POLITECNICO DI TORINO

PhD. Course – XXV Cycle
Aerospace Engineering

Doctorate Thesis

Navigation system for the remote management of unmanned aircraft

ING/IND 03

Tutors:
Prof. Giulio Romeo
Dott. Stefano Frazzetta

PhD. Candidate:
Marco Pacino

December 2012

Index

Figure Index

1 Introduction

The development of different kinds of Unmanned Aerial Vehicles (UAV) brought to an increment of employment of such platforms both in military and civil applications. While in civil applications the range is usually limited to the presence of a pilot in Line Of Sight (LOS) for airworthiness requirements, the unconstrained use of UAVs in military scenarios brought to extend the range of control to a Beyond Line Of Sight (BLOS) satellite communication. Anyway, in vision of airworthiness improvement, also for civil applications the BLOS control has to be considered. The remote control of a BLOS UAV introduces some complexities that will be presented in this work, in particular due to the latency of the signal and to the need of hand-over to switch from LOS control to BLOS and vice versa.

In this work will be discussed the problematic of the UAV navigation for the remote management in vision of a BLOS capability showing the research experience developed in collaboration with Alenia Aermacchi, focusing on the solution adopted for increasing the navigation automations and the system interoperability. The first section presents an overview of such problems and the guideline followed to provide a relevant solution. The second section exhibits the effective implementation of the studied cases on the Alenia Aermacchi Sky-Y UAV focusing on the navigation issues. During the navigation functions development a large use of simulators occurred: one Matlab® model and one Simulink® simulator have been developed for this purpose, then the proper Alenia Aermacchi Sky-Y flight simulator has been used before the RIG tests. In the last part of this work some results of such simulations are presented by showing some significant cases of navigation test done with the different tools mentioned above.

2 The UAV

The UAV: Unmanned Aerial Vehicles, represent a great growing sector of the aeronautic both military and civil since the last ten years of the XX century. This has been possible thanks to the technologies rapid development as electronic components miniaturization, high calculation capability, global position systems, that, thanks to the mass diffusion, have strongly reduced the costs permitting the growth of many research activities [1]. Today almost all the kind of configurations have been considered for the unmanned aircrafts: fixed wing, rotorcraft, seaplane, balloons, propeller engine, turbine propulsion, electric motor, single engine, multi engine, etc. Many configuration with the same purpose: to perform the so called Dull, Dirty and Dangerous missions, all situations in which the pilot onboard is not recommended. The so fast growing up of the UAV technology has not been supported from so much fast development of regulations for the UAVs employment, fact that represents a limitation specially for the civil applications, anyway the experimentation in civil field is usually possible in segregated airspaces waiting for a future integration with the civil air traffic.

Different classifications of UAV can be made, the Fig 1 shows the subdivision of UAV respect to the aerial vehicles, while in Fig 2 is tabled the UVS International classification for UAV based on the range, altitude, endurance, and maximum take-off weight (MTOW).

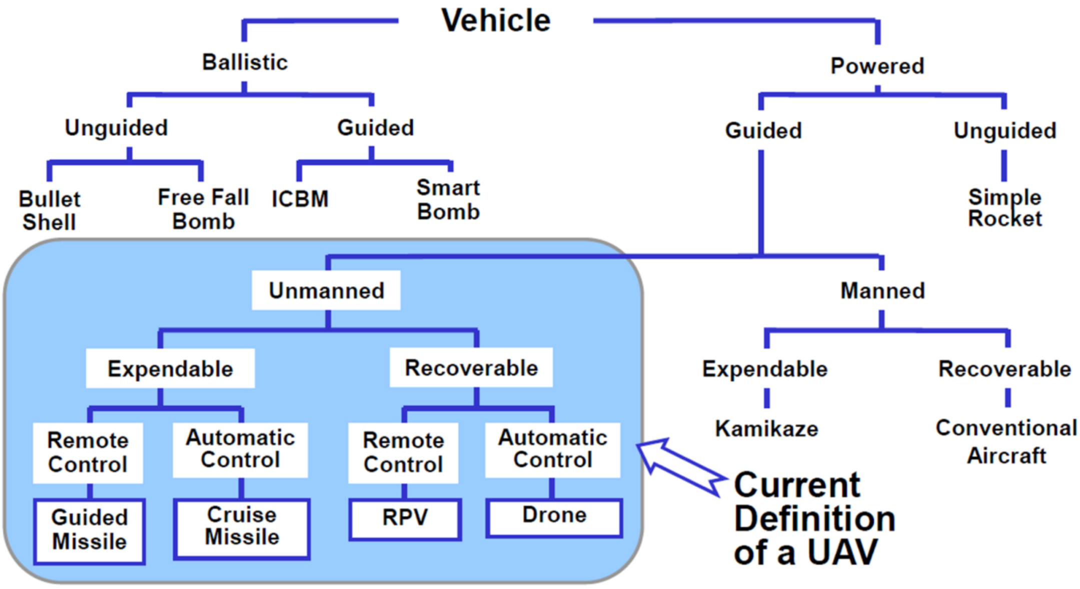


Fig 1 UAV subdivision respect to the aerial vehicles [2].

Category	Acronym	Operative range [km]	Operative Altitude [m]	Endurance [h]	MTOW [kg]
Tactical UAV					
Nano	η	< 1	100	< 1	< 0,0250
Micro	μ	< 10	250	1	< 5
Mini	Mini	< 10	150 - 300	< 2	< 30
Close Range	CR	10 - 30	3 000	2 - 4	150
Short Range	SR	30 - 70	3 000	3 - 6	200
Medium Range	MR	70 - 200	5 000	6 - 10	1 250
Medium Range Endurance	MRE	> 500	8 000	10 - 18	1 250
Low Altitude Deep Penetration	LADP	> 250	50 - 9 000	0,5 - 1	350
Low Altitude Long Endurance	LALE	> 500	3 000	> 24	< 30
Medium Altitude Long Endurance	MALE	> 500	14 000	24 - 48	1500
Strategic UAV					
High Altitude Long Endurance	HALE	> 2 000	20 000	24 - 48	12 000

Special purpose UAV					
Unmanned combat aerial vehicle	UCAV	1 500	10 000	2	10 000
Lethal	LETH	300	4 000	3 - 4	250
Decoy	DEC	0 – 500	5 000	< 4	250
Stratospheric	STRATO	> 2 000	> 20 000 & < 30 000	> 48	Tbd
Exo – stratospehric	EXO	Tbd	< 30 000	Tbd	Tbd
Space	SPACE	Tbd	Tbd	Tbd	Tbd

Fig 2 UVSInternational UAV classification [3].

2.1 UAV history

The story of UAV starts prior to that of the invention of the aircraft with the use of unmanned balloons for military use. The earliest recorded use of such a vehicles occurred on August 22, 1849, during the Austrians attack of Venice where these unmanned balloons were loaded with explosives [4]. The use of unmanned balloons carried with explosive with a time dropping mechanism was recorded also in 1863 during the American Civil War, although the vehicle control was not possible.

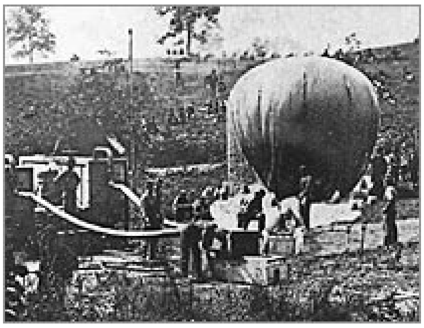


Fig 3 American Civil War balloon bomber.

In 1883 Douglas Archibald was the first in taking aerial photography by mounting a camera on a kite. A similar system was used in 1898 as a surveillance method during the Spanish-American war. The first radio-controlled UAV appeared in 1917 thanks to a gyro stabilizer system for straight flight, it was thought for military purpose but, despite several flight tests, it was never used in real operations. In 1935 in Britain was built the *Queen* radio controlled airplane and after that the *Queen Bee* RC biplane were built in great number and used as targets.

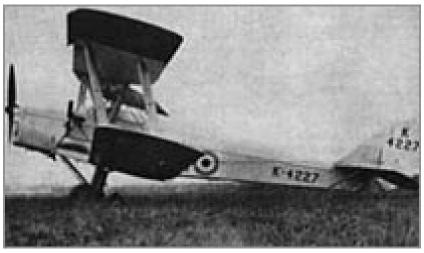


Fig 4 Queen Been.

The same use were made by the U.S. Army with the RC aircrafts built by the Radioplane Company: in 1935 the first prototype, the *RP-1*, as military target was demonstrated; then, after two following versions, in 1940 the *RP-4* won an army contract and became the *Radioplane OQ-2*. After this a large number of remotely controlled aircraft have been developed and used in the following years, mainly for military purposes.

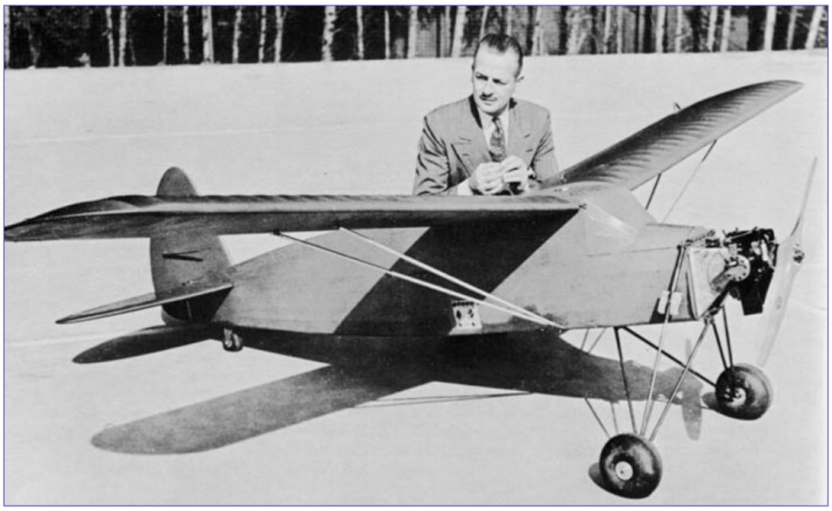


Fig 5 RP-1.

In the early years a large use of UAV has been done by the army and many UAV have been developed for civil purposes.

.

Fig 6 Global Hawk.

Since the 1995 the General atomics Global Hawk HALE UAV has been largely used by the US army starting with the first deploy in the Balkans, then in Iraq, and after that in Afghanistan. In the same years made the first flight also the MALE UAV Predator, always manufactured by the General Atomics, for monitoring and fighting purposes.

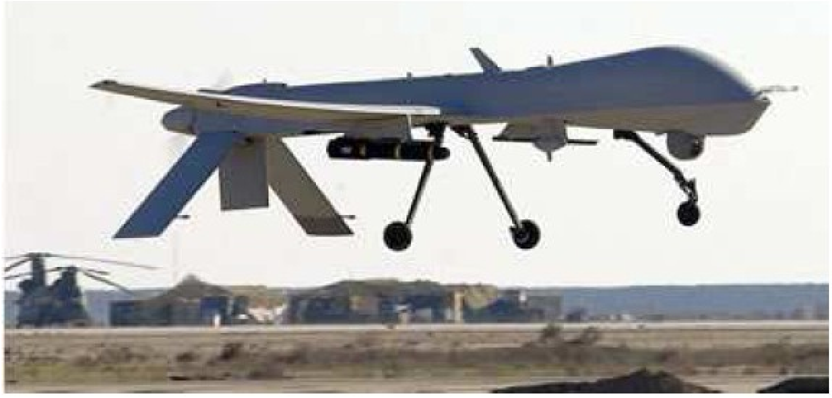


Fig 7 Predator.

Continuing in the military field, a great number of mini UAV have been developed, to be used by the marines as portable low cost strategic survey platforms for low range missions defined ‘over the hill’ or ‘around the corner’[5].

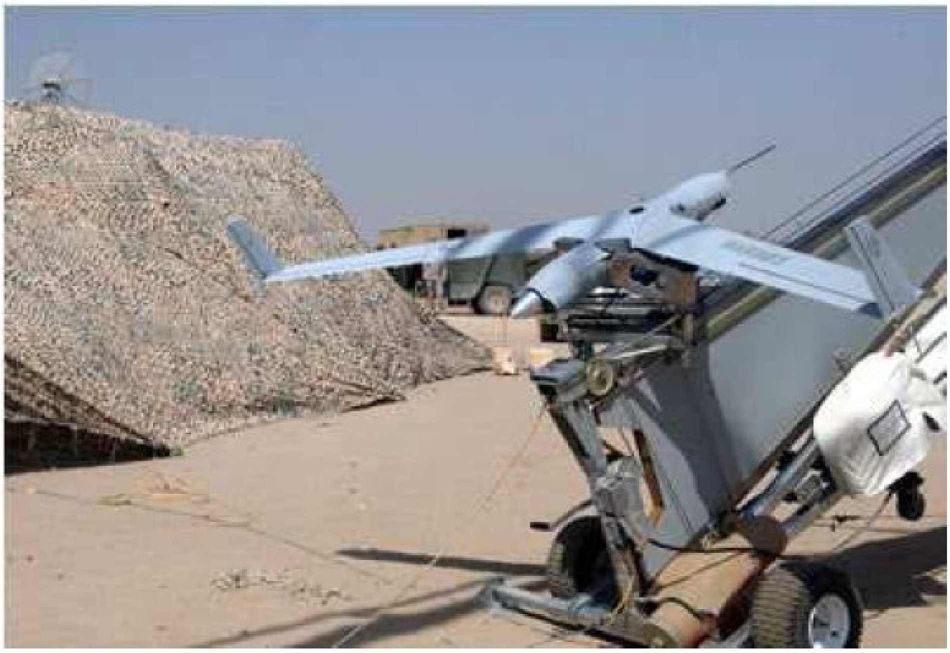


Fig 8 ScanEagle.

Also in Italy the aeronautic industry research started the UAV study, and in 2005 the Alenia Aeronautica Sky-X technological demonstrator made the first flight.

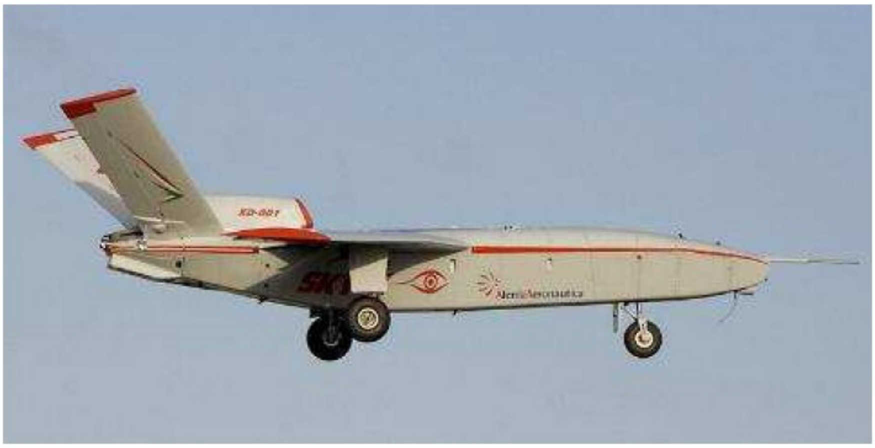


Fig 9 Sky-X.

Then the Alenia Aermacchi research continued with a propeller MALE UAV: the Sky-Y, that, as part of the SMAT project, it would be tested for civil applications.

For civil applications in fact the UAVs can be used for a wide number of purposes: agriculture, pipeline monitoring, border patrol, telecommunication, urban photography, fire detection, search and rescue, etc. One of the reason for using unmanned vehicles for this kind of operations is the mission repetitiveness, but, at the same time, without man on board, the need of landing for crew rest is also avoided. So that started a field of research on very long endurance UAV mainly for monitoring purposes. Such research leaded in the 1980s in the solar powered UAV development for increasing the endurance. NASA developed the Helios HALE UAV intended to have a night and day continuous flight by storing energy during the day with photovoltaic solar cells mounted over the wing. One technological demonstrator was built, as follower of 10 years of previous research and prototypes, but in June 2003 crashed for a structural failure [6].

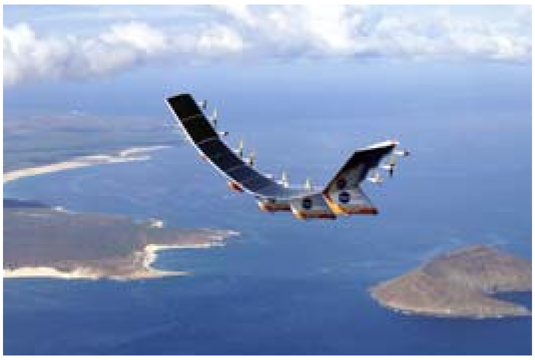


Fig 10 Helios.

Also in Europe the study of this kind of UAV occurred and between the 200 and the 2003 the European Program Heliplat run the study of a 73m wingspan HALE stratospheric platform, of which the Poltecnico di Torino manufactured a scaled prototype to perform the structural tests [7].

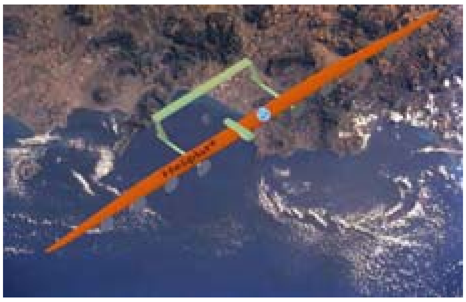


Fig 11 Heliplat.

Then a shorter wingspan solar UAV have been designed as the 2005 Solong which realized a 48 hours continuous flight over the Colorado desert by using both solar cells and thermals from the desert. In 2005 the QinetiQ developed the Zephyr that in 2008 performed the world record of 82 hours continuous flight at an altitude of 61000 ft.

Fig 12 Zephyr.

This kind of UAV, flying at stratospheric altitudes and covering high distances, requires a communication system with beyond line of sight capability, and so an high automated navigation system as well as the military UAVs to adopt in far foreign landscapes. So that the satellite communication and the integration of satellite data-link began to be considered for the UAV management.

3 Satellites

In this contest satellites are defined as artificial object placed into orbit around the Earth by the man. Since the first artificial satellite, the Sputnik I, was launched in 1957 from the Soviet Union, thousand of satellites have been launched into orbit from more than fifty nations. Satellites are used for many purposes both military and civil: communications, Earth observation, navigation, weather and research.

Fig 13 Communication satellite.

Many different classifications can be done according to different satellite characteristics. The altitude classification includes:

- Low Earth orbit **(**LEO**)**: Geocentric orbits ranging in altitude from 0–2000 km.
- Medium Earth orbit **(**MEO**)**: Geocentric orbits ranging in altitude from 2,000 km to just below geosynchronous orbit at 35,786 km. Also known as an intermediate circular orbit.
- High Earth orbit **(**HEO**)**: Geocentric orbits above the altitude of geosynchronous orbit 35,786 km.

HEO are also called Geostationary Earth Orbit, this means that the satellite angular velocity is the same of the earth, so the satellite would always be over the same point of the earth. This allows constant coverage of the same area and eliminates blackout periods typical of the other type of satellites. However their high altitude causes a long signal delay: a signal from the ground that has to reach the satellite and come back, has to cover a distance of 72,000 km.

LEO type reduces transmission times as compared to HEO. It should be also used to cover a polar region, which the HEO cannot do. However, since it is not geostationary, the earth stations need to track the motion of the satellite with their antennas [8].

MEO type have characteristic in the middle of the two mentioned above.

To cover simultaneously every point on the earth a constellation is required, so that a series of satellites would cover all the desired zones and/or alternately pass over the target zone.

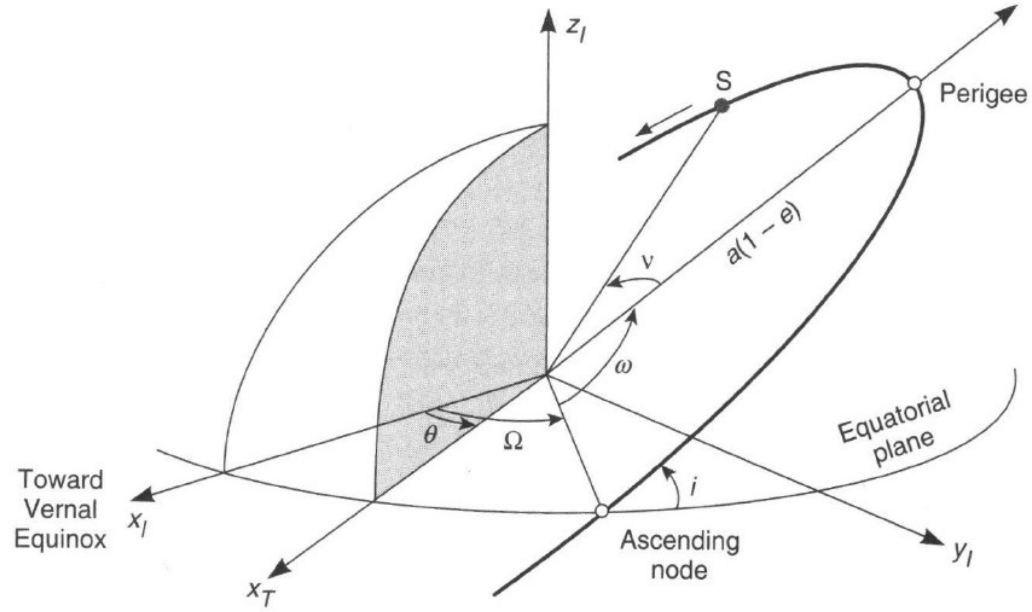


Fig 14 Keplerian orbit elements.

3.1 Satellite and UAV

Satellites and UAV have been often discussed together for many reasons. From one hand because a UAV branch involved some functions usually provided by satellites, like global monitoring and communication; but mainly because of the satellite use for the UAV management. In the UAV field the use of satellite occurred at the beginning for navigation purposes: the GPS; then, in the last years, for communication: to control vehicles and payload when the direct link falls out of the range.
The following two sub-sections will briefly introduce these two functions.

GPS

The Global Positioning System (GPS) is a space-based radio-positioning and time transfer system. In the past decade, GPS has grown into a global utility providing space-based Positioning, Navigation and Timing. Nowadays GPS provides a global coverage and is owned and operated by the U.S. Government. The GPS service is a one-way broadcast (like FM radio), with an unlimited number of users [9].
It is constituted by the satellites that compose the constellation, each of them transmits continuously its ranging signal including the navigation message [10].
Fig 15 presents the GPS satellites constellation with the relative planar projection referenced to the epoch of July 1st, 1993 UTC (Universal Time Coordinates).

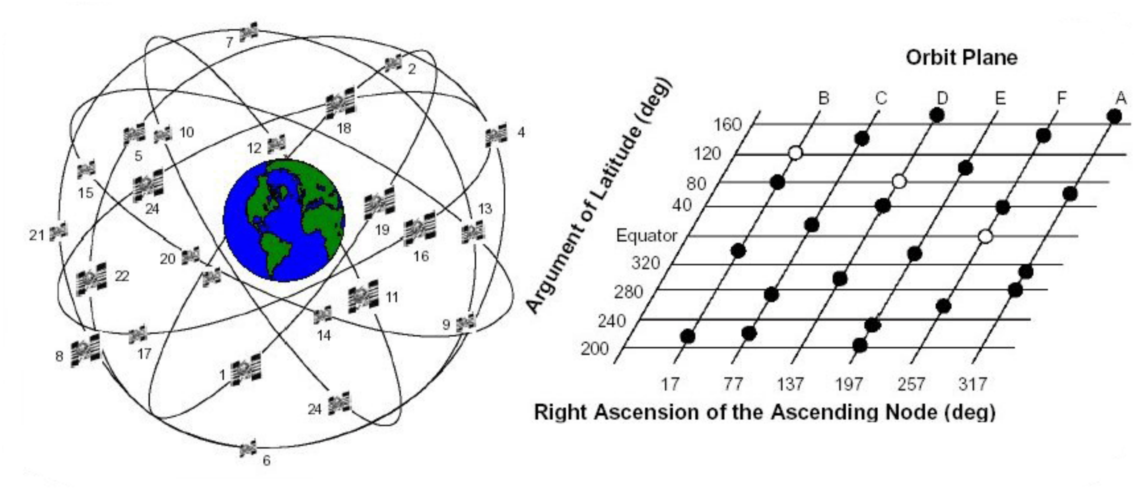
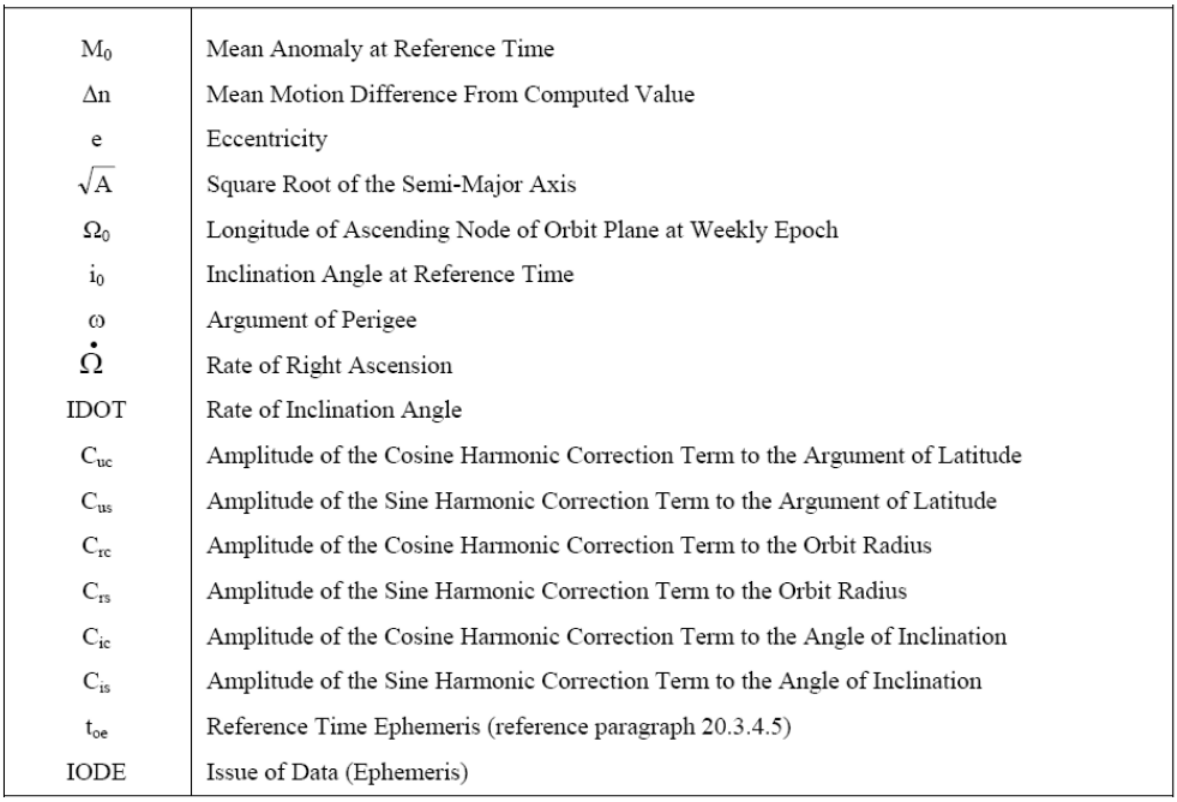


Fig 15 GPS constellation.

A proper use of the signals broadcast by every GPS satellite allow the users to estimate instantaneously and in real time its Position, Velocity and Time on the Earth surface (or in flight). Such information, known as ephemeris, are shown in Fig 16.



M_0	Mean Anomaly at Reference Time
Δn	Mean Motion Difference From Computed Value
e	Eccentricity
$\sqrt{A}$	Square Root of the Semi-Major Axis
Ω_0	Longitude of Ascending Node of Orbit Plane at Weekly Epoch
i_0	Inclination Angle at Reference Time
ω	Argument of Perigee
$\dot{\Omega}$	Rate of Right Ascension
IDOT	Rate of Inclination Angle
C_{uc}	Amplitude of the Cosine Harmonic Correction Term to the Argument of Latitude
C_{us}	Amplitude of the Sine Harmonic Correction Term to the Argument of Latitude
C_{rc}	Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius
C_{rs}	Amplitude of the Sine Harmonic Correction Term to the Orbit Radius
C_{ic}	Amplitude of the Cosine Harmonic Correction Term to the Angle of Inclination
C_{is}	Amplitude of the Sine Harmonic Correction Term to the Angle of Inclination
t_{oe}	Reference Time Ephemeris (reference paragraph 20.3.4.5)
IODE	Issue of Data (Ephemeris)

Fig 16 Ephemeris data.

GPS utilizes the Time-of-Arrival concept in order to determine the user position: it consists of the measure of the time for a signal transmitted by a satellite at a known location to reach a user receiver. This time is then multiplied by the speed of light to obtain the distance between the receiver and the satellite. By measuring the propagation time of signals broadcast from multiple satellites at known locations, it is possible to determine the receiver position [10].

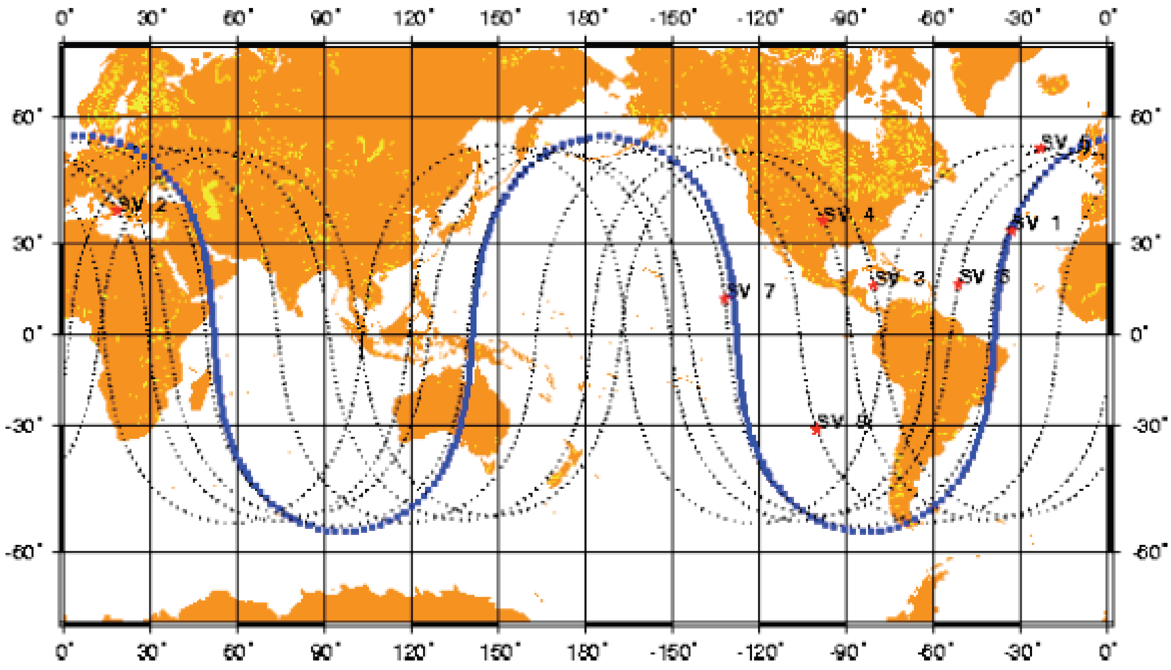


Fig 17 orbit in Earth-fixed frame.

Satellite communication

In satellite communication the information transmitted by a source use the satellite as a relay to reach the receiver. This is used in UAV to increase the range of control beyond the line of sight. The signal, which is a modulated beam, is received by the satellite amplified and then sent back to the receiver. Typically this happens, in UAV, bi-directionally between the control station and the unmanned aircraft.

The satellite main component for communication are antenna and transponder. The antenna receives the original signal from the Earth transmitting source and re-transmit this signal to the receiver. Modern satellites uses high gain antennas (for *antenna gain* see section 4.2.2) pointing towards the area they are providing the service.

The transponder filters and translates the signals received from Earth and then redirects them to the transmitting antenna on board. A large number of transponders (even more than 24) are usually carried on communication satellites to deliver multiple channels (carriers) of communication at the same time [8].

In the next section will be presented an overview of the data-link communication basic principles where the antenna behavior is described as well as the satellite frequency bands (section 4.3).

4 Data-Link

The data-link represents the connection between the pilot in the control station and the aerial vehicle, for the command and control and for the payload management and video transmission. To understand how does it work and what are the characteristics of a data-link communication signal is necessary to understand the electromagnetic wave propagation in the radio frequency band.

4.1 The electric communication

The electric communication consists in the transmission of information, in form of electric signal, between two points separated by a certain distance. The information in the telecommunication field is intended as an electric signal representing sounds, images, or simply codified impulses. One of the fundamental characteristics of the information type is the frequency band in which is bounded. The bound defines the signal quality: an acoustic signal is understandable also if its electric signal is bounded in a range of 300 Hz and 3400 Hz, anyway the characteristic that results altered is the tone (the signal intensity) due to the modification of the harmonics into which the signal can be divided [11].
The scheme of Fig 18 represents the block diagram of the electromagnetic signal transmission and reception.

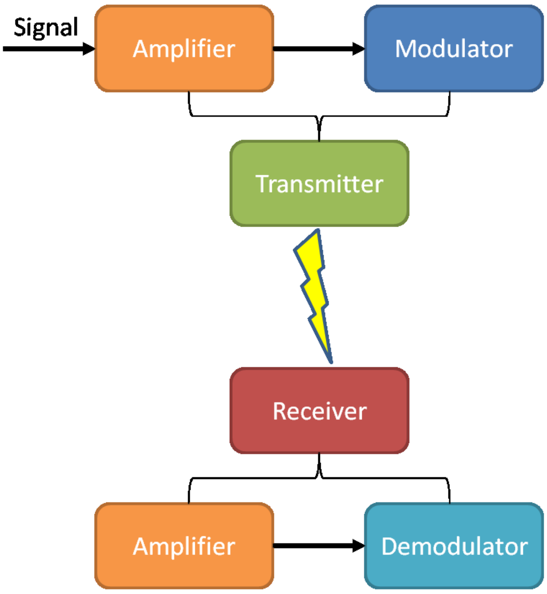


Fig 18 Transmitter and receiver block diagram.

In the Amplifier module the electric signal representing the information is introduced. But since the amplitude (tension, current, power) is low, the signal is first amplified, then modulated in amplitude, frequency or phase from the Modulator, and so propagated in the field that separates the two points. During the propagation anyway the information degrades, it is so necessary to re-amplify the signal before demodulate it. After the propagation the received signal results also affected by new components not present in the original signal known as noise.
The propagation signal power reduction represents the first limitation in the power transmitted and so received. In first approximation the spreading of signal of wavelength λ between two points at distance R can be modeled in as follows:

$$\left(\frac{\lambda}{4\pi R}\right)^2$$

As the wavelength is the inverse of the frequency with the light speed propagation constant, can be noted that halving the frequency the signal spreading increase of four times.

Between the two points the signal propagates as an electromagnetic wave. Considering two conductors, insulated between them, applying with a generator a sinusoidal signal of a certain frequency and power, between the two conductors an electric field E will be generated. The electric field will vary in the time at the alimentation frequency and can be described with a vector in the plane of the two conductors. So that the insulator between the two conductors is covered by an electric current at the same frequency and can be modeled as a capacity. Such a current generates a magnetic field H variable that can be modeled with a vector orthogonal to the two conductors plane. The simultaneous presence of an electric field and a magnetic field orthogonal each other generates a third vector orthogonal to the previous two and called Poyinting vector *S*. The Poyinting vector express the electromagnetic wave power by defining the instantaneous power density (E/m^2) of the electromagnetic radiation in each point.

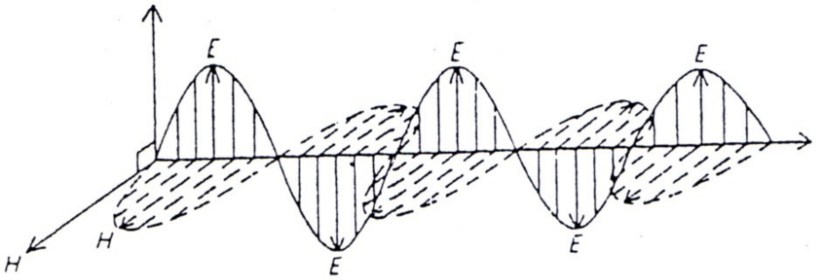


Fig 19 Electromagnetic wave propagation.

4.2 Antennas

In the telecommunication field the antennas have the role of the two conductors above described. The antennas should irradiate as much power as possible, can be demonstrated that the power irradiate from a generic dipole is:

$$P_i = \left(\frac{l}{\lambda}\right)^2 I^2$$

Where l is the dipole length, I is the effective value of the signal current in the wave body, and λ is the signal wavelength. As can be noted from the above formulation, the antenna works as a resistor:

$$P_i = RI^2$$

Where R results to be the radiation resistance: the value that, inserts in series with the antenna and crossed by the signal current, would dissipate the same value of power irradiate by the antenna.

In the real case, the antenna does not provide exactly the same power received by the generator, but it absorb part of the power during its work. Modeling the antenna power dissipation with a resistance cross by the signal current, called equivalent resistance, the overall power required by the generator is $P = P_R + P_i$, where P_R is the power dissipated. So that the antenna efficiency is given by:

$$\eta = \frac{P_i}{P_i + P_R}$$

Such considerations are valid both for transmitting and receiving antennas.

4.2.1 Directive antennas

In many cases is useful to use omnidirectional antennas to cover all the possible directions. Anyway, where the receiver position is note, is more convenient to direct the antenna transmission signal in a specific direction in order to increase the signal power in that direction and so augment the effective range. A directive antenna produce a radiation with a particular shape as shown in Fig 20.

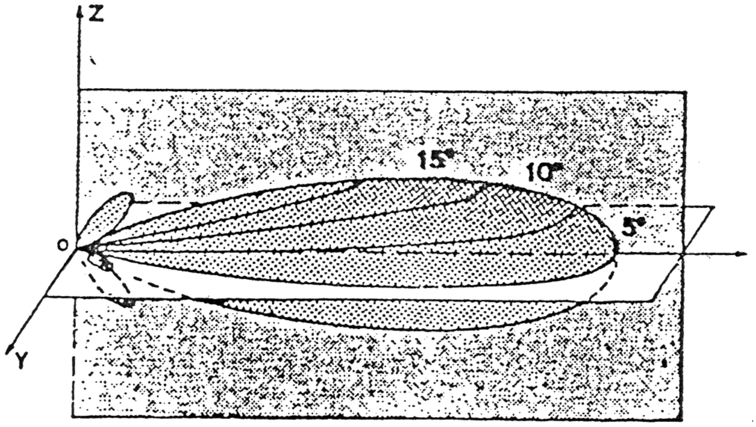


Fig 20 Directive antenna lobe.

The radiation of a directive antenna has a main lobe useful for the transmission and secondary lobes useless and dissipating energy, but often inevitable.

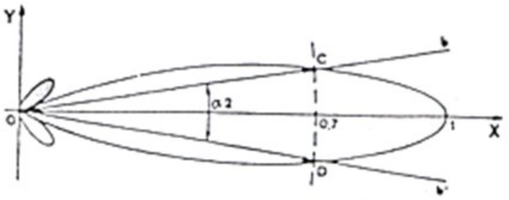


Fig 21 Directive antenna lobe in XY plane.

In Fig 21 is described the radiation of a directive antenna in the XY plane. The directivity is provided by an angle defined as the angle between the main lobe axis and the direction in which the power intensity is 0.707 times the maximum power.

4.2.2 Antenna gain

The ratio between the power of an isotropic antenna and a directive one to generate the same field at a certain distance, along the maximum power direction, is called gain.

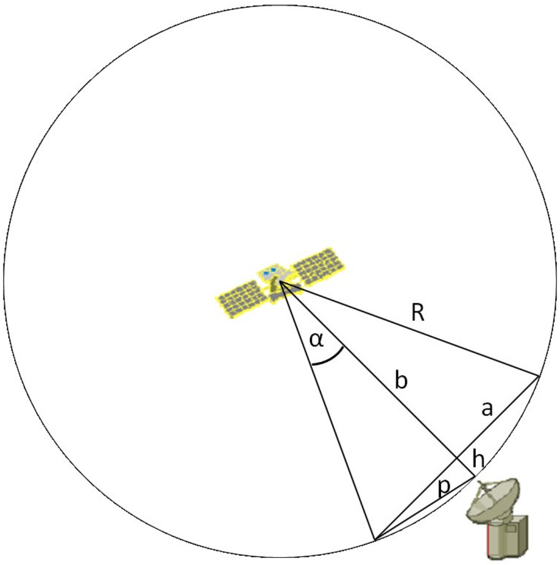


Fig 22 Gain calculation geometrical elements.

The gain is defined as $G = \frac{4\pi R^2}{\pi p^2}$, considering Fig 22:

$a = R\sin\alpha\,, \qquad b = R\cos\alpha$

$h = R - b = R(1 - \cos\alpha)$

$p = \sqrt{h^2 + a^2} = R\sqrt{2 - 2\cos\alpha}$

So the gain result: $G = \frac{2}{1-\cos\alpha}$

Being a value dependent from the angle α, which is half of the directivity cone, the gain is a measure of how directive is the antenna.

4.3 Frequencies

In relation with the frequency and the use, an international classification of the electromagnetic waves has been done:

- radio band for civil applications:

1-10 kHz	VLF (very low frequency)
10-100 kHz	LF (low frequency)
100-1000 kHz	MF (medium frequency)
1-10 MHz	HF (high frequency)
10-100 MHz	VHF (very high frequency)
100-1000 MHz	UHF (ultra high frequency)
1-10 GHz	SHF (super high frequency)
10-100 GHz	EHF(extremely high frequency)

- Military band for radar applications:

1-2 GHz	L Band
2-4 GHz	S Band
4-8 GHz	C Band
8-12 GHz	X Band
12-18 GHz	Ku Band
18-27 GHz	K Band

27-40 GHz	Ka Band
40-75 GHz	V Band
75-110 GHz	W Band
110-300 GHz	mm Band
300-3000 GHz	µmm Band

- Satellite band:

S Band	1700-3000 MHz
C Band	3700-4200 MHz
Ku1 Band	10.9-11.75 GHz
Ku2 Band	11.75-12.5 GHz
Ku3 Band	12.5-12.75 GHz
Ka Band	18.0-20.0 GHz

In Fig 23 is shown the data rate of the band listed above positioned according to the range

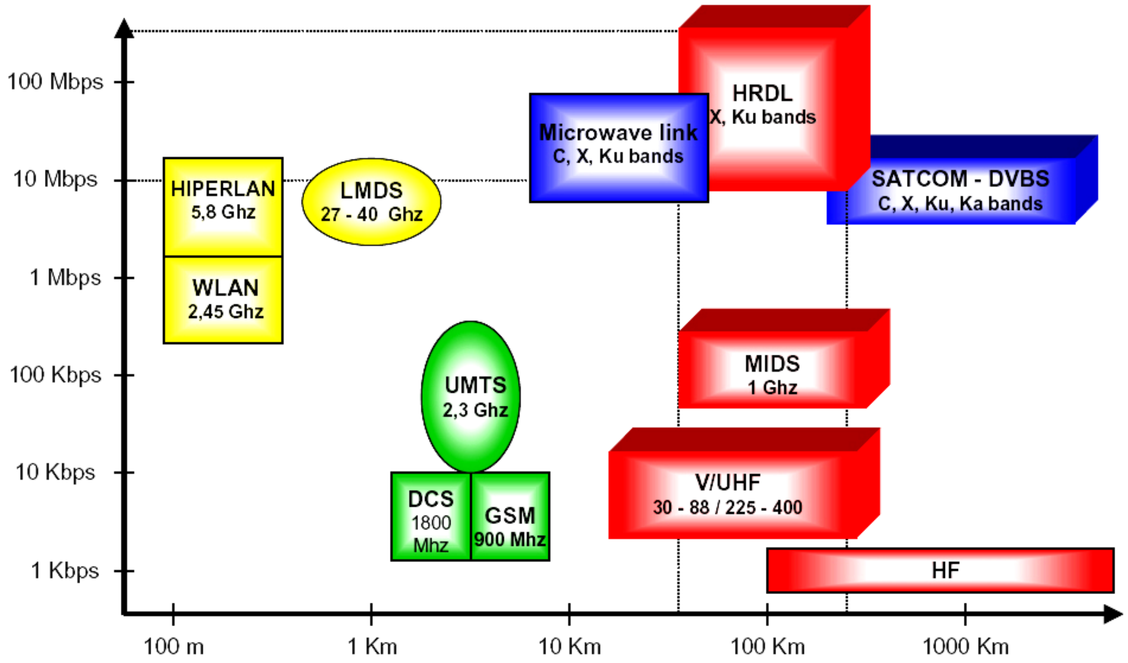


Fig 23 Band data rate according to the range.

4.4 Modulation

As seen before the signal to be broadcast consist in a information amplified and modulated. The modulation consists into impress considerable variation to a radio-frequency wave. Usually a wave called carrier is modulated by varying frequency, amplitude or pulse duration, with a modulating signal that typically contains the information to be transmitted. Two type of modulation are used: analog and digital. The analog modulation is done by an analog modulating signal such a AM and FM radio diffusion and the video diffusion, and the pulse modulation. In the digital modulation instead, the modulator is a sequence of bits as used for the radio telephony of second and third generation and the satellite broadcasting.

4.4.1 Pulse modulation

The first modulation adopted was the pulse modulation that actually is still used in the radar techniques. It consists into the carrier interruption at known intervals. The carrier amplitude rests constant and the modulate signal assumes value null or maximum (equal to the carrier one). It grants high ranges and easy signal understandability, but is a slow communication technique.

4.4.2 Amplitude modulation

It is a technique in which the carrier amplitude by adding the modulator signal containing the information to send. It is used for vocal transmission or high frequency information messages. The carrier $A = A_0 \cos \omega_c t$ is modulated with the modulator signal, for example $1 + m \cos \omega_m t$ where m is the modulation index. The modulated signal results:

$$A = A_0 \cos \omega \;+ \frac{1}{2} m A_0 \left[\cos \omega + \omega \;+ \cos \omega - \omega\right]$$

This is what happens for a pure note, for a more complicated signal, as a vocal message, a Fourier transformation is required.

4.4.3 Frequency modulation

The frequency modulation consist into vary the carrier with the modulator signal frequency. This technique has to keep into account the band limitations that impose a limitation in the frequency maximum deviation: at low frequencies this technique cannot be used. Called Δf the frequency deviation and f_m the modulator frequency, the signal assumes the form:

$$A(t) = A_0 \left[\sin \omega \;\; \cos\left(\frac{\Delta f}{f_m} \sin \omega \right) + \cos \omega \;\; \sin\left(\frac{\Delta f}{f_m} \sin \omega \right) \right]$$

the terms inside the round parenthesis are like Besser series bringing infinite lateral lines at frequencies $f_c + nf_m$ and $f_c - nf_m$ of amplitude decreasing with the order. This means that increasing the carrier frequency the signal increase the band.

This modulation has the advantages of increasing the broadcasting efficiency due to the constant amplitude (the power does not depend from the broadcasted signal). Moreover the transmission is not affected by the weather discharges as them affect only the amplitude.

4.4.4 Digital modulation

The frequency modulation has many advantages but is not efficient in band compared with the amplitude modulation due to the increase of the transmitted band respect to the modulator. the digital modulation solves this problem by converting the signal into a binary sequence before transmitting. It is anyway completely incompatible with the analog modulation. This method is less affected by cannel troubles, higher band efficiency, power efficiency and offers an elevate bit rate. Also in this case the transmitted signal can be modulated in frequency, amplitude or phase.

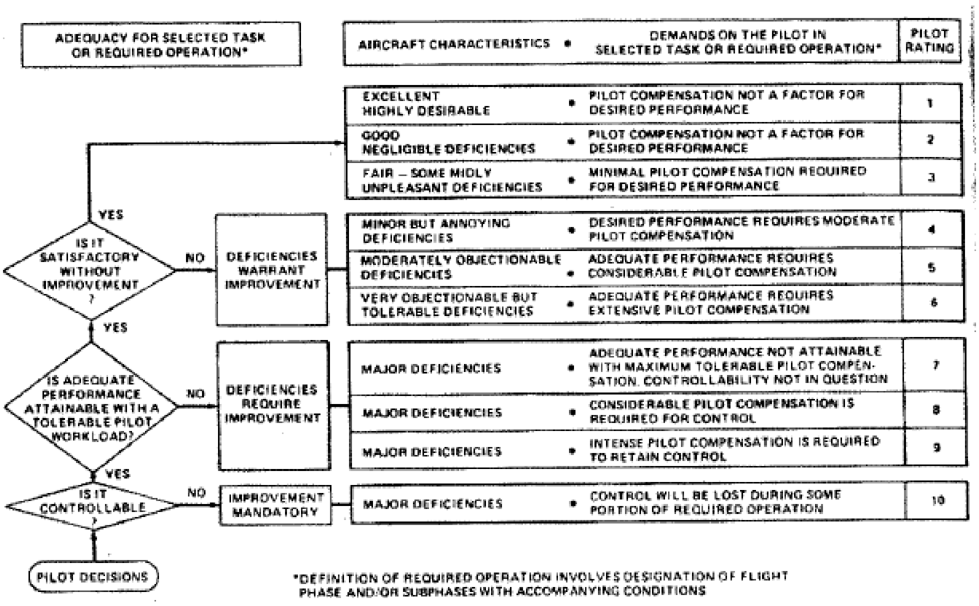
4.4.5 Bit error rate

One of the fundamental aspects to take under control during the digital modulation is the rate of the errors on the bits. This is done with the bit error rate, which is the rate at which the number of errors received over the total bits transmitted. This measure is significant of all the radio system efficiency: electronics, antennas, signal path.

4.5 Latency & handling qualities

The direct control of a UAV is strongly affected by the control loop latency. The increase of latency in a feedback control loop decrease the stability of the system until a certain value of latency would make the system unstable. The overall latency of the system, also called delay, can be defined as the time required by a command to start, execute the action, and provide a feedback. For a UAV, it results as the sum of many efforts: pilot reaction time, CS processing time, uplink latency, UAV command processing time, UAV dynamic, downlink latency, CS display latency. The resulting time delay affect the UAV handling qualities for the direct control.

The effect of time delay in human control have been studied extensively in manned aircraft for handling qualities investigation and the most common used rating scale has been defined by Cooper-Harper Fig 24. [4]



ADEQUACY FOR SELECTED TASK OR REQUIRED OPERATION*
AIRCRAFT CHARACTERISTICS
DEMANDS ON THE PILOT IN SELECTED TASK OR REQUIRED OPERATION*
PILOT RATING
EXCELLENT HIGHLY DESIRABLE
PILOT COMPENSATION NOT A FACTOR FOR DESIRED PERFORMANCE
1
GOOD NEGLIGIBLE DEFICIENCIES
PILOT COMPENSATION NOT A FACTOR FOR DESIRED PERFORMANCE
2
FAIR – SOME MIDLY UNPLEASANT DEFICIENCIES
MINIMAL PILOT COMPENSATION REQUIRED FOR DESIRED PERFORMANCE
3
YES
IS IT SATISFACTORY WITHOUT IMPROVEMENT ?
NO
DEFICIENCIES WARRANT IMPROVEMENT
MINOR BUT ANNOYING DEFICIENCIES
DESIRED PERFORMANCE REQUIRES MODERATE PILOT COMPENSATION
4
MODERATELY OBJECTIONABLE DEFICIENCIES
ADEQUATE PERFORMANCE REQUIRES CONSIDERABLE PILOT COMPENSATION
5
VERY OBJECTIONABLE BUT TOLERABLE DEFICIENCIES
ADEQUATE PERFORMANCE REQUIRES EXTENSIVE PILOT COMPENSATION
6
YES
IS ADEQUATE PERFORMANCE ATTAINABLE WITH A TOLERABLE PILOT WORKLOAD?
NO
DEFICIENCIES REQUIRE IMPROVEMENT
MAJOR DEFICIENCIES
ADEQUATE PERFORMANCE NOT ATTAINABLE WITH MAXIMUM TOLERABLE PILOT COMPENSATION. CONTROLLABILITY NOT IN QUESTION
7
MAJOR DEFICIENCIES
CONSIDERABLE PILOT COMPENSATION IS REQUIRED FOR CONTROL
8
MAJOR DEFICIENCIES
INTENSE PILOT COMPENSATION IS REQUIRED TO RETAIN CONTROL
9
YES
IS IT CONTROLLABLE ?
NO
IMPROVEMENT MANDATORY
MAJOR DEFICIENCIES
CONTROL WILL BE LOST DURING SOME PORTION OF REQUIRED OPERATION
10
PILOT DECISIONS
*DEFINITION OF REQUIRED OPERATION INVOLVES DESIGNATION OF FLIGHT PHASE AND/OR SUBPHASES WITH ACCOMPANYING CONDITIONS

Fig 24 The Cooper-Harper rating scale

This scale has been used for empirical evaluation of aircraft handling performances according to the specific tasks.

The latency value that degrades the controllability is of course dependent from the aircraft characteristics, anyway is intended that high values are not compatible with the manual control. The Cooper-Harper scale is used for manned aircraft, but the same consideration can be done for an UAV directly controlled by the grounded pilot. Respect to a manned aircraft the UAV adds to the system also the data-link latency in uplink and downlink. Considering just this effort to the latency the time delay can be easily calculated. The

signal propagation in the air is close to the light speed (about 300000 km/sec), hypnotizing a line of sight (LOS) of 200 km, the sum of uplink and downlink time result to be about 1.33 msec [11]. So, respect to the others contributes, the LOS signal latency does not greatly affect the overall delay.
Things are different in case of satellite communication for beyond line of sight (BLOS) data-links and will be discussed in the next section.

5 Beyond Line Of Sight (BLOS)

Since the UAV have been used for operative missions, specially for military use, where the operative scenario resulted to be very far from the CS position, the direct link range started to be a limitation for the UAV control. The solution was find by using satellite communication for granting the UAV control also in BLOS cases. The use of satellite communication for UAV control requires some more considerations respect to the direct link case. Over the band , the main problem is given by the time of the signal:

A geostationary satellite has an altitude of about 36000 km, a GS signal travelling at the speed of light would reach the satellite and come back in about 240 msec. In this case the delay due to the data-link signal path is not negligible. This means that the UAV control in full manual mode is in many case more difficult if not completely unsafe, particularly during take-off and landing phases. Considering a LEO satellite the distance would be inferior, anyway other problems have to be considered over the signal path distance. A study about the use of commercial satellite constellations for the Flight Termination System (FTS) BLOS command [12] collected data about the satellite communication latency due to buffering. Tests made on IRIDIUM satellite network show the probability of receiving a the commanded signal in a certain amount of time Fig 25.

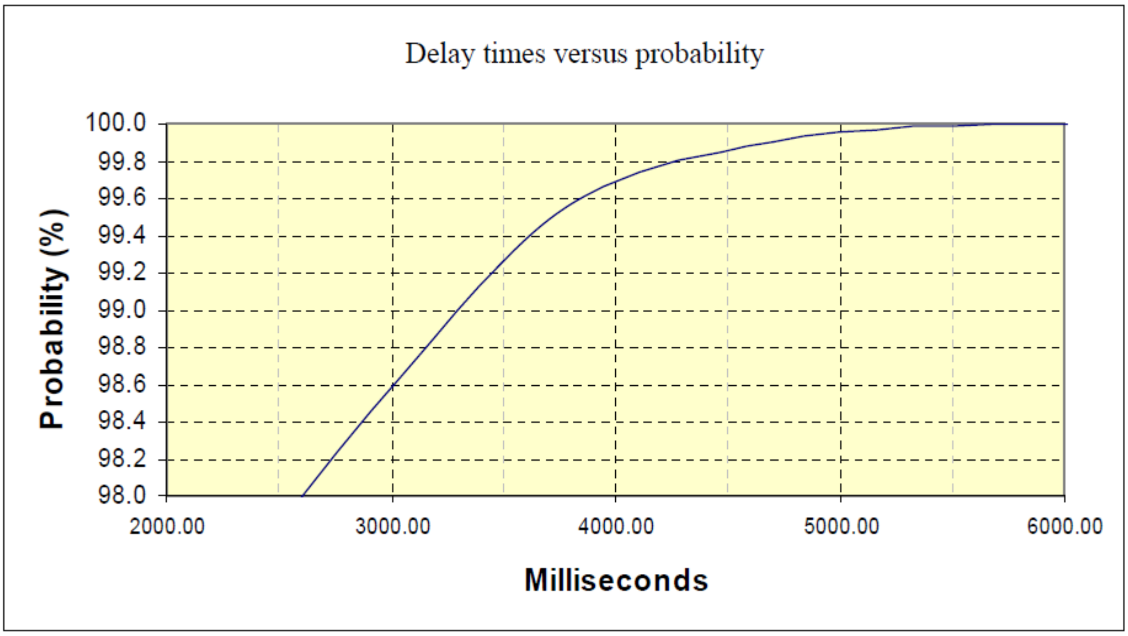


Fig 25 Probability of receiving the signal in a certain amount of time with IRIDIUM.

According to this study there is a probability of 96.7% to have a signal packet delay of 4 seconds due to Acquisition-No acquisition process needed to guarantee data to delivery. That is not acceptable at all for manual controllability. According to IRIDIUM this process of guarantee data delivery can be disabled for a data call, anyway a buffering delay of 600 msec is always present.

The delay problem is one reason for going in the direction of increasing the UAVs level of Automation (LOA) in order to reduce the need of pilot direct control at least for the functions affected by controllability delay degradation.

In addition the LOA increase gives also the advantage of reducing the pilot workload and brings the possibility to manage multiple UAVs at the same time.

Another problem is the need of switch the UAV control from one CS to other CSs. An example is the case of a mission that requires take-off in manual mode from a LOS CS, hand-over to a BLOS CS to reach the operative scenario, and hand-over to a CS displaced near the operation area in LOS mode for granting manual control during the mission operations or for landing. Because of this the hand-over process is an aspect that need to be considered.

6 Level Of Automation (LOA)

The way for not be affected from the latency problems is augment the platform level of automation and left to the pilot just a role of supervisor monitoring the UAV and interacting for taking decisions without controlling directly the vehicle in manual mode.

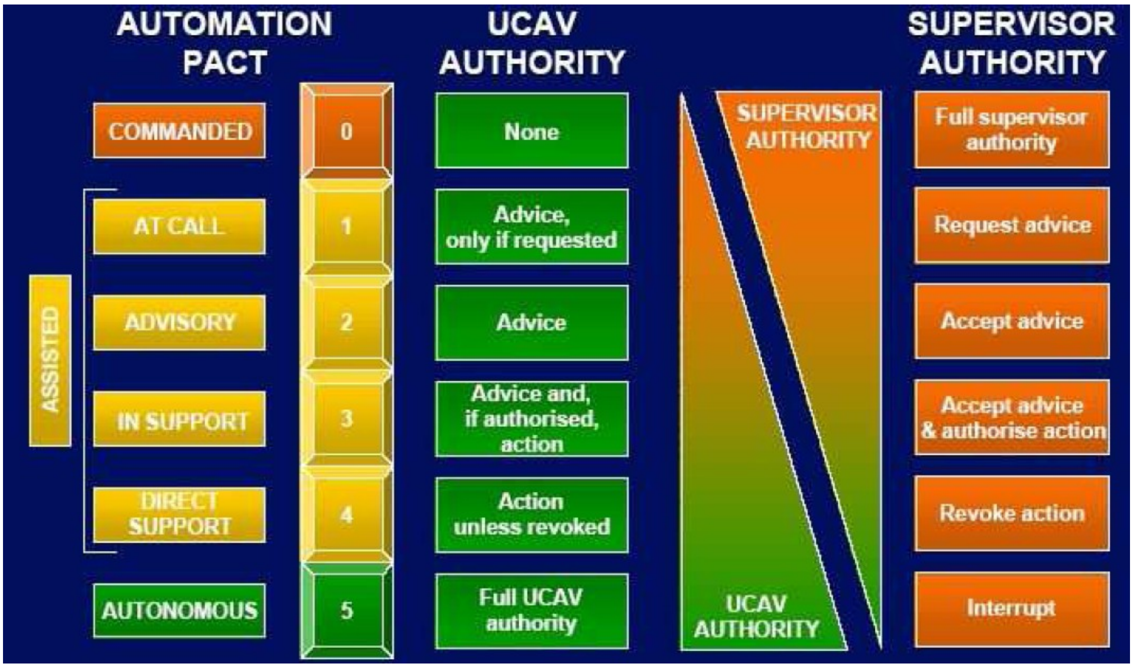


Fig 26 PACT framework.

Different level of automation can be individuated and the proposed scales are many, one of the possible is the PACT scale (Pilot Authorization and Control Tasks) [13] in which a rate from 0 to 5 is assigned for marking the level of automation from no automation (level 0) to full autonomous (level 5). Intermediate levels represent the different demand of interaction asked to the pilot, the possibility to ask for an advice to the system (level 1), the receipt of a system advice (level 2), the advise acceptance and action authorization (level 3), the possibility to revoke an autonomous action (level 4). The increase of UAV automation overcomes the problem of data-link latency and changes the pilot role from an active role to a supervision role and his attention can be concentrated to the task instead of the piloting.

Another scale developed by the US Air Force Research Laboratory is the ACL[18]. That study specify the difference between the automation and the autonomy defining the automation as the capacity to execute programmed functions, and the autonomy as the capability to choice a solution.

7 Hand-Over

As seen in section 5 the hand-over between two CSs is an issue to consider for a BLOS UAS. An hand-over occurs any time the command and control should pass from a CS to another, and this may happens mainly for reasons of range. Not to limit the range of operations to the LOS of the commanding CS, the control can pass to another CS always in LOS, adjacent to the range of the previous. This solution can be used for short range missions if the satellite communication is not implemented. Anyway the use of satellite communication for BLOS operations is the best solution for increasing the range, so the hand-over in that case occurs for passing the control from a LOS CS to a BLOS CS and vice versa, but also for doing the same thing from the same CS and just switching the data link system. The reasons for not using since the begin a BLOS data link are simply the need of reducing the delay for the more critical phases where a direct control is or always required or, in case of more automated systems, provided for back up. Those phases are usually the take off, the landing and the active part of the mission. A typical BLOS mission can start with a manual take off in LOS, then a BLOS control brings the UAV to the area of operation where a local CS takes the control in LOS for granting the direct control capability in case of needs. During the hand-over the control should pass from the first CS, currently in control, to the second one receiving the control. During this procedure one of the two CS should always have the control of the UAV, and no overlapping is permitted to avoid conflicts of command. Both the CS should have the UAV data in downlink for monitoring the platform.

8 STANAG 4586 for interoperability

Up to now the UASs are typically formed by a UAV and a CS designed to interact each other as a closed system in which the CS speaks with its own UAV but only with it and vice versa. Type of data-link, communication protocol, message format are proper of that system and each UAS has a different protocol. That is a problem for coordinated operations in which more UAS of different kind are used by the same operator. NATO proposed a solution for this problem in order to have interoperable systems for military operations, anyway that solution can be adopted also for civil UAVs. For encouraging interoperability NATO promulgated a standardization, the STANAG 4586, proposing common references for interfaces, communication protocol and transmitted frames [14].

8.1 STANAG 4586 architecture

The STANAG 4586 describe the system architecture as a group of five elements as shown in Fig 27.

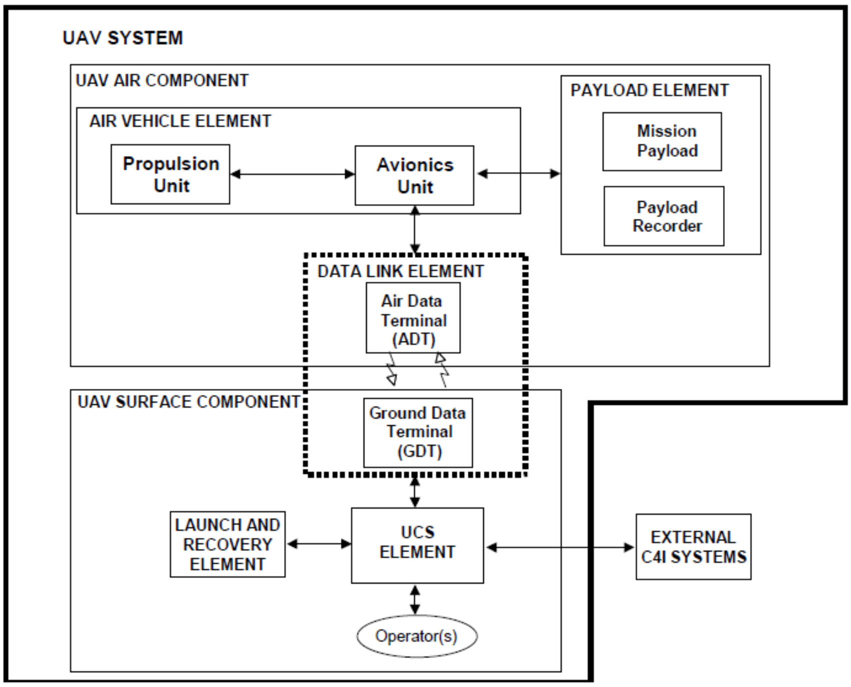


Fig 27 STANAG 4586 UAV system elements [14]..

The air vehicle element which consist of the airframe, the propulsion and the avionic unit for the flight management. The payload element consisting of sensors and recording systems, or weapons system and control/activation mechanism, or both. The data-link element consisting of two units, one air data terminal and one ground data terminal (that can be located also on another air platform). It is described as one unique link but can be formed by separate independent data-links. The UCS element which is the ground element (but can be located also on another air platform) generating, loading and executing the UAV mission and exchanging information with the various C4I. The launch and recovery element that incorporate the functionalities of launch for taking off and recovery for landing. This system present advantages respect to the traditional way of managing a interoperable system. The current way of having an interoperable system is to have a so called ‘stove pipe’ system Fig 28 in which each UAV has an individual communication system, protocol and messages format with its own UCS, and each UCS exchanges information with a different C4I, then all the data are provided to a common tactical node.

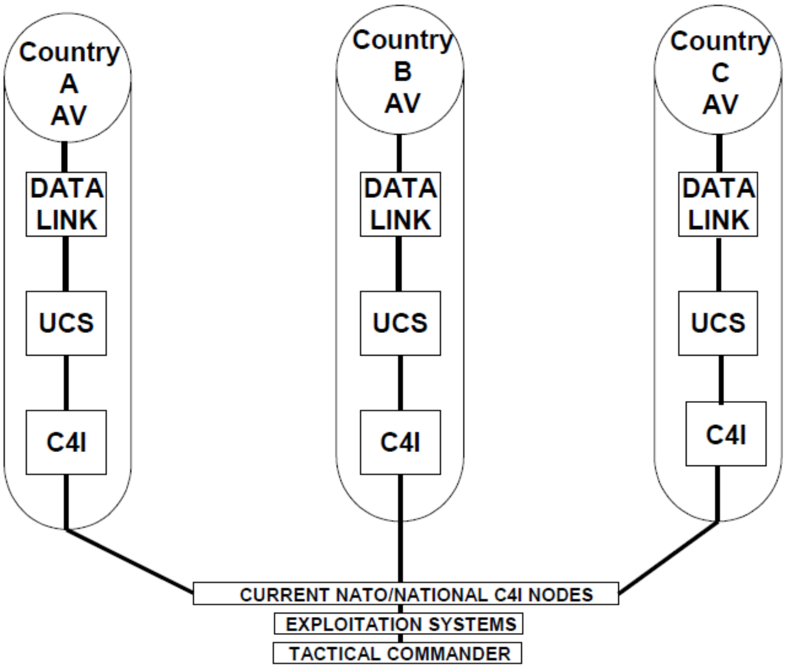


Fig 28 Current UAV system operations example [14].

The STANAG 4586 architecture instead prescribe a configuration for granting a unique data-link and UCS and common communication protocol between the UAVs and the UCS and the UACS and the C4I nodes. This would ensure a near real-time interaction between the two side of the system, as desirable by an interoperable mission, which a classical 'stove pipe' cannot provide.

8.2 Level of interoperability

The STANAG 4586 describe five levels of interoperability (LOI) each one representing a different UAS capability:

LOI 1: indirect receipt of UAV related data

LOI 2: direct receipt of ISR/other data

LOI 3: control and monitor of the UAV payload in addition to the LOI 2 capabilities

LOI 4: control and monitoring of the UAV except launch and recovery

LOI 5: LOI 4 capabilities with the inclusion of launch and recovery

If the UAV has more than one payload the LOI 2 and LOI 3 can be different for each payload.

These interoperability levels are obtainable by the standardization of the UAS components interfaces and the standardization of the interfaces between the UCS and the C4I systems. The STANAG 4586 provide such standardizations and for enabling the desired LOI the UAS shall be compliant with existing or new standards for:

- A data link system(s) that provides connectivity and interoperability between the UCS and the AV(s). The data link system(s) shall accommodate legacy as well as future systems. STANAG 7085, Interoperable Data Links for Imaging Systems, specifies a data link system that would provide the required connectivity and interoperability. Users that require encryption should reference work being done for data links by NAFAG Air Group IV and NATO International Military Staff (IMS) for interoperable encryption standards. A standard for a secondary or "back up" data link for UAV systems requiring one, or for use in tactical UAS not requiring the capability of a STANAG 7085 Data Link is not currently available and needs to be developed.
- Format for payload/sensor data for transmission to the UCS via the data link and/or for recording on the on-board recording device. STANAG 7023, Air Reconnaissance Primary Imagery Data Standard, with addition for non-imagery sensors, (e.g., Electronic Support Measures (ESM)), STANAG 4545, NATO Secondary Imagery Format, STANAG 4607, NATO GMTI Format, and STANAG 4609, NATO Digital Motion Imagery Format provide standard formats for transmitting payload data to the UCS or for storage on the on-board recording device.
- Recording device for on-board recording of sensor data, if required, STANAG 7024, Imagery Air Reconnaissance Tape Recorder Standard, and STANAG 4575, NATO Advanced Data Storage Interface (NADSI), specify standard recording devices and formats for wideband tape and other advanced media (e.g. solid state, RAID) recorders, respectively.
- UCS interfaces with the data link system (e.g., DLI); UCS interface with command and control systems (e.g., CCI); and HCI top level requirements for a UCS to support the UAV System operators. STANAG 4586 defines the UCS Architecture and interface requirements.
- Although beyond the scope of this STANAG, operational guidelines or standards that define the minimum level of operator proficiency needed to operate a given UAV at the desired LOI are also required.

8.3 Functional architecture

The functional architecture of the UCS is shown in Fig 29 where the system is composed by:

- Core UCS (CUCS)
- Data Link Interface (DLI)
- Command and Control Interface (CCI)
- Vehicle Specific Module (VSM)
- Command and Control Interface Specific Module (CCISM)

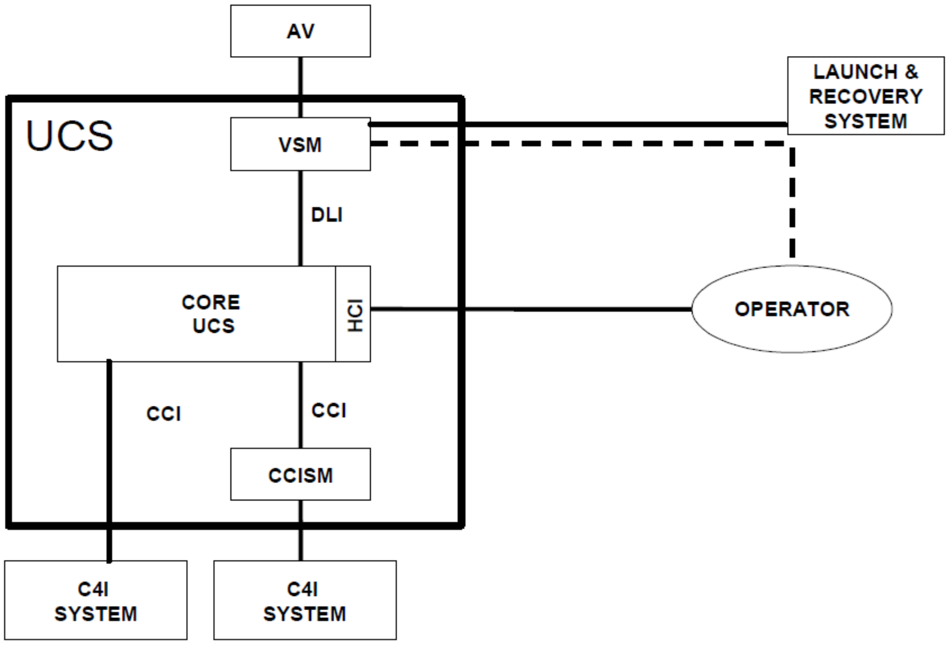


Fig 29 UCS functional architecture [14].

The VSM function is to provides unique/proprietary communication protocols, interface timing, and data format required by the respective air vehicles; and also provide a DLI protocol and communication format 'translation' to the format required by the platform. The VSM can be on board, on ground, or both. It has also to provide the UCS DLI connection to the GDT associated to non STANAG 7085 compliant data-link, as for example satellite communications for BLOS operations, if such a data-link is present. This module, as proper of a specific air vehicle, is usually provided by the air vehicle manufacturer.

The CCISM provide a similar function of the VSM between the UCS and the C4I.

8.4 Data link interface

The DLI is the interface between the UAV data-link and the CUCS. For establishing the DLI message set a wide range of UAVs system requirements have been considered. The DLI role in the AV/UCS concept is shown in Fig 30 where four vehicles are described. The firs is an UAV that totally support the DLI interface, so does not need a VSM. The other three possible configurations are UAVs that partially support the DLI messages and so require a VSM for the remaining part of the DLI interface functionalities. It is also possible to have an UAV don't supporting any DLI messages demanding all the DLI functionalities to the VSM, but it represents the extreme case. The second case uses a VSM on ground between the CUCS and the GDT, the third case brings the VSM on board between the ADT and the GDT, and the last case presents one VSM on board and another on ground.

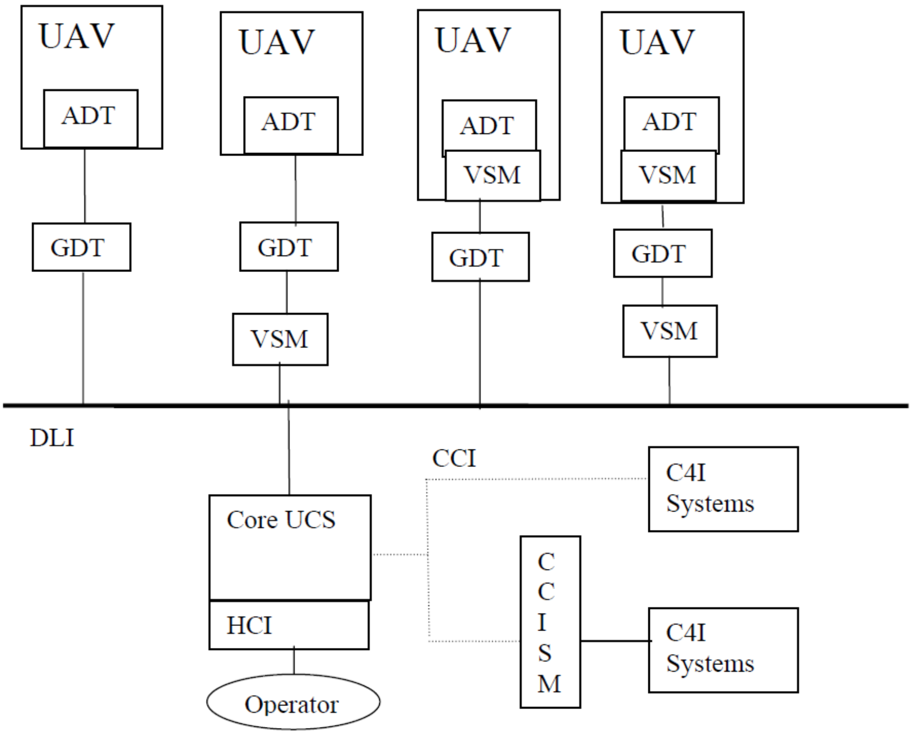


Fig 30 DLI role in the AV/UCS concept [14].

8.5 UCS communication and information technology protocol and standards

The UAV is seen as a terminal element of a network in which the UAV and C4I system should be able to interoperate. For reaching this a standard protocol for the electronic exchange of information has been selected: the NATO Command, Control, Communication (NC3) Technical Architecture (TA), Volume 4, NC3 Standard Profile (NCSP). This document divided the standards into different areas:

- User Interface
- Data Management
- Data Interchange
- Graphics
- Communications
- Operating Systems
- Internationalization
- System Management
- Security
- Distributed Computing
- Software Engineering

To achieve interoperability a minimum of standards about interchange and communications protocols should be implemented:

For interchange service geographical standards should be granted by using Digital Geographic Information Exchange Standard (DIGEST Version 1.2a), STANAG 7074:1998; Digital Terrain Elevation Data (DTED) Geographic Information Exchange Standard, STANAG 3809; Digital Feature Analysis Data (DFAD); and World Geodetic System - 84 (WGS-84), Mil-STD-2401.

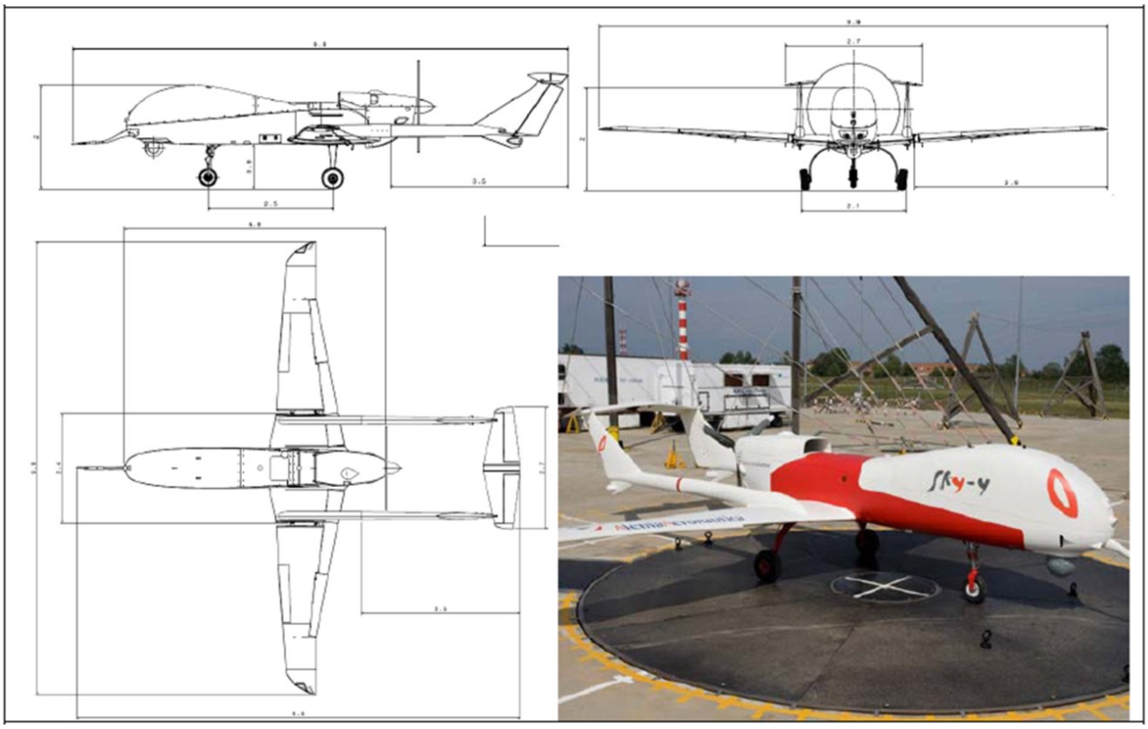
For communication service the architecture should be compliant to the IP version adopted by the community in which the UCS is integrated, that in near-term is the IPv4. In future will be replaced by the IP version IPv6 to overcame IPv4 weakness, increase available address space, and improve security, throughput, latency, error rate and costs.

It should adopt also the Transport Control Protocol (TCP), the User Datagram Protocol (UDP), the Hypertext Transfer Protocol (HTTP), the File Transfer Protocol (FTP), and the Network Time Protocol (NTP).

9 Sky-Y experimentation purposes

From the previous analysis, for a BLOS UAV management, the increase of the on-board automation and the hand-over capability have to be investigated. Considering the advantages offered by the STANAG 4586 adoption, the new system should be improved by the STANAG 4586 compliance. This would increase the interoperability between UAV and CSs, so in vision of a future BLOS management, the hand-over capability and the multiple platform management would result easier. Moreover the STANAG 4586 adoption prescribe the use of contingency routes for the lost link event management, that represents an improvement in safety and an increase of the automation level.

The Alenia Aermacchi technological demonstrator MALE UAV: the Sky-Y (Fig 31), has been selected as platform suitable this experimentation.



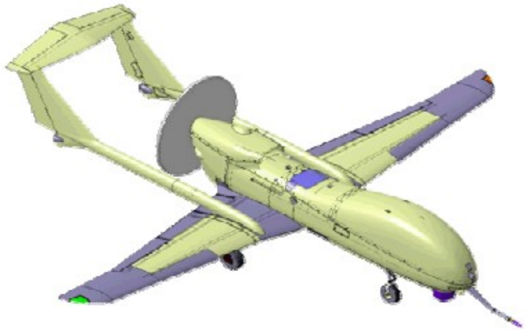


Fig 31 The Sky-Y MALE UAV.

Starting from a LOS remotely controlled UAV, the first step for reaching the BLOS management has been the improvement of the on-board navigation functions to increase the platform automations. This imposed the design and development of a new navigation and steering functions. At the same time, the new design has been done taking into account the STANAG 4586 prescription. By imposing the transmission of specific messages format and content in fact, the navigation functions design resulted to be impacted by the STANAG 4586.

For the development of the software algorithms concerning the navigation and steering, a large use of the simulator have been done. The approach used consisted into generate the final low level software requirements by implementing high level requirements into a simulation model. Three steps have been affronted: the modeling of a Matlab® simulator for the steering functions development; a Simulink® model implementation for the functions evaluation; the real time Sky-Y flight simulator tests for final code validation and last developing phase. Then the functions have been integrated on the real hardware component and tested, first stand-alone and then integrated with the other systems.

The validation of the new on-board navigation functions brought the UAV to operate in 4D autopilot mode with pilot supervision. These functionalities, allowing the BLOS control capability, have been finally implemented on the aircraft for flight tests.

At the beginning, before the experimentation, the Sky-Y was equipped with a traditional command and control system, a LOS data-link, and a navigation computer providing both navigation and surfaces commands. During the flight test experimentation the new system STANAG 4586 compliant has been placed beside the original one. The old system has not been removed in order to have at any moment the possibility of switching the command and control to the old consolidate control line considerate safe. As will be discussed in detail later the switch from one control line to the other is an hand-over that offers many similitude to the LOS to BLOS case and vice-versa, for that reason has been studied as a significant test case.

The last study consisted into new advanced functionalities investigation to increase again the LOA in vision of a future BLOS capability improvement.

9.1 Sky-Y system architecture

The overall architecture (including the consolidated control line) is described in Fig 32. The ground segment is composed by a Tactical control Station (TCS) STANAG 4586 compliant and the Data Link modules (D/L). The TCS is split in two, the Remote Operator Station (ROS) Ground Control Station (GCS) communicating with a dedicate D/L, which is the traditional command and control line considered safe, and the new STANAG 4586 compliant ROS TCS communicating by two separate D/L: a Wide Band Data Link (WBDL) and a Narrow Band Data Link (NBDL).

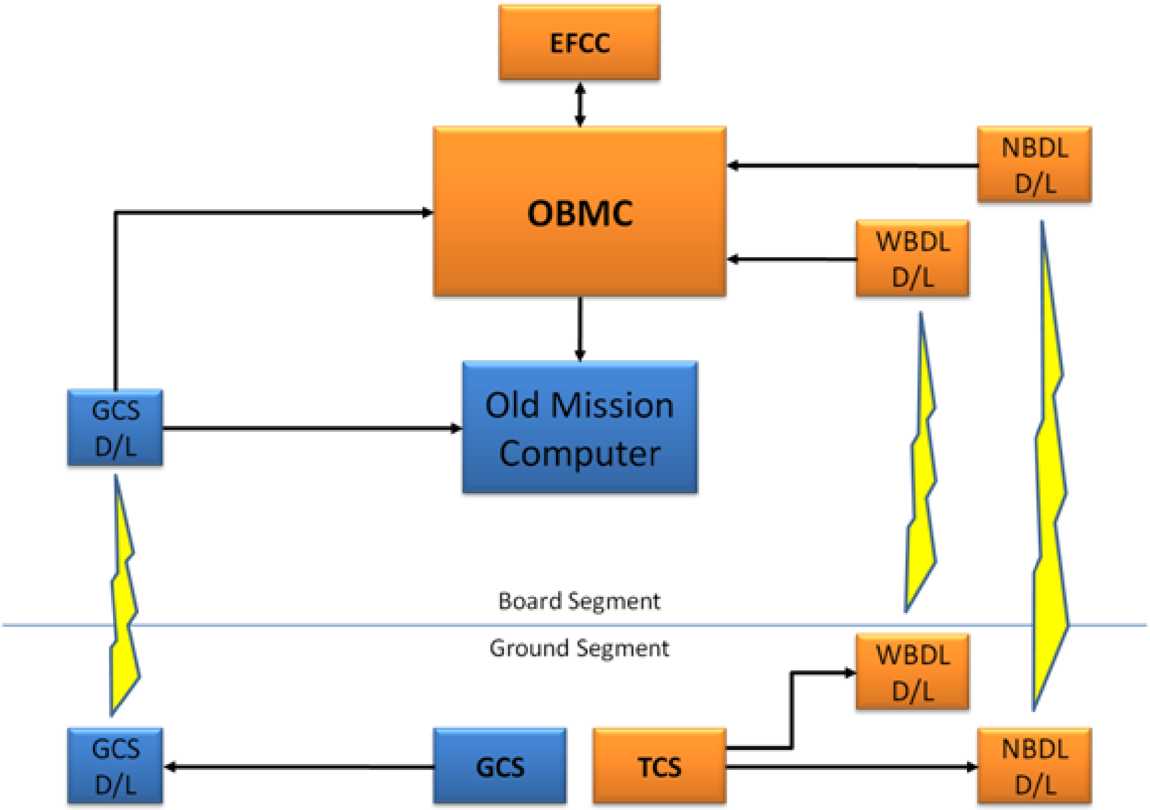


Fig 32 Sky-Y first step system architecture.

The board segment is composed by an On Board Mission Computer (OBMC) managing all the data, the WBDL and NBDL module for the TCS communication, the GCS D/L module for the GCS communication, an Experimental Flight Control Computer (EFCC), and the old navigation and flight control computer.

The purpose of this configuration is to test the new command line STANAG 4586 compliant. The previous configuration was using the GCS communicating with the old mission computer which provided both the navigation calculation and the control law elaboration for the actuator deflection. That system didn't support the STANAG 4586. The new system is composed by a OBMC managing the D/L, the sensors, the navigation calculation, with STANAG 4586 interfaces and communication protocol. Then a separate element, the EFCC, is demanded to elaborate the control law for the actuator deflection. The first step configuration is composed by both the systems for safety reasons and demand in addition the OBMC to support the hand-over process to switch the command from the old line to the new one and vice versa. Anyway, in case of failure, the GCS can at any time take the command bypassing the OBMC. The exact procedure of hand-over will be discussed in detail later in a dedicated section (9.5).

9.2 The OBMC

The scope of the OBMC is to practice functionalities generally performed by pilot on ground. These scopes involve mainly the communication among OBMC and the following sub-system:

- FMS: it includes old mission computer, actuators and sensors; it controls the flight, by means of an internal autopilot and provides data coming from on board equipment (GPS, Radar Altimeter and other integrated sensors).
- MMS: it includes Mission Manager Equipment, also named OBMC; it controls the Sensor System (SNS) and EFCC in order to manage its functionality using received commands from CS.
- COMS: The acronym stands for Communications System and includes a Command & Control Data Link (GCS DL), Wide Band Data Link (WBDL) and Narrow Band Data Link (NBDL).
- SNS: The Sensor System it represented by an Electro Optical Sensor which is connected to OBMC and controlled by TCS for manual operation or by OBMC for automatic mode.
- EFCC: Experimental Flight Control Computer. When engaged it send surface commands (aileron, rudder, elevator and throttle) to the actuators.
- DGPS and IRS: provides accurate information about aircraft position, angles, velocities etc.
- Video Encoder: acquire sync information from OBMC and provides video stream from SNS.

The OBMC main tasks are:

- To allow piloting the aircraft by TCS using navigation functions present on OBMC and sensors information coming from EFCC and DGPS/IRS.
- To Monitor health status of connected equipments and OBMC itself.
- To control communications between the Electro Optical Sensor and the Tactical Control Station and to add automatic procedures to control autonomously the Electro Optical Sensor (Automatics Mode).

9.3 Navigation and Steering

The autonomous navigation of the UAV is provided by the two OBMC modules: NAV and Steering. The NAV module is composed by 12 functional blocks (Fig 33) providing the following functions:

- The health monitoring
- The data exchange with EFCS
- The Auto Pilot (A/P) Mode determination
- The Best Data calculation
- The STANAG 4586 interface and data conversion
- The navigation sensors management
- The Best navigation mode determination

- The ATOL data calculation
- The navigation data transmission with WBDL
- The Lost Link state calculation
- The Phase of Flight determination
- The FOM evaluation

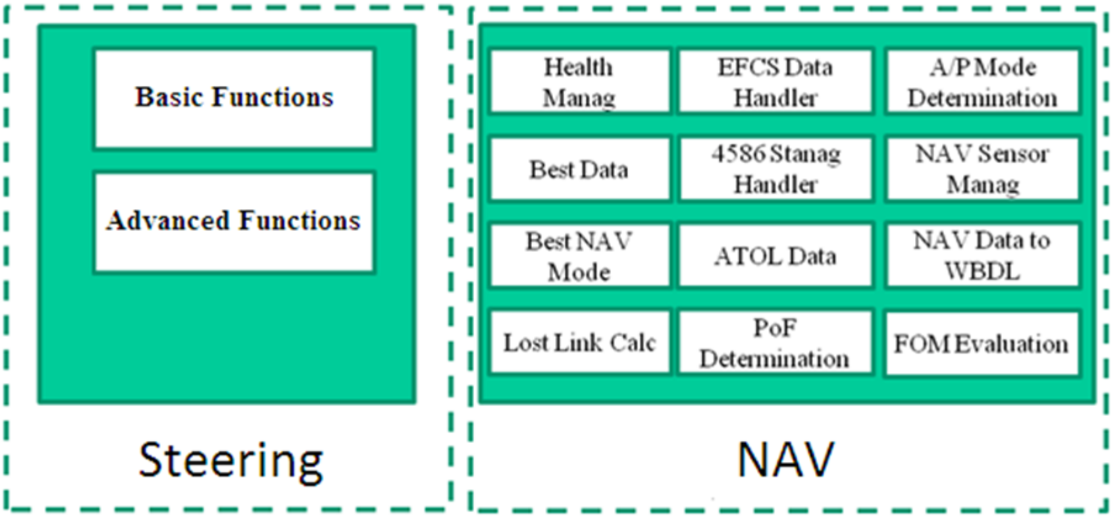


Fig 33 Steering and NAV module overview.

The Steering module instead is asked to manage the waypoint and elaborate the route flight calculations. A simplified scheme of the process for providing the autonomous navigation is exposed in Fig 34.

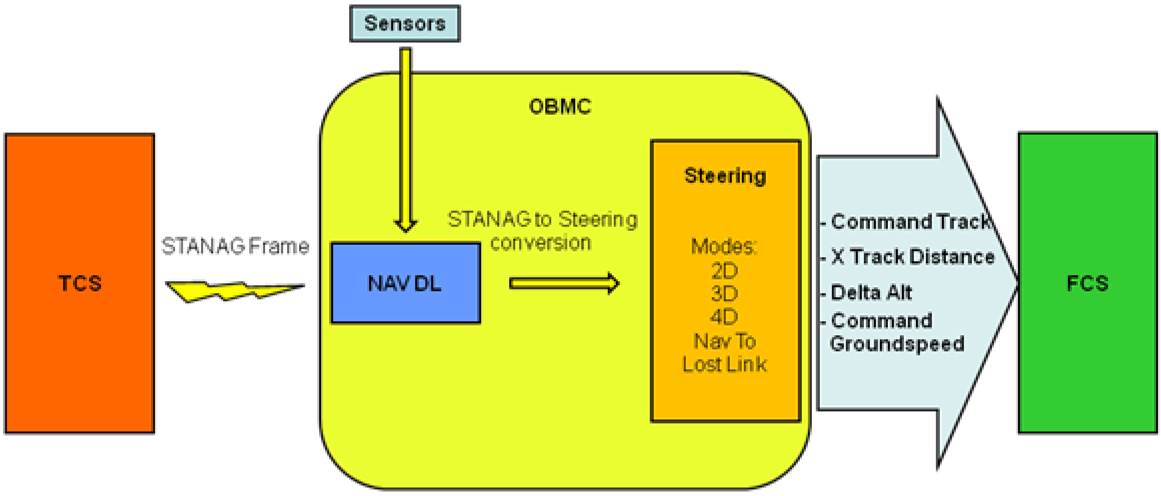


Fig 34 Navigation data flow concept.

The TCS send and receive navigation data communicating with the NAV following the STANAG 4586 protocol, the NAV operate the data conversion from the STANAG 4586 protocol to the Steering format, collect the sensors data and send all the information to the Steering partition. The Steering module operate the calculations for the required navigation mode and send the proper correction to the FCS (always passing through the NAV again).

9.3.1 NAV functions

In this section will be described the 12 NAV functional blocks of Fig 33.

9.3.1.1 A/P mode determination

The possible autopilot modes considered for the Sky-Y are:

Basic modes

- Pitch & bank
- Altitude/heading
- Vertical speed/heading

Steering modes

- Nav2D: waypoint route navigation with lat & lon from WP attribute, altitude and speed imposed by the pilot
- Nav3D: waypoint route navigation with lat & lon and altitude from WP attribute, speed imposed by the pilot
- Nav4D: waypoint route navigation with all the data from WP attribute

Advanced Steering modes

- Nav-To: navigation to a loiter WP with altitude, speed and loiter parameters imposed by the pilot; lat & lon imposed by the pilot or choose from database
- Slave To Sensor: navigation to a circular loiter point linked to the sensor footprint position; altitude, speed, loiter radius, direction and offset imposed by the pilot
- Lost Link: navigation to a safety loiter point following a contingency route, all the data taken from WP attribute

The A/P mode is selected from the TCS apart from the Lost Link mode which is automatically selected from the OBMC in case of degradation of the signal. The selection of a mode transit from the TCS to the NAV in STANAG 4586 format as a combination of frame values, the NAV is asked to interpret the frames and figure out the correct value of navigation mode for the Steering partition. The STANAG 4586 messages used for the A/P mode determination are:

- Message #42 - Vehicle Operating Mode Command, Field #4 - Select Flight Path Control mode, possible values 2 = Flight Director, 11 = Waypoint, 12 = Loiter, 22 = Slave To Sensor, 32 = Pitch/Bank.
- Message #43 - Vehicle Steering Command, Field #4 - Altitude Command Type, possible values 1 = Altitude, 2 = Vertical Speed. Field #7 - Heading Command Type, possible values 1 = Heading.

- Message #48 - Mode Preference Command, Field #4 - Altitude Mode, possible values 0 = Configuration, 2 = Manual/Override. Field #5 - Speed Mode, possible values 0 = Configuration, 2 = Manual/Override.

The combination for providing the desired A/P mode is determinate by the following table (Fig 35):

4586 input					OBMC data
Select Flight Path Control Mode (msg42)	**Heading Command Type(msg43)**	**Altitude Command Type (msg43)**	**Altitude Mode (msg48)**	**Speed Mode (msg48)**	**AUTOPILOT-MODE**
32 = Pitch/Bank	n/A	n/A	n/A	n/A	**Pitch/Bank**
2 = Flight Director	1- HDG	1-ALT	n/A	n/A	**Altitude / HDG**
2 = Flight Director	1- HDG	2-VS	n/A	n/A	**Vertical Speed/HDG**
11 = Waypoint	n/A	n/A	2-MAN	2-MAN	**NAV2D**
11 = Waypoint	n/A	1-ALT	0-CONF	2-MAN	**NAV3D**
11 = Waypoint	n/A	1-ALT	0-CONF	0-CONF	**NAV4D**
12 = Loiter	n/A	n/A	n/A	n/A	**Nav To**
22 = Slave To Sensor	n/A	n/A	n/A	n/A	**Slave To Sensor**

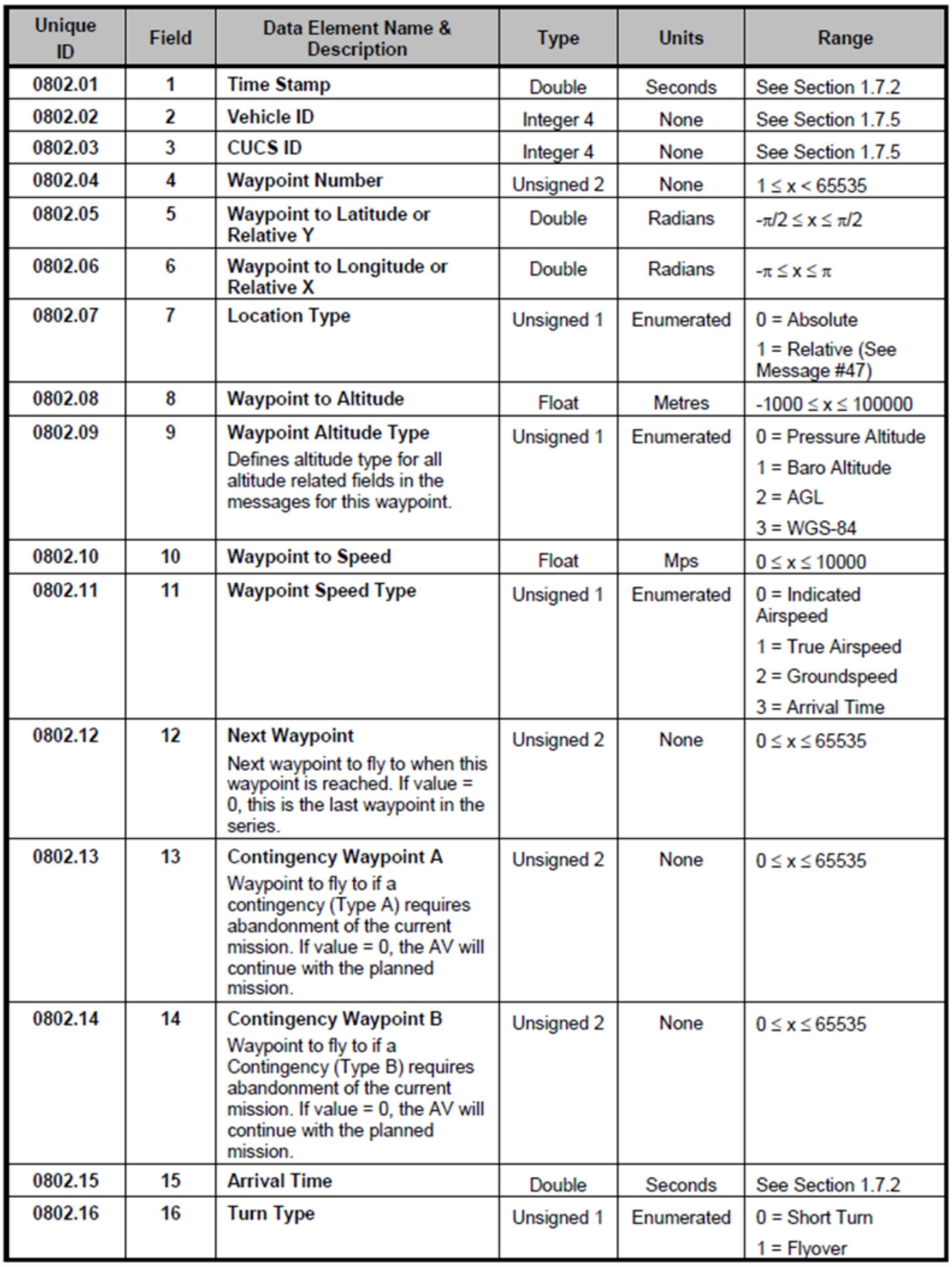
Fig 35 A/P mode determination table.

9.3.1.2 STANAG 4586 conversion – The route loading

The OBMC NAV module STANAG4586 conversion function provide all the conversion between the Steering and the GS or FCS messages. The Steering utilizes the STANAG 4586 data but in a different format specific to that partition. The function in question is asked to elaborate this conversion and the most important re-assembling data regard the WPs and route loading.

The Route Loading

The STANAG 4586 route loading is made with the messages series #800. The #800 is used for the upload demand and brings the information of the number of WPs that will be transmitted, the #801 is used for specify the route type and tells the OBMC the ID number of the first WP of the route. Then a sequence of #802 is sent, each message bringing the information data of a single WP of the route. Moreover, for each loiter WP, an additional #803 message is sent bringing the loiter data. Looking the #802 frame Fig Fig 36 can be noted that, further the data relative to the WP, as WP Number, Lat, Lon, Alt, Speed, the frame marks also the *Next Waypoint* as the ID number of the following WP (Field 12). This is used by the NAV to generate the route independently from the messages sequence.



Unique ID	Field	Data Element Name & Description	Type	Units	Range
0802.01	1	**Time Stamp**	Double	Seconds	See Section 1.7.2
0802.02	2	**Vehicle ID**	Integer 4	None	See Section 1.7.5
0802.03	3	**CUCS ID**	Integer 4	None	See Section 1.7.5
0802.04	4	**Waypoint Number**	Unsigned 2	None	1 ≤ x < 65535
0802.05	5	**Waypoint to Latitude or Relative Y**	Double	Radians	$-\pi/2 \le x \le \pi/2$
0802.06	6	**Waypoint to Longitude or Relative X**	Double	Radians	$-\pi \le x \le \pi$
0802.07	7	**Location Type**	Unsigned 1	Enumerated	0 = Absolute 1 = Relative (See Message #47)
0802.08	8	**Waypoint to Altitude**	Float	Metres	-1000 ≤ x ≤ 100000
0802.09	9	**Waypoint Altitude Type** Defines altitude type for all altitude related fields in the messages for this waypoint.	Unsigned 1	Enumerated	0 = Pressure Altitude 1 = Baro Altitude 2 = AGL 3 = WGS-84
0802.10	10	**Waypoint to Speed**	Float	Mps	0 ≤ x ≤ 10000
0802.11	11	**Waypoint Speed Type**	Unsigned 1	Enumerated	0 = Indicated Airspeed 1 = True Airspeed 2 = Groundspeed 3 = Arrival Time
0802.12	12	**Next Waypoint** Next waypoint to fly to when this waypoint is reached. If value = 0, this is the last waypoint in the series.	Unsigned 2	None	0 ≤ x ≤ 65535
0802.13	13	**Contingency Waypoint A** Waypoint to fly to if a contingency (Type A) requires abandonment of the current mission. If value = 0, the AV will continue with the planned mission.	Unsigned 2	None	0 ≤ x ≤ 65535
0802.14	14	**Contingency Waypoint B** Waypoint to fly to if a Contingency (Type B) requires abandonment of the current mission. If value = 0, the AV will continue with the planned mission.	Unsigned 2	None	0 ≤ x ≤ 65535
0802.15	15	**Arrival Time**	Double	Seconds	See Section 1.7.2
0802.16	16	**Turn Type**	Unsigned 1	Enumerated	0 = Short Turn 1 = Flyover

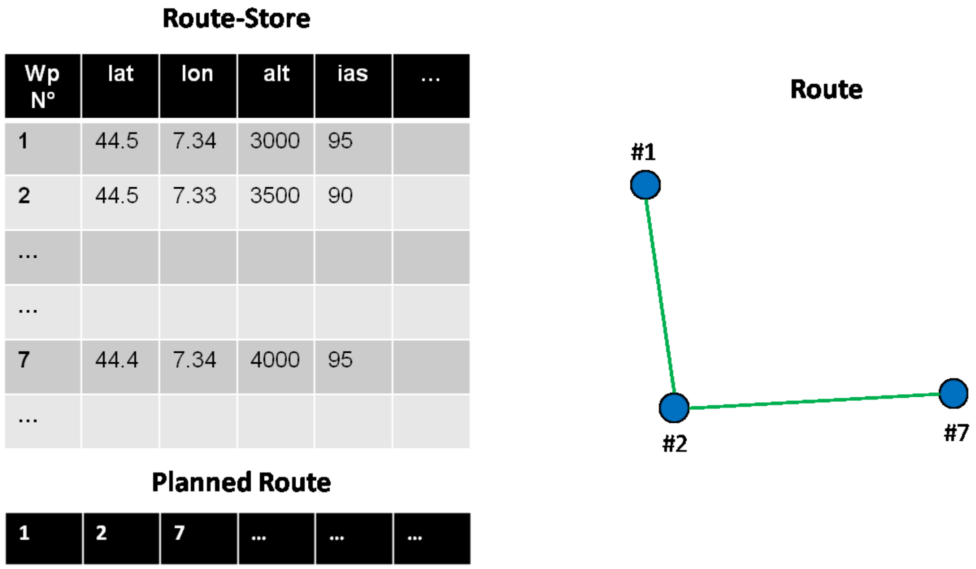
Fig 36 STANAG 4586 extract: message #802 - AV Position Waypoint [14].

In this way the NAV allocates each message #802 data into a structure like Fig 37 called *New WP Data* and sends this data to the Steering partition in which a specific function, called *New WP Data Action,* copies the data into a free field of the *Route-Store*: a vector of structures able to store up to 500 WPs, 250 for the main route and 250 for the *Contingency Routes* (see section 9.3.3).

New WP Data	.wp_number
	.type
	.lat
	.lon
	.alt
	.groundspeed
	.sched_time_to
	.loiter_radius
	.loiter_direction
	.loiter_speedloiter_time
	.loiter_time
	.loiter_type
	.loiter_bearing
	.loiter_length

Fig 37 New WP Data structure.

Knowing the *Waypoint Number* from the message #801 and the *Initial WP Number* from the #800 the OBMC can understand when the route is finished and no more messages #802 have to be received. After this the NAV edit the *Planned-Route*, a sequence of *WP* N*umber* forming the main route, by using the *Next Waypoint* information of each message #802.



Route-Store

Wp N°	lat	lon	alt	ias	…
1	44.5	7.34	3000	95	
2	44.5	7.33	3500	90	
…					
…					
7	44.4	7.34	4000	95	
…					

Planned Route

1	2	7	…	…	…

Fig 38 Example of Route-Store and Planned-Route for the route generation.

In case of loiter WPs the STANAG 4586 provides a message #803 for each loiter WP, Also these messages are not dependent from the receiving sequence thanks to the field #4 *Waypoint Number* which links the loiter data to the respective WP.

Another conversion operated by the NAV is the interpretation of the messages #43 which alone can be used for the selection of a WP of the route to impose as destination (Skip WP function), and together with the message #41 is used for defining the Nav-To data and select that mode. The use of the message #43 for selecting the destination WP is always used at any route engage independently from the message #800 field *Initial WP Number*. It is important to note that these two values have different meanings: in the message #800 is prescribed the first WP of the route for the route definition, but is not intended to be the first WP to fly. Whit the message #43 is possible to select another WP as first WP to fly if necessary without changing the planned route.

9.3.1.3 The Health Management

The Health Management is the function responsible to monitor the correct working of the system and eventually report the failure to the ground segment. The module providing this function is the Health Monitoring module consists of a Failure Detection and a Health Management sub-modules as shown in Fig 39.

The Failure Detection is responsible to:

- Detect OBMC internal failures via Built-In Tests (BIT)
- Monitor periodically input frames in order to detect silence, CRC errors and Time Tag Frozen failures
- Monitor avionic equipment BIT in order to detect avionic systems failures
- Monitor the MIL-STD-1553B avionic bus in order to detect transmission failures.

The Health Management is responsible to:

- Activate and deactivate relevant warning considering the detected failures
- Report relevant warning to GCS
- Report relevant warning to TCS in accordance with the STANAG 4586 standard
- Reply to STANAG 4586 Sub-System Status Requests (message #1000) and Sub-System Status Detail Requests (message #1001)

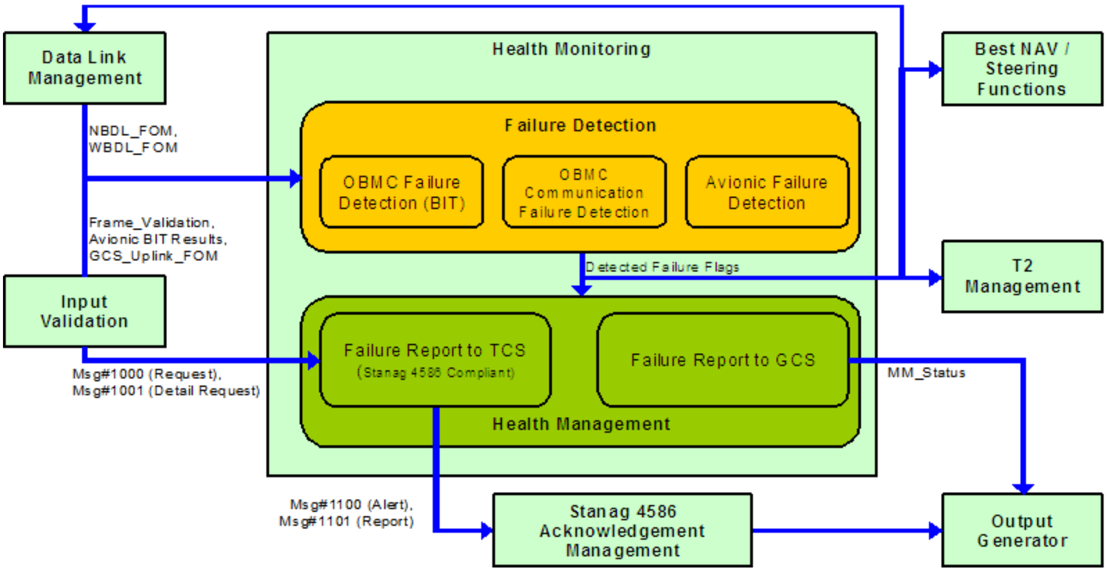

Data Link Management
Health Monitoring
Best NAV / Steering Functions
Failure Detection
NBDL_FOM, WBDL_FOM
OBMC Failure Detection (BIT)
OBMC Communication Failure Detection
Avionic Failure Detection
Frame_Validation, Avionic BIT Results, GCS_Uplink_FOM
Detected Failure Flags
T2 Management
Input Validation
Failure Report to TCS (Stanag 4586 Compliant)
Failure Report to GCS
MM_Status
Msg#1000 (Request), Msg#1001 (Detail Request)
Health Management
Msg#1100 (Alert), Msg#1101 (Report)
Stanag 4586 Acknowledgement Management
Output Generator

Fig 39 Health Management architecture description.

9.3.1.4 The EFCS Data handler

The EFCS Data Handler routes the Experimental Flight Control Computer commands to the module providing the surface actuation.

9.3.1.5 The NAV Sensor Management

The Navigation Sensor Management provide the sensor information to the others OBMC functions by managing the sensors linked to it. Two sources are provided for the inertial data and positioning: the AHRS and the INS; while the air data, such the airspeed and barometric altitude, are provided by the Air Data System (ADS) Fig 40.

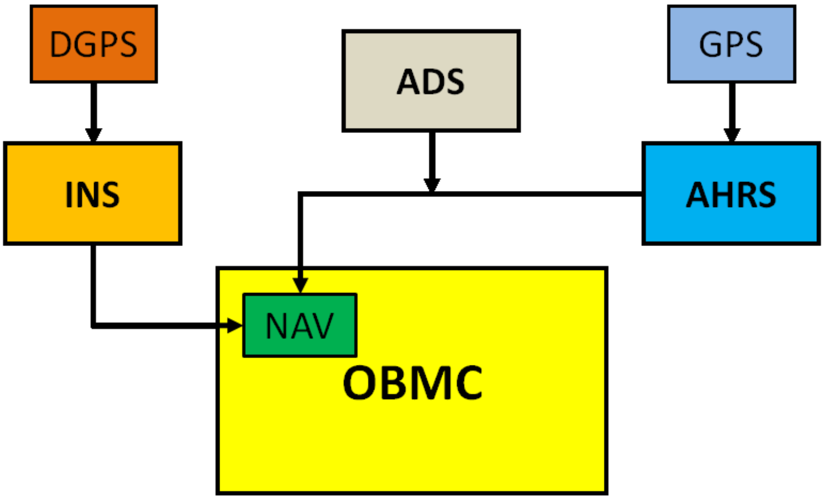


Fig 40 Navigation sensors system architecture.

AHRS

The Attitude and Heading Reference System (AHRS) is a component providing aircraft attitude, such pitch, roll and heading, thanks to magnetic sensors on and a software for the sensor data conversion. In the Sky-Y navigation system it is also provided by a GPS (secondary GPS) for the aircraft position detection.

INS

The Inertial Navigation System is made by inertial platform measuring the aircraft attitude and position. The INS is connected to a DGPS (primary GPS) for the bias error correction in the aircraft positioning estimation. In case this source is not available or not valid, the position is elaborated just by the INS integrating the inertial platform data. The INS is also provided to a Kalman Filter to improve the precision.

DGPS

The primary GPS for the Sky-Y navigation is a Differential GPS. The DGPS improves the traditional GPS precision by acquiring information from ground station displaced in different locations. The ground stations, knowing exactly their position, can estimate the error between their position measured by the GPS in that moment and the real one; then the correction is sent to the DGPS of moving vehicles to improve their position measure calculation.

Kalman Filter

The Kalman Filter (KF)is an algorithm used to process real-time measures to estimate the inaccuracies. The algorithm has a recursive nature and elaborates the current data respect to the previous steps and so produce the statistic optimal estimate of the system state.

9.3.1.6 The Best NAV Mode determination

The OBMC continuously determines the best source available for navigation, according to sensors validity. In order to grant graceful degradation of performance in case of failures, different modes are supported combining the current available sources.

9.3.1.7 The Best Data calculation

This function provide the AV sensors data selecting the best available sources based on the best navigation mode.

9.3.1.8 NAV data to WBDL

The OBMC sets up the Navigation messages to WBDL. For each message is also provided the Time Tag as an incremental value and the Validity flags.

9.3.1.9 FOM Evaluation

The OBMC is asked to evaluate the goodness of the received signal by computing the Figure Of Merit (FOM) of both the NBDL and WBDL uplinks.
The FOM calculation considers the numbers of "good" uplink messages #2000 received.
(message #2000 is a Private message: TCS Flight Controls Command).

9.3.1.10 Pof Determination

The Phase of Flight determination is used to automatically detect in which of four possible phases (Ground, Climb, Approach, Navigate) the UAV is. The PoF output is used for the lost link calculation, once the uplink is lost, the lost link state is latched and the defined actions for that state are executed. The state of the lost link process is provided in downlink. The transition between lost link states occur until the current state time-out expires.

9.3.1.11 Lost Link Calculation

The Lost Link calculation provide the Lost Link status by verifying the uplinks FOM according to the current Phase of Flight. Different values of NBDL and WBDL uplink FOM thresholds are set for each Phase of Flight as well as the time-out. The algorithm rises the Lost Link state in case of both the FOM are below the thresholds and rest below also after the time-out. The Lost Link state is lowed when at least one FOM returns over the proper threshold and a TCS recovery command is received.

9.3.1.12 ATOL Data

The Auto Landing Data Calculation determines the approach and land path on the desired runway and calculates the UAV divergences respect to that path in order to activate the FCS corrections. Such algorithm has been implemented in Simulink® for testing purposes and then has been loaded into the OBMC for data gathering flight tests.

9.3.2 Steering functions

The Steering partition is a set of 23 basic functions providing a waypoint based route navigation plus 3 advanced function: one for the navigation to a specific point (Nav-To), one for the sensor slaved navigation (Slave to Sensor), and one for the lost link event management during the route navigation. In case of basic navigation modes the NAV send the data directly to the FCS. In case of Steering navigation modes, independently from the mode selected, the Steering execute all the calculations and always output 4 values to the NAV for the FCS:

- Command Track = Value of track to keep (if the X-Track is zero)
- X-Track = distance of the UAV from the leg (if the leg has to be reached)
- Delta alt = difference between the desired altitude value and the actual
- Command Groundspeed = GS to keep

Then the NAV exclude the Command Groundspeed and/or the Delta Alt value coming from the Steering and use the value coming from the TCS if the navigation mode selected requires a pilot override of that attribute (see section 9.3.1.1). In this way the Steering works always as the mode selected was Nav4D, then a manual override of speed occurs in case of A/P mode = Nav3D, and an override of speed and altitude in case of A/P mode = Nav2D, Nav-To and Steer To Sensor.

The route navigation is based on the WP loaded in the *Route-Store* bringing all the WP data and the *Planned-Route* listing the sequence of flight of the WPs by reporting their *WP-Number* (see section 9.3.1.2). The route result to be the sequence of the segments connecting the WPs (Fig 41), plus the eventual loiter pattern due to loiter WPs (see section 9.3.2.1.2).

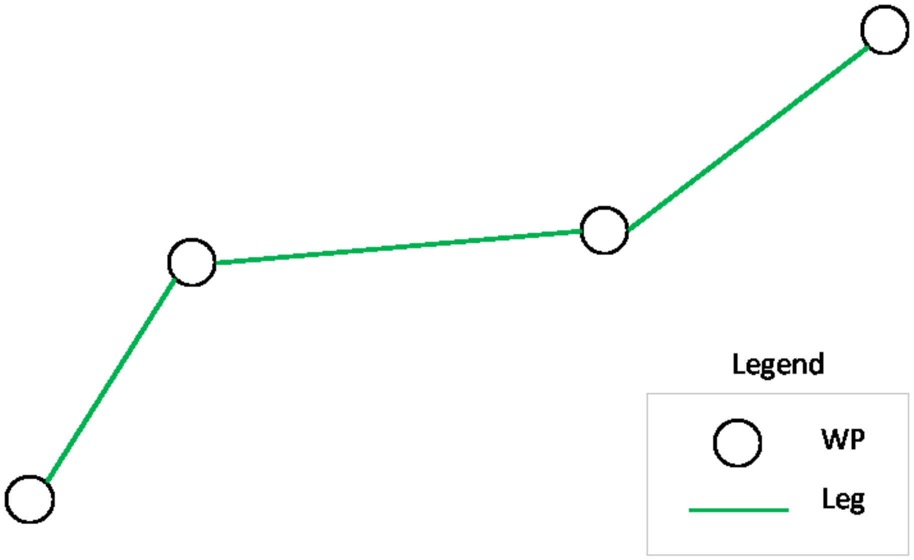


Fig 41 Route example.

The flight path depends from the *WP-Type* and from the AV position respect to the WP and leg. For going on with the description is necessary to introduce some variables used by the Steering algorithms (Fig 42), some are input variables, others are calculated by the Steering itself.

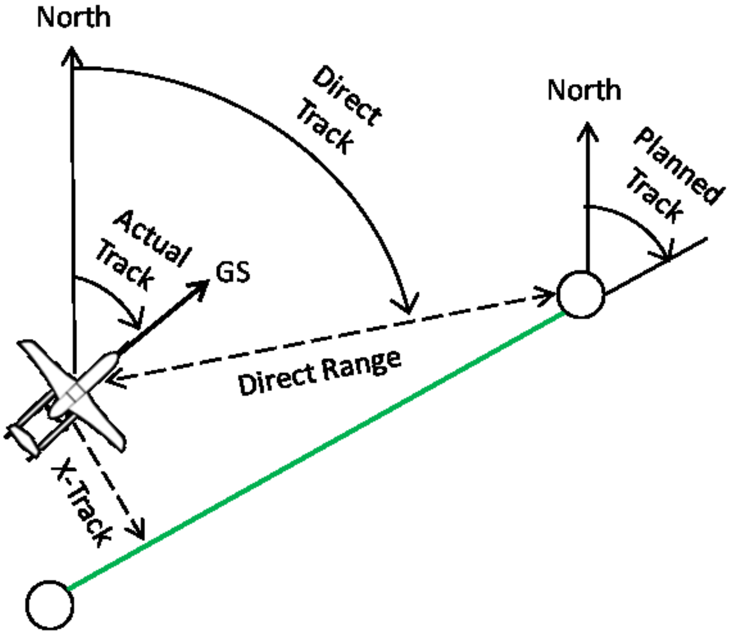


Fig 42 Steering variables convention.

9.3.2.1 Waypoint Type

The WPs along the route can be of three different type depending of the path required in that specific point. Two types are used for defining the *Turn Type*, one specifics that the WP is a *Loiter* WP. The *WP-Type* attribute is loaded in the *Route-Store* at the line corresponding to the relative *WP-Number*, the three possible values are:

- Fly-By
- Fly-Through
- Loiter

9.3.2.1.1 Turn Type

The *AV Position Waypoint* STANAG 4586 frame (#802), as seen before (section 9.3.1.2), is responsible to the WP data upload. One of its fields brings the 'Turn Type' and the STANAG 4586 fixes its value to 2 possible: *Short Turn* and *Flyover*. To be compliant to this field the steering partition is asked to command the change of the leg in 2 ways as shown in Fig 43. In order to perform the *Short Turn*, corresponding to the Steering *Fly-By*, the Steering algorithm should calculate the *Roll In Point* as the distance to the WP at which the change leg command should be done to allow the UAV to join the next leg with a constant turn radius without overshoot (Fig 43-b).

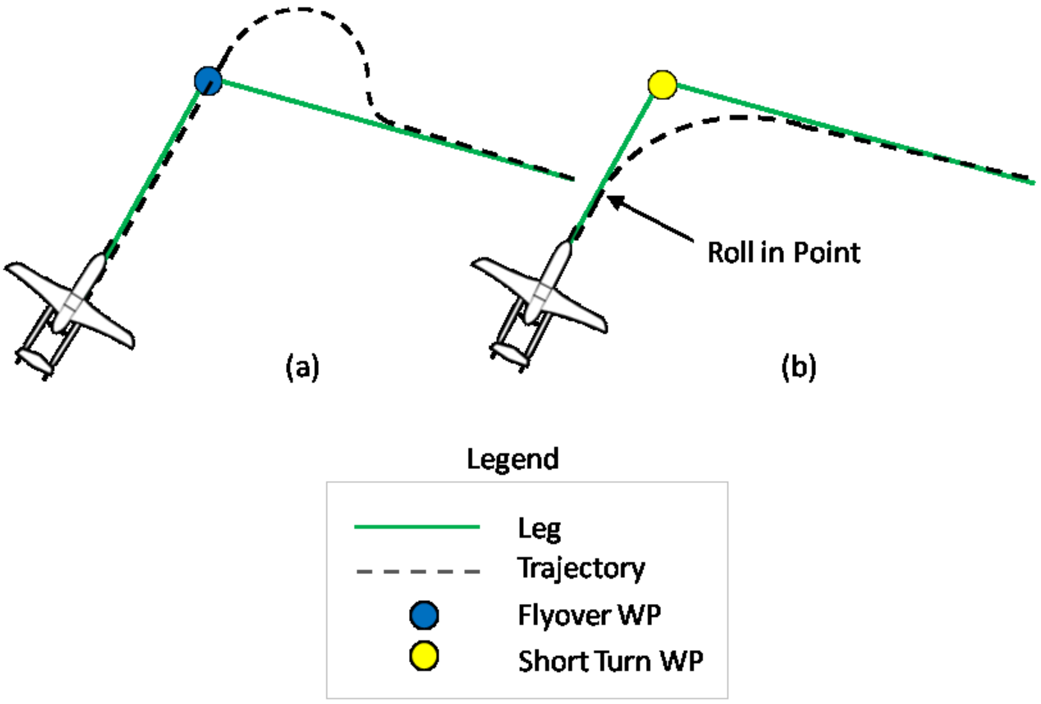


Fig 43 Turn Type: (a) Fly Over, (b) Short Turn.

For this aim the correct turn formulation is used taking into account the actual vehicle altitude, speed, bank angle, WP direct distance, direct track, and next leg track.

Nevertheless as the *Roll In Point* distance calculation can show out a infinite range of values is opportune to fix a limitation in acceptable range in order to keep ‘Short’ the turn by putting a maximum threshold. Because of this a leg overshoot is tolerated either on a *Short Turn* WP just in case of the WP is linking two legs forming a too little angle compared with the vehicle performances (Fig 44).

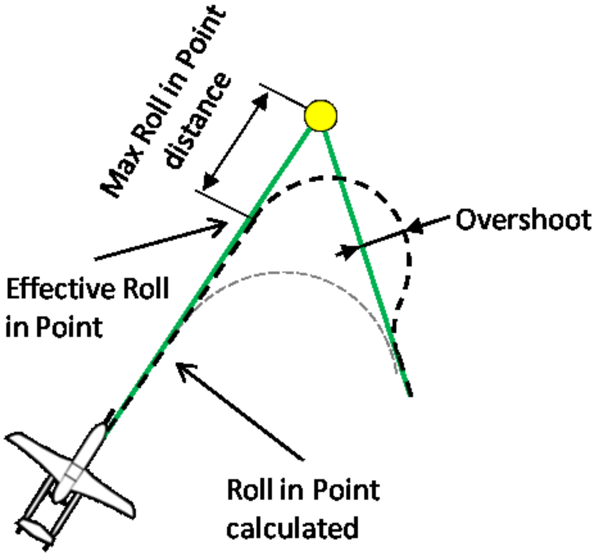


Fig 44 Roll In Point threshold.

The *Flyover* instead required an easier algorithm that just asks the system if the UAV position coincides with the WP coordinates by measuring the vehicle distance to the WP and evaluating if it results inside a fixed radius used as WP dimension tolerance. This provides the UAV to fly over the WP and then start the next leg approach (Fig 43-a). This results in a great overshoot on the new leg.

It is important to note that whatever the *Turn Type* is the turn is commanded just by changing destination WP, it means that all the variables described in Fig 42 (except *Actual Track*) and the output variables (see section 9.3.2) would be suddenly calculated respect to the new WP without any transition. Is the FCS to control the turn with the Sky-Y control laws according to the new variables. So the Steering partition task is to calculate the right moment for commanding the change leg by keeping in account the FCS behavior in order to obtain a coordinated turn.

The last important consideration is that the *Delta-Alt* is not considered for determinate the WP acquisition; it means that a change leg can occur also if the desired altitude is not reached.

9.3.2.1.2 Loiter

The *Loiter* WPs are need some more information respect to the ordinary one. As seen in section 9.3.1.2 the STANAG 4586 itself sends an additional message (#803) for each loiter WP of the route. In the Steering partition the loiter attributes are written in the *Route-Store* at the line of the corresponding WP together with all the other WP data. The *Loiter-Type* prescribed by the STANAG 4586 are 3 (for a fixed wing *Hover* has been excluded):

- Circular
- Racetrack
- Figure8

All the loiters are intended at constant altitude and speed (which is also the altitude and speed of the approaching leg). For each is possible to define the *Radius*, the turn *Direction*, and the *Loiter Time*; for the *Racetrack* and *Figure8* it is also definible the *Length* and the *Bearing*.

From the Fig 45 taken from the STANAG 4586 is possible to see that *Racetrack* and *Figure8* are described with a *Fly To Point* that has been interpretated as a loiter pattern *Entry Point*. For the *Circular* instead the *Entry Point* is not described so the the decision has been to leave the leg on time for have a smooth joint of the circle.

The exit condition is just the end of the *Loiter Time* countdown (started at the loiter path entrance). This means that does not exist a fixed *Exit Point*, or an exit condition as the number of laps. Of course a pilot command can exit any time the loiter path by selecting a different destination WP.

The Steering partition task is to alaborate the loiter data and figure out the way of producing the right correction to demand the FCS to conduce the UAV along the loiter path. For the *Circular Loiter* the solution is easier: entering the loiter the Steering output of *Command Track* beacomes always tangent to the circle and the *X-Track* becomes the distance from the circle (see *Route Section* = *Loiter Circular* in section 9.3.2.2).

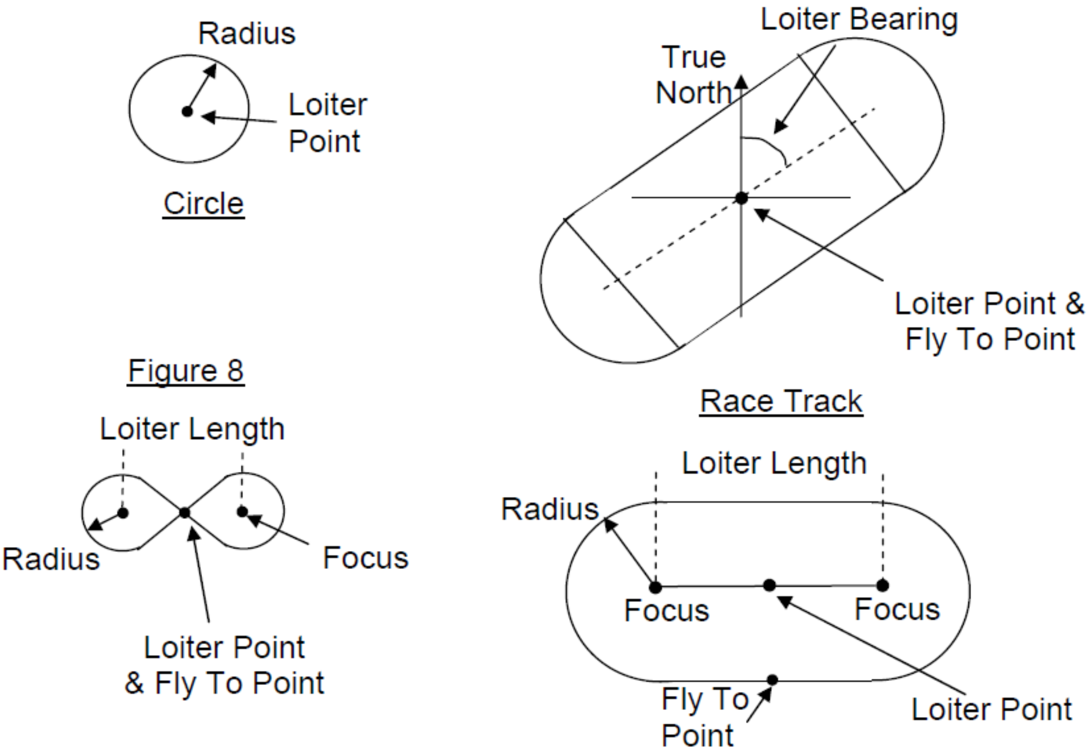


Fig 45 STANAG 4586 loiter pattern [14].

For the Racetrack and the Figure8 the solution adopted is more complicated due to the path shape. A specific Steering function generates new WPs, according to the loiter data, to recriate the loiter path and permit the UAV to navigate the loiter as a normal route. The *Loiter Path WP*s data and *Entry Poin WP* (always a Fly-By WP) are calculated any time a loiter non *Circular* is near to be flown and stored into a dedicated area of the *Route-Store* that is overwritten at any *Loiter Path WP*s generation.

This approach requires the Steering automatic on board replanning of the loiter route, the initialization of such a route at any lap, and the replanning of the original route after the loiter completion. Morover also for the loiter approach and entrance is required a replanning of a transition route passing through the *Entry Point*. An example of this internal replanning for a *Racetrack* is shown in Fig 46. In the *Loiter Path WP*s generation of a *Loiter Racetrack* 6 WPs are generated. 4 WPs are *Fly-Through* WPs and 2 are navigated as a *Circular Loiter* for half a circle (see *Route Section* = *Loiter Curve* in section 9.3.2.2). One *Entry Point* WP is placed at the first internal replanning (Fig 46-b), then, afrher the first lap, during the second end the following internal replanning the *Entry Point* WP disappears (Fig 46-c).

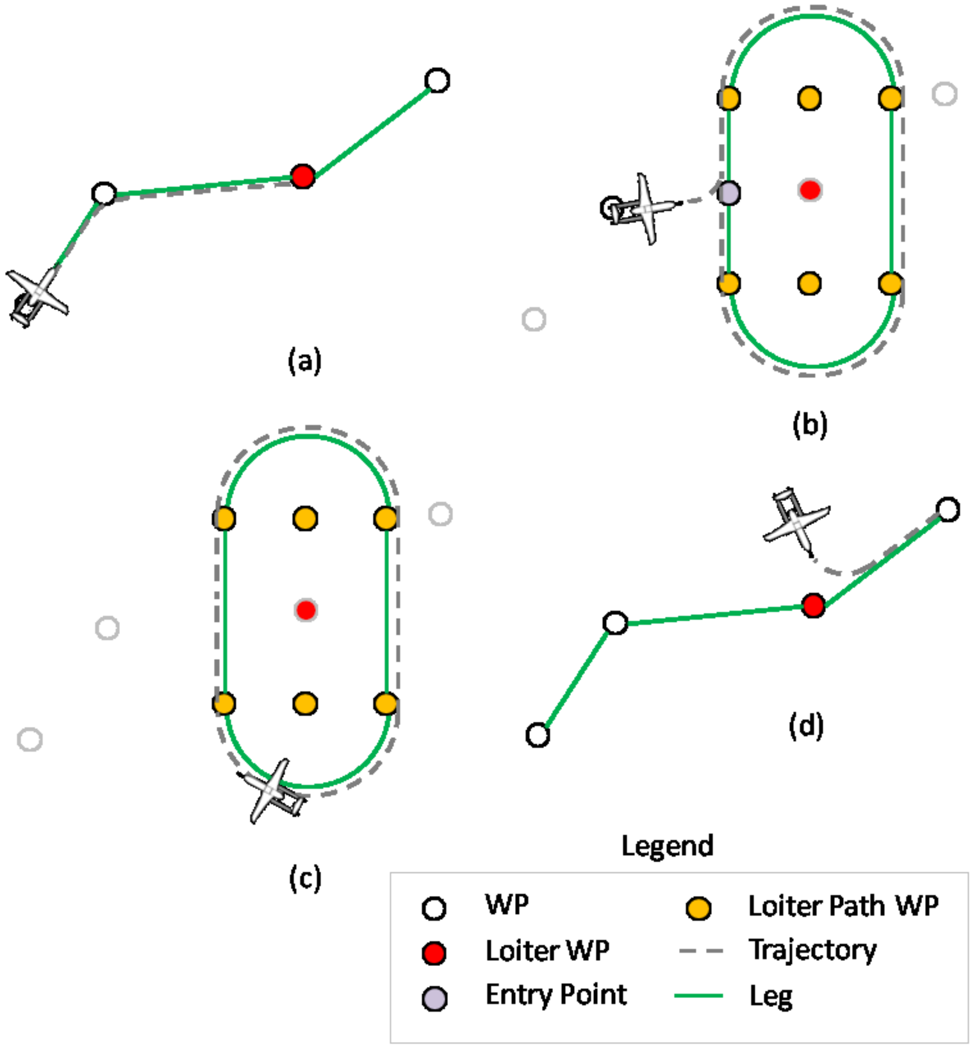
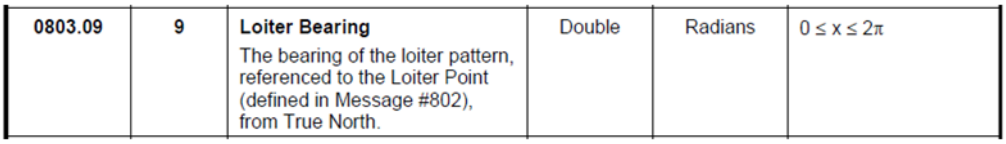


Fig 46 Steering Racetrack Loiter path automatic re-planning: (a) path before the loiter leg, (b) Loiter Path WPs generation and Steering internal route re-planning,(c) loiter path internal re-planning without Entry Point for the second and following laps,(d) loiter exit with Steering internal Original Route re-planning.

The sequence of flight depends on the *Loiter Direction*, but accordning to the STANAG 4586, the *Entry Point* is always on one straight leg. This interpretation is due to the loiter explanation figure of the STANAG 4586, the same reported in Fig 45, and becouse of the absence of a specific field in the *AV Loiter Waypoint* STANAG 4586 message.



0803.09	**9**	**Loiter Bearing** The bearing of the loiter pattern, referenced to the Loiter Point (defined in Message #802), from True North.	Double	Radians	0 ≤ x ≤ 2π

Fig 47 STANAG 4586 extract: message #803 - AV Loiter Waypoint, field #9 - Loiter Bearing [14].

Furter the same message shows the ranges of each variable and the *Loiter Bearing (*Fig 47*)* goes from 0 and 2π, but as both the *Racetrack* and the *Figure8* have a symmetric shape, a 0 to π rotation would be sufficient. This brought to the conclusion that the *Entry Point* is prescribed to be just on one side of the *Racetrack* leg and the entrance on the opposite side should be obtained by rotating the path with the additional bearing range (from π to 2π). The Fig 48 shows how the same Loiter Racetrack path displacement can be obtained by adding π to the Loiter Bearing just for changing the Entry Point side position (Fig 48).

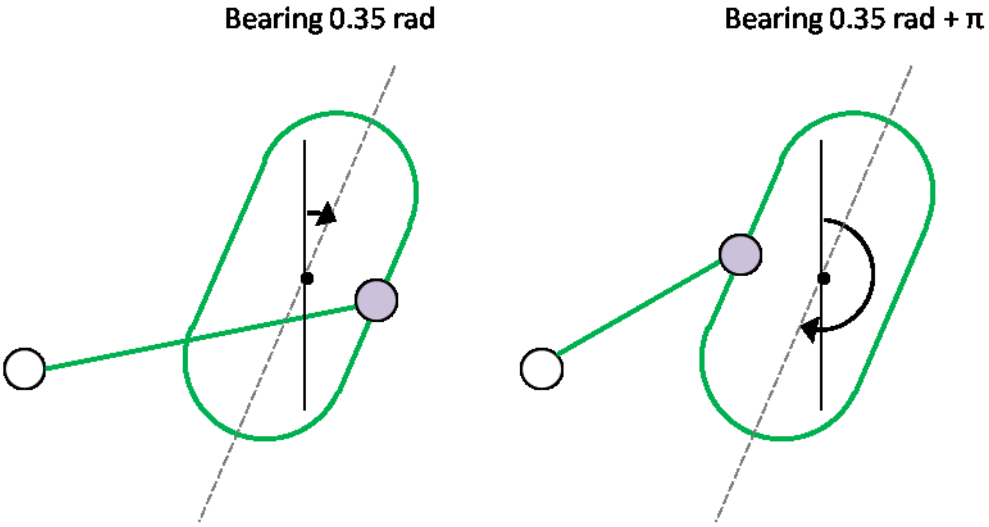


Fig 48 Entry Point positioning on the same Racetrack displacement by varying of π the Loiter Bearing.

For the *Figure8* path the *Entry Point* is set to be at the middle of the figure where the two straight legs cross each other, so it does not vary with the *Bearing*. Otherwise the replanning mechanism is the same of the *Racetrack* case: When the loiter WP is angaged as a destination WP the Steering calculates the *Loiter Path WP*s position and generates the new loiter path route with the *Entry Point* WP at the same coordinates of the route *Loiter WP* (Fig 49-b). After the first lap for all the following laps the Steering replans the loiter path route without the *Entry Point* (Fig 49-c). When the *Loiter Time* ends the Steering replan the *Original Route* (Fig 49-d).

Another point to analyze is the *Fugure8 Loiter Direction* verse. The STANAG 4586 prescribes four values two of which are: *Clockwise* and *Counter-Clockwise*. These are evident for the *Racetrack*, but for the *Figure8* having two counter-rotating circles has to be decided a convention. For a Figure8 with bearing 0 the conventional clockwise direction has been choose as the clockwise direction of the upper circle.

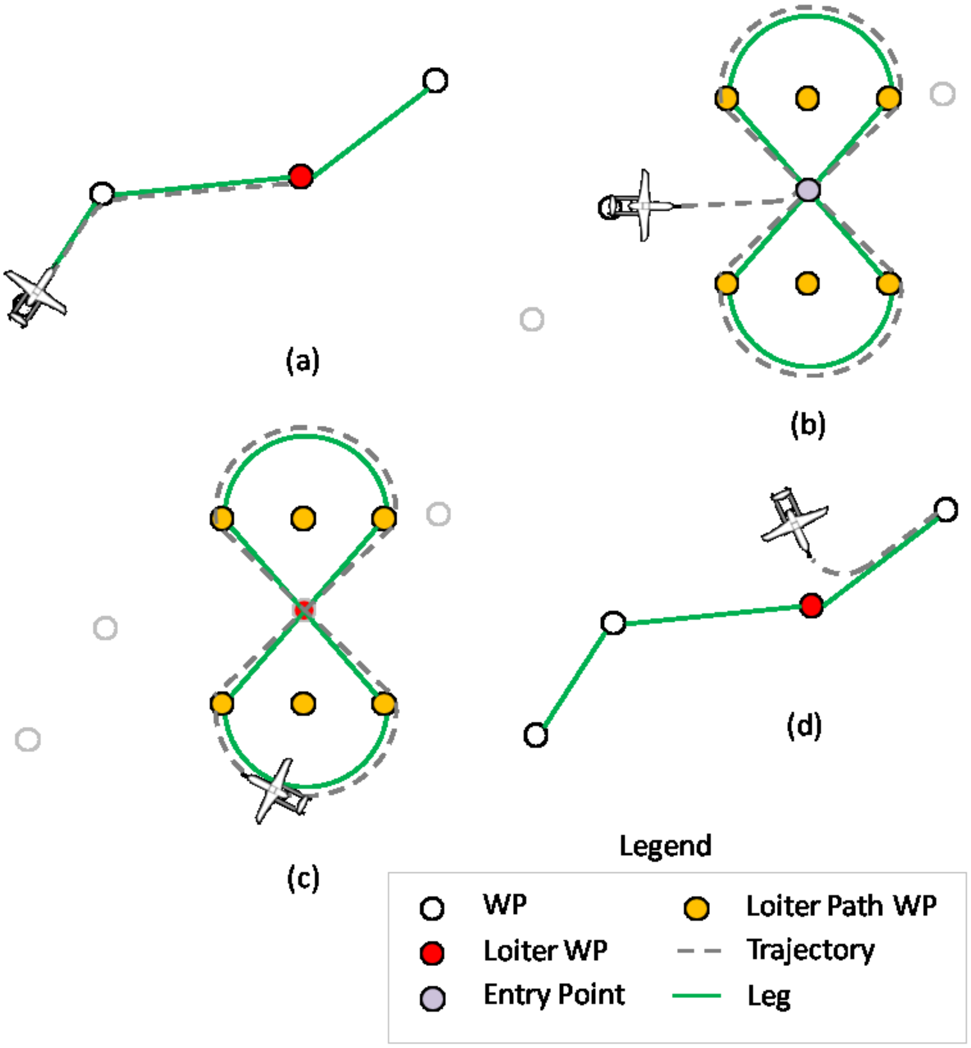


Fig 49 Steering Figure8 Loiter path automatic re-planning: (a) path before the loiter leg, (b) Loiter Path WPs generation and Steering internal route re-planning,(c) loiter path internal re-planning without Entry Point for the second and following laps,(d) loiter exit with Steering internal Original Route re-planning.

About the path entrance the first WP after the *Entry Point* results to be dependent from the verse and from the UAV position to avoid too high turn angles. 4 sectors are delineated by the crossing legs prolongation for defining the flight order. In case of *Clockwise Figure8* loiter, as shown in Fig 50, if the UAV is coming from the sectors I, II or IV the WP following the *Entry Point* is the number 2; in case of UAV coming from sector III the following WP is the number 4. In case of *Counter-Clockwise Figure8* loiter instead (Fig 51), if the UAV is coming from the sectors II, III or IV the WP following the *Entry Point* is the number 4; in case of UAV coming from sector III the following WP is the number 2.

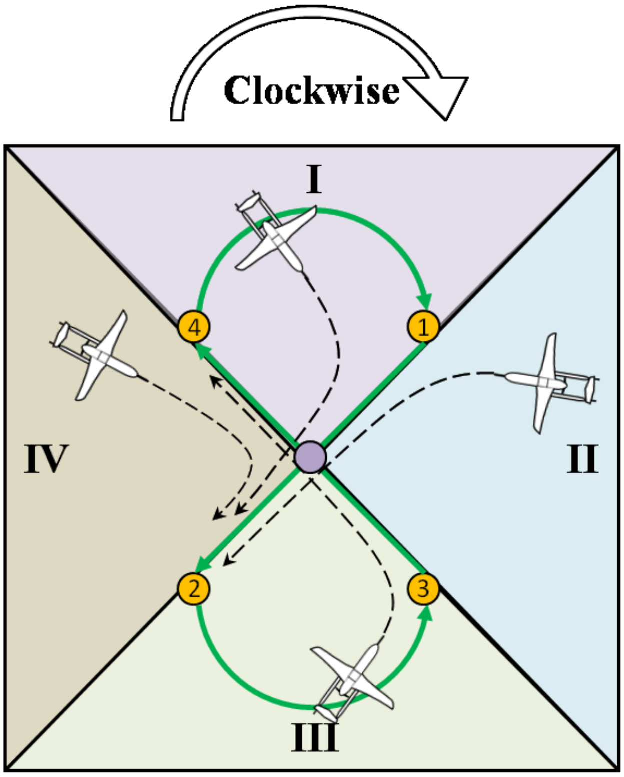


Fig 50 WPs sequence at the entrance of a Clockwise Figure8 loiter based from the UAV sector position.

The solution showed is what has been implemented in the first iteration trial for being as much as possible STANAG 4586 compliant. Nevertheless after some simulation flight trials a different solution in the *Racetrack* loiter approach has been preferred and implemented for having a path approach similar to the *Circular* one. The new solution uses the same sectors found out for the *Figure8* case also for the *Racetrack*. In this case the *Entry Point* WP has been eliminated and the loiter path approach results to be: on the upper curve in case of UAV coming from sector I, on the right straight leg in case of UAV coming from sector II, on the lower curve in case of UAV coming from sector II, and on the left straight leg in case of UAV coming from sector IV.

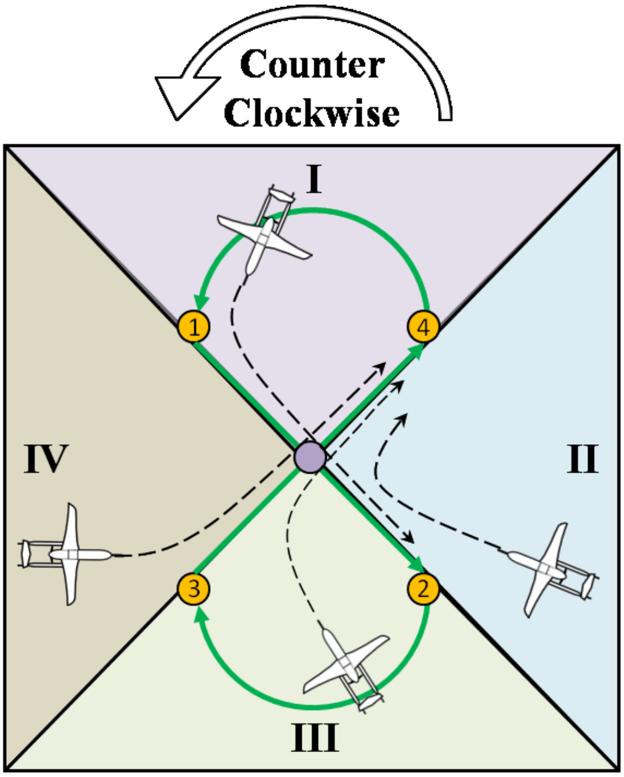


Fig 51 WPs sequence at the entrance of a Counter-Clockwise Figure8 loiter based from the UAV sector position.

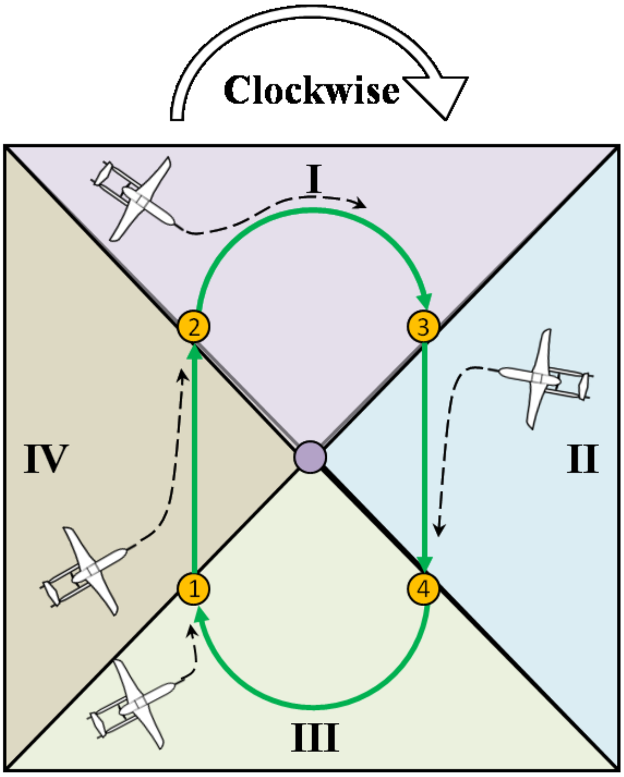


Fig 52 Loiter path approach with no Entry Point for Racetrack loiters based on the UAV sector position.

9.3.2.2 Route Sections

The way of navigating a route does not depend only from the *WP Type*, but also from the position of the UAV respect to the leg or respect to the path section in case of loiter. This fact is expressed by the Steering variable *Route-Section*. The *Route-Section* is individuate by the Steering and its value is transparent to the ground segment, it is just an internal variable used for performing correctly the expected trajectory. For doing this the 2D output variables to the FCS (*Command Track* and *X-Track*) are calculated in different ways. The *Route-Section* possible values are 6:

- *No Sec*
- *Straight*
- *Direct*
- *Overfly*
- *Loiter Circular*
- *Loiter Curve*

No Sec is used just for defining an initialization value of the variable when a route is not present on board. The Steering is not providing any correction to the FCS.

Straight is the value used in case the UAV has to follow the current leg. This happens when the UAV is inside a cone of ± 55° respect to the leg with vertex centered on the WP of destination and distance to the WP greater then a 2M radius (Fig 53-White Aircraft).

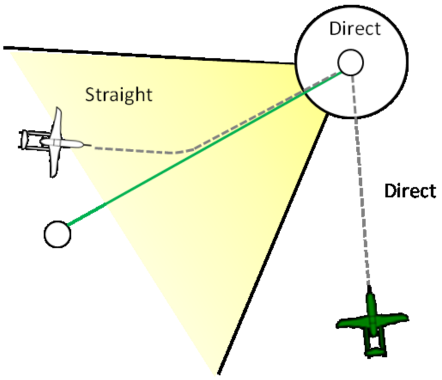


Fig 53 Straight and Direct Route-Section conditions.

In this condition the UAV is demanded of reaching the leg and keep that trajectory until the destination WP. For doing this the Steering output variables are give in this way: the *Command Track* is equal to the *Planned Track* (that is the track of the leg), and the *X-Track* is calculated as the distance between the UAV and the leg (positive at the right of the leg). The FCS is programmed to reach the leg by annulling the *X-Track* with a 45° approach and, when the *X-Track* is near to zero, to keep the *Command Track* (that, in this case, is the leg track) Fig 54.

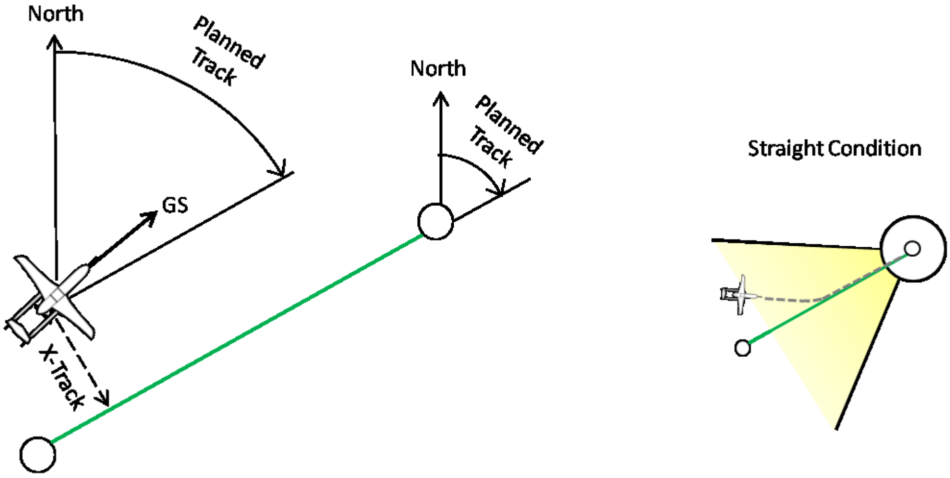


Fig 54 Current Leg Section = Straight, Steering output variables and Straight Condition.

Direct is the value used for reaching directly the WP without considering the leg. It happens when the UAV is outside the cone (Fig 53-Green Aircraft) or the distance from the WP is closer than 2M. In this case the Steering output are calculated in the following way: the *Command Track* is equal to the *Direct Track*, which is the track of the conjunction between the UAV and the WP, and the *X-Track* is fixed to zero in order to avoid the FCS correction of this error. This condition, due to the zero value of the *X-Track*, is interpreted from the FCS as a Straight condition in which the leg has been acquired, so that the track kept is exactly the commanded one Fig 55.

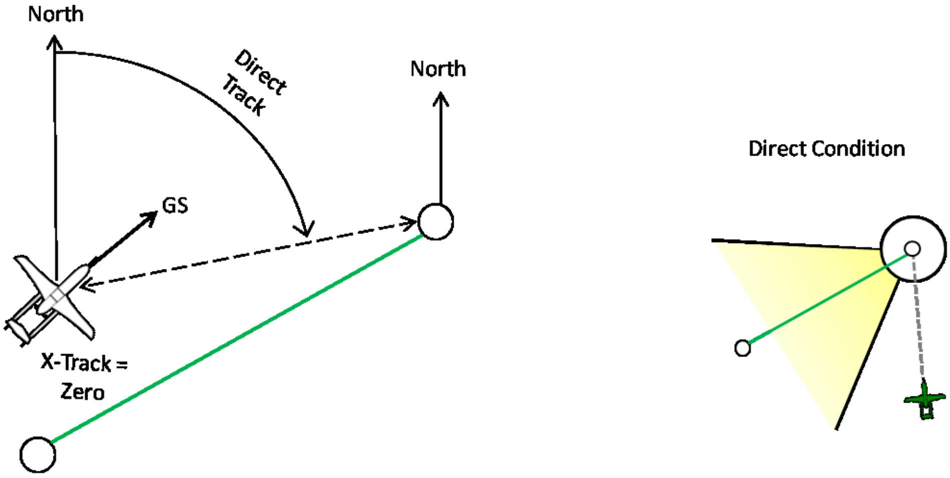


Fig 55 Current Leg Section = Direct, Steering output variables and Direct Condition.

Overfly is the condition that occurs during the WP acquisition. It is used just for one computational cycle, due to the Steering functions sequence, to reinitialize the Steering variables at the following WP values avoiding biased interpretation.

Loiter Circular is the condition used for flying the *Circular* loiter WPs.

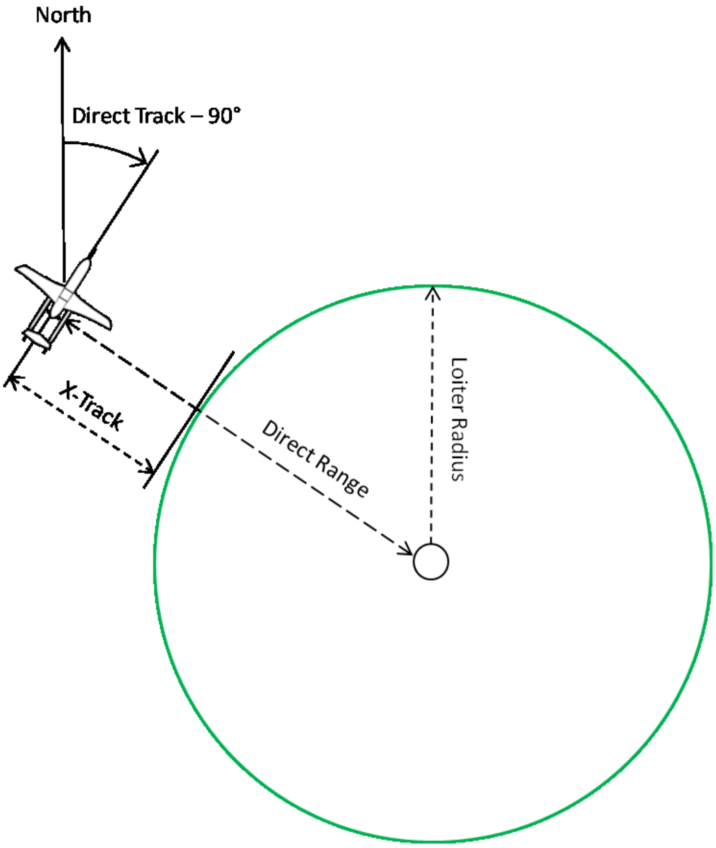


Fig 56 Leg Section = Loiter Circular, Steering output variables.

When the loiter starts the output values to the FCS are calculated as follows: the *Command Track* is calculated as the *Direct Track* ± 90° in order to require a direction always parallel to the circle tangent (+ if *Counter-Clockwise*, – if *Clockwise*), and a *X-Track* calculated as the *Direct Range – Loiter Radius* so that the FCS, correcting the *X-Track* distance, asks the UAV to reach and keep the radius distance Fig 56.

Loiter Curve is used in case of *Racetrack* and *Figure8* loiters. As seen in section 9.3.2.1.2 the non *Circular* loiters have two straight segment connected to two semicircles. The straight segments are flied like normal route segments, so in *Straight* mode or *Direct* mode according to the cone mechanism, while the semicircles are flied like the *Loiter Circular* case, but to separate these cases from the *Loiter Circular* one this section type has been called *Loiter Curve*.

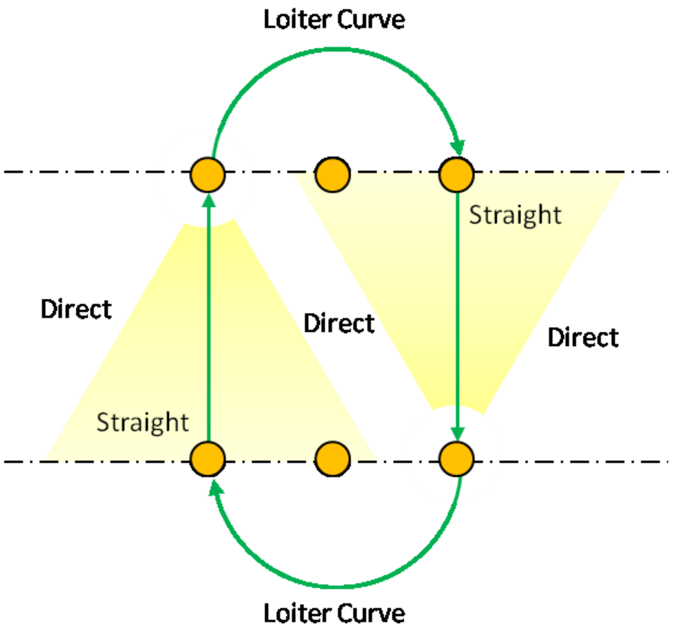


Fig 57 Leg Section = Loiter Curve, Straight and Direct in a non Circular loiter.

9.3.2.3 Altitude and Speed

The altitude command to the FCS is given as the difference between the *Next WP* altitude attribute and the current UAV altitude, so is positive in case of climb and negative in case of dive. In any case the FCS will command an altitude variation to annul the difference as soon as possible, that means that the UAV is not asked to follow the line connecting the two WPs as ramp (Fig 58). The legs are followed just in the two dimensions. Moreover the altitude is not considered a requirement for the WP acquisition. The WP acquisition consider just the two dimensions.

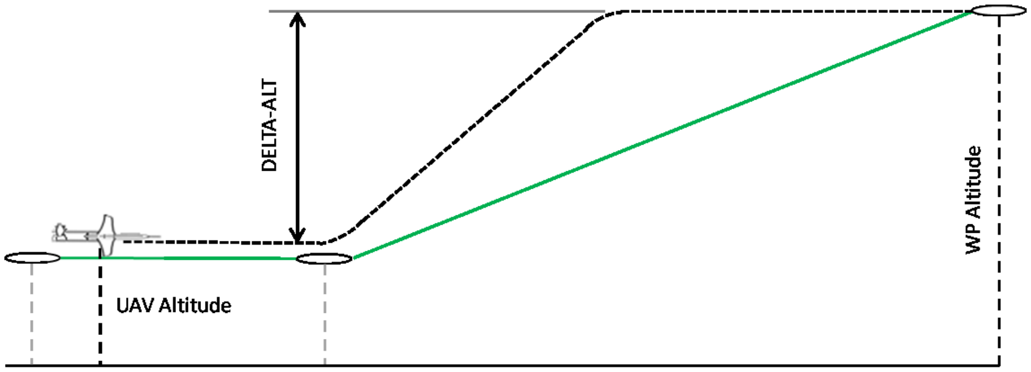


Fig 58 Altitude acquisition example.

About the speed the STANAG 4586 message #802 (Fig 36) prescribe two fields mutually exclusive: *Waypoint to Speed* and *Arrival Time*. In the first case the WP brings the value of speed to keep during the leg of its acquisition, in the second is marked the time at which is desirable to reach the WP. The STANAG 4586 also prescribe the possibility of choosing a speed type, but for the Sky-Y TCS this function has not been implemented and only the IAS is accepted. As the Steering command to the FCS a groundspeed, in case of WP speed attribute, the Steering should perform the conversion from IAS to GS taking into account the speed and direction of the wind. The case of *Arrival Time* instead requires an iterative calculation for providing the exact value of GS to reach on time the WP. If the required time is in the past the maximum speed is commanded, if is too in the future the minimum speed is commanded even if not sufficient to retard the arrive, but the trajectory is not modified with possible retarding path. Both the speed attribute are affected by the altitude variation due to the FCS way of managing the climb and the descent. The FCS in fact imposes the IAS of climb to 95 kts regardless a different Steering command, and in case of descent imposes a constant vertical speed value. This behavior affect also the manual speed override in case of 2D and 3D *A/P Mode*. The Steering speed calculation for the *Arrival Time* accomplishment takes in consideration this behavior when a difference of altitude is verified. The last consideration about the *Arrival Time* attribute is that, as in the STANAG 4586 message is not present a *No Attribute* value, result impossible to have a WP without a speed or time value. This means that is not possible to have a final *Arrival Time* for more than one leg but all the legs should have or a time attribute or a speed attribute.

9.3.2.4 Change Destination WP

During the route flight the pilot can intervene to skip one or more WPs or come back to a previous WPs of the route by changing destination WP. With this command a message #43 start from the TCS to the OBMC and the NAV.DL operate the data interpretation for extracting the *Commanded Waypoint Number* and trigger the Steering partition. A specific Steering function called *Change Destination Actions* verify the correctness of the command checking the presence of the commanded WP inside the route and, in case of success, operate the change of destination WP. The UAV *Present Position* is acquired and connected to the selected WP with a virtual leg so that the UAV is asked to fly directly to the new WP in a *Straight* leg section mode. The expected behavior is that the UAV would start a curve for leaving the current leg and approach the new virtual leg connecting the selected WP to the UAV position at the *Change Destination Command* time Fig 59 and Fig 60.

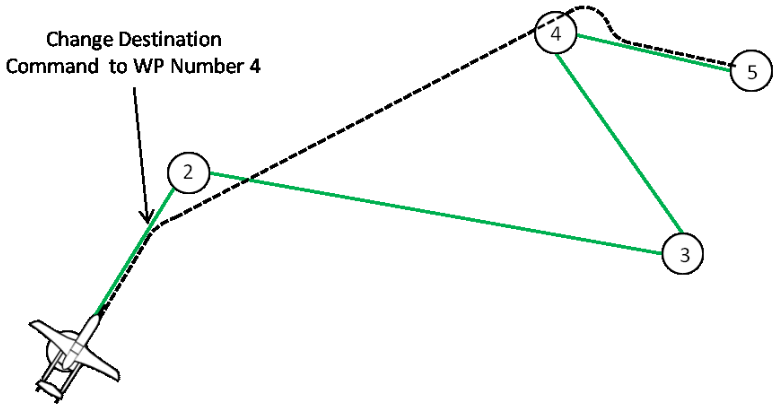


Fig 59 Change Destination Command for skipping two WPs.

The message #43 occurs also during the route engagement. The *Change Destination* function in fact is used automatically any time the UAV has to begin a route navigation to select the starting WP.

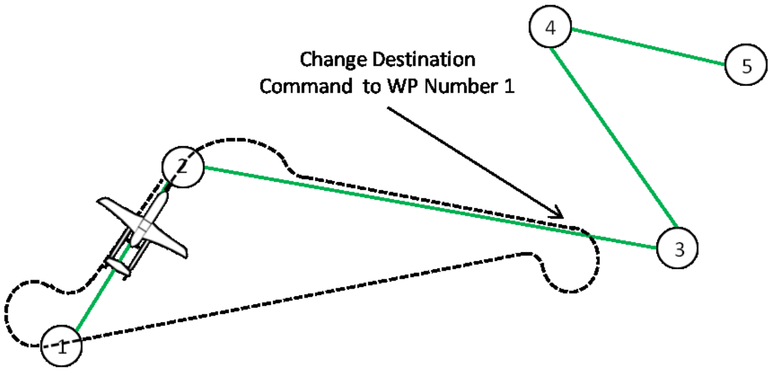


Fig 60 Change Destination Command for coming back to a previous WP.

By default the starting WP is the first WP of the route, but at the engage the pilot can select any other WP as starting WP. To be noted that the use of this function at the route engagement imply the approach to the selected WP always in a *Straight Leg Section* mode due to the virtual leg generation.

9.3.2.5 Tracks calculation – Lambert approximation

For the route navigation is necessary to combine the need of visualize and measure the route in a two-dimensional sheet with the real case of spherical world. This fact affect the navigation any time a track to a WP and a distance between two points have to be calculated. For solving this problem a sub-function implementing the Lambert transformation has been implemented in the Steering partition to be called by all the others Steering functions needing this calculation.

Lambert projection

The Lambert map is based on a conical projection of the sphere [19]. Considering a cone coaxial with the heart axis tangent to the sphere in a point. The tangent parallel projection is a parallel with the same length of the original one. Connecting the center of the sphere with a parallel of latitude φ a coaxial cone with respect to the tangent cone is obtained. Their intersection generates a circle, with center on the cones common axe, that represents the parallel projection on the tangent cone Fig 61.

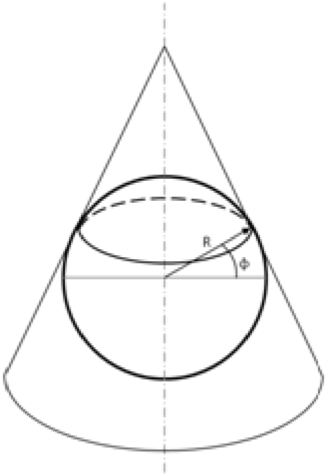


Fig 61 Lambert projection.

Unfolding the tangent cone on a sheet the parallels will be represented by arcs of circle with center in the pole Fig 62.

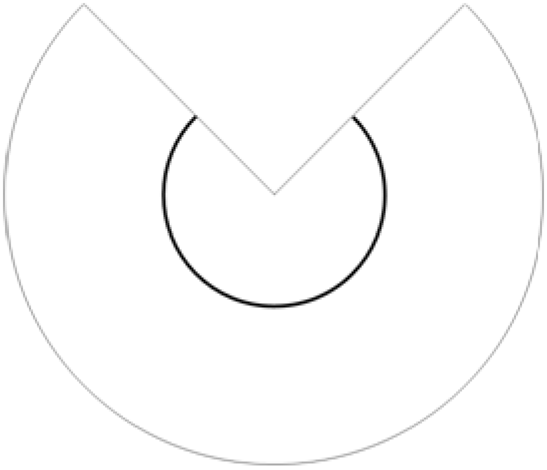


Fig 62 Unfolded tangent cone.

The meridian projection from the center of the hearth on the cone will be represented by straight lines converging in the pole and passing on the corresponding meridian in the tangent point of the cone with the sphere surface Fig 63.

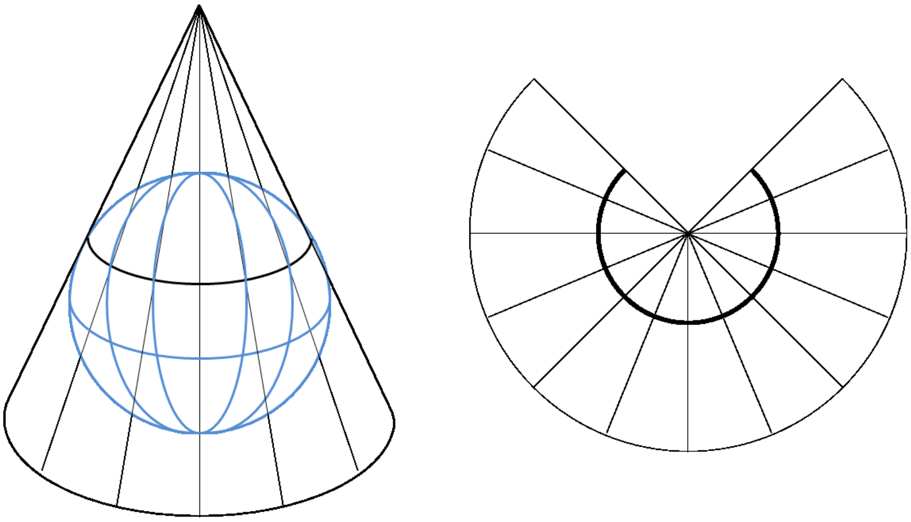


Fig 63 meridian projection on the cone.

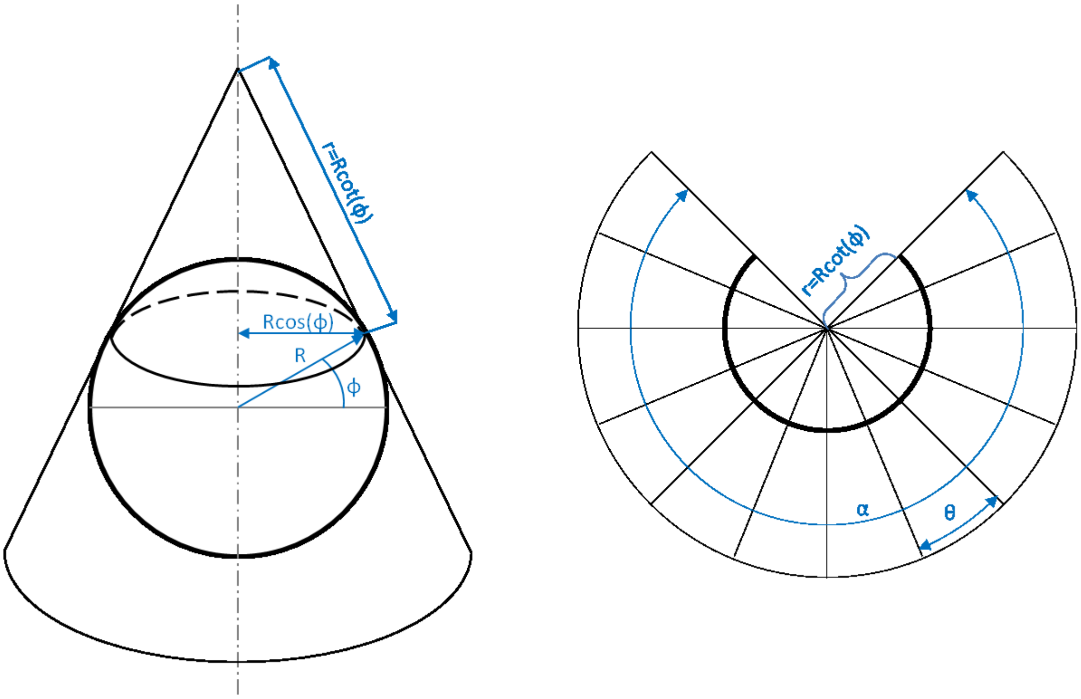


Fig 64 Parallel arc of circle calculation.

The tangent parallel is an arc of circle of radius Fig 64:

$$r = R \cdot cot\varphi$$

with:
r = radius of the tangent parallel circle
R = radius of the sphere
φ = latitude of the tangent parallel

The angle between two meridians θ represents the projection of the difference of longitude Δλ between the two meridians and is called *meridian convergence*. Called α the angle of the circle arc the equivalence between the circle arc on the unfolded cone and the tangent parallel circle is:

$$\alpha \cdot R \cdot cotan\varphi = 2 \cdot \pi \cdot R \cdot cos\varphi$$

simplifying:

$$\alpha = 2 \cdot \pi \cdot sin\varphi$$

knowing the longitude difference Δλ, by imposing the proportion:

$$\frac{2 \cdot \pi \cdot sin\varphi}{2 \cdot \pi} = \frac{\theta}{\Delta\lambda}$$

it is possible to find out the *meridian convergence*:

$$\theta = \Delta\lambda \cdot sin\varphi$$

Track and distance calculation

For the Sky-Y navigation the previous theory have been used any time a track or a distance between two points on the hearth has to be calculated. Called pos1 and pos2 the positions (lat & lon) of the two points the cone has been considered tangent to pos2. Moreover, as the earth is not perfectly spherical, different values for the north and east radius in the pos2 have been calculated with the WGS84:

```
sin_lat = sin(pos2.lat);

cos_lat = cos(pos2.lat);

east_radius = (WGS84_A * cos_lat) / sqrt(1 - WGS84_EPS2 * sin_lat * sin_lat);

north_radius = WGS84_A * (1 - WGS84_EPS2) / pow((1 - WGS84_EPS2 *
sin_lat * sin_lat),1.5);
```

Where WGS84_A is the major semi-axis (6378137m) and WGS84_EPS2 is the square of the eccentricity (0.00669437999013). Then the Lambert variables are calculated:

```
delta_lat = (pos1.lat - pos2.lat);

delta_lon = (pos1.lon - pos2.lon);

lambert_e = east_radius * delta_lon;

lambert_n = north_radius * delta_lat;

merid_conv = delta_lon * sin(pos2.lat);
```

So if the meridian convergence is zero:

```
lamb_x = lambert_e;

lamb_y = lambert_n;
```

If the meridian convergence is different from zero:

```
lamb_x = (lambert_e/merid_conv - lambert_n) * sin(merid_conv);

lamb_y = lambert_e/merid_conv - (lambert_e/merid_conv - lambert_n) * cos(merid_conv);
```

Then for calculating the track and the distance, for example the *Direct-Track* of the UAV respect to the destination WP and the corresponding *Direct-Range*:

```
If (lambert.lambert_x =! 0)

DIRECT_TRACK = RAD_90 - atan(lambert.lambert_y/lambert.lambert_x) + lambert.meridian_conv;

if ( lambert.lambert_x > 0)

DIRECT_TRACK = RAD_270 - atan(lambert.lambert_y/lambert.lambert_x) + lambert.meridian_conv;
```

Then the if the *Direct-Track* results outside the 0 to 360 deg bound the value is converted inside that bounds.

Instead

```
if (lambert.lambert_y > 0)

DIRECT_TRACK = RAD_180;
```

Otherwise

```
if ( lambert.lambert_y <= 0 )

DIRECT_TRACK = RAD_0;
```

At the end, as the Steering uses *Direct-Track* bounded between -180 and 180 deg, the value is converted into this range.

The distance calculation:

```
DIRECT_RANGE = sqrt(lambert.lambert_y*lambert.lambert_y + lambert.lambert_x*lambert.lambert_x);
```

9.3.3 Advanced Steering functions

9.3.3.1 Direct navigation to a specific point – Nav-To flight mode.

During a mission flight is sometimes useful to have the possibility of directing the UAV to loiter on a specific point. For example while the UAV is flying a route if the sensor operator detects something interesting in a particular point the best way for investigating that point is to bring the UAV to loiter near or over that point. The same function can be used also to freeze the mission and hold the UAV in a waiting loiter. Because of this the navigation mode Nav-To has been introduced. The pilot has the possibility of choosing the loiter type (circular, racetrack or figure8), the loiter parameters (radius, length, bearing and direction), the point coordinates (lat & lon), then the altitude and the speed are imposed with an override. The Steering function Nav to steer init was designed to perform the direct steer of the UAV to the loiter Nav-To point and, if required, to bring back the UAV to the point of the route from which the route wad left. The algorithm generates a virtual leg connecting the UAV present position at the Nav-To command time to the loiter Nav-To point Fig 65. The main route is stored in order to be re-planned in case of exit Nav-To commanded demanding the previous route re-acquisition. For the same reason the UAV present position is acquired and converted into a WP for planning the back to route path. The Nav-To function can be activated also starting from a different navigation mode like the Altitude-Heading or the Vertical_Speed-Heading. In such cases the Nav-To exit command is given just by selecting a different navigation mode.

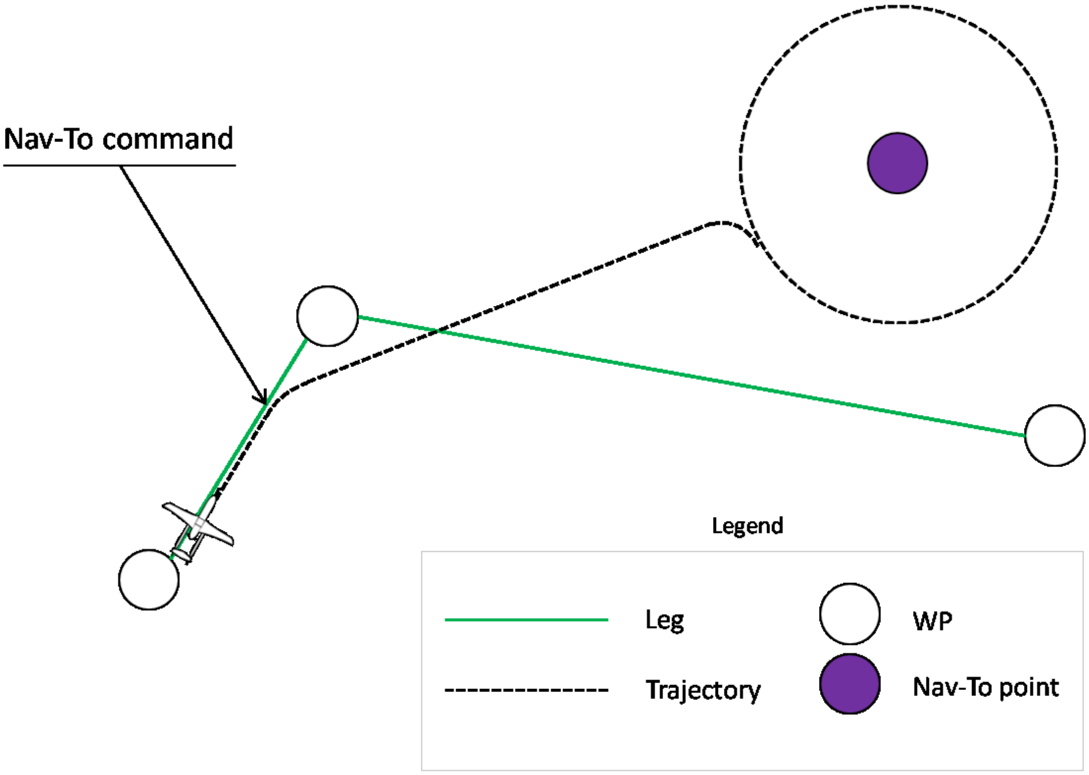
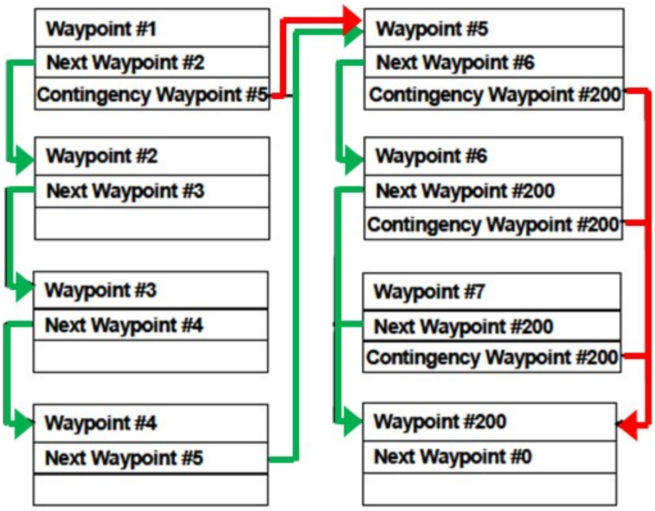


Fig 65 Nav-To mode command from route navigation.

9.3.3.2 Lost Link event management – Contingency Routes

One of the most criticalities for a UAV is represented by the data-link signal. For this reason both the up-link and the down-link have a redundancy, but in case of lost of both the up-link signals the AV result to be uncontrollable. The level of automation of the platform should be enough to activate an autonomous function of *Lost-Link* event management. The level of the data-link signal is measured by the NAV and in case of value below a prescribed threshold the *Lost-Link* navigation mode is activated.

The concept is to leave temporally the main route and wait for the link recovery. The SANAG 4586 prescribe the use of Contingency Routes, loaded with the main route, to be engaged in case of needs as for example the lost link. The STANAG 4586 approach is to assign two Contingency WPs to each WP using the proper fields in the AV Position Waypoint frame (#802 Fig 36). This means that each WP of the route can have two Contingency WPs each one bringing to other WPs having their own Contingency WPs and so on. The result is a very wide range of potential routes to be used and concatenate as in fact the Contingency WPs can bring also to main route WPs or to other Contingency routes WPs. An example of STANAG 4586 Contingency Routes concept is shown by the STANAG 4586 itself in a table presenting the WPs number, the Destination WP, and the Contingency WP. The table present only Contingency WP '*A*', but the same can be done for Contingency WP '*B*'. In Fig 66 the table has been shown underlying in green the connection between the WPs of the main route and in red the connection between the WPs of the Contingency Routes.



Waypoint #1
Next Waypoint #2
Contingency Waypoint #5

Waypoint #2
Next Waypoint #3

Waypoint #3
Next Waypoint #4

Waypoint #4
Next Waypoint #5

Waypoint #5
Next Waypoint #6
Contingency Waypoint #200

Waypoint #6
Next Waypoint #200
Contingency Waypoint #200

Waypoint #7
Next Waypoint #200
Contingency Waypoint #200

Waypoint #200
Next Waypoint #0

Fig 66 STANAG 4586 Contingency Routes example [14].

The routes resulting from this example are shown in Fig 67.

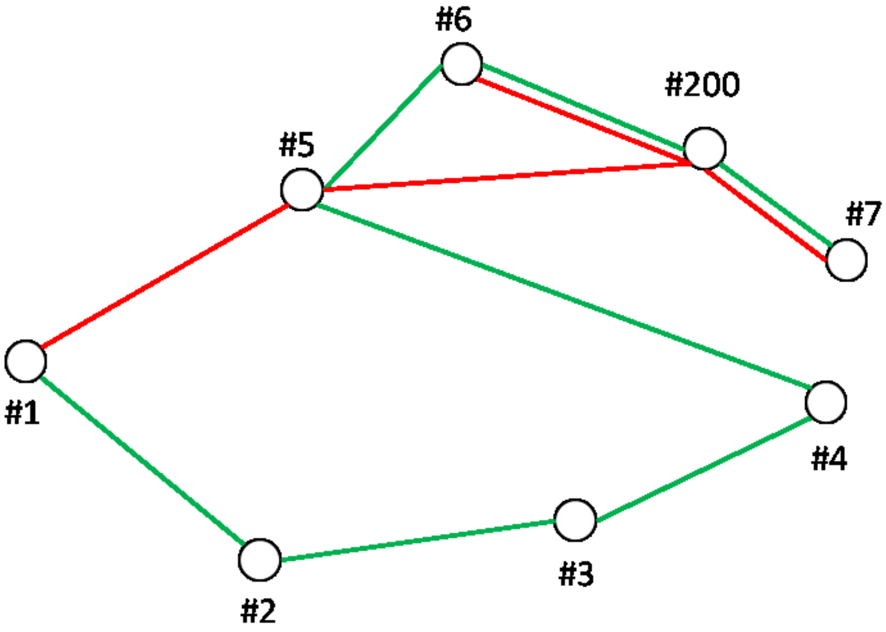
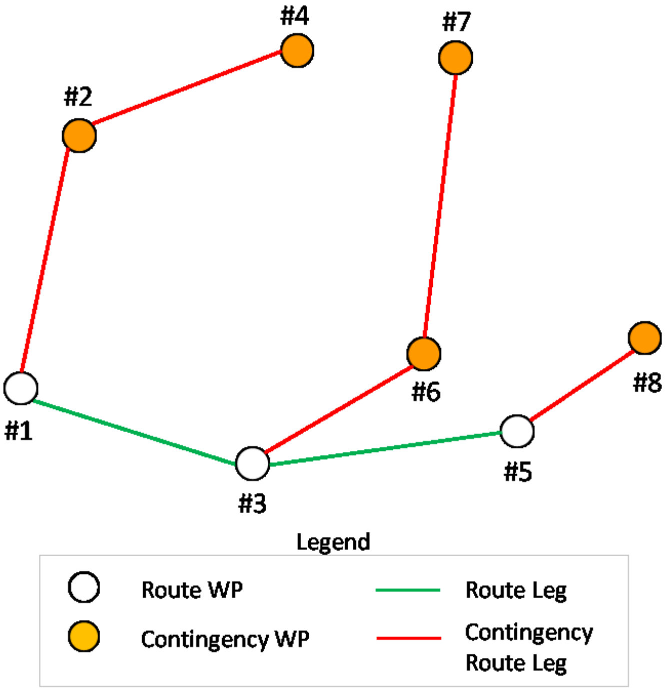
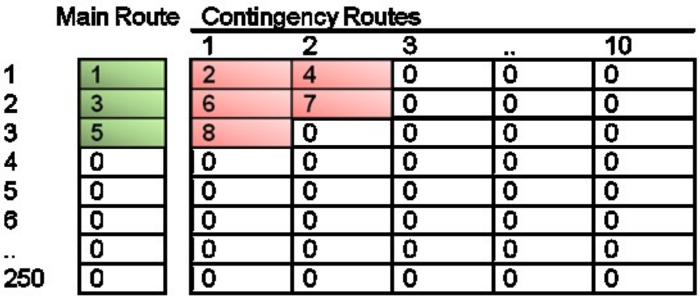


Fig 67 STANAG 4586 Contingency Routes representation.

For the Sky-Y Lost-Link management has been used just a sub-case of the wide possibilities offered by the STANAG 4586 in order to have a more evident separation between the main route and the contingency routes and only the Contingency WP '*A*' has been considered. In the planning phase each main route WP is provided with a Contingency Route formed only by Contingency WPs (*Fig* 68).





	Main Route	Contingency Routes				
		1	2	3	..	10
1	1	2	4	0	0	0
2	3	6	7	0	0	0
3	5	8	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
..	0	0	0	0	0	0
250	0	0	0	0	0	0

Fig 68 Sky-Y Contingency Routes.

So that at any point of the route the UAV would lost the data-link, the OBMC would autonomously activate the Lost-Link state: the NAV switch immediately to the Lost-Link Navigation Mode providing a full 4D UAV control, while the Steering partition re-plans the flight path linking the destination WP to the relative Contingency Route WPs and stores the main route in a local variable to be used in case of link recovery. The last WP of each Contingency Route is intended to be a loiter. If not a Steering function is called to set it as circular loiter with default loiter parameters and the same function set to infinite the loiter time attribute to each last Contingency route WP. This management of The Contingency Routes resulted to be very useful for bringing the UAV over a safety area thanks to the possibility of planning the best flight path to be adopted at any point of the route. That is an improvement of safety respect to a traditional lost link management made of only one or few contingency WPs for the entire route and helps also in case of flight test in a civil aerospace granting the no fly zone avoidance also in case of lost-link (Fig 69).

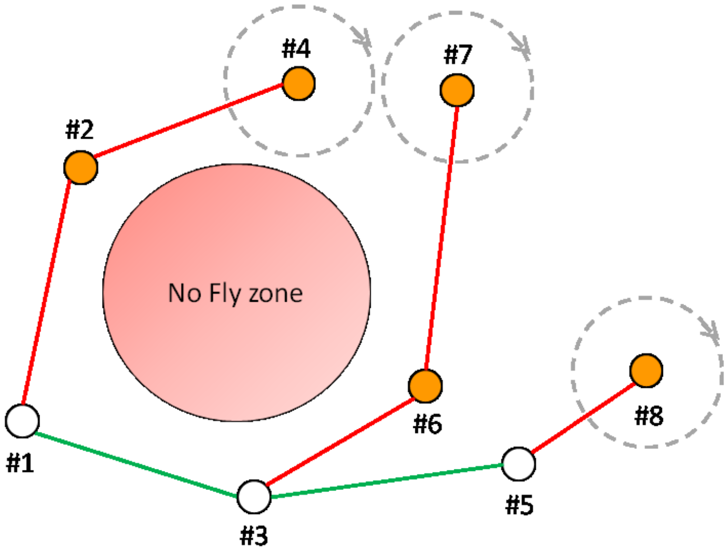


Fig 69 Contingency Routes avoiding no fly zones.

In case of link recovery the pilot has the possibility of recover the control. In that case the proper Steering function re-plan the flight path inverting the WPs sequence in order to navigate back the same Contingency Route and return on the WP of the route at which the main route had been abandoned. At that point the Steering re-plan the main route to continue the mission (Fig 70).

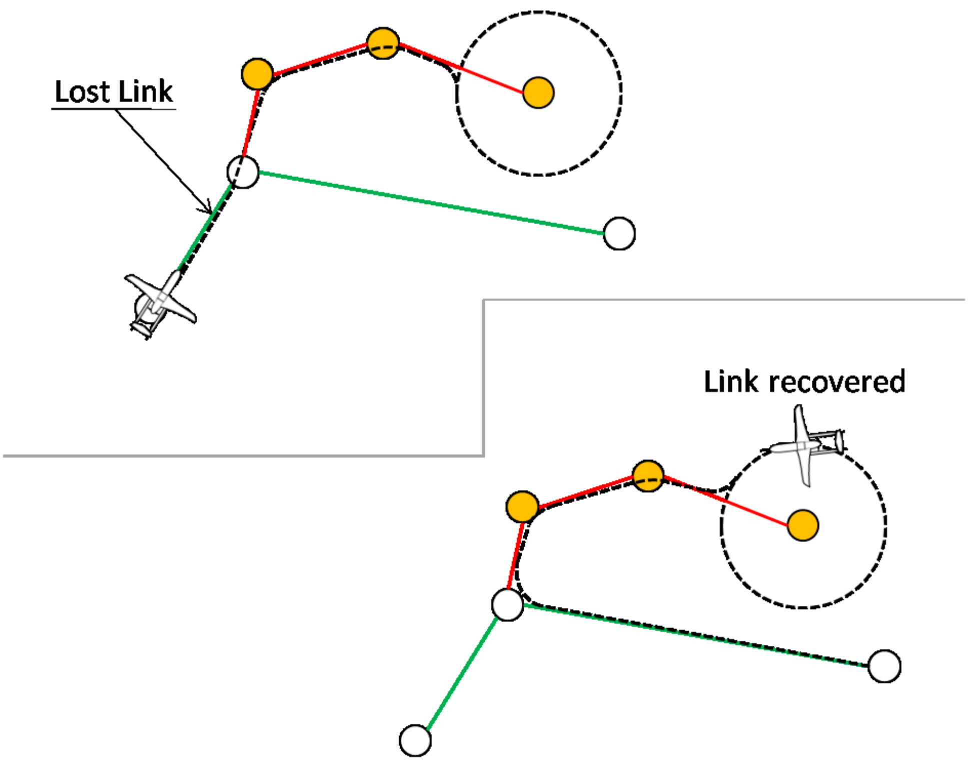


Fig 70 Lost Link and Link Recovery flight path.

In case of permanent lost of the link, after a certain loiter time on the last Contingency route WP, the termination process is engaged.

9.3.3.3 Slave to sensor navigation

UASs conceived for monitoring missions have as primary goal the images acquisition respect to the flight path. These systems have their core in the Mission System (the equipments used to accomplish the mission goals: the payloads), because of this a new navigation mode has been developed and introduced for modifying the concept of conducting the UAV. The most common sensors installed on medium size UAVs are the electro-opticals, over the RADARs and the Hyperspectrals. The Slave to Sensor Navigation defines a set of different navigation and sensor operational modes designed to slave the aircraft flight to the sensor objectives. This includes fully flight enslavement or momentary diversions from preplanned routes.

The Slave to Sensor Navigation is conceived for searching or monitoring tasks in zones where the vehicle is not constrained to fly in assigned corridors. Operatively this mode is typically used for a free search in the operational area with the vehicle slaved to the sensor observed point, or as diversion from the planned monitoring route if an opportunity target is detected. It is therefore suitable both for ground and maritime operations. Examples could be a diversion from a planned oversea ladder patter if a ship is detected, or the observation of a fire in order to coordinate the firefighting ground team operations. According to the previous examples, this mode is utilized especially for opportunity targets, that are in many case of punctual type and so can be observed with different sensor mode (e.g. manual, ground stabilized, inertial pointing, etc.). In any case, the pilot have the possibility to specify the UAV range and bearing with respect to the sensor aiming point to permit the use of this mode also in presence of opportunity targets placed in No Fly Zones (e.g. out of national borders).

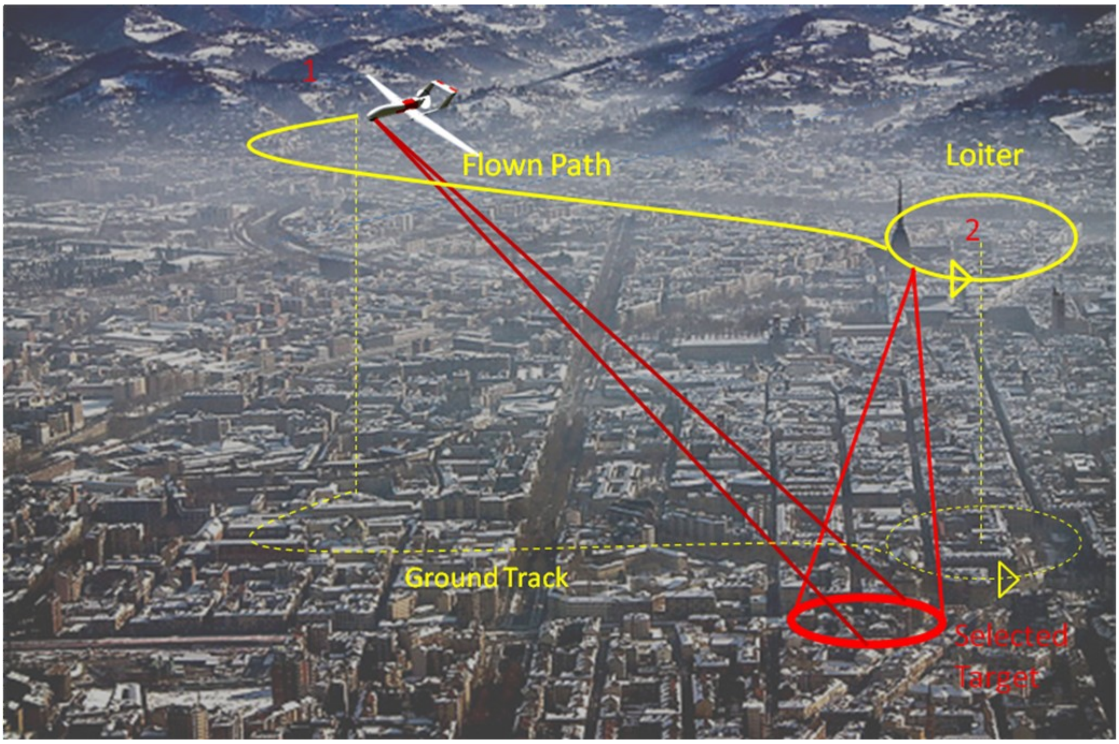


Fig 71 Slave To Sensor navigation mode: static target observation example.

Static target

After having identified a point of interest and locked the sensor to the target, the operator can command the engagement of the Slave to Sensor navigation mode Fig 71 position 1. The effect of the Slave To Sensor navigation is that the aircraft will move to a position of better observation of the target Fig 71 position 2. In particular when the target is distant from the aircraft, the aircraft will fly a straight line from the aircraft current position to a definable loiter point close to the target. This loiter point (over the usual loiter attribute) is defined by providing bearing and distance information from the target. The aircraft will loiter on the specified point until a new command is issued. If the previous navigation mode was a route navigation mode the Steering partition stores the route with the purpose of re-plan it at the end of the Steer To Sensor mode if required by the pilot. in particular the Steer to Sensor function, which is the Steering function providing this mode, places a WP having the coordinate position of the UAV at the time of the Slave To Sensor engagement in order to re-acquire the previous route exactly at the point from which it had been left.

Moving target

The principle of flying to a better observation point that is located at a defined distance and bearing from the target is still applicable also in case of a moving target. In this case the target location changes continuously and the position of the loiter changes accordingly.

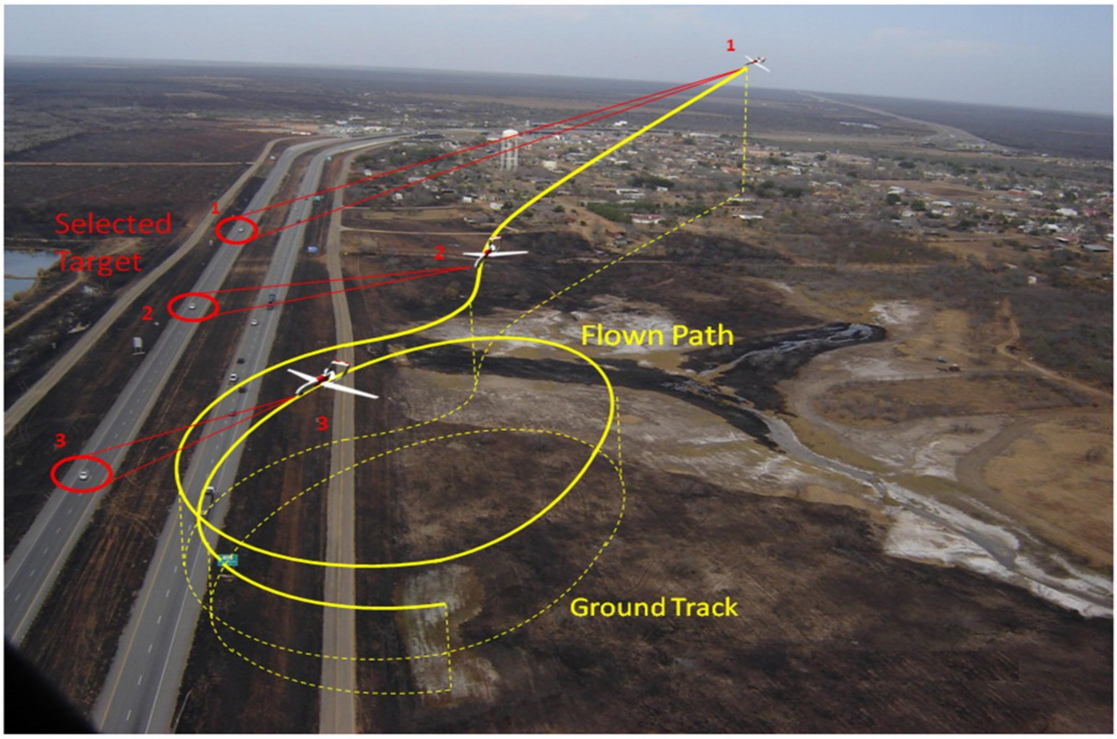


Fig 72 Slave To Sensor navigation mode: moving target observation example.

In Fig 72 position 1 the target is identified and selected. The UAV flies toward the loiter position identified by a given bearing and distance from the moving target. The aircraft will continuously change its destination. Because the aircraft speed is higher than the target speed, the aircraft will eventually reach a loiter position Fig 72 position 2 and enter a loiter. The centre of the loiter is moving with the target, therefore the aircraft will fly a circular path with a continuously changing centre Fig 72 position 3.

9.3.4 STANAG 4586 impact in navigation design

The STANAG 4586 adoption impacted many aspects of the navigation design due to the different approach respect to the traditional navigation imposed by the STANAG 4586 messages structure.

One of the aspects mainly impacted is the route loading procedure Previously the route definition was a separate message respect to the WPs attributes and was just a list of WPs ID ordered in the sequence of flight. With the STANAG 4586 the WPs data messages contains also the route definition (see section 9.3.1.2). Therefore a specific software algorithm had to be generated and allocated in the board segment (a NAV function) allowing the route arrangement in order to make the STANAG 4586 route loading frame management transparent to the Steering partition. That represented one of the main impact of the STANAG 4586 in the board segment respect to a traditional approach, but the solution figured out just by adding a new specific function so that the rest of the navigation resulted to be not impacted. A similar algorithm was implemented in the ground segment for showing the route to the pilots on the CS navigation screens.

The loiter management is another point that impacted the on board navigation design specially about the Racetrack and Figure8 that required a specific Steering function implementation for the pattern generation and a specific flight mode introduction for the path covering. Beside, respect to the traditional approach, the STANAG 4586 don't supply speed and altitude attributes of the loiter path different from the attribute of the approaching leg, so that represented a little simplification in the Steering design. Moreover the exit condition for the loiters is just the loiter time without an exit point and this requires more accuracy during the planning phase (see section 9.3.2.1.2).

A great reduction in the navigation functions design instead was the speed determination based on the *Arrival Time*. The impossibility of having just one *Arrival Time* WP for more than one leg (see section 9.3.2.3) reduced the complexity of the algorithm calculating the speed because the path to cover prevision over all the legs before the Arrival WP was not required anymore. Such a function, for its iterative nature due to the path forecasting (Roll In Points and Overshoots) according to the actual speed (that is also the output), resulted to be, very heavy from a computational point of view. The STANAG 4586 adoption avoid this problem on board by charging the ground segment of a more accurate planning phase.

Impacted area	**Traditional approach**	**STANAG 4586 modification**	**improvement**
Route Loading	WPs data separate from the Route definition messages	Route definition enclosed in WPs data	enhanced route data loading flexibility
Loiter management	Circular loiter only; Different loiter speed and altitude attribute	Racetrack and Figure8 loiter added; Same speed and altitude of the leg	alternative loiter pattern introduced
Arrival Time speed determination	Admitted one Arrival Time for more legs	Speed or Arrival Time attribute mandatory for each WP	More accuracy in the planning phase
Lost Link management	Two loiter WPs for all the route	A Contingency Route for each WPs of the main route	New functionality supported (contingency routes)

Fig 73 Summary of the STANAG 4586 impact on the navigation design respect to a traditional approach [17].

About the Lost Link Contingency management the adoption of this STANAG 4586 functionality impacted both the board and ground segment in the same way of the route loading function due to the STANAG 4586 way of conceive of WPs messages for the routes definition. Moreover the board Steering segment was asked to perform the in-flight auto re-planning during the Lost-Link and link-recovery phases which comported the development of new functionalities compliant to the STANAG 4586 requirement. The result was a great improvement of the UAV safety capability respect to the previous Lost Link management made of just two possible loitering WP for all the route.

It is important to note that sometimes during the STANAG 4586 implementation some limitation of the prescribed frames or in their management came out, in many cases that was due to the early application of the standardization. The navigation function impacted by the STANAG 4586 discussed in this section instead cannot be considered effective limitations, but just a different approach respect to the traditional implementation. In certain cases the solution has been to introduce navigation design changes in order to arrange data and message management to the STANAG 4586 requirement. That was the case of the route loading procedure and the Contingency Routes management. In other cases solution has been to increase the planning phase accuracy, as for example the loiter exit point and the Arrival Time. The only real limitation found out during the implementation was the route loading consistency problem about the loiter WP messages. As seen in section 9.3.1.2 the loiter attributes for a loiter WP are provided by an additional STANAG 4586 message (#803) separate respect to the WP position attributes. The link between the two messages is only the WP number present in the loiter data message that corresponds to the WP number of the respective WP to be set as loiter. Unfortunately is not present any field in the WP position data message advising the route building algorithm to wait or not for a loiter data message linked to it. As is not prescriber a forced sequence of loading messages or a message of end loading, the on board algorithm should potentially wait forever a possible loiter data message. Also if a sequence is forced, as for example imposing that the loiter data messages should always follow the respective WP position data, the algorithm will anyway wait for the possible last WP loiter data message. Fortunately a solution for this problem is planned in the proposed next edition of the STANAG 4586 [16], where a significant improvement will be made by covering feedbacks originated in these early applications of the Standardization. About this problem an additional field is imposed in the WP position data message advising if that WP is followed by additional data messages or not.

The last consideration is that the STANAG 4586 allows up to 400 Private messages (16000 in the future third edition [16]) customizable by the user in order to transfer specific information not prescribed by the standardization. Such messages result to be very useful in case of functions specific for the own platform, but also for cover eventual STANAG 4586 misleading.

9.4 Simulation in Steering development and RIG tests

The development of the Steering functions take different steps with large amount of simulations. In the first phase the functional requirements have been outlined, then the main work has been the translation of these high level requirements into more specific software requirements. The functional requirements have been split into separated functional blocks providing a different action, but cooperating each others for the overall Steering output determination. This second phase required two steps: first of all the test of each single function algorithms, then the test of the interaction between all the functions. As many variables are used and modified by many functions, each single function should be tuned within the order of execution of all the other functions. Because of this the use of Matlab® resulted to be a great effort in the single function development and test, and also for tuning the right interaction between them.

9.4.1 Matlab® model

A Matlab® model of each Steering function was so compiled according to the requirements and used for testing the algorithms to evaluate and eventually correct the requirements itself. This iterative process occurred also during the second step, where all the functions have been inscribed into a overall Steering simulation always implemented in Matlab®. To test the overall Steering model in a more accurate and efficient way in fact, a dynamic iteration process was required. For this reason in the Matlab® model was introduces also a basic AV simulator to generate coherent output to give in input to the Steering model (Fig 74). Such a simulator, in a very simplified way, incorporated also the role of the FCS by reading the Steering model output and generating the correction to the AV position, attitude and speed (Best data of Fig 74). The Best data, in addition to the TCS data (route commands), alimented the Steering model which, at each iteration, provided the route data (just for display purpose) and the output to FCS, used by the AV model to update the Best data in an iterative loop. This process granted the evaluation of all the Steering packet in a dynamic way with a complete set of cases by simulating the UAV behavior.

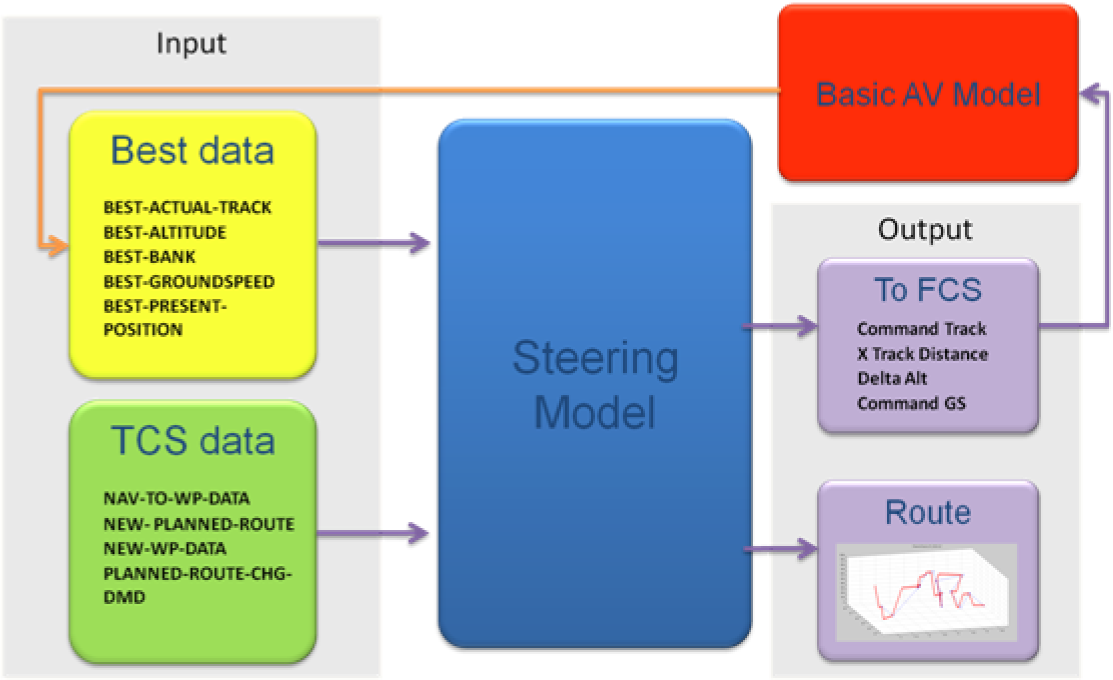


Fig 74 Matlab® Steering Model.

This Phase outlined many particular cases in which the Steering functions resulted to cooperate in a wrong way and was so a very easy and powerful method for improving the software requirements. At the end of this phase the Steering functions and their calling sequence were consolidated.

In Fig 75 and Fig 76 is possible to see the result of a Matlab® simulation in which the blue lines are the route legs, the full blue dots are the fly-through WPs, the blue rimmed dots are the fly-by WPs and the red line is the UAV simulated trajectory.

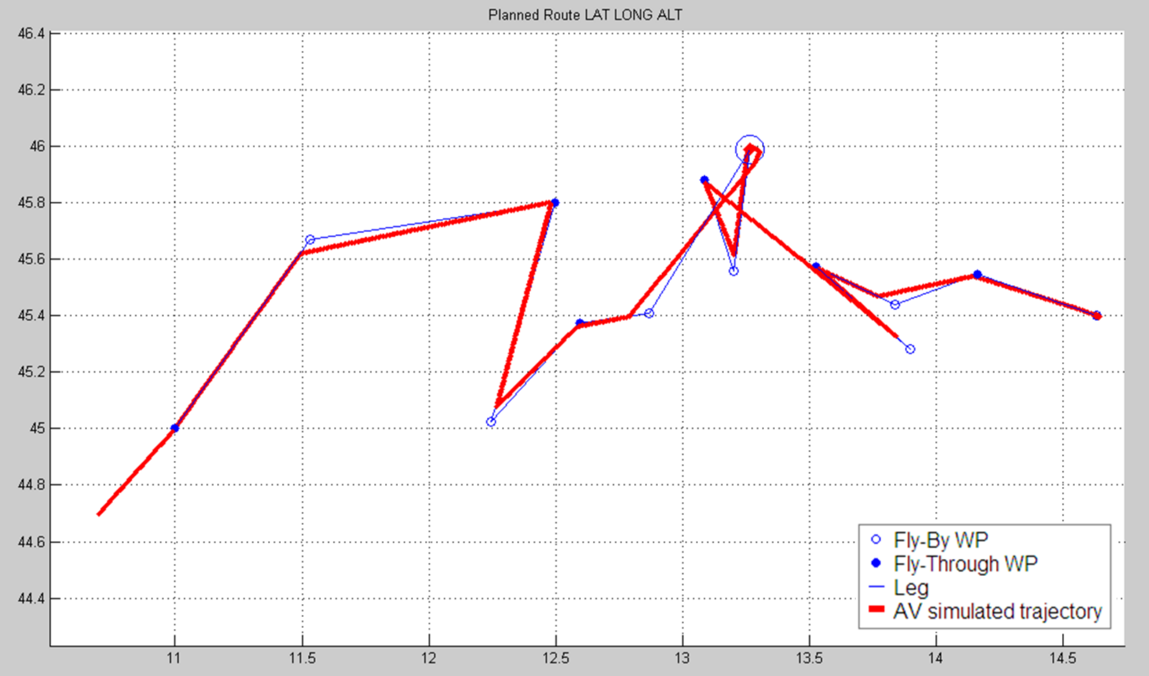


Fig 75 Matlab® model Steering simulation: 2D view of a route navigation.

In the 3D graph (Fig 76) it is possible to note that the altitude is reach with a constant ramp independent from the inclination of the legs, exactly as the real FCS correction. While in the 2D view (Fig 75) is evident that the curves are simulated as a sudden change of direction without transitions, anyway it is possible to evaluate that the fly-through WPs are over-flown while the fly-by are acquired at a certain distance (roll-in-distance). Always from the 2D graph is possible to see that, mainly after the fly-by WPs, the legs are not acquired by the UAV simulated trajectory. This is not due to the *Current Leg Section* value = '*Direct*', but it is due to the basic AV simulator FCS model that does not correct the *X-track* errors. For the software requirements generation this order of approximation was enough, but for going on with more accurate tests an improvement in the UAV and FCS simulation appeared desirable.

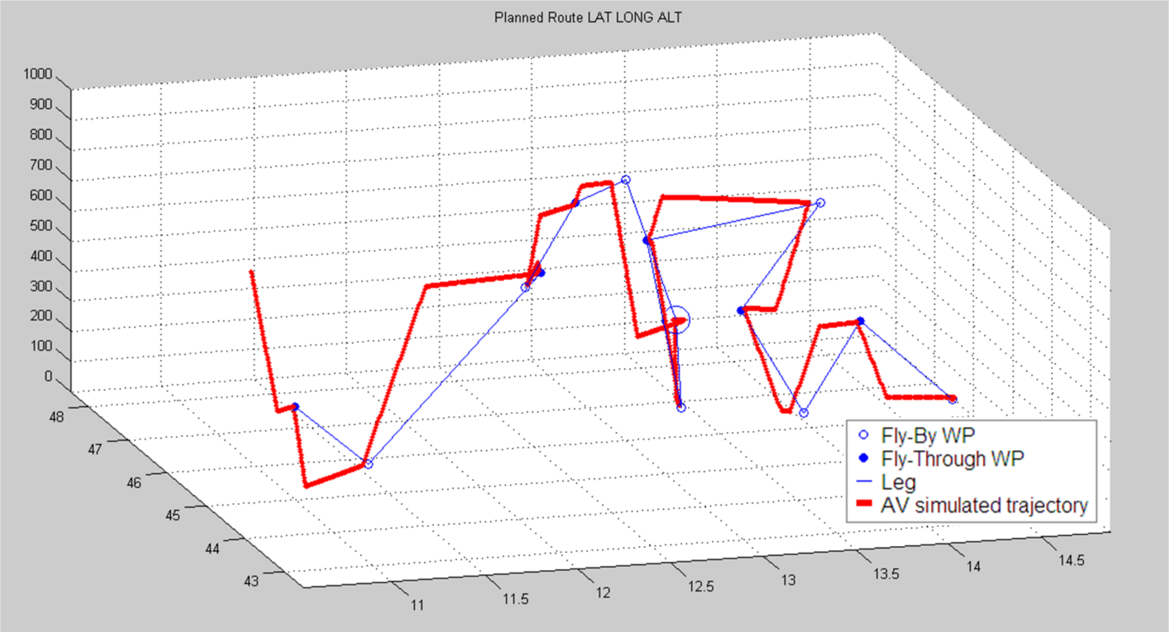


Fig 76 Matlab® model Steering simulation: 3D view of a route navigation.

For this reason the following step has been the implementation of the Steering model in Simulink® in order to have a more precise simulation model of the aircraft and FCS, and so evaluate the UAV behavior along the route.

9.4.2 Simulink® model

The Simulink® model (see Appendix B) was realized for generating a real time simulator in which the Steering partition would be interfaced with a more realistic flight dynamic AV model. Also the FCS model was improved to take into account also the *X-Track*. Six main blocks have been used for the Simulink® model (Fig 77).

The UAV Model (red block of Fig 77) is the AeroSim/Complete Aircraft 6 degree of freedom model block. It is used to simulate the motion of a standard aircraft by taking in input the controls commands and providing in output the states of the AV.

The Flight Gear® Interface (gray block of Fig 77) was introduced to use Fly Gear® Flight Simulator as a graphic interface to the Simulink® model simulation. This results useful for having a qualitative evaluation of the AV motion during the route navigation (Fig 78).

The TCS Command (light green block of Fig 77) is the model of the TCS main command for the Steering. It provides the WPs data and the route.

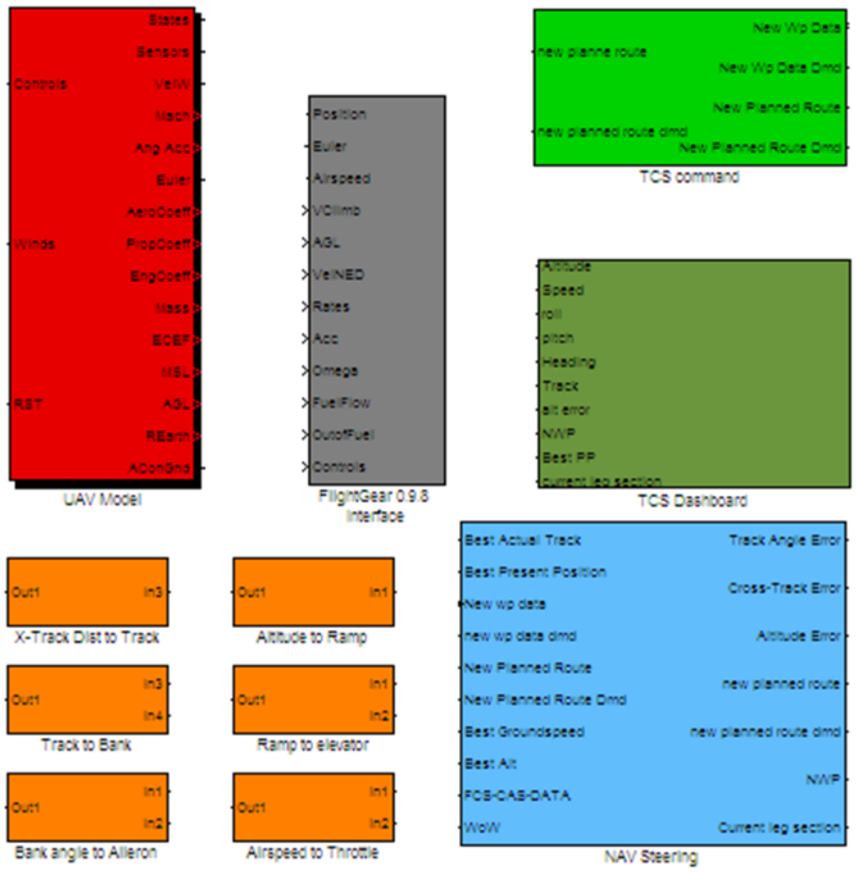


Fig 77 Simulink® Model main blocks.

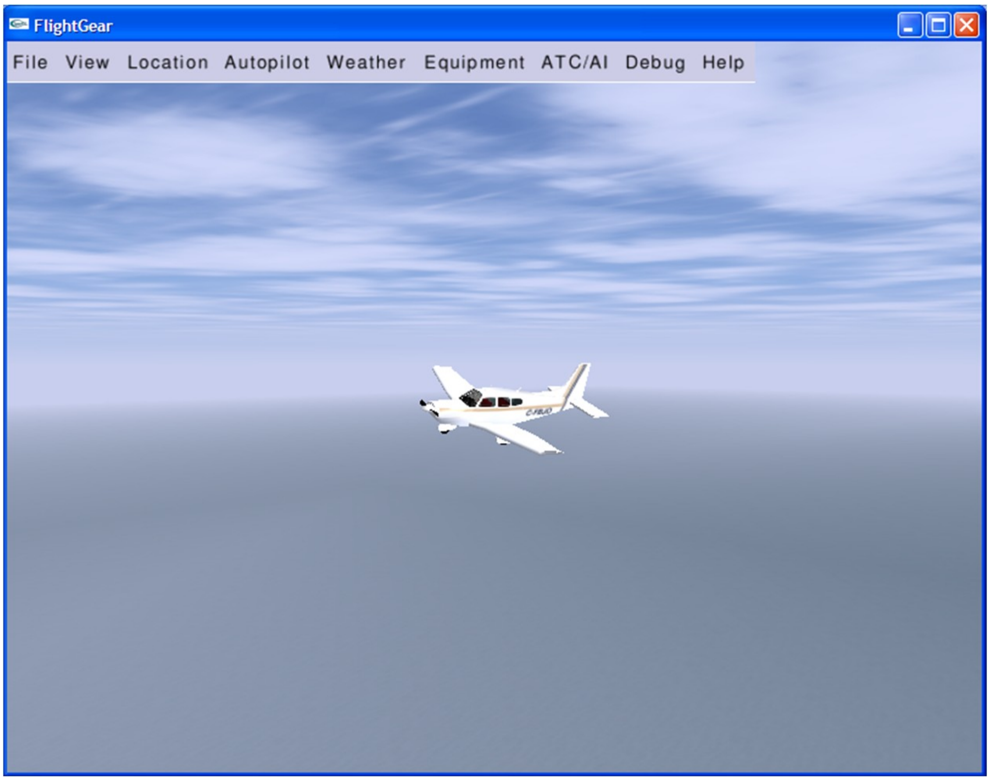


Fig 78 Flight Gear® Flight Simulator graphic interface to the Simulink® model.

The TCS Dashboard (dark green block of Fig 77) is the block used for the main navigation data visualization as shown in Fig 79. By using the Simulink® Gauges Blockset/ActiveX Library was possible to reproduce a real aircraft dashboard with the altimeter, the artificial horizon, the airspeed indicator, the heading indicator, and the Track indicator. Moreover additional information useful for the navigation are displayed like the destination WP coordinates and altitude, the AV position, the altitude error and the current-leg-section indication. Then also the simulation time has been displayed to compare the AV performances with the simulation time that result to be different respect to the real time (usually slower due to the high computational workload).

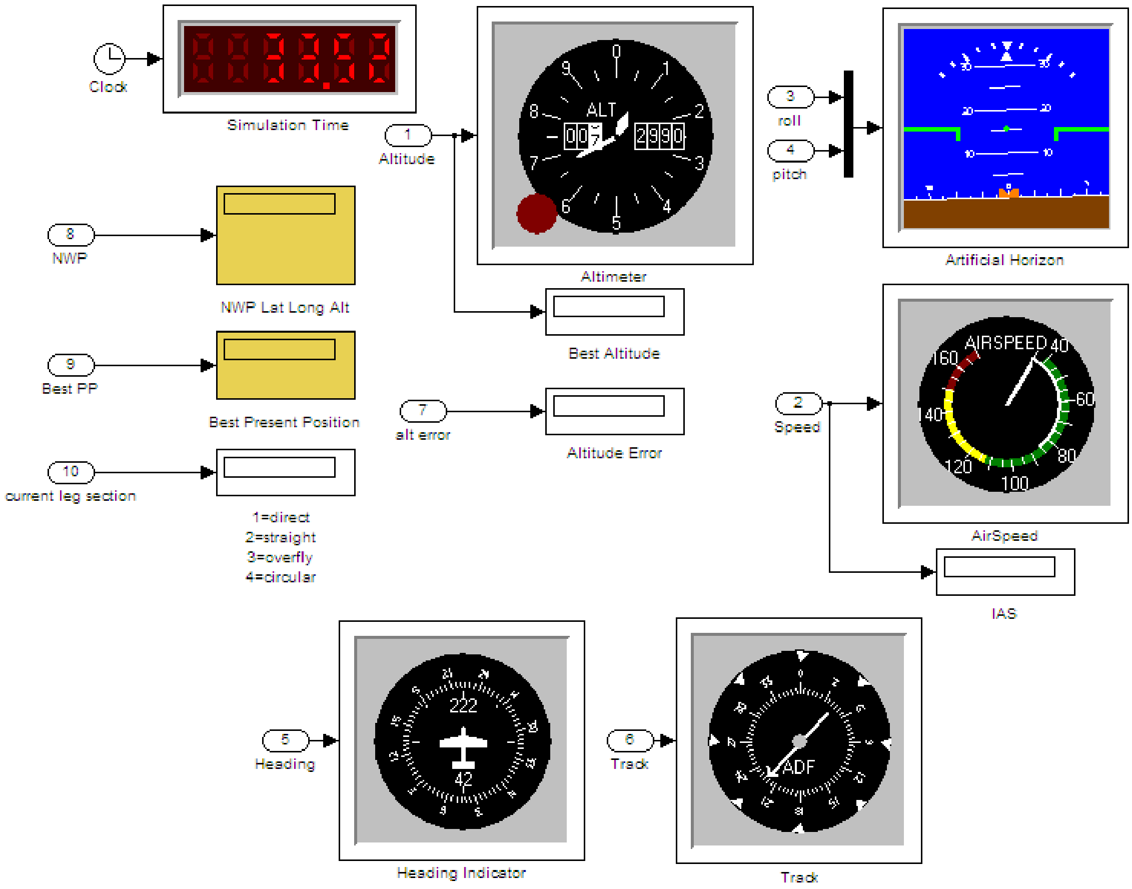


Fig 79 TCS Dashboard Simulink® block contents.

The FCS blocks (the 6 orange blocks of Fig 77) are PID controllers used to convert the Steering output into corrections to the AV surfaces and throttle. One block is used for the throttle control according to the required speed. Two blocks are used for the longitudinal control, one to determinate the ramp according to the altitude error, the other for the elevator deflection according to the ramp. Three blocks are used for the latero-directional control, one to determinate the track correction to annul the x-track-error, one to merge the previous output with the planned track and determinate in base to the AV track the bank to reach, the last one to convert the required bank into the aileron commands.

The NAV Steering block (the blue block of Fig 77) is the module in which the basic Steering functions have been implemented.

The Simulink® model was developed to have a real time simulation of the Steering functions alimented by an AV dynamic model. Some tests have been performed with this tool, but soon became possible to use the Alenia Aermacchi Sky-Y flight simulator which represented the best solution for conducing accurate tests. Anyway as the Sky-Y simulator works with the algorithm codified in C from the same software requirements, the use of Simulink® and Matlab® models occurred also in some successive cases where the comparison between the C code and the Matlab® or Simulink® one was required for debugging purposes.

9.4.3 Sky-Y Flight Simulator

The Sky-Y flight simulator is the main facility used for the functionality tests and for the advanced steering functions development. It is formed by a real TCS and a simulated board segment where the AV model has the Sky-Y dynamic model and the simulated OBMC has the real NAV and Steering software with the interfaces simulated (Fig 80).

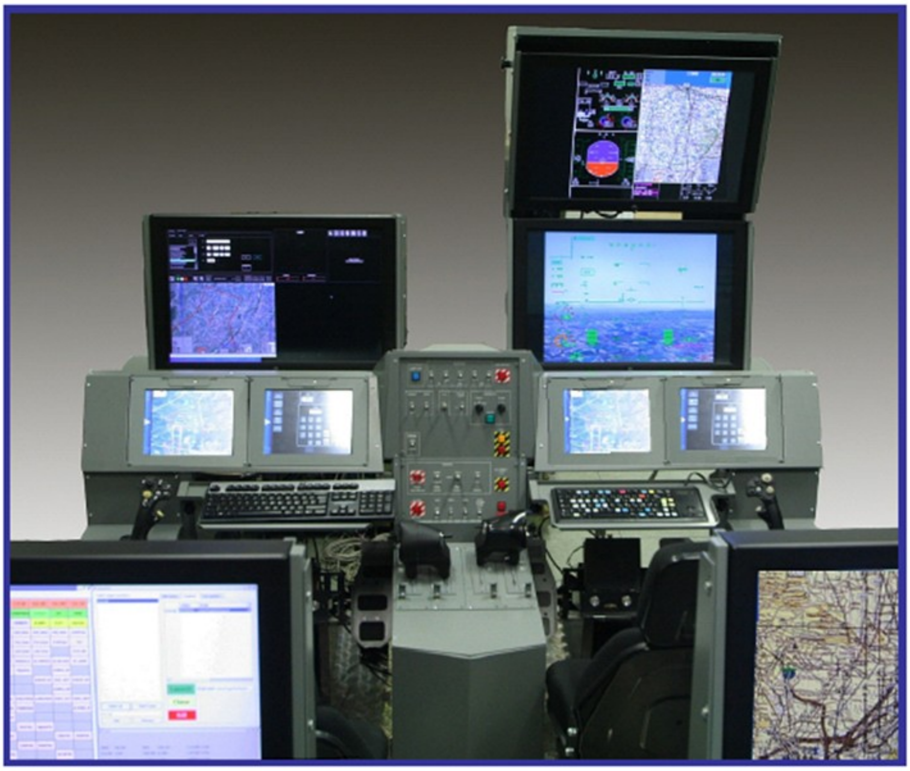


Fig 80 Sky-Y flight simulator.

In this way was possible to include the steering code inside the complete loop evaluating the interaction between the TCS, the NAV, the FCS, the simulated sensors and UAV. In this ambient was also possible to test for the first time the STANAG 4586 messages management taking into consideration also the synchronization of the frame exchanged between the ground segment and the board segment, and the sequence of transmission of certain messages packets, over the NAV route assembling and data conversion to the Steering partition.

The use of the Sky-Y simulation model, coupled with the real FCS control laws, permitted a fine tuning of the Steering flight parameters as the maximum and minimum roll-in-distance, the maximum bank angle, the WPs acquisition tolerance etc.

Adopting the C codes of the Steering and NAV, that are the same codes used in the real OBMC, was also possible to directly compare the behavior of the simulated board partition respect to the real one at the RIGs.

9.4.4 RIGs tests

The NAV and Steering software have been tested on target at the RIGs in two steps. First the OBMC stand-alone tests have been performed at the Software Bench RIG in which the navigation algorithms have been alimented by static inputs to validate the single states of the software. In this phase the comparison between the Matlab® model resulted to be very useful for the debug. Then occurred the OBMC integration with the others systems at the Sky-Y RIG in which the real equipments are linked together and alimented with a flight simulator providing the Sky-Y UAV flight dynamic and simulating the sensors data. In this phase the comparison between the behavior verified at the RIG and the Sky-Y flight simulator behavior permitted to outline some integration problems. At the end the formal tests done at the Sky-Y RIG brought to the formal validation of the codes for the flight tests.

9.5 Hand-over experimentation

As discussed in section 7 the hand-over is a considerable issue for a UAS. In a operative mission the hand-over can be necessary in case of data-link operative range limitation so that a different CS is asked to take the UAV command in order to increase the total controllability range. A typical example a BLOS mission in which the take off is made from a CS in LOS control, then the hand-over occurs for the UAV transfer to the operative scenario in BLOS control, and finally another hand-over may be required for the mission control from a LOS CS located near the area of operation.

The issues of this operation are represented by the overlapping of the two different CS control for the command authority transfer from the first CS to the second. In this critical phase the UAV control should be always possible from one of the two CS, otherwise the flight safety results compromised, but should also be avoided the possibility of double control from both the CS to avoid contrasting information to the UAV. The STANAG 4586 compliance, prescribing the standards for the communication messages, increase the interoperability between UASs and so also the capability of switching the UAV control from a CS to another is prescribed. For this aim the message #700 is used in the STANAG 4586.

During the Sky-Y flight tests an hand-over experimentation occurred for the safety test of the new FCS and TCS. The test had not the scope of providing the UAS of a hand-over capability as described above, but was used for testing the new control line with the safety support of the old consolidated one, anyway the problematic analyzed and the solution find out resulted to be the same, with just some differences, respect to the CS to Cs hand-over case. In particular the tested hand-over, for an architectural reason (the use of two separate data-link systems), resulted to be a significant test for a LOS to BLOS (or vice versa) hand-over. Let start analyzing the test procedure as proved in flight for concluding with the LOS to BLOS analogy considerations.

The test architectural configuration is the one described in section 9.1 Fig 32. The consolidated command line is made by the GCS with a single D/L directly connected to the old mission computer. In this mode the old mission computer is asked to provide the UAV control with its own control laws, while the TCS command line, the OBMC and the EFCS are in stand-by Fig 81.

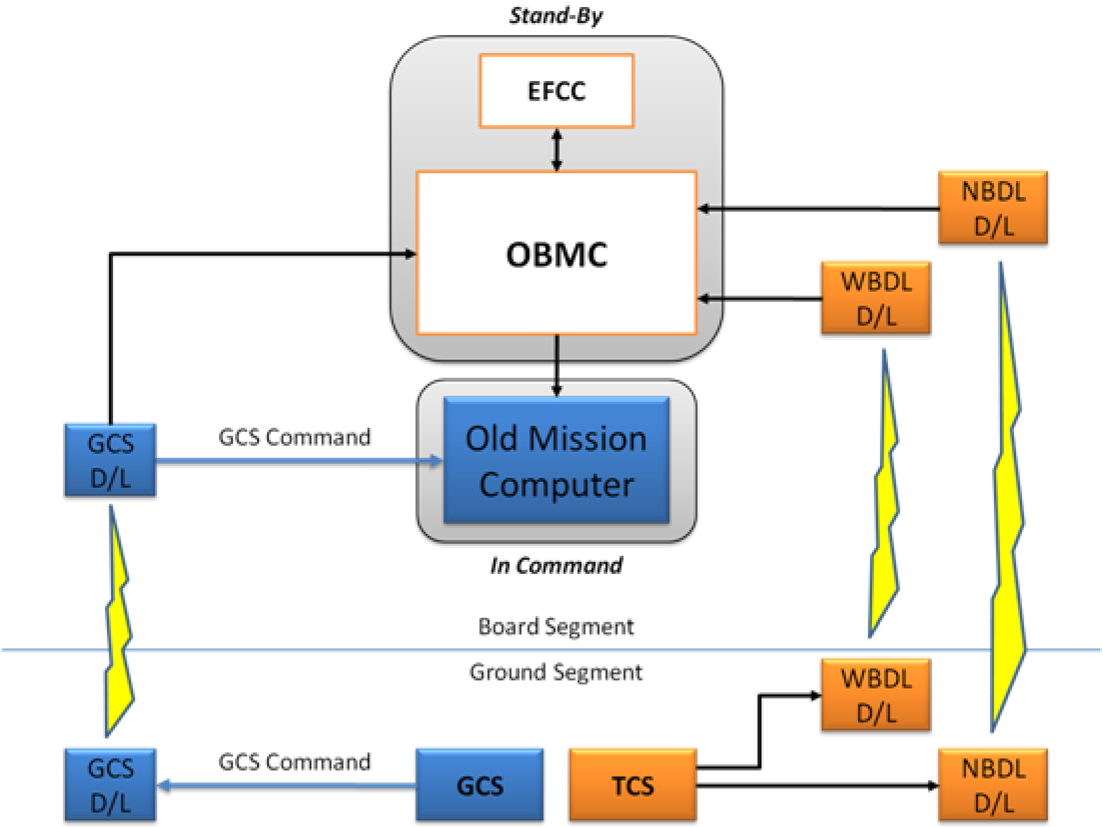


Fig 81 ROS GCS in command, OBMC in idle.

The hand-over, occurred for testing the OBMC, the EFCS and the TCS command line, was made of three steps: the first for hand-over request and data-link FOM evaluation, the second for the OBMC enter in the loop, and the third for the EFCS enter in control.

The first phase starts the hand-over with the request to the OBMC (Request Command of Fig 82). The OBMC process the request calculating and evaluating the Uplink FOM of all the data-links, but the control is still bypassing the OBMC.

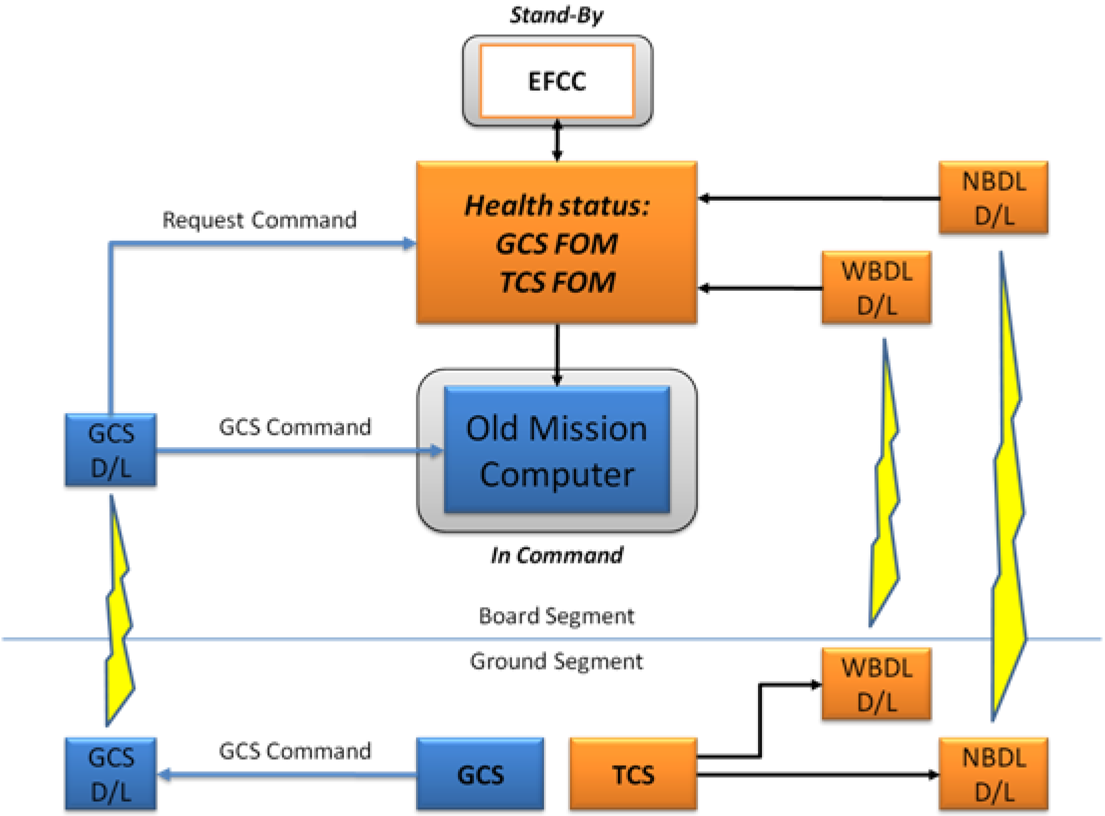


Fig 82 Hand-over first phase: OBMC interrogation and FOM calculation.

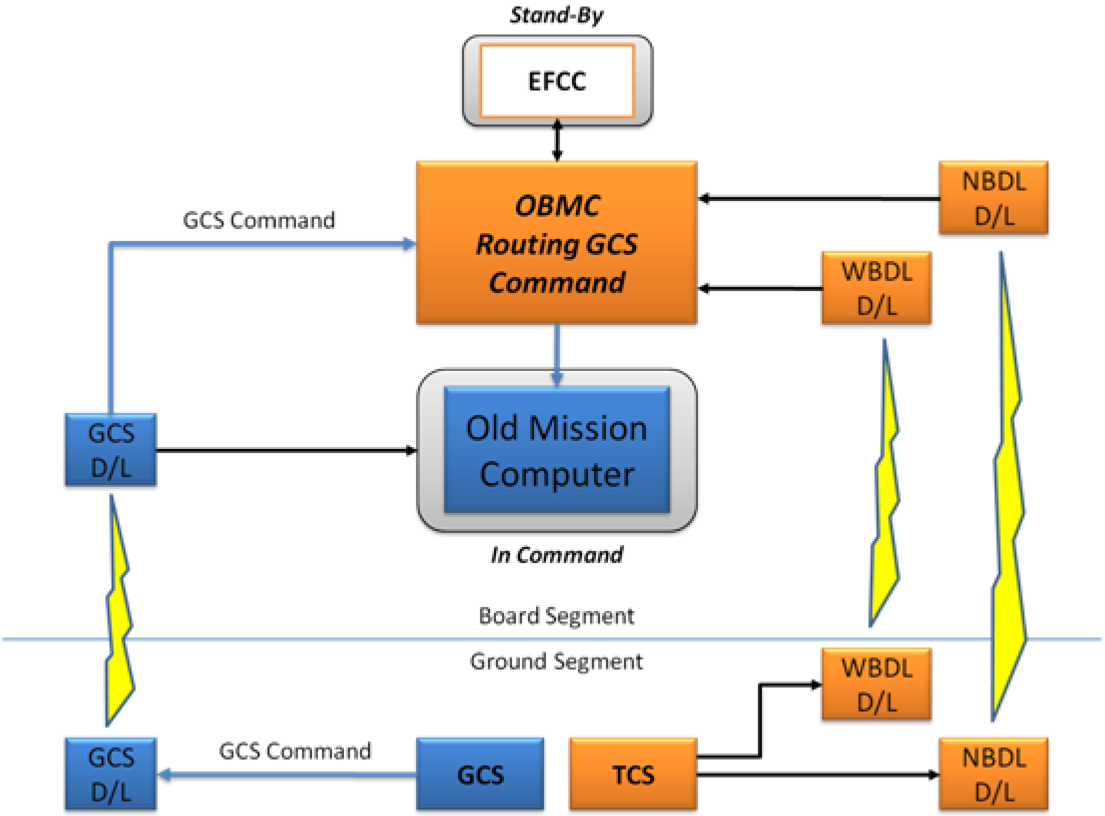


Fig 83 Hand-over second phase: OBMC in the loop.

In the second phase, if the FOM resulted valid, the OBMC takes the UAV control. In this phase the control is still from the GCS but the OBMC enters in the loop (Fig 83).

In the last phase the EFCS is engaged (Fig 84).

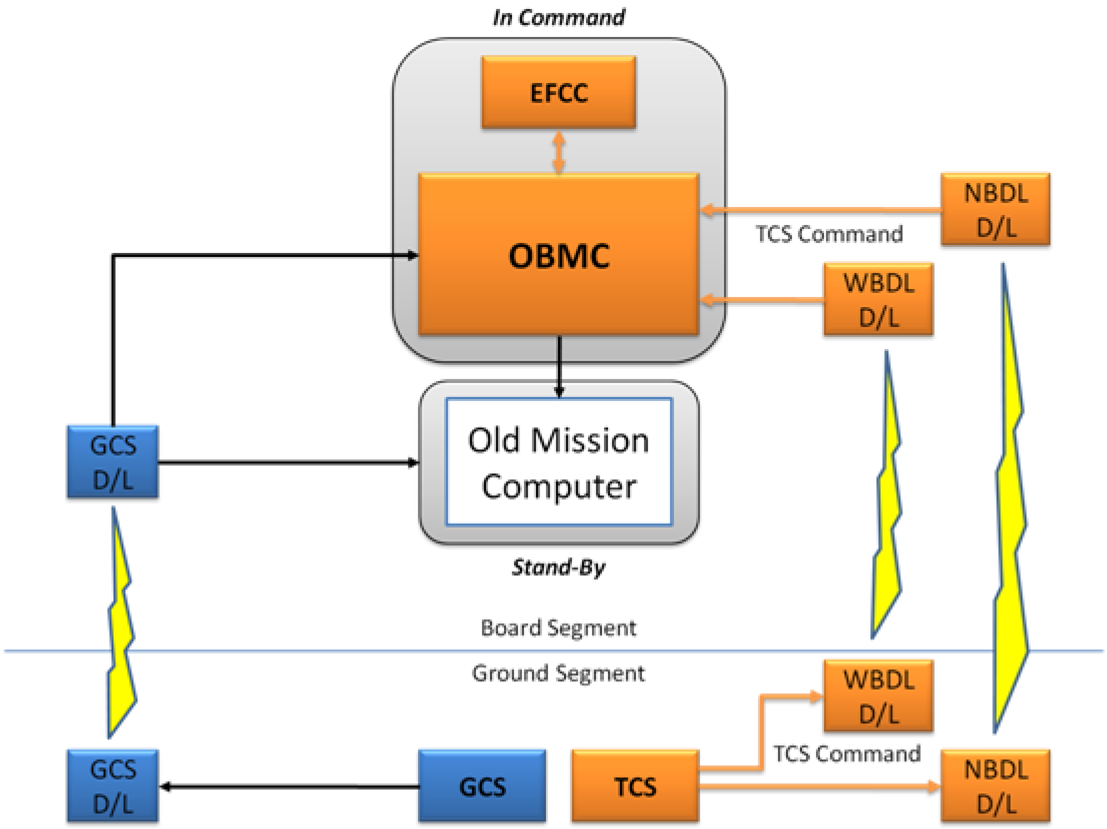


Fig 84 Hand-over third phase: EFCS control.

This mechanism, studied for the new on-board systems and STANAG 4586 TCS testing, presents some analogies with the hand-over between a LOS and a BLOS CSs, not just for the data-link as both the lines used are in LOS, but because of the presence of two separate data-link modules. Excluding first phase and the alternant use of two FCSs (the EFCC and the one integrated in the old mission computer), fact that represent an increase of complexity, the second and third phases described above can be a valid simulation of such hand-over. Imagining the GCS in LOS and the TCS in BLOS (or vice versa), the OBMC is demanded to manage the hand-over request from the first CS by analyzing the data-link FOM and switch the command from the first CS to the second without lack or overlapping. The flight test demonstrated the good reliability of this design for the proposed test. For a possible future application in a LOS to BLOS hand-over this logic experiment should be take into consideration as a starting set for developing the functionality.

9.6 Data-Link LOS and BLOS requirements

As seen in section 8.4 the system is made of a Ground Data Terminal (GDT) and a Air Data Terminal (ADT) communication segments. Both the terminals can have different data-links modules according to the system requirement. In the Sky-Y two LOS data-links are provided for redundancy and described below. Then the requirements for a BLOS data-link are proposed.

9.6.1 LOS Data-Link

The communications between the ground segment and the UAV is grant by the Data-Link systems providing a full duplex redundant command and control data exchange.

The communications system should also be able to provide connectivity through the Air Traffic Control (ATC) channels to connect the UAV and the ATC operator on the ground.

Summarizing the system is asked to provide:

- Commands and Controls Data between the AV and ground segment
- Data, Image and Video from AV to ground segment
- ATC connectivity

For this purposes two different Data-Link modules have been implemented:

- A Narrow Band Data Link (NBDL) for data exchange between the UAV and the ground segment in LOS
- A Wide Band Data Link (WBDL) for data exchange between the UAV and the ground segment in LOS

Narrow Band Data Link LOS

This Data Link operates as the main command and control channel: it will provide a bidirectional, low-data rate, high-integrity data exchange between the CS and the UAV platform.

The expected data to be exchanged through the NBDL are relevant to the following categories:

Uplink messages

- UAV platform commands (e.g. near-real-time steering commands, mission re-planning, etc.)
- ATC/ATM voice/data from the remote pilot in the GCS to the UAV (uplink)

Downlink messages

- UAV platform controls (e.g. platform position & attitude, flight telemetries, health status, etc.)
- ATC/ATM voice/data from the UAV to the remote pilot in the CS

In case of failure or temporary unavailability of the NBDL, the WBDL, may be used as a back-up system.

Wide Band Data Link LOS

The WBDL operates as the Command and Control channel, ISR and mission data. The WBDL provides point-to-point, full-duplex, high-data rate, bidirectional communications between the CS and the UAV platform through a wide-band downlink channel. This system operates over the full Ku-Band frequency. The expected data to be exchanged through the WBDL are relevant to the following categories:

Downlink channel

- ISR payload data (e.g. sensors data)
- ISR payload status data (e.g. aiming point, health status, etc.)
- UAV mission controls/status
- UAV platform status data (e.g. position, attitude, flight telemetries, platform health status, etc.)
- ATC/ATM voice/data from the UAV to the remote pilot in the CS

Uplink channel

- ISR payload commands (e.g. steering commands, operative mode changes, etc.)
- UAV mission commands (e.g., mission re-planning)
- ATC/ATM voice/data from the remote pilot in the CS to the UAV (uplink)

The WBDL also acts as a back-up system for the NBDL for the transmission of platform commands and controls/status messages.

9.6.2 BLOS Data-Link requirements

In this section, the architecture and characteristics of a WBDL Satellite data-link for the BLOS communication between the CS and the UAV are proposed. The BLOS communication considered will be used for mission purposes as a main role and for command and control as a backup function.

Wide Band Data Link BLOS

The WBDL-BLOS Data Link shall operate as the main transmission channel for ISR and mission data, while the UAV is operating in BLOS conditions, with operational range of 1000 NM. The WBDL-BLOS shall provide point-to-multipoint, full duplex, bidirectional communications between the GCS and the UAV platform through a wide-band downlink satellite channel. This system operate over the full Ku-Band or Ka-Band frequency.

The expected data to be exchanged through the WBDL-BLOS are relevant to the following categories:

Downlink channel:

- ISR payload data (e.g. sensors data)
- ISR payload status data (e.g. aiming point, health status, etc.)
- UAV mission controls/status
- UAV platform status data (e.g. position, attitude, flight telemetries, platform health status, etc.)

Uplink channel:

- ISR payload commands (e.g. steering commands, operative mode changes, etc.)
- UAV mission commands (e.g., mission re-planning/rehearsal)

The WBDL-BLOS will also act as a back-up system for the system LOS for the transmission of platform commands and controls/status messages in case of failure or temporary unavailability of the NBDL-LOS and WBDL-LOS.

The system main characteristics are summarized below:

- Data-link type: Bi-directional and Full Duplex link
- Operational Frequency: Military Ka (20.2÷21.2 downlink, 30.0÷31.0 uplink)
- Modulation Type:
  Downlink O-QPSK;
  Uplink BPSK
- Data Rate:
  Return link (ADT-to-GDT)) bit rate Up to 45 Mb/s
  Forward link (GDT-to-ADT) bit rate: 200 Kb/s
- Range: 1000 NM depending from satellite coverage with at least 10 dB margin
- Bit Error Rate (BER):
  10-8 without encryption
  10-6 when encryption is employed
- Anti-jamming capabilities
- Encryption type: software encryption

9.6.2.1 Multiple UAVs BLOS management

The advantage of an higher LOA, apart the possibility of permitting a BLOS capability, is also the possibility to control multiple UAV by the same CS pilot/operator. This feature impacts mainly the CS, but also the data-link should keep into account the transmission of data for more than one UAV from the same source.

The hypnotized management of multiple UAVs considers a mission in which only one UAV can be controlled at the same time, while the others are just monitored in BLOS.

The architecture for the UAV has been based on the following logical approach:

- The primary requirements expressed in above are compared with the obtainable performances in terms of satellite access scheme, link budget and the constraints imposed for satellite transmission from user terminals with limited antenna size.
- A traditional airborne Parabolic antenna is considered as a baseline implementation for the WBDL Satellite Airborne Data Terminal (ADT)
- Optionally, a solution based on an airborne Phased Array antenna is considered

Satellite access scheme

From the point of view of system architecture, one of the most important parameters to be considered for the implementation of a satellite data link is the satellite access method.

The WBDL Satellite data link must support high throughput data transmissions for long periods of time: for this reason, an FDMA satellite access, based on pre-planned transponder and frequency assignment appears as the most suitable approach. In the following figure, the WBDL Satellite allocation within the overall system-level architecture is depicted, though other ones may be defined.

In Fig 85, the mission BLOS ground data terminal is co-located with the Control Station, which can, if necessary, be configured in order to exchange data with more than one UA.
It is assumed that the satellite resource management is performed by the SATCOM service provider through an Earth Station (ES) located elsewhere.
This scenario requires a pre-emptive coordination between the CS and the ES in order to ensure the allocation of an appropriate bandwidth and number of satellite transponders to support the required connections.

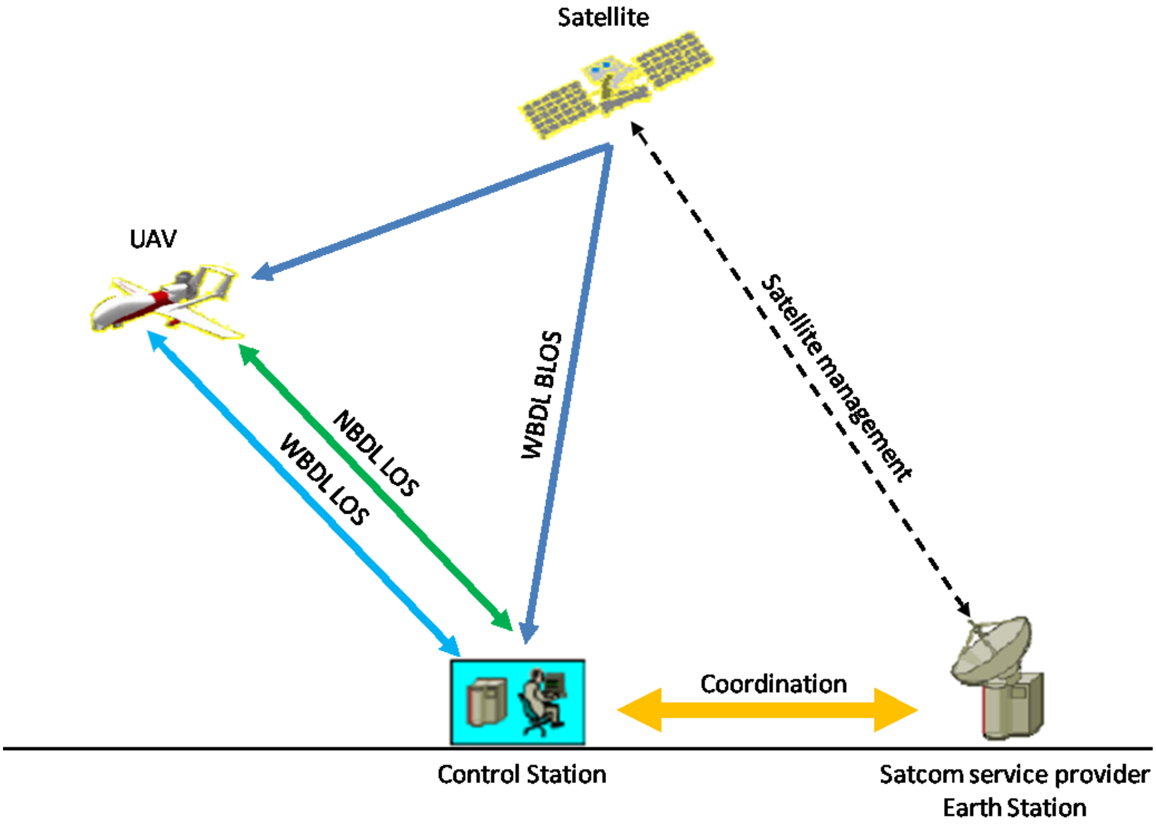


Fig 85 LOS and BLOS data-link scheme.

Considering that the typical WBDL Satellite operational usage involves the usage of a significant Transponder bandwidth (especially in the return link direction) on a time-continuous base, the usage of TDMA satellite access scheme appear as non suitable for the application.
Conversely, an FDMA satellite access scheme is proposed as a solution: in this scenario, it is assumed that a suitable bandwidth is allocated on one or more satellite transponder and dedicated to the UAS mission for its whole duration.

To consider the requirement of monitoring multiple UAVs (up to 3 UAVs are considered), some further consideration is necessary in order to optimize the transponder usage, while not exceeding its overall bandwidth (assumed equal to 54 MHz).

- The WBDL satellite requirements can be refined as follows: The UAV executing the mission shall have the capability to transmit at full rate on the return link (UAV to Satellite to ground)

- The UAVs under monitoring shall have the capability to transmit on the return link at the maximum rate that allows multiplexing the signals with the one of the main UAV while not exceeding the bandwidth of a single transponder
- Each UAV (in mission or under monitoring) shall share a single forward link (ground to Satellite to UAV) with an overall data rate of 200 Kbps at OSI physical layer level. As the data exchange will be based upon IP packets transmission, each UAV will have the capability to select its own data by checking the packet address at OSI layer 3.
- According to these hypotheses, the following architecture will be considered in this proposal: Each UAV will be equipped with a WBDL Satellite ADT capable of sustaining a transmission rate of at least 10 Mbps and a reception rate of 200 Kbps at physical layer level
- The three return link will be associated to three different carriers separated in frequency so that the multiplex signal can be contained within the bandwidth of a single satellite transponder
- The multiplex signal containing the three return links is converted in frequency at satellite level and transmitted to the WBDL Satellite GDT
- At GDT level, the multiplex signal is down-converted and passed to three dedicated demodulators, which extract the different return link data stream at physical layer
- After that, the IP packets streams are reconstructed and passed to the Ground data-link management system for further routing to the final users
- On the forward link, the IP packets directed to the various UAVs are serialized, converted into a single physical layer stream and transmitted in broadcast mode over a single carrier to all the UAVs

The above architecture is schematically represented in Fig 86.

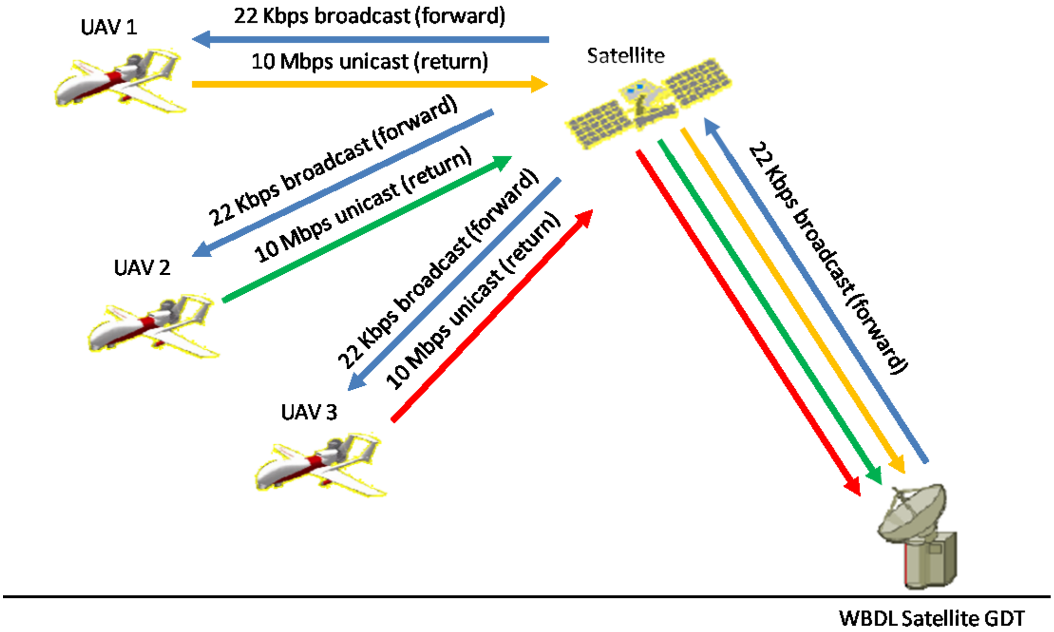


Fig 86 WBDL Satellite for BLOS multi UAV management scheme.

9.7 Improvement for increasing the LOA

Reached the automated navigation capabilities, developed as discussed in the previous chapters, the provision of new functions, improving the level of autonomy, can be evaluated. Such functions can be considered as an enhancement in the UAV autonomy for the management of the route navigation and sensor management in both LOS and BLOS conditions with the purpose to reach the ACL3 [18]. In ACL3 the system is able to determinate criticalities and modify in real time the flight or mission parameters to avoid the criticality, choosing between fixed roles. The system would adapt the action in base of the presence or not of LOS/BLOS link, the emergency level, the airport availability, etc.

The enhancement in the mission autonomy would consist in a module implementing a series of cooperating functions providing a partial or total route deviation in case of needs and/or a data-link source switch. According to the data-link condition (LOS or BLOS) and the eventuality of sub-systems criticality (Normal or Emergency Condition) the system is asked to perform different specific actions. The pilot is always demanded to confirm or reject an autonomous decision (if the command & control link is available).

The mail advanced Functions individuated to be part of such a module are:

- Collision Avoidance
- Autonomous On Board Re-plan
- Data-Link Coverage
- Fuel Consumption Prediction

Plus two functions specific for the sensor:

- BLOS Sensor management
- Sensor Coverage

Such functions have been defined as requirements for future implementation and some of them have been developed by external suppliers. Below is reported a brief description of their behavior.

If no failures are detected (Normal Condition) the provided function (excluding the sensor functions) is the Collision Avoidance:

In case of unexpected event requiring an evasive maneuver like risk of collision with other AVs or weather condition to be avoided, the function is demanded to calculate the probability of collision and the interception time and, based on the time, or sudden command an evasive maneuver or re-plan the mission. Both the solutions should keep into account to avoid possible no-fly zones stored in the mission plan. The time evaluation should consider the BLOS signal delay and if opportune the re-planning should occurs without waiting the pilot confirmation.

This function would cooperate with the Autonomous On Board Re-plan that is the function demanded to recalculate the route receiving as input data the output of the others listed functions (as the previous one).

In case of Emergency Condition instead the system should follow a priority list:

1) If the LOS data-link is available the pilot is demanded to take the command in LOS

2) If the LOS data-link is not available the system should evaluate the route re-planning in order to recover the link according to the link coverage.

3) If the previous solution is not possible the system should try to establish a connection with a BLOS data-link (if not already present)

4) If the BLOS connection is not available the system should verify the possibility of landing in an alternative airport

5) If the previous solution is not possible the system should activate the termination procedure

To perform the action number 2, and in any case an automatic on board re-plan occurs, the Data-Link Coverage function should be called. Its role is to estimate the data-link coverage according to the transmitters, the UAV positions and the terrain orography. With this information the validation of a route for the link recovery can be done. For example, in case of BLOS navigation, if a failure occurs and the point number 1 of the priority list is not available, this function would cooperate with the Autonomous On Board Re-plan to calculate the best route to reach the LOS link recovery. Then the system would evaluate if such a route is reachable by the UAV in that condition.

One more factor that the system should keep into account, any time an autonomous re-plan is required, is the Fuel Consumption Prediction. This function is used to discern if the new route re-plan can be validated by estimating the fuel consumption of the proposed route compared to the current fuel level.

About the sensor the first function considered is the BLOS Sensor Management. It is asked to control the mission sensors in a more autonomous way according to the mission in order to alleviate the operator on ground from the sensor command when a BLOS latency would add delay to the command signal and to the acquired video. The function would select the more efficient sensor according to the target typology and distance and would track the target commanding automatically the azhimuth and elevation of the sensor. The target selection is demanded to a priority list loaded by the CS. The system would point and track the target in view with higher priority level and, in case of an higher target enters in the view, the system would switch to the new higher target. The system would also transmit to the CS the information about the target and be able to receive a variation to the target list or to the priorities associated. Then the sensor pointing would consequently change according to the new data. In case of failure the system would reconfigure itself with the purpose of keep tracking the selected target.

The last step for increase the sensor autonomy is the Sensor Coverage function providing the forecast of the sensors possible fields of views respect to the planned route and the terrain orography. The scope of this function are two: the first is to validate the proposed route according to the sensor coverage and the target position, the second is to cooperate with the Automatic On Board Re-plan to propone a route with the best target visibility.

10 Results

As discussed in section 9.4, the approach for the steering functions development, has been to follow successive steps of simulation in different ambient. The first tool used was Matlab®, where mainly the single algorithms have been outlined and then the test involved most of all the different function interaction. Anyway, thanks to the developed 'Basic AV Model' (see section 9.4.1), also some simulation of basic flight behavior could be done to understand qualitatively the steering function answer.

10.1 Matlab® results

Considering a route made of four WPs, different functions have been tested by varying the second WP attributes. The first test exposed in this section is the flight of Fly-Through WPs at the same altitude and with constant speed.

10.1.1 Fly-Through WP

The WP attributes are summarized in the following table:

WP	Lat [deg]	Lat [rad]	Lon [deg]	Lon [rad]	Alt [ft]	Alt [m]	IAS [kts]	IAS [m/sec]
1	44,2	0,7714	7,2	0,1256	3280	1000	90	46,3
2	44,4	0,7492	7,4	0,1291	3280	1000	90	46,3
3	44,2	0,7714	7,2	0,1256	3280	1000	90	46,3
4	44,2	0,7714	8,0	0,1396	3280	1000	90	46,3

All WPs are Fly-Through. The resulting route is the following:

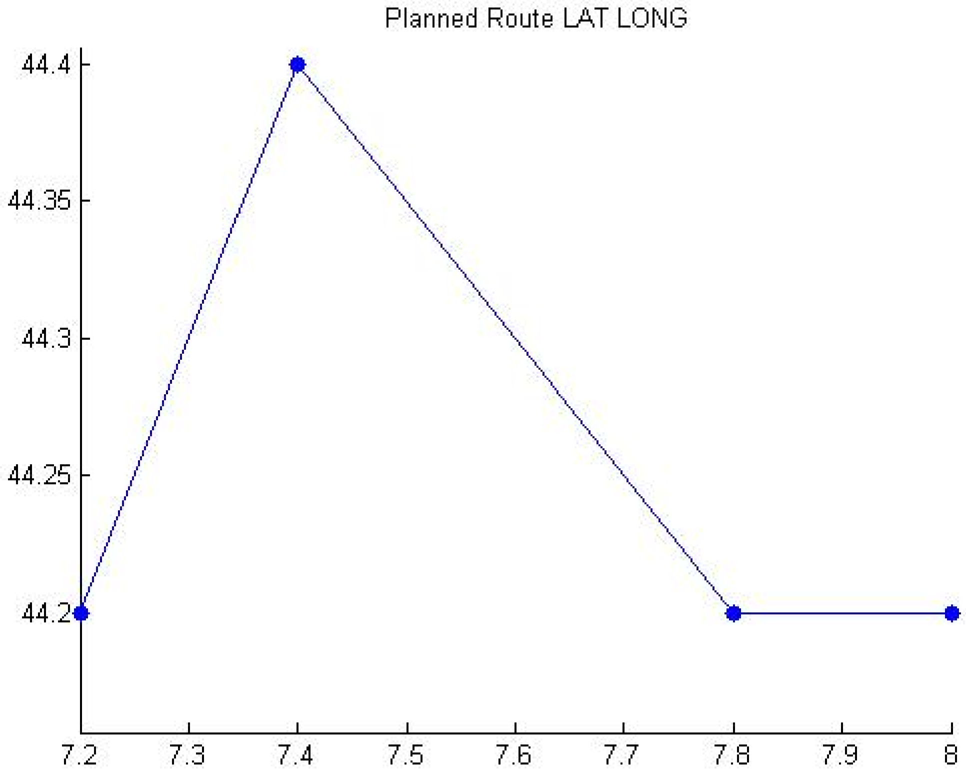


Fig 87 Matlab® test: Planned Route.

And the flight path performed by the Basic AV Model, according to the Steering Model output results cover exactly the planned path (red line):

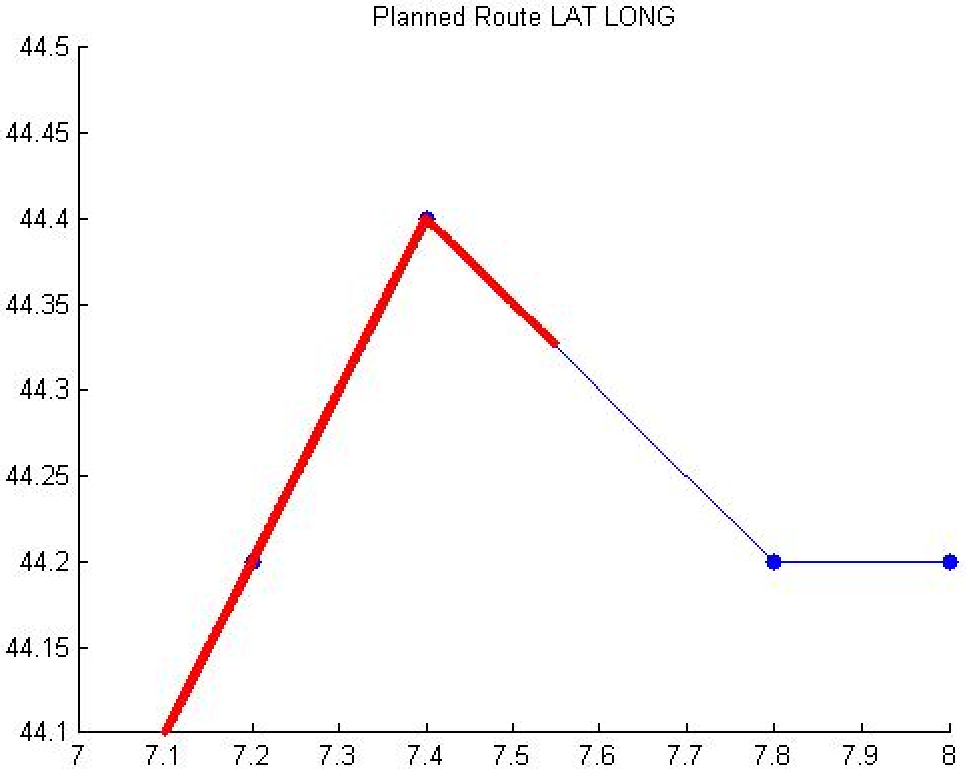


Fig 88 Matlab® test: Flight path.

Zooming the WP2 overfly can be noted the overshoot due to the Fly-Through WP-Type.

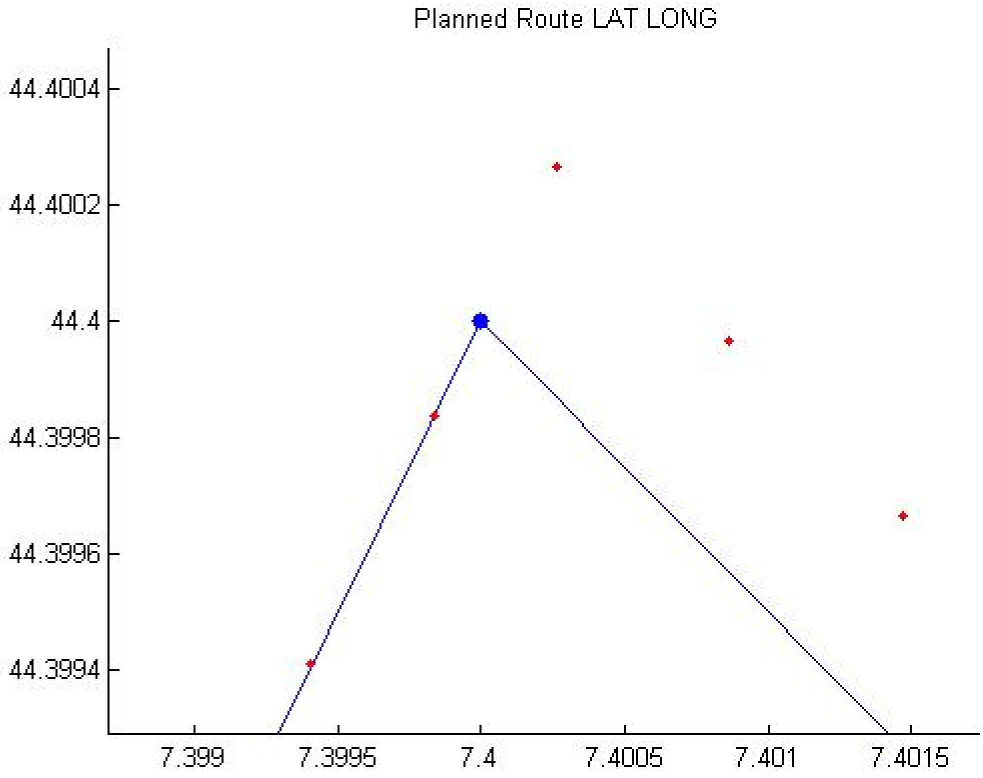


Fig 89 Matlab® test: Fly-Through WP overshoot.

As seen in section 9.4.1 the Basic AV Model does not perform a realistic turn and do not annul the X-Track. Anyway the purpose of this step was just validating the overall steering functions, so this level of approximation resulted to be sufficient.

Analyzing the Direct Range record data is possible to evaluate the change leg by the presence of discontinuities when the new WP is acquired as destination.

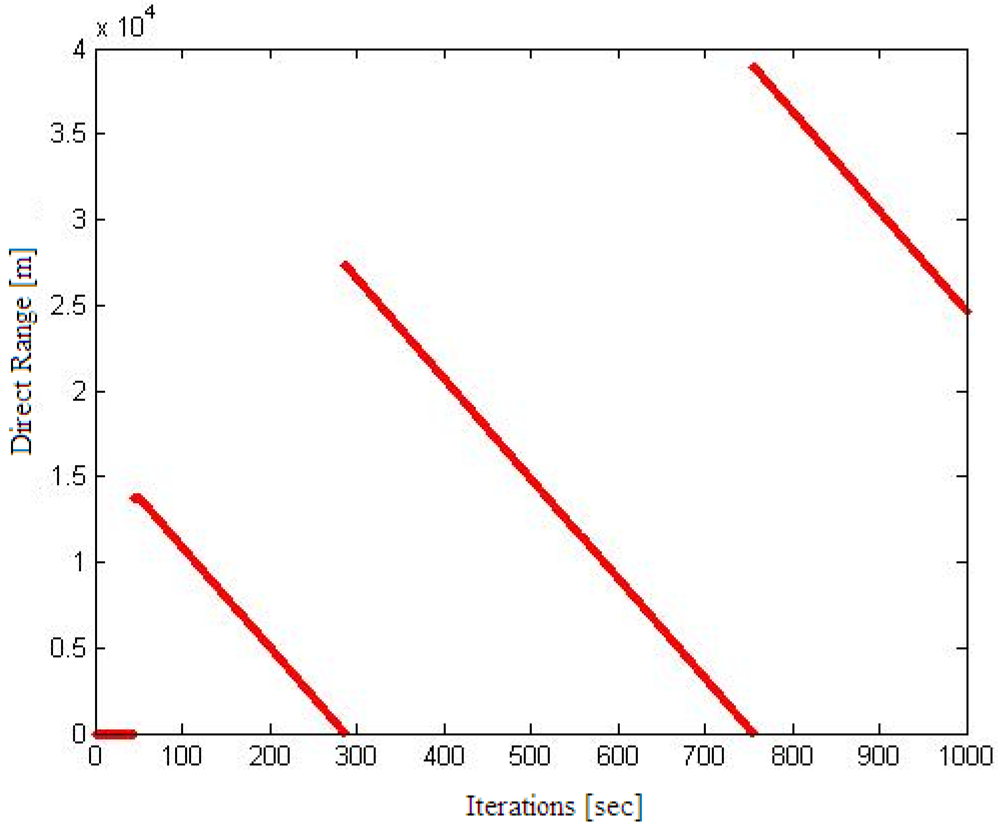


Fig 90 Matlab® test: Direct Range of Fly-Through WPs.

Zooming the Direct Range at the lower levels can be noted that the change leg occurs when the range is lower than a threshold of 80m.

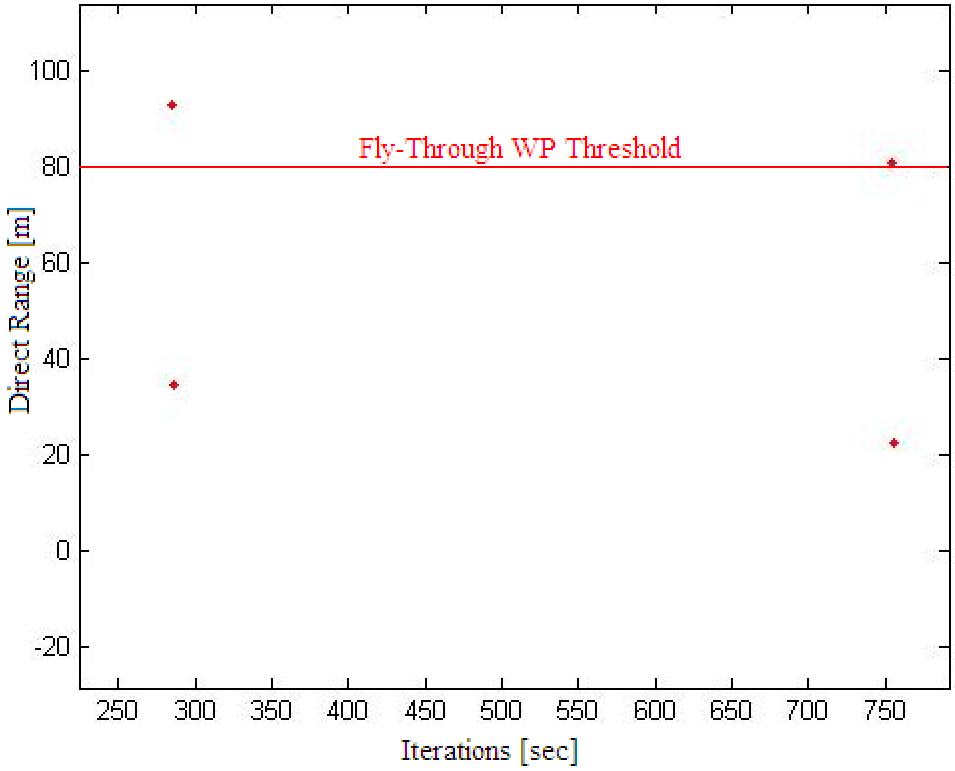


Fig 91 Matlab® test: Fly-Through WP threshold for WP acquisition.

10.1.2 Fly-By WP

The Fly-By test was performed with the same route by just varying the WP2 Type attribute to Fly-By.

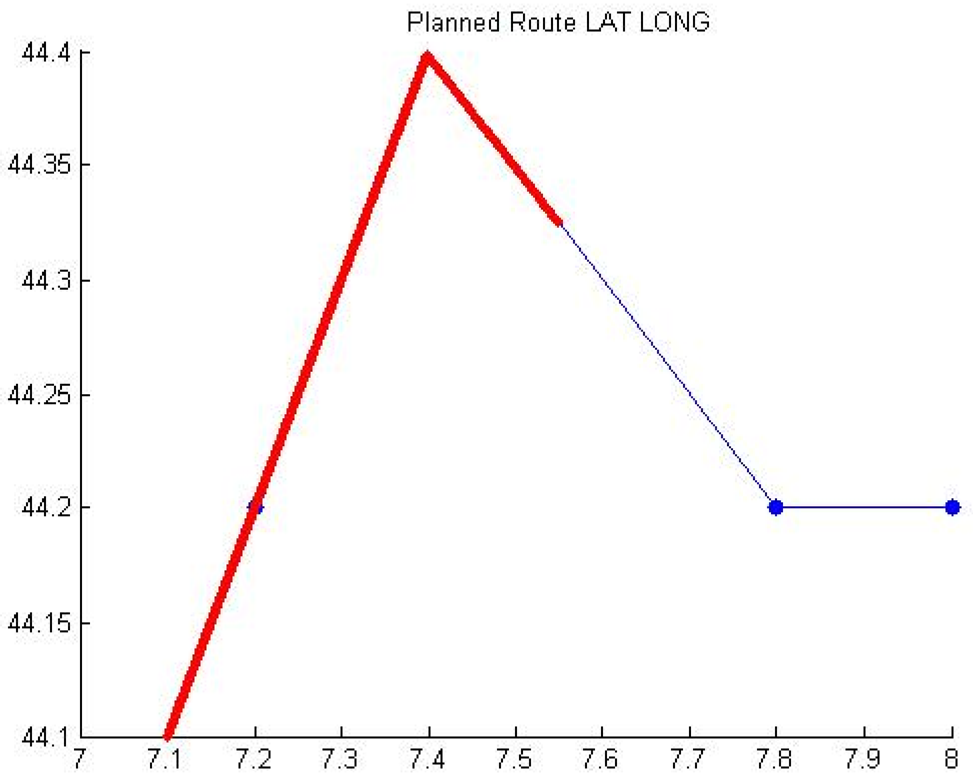


Fig 92 Matlab® test: Flight path

Apparently the path is similar to the previous case, but it is just due to the simplified flight simulation. Going more in detail is possible to see that in this case the change leg occurs before the WP2:

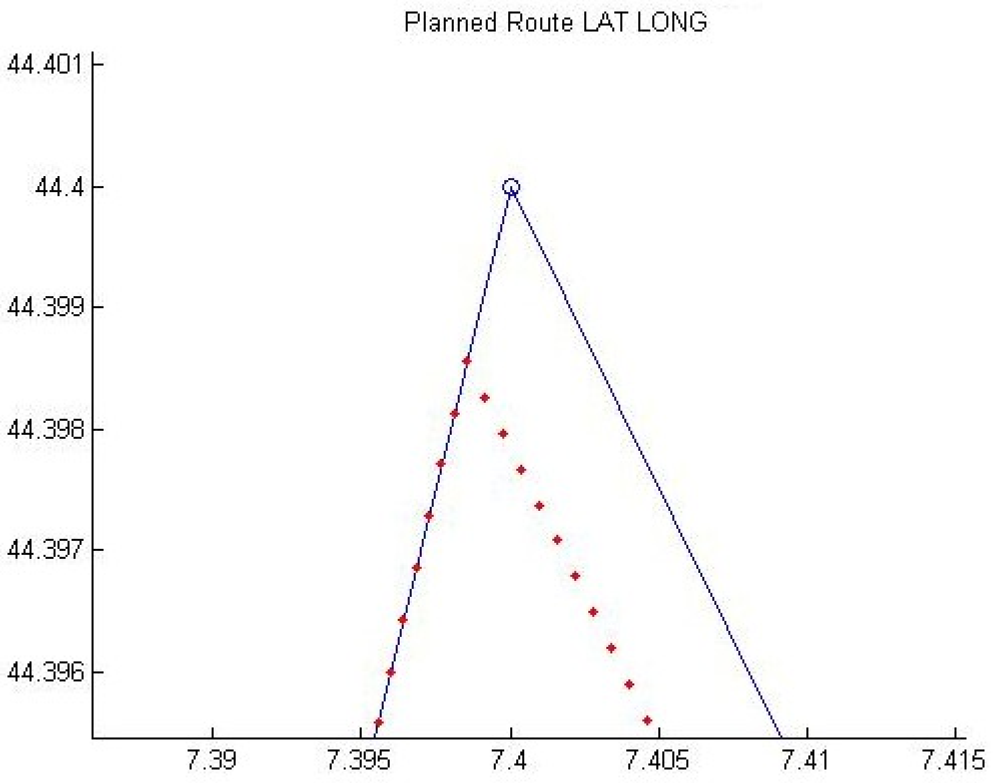


Fig 93 Matlab® test: Fly-By WP acquisition.

The Dist To Roll In value calculated for all the WPs is shown in the following figure

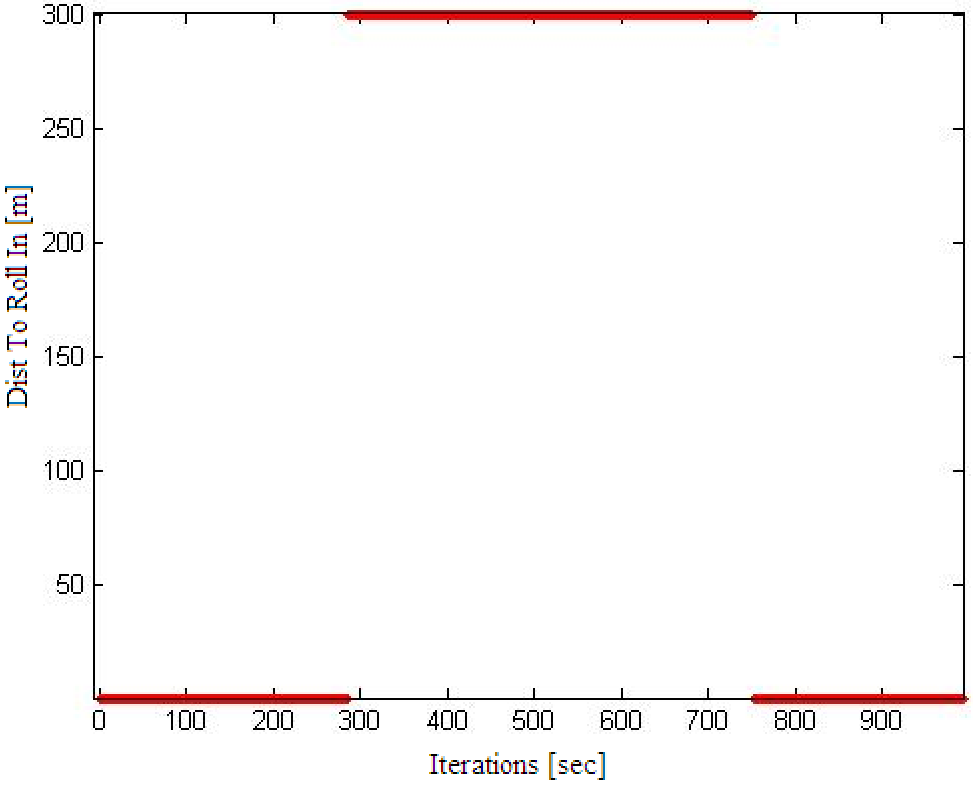


Fig 94 Matlab® test: Dist To Roll In.

Can be noted that for the WP2 results to be 300m, while for all the rest of the route is zero as the other WPs are Fly-Through.

From the Direct Range graph is possible to compare the distance of acquisition of WP1 and WP2.

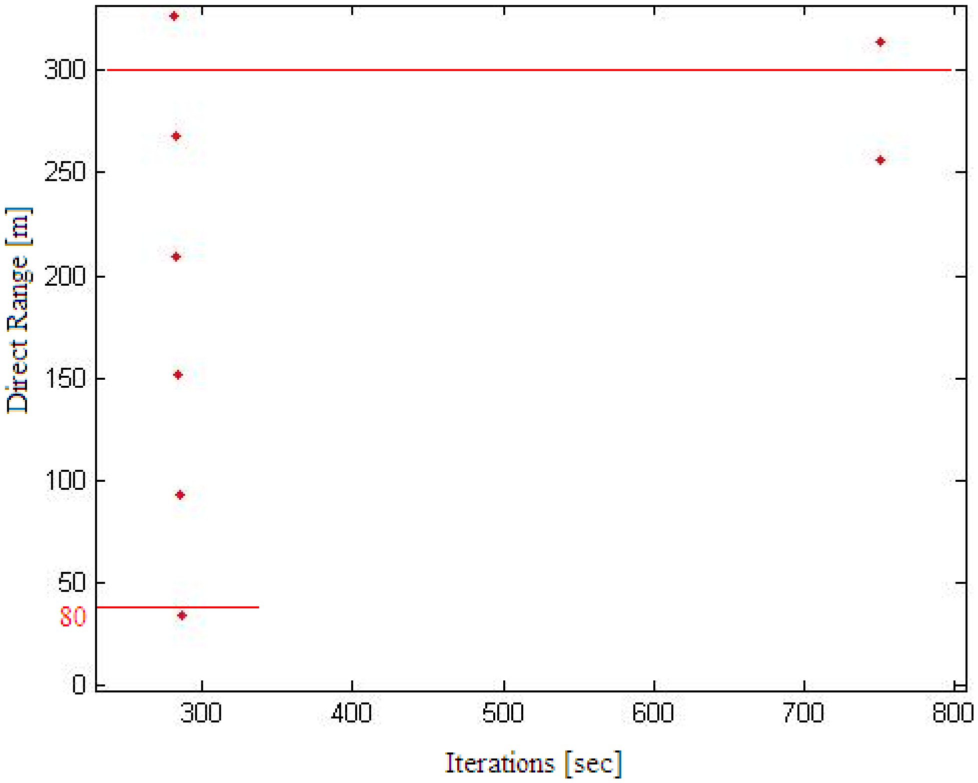


Fig 95 Matlab® test: Direct Range for Fly-Through and Fly-By acquisition comparison.

The first WP is Fly-Through and the change leg occurs at the iteration at which the range is lower than 80m, the second WP is Fly-By and the iteration of change leg is the first surpassed the value of 300m which is the Roll In Distance seen before.

10.1.3 Altitude variation

Using always the same route with all Fly-Through WPs, the test of the altitude command was done. In this case all WP altitude is set to 1000m except to the WP2 which is 2000m.

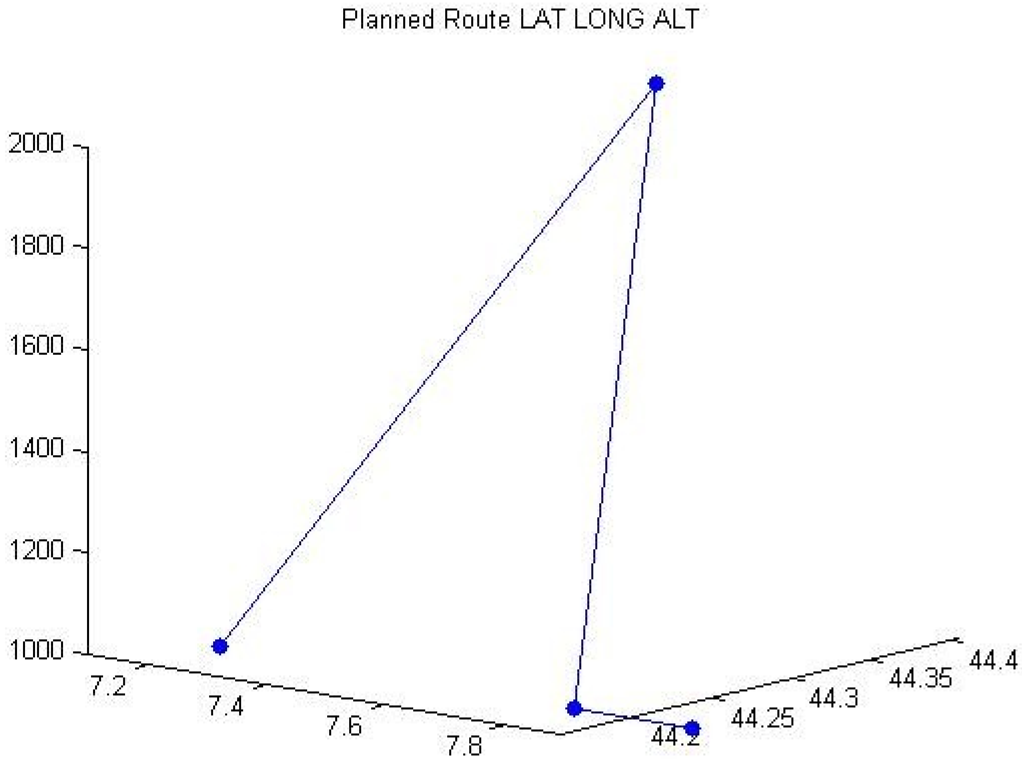


Fig 96 Matlab® test: Planned route 3D.

The path performed by the Basic AV Model respect the expectative: it reach the altitude as soon as possible without following the leg inclination. Of course the ramp adopted is not significant of a real aircraft, but is just used for testing the algorithms.

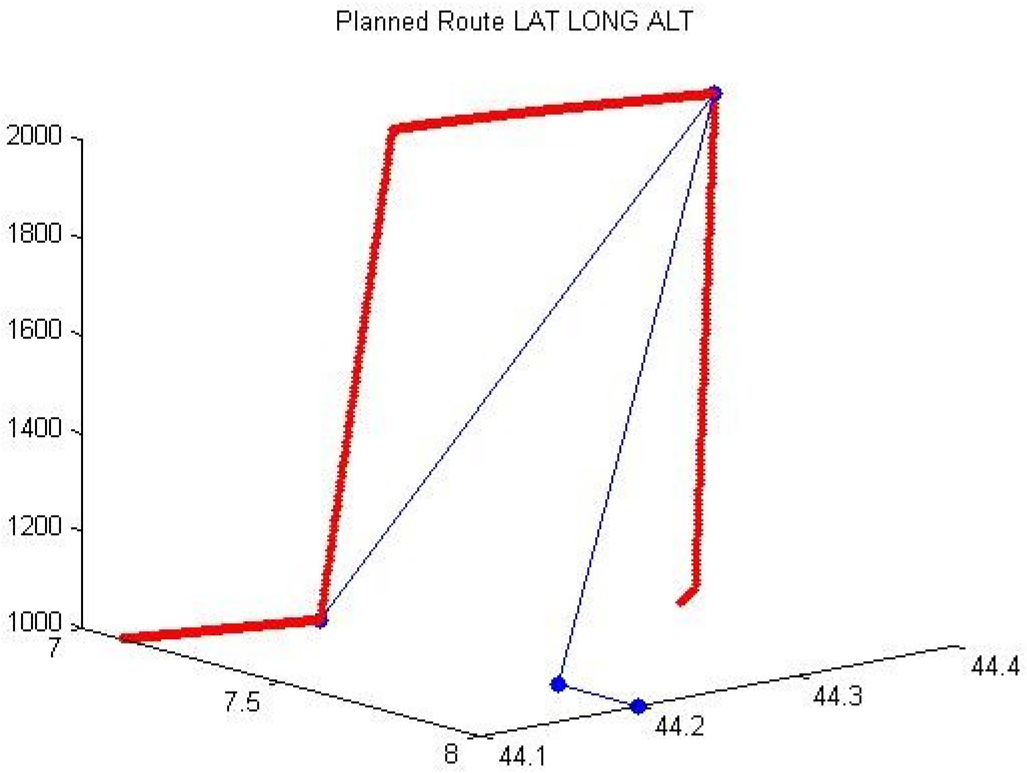


Fig 97 Matlab® test: 3DFlight path.

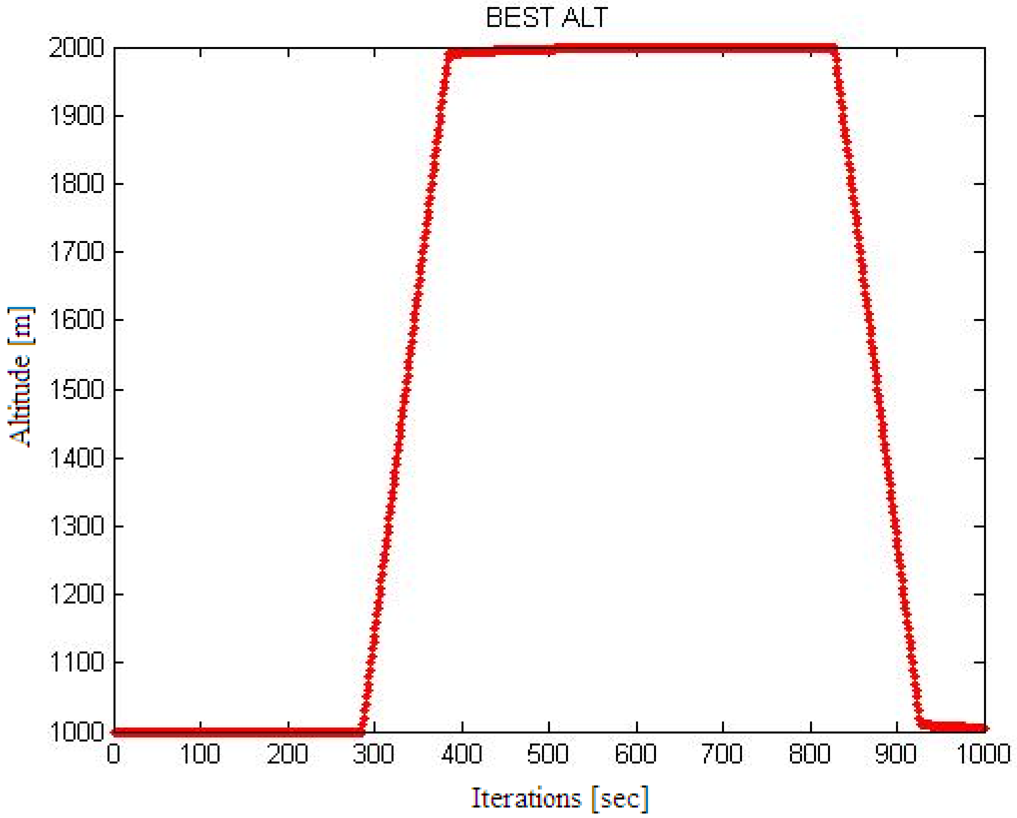


Fig 98 Matlab® test: Best Altitude path.

10.1.4 Speed variation

For the speed command evaluation the WP2 IAS was increased to 100kts (51.4m/sec), while all the other WP's IAS remained 90kts (46.3m/sec). The test was simulated at sea level without wind, so IAS and Groundspeed resulted to be the same.

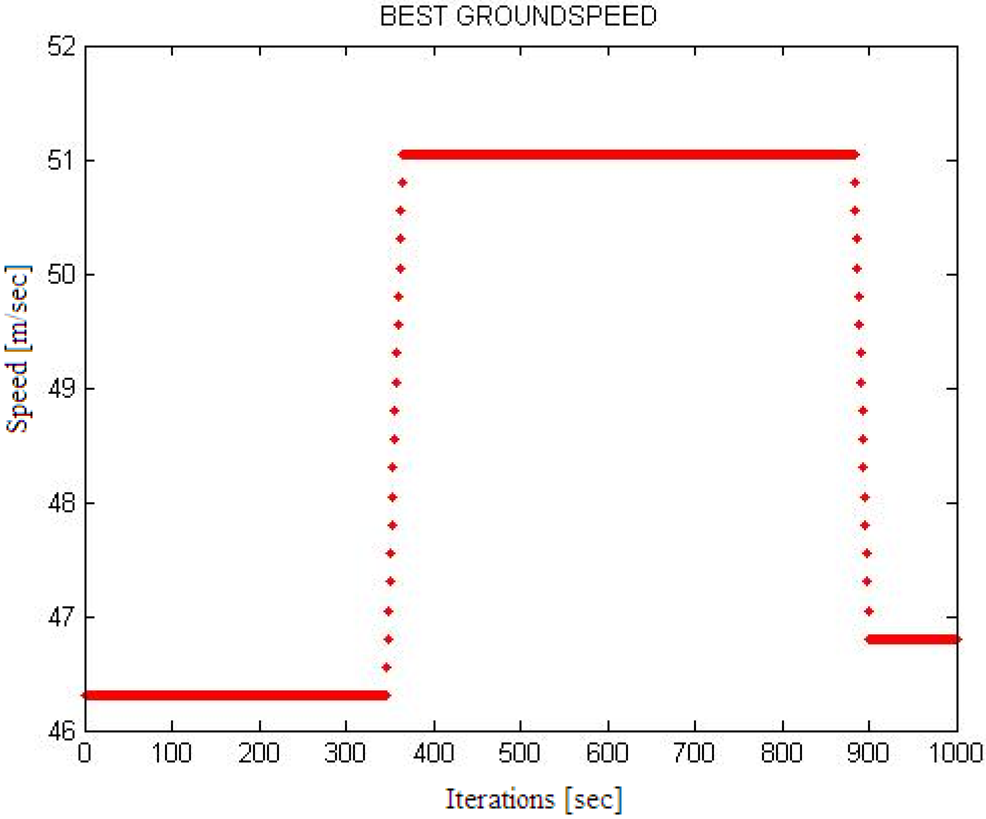


Fig 99 Matlab® test: Speed path.

Also in this case the acceleration is not realistic due to the model, but this behavior does not affect the test.

Comparing the groundspeed of the simulated vehicle (blue line) to the altitude error from the steering (red line) can be noted the expected parallel:

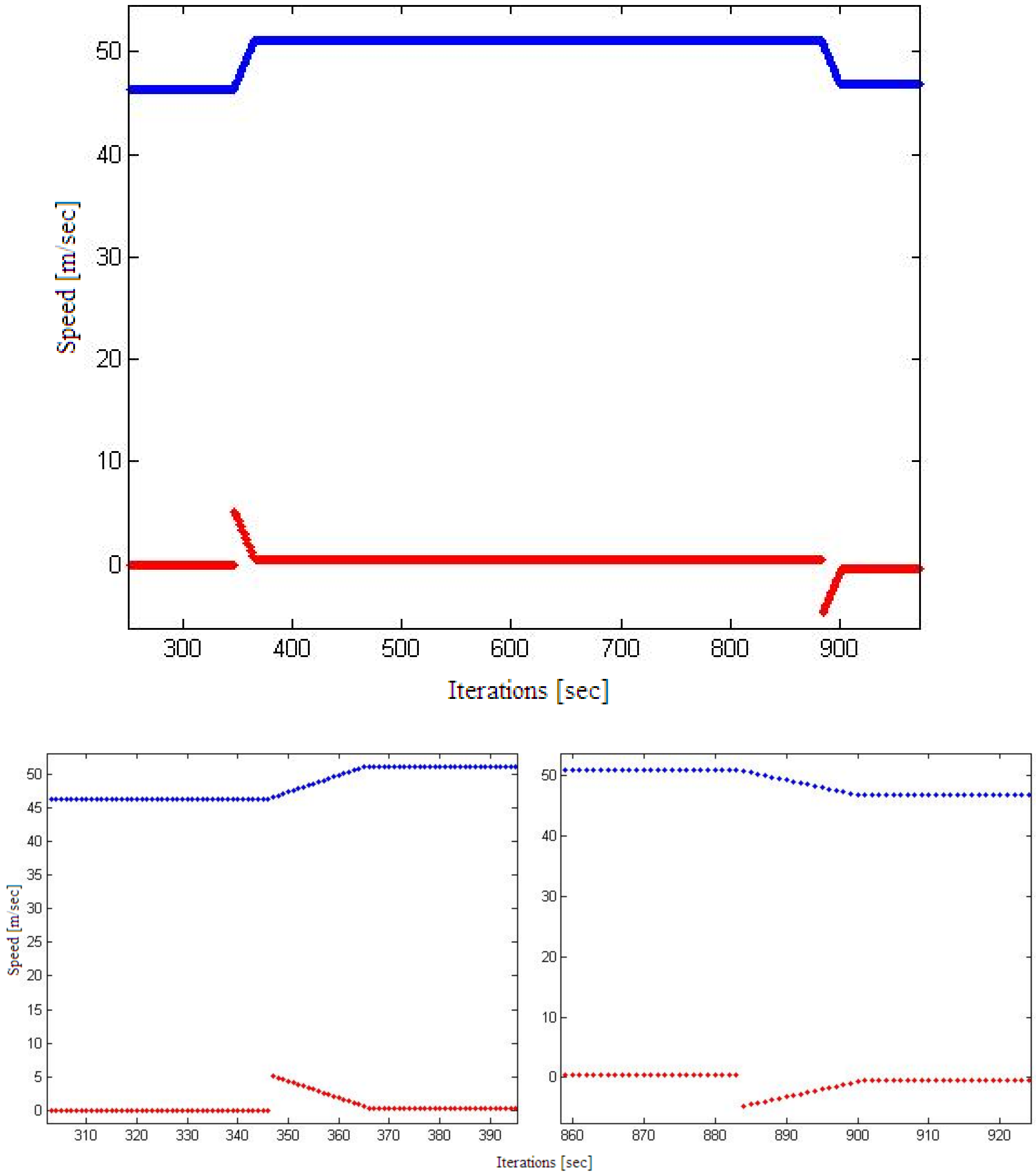


Fig 100 Matlab® test: Speed (blue line) VS Speed Error (red line).

10.2 Simulink® results

After the Matlab® tests a Simulink® flight simulator has been developed to have a more realistic UAV flight and to implement a more realistic FCS, so was possible to have a more realistic real time steering stimulation. The UAV model was not the Sky-Y model because of the company reserved information. The same occurred for the FCS, so that a generic UAV model was implemented and also a generic FCS was built by adding DIP controllers receiving the steering output and producing the proper surfaces deflections. Respect to the Matlab® model with this simulator was possible to have also the X-Track correction, a smooth curve to join the legs, and so also the correct loiter pattern flight.

Moreover the integration of the Fly Gear® Flight Simulator graphic interface permitted to have an immediate evidence of the aircraft qualitative behavior.

Also for the Simulink® tests a short route made of 4 WPs was selected. Due to the heavy computation of the model respect to the platform used for the run, the simulation would result too slow with a long route. So that, respect to the Matlab® tests, the route selected for the following Simulink® test results is shorter, moreover, for the gain tuning assessment, the turn provided with such route is a left turn to show a different case respect to the previous.

The route provided for this test is made of the following Fly-Through WPs:

WP	**Lat** **[deg]**	**Lat** **[rad]**	**Lon** **[deg]**	**Lon** **[rad]**	**Alt** **[ft]**	**Alt** **[m]**	**IAS** **[kts]**	**IAS** **[m/sec]**
1	44,20	0,7714	7,2	0,1256	3280	1000	90	46,3
2	44,22	0,7718	7,22	0,1260	3280	1000	90	46,3
3	44,24	0,7721	7,2	0,1256	3280	1000	90	46,3
4	44,26	0,7725	7,2	0,1256	3280	1000	90	46,3

10.2.1 PID Tuning

As discussed before the development of this simulator involved the model of a module providing the FCS function of converting the steering output into surfaces deflections. This module required the tuning of the PID controllers of which is made for a proper aircraft behavior.

After some trials the route resulted flown properly, but the PID tuning had to be improved.

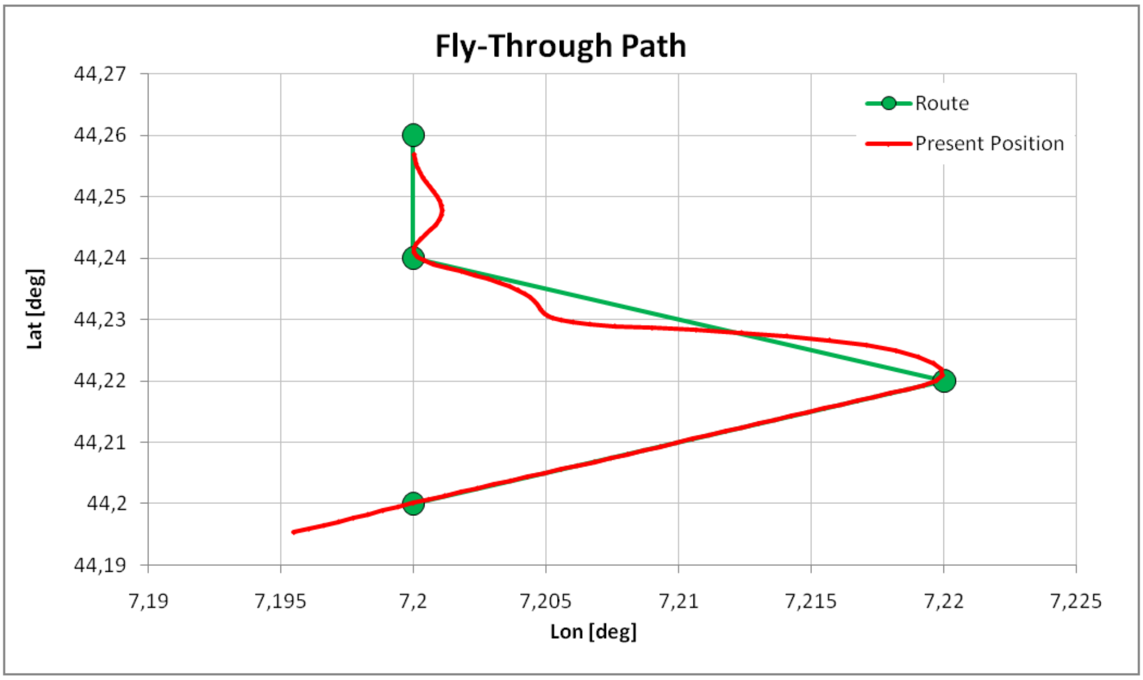


Fig101 Simulink® test: Flight path during the PID tuning.

As expected the Fly-Through WPs are over-flown, but the X-Track correction generates a too high oscillation at the new leg acquisition. The overshoots are observable from the X-Track time history:

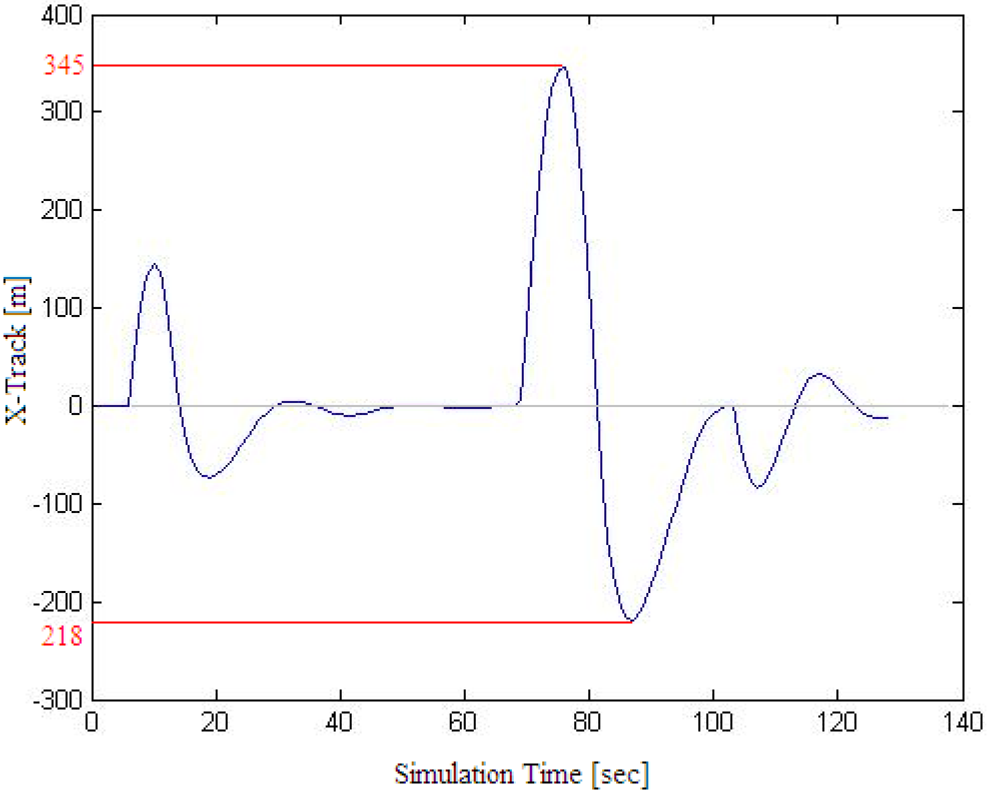


Fig 102 Simulink® test: X-Track error during the PID tuning.

The Fly-Through produces an overshoot of about 345m, that considering the UAV model adopted can be considered good, anyway the controllers produce an oscillation in the other direction of about 218m that, being about the 63% of the main peak, cannot be accepted. As well as the second overshoot takes exactly double the time of the first (18sec and 36sec).

The controllers gains were changed and, after some trials, a new behavior was obtained:

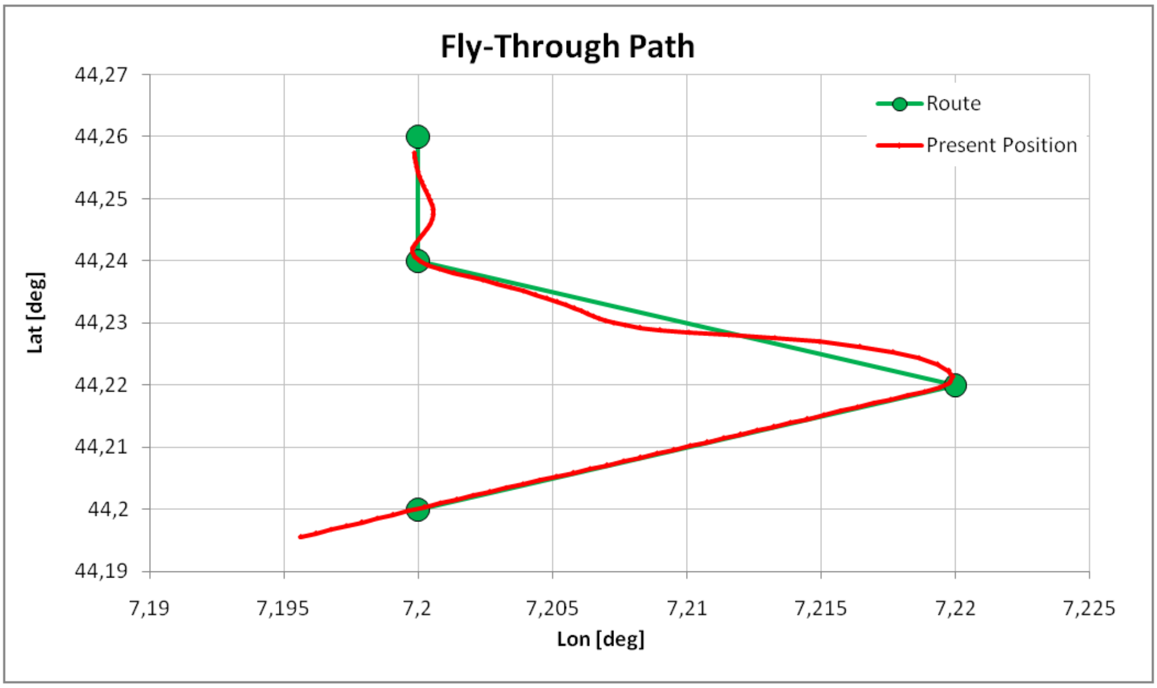


Fig 103 Simulink® test: Flight path during the PID tuning improvement.

The oscillation is reduced and, from the X-Track time history, can be noted that the main overshoot peak is quite the same (347m), while the oscillation peak is reduced to 95m, which represents only the 27% of the main one.

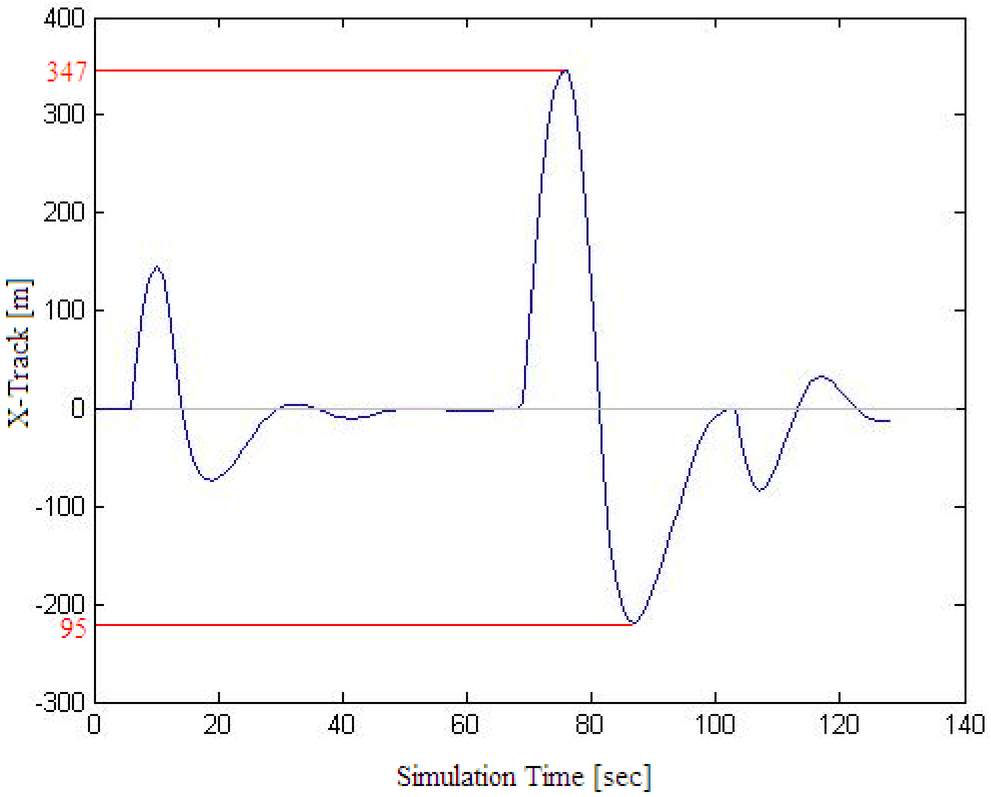


Fig 104 Simulink® test: X-Track error during the PID tuning improvement.

Also the dumping time of the oscillation is reduced to 24sec.

Observed an improving of the performances the gains were tuned again in the same direction to try to obtain a better behavior. Some more trials were done and a new behavior was obtained.

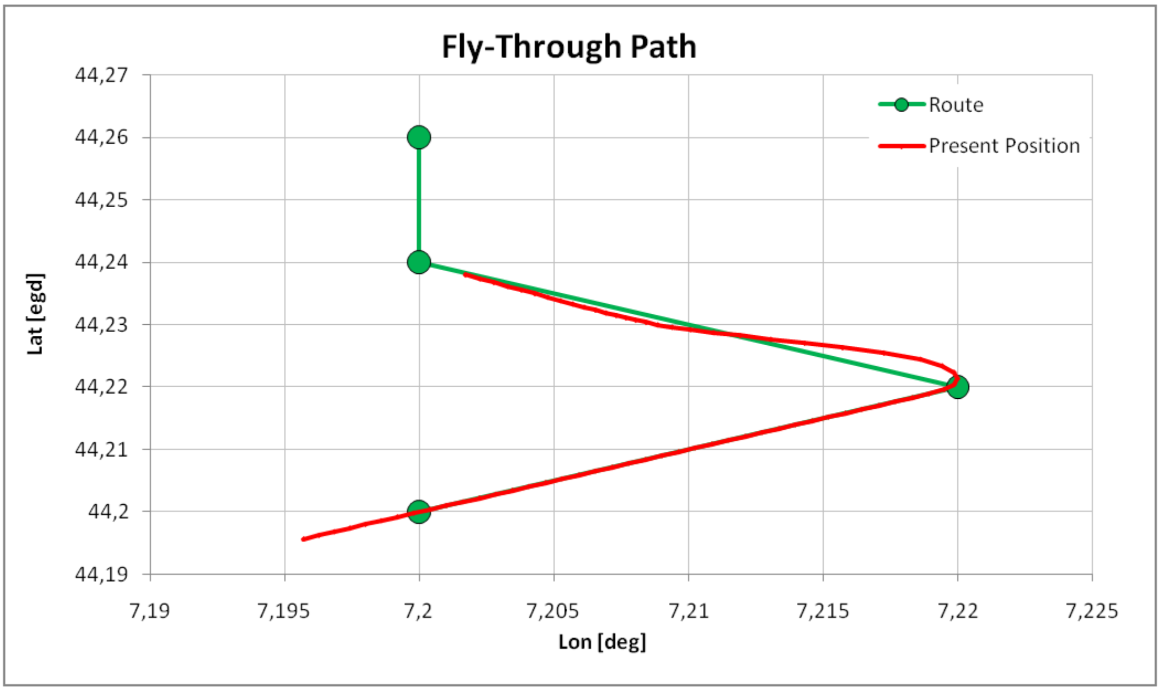


Fig 105 Simulink® test: Fly-Through WP flight path after the last PID tuning.

The new behavior results to be better than the previous: as can be observed from the path and from the X-Track time history the oscillation peak is avoided (only 6m).

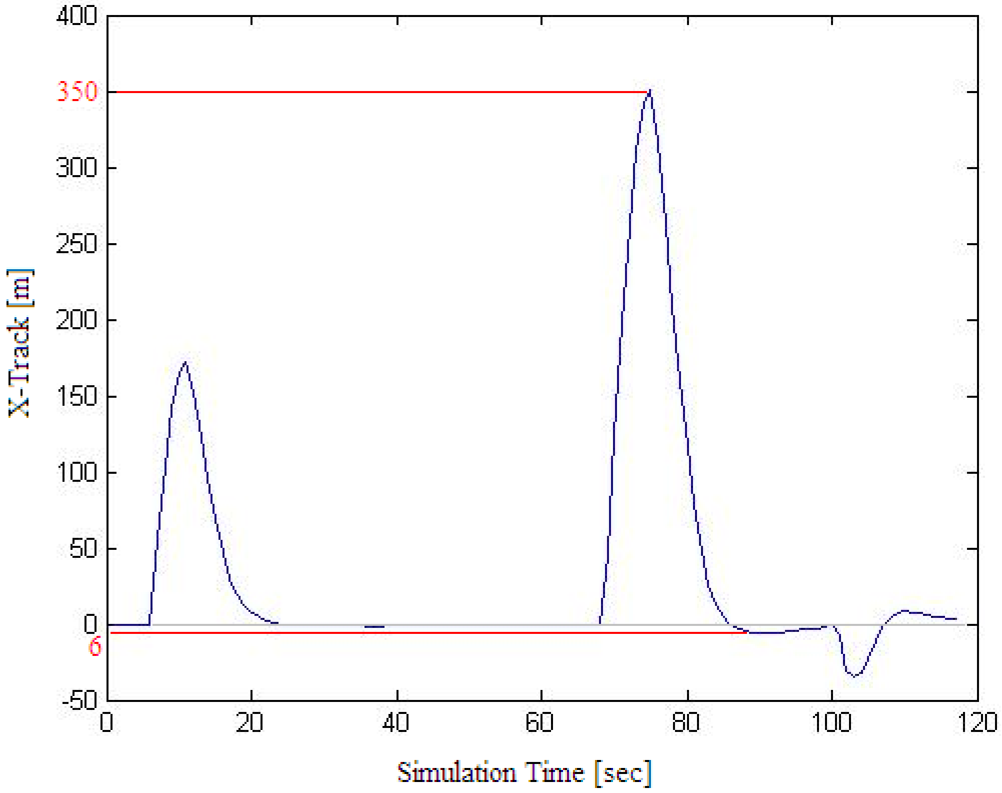


Fig 106 Simulink® test: X-Track error after the last PID tuning.

As the purpose is to test the steering functions and not the FCS controls performances (as the FCS for the real UAV would be provided by another Alenia Aermacchi team), this behavior is considered good enough for the scope. Moreover, as said before, the aircraft used for this simulation is not the UAV that would be implemented with such navigation functions. So the tuning of a FCS that would not be used on the real aircraft for controlling a simulate UAV that would not be representative of the real one, would just take enough time to provide a good behavior compared to the functions to be tested. This last tuning step resulted to be enough for that purposes, so for the following examples these value of gain for the FCS PID have been considered.

10.2.2 Fly-Through

The route provided for the Fly-Through test is similar to the route used for the Matlab® test, so the WP2 would provide a right turn, but, in this case, it is shorter due to the computational heaviness discussed previously. This route is made of the following Fly-Through WPs:

WP	Lat	Lat	Lon	Lon	Alt	Alt	IAS	IAS

	[deg]	[rad]	[deg]	[rad]	[ft]	[m]	[kts]	[m/sec]
1	44,20	0,7714	7,20	0,1256	3280	1000	90	46,3
2	44,22	0,7718	7,22	0,1260	3280	1000	90	46,3
3	44,20	0,7714	7,24	0,1263	3280	1000	90	46,3
4	44,20	0,7714	7,26	0,1267	3280	1000	90	46,3

Focusing on the second WP can be seen that the steering model command correctly the change leg after the WP overfly, and the Simulink® model executes, differently respect to the Matlab® model, a smooth curve and also the X-Track correction.

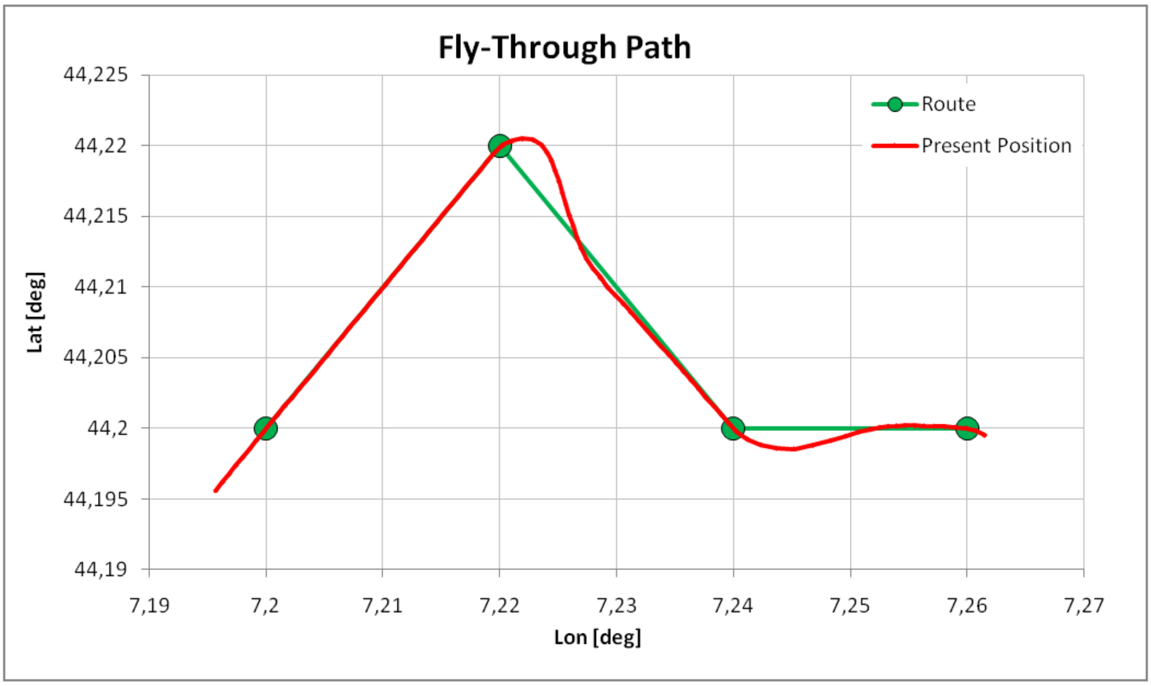


Fig 107 Simulink® test: Fly-Through WP flight overshoot.

The Fly-Through WP acquisition is intended to be an overfly WP, anyway has a settable threshold distance at which the steering consider the WP overflawn. For the Simulink® tests this threshold was set to 80m as can be observed from the lower values of the Direct Range (the peak after the discontinuities mark the change leg).

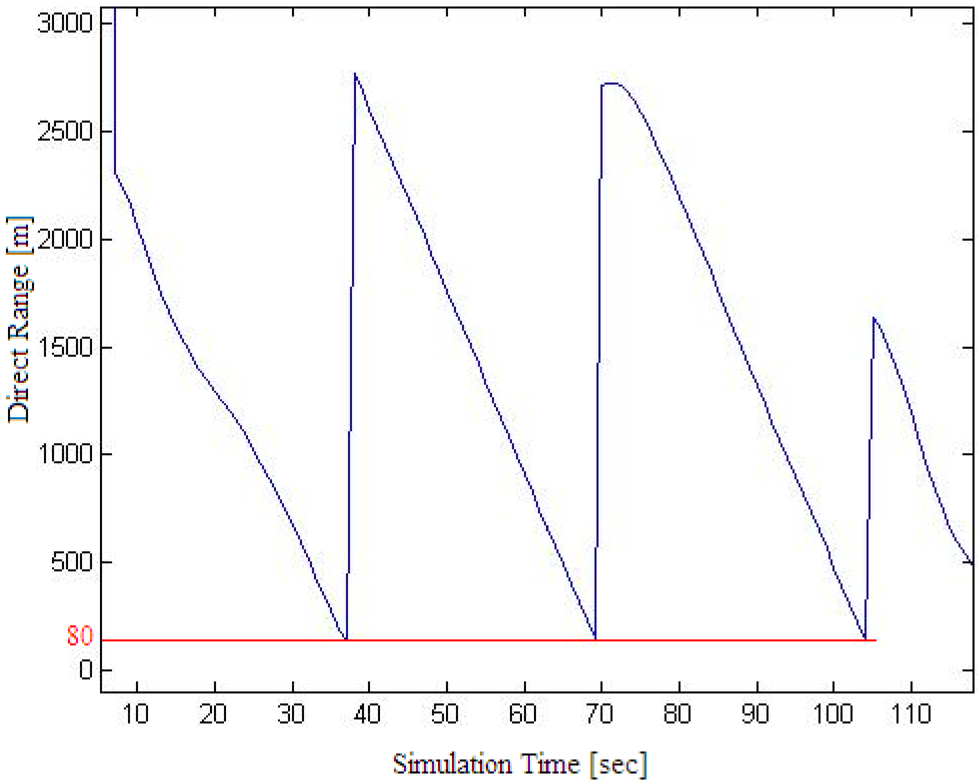


Fig 108 Simulink® test: Direct range.

On the graph presenting the Commanded Track (green line) and the Track (blue line) is possible to see the overshoots during the Fly-Through turns by the oscillations of the Track respect to the Commanded.

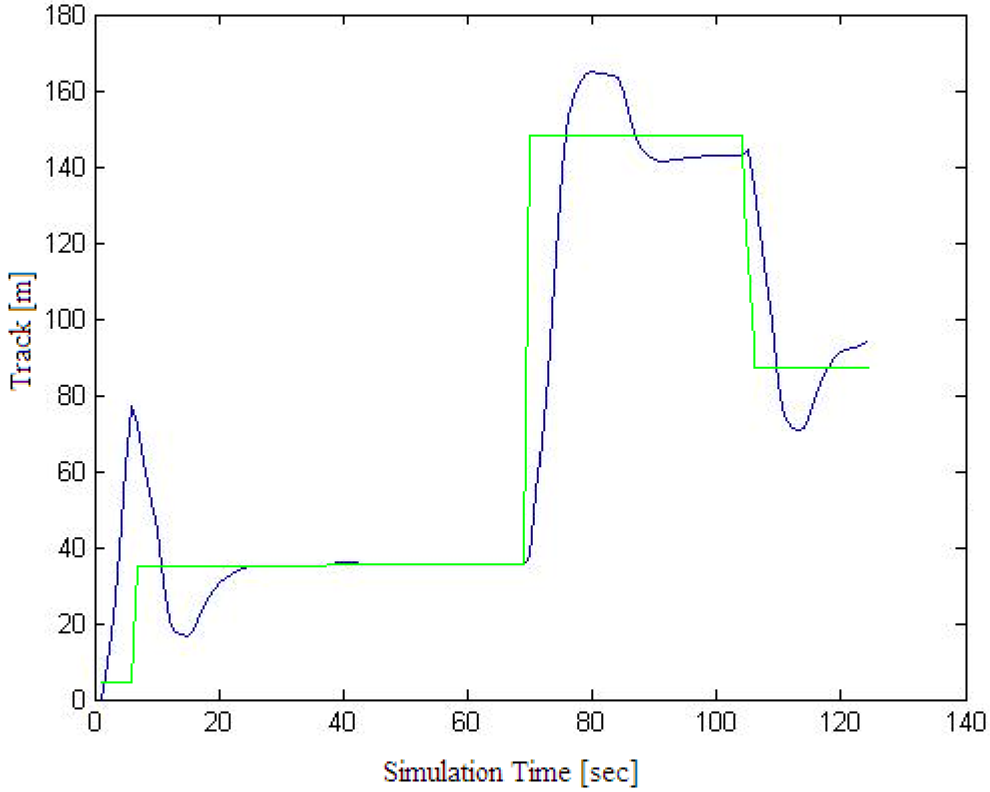


Fig 109 Simulink® test: Commanded Track (green line) VS Track (blue line).

As seen before the Simulink® model corrects also the X-Track Error:

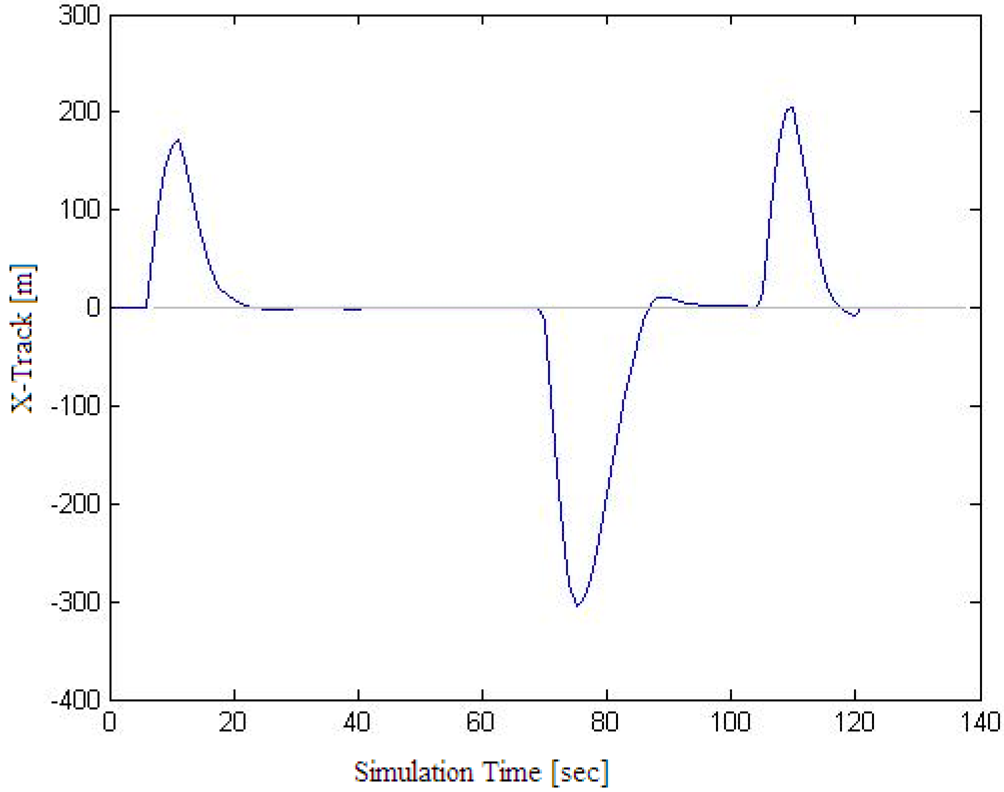


Fig 110 Simulink® test: X-Track Error.

10.2.3 Fly-By

The Fly-By test was performed with the same route with the WP2 Type set to Fly-By:

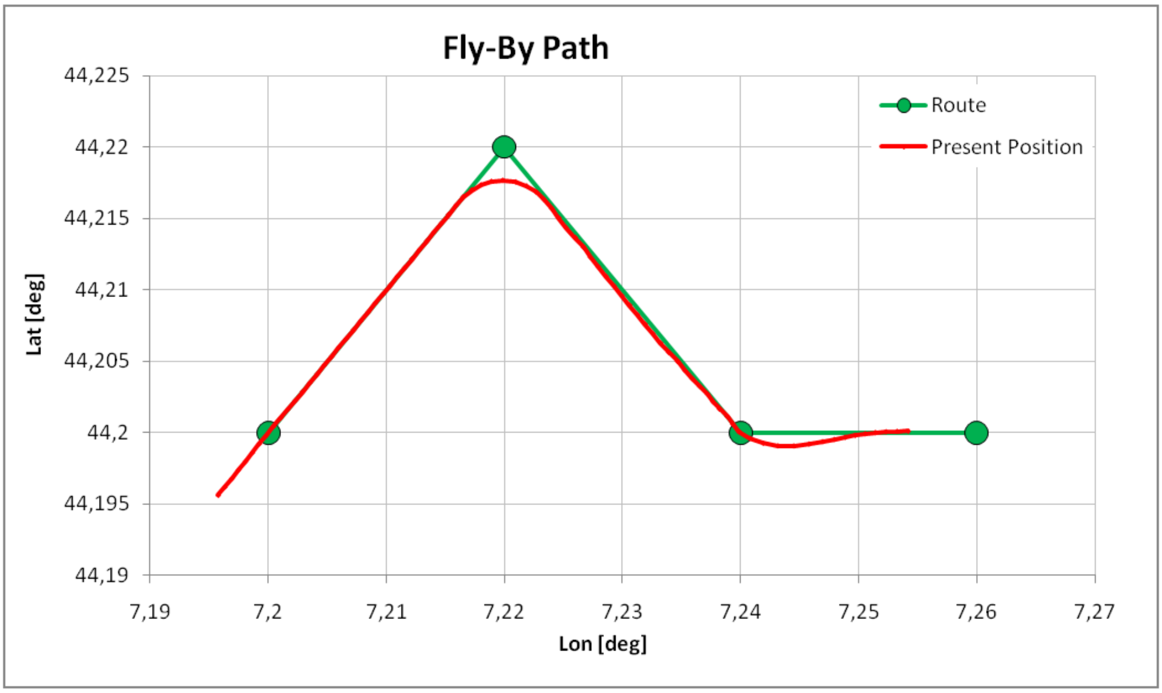


Fig 111 Simulink® test: Fly-By WP flight path.

As expected the UAV leaves the first leg before the WP2 in order to join the second leg with a smooth curve without overshoot. From the Direct Range it is possible to evaluate the Dist To Roll In calculated by the steering (560m):

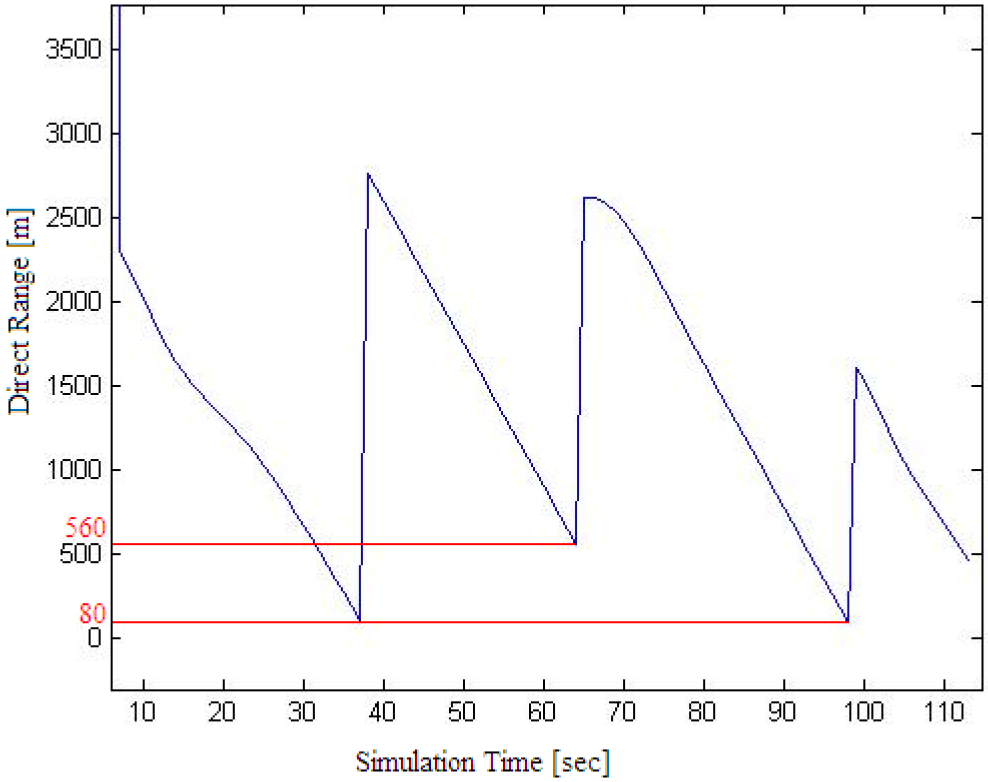


Fig 112 Simulink® test: Direct Range.

The X-Track error mark the differences between the Fly-Through WPs (WP1 and WP3) respect to the Fly-By WP2: the first and the last peaks have positive value due to the left turn and their amount is close to the change leg threshold, while the central peak has a negative value due to the right turn and its amount is close to the Dist To Roll In (to be noted that the X-Track at the change leg is not equal to the Direct Range value because the two legs are not perpendicular).

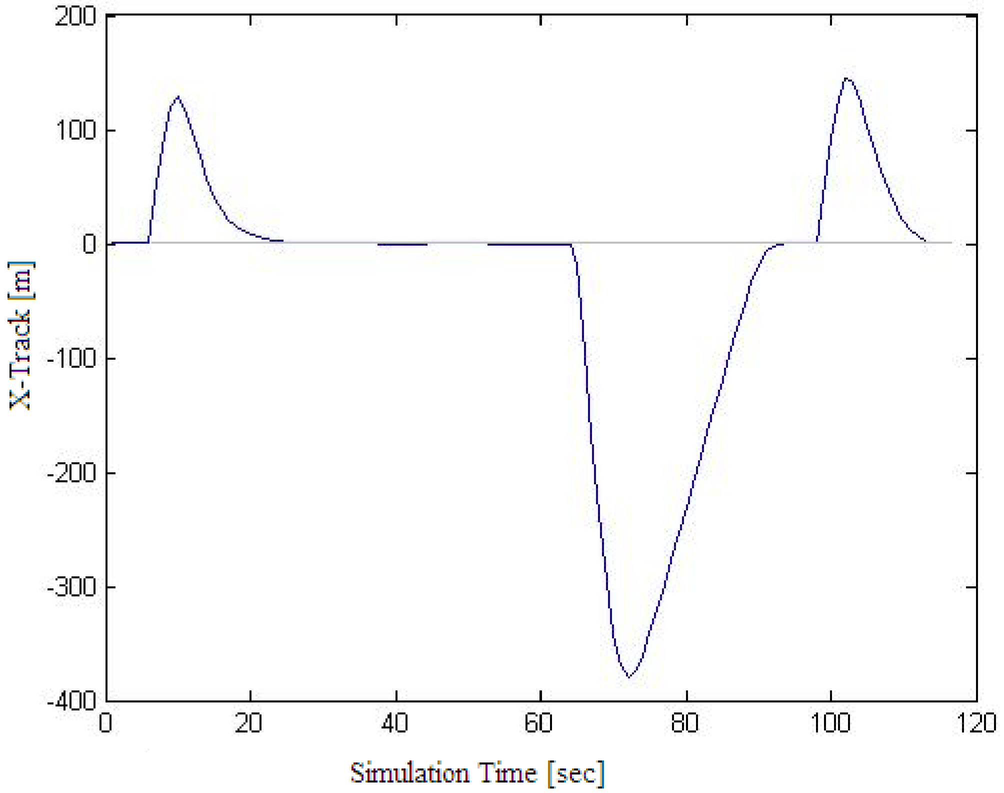


Fig 113 Simulink® test: X-Track.

10.2.4 Altitude test

The Altitude test was performed with the same route of the Fly-Through case with WP2 altitude set to 1250m (4100ft). From the Altitude time history is possible to note that the altitude is reached following the steering command (Altitude error).

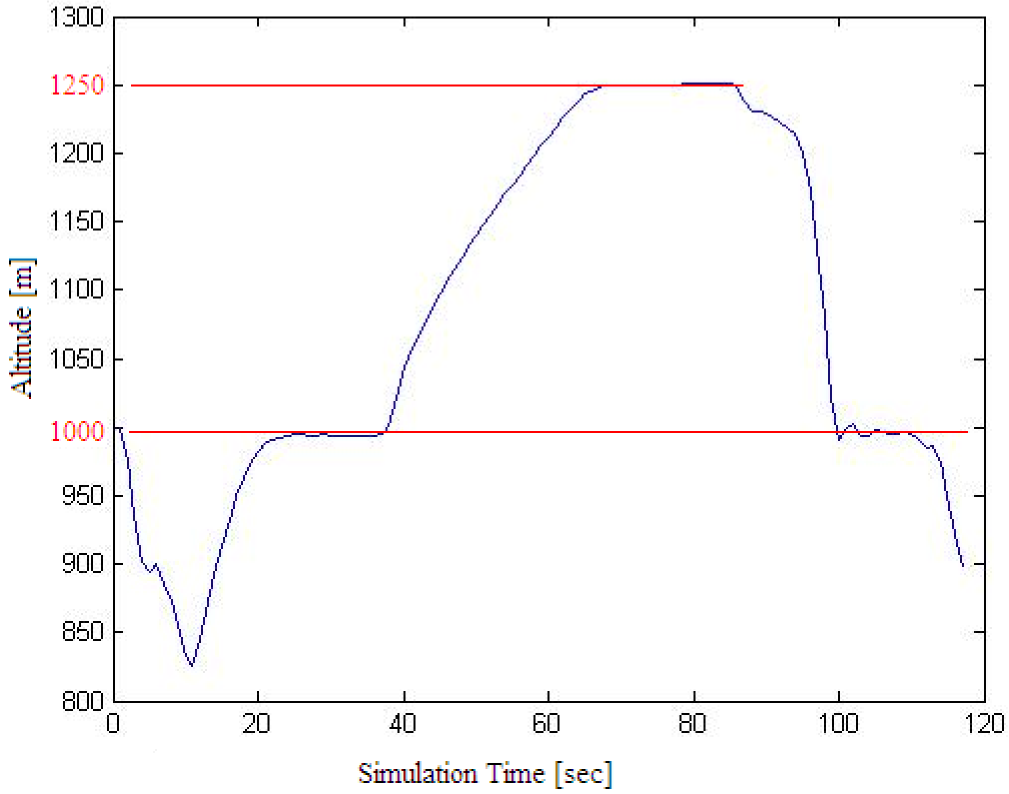


Fig 114 Simulink® test: Altitude.

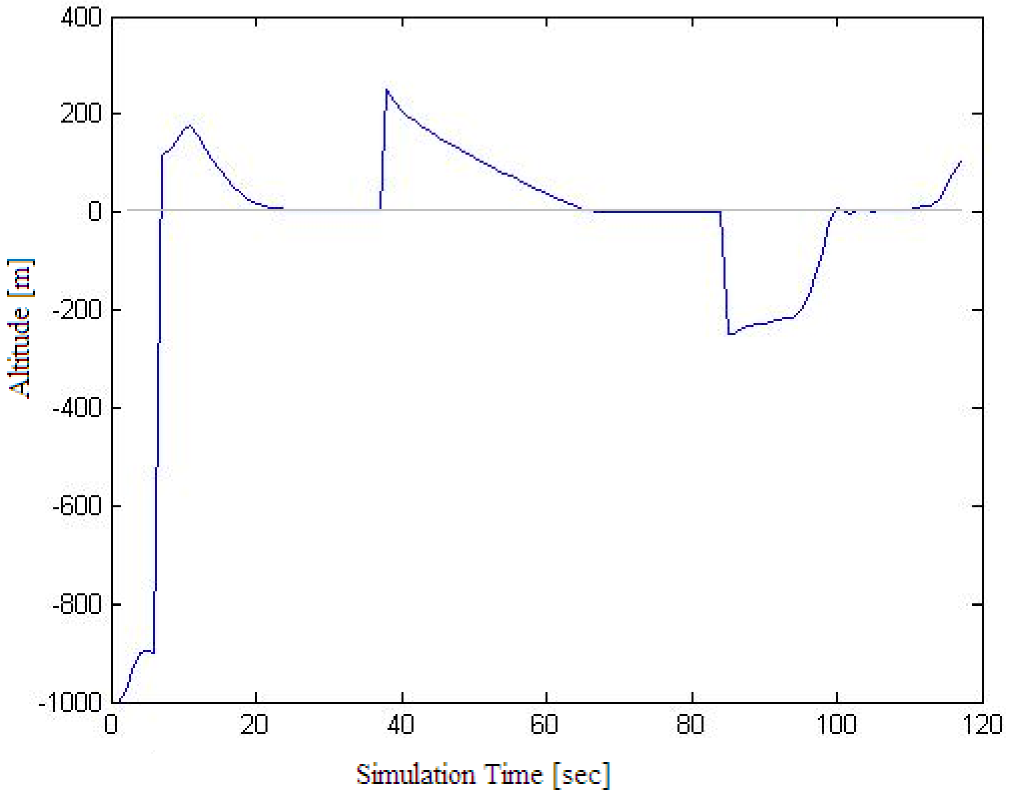


Fig 115 Simulink® test: Altitude error.

As soon as the first tests on the basic steering functions have been made, and the low level requirements for the target code basic functions have been provided, the possibility of testing and developing the functions at the Sky-Y flight simulator occurred. So that all the following development was done at such simulator and Matlab® and Simulink® were used just for comparisons during the debugging of the target code.

10.3 Sky-Y Flight Simulator results

At the Sky-Y Flight Simulator the integration of the navigation functions with the real TCS could be done for the first time. This allowed the implemented STANAG 4586 use for communicating between the ground and the board segments. So all the test of this section are valid also for the STANAG 4586 messaging test of the navigation and steering modules. Moreover the flight simulator was equipped with the real FCS control laws and the real UAV flight dynamics. In this way a more accurate study of the steering functions performance could be done for continuing the development.

10.3.1 Fly-Through WP

The test shown in this section are performed with a route similar to the one used for the Matlab® tests, just translated for avoiding the mountains:

WP	Lat [deg]	Lat [rad]	Lon [deg]	Lon [rad]	Alt [ft]	Alt [m]	IAS [kts]	IAS [m/sec]
1	44,317	0,7734	8,3242	0,1452	3280	1000	90	46,3
2	44,401	0,7749	8,406	0,1467	3280	1000	90	46,3
3	44,355	0,7741	8,4911	0,1481	3280	1000	90	46,3
4	44,350	0,7740	8,6666	0,1512	3280	1000	90	46,3

The WPs are all Fly-Through.

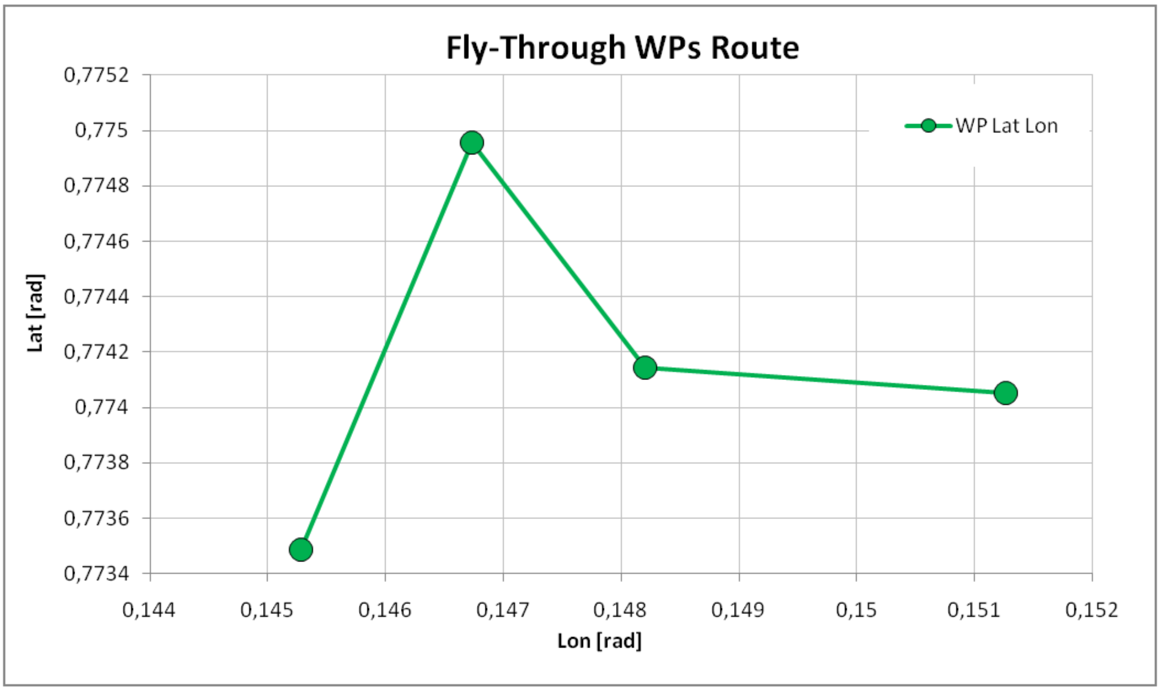


Fig 116Flight Simulator test: Planned Route.

As seen in the Matlab® test the WPs are overflown, but thanks to the real aircraft flight simulator the turning path is realistic.

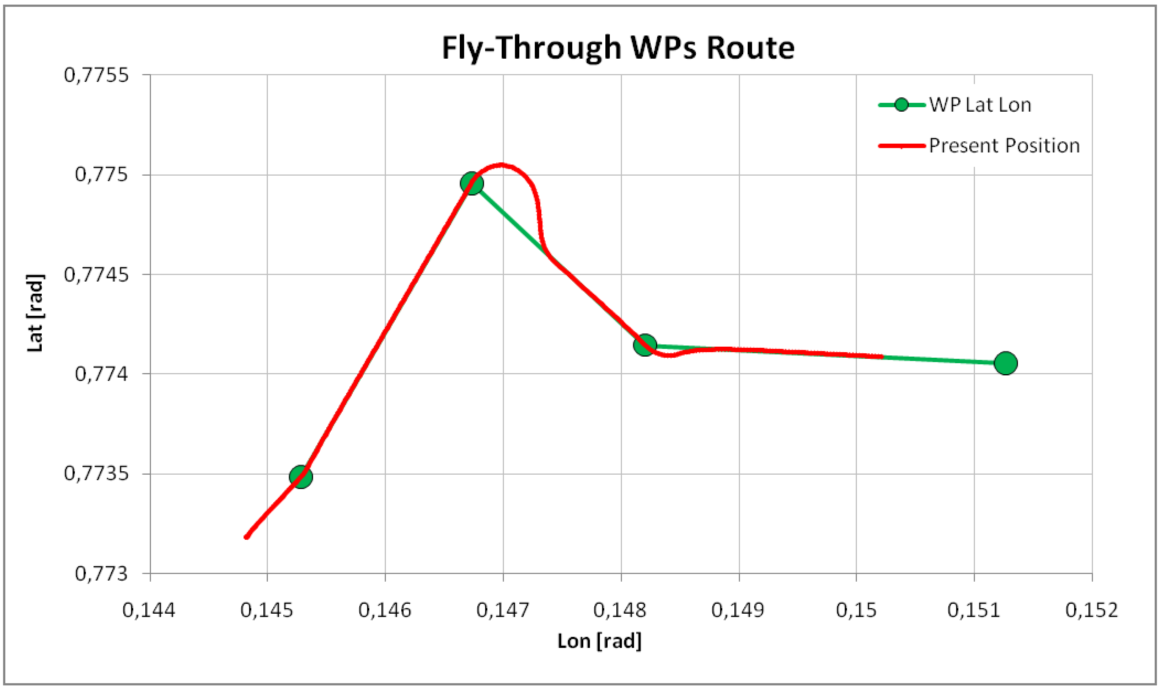


Fig 117Flight Simulator test: Flight path.

Zooming on the WP2 is observable the overshoot ant than the joint to the following leg:

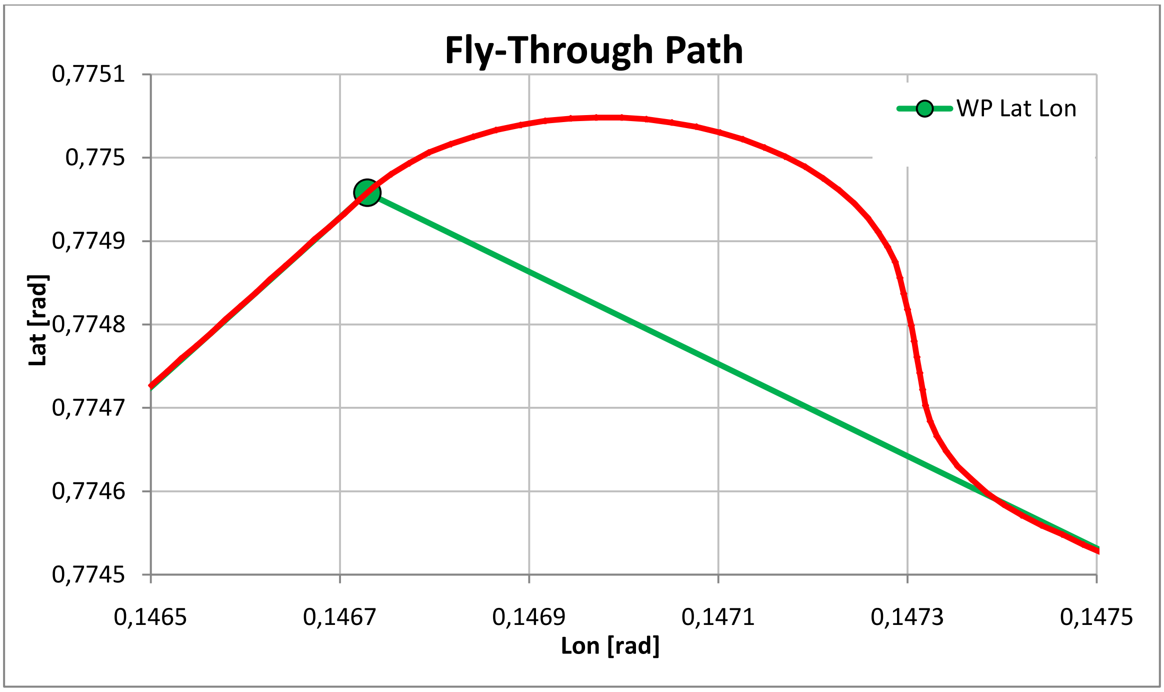


Fig 118 Flight Simulator test: Fly-Through overshoot.

The Direct Range shows the WP acquisition distance:

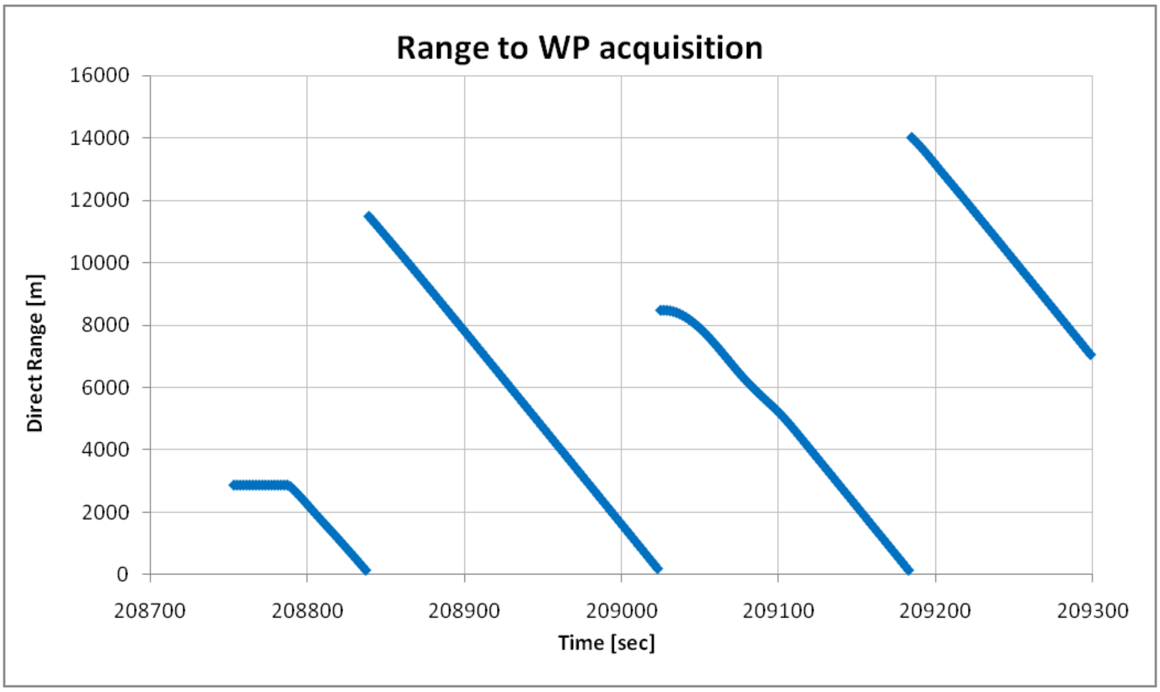


Fig 119 Flight Simulator test: Direct Range.

As expected the change leg occurs when the value of 80m is passed for all the WPs.

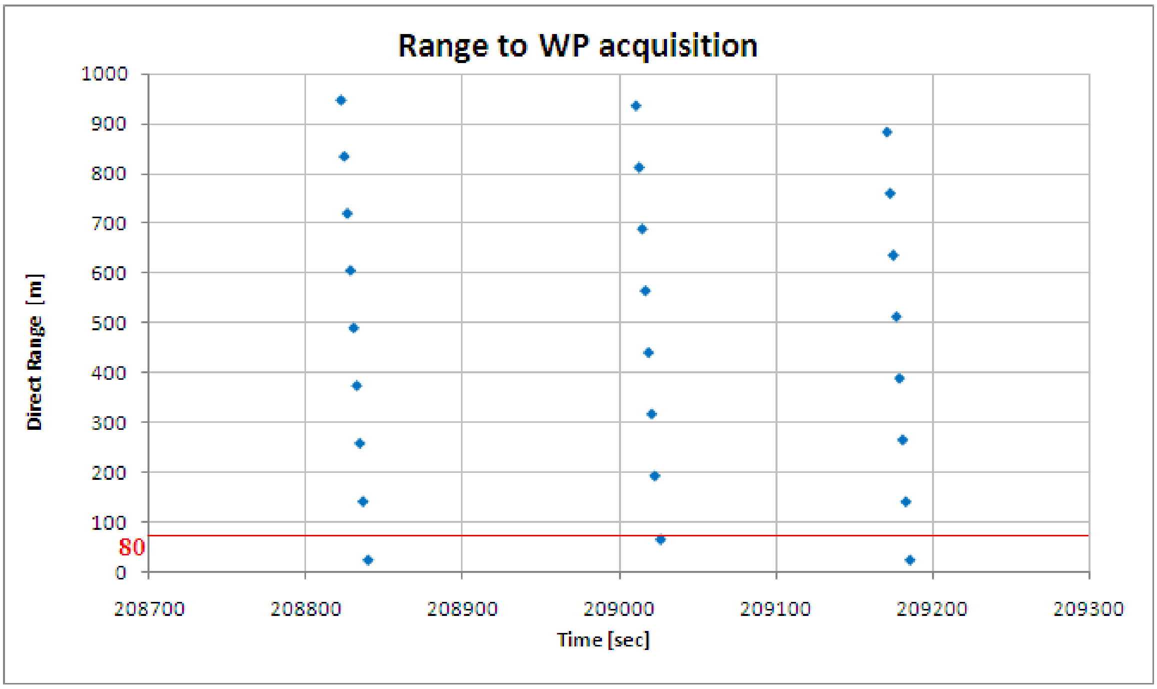


Fig 120 Flight Simulator test: Range of WP acquisition.

Thanks to the simulator is possible to evaluate also the track respect to the commanded

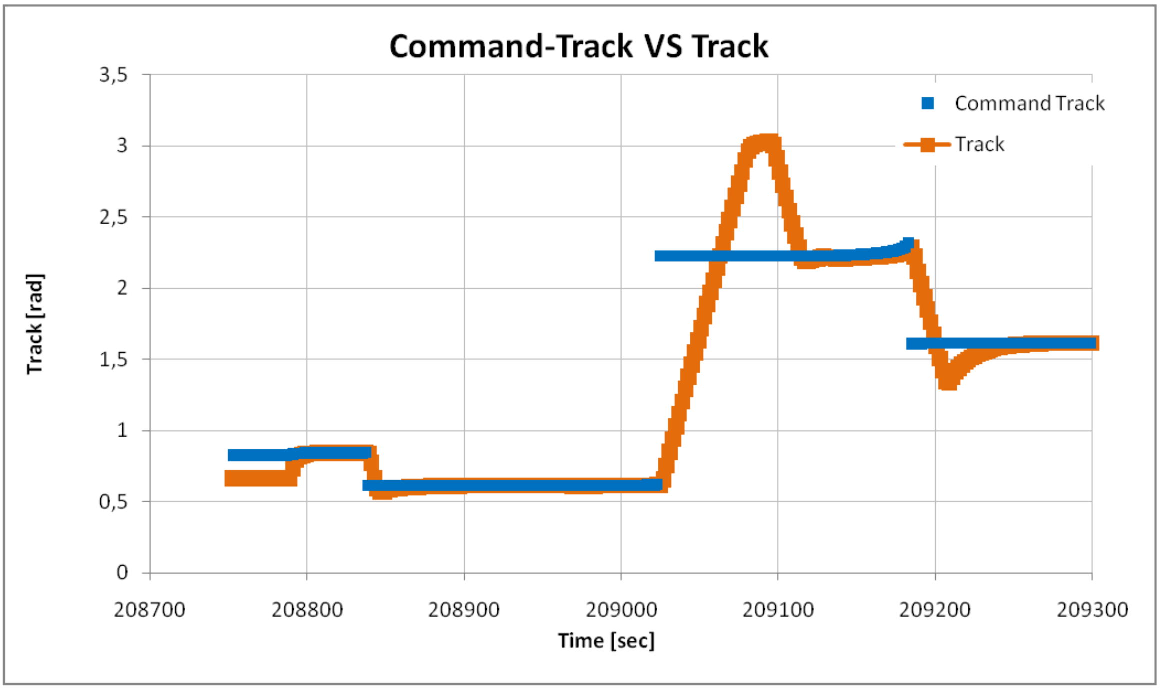


Fig 121 Simulator test: Command Track VS Track.

As described in section 9.3.2 the Commanded Track inside the cone is given by the calculation of the leg track, while the aircraft Track shows a continuous path performing the curves and converging to the leg when the X-Track is different from zero. It is notable the Fly-Through overshoot from the Track overshoot respect to the Commanded Track.

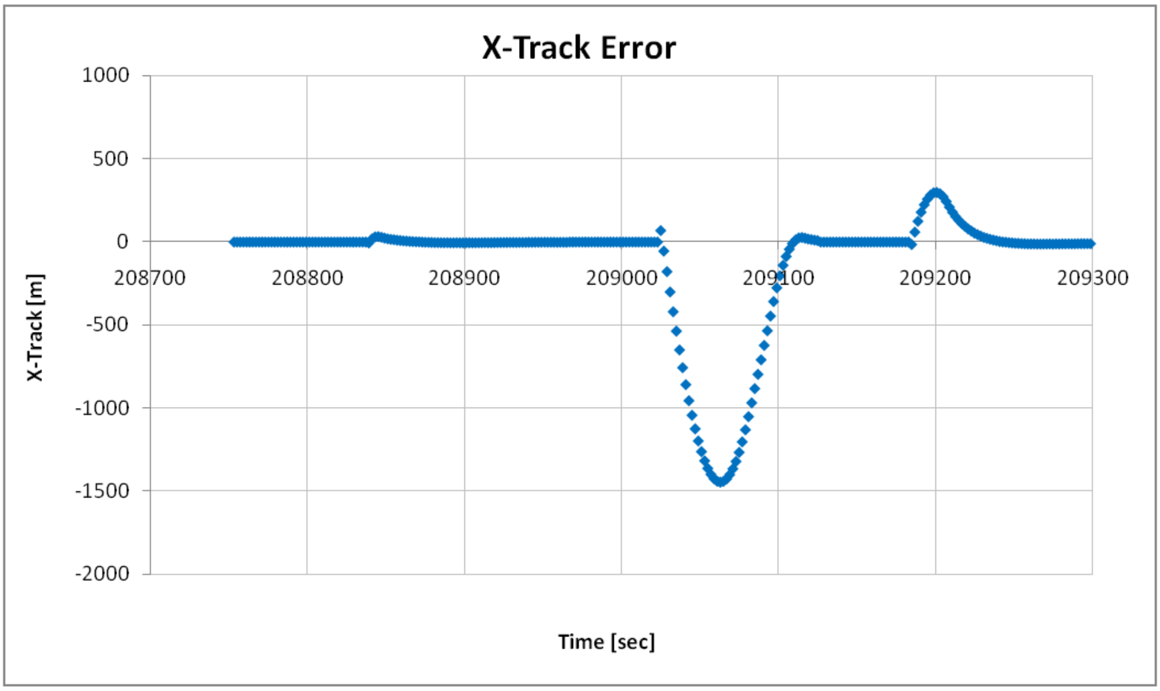


Fig 122 Simulator test: X-Track error.

10.3.2 Fly-By WP

The study of the Fly-By case was done by changing the WP2 Type attribute. All the other WPs remained Fly-Through.

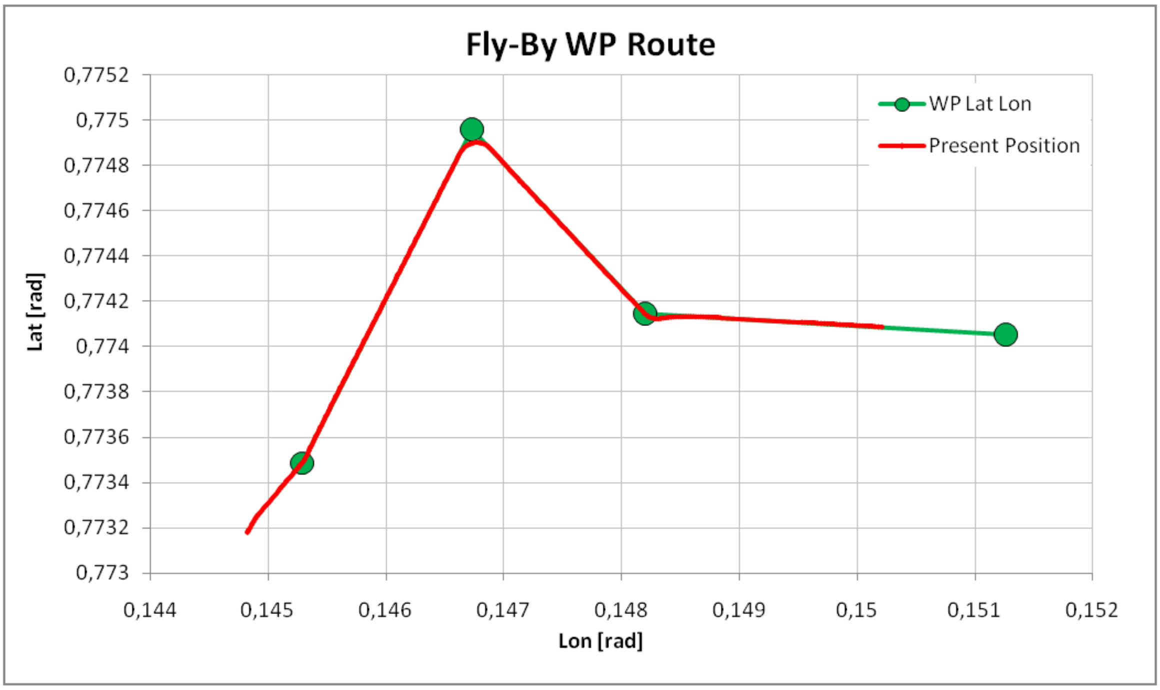


Fig 123 Simulator test: Flight path.

The flight path shows that the change leg occurred before the WP2 at a certain Roll In distance.

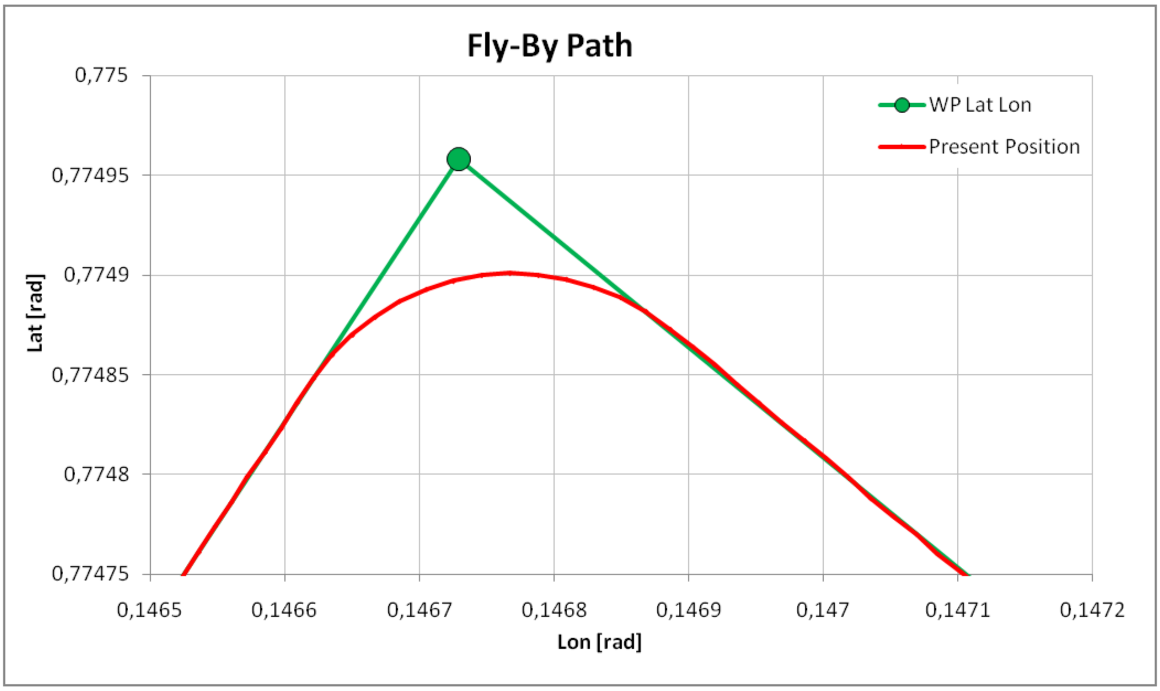


Fig 124 Simulator test: Fly-By acquisition path.

Looking at the Direct Range is evident the different acquisition distance between the Fly-Through WPs and the Fly-By WP2

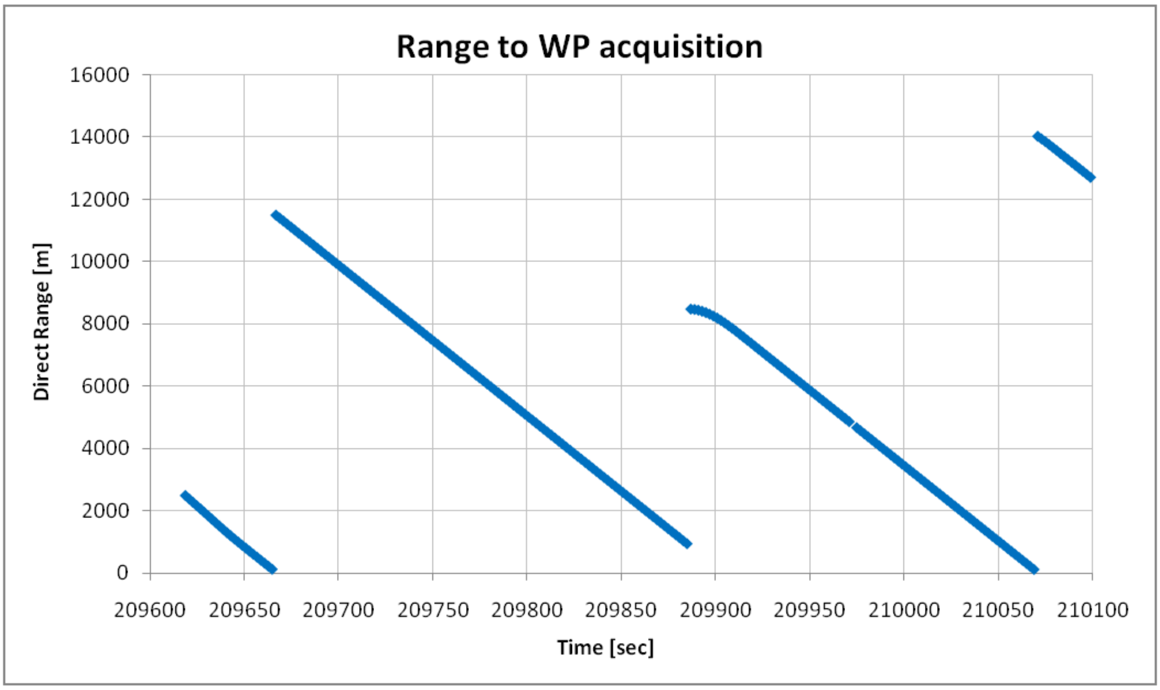


Fig 125 Simulator test: Direct Range.

The Dist To Roll In for the WP2 calculated by the steering in that condition resulted to be about 850m as seen by the graph; while for WP1 and WP2 the change leg is always under 80m:

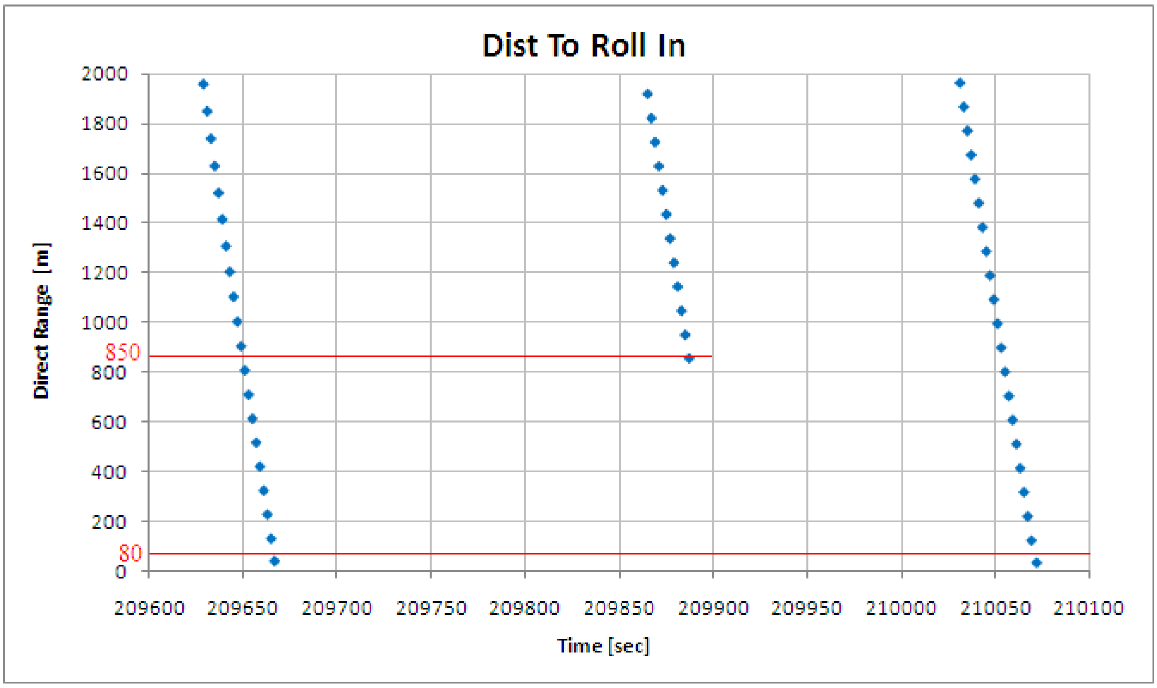


Fig 126 Simulator test: Dist To Roll In.

From the comparison between the Track and the Command Track is possible to note the absence of overshoot passing from the first leg to the second due to the Fly-By Dist To Roll In correct calculation.

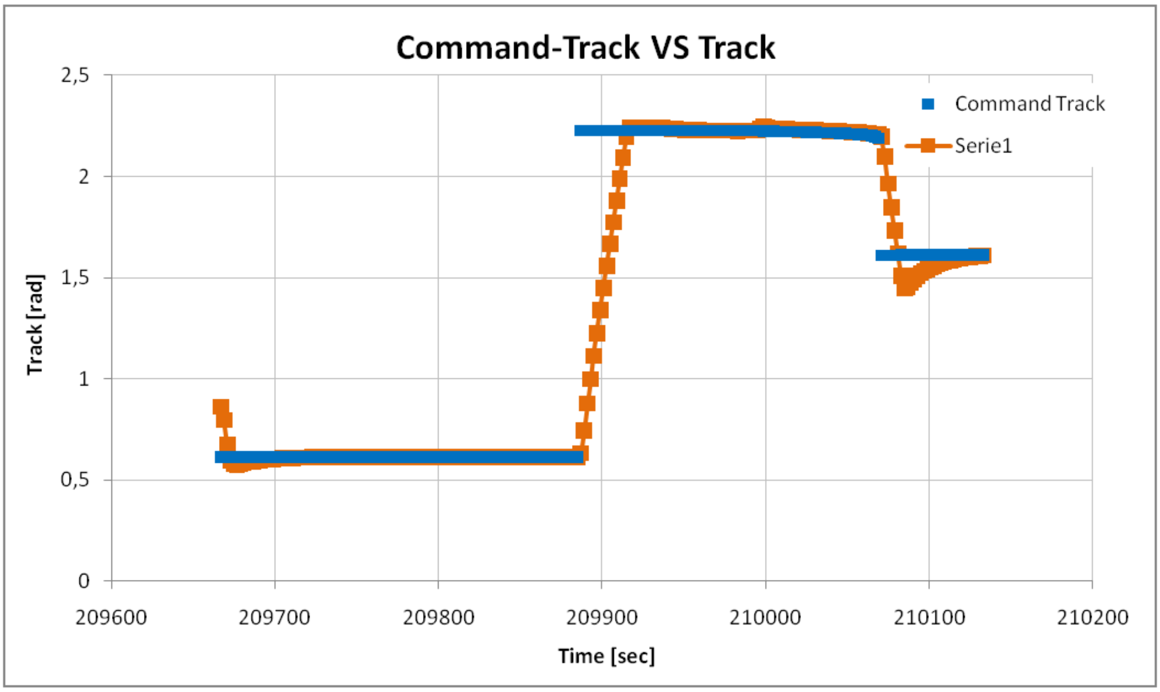


Fig 127 Simulator: Command Track VS Track.

From the X-Track error recorded data result evident the Fly-By leg change by the high value of X-Track crossing the WP2. Such value, as the two legs are quite perpendicular, is very close to the Dist To Roll In.

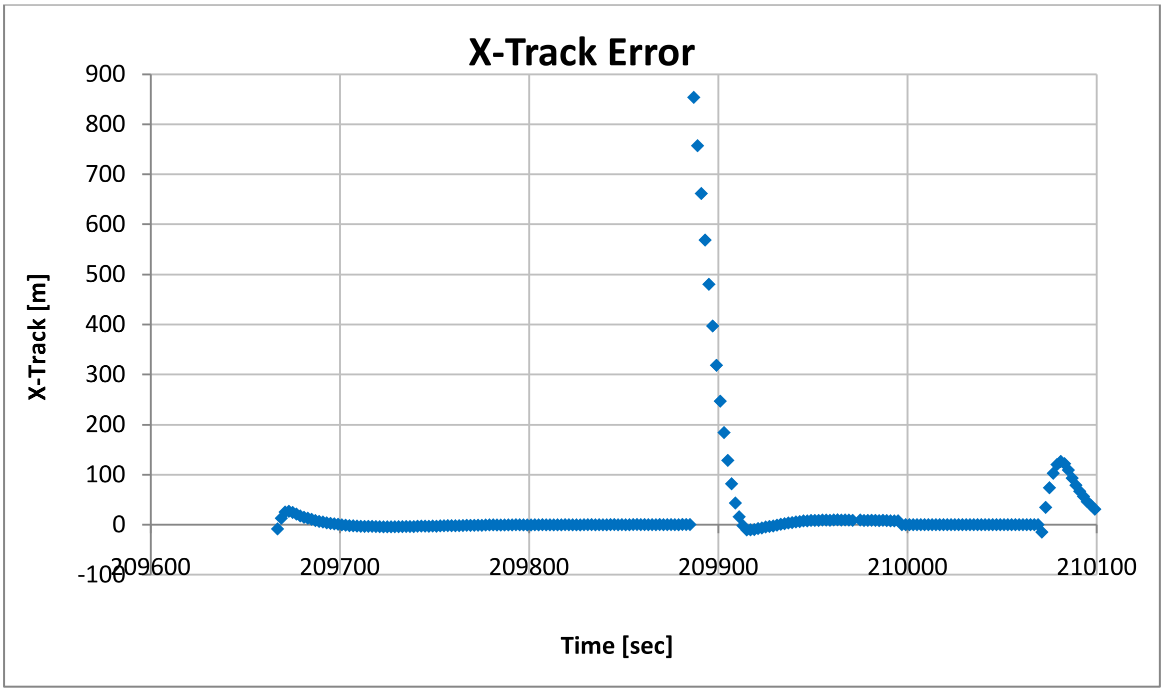


Fig 128 Simulator test: X-Track Error.

10.3.3 Altitude

The same route was used for the altitude test. Two test were performed, in the first the WP2 altitude was set to 2000m (6560ft) and in the second to 1250m (4100ft), all the other WP's altitude attribute was set to 1000m (3280ft). In the first case the UAV doesn't have enough space to reach the altitude before the WP2 acquisition, while in the second the altitude is reached.

First test (2000m)

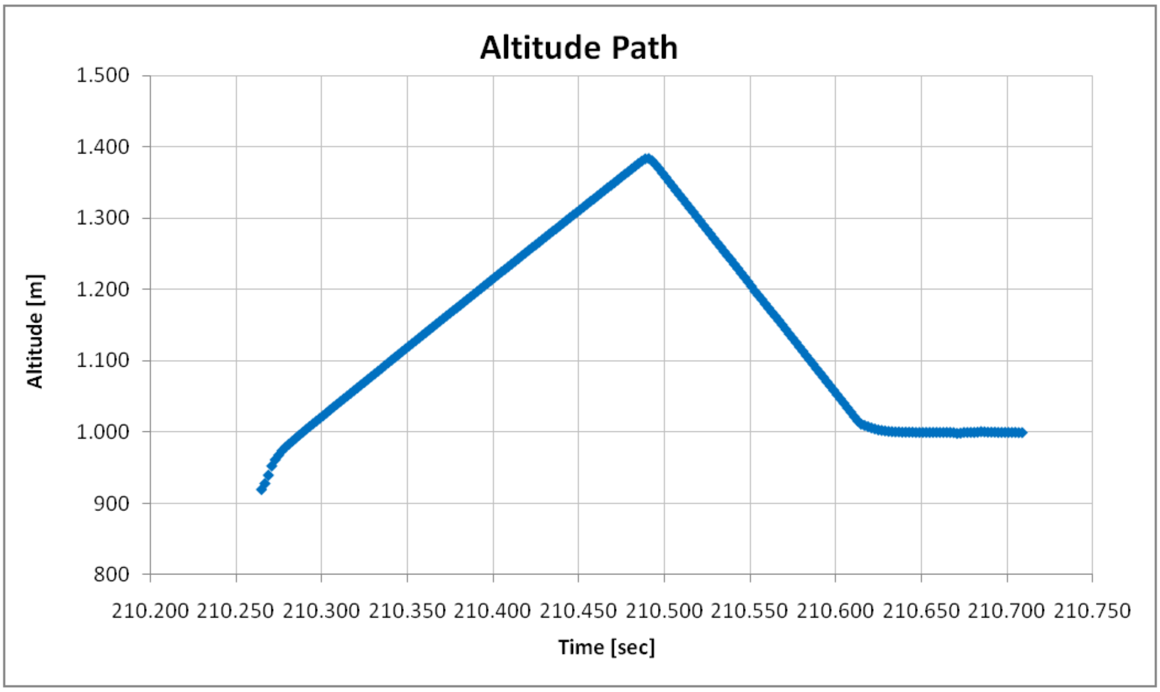


Fig 129 Simulator test: First test altitude path.

From the figure is possible to see that the UAV start the climb, but the WP2 acquisition occurs when a maximum altitude of only 1380m is reached. Comparing the altitude path to the altitude error calculated by the steering can be noted that the demand was correct, but the altitude gap was too much, compared to the distance, for the simulated UAV performances. The altitude error in fact never reach zero, so the steering demand was to continue the climb. On the contrary the descent to return at 1000m altitude shows that the steering command goes correctly to zero when the UAV reach the WP3 altitude.

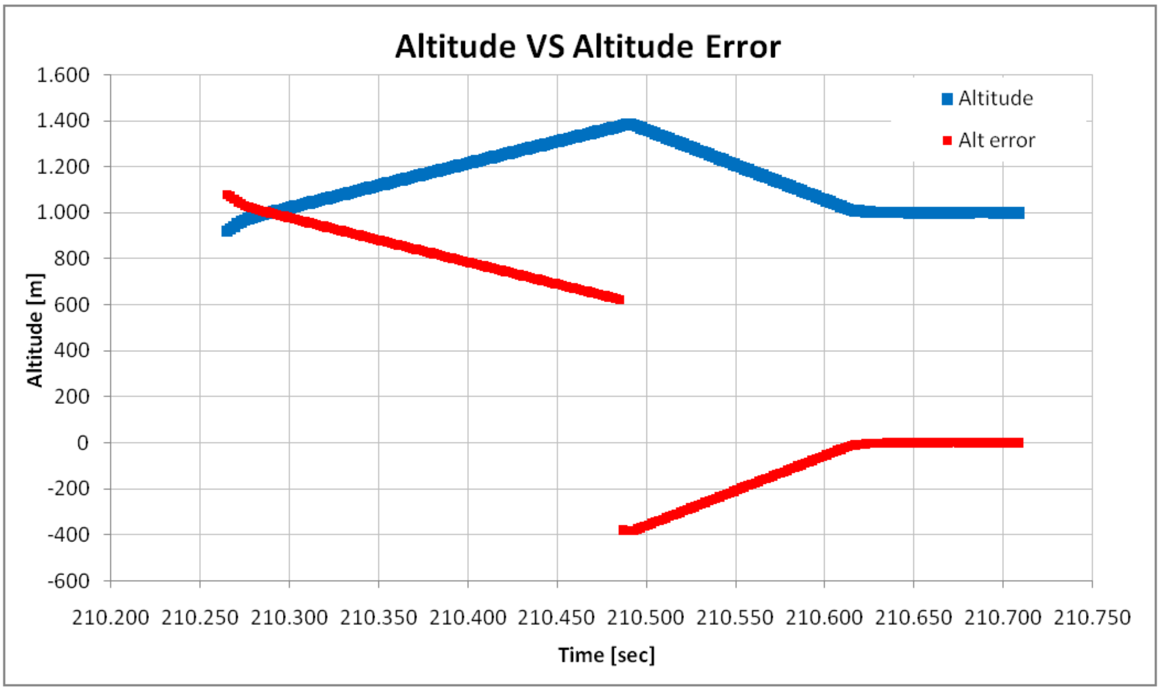


Fig 130 Simulator test: First test altitude VS altitude error.

Second test (1250m)

In the second test the altitude is reached.

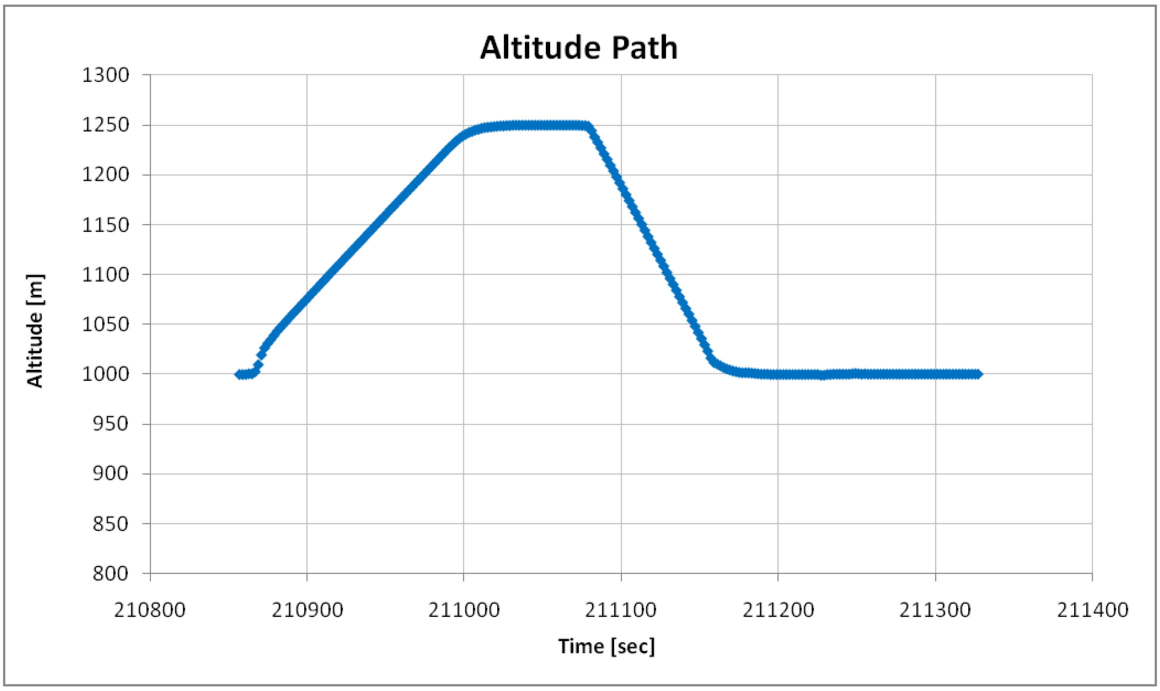


Fig 131 Simulator test: Second test altitude path.

The altitude error steering output command correctly to the FCS the gap to annul both in climb and in descent. The reached altitude is kept constant until the WP is acquired.
In Fig 132 the UAV altitude is shown in blue (left scale) and the altitude error in red (right scale).

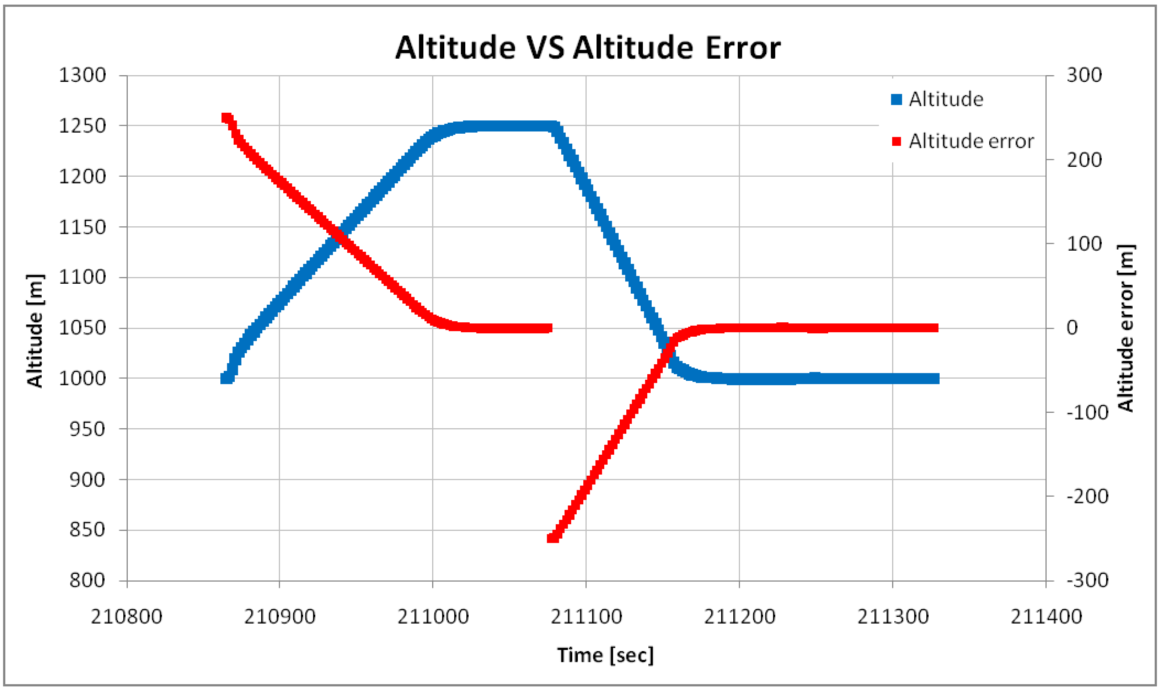


Fig 132 Simulator test: Second test altitude VS altitude error.

Last consideration about the altitude is about the associated speed. As seen in section 9.3.2.3 during the climb the FCS fixes the IAS to 95kts while during the descent fixes the vertical velocity independently from the WP speed attribute. This behavior is respected from this test as can be seen in Fig 133:

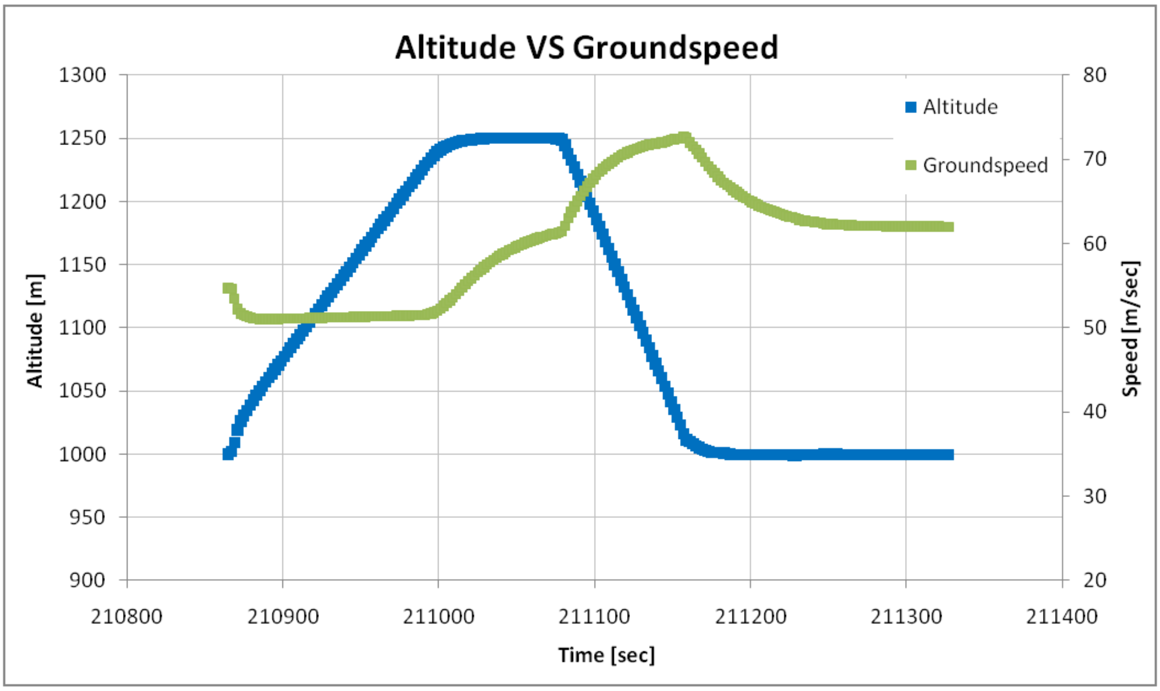


Fig 133 Simulator test: Second test altitude VS groundspeed.

Until the UAV is climbing the speed is constant with a groundspeed of about 51m/sec (about 99kts as the test is not at sea level). When the altitude of 1250m is reached the steering command the groundspeed calculated on the WP2 IAS of 110kts (corresponding to a groundspeed of about 60m/sec at 1250m altitude). At the change leg the diving at constant vertical speed brings the UAV to increase the groundspeed until the WP3 1000m altitude is reached. Then the speed decrease to the WP3 peed and remains constant for all the leg.

10.3.4 Speed

A pure speed test was performed by changing the IAS attribute of WP2 to 100kts (51.4m/sec), while the other WP's IAS were set to 90kts (46.3m/sec).

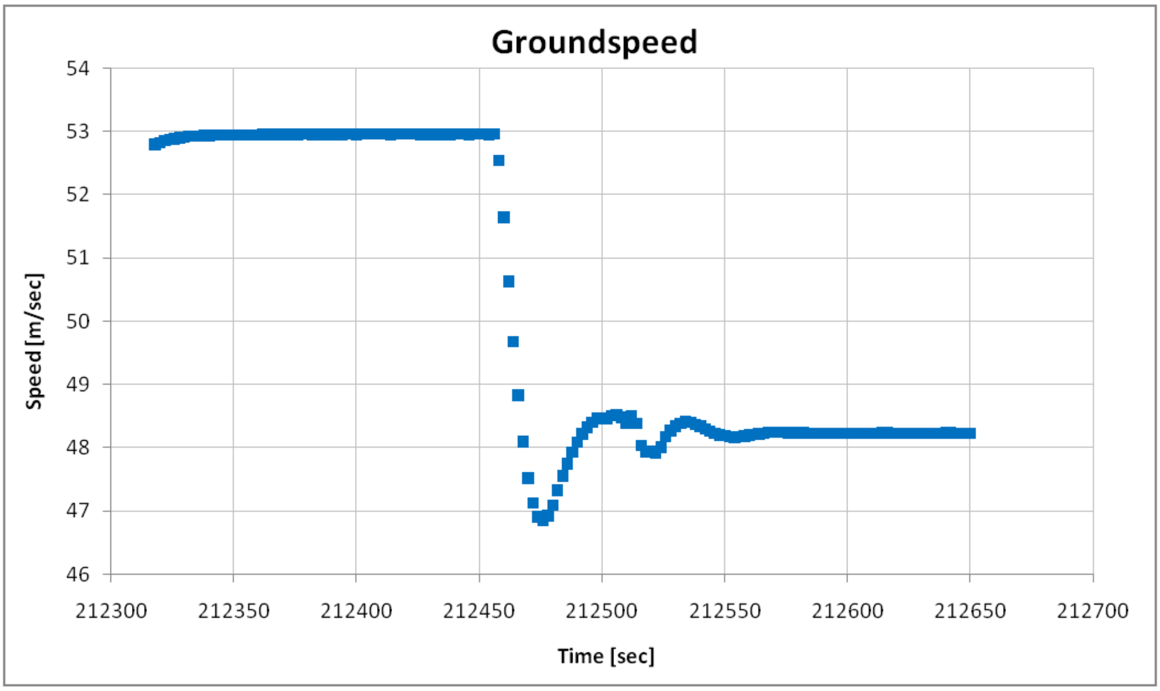


Fig 134 Simulator test: Groundspeed path.

During all the first leg the groundspeed of 53m/sec is kept (resulting groundspeed of a IAS=90kts at 1000m of altitude). During the change leg the speed is decreased, some oscillations around the final value are observable due to the coupling maneuvers of turning and changing altitude, this effect in fact is present only along the turning maneuver as can be seen from the track recorded data in Fig 135. After the latero-directional stabilization the speed is kept constant to the groundspeed value of about 48m/sec corresponding to the WP3 IAS of 90kts at 1000m of altitude.

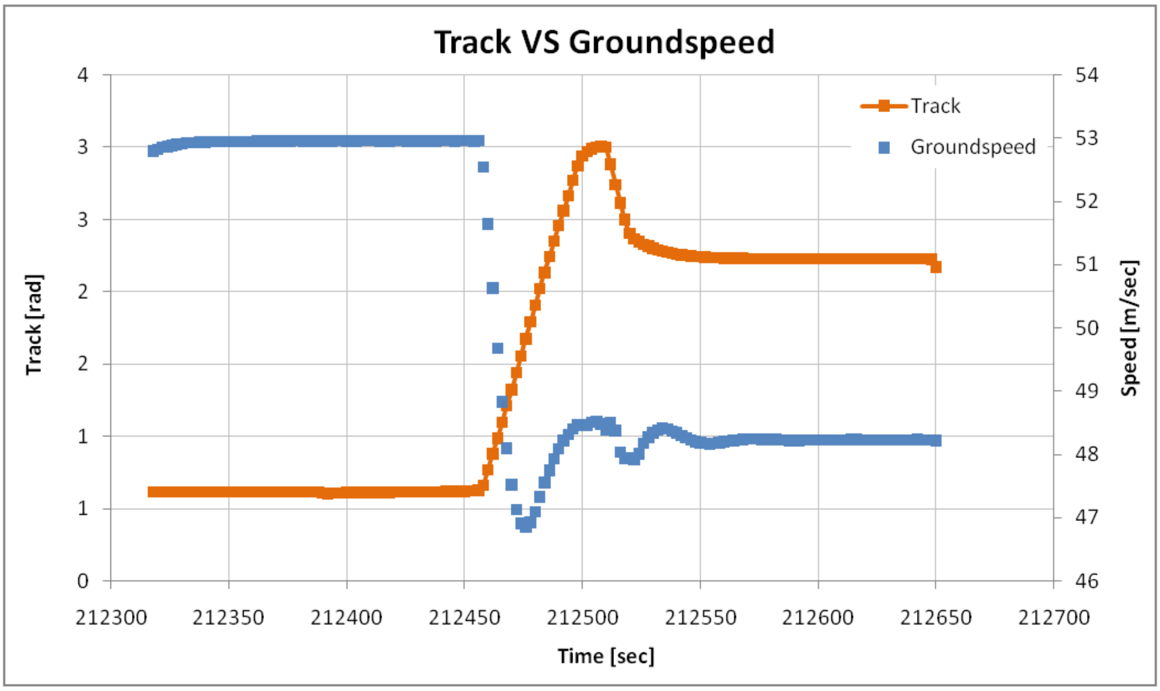


Fig 135 Simulator test: Track VS groundspeed.

10.3.5 Loiters

Thanks to the simulator and the possibility of fruit the real UAV flight dynamics, also the loiters flight path could be tested.

Circular Loiter

The route loaded was always the same used for the previous simulator test. In this case the WP2 is a clockwise loiter of circular shape with radius 1.5nm, altitude of 1000m as the other WPs and loiter time of 9 minutes.

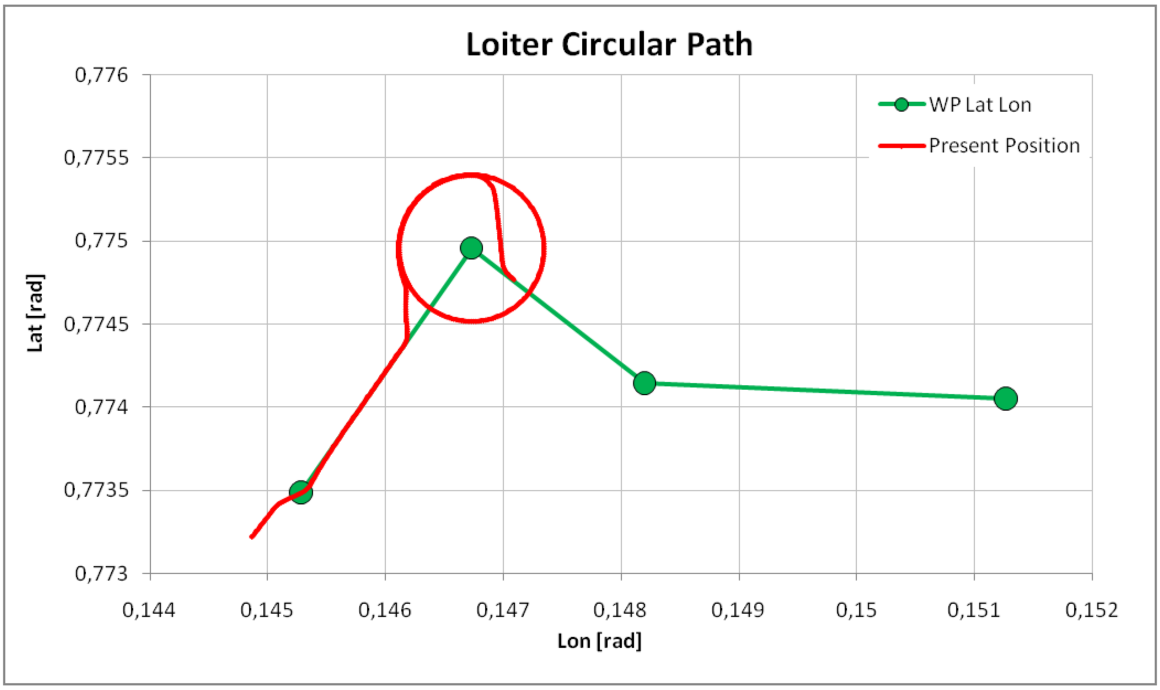


Fig 136 Simulator test: Loiter circular flight path.

As expected the UAV leaves the leg before the circular path not to approach the circle perpendicularly. Then the path is performed around the WP2 at a distance of 1.5nm; at the end of the loiter time the UAV exit the loiter and approach the next leg.

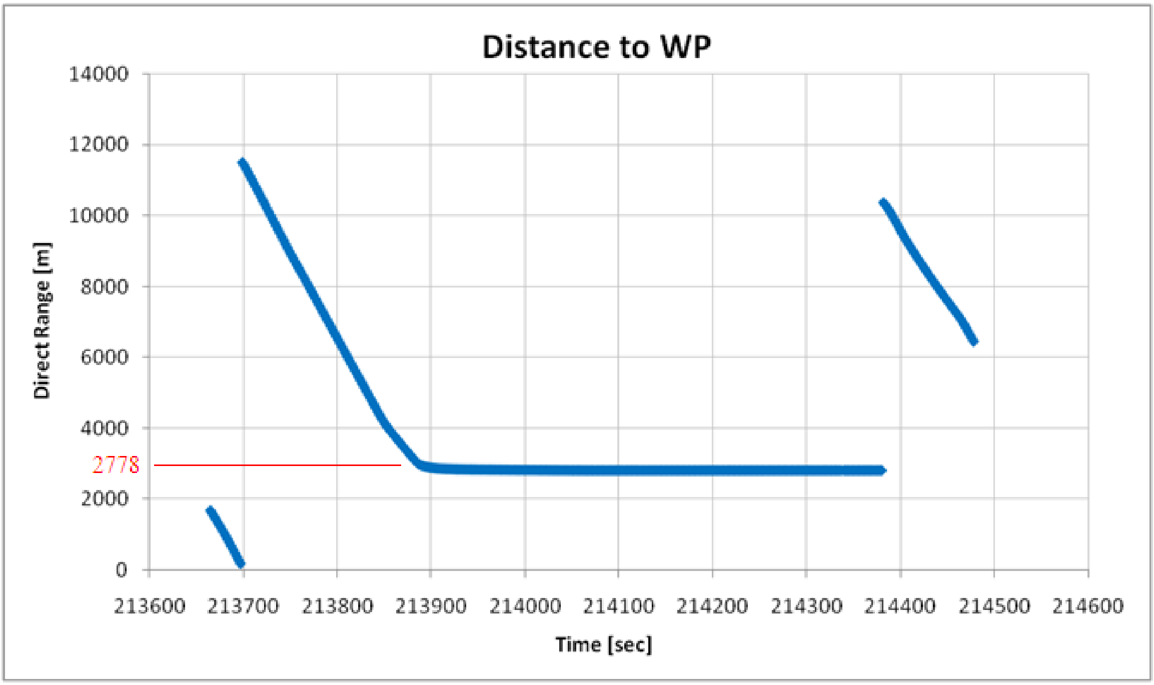


Fig 137 Simulator test: Direct range.

From the Direct Range recording is possible to note the distance of approach to the loiter and the distance of the UAV from the WP2 of about 2778m which is exactly the loiter radius value of 1.5nm. In Fig 138 is compared the altitude with the altitude error from which is possible to see that, after the climb of 200m due to the starting UAV position, the altitude remains constant for all the flight, even during the loiter. there are only few oscillations during the two maneuvers for jointing the loiter circle.

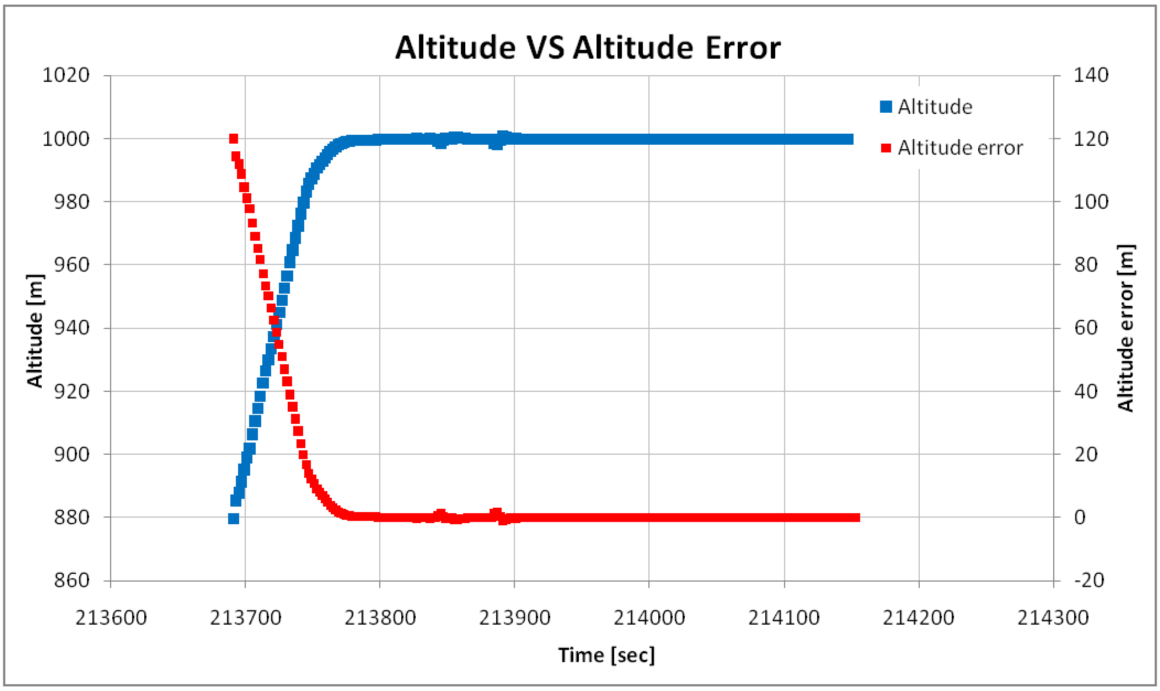


Fig 138 Simulator test: Altitude VS altitude error.

Racetrack Loiter

For the racetrack loiter test the previous route was used changing the WP2 attributes to a racetrack counter-clockwise loiter with a radius of 1.5nm, 2nm of length, 0deg of bearing and 9 minutes of loiter time. Moreover the WP2 IAS was increase to 100kts.

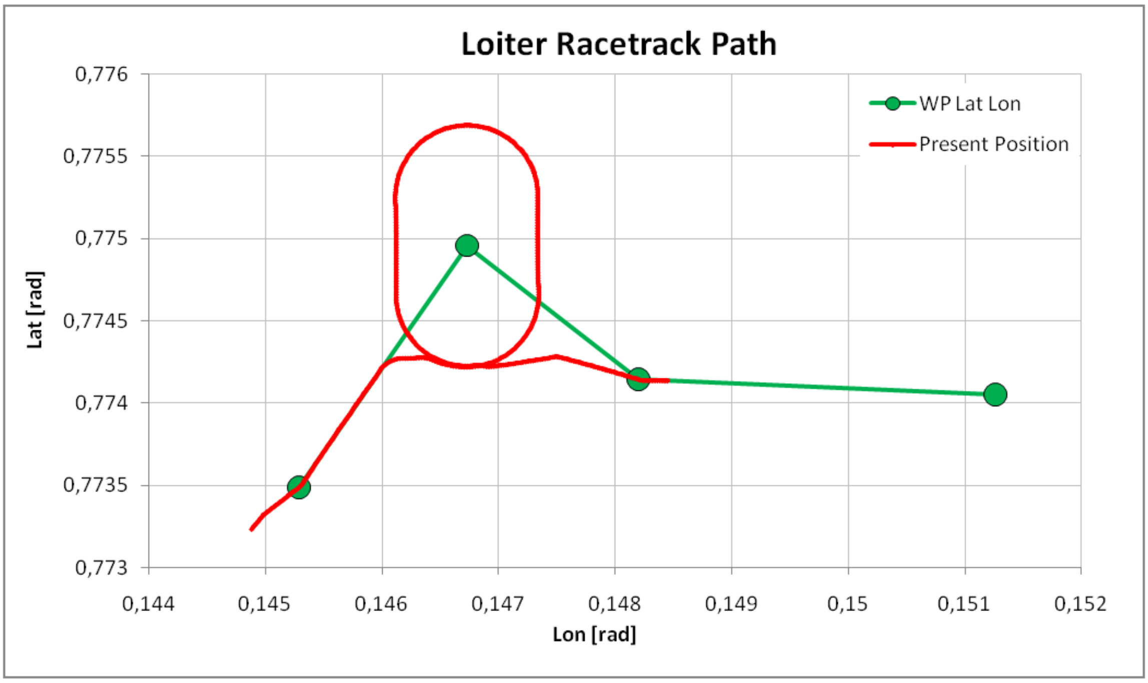


Fig 139 Simulator test: Racetrack loiter path.

The path shown in Fig 139 describe the loiter flight where the acquisition of the loiter path starts from the section from which the UAV was approaching as described in section 9.3.2.1.2.

It is interesting to analyze the Direct Range recording for this kind of loiter.

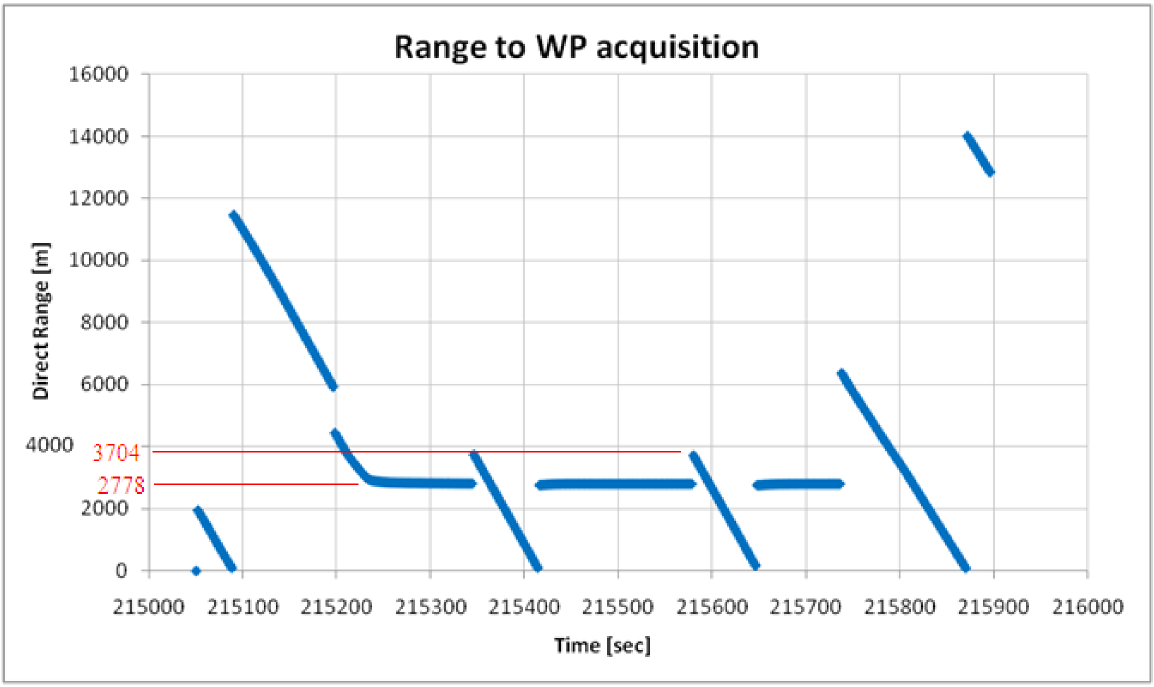


Fig 140 Simulator test: Distance to WP.

As usual the WP1 is acquired at 80m, then, going in the WP2 direction, the leg is left at a distance allowing the UAV to join the loiter with a smooth curve. But what is interesting to note is the behavior of the steering algorithm during the loiter path. As described in section 9.3.2.1.2, the steering re-plan automatically the route substituting for all the loiter time the original route with a calculated route forming the loiter path. From the Direct Range is possible to see that to complete one loiter turn, the aircraft flies two times a straight leg (two central inclined segments) which are the straight segments of the racetrack, while the three horizontal segments represent the moment in which the UAV was flying the loiter curves. The Direct Range values in fact are 3704m at the beginning of the straight legs which corresponds to the 2nm loiter length; and 2778m value which represents the 1.5nm loiter radius, and it remains constant for all the circular sections of the loiter path.

Last consideration is on the speed. The WP2 IAS was set to 100kts while other WP's IAS was set to 90kts. The graph of Fig 141 plot the groundspeed recording of the test starting from the WP1 acquisition: at the beginning the UAV was flying at the WP1 IAS (corresponding to about 48m/sec at 1000m of altitude), as soon as the WP1 is acquired the speed increase to about 53m/sec and rests constant to this value for all the second leg, the loiter approach, and all the loiter sections path; then the speed is decreased to the WP3 value when the loiter time ended. There are some oscillations of the speed just during the acquisition curves and, after the loiter exit, during the curves for directing the UAV to the WP3, but since the oscillation range is less than 1m/sec (less than 2kts) are considered acceptable..

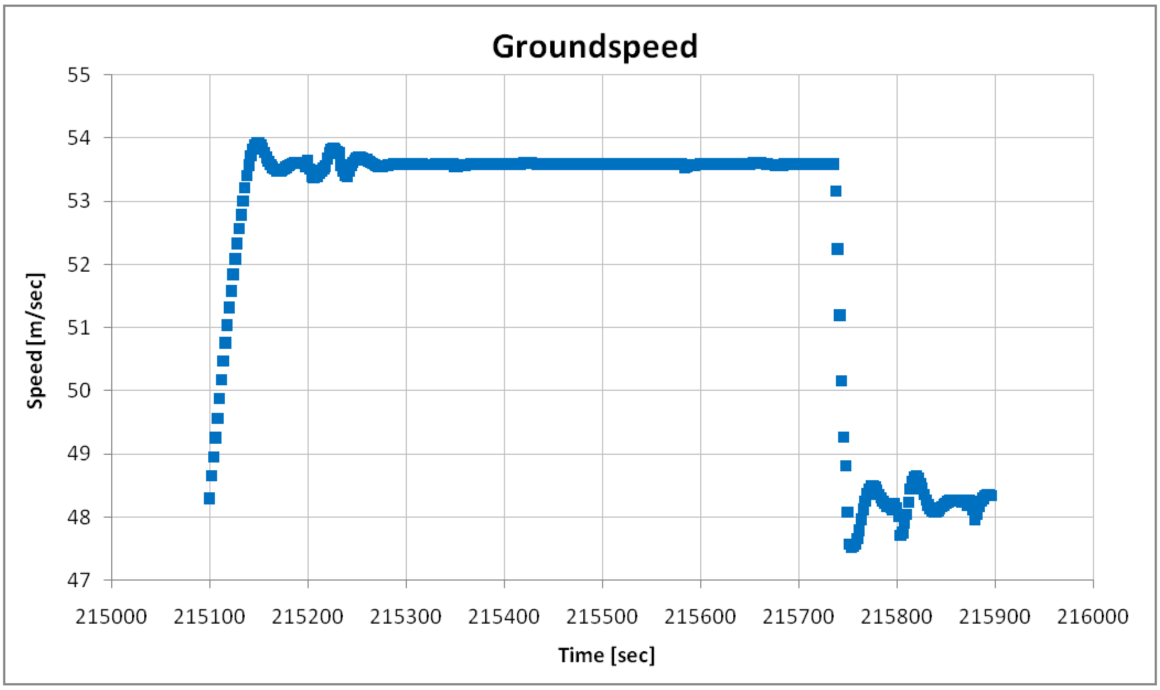


Fig 141 Simulator test: Groundspeed recording.

Zooming the graph and isolating the loiter path groundspeed recording is possible to note the speed oscillation at the acquisition of the different loiter sections. It is supposed to be variable passing from a straight segment to a circular path and vice versa due to the augment and decrease of drag due to the UAV inclination. The result underline the the speed variation is soon compensated with a very short range of oscillation if compared to the scale of the graph (Fig 142).

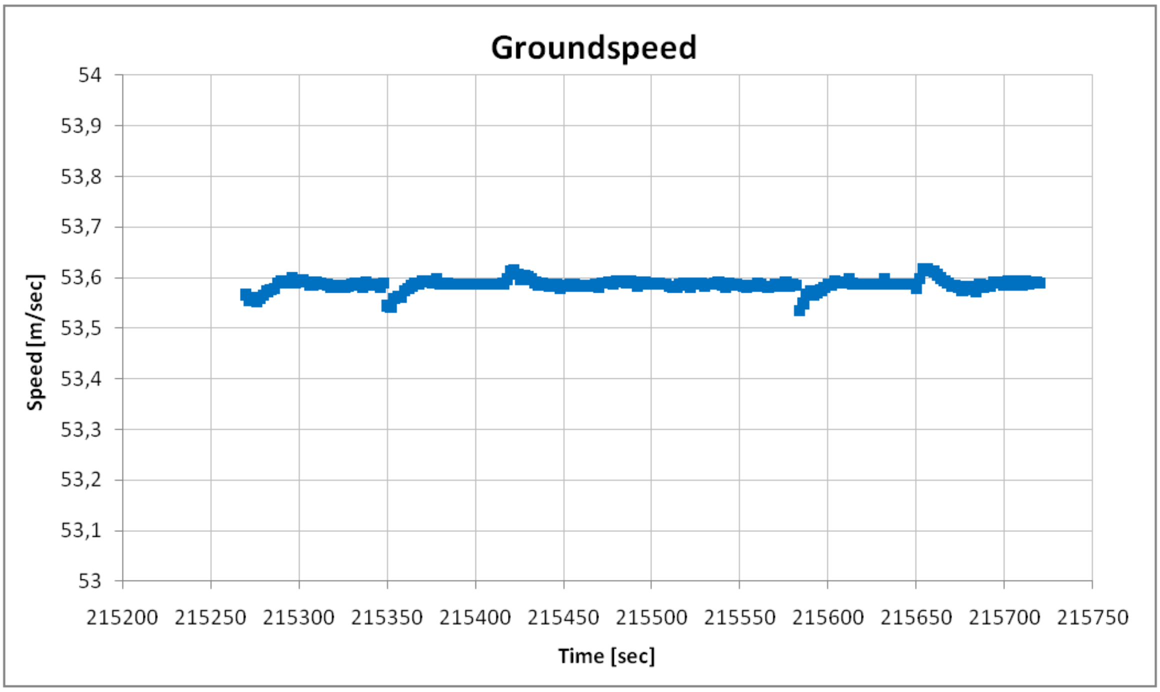


Fig 142 Simulator test: Loiter groundspeed recording.

Figure8 Loiter

The last test on the loiter is about the Figure8, this path is introduced by the STANAG 4586. The same attribute of the previous route were used except from the loiter Type.

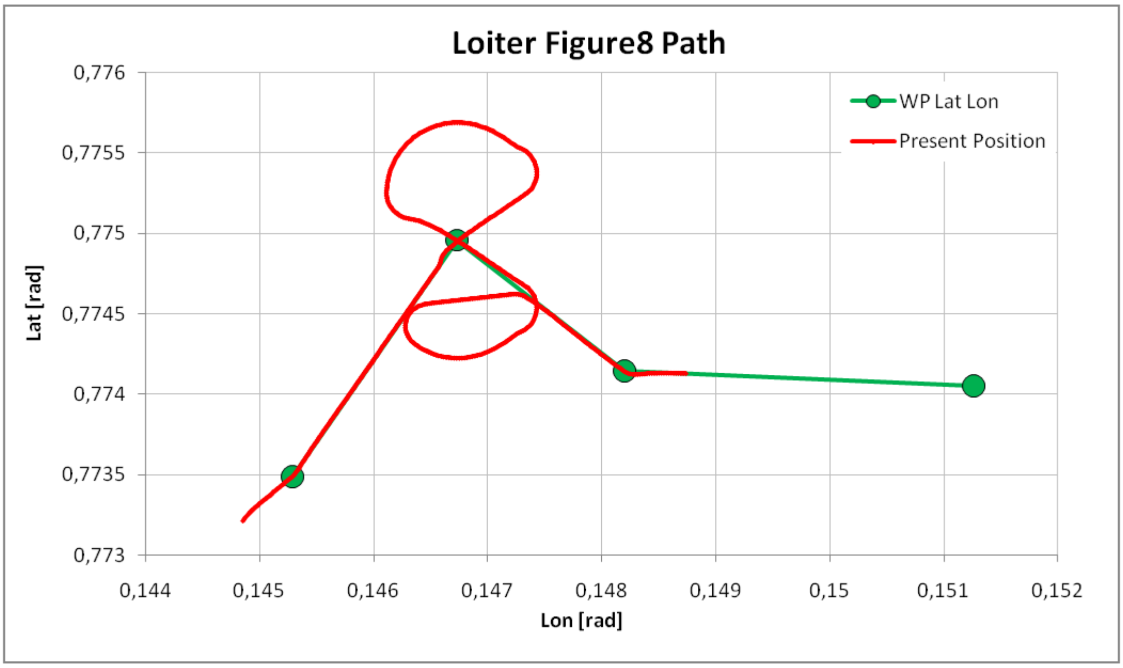


Fig 143 Simulator test: Figur8 loiter path.

The path of Fig 143 shows the 8 shape of the loiter, due to the high curves ratio compared to the speed and UAV performances, are observable some overshoot at the joint of the loiter curves with the straight segments (Fig 144).

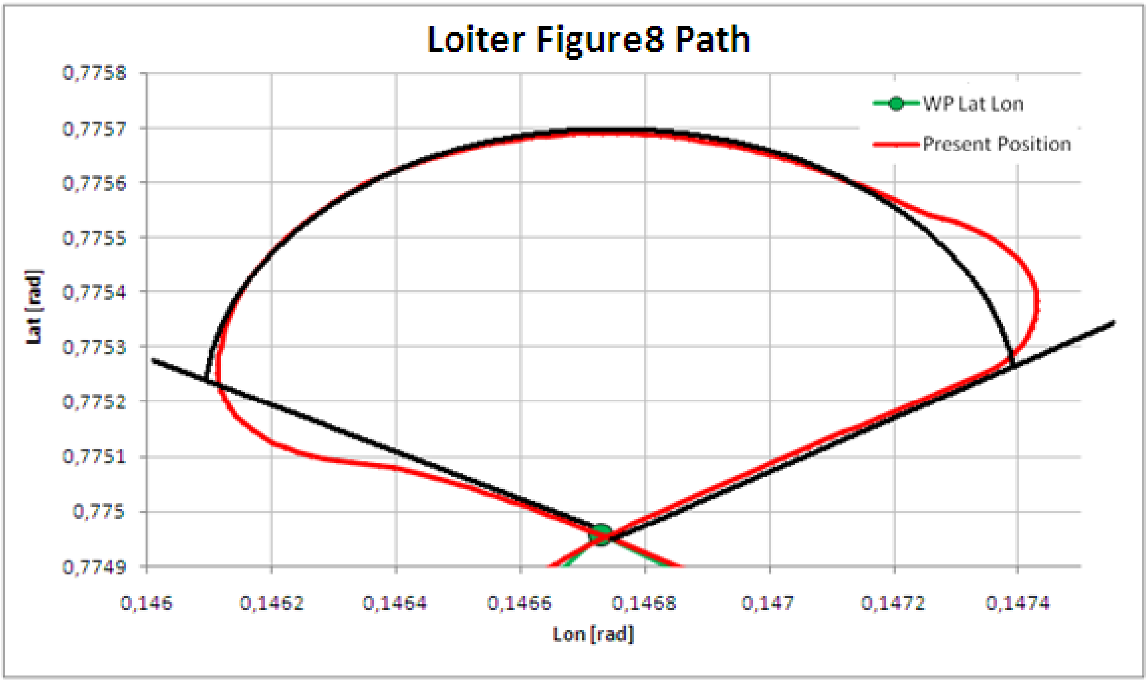


Fig 144 Simulator test: Figure 8 loiter path overshoot.

Anyway the shape is completed correctly and at the end of the loiter time the UAV engage the second leg of the route to reach the WP3 as prescribed.

Looking the Direct Track recording the overshoot is present in the peaks at the beginning of the two horizontal segments representing the distance from the center of the curve of the two figure8 circular sections. After the overshoot anyway the distance value is 2778m which correspond to the loiter radius of 1.5nm.

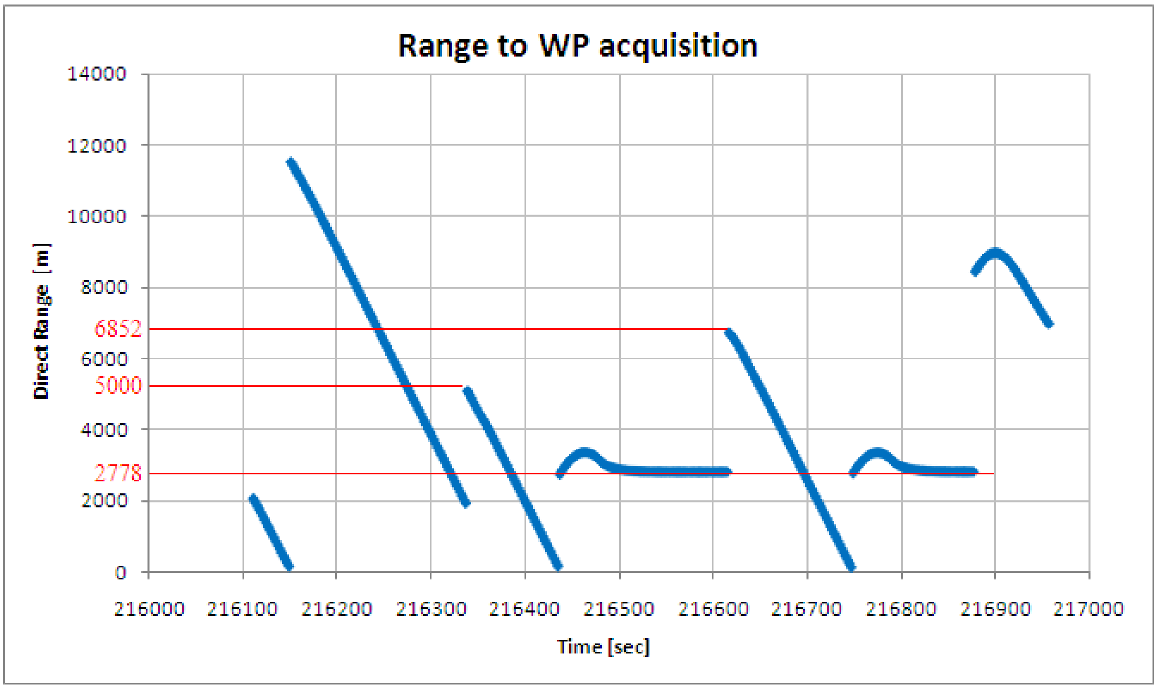


Fig 145 Simulator test: Direct range recording.

The two inclined line, representing the straight loiter sections, instead have not the same length as in the racetrack case. This is due to the fact that the first leg is acquired at the UAV entrance in the loiter, so the straight path to cover is just half of the segment plus the threshold value of 1nm. The two values at the change leg are in fact about 5000m and 6852m (about 2.7nm and 3.7nm), the difference is exactly 1nm.

About the speed, the same considerations of the Racetrack case can be done:

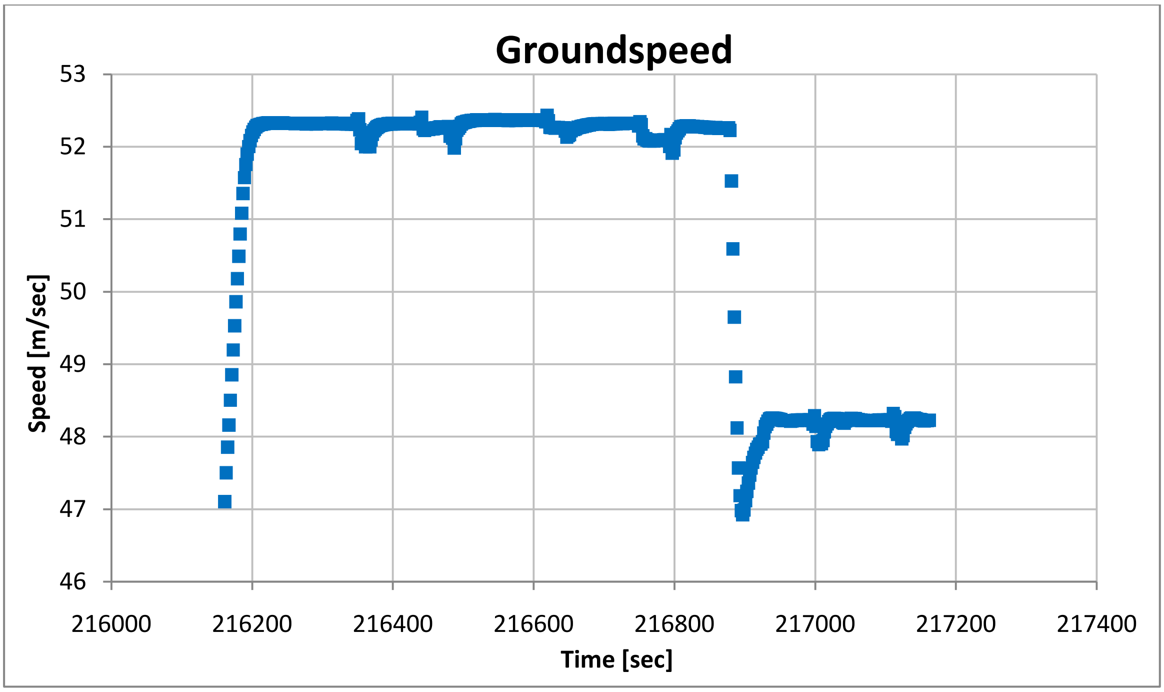


Fig 146 Simulator test: Groundspeed recording.

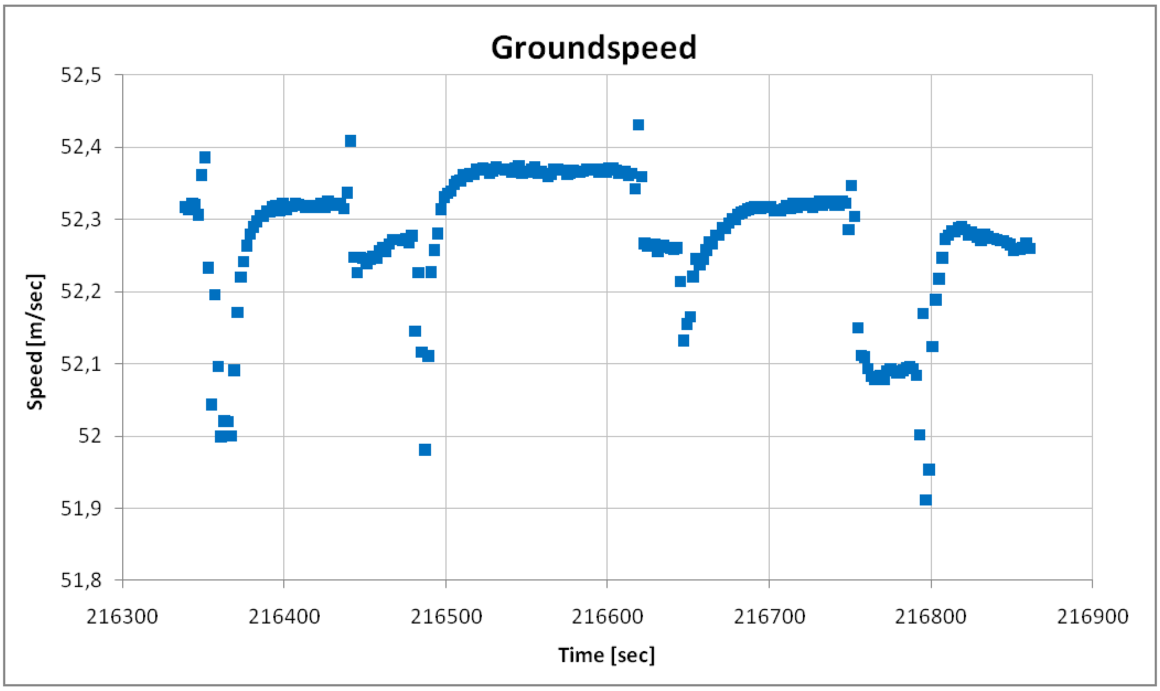


Fig 147 Simulator test: Loiter groundspeed recording.

Also the altitude is affected by some oscillation during the change of loiter section, specially because as seen before the turns are very close, but also in this case the magnitude is just in the order of the meter, so can be considerable acceptable. Moreover this altitude oscillation augment the consideration of the speed correction performances which result efficient even thought the altitude variation.

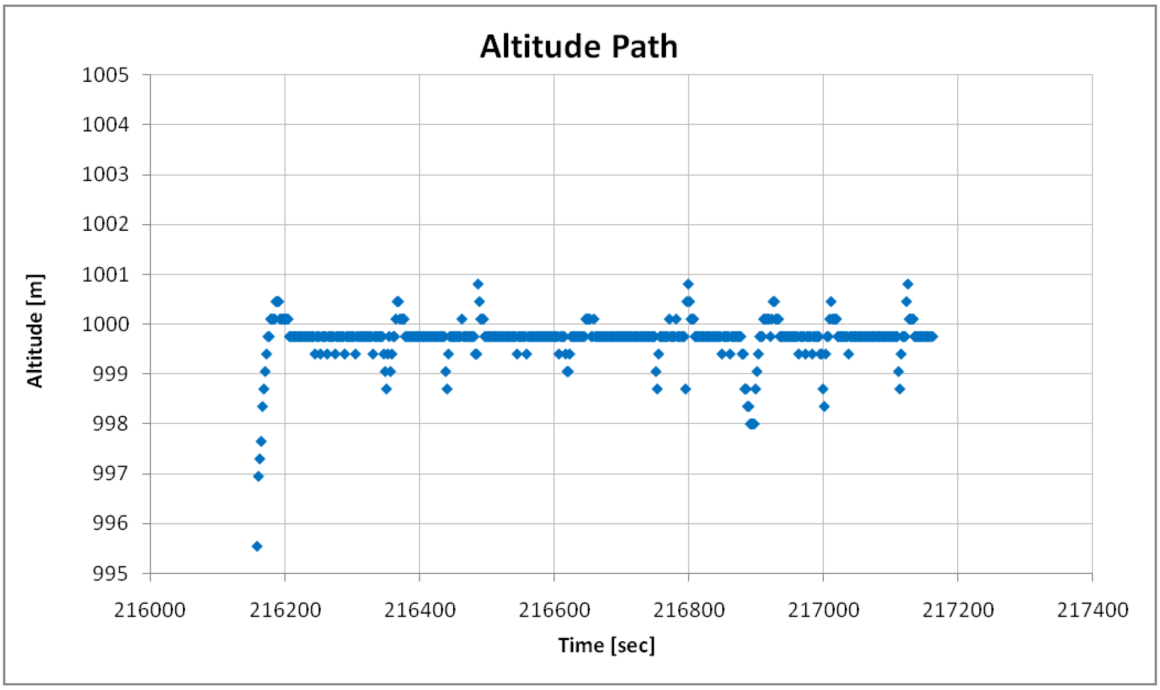


Fig 148 Simulator test: Loiter altitude recording.

10.4 RIG results

A route similar to the previous one was used also for RIG test with WP2 Type set to Fly-By and the other WP Type set to Fly-Through:

WP	**Lat [deg]**	**Lat [rad]**	**Lon [deg]**	**Lon [rad]**	**Alt [ft]**	**Alt [m]**	**IAS [kts]**	**IAS [m/sec]**
1	44,5474	0,7775	7,6200	0,1330	2000	910	90	46,3
2	44,6334	0,7790	7,7177	0,1347	2500	762	90	46,3
3	44,2896	0,7773	7,8151	0,1364	2000	610	90	46,3
4	44,2896	0,7773	7,9182	0,1382	2000	610	100	51,4

The resulting flight path is shown in Fig 149.

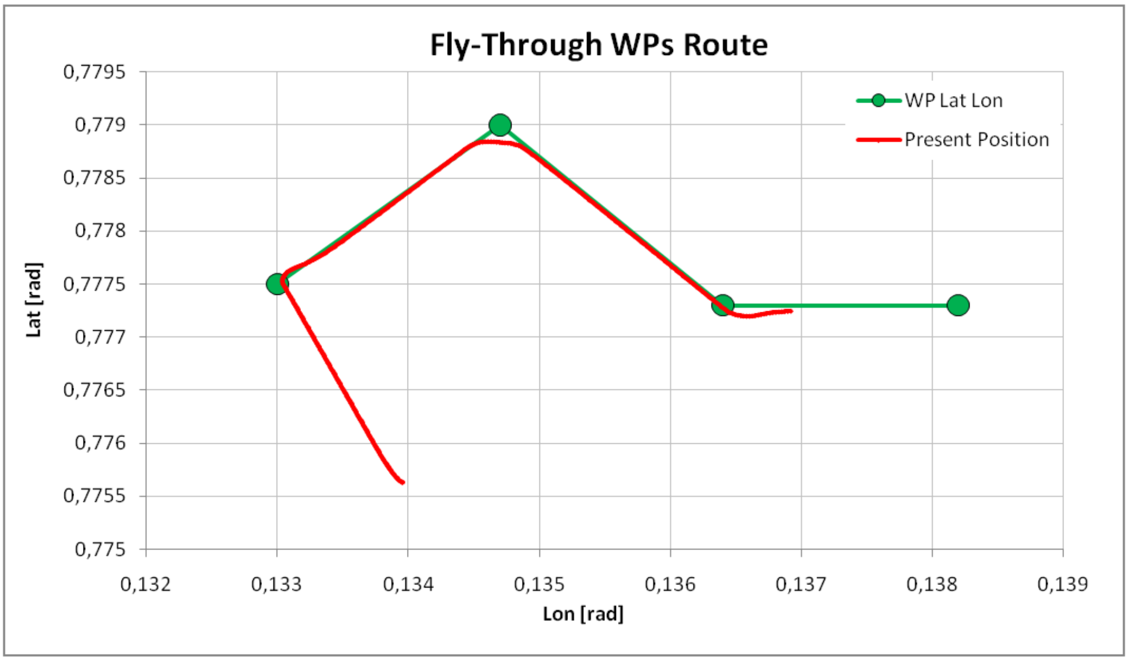


Fig 149 RIG test: Route flight path.

The Direct Range shows the distance to WP acquisition: WP1 and WP3 are Fly-Through, while WP2 is Fly-By

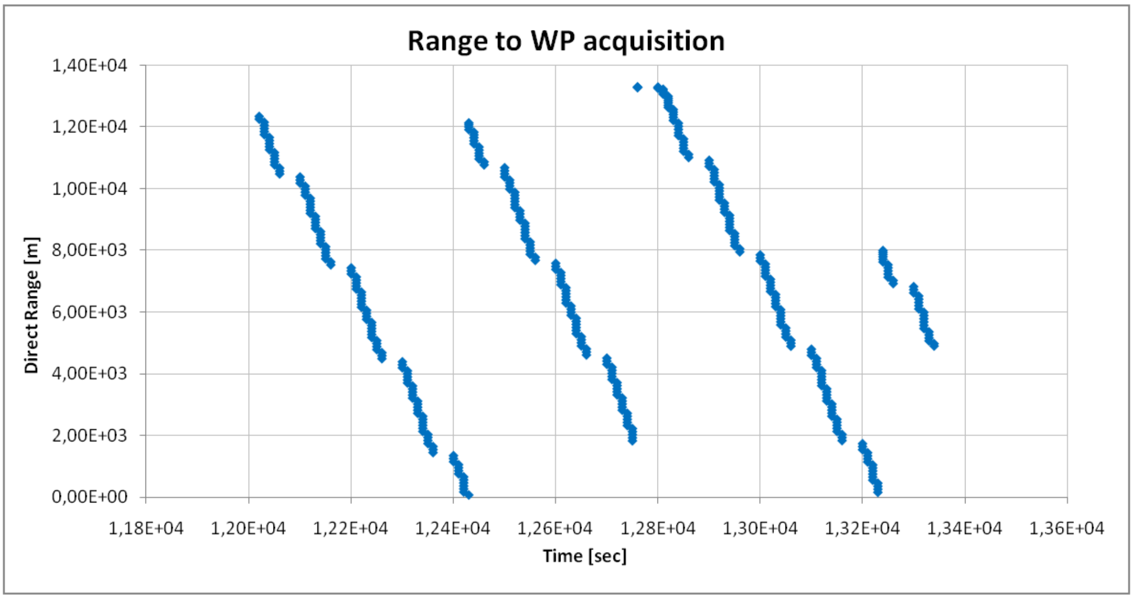


Fig 150 RIG test: Direct Range.

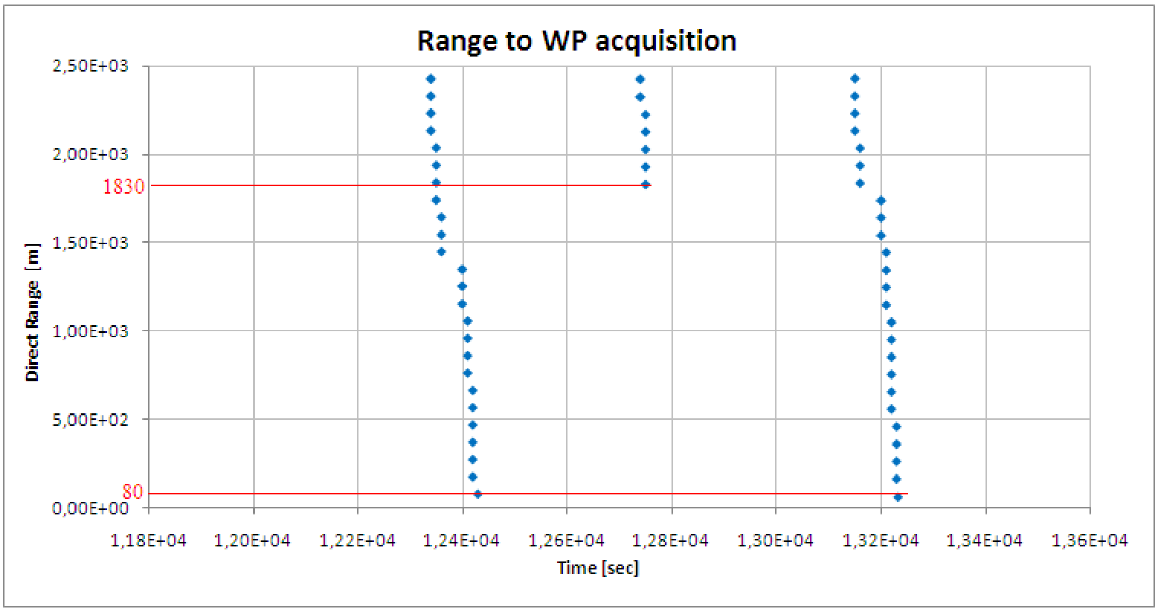


Fig 151 RIG test: Direct Range zoom.

The Fly-Through threshold is 80m and the steering command the change leg under that value, while the calculated Dist To Roll In resulted to be 1830m.
The following graph instead shows the X-Track Error time history:

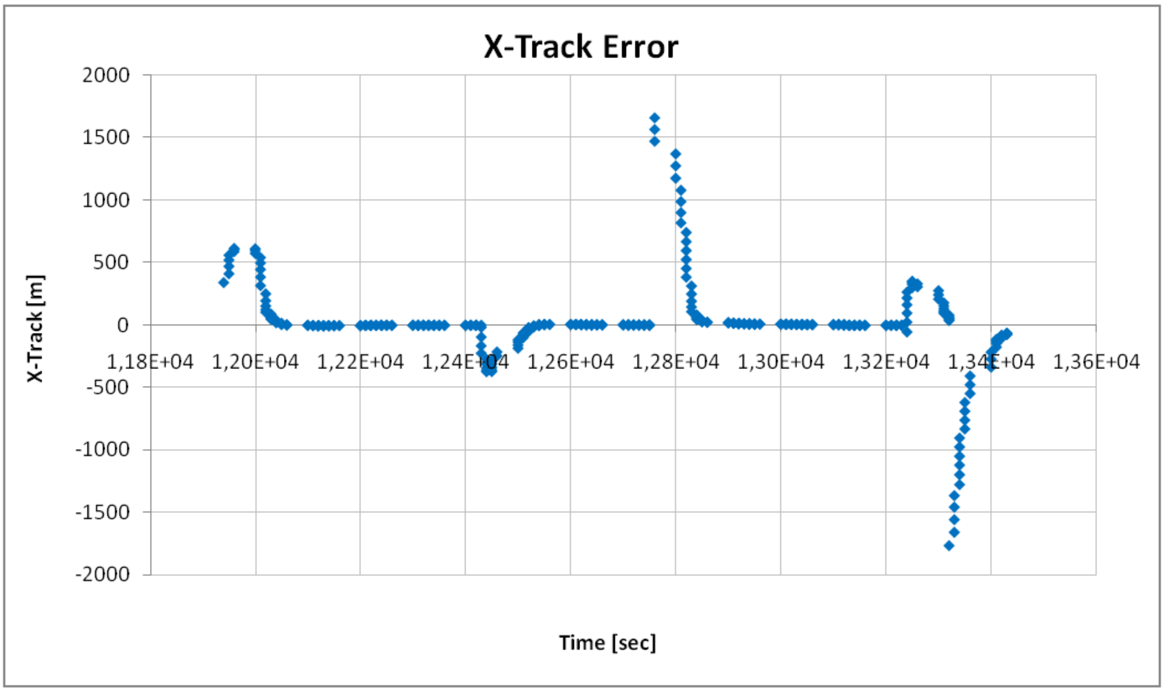


Fig 152 RIG test: X-Track Error time history.

Then the comparison between the Commanded Track and the UAV Track:

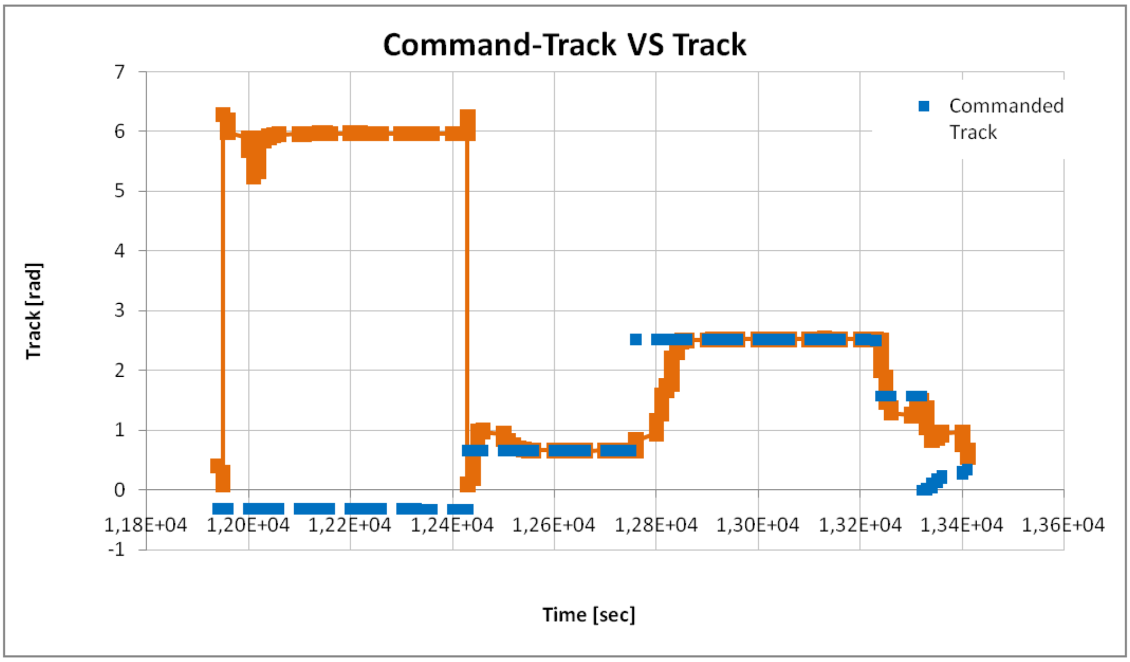


Fig 153 RIG test: Command Track VS Track.

The altitude graph shows in red the Altitude Error (right scale) and in blue the UAV Altitude (left scale). It is possible to note the answer to the steering command that brings the UAV to the correct WP altitude: 762m (2500ft) for the WP2 and 610m (2000ft) for the other.

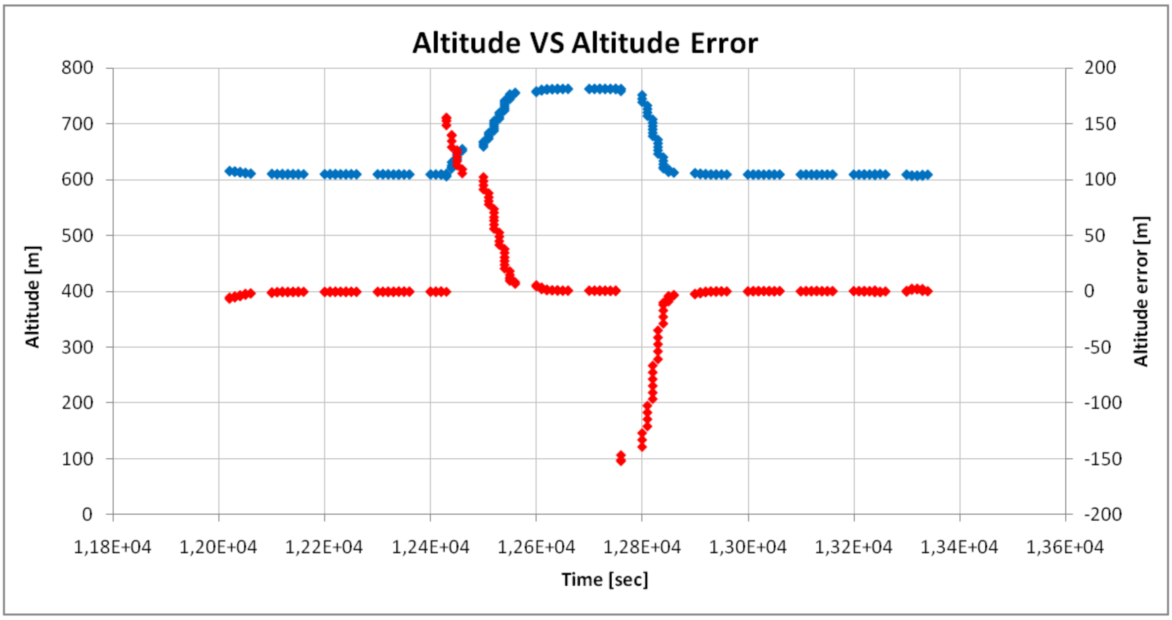


Fig 154 RIG test: Altitude VS altitude error.

11 Conclusions

The problem of the UAV remote control, focusing on the on-board segment, has been discussed, considering the need of incrementing the range of controllability to beyond the line of sight. The use of satellite communication for this purpose has been selected and the data-link topics have been described. The use of satellites, anyway, introduces latencies on the signal, this issue has been evaluated and the solution adopted has been the increase of the on-board UAV level of automations. Because of this also the possibility of multiple UAV control resulted enhanced. For the BLOS control the hand-over procedure has been investigated and to provide interoperability with other CS the STANAG 4586 was implemented and tested in flight. This adoption, impacting the communication protocol, affected also the navigation functions behavior.

So that the navigation functions (In particular the steering functions) increasing the LOA have been developed keeping in account the STANAG 4586 prescriptions. The functions development occurred with successive steps by a large use of simulations: First a Matlab® model was built to help the functions development and to test each function stand-alone and in the loop with the other functions. Then a Simulink@ flight simulator was built to test the steering functions behavior alimenting a simulated UAV controller. After this the Sky-Y flight simulator was used to test the final steering functions code alimented by the STANAG 4586 messages from the real CS and providing the navigation output to a simulation mode implementing the real UAV flight dynamics. Moreover this ambient was used also for developing the advanced steering functions as the lost-link contingency routes re-planning and the navigation slaved to the sensor pointing function. In the end the final tests on the target OBMC were made at the Software Bench for the stand-alone tests, and at the Sky-Y RIG for the integration with the other systems.

This navigation automations development allowed the provision of more advanced functions for future LOA increase.

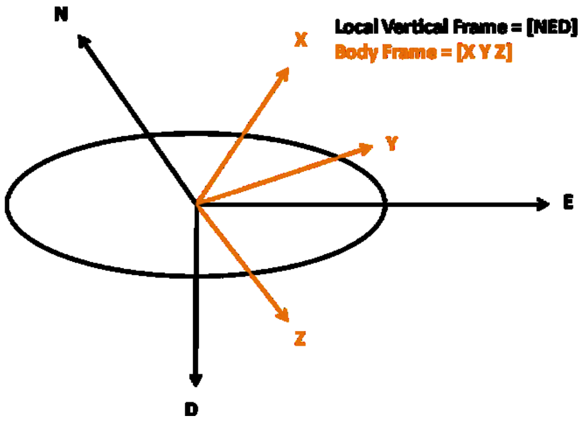
The hand-over problems has been evaluated and a test in flight has been performed to pass the control from a CS to another by using two different data-link systems like in the case of LOS to BLOS (or vice versa) hand-over.

Finally the satellite data-link requirements have been outlined for the control of up to 3 UAVs.

Appendix A STANAG 4586 adopted convention

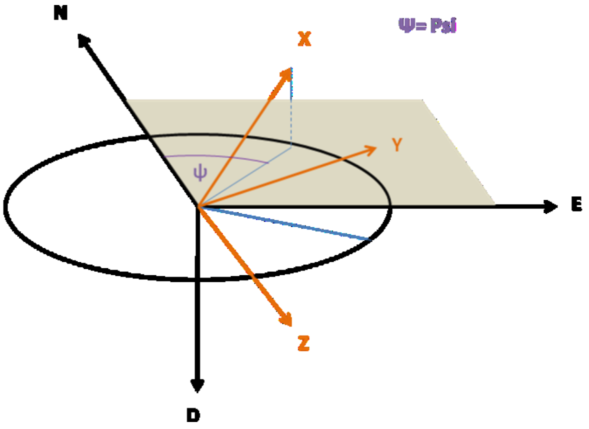
i. STANAG 4586 message #101 - Inertial States conventions

Field 14, 15, 16: Phi Theta Psi, are the Euler Angles for passing from Local Vertical NED frame to Body frame:

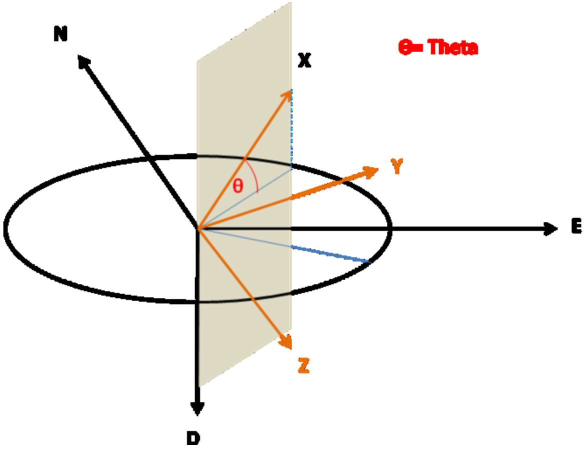


Psi is defined the angle between the Local Vertical frame North axis and the X Body axis projection on the Local Vertical frame Horizontal Plane. The positive verse is marked by the clockwise deviation of the X body projection respect to the Local Vertical North axis.

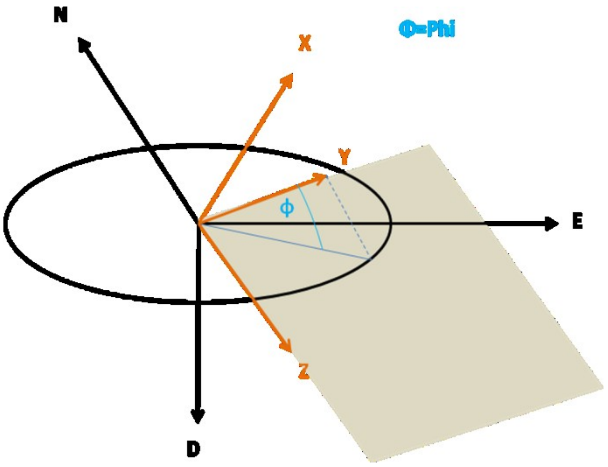
It coincides with the True Heading.



Theta is the angle between the X Body axis and its projection on the Local Vertical frame Horizontal Plane. Positive values for the X Body axis Pitch-Up deviation from the Horizontal Plane.



Phi is the angle between the Y Body axis and the Local vertical frame horizontal plane obtained turning around the X Body axis. It is positive rolling right.

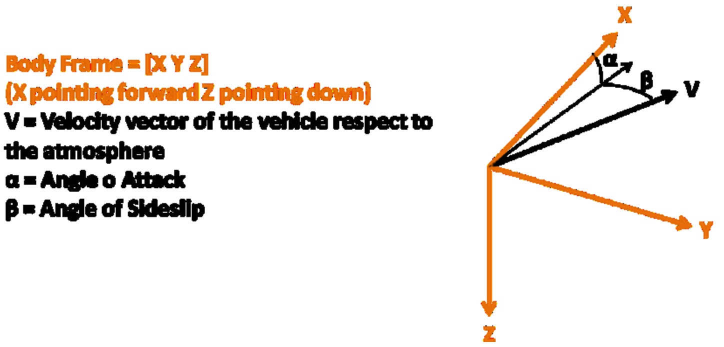


Field 20: Magnetic Variation, is the angle between True North and Magnetic North, is positive clockwise so that True = Magnetic + Variation.

STANAG 4586 Frame #102-AIR AND GROUND RELATIVE STATES conventions Field 04 and 05: Angle of Attack and Angle of Sideslip, are the angle formed by the Velocity vector of the vehicle respect to the atmosphere and the Body Frame

Angle of Attack is the angle between the Velocity vector projection on the Body Vertical Plane and the X Body axis, positive when the Velocity vector projection has positive Z Body component.

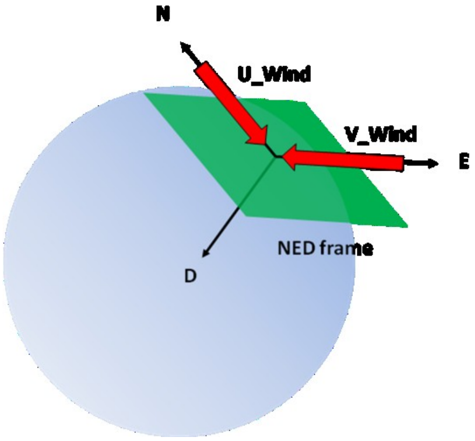
Angle of Sideslip is the angle between the Velocity vector direction and the Body Vertical Plane, positive when the Velocity vector has positive Y Body component.


Body Frame = [X Y Z]
(X pointing forward Z pointing down)
V = Velocity vector of the vehicle respect to the atmosphere
α = Angle o Attack
β = Angle of Sideslip
X
α
β
V
Y
Z

Field 09 and 10: U_Wind and V_Wind, are the Wind Speed components on the Local Vertical NED frame Horizontal Plane

U_Wind is aligned with the North axis with opposite verse

V_Wind is aligned with East axis with opposite verse



ii. STANAG 4586 message #103 – Body Relative Sensed States conventions

Field 04, 05, 06: X_Body_Accel, Y_Body_Accel, Z_Body_Accel, are the inertial measure of the body accelerations:

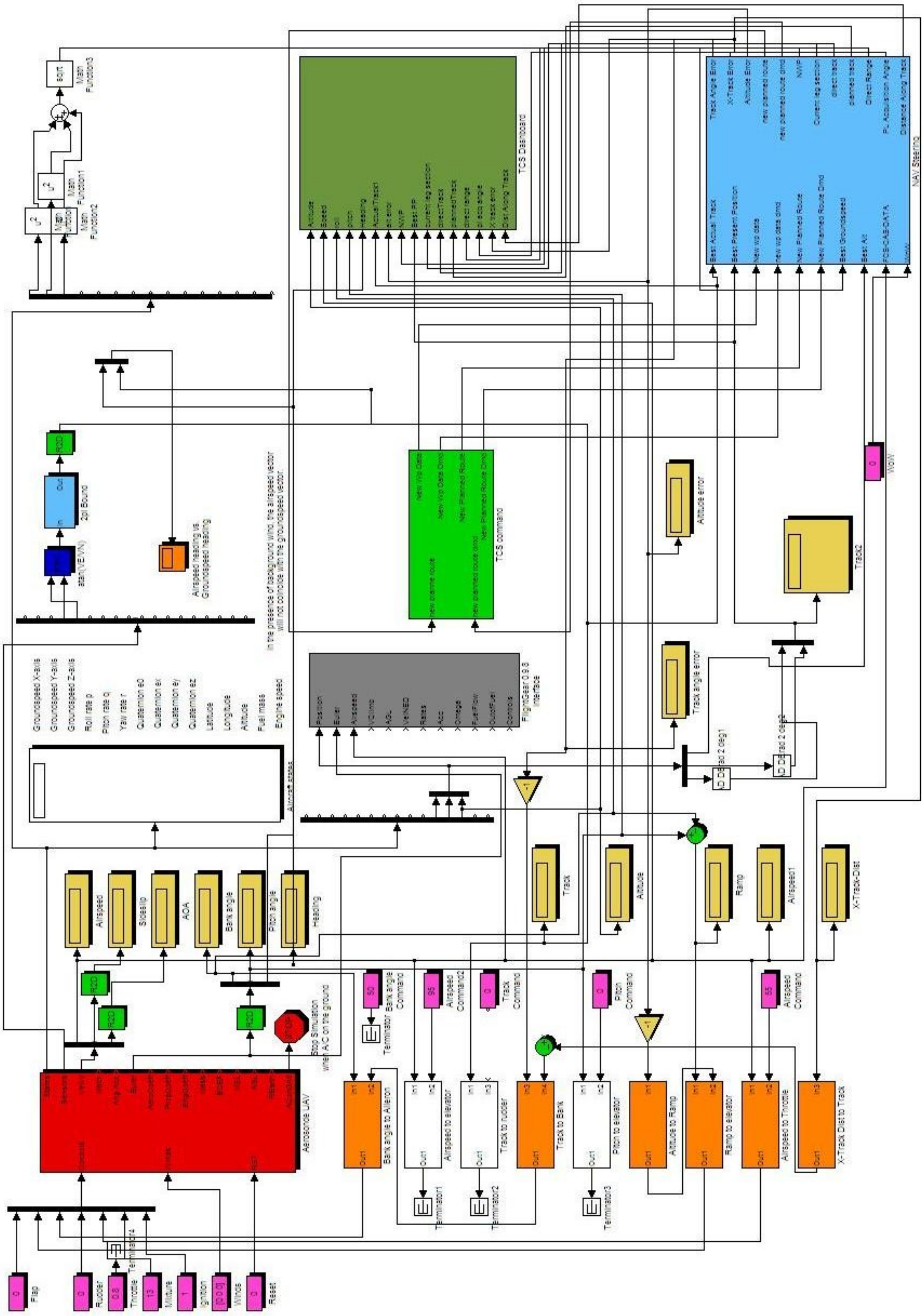
X_Body_Accle is the aircraft acceleration along X body axis positive accelerating forward

Y_Body_Acceleration is the aircraft acceleration along Y body axis positive accelerating right

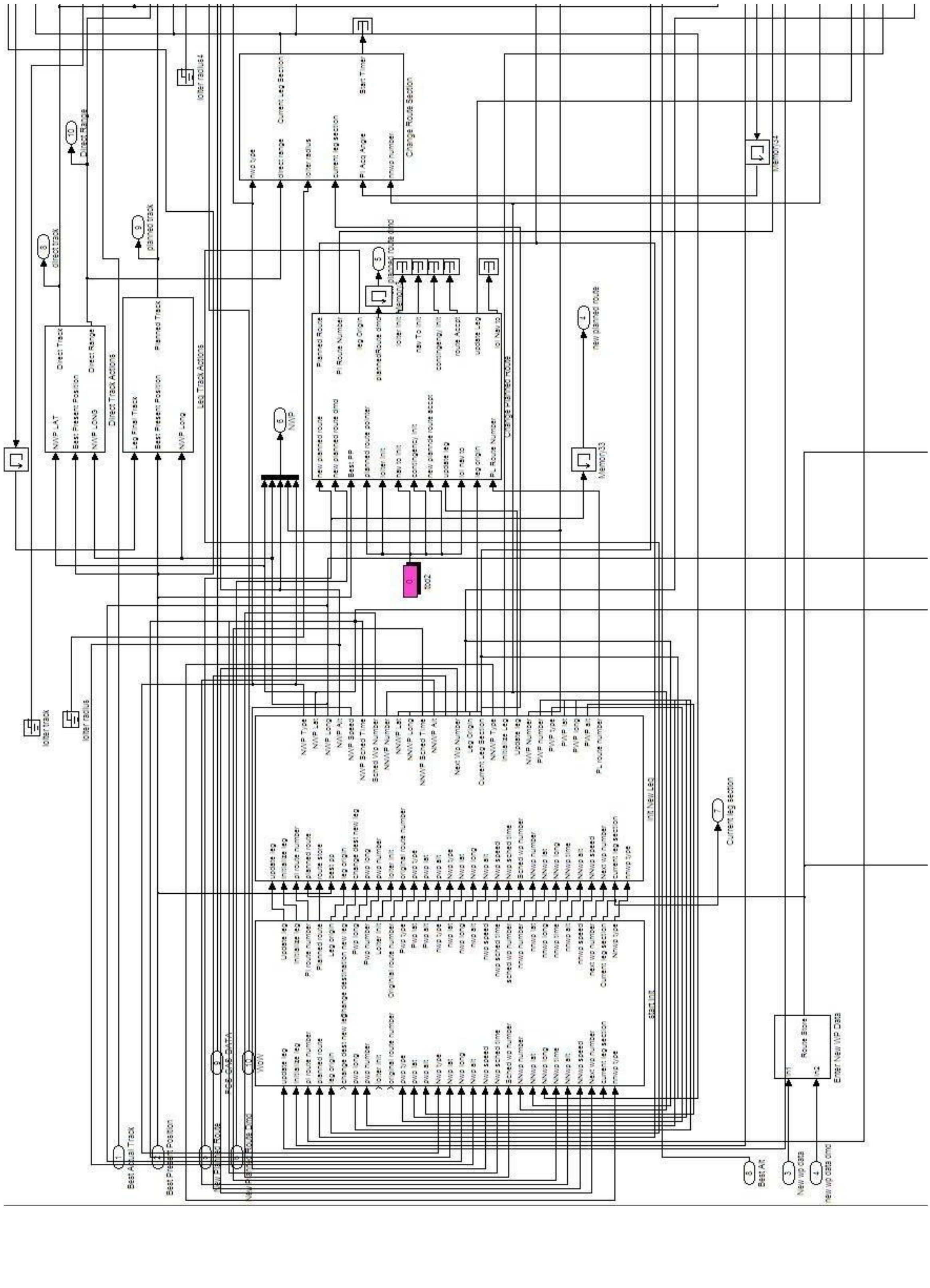
Z_Body_Acceleration is the aircraft acceleration along Z body axis positive accelerating downward

On ground in steady condition the measure are [00-g] with g=gravity acceleration.

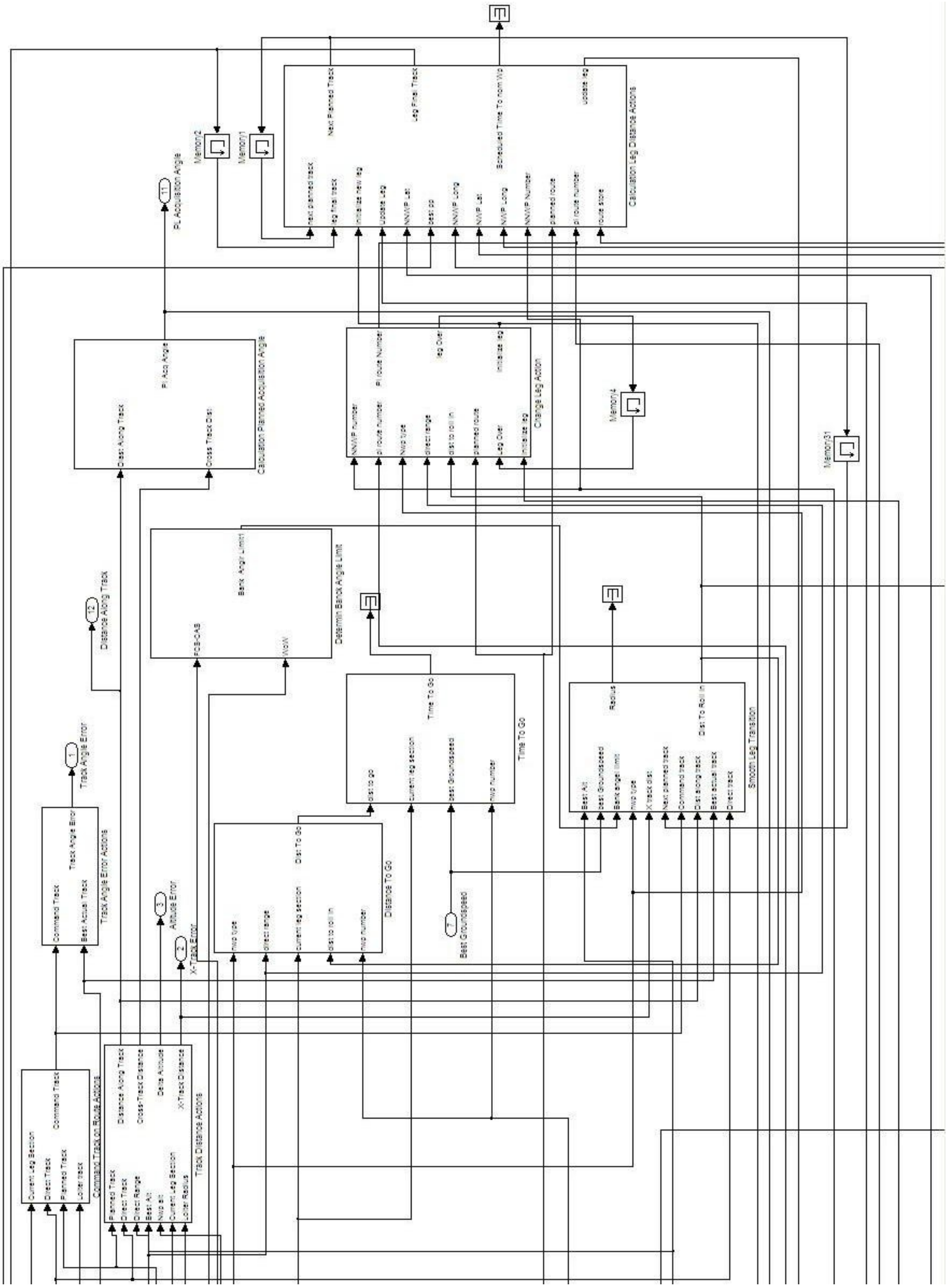
Appendix B - Simulink® model



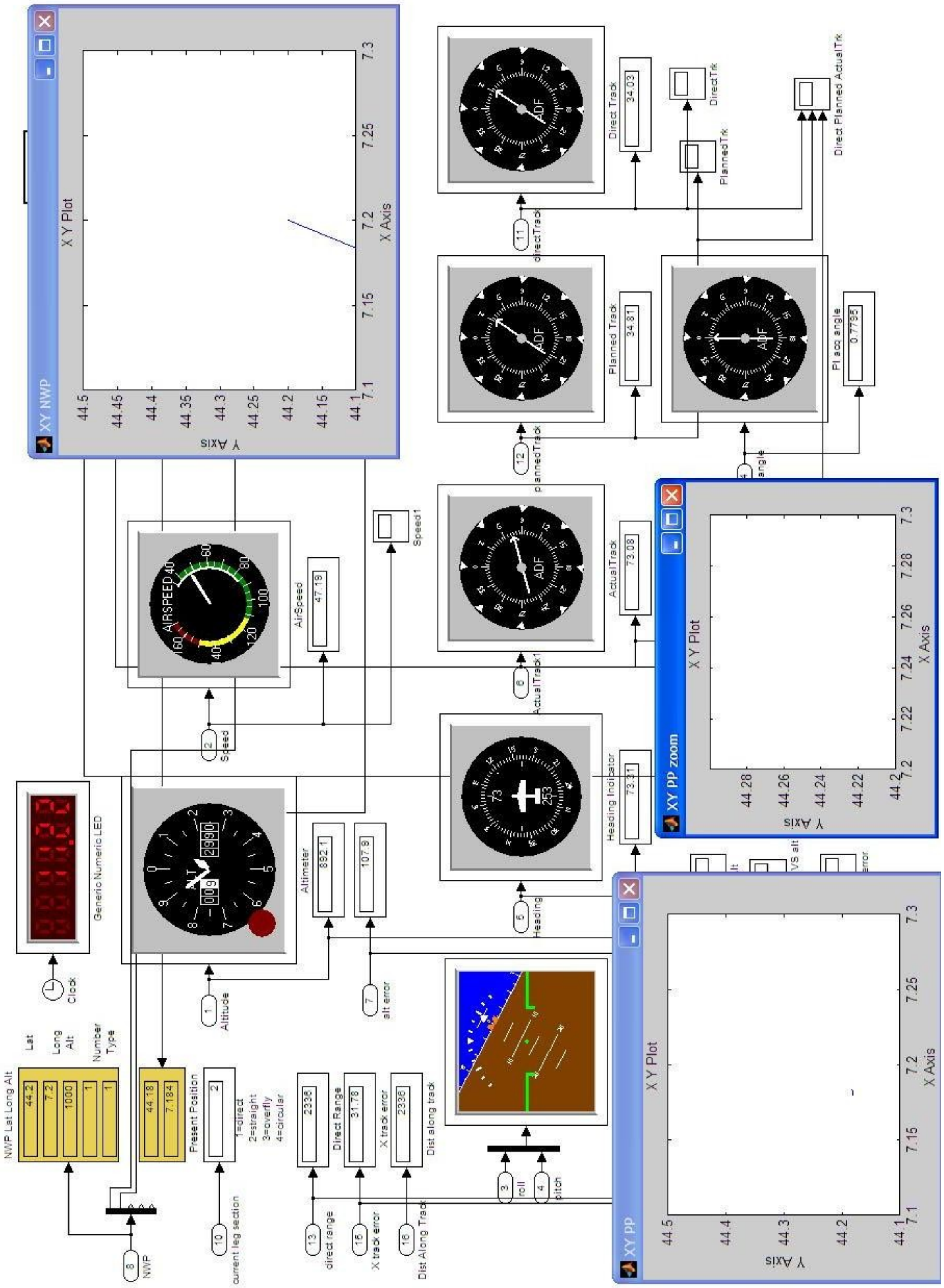
Board and Ground simulation model



Steering model part1



Steering model part2



Dashboard model

Abbreviations and acronyms

A/P	Auto Pilot
ADS	Air Data System
AGL	Above Ground Level
AHRS	Attitude and Heading Reference System
ATC	Air Traffic Control
ATM	Air Traffic Management
ATOL	Automatic Take Off and Landing
AV	Air Vehicle
BER	Bit Error Rate
BIT	Built-In Test
BLOS	Beyond Line Of Sight
C4I	Command, Control, Communications, Computers and Intelligence
CCI	Command & Control Interface
CCISM	Command and Control Interface Specific Module
CRC	Cyclic Redundancy Check
CS	Control Station
CSCI	Computer Software Configuration Item
CUCS	Core UAV Control System
DFAD	Digital Feature Analysis Data
DGPS	Differential Global Position System
DIGEST	Digital Geographic Information Exchange Standard
D/L	Data Link
DLI	Data Link Interface
DTED	Digital Terrain Elevation Data
EIRP	Equivalent Isotropic Radiated Power
EFCC	Experimental Flight Control Computer
EFCS	Experimental Flight Control System
ES	Earth Station
ESM	Electronic Support Measures
FCS	Flight Control System
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FMS	Flight Management System
FOM	Figure Of Merit
FTP	File Transfer Protocol
FTS	Flight Termination System
GCS	Ground Control Station
GMT	Greenwich Mean Time
GPS	Global Position System
GS	Ground Speed

HCI	Human Computer Interface
HDG	Heading
HEO	High earth Orbit
HTTP	Hypertext Transfer Protocol
ID	Identification
IMS	International Military Staff
INS	Inertial Navigation System
IP	Internet Protocol
IRS	Inertial Reference System
ISR	Intelligence, Surveillance, Reconnaissance
KF	Kalman filter
LEO	Low Earth Orbit
LOI	Level Of Interoperability
LOS	Line Of Sight
MALE	Medium Altitude Long Endurance
MEO	Medium Earth Orbit
MMS	Mission Management System
MTOW	Maximum Take Off Weight
n/a	not applicable
NADSI	NATO Advantaged Data Storage Interface
NBDL	Narrow Band Data Link
NC3	NATO Command, Control, Communication
NCSP	NC3 Common Standard Profile
NTP	Network Time Protocol
OBMC	On Board Mission Computer
OSI	Open System Interconnection model
PID	Proportional Integrative Derivative
RAID	Redundant Array of Inexpensive/Independent Disks
ROS	Remote Operator Station
RTI	Run Time Input
SNS	Sensor System
TCP	Transfer Control Protocol
TDMA	Time Division Multiple Access
TCS	Tactical Control Station
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UCS	UAV Control System
UDP	User Datagram Protocol
ULS	UpLink Switch
VSM	Vehicle Specific module
UTC	Universal Time Coordinates
WBDL	Wide Band Data Link
WGS-84	World Geodetic System – 84
WP	Waypoint

References

1. Natalini A., ‘Origini e sviluppo dei velivoli senza pilota (1848 - 1990)’, Editrice uni Service
2. ‘UAV Development and History at Northrop Grumman Corporation’, Ryan Aeronautical Center, 2004
3. Blyenburgh & Co, ‘UAS: The Global Perspective’, 2011-2012 UAS Yearbook 9th Edition, June 2011
4. Hodson Chris J., “Civil airworthiness for UAV control station”, University of York, Sept 2008.
5. Di Loreto M., ‘Impiego joint e combined di Unmanned Aerial Vehicles (UAV): stato dell’arte prospettive future di impiego’, Centro Militare di Studi Strategici, 2006
6. Noth A., ‘History of Solar flight’, Autonomous Systems Lab, Swiss Federal Institute of Technology, Züric
7. Romeo G., Frulla G., ‘HELIPLAT: high altitude very-long endurance solar powered UAV for telecommunication and Earth observation applications’, The Aeronautical Journal
8. Basic of satellite communications, JISC , www.jisc.ac.uk
9. Dovis F., Mulassano P., ‘Introduction to Global Navigation Satellite Systems’, Politecnico I Torio, August 2009
10. Kaplan E. D., Hegarty C. J., ‘Understanding GPS: principles and applications’, Second Edition, Artech House, Norhood, MA, 2006
11. Zippo G., ‘Sistemi di controllo di velivoli non pilotati’ Tesi di Laurea in Aerospace engineering, POLITECNICO DI TORINO March 2004
12. Range Safety Group, ‘Flight Safety System (FSS) for Unmanned Aerial Vehicle (UAV) Operation’, Range Commanders Council, March 2008
13. Taylor R. M, Abodi S, Dru-Drury R, Bonner M. C, ‘Cognitive cockpit systems: information requirements analysis for pilot control of cockpit automation’, Ch 10, p81-88 in Engineering psychology and cognitive ergonomics Vol 5, Aerospace and transportation systems Ed Harris, pub Ashgate, Aldershot, 2001
14. NATO Standardization Agency, ‘STANAG 4586 (edition 2)’, ANNEX B, November 2007
15. NATO Standardization Agency, ‘STANAG 4586 Edition 2 Implementation Guideline Document’, Allied Engineer Publication 57 Volume1, January 2009
16. NATO Standardization Agency, ‘Ratification of STANAG 4586 (edition 3)’, December 2008
17. Frazzetta S., Pacino M., ‘A STANAG 4586 Oriented Approach to UAS Navigation: The case of Italian Sky-Y Flight Trials’, International Conference of Unmanned Systems, Philadelphia, June 12-15 2012.

18. Clough B., T., ‘Metrics, Schmetrics! How The Heck Do You Determine A UAV’s Autonomy Anyway?’, Proceedings of the Performance Metrics for Intelligent Systems Workshop, Gaithersburg, Maryland, 2002
19. www.navigazione-aerea.com
20. www.wikipedia.com