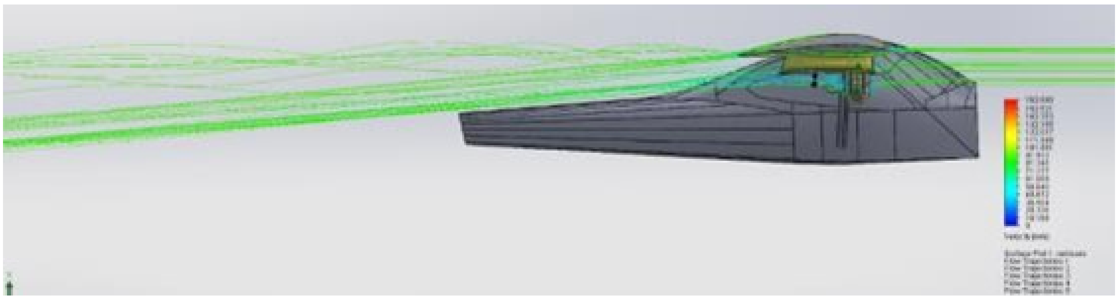

Figure 8.9: The streamline around the wing (lateral view)

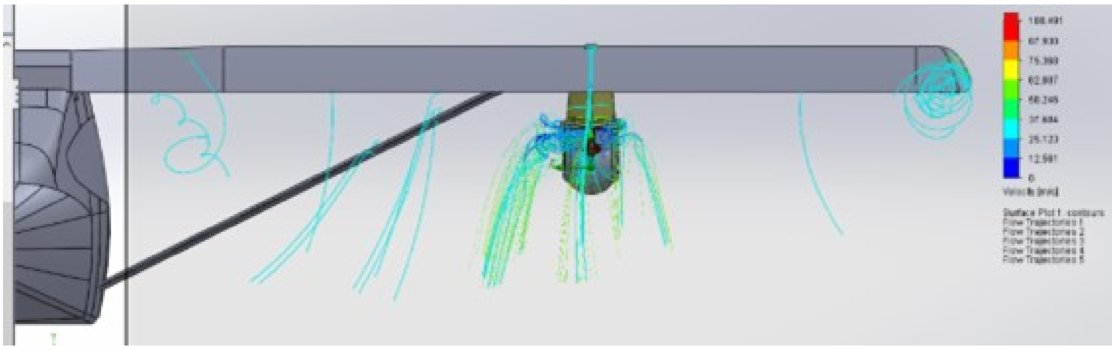

Figure 8.10: The streamline around the wing (rearview)

In the figures is observable behind the *SmartGimbal* a bordeux appendix, it is a support to install a further camera, during the test this support will non be installed;
Being a little part compared to the *SmartGimbal* and being in the wake of the gimbal, its influence can be neglected.

Force on X axis	-260 N
Force on Y axis	125 N
Force on Z axis	2750 N
Moment on X axis	300 Nm
Moment on Y axis	-1050 Nm
Moment on Z axis	125 Nm

Table 8.2: Forces and moments on the wing obtained in the CFD **case B** simulation

8.3 Case 3

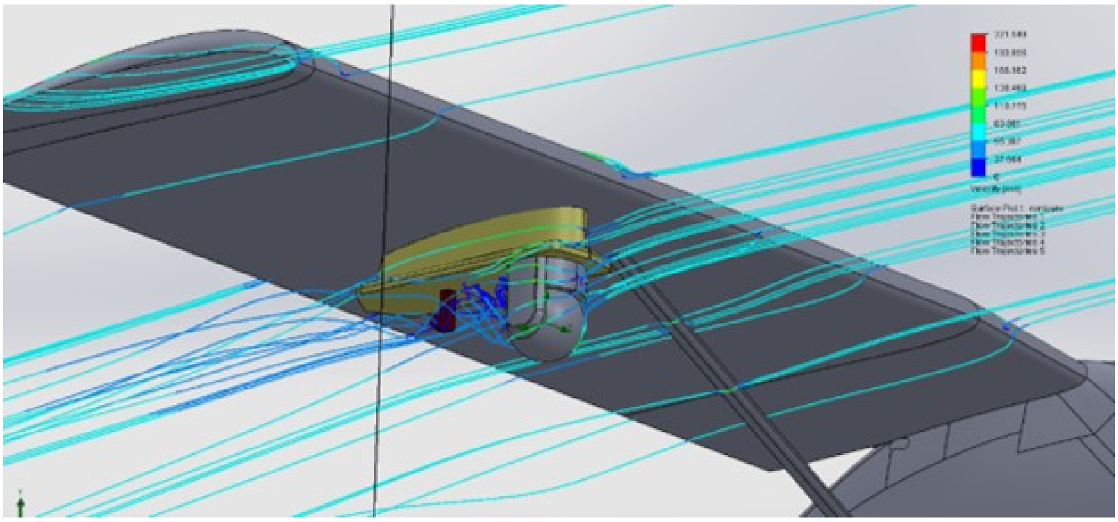

Figure 8.11: The streamline around the wing

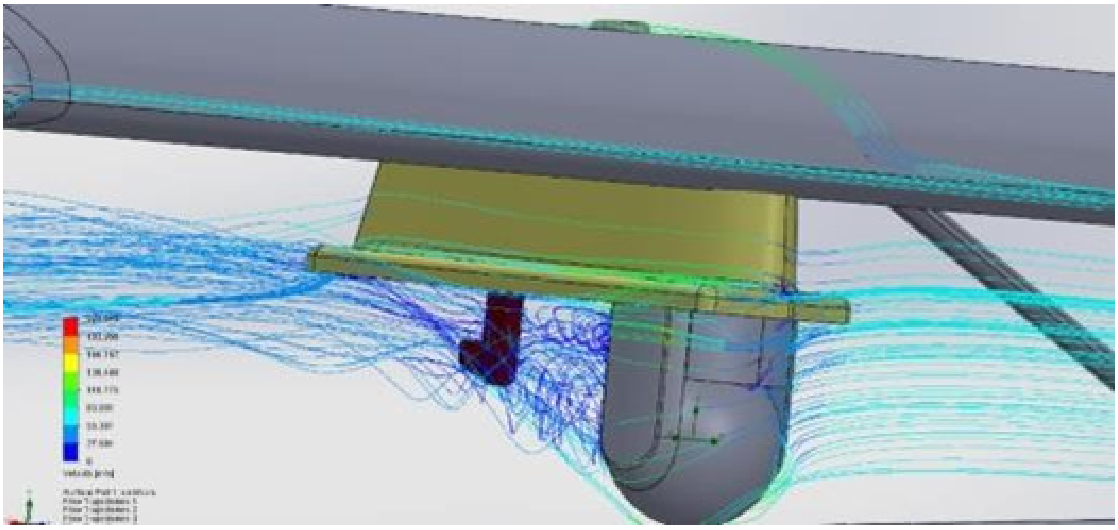

Figure 8.12: A zoom of the streamline around the *SmartGimbal*

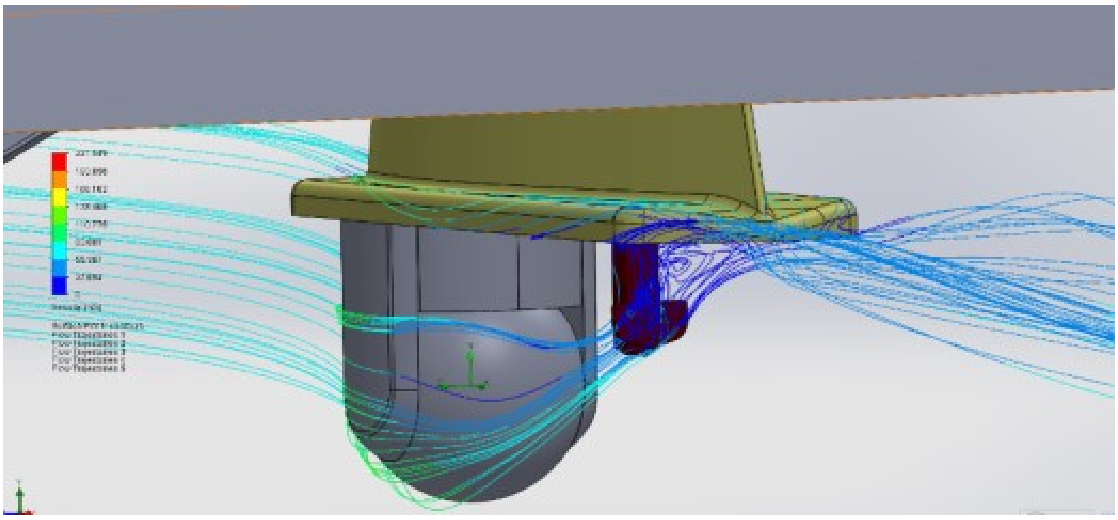

Figure 8.13: A zoom of the streamline behind the *SmartGimbal*

Force on X axis	-500 N
Force on Y axis	70 N
Force on Z axis	5000 N
Moment on X axis	570 Nm
Moment on Y axis	-1750 Nm
Moment on Z axis	250 Nm

Table 8.3: Forces and moments on the wing obtained in the CFD **case C** simulation

8.4 Case 4

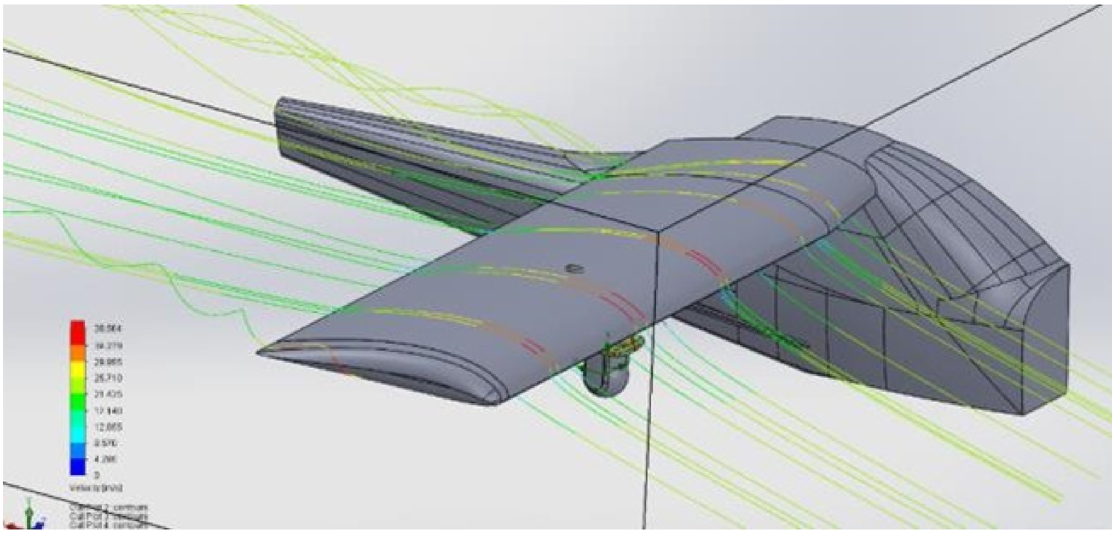

Figure 8.14: The streamline around the wing

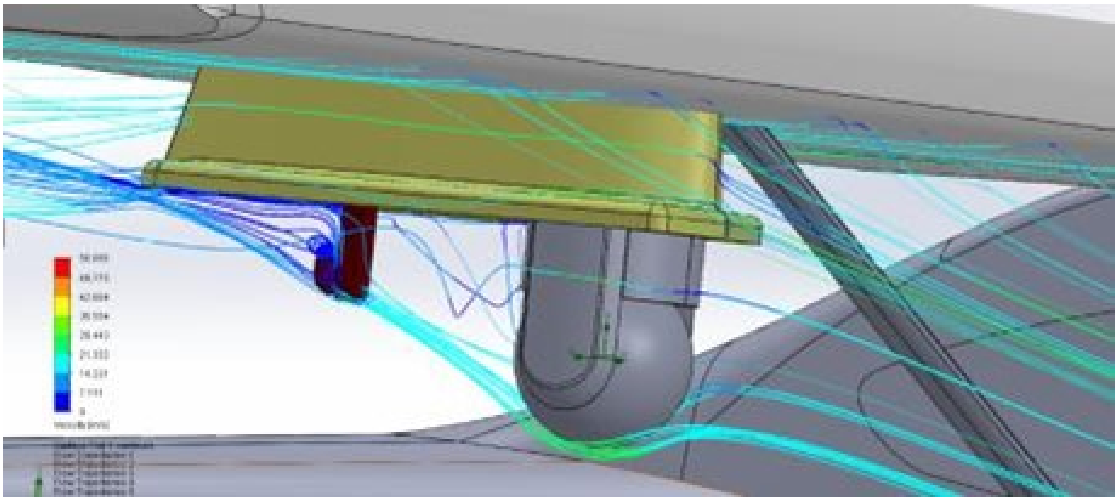

Figure 8.15: A zoom of the streamline around the *SmartGimbal*

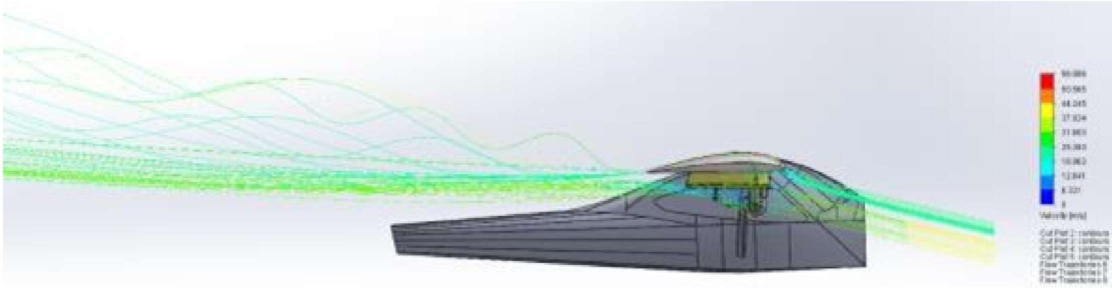

Figure 8.16: The streamline around the wing (lateral view)

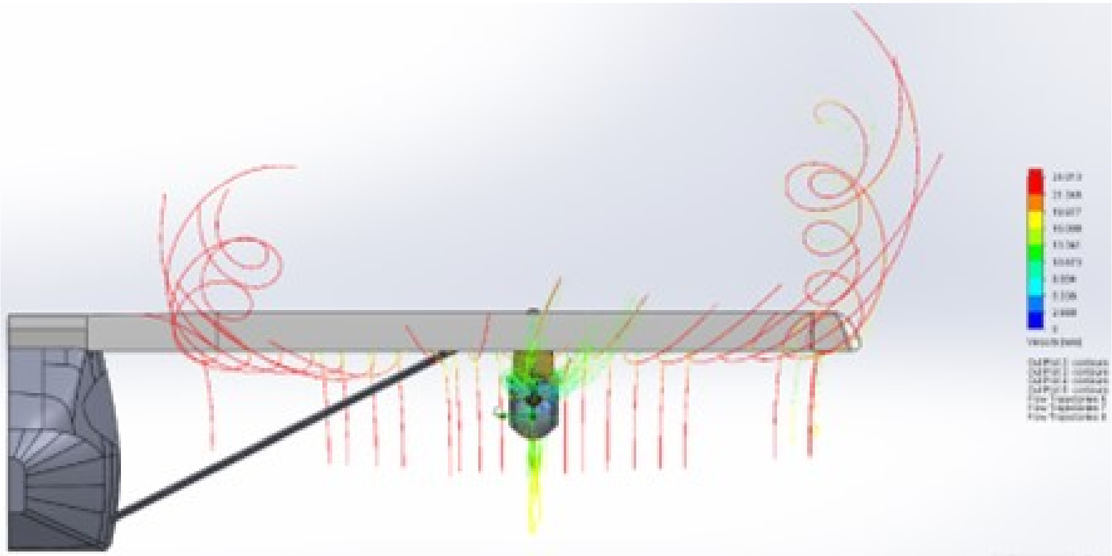

Figure 8.17: The streamline around the wing (rearview)

Force on X axis	100 N
Force on Y axis	-250 N
Force on Z axis	1360 N
Moment on X axis	-150 Nm
Moment on Y axis	60 Nm
Moment on Z axis	20 Nm

Table 8.4: Forces and moments on the wing obtained in the CFD **case D** simulation

Chapter 9

Flight tests

This was the most important phase of the project. The flight tests can be divided in two sections, **Objective test** and **Subjective test**, the latter were carried out after the objective test and had the purpose to perform a "fine calibration" of the control parameters on the basis of the feedback provided by the pilot or operator.

9.1 Objective test

This test has to be conduced in what can be defined as the "operative condition" of the *SmartGimbal*, so with a constant plane speed and trim with a leveled flight, the only variable in this case was the airspeed that applying a pressure on the gimbal structure, changes the loads on the main bearing of the structure.

The test procedure was the same as the ground test, that means to measure the time used to complete a path monitoring its variation for different IAS, these travel times should have a variation less than 10 % assumed as pass criteria.
Again all the test steps have been inserted in a checklist that foresaw to execute all the passages flying every time with a different IAS, his in order to obtain enough data to draw a diagram for time/IAS, in particular the IAS selected were: 110 Km/h, 130 Km/h, 160 Km/h, 190 Km/h.
It was not considered necessary to carry out the test at different altitudes as the excursion of the atmospheric parameters such as density was negligible net of the effects of air velocity;
The test foresaw to execute four tasks for every IAS step, these tasks are:

- PAN rotation test with 5 deg/s angular speed
- TILT rotation test with 5 deg/s angular speed
- PAN rotation test with 17 deg/s angular speed
- TILT rotation test with 17 deg/s angular speed

The first test was performed on the 10-8-2022 at the "Torino Aeritalia" airport (ICAO code: LIMA) on the Tecnam P92 *echo super* marked I-A248, the aircraft used for the basic test of the equipment developed by *DigiSky*.

Figure 9.1: The I-A248 ready for the test

Figure 9.2: The *SmartGimbal* installed on the *SmartBay*

The test has been completed in 38 minutes and the results are showed in these two tables:

IAS [Km/h]	**Travel time 5 deg/s**	**Travel time 15 deg/s**
0	61.68 s	22.27 s
110	67.76 s	21.62 s
130	67,95 s	21.62 s
160	68.04 s	22.27 s
190	68.23 s	21.71 s
AVG Travel time	66.33 s	21.9 s

Table 9.1: PAN axis test results

IAS [Km/h]	Travel time 5 deg/s	Travel time 15 deg/s
0	31.32 s	22.27 s
110	32.94 s	20.78 s
130	33.50 s	20.78 s
160	33.32 s	20.78 s
190	22.18 s	20.78 s
AVG Travel time	30.65 s	21.52 s

Table 9.2: TILT axis test results

This table shows the maximum error encountered in the test for the four conditions listed below, the errors are expressed in terms of percentage of the relative average time shown in the two tables above:

Axis	5 deg/s	15 deg/s
PAN	7%	1.6%
TILT	27.6%	3.4 %

Table 9.3: Errors encountered

The results shown that apart from the TILT axis at 5 deg/s, the maximum error is under the fixed threshold of 10%. About the huge error of 27.6% of the TILT axis the only logic explanation found was an error from the pilot during the test execution; This strong statement comes from the fact that the checklist passage that provides this number, was the final one before the checklist action to switch the test speed from 5 deg/s to 15 deg/s, reading the checklist is possible that the pilot has changed the test speed before doing the current test tasks, so the recorded time actually, corresponds to the condition of TILT axis at 15 rad/s.
Furthermore, to evaluate this hypothesis, there is the fact that the registered time is quite similar to the ones recorded in the following checklist actions, so with the TILT axis of 15 rad/s.
Under this assumption, the *correct* error calculated for the TILT axis at 5 rad/s is 3.4%. Apart from this mistake, no other issues were reported by the pilot.

9.2 Subjective test

The subjective test aim is to obtain a "usable product", it means a recorded video of the flight, that can be analyzed to identify system critical issues such as vibrations and irregular movements.

The plane setup for this test was a bit more complex due to the presence of the video data cable and the display into the cockpit, for this test the *SmartGimbal* was programmed with the Simulink developed SW exposed at the beginning of the document.

9.2.1 Cockpit setup

This test was conduced as a *single pilot test* where the pilot had to manage both the plane and the installed device.

The console had to be installed on a solid support so that the pilot could manage it with a single hand, the used display was a *BlackMagic Design Video Assist* specially designed to work with the BlackMagic cameras;
This display had to be placed so that the pilot could see such as was part of the aircraft flight instrumentation;
This was not the first test on the *DigiSky* P92 plane requiring this setup, so a solution was already developed, it consists of an aluminum panel that can be installed on the copilot side replacing the EFIS panel, this display has also the capability to record the video coming from the camera on a SD card. Here is shown the complete cockpit setup for the test:

Figure 9.3: The cockpit setup for the test

9.2.2 Test procedure

Once installed and tested on the ground the system was ready to take off, the purpose of the test was to point the camera to a target manually and keep it into the field of view using the console joystick in order to verify and eventually modify the PID parameters to obtain a comfortable motion of the gimbal.
The test was conduced in the Susa valley placed perfectly at the end of the LIMA airport take off path, this area is a *G airspace* with very low air traffic so ideal to allow the pilot to focus on the mission.

At the entrance of the valley there the "Sacra of San Michele" monument, this building is located at an altitude of 960 m (3150 ft) above the S.L. and is clearly visible from every direction so its easy to localize by a pilot in flight, moreover being a church it has a lot of details like towers, spiers, windows that can be exploited to analyze the video quality once terminated the mission.
The test foresaw to flight along the stretch of the valley in front of this building, trying tho keep it at the center of the video.

Figure 9.4: The Sacra of San Michele

Figure 9.5: The I-A248 ready for take off

9.2.3 Test results

The test was conduced on the 12-10-2022 in the morning and the complete flight takes about 30 min, visibility was not the best, there was haze which made the video a bit blurred, here are showed some frames of the recorded video:

Figure 9.6: The LIMA airport view after the take-off

Figure 9.7: Some highways near the LIMA airport

Figure 9.8: Approaching the target

Figure 9.9: The south facade of the building

Figure 9.10: The west facade of the building

Figure 9.11: The east facade of the building

Figure 9.12: A digital zoom on the building

Figure 9.13: The final approach to the LIMA airport

The critical issues emerged during the test were discussed during the test debriefing after the flight, the video quality seems great, but the haze did not allow evaluating it at the best, the stability of the system is good only very slight vibrations are present but nothing that could affect the video quality. The only appreciable flaw based on the pilot opinions was the joystick sensitivity, the gimbal response is a bit impulsive;
Being a device designed for slow and precise movements, will be necessary to reduce the "system reactivity" decreasing for example, the PI proportional parameter.

Chapter 10

Conclusion and future improvements

After the tests have confirmed the functionality of the project, the *SmartGimbal* system can be considered terminated, however technical improvements for several components are still possible to make the system more powerful.
These possible modifications will be taken into consideration together with those that will come out from the operative life and the new technologies and methodologies that will be developed by *DigiSky* company. At the moment the possible improvements of the system are listed below:

- **PAN actuator issue:** the location of the PAN actuator and its toothed gear causes the belt to be in a position for which its correct tensioning is difficult. This lack of belt tension causes a delay in the movement of the PAN axis, resulting in a loss of precision of the rotation. The belt tensioning mechanism needs improvements.

- **VR viewer implementation:** currently the operator looks at the camera video using a monitor installed into the plane, to improve the quality of the recorded video helping the operator focusing on the target object, a VR viewer device will be helpful, this in order to isolate the operator from external factor such as the sunlight reflecting on the screen disturbing the visualization.
  The feasibility of using a highly immersive device such as a viewer in an environment full of movements, such as that of an airplane in flight, must be carefully analyzed to avoid discomfort for the operator due to very different and also contrasting somatosensory information.

- **Aerodynamic fairing improvement:** to install the camera inside the gimbal is necessary to remove the front spherical case of the fairing, not a quite comfortable solution, a possible improvement will be to develop a locking system to ensure an easier and faster access to the gimbal interior.

- **C code development**: At the current state of the art of the project the working software is implemented with **Simulink**, and it is able to work ONLY with Arduino

board from the UNO to powerful models [1]. To ensure the functionality of the software with other microcontrollers it is necessary to have a suitable software for all, that's mean a C code with libraries that can be changed basing on the used microcontrollers.
A base implementation of this code already exists, it is the software used for the "objective test". In this code are already implemented the PID logic and the UART routines.
What still need to be implemented is the code relative to the joystick reading and the related actuators commands. Luckily the Arduino community provide a huge amount of library compatible or dedicated to a specific microcontroller so this code libraries and eventually commands can be easily updated consequently.

- **Auto-tracking mode**: a useful feature for the *Smartgimbal* system will the automatic tracking of a target, using the data coming from the NMEA flight data line available on the *SmartBay* platform. To do this a specific software must be implemented, in particular is necessary to write an algorithm that allows the conversion of a 3 DOF reference system (pitch, roll, yaw) in a 2 DOF reference system (pan, tilt).

[1] https://it.mathworks.com/hardware-support/arduino-simulink.html

List of acronyms

SDI	Serial Digital Interface
UI	User Interface
GNSS	Global Navigation Satellite System
IMU	Inertia Measurement Unit
EASA	European union Aviation Safety Agency
NAVCOM	aeronautical radio-Navigation and Communication services
EMI	Electromagnetic interference
SW	SoftWare
UART	Universal Asynchronous Receiver-Transmitter
GPIO	General-Purpose Input/Output
I/O	Input/Output
I2C	Inter-Integrated Circuit, eye-squared-C
RISC	Reduced Instruction Set Computer
AVR	Alf and Vegard's RISC
PWM	Pulse Width Modulation
IC	Integrated Circuit
RC	Radio Controlled
TTL	Transistor Transistor Logic
NRZ	Not Return to Zero
PCB	Printed Board Circuit
ESD	Elettro-Static Discharge
PI	Proportional Integral
IAS	Indicated Air Speed
EFIS	Electronic Flight Instrumentation system
IDE	Integrated Development Environment
CFRP	Carbon Fiber Reinforced Polymer
IAS	Indicated Air Speed
RS	Reference System
CFD	Computational Fluid Dynamics
NMEA	National Marine Electronics Association
ESC	Electronic Speed Controller
DC	Direct Current
LCD	Liquid Crystal Display