

UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”



FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA AEROSPAZIALE

Corso di Dottorato in Ingegneria Aerospaziale, Navale e della Qualità
XXIV Ciclo

**Autonomous Approach and Landing Algorithms
for Unmanned Aerial Vehicles**

Anno 2011

Autore:	Ettore De Lellis
Tutor Aziendale:	Dr. Federico Corraro
Tutor Universitario:	Prof. Domenico Accardo
Coordinatore:	Prof. Antonio Moccia

*To my family Nora, Carlo, Sissi
and to Annachiara
with love*

ACKNOWLEDGEMENTS

First of all I would like to sincerely express my gratitude to my academic tutors Prof. Domenico Accardo and Prof. Antonio Moccia and to my corporate tutor Dr. Federico Corraro for giving me the opportunity to grow professionally and humanely with this experience in a Phd course at the University of Naples supported by CIRA (Italian Aerospace Research Center) with the hope that the interesting activities started in this three years will continue in the future.

All the scientific work carried out during these three years has been possible only thanks to my colleagues involved in the System Unit of CIRA, LGNC and LATM laboratories. More specifically I am grateful to the CIRA TECVOL project team because only with the inventive and high proficiency of their members was it possible to reach so many interesting research results validated through very challenging flight campaigns. In particular my specific acknowledgements are for the aforementioned Federico Corraro and for Umberto Ciniglio, Carmine Marrone, Vittorio Di Vito, Gianfranco Morani, Nicola Genito and Luca Garbarino, for their passion and care in the research activities in which we worked together.

I also wish to express my thanks to my friend Caterina Calabrò who helped me with a quick but precious revision of this thesis and to Giuseppe Greco who, with the work for his master degree, collaborated with me in parts of the research activities for my Phd.

However dedicating so much time to this interesting research was only possible thanks to my mother Nora and my father Carlo who, with their love and faith in me and in their own way, gave me the strength and the stability to face any kind of problem. I am also grateful to my sister Sissi for her encouragement to always tend towards life and professional improvements, I am so proud of her recently becoming an Italian judge.

Finally, I would like to express my gratitude to my darling Annachiara for his constant presence and patience in me. She supported me in writing this thesis becoming a better expert than me in UAVs history in spite of having a completely different educational background. I sincerely hope to be part of her life also in the future.

TABLE OF CONTENTS

1	PREFACE	6
1.1	Abstract	6
1.2	List of Abbreviations	8
1.3	List of Symbols	11
1.4	Reference Systems	14
2	INTRODUCTION	16
3	SURVEY OF AUTONOMOUS APPROACH AND LANDING ALGORITHMS	26
3.1	Algorithms for Autonomous Landing Systems	26
3.2	Algorithms for Optimal Path Generation	29
3.2.1	<i>Algorithms for Optimal Path Generation without No-Fly Zones</i>	29
3.2.2	<i>Algorithms for Optimal Path Generation considering No-Fly Zones</i>	30
3.3	Sensor Fusion Algorithms	31
3.3.1	<i>EGNOS System Overview</i>	32
3.3.2	<i>Digital Elevator Map (DEMs) devices currently available</i>	38
4	PROPOSED AUTONOMOUS APPROACH AND LANDING ALGORITHMS	39
4.1	On-board SW Functional Architecture	41
4.2	UAV Automation and Reconfiguration Logic	43
4.3	Path Generation and Tracking for En-Route and Terminal Area Operations	50
4.3.1	<i>Proposed Algorithm for Optimal Path Generation without No-Fly Zones</i>	52
4.3.2	<i>Proposed Algorithm for Optimal Path Generation considering No-Fly Zones</i>	54
4.3.3	<i>Trajectory tracking</i>	61
4.4	Path Generation for Flare phase	63
4.5	Sensor Fusion Algorithms	65
4.5.1	<i>Positioning Estimation</i>	65
4.5.2	<i>Above Runway Level Estimation</i>	68
4.6	Configuration and Design Constraints	73
5	DEVELOPMENT OF VERIFICATION TOOLS	76
5.1	Numeric Simulation Environment	76
5.1.1	<i>Laser Altimeter Model</i>	79
5.1.2	<i>DEM Error Model Development</i>	82
5.2	Laboratory Test Rig	83
5.3	Flying Demonstrator	84
6	NUMERICAL AND EXPERIMENTAL ASSESSMENTS	86
6.1	Development and Implementation Process	86
6.1.1	<i>Requirement definition and System Design</i>	86

6.1.2	<i>Implementation & On-Ground Testing</i>	87
6.1.3	<i>Integration and Flight Validation</i>	88
6.2	Numerical and Laboratory Assessments	88
6.2.1	<i>Numerical Assessments</i>	88
6.2.2	<i>Laboratory Assessments</i>	94
6.3	Flight Demonstrations	100
7	CONCLUSIONS	107
8	AUTHOR REFERENCES	109
9	BIBLIOGRAPHY	111

1 PREFACE

1.1 Abstract

In recent years, several research activities have been developed in order to increase the autonomy features in Unmanned Aerial Vehicles (UAVs), to substitute human pilots in dangerous missions or simply in order to execute specific tasks more efficiently and cheaply. In particular, a significant research effort has been devoted to achieve high automation in the landing phase, so as to allow the landing of an aircraft without human intervention, also in presence of severe environmental disturbances. The worldwide research community agrees with the opportunity of the dual use of UAVs (for both military and civil purposes), and because of this it is very important to make the UAVs and their autoland systems compliant with the present and future rules and with the procedures regarding autonomous flight in ATM (Air Traffic Management) airspace in addition to the typical military aims of minimizing fuel, space or other important parameters during each autonomous task.

Developing autoland systems with the desired level of reliability, accuracy and safety involves an evolution of all the subsystems related to the guide, navigation and control disciplines. The main drawbacks of the autoland systems are lack of “adaptivity” to the trajectory generating and tracking process to unpredictable external events, such as varied environmental conditions and unexpected threats to avoid, or the missed compliance between the guidelines imposed by certification authorities and the technologies used to get the desired above mentioned adaptivity.

During his PhD period the author contributed to the development of an autonomous approach and landing system considering all the indispensable functionalities AS mission automation logic, runway data managing, sensor fusion for optimal estimation of vehicle state, trajectory generation and tracking considering optimality criteria and health management algorithms.

In particular the system addressed in this thesis is capable of performing a fully adaptive autonomous landing starting from any point of three dimensional space. The main novel feature of this algorithm is that it generates on line, with a desired updating rate or at a specified event, the nominal trajectory for the aircraft, based on the actual state of the vehicle and on the desired state at touch down point. Main features of the autoland system based on the implementation of the proposed algorithm are: on line trajectory re-planning in the landing phase, fully autonomy

from remote pilot inputs, weakly instrumented landing runway (without ILS availability), ability to land starting from any point in space and autonomous management of failures and/or adverse atmospheric conditions, decision-making logic evaluation for key-decisions regarding possible execution of altitude recovery manoeuvre based on the Differential GPS integrity signal and compatible with the functionalities made available by the future GNSS system.

All the algorithms developed allow reduction of computational tractability of trajectory generation and tracking problems so as to be suitable for real time implementation but still obtaining a feasible, robust and adaptive trajectory for the UAV.

All the activities related to the current study have been conducted at CIRA (Italian Aerospace Research Center) in the framework of the aeronautical TECVOL project whose aim is to develop innovative technologies for the autonomous flight. The autoland system was developed by the TECVOL team and the author's contribution to it will be outlined in the thesis. Effectiveness of proposed algorithms has been then evaluated in real flight experiments, using the aeronautical flying demonstrator available at CIRA.

1.2 List of Abbreviations

3D	Three Dimensional
3DoF	Three degree of Freedom
4D	Four Dimensional
6DoF	Six Degree of Freedom
A/P	AutoPilot
ADC	Air Data Computer
ADS	Air Data System
ADSB	Automatic Dependent Surveillance – Broadcast
AHRS	Attitude and Heading Reference System
AIT	Assembly, Integration, Test
AIV	Assembly, Integration, Verification
ALI	AutoLanding Interface
AMF	Adaptive Model Following
ANT	GPS ANTenna
AoA	Angle of Attack
AoB	Angle of Bank
AoS	Angle of Sideslip
ASA	Active Set Algorithm
ASQF	Application Specific Qualification Facility
ATC	Air Traffic Control
ATM	Air Traffic Management
ATOL	Automatic Take-Off and Landing
CCF	Central Control Facility
CIRA	Italian Aerospace Research Center
COTS	Component Off The Shelf
DCA	Dynamic Control Allocation
DGPS	Differential Global Positioning System
DTFT	Dropped Transonic Flight Test
EGNOS	European Geostationary Navigation Overlay System
EWAN	EGNOS Wide Area communication Network
FCC	Flight Control Computer
FCL	Flight Control Laws
FCS	Flight Control System
FDSW	Facility Dependent SW

FHM	Failure & Health Monitoring
FLARE	Flying Laboratory for Aeronautical REsearch
FMS	Flight management System
FOG	Fibre Optics Gyroscopes
FTB	Flying Test Bed
FTE	Flight Technical Error
GCL	Guidance and Control Laws
GCS	Ground Control Station
GNC	Guidance, Navigation, Control
GNC-OBC	GNC On Board Computer
GPS	Global Positioning System
HALE	High Altitude, Long Endurance
HMI	Human Machine Interface
HW	Hardware
HYSY	HYdraulic SYstem
I/O	Input/Output
IGP	Ionospheric Grid Point
INS	Inertial Navigation System
IOD	Issue Of Data
IODF	Issue Of Data Fast correction
L/D	Lift to Drag ratio
LPV	Linear Parameter Varying
MAG	MAGnetometer sensor
MALE	Medium Altitude, Long Endurance
MAV	Micro unmanned Aerial Vehicle
MCC	Mission Control Centres
MEMS	Micro Electro-Mechanical Systems
MILP	Mixed Integer Linear Programming
MRAC	Model Reference Adaptive Control
MRAS	Model Reference Adaptive System
NASA	National Aeronautics and Space Administration
NED	North, East, Down
NGATS	Next Generation Air Transportation System
NLES	six Navigation Land Earth Stations
NP	Non Polynomial
NSE	Navigation Sensors Error

OBDH	On Board Data Handling
PACF	Performance Assessment and system Checkout Facility
PVTOL	Planar Vertical Take-Off and Landing
PWM	Pulse Width Modulation
RHC	Receding Horizon Control
RIMS	thirty-four Ranging and Integrity Monitoring Station
RLV	Reusable Launch Vehicle
RLV	Reusable Launcher Vehicle
RNP	Required Navigation Performance
RNAV	aRea NAVigation
RTK	Real Time Kinematic
SAR	Synthetic Aperture Radar
SBAS	Satellite Based Augmentation System
SCAS	Stability and Control Augmentation System
SESAR	Single European Sky ATM Research
SISO	Single Input Single Output
SNS	Sensor and Navigation System
SUDA	See, Understand, Decide, Act
SW	SoftWare
TAEM	Terminal Area Energy Management
TAS	True Air Speed
TECVOL	TECnologie per il VOLO autonomo (Autonomous Flight Technologies)
TPBVP	Two Point Boundary Value Problem
UAV	Unmanned Aerial Vehicles
UCAV	Unmanned Combat Aerial Vehicle
UKF	Unscented Kalman Filter
USA	United States of America
USV	Unmanned Space Vehicle
VLA	Very Light Aircraft
WP	Way-Point

1.3 List of Symbols

J	Trajectory planning cost function
t_0	Initial time instant, [s]
t_f	Final time instant, [s]
f	Vehicle state equations
X	Vehicle states
U	Commands for Trajectory planning
Y	Outputs of a generic state-space model
v	Forbidden zone specification
V	Vehicle velocity vector, [m/s]
P	Vehicle position vector, [m]
$(x_{NED}, y_{NED}, z_{NED})$	NED system of coordinate axis, [m]
(V, χ, γ)	Inertial Velocity: module, track and path angles w.r.t. NED, [m/s, rad]
$(V_{TAS}, \alpha_{TAS}, \beta_{TAS})$	Air Velocity: module, AoA, AOs w.r.t. Body, [m/s, rad]
(φ, θ, ψ)	Euler's angles, respectively roll, pitch and heading angles, [rad]
(σ, α, β)	Incidence angles, respectively bank, angle of attack and sideslip, [rad]
(n_x, n_y, n_z)	Axial, lateral and vertical load factors in body axis
n_{xmin}, n_{xmax}	Axial minimum and maximum load factors
n_{zmin}, n_{zmax}	Axial minimum and maximum load factors
V_{min}, V_{max}	Velocity module minimum and maximum values, [m/s]
$\gamma_{min}, \gamma_{max}$	Flight path angle minimum and maximum values, [rad]
R^H	Horizontal Trajectory Curvature Radius, [m]
R^H_{min}	Minimum Horizontal Trajectory Curvature Radius, [m]
R^V	Vertical Trajectory Curvature Radius, [m]
R^V_{min}	Minimum Vertical Trajectory Curvature Radius, [m]
η	Derivative of track angle [rad/s]
μ	Derivative of flight path angle [rad/s]
Γ	Trajectory path
d	Length of a path, [m]
M	Number of no-fly zones
R_j	Radius of j no-fly zone of circular shape, [m]
(x_j, y_j)	Center of a no-fly zone of circular shape, [m]
Π	Set of no-fly zones for a given mission
$\Pi'=(D'_1, ..D'_n)$	Set of no-fly zones $D'_1, ..D'_n$ violated by a given path
l'_1	Straight segment of a tentative path

s	Curvilinear abscissa of a given trajectory
σ_{UDRE}	Satellite user differential range error standard deviation
σ_{GIVE}	Satellite residual ionospheric vertical error standard deviation
(e_x, e_y, e_z)	Shortest UAV displacements to reference trajectory, [m]
e	Module of shortest UAV displacements to reference trajectory, [m]
Ω	Straight line tangent to a flight path
K_P	Inverse of the Look ahead distance, [m ⁻¹]
X	Current flare position along X-Runway axis [m]
$V_X(X)$	Inertial velocity profile along X-Runway axis for the flare trajectory; [m/s]
$V_Z(X)$	Inertial velocity profile along Z-Runway axis for the flare trajectory; [m/s]
H_0	Initial flare position along Z-Runway axis; [m]
H_0	Initial flare position along Z-Runway axis; [m]
X_0	Initial flare position along X-Runway axis; [m]
V_{X0}	Initial flare inertial velocity along X-Runway axis; [m/s]
H_F	Desired final position (at the touch down) along Z-Runway axis; [m]
X_F	Desired final position (at the touch down) along X-Runway axis; [m]
V_{XF}	Desired final inertial velocity (at the touch down) along X-Runway axis;
V_{ZF}	Desired final inertial velocity (at the touch down) along Z-Runway axis;
(α_L, β_L)	Laser altimeter mounting angles
x_k	State vector of a system at the time step k (k)
y_k	Output vector of a system (k)
v_k	Noise vector associated with the measurement noise (k)
w_k	Noise vector associated model/input error (k)
\hat{x}_k	A-posteriori state estimation using the knowledge of y_k (k)
\underline{e}_k^-	A-priori estimation error (k)
\underline{e}_k	A-posteriori estimation error (k)
P_k^-	A-priori error covariance matrix (k)
P_k	A-posteriori error covariance matrix (k)
R_K	Covariance matrix associated with the measurement noise (k)
Q_K	Covariance matrix associated with the model/input noise (k)
A_K	Linearized system matrix (k)
C_K	Linearized output matrix (k)
K_K	Kalman gain matrix
h	Above runway level (ARL) of the vehicle
ΔR	Terrain elevation above the runway

V_{Zm}	Vertical Speed measured by GPS
ΔR_m	Terrain elevation as indicated by the DEM
r_m	Laser Range
q_m	Laser altitude elaborated using the measured laser range r_m
h_0	runway elevation above the reference geode (WGS84)

1.4 Reference Systems

This paragraph introduces the coordinate system of axes that will be used for modelling purposes [B5].

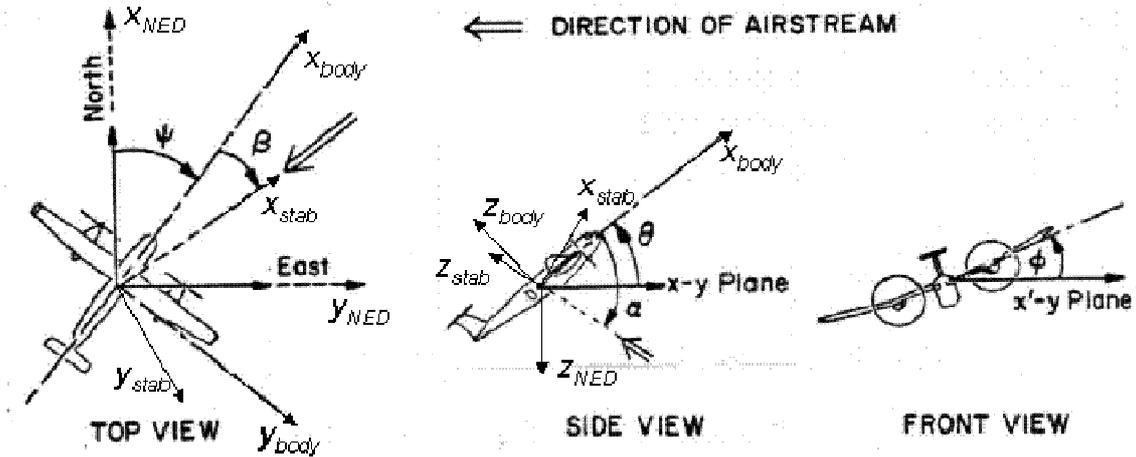


Figure 1.1 – Axis Coordinate Systems for an Aircraft

The *North-East-Down* (NED) coordinate system is used in combination with flat non rotating Earth assumption in order to be considered an inertial reference system. This orthogonal reference frame has its origin on a point of the Earth surface, typically assumed to be 0m altitude with a vertical axis (z) directed positively towards the Earth centre and the horizontal plane with x axis directed through the North and y axis directed to East. In this reference frame, the velocity vector of an aircraft is usually defined in spherical coordinates, with a vector module V , a *Track* angle χ that indicates the horizontal direction with respect to North (x NED axis) and a *Flight Path* angle γ that indicates the vertical direction with respect to the NED horizontal plane (negative downwards).

The *Body-fixed Reference* (BFR) is a body relative coordinate system of axes used to express aircraft attitude with respect to a horizontal plane. The origin is fixed in the vehicle Center of Gravity, with x body axis is aligned along the vehicle body and is usually positive toward the normal direction of motion. The y body axis is at a right angle to the x body axis and is oriented along the wings of the vehicle. If there are no wings (as with a missile), a "horizontal" direction is defined in a way that is useful. The y body axis is usually taken to be positive to right side of the vehicle. The z body axis is perpendicular to wing-body (XY) plane and usually points downward. The angles between the body-fixed coordinate system and the NED axes are called *Euler's angles*. Specifically, ϕ or *Roll* angle is the rotation that the body-fixed reference frame

shall perform around its x axes to align with the yz plane of NED, θ or *Pitch* angle is the rotation along y axis to align the xz plane and, finally, ψ or *Heading* angle is the rotation along z axis to align the xy plane.

A Sensor reference frame is a body relative frame usually used to express the measures of a specific sensor mounted on-board the vehicle. For the scope of this thesis it is useful to define the *Laser Sensor Reference* (LSR) frame. The origin is in the laser beam generator, with the z *laser* direct towards the laser beam (typically headed towards the terrain). The two laser mounting angles α_L and β_L are such as rotating the BFR along x axis of α_L and along the current y of β_L the z axis will coincide with the laser beam direction. The current x and y axes after this two rotations are the x and y axis of the LSR.

2 INTRODUCTION

In the field of aeronautics, one of the most important researches in recent years has focused on UAVs, Unmanned Aerial Vehicles. This term is defined for the special class of vehicles that fly without the aid of a pilot on board. But this simple and intuitive definition cannot be exhaustive since the acronym UAV encompasses a wide variety of different systems, so a univocal and concise definition of UAVs is not applicable.

We can affirm that UAVs are powered reusable aircrafts without crew, which can be remotely piloted or can fly autonomously or semi-autonomously. They can board a large variety of payloads and execute a large variety of specific tasks, for a certain period of time, determined by their mission and within or beyond the Earth's atmosphere.

This inclusive definition helps to distinguish clearly UAVs from missiles that a superficial first analysis could associate. In fact, although also missiles do not have human operator on board and usually are remotely piloted, they are not a "means" to carry a payload but they are the load themselves. Moreover a missile cannot be reused.

A remarkably wide spectrum of UAV configurations is currently in use or under development, ranging from fixed and rotary wing, going from micro to jet-sized. Initially they were simple drones, unmanned aircrafts used mainly in military operations controlled by a remote pilot potentially far thousands of miles away or even placed in another continent. The drones are improperly so called due to the typical noise produced by this kind of aircraft. But over the last decade, the advent of new sensor technology and the successes of already deployed platforms have bolstered a worldwide interest on developing and expanding the capabilities of UAVs.

The great importance of these vehicles and their growing use in various fields is mainly due to the absence of human operators on board, that is a very attractive feature in civil application but above all in military ones, since it allows to perform very risky missions without risking to lose human life and to execute lingering and boring routine tasks implying an unsustainable workload and stress to a human pilot.

The first experiment regarding an UAV was tried by an English pioneer, Archibald M. Low, who designed the so-called "aerial target", the aim of this experimental guided missile was to verify the possibility of using radio signals to drive a "flying-bomb" towards its target. This "radio-guide" device was developed and installed on small monoplanes powered by an engine of 35 horses. Two test flights were made around 1917 and, although in both experiments, the aircraft

crashed due to engine failure, the experiment showed that the radio-guide was feasible. Despite this important demonstration, the program "aerial target" was discarded because considered, wrongly, with a limited military potential.

In 1959 the United State Air Force worried of losing pilots in hostile territories began research activities for the use of unmanned aircraft (initially called RPV). UAV experimentation was intensified above all after that Lieutenant Gary Powers was shot down in the URSS skies during the last spy mission on soviet territory planned for his plane Lockheed U-2, made easily detectable by soviets radars due to its advanced technologies.

Lockheed U-2	Boeing X45
	
Mirach CAE	Alenia Sky Y
	

Figure 2-1 – The aeroplane Lockheed U-2 and some UAV models

Starting by the years 1980/1990, with the miniaturization and the development of applicable technologies, interest in the use of UAVs in the U.S. armed forces has grown significantly since they are considered the fighting machines cheaper and more capable compatibly with the least loss of life. By means of numerous flight experiments it began the systematic use of these vehicles by military forces in operations of such extreme delicacy: the war in the Balkans, the Gulf war, the war in Afghanistan and Pakistan, the war in Gaza, and more recently the war in Libya.

In order to increase safety, autonomy and reliability of UAVs many research projects have been launched around the world [B4]. In the U.S., for example, projects have been launched for the development of UAVs prototype more autonomy for carrying out flight missions that include more complex coordination between manned vehicles, UAVs and other military ground forces. Some of this 4 prototypes are part of the class UCAV (Unmanned Combat Air Vehicle) as the Boeing X45, others are of the class of HALE (High Altitude, Long Endurance) and others of the class Mini and Micro UAVs (MAV) with sizes comparable with a mosquito.

Despite the global leaders in the UAVs business are currently the United States and Israel, whose vehicles are engaged in missions of reconnaissance and combat operations, even Europe is investing heavily in this market and there are many national and international projects under way. The Italian industry has a long tradition in using drones for reconnaissance and combat exercises such as the series of Mirach Meteor, the Meteor CAE (now Galileo Avionics founded in 1974) pioneered firstly target aircrafts and then aircrafts suitable to observe the battlefield. Besides also Alenia Aeronautica started the development of the last generation MALE (Medium Altitude, Long Endurance) UAVs with prototypes such as Sky X and Sky Y.

Large industrial and commercial implications of UAVs and thus their evolution require large investments in research with the aim to achieve in the close future the possibility of easily use them for civil applications. Currently, however, civilian applications are very less widespread than military ones, in fact, UAVs could be a valid alternative to human missions only in specific cases because they are not still so safe and reliable to allow them for a convenient use in civil airspace where the rules dictated by the aviation authorities, are applicable, at the moment, only to aircraft with crew. However, there are several ongoing research and experimentation in order to improve the reliability of UAVs and define the rules to allow UAVs to fly in civil airspace (for example the project INOUI - Innovative Operational Uas Integration).

UAVs typically fall into one of six functional categories (although multi-role airframe platforms are becoming more prevalent):

- Target and decoy – providing ground and aerial gunnery a target that simulates an enemy aircraft or missile
- Reconnaissance – providing battlefield intelligence
- Combat – providing attack capability for high-risk missions
- Logistics – UAVs specifically designed for cargo and logistics operation

- Research and development – used to further develop UAV technologies to be integrated into field deployed UAV aircraft
- Civil and Commercial UAVs – UAVs specifically designed for civil and commercial applications

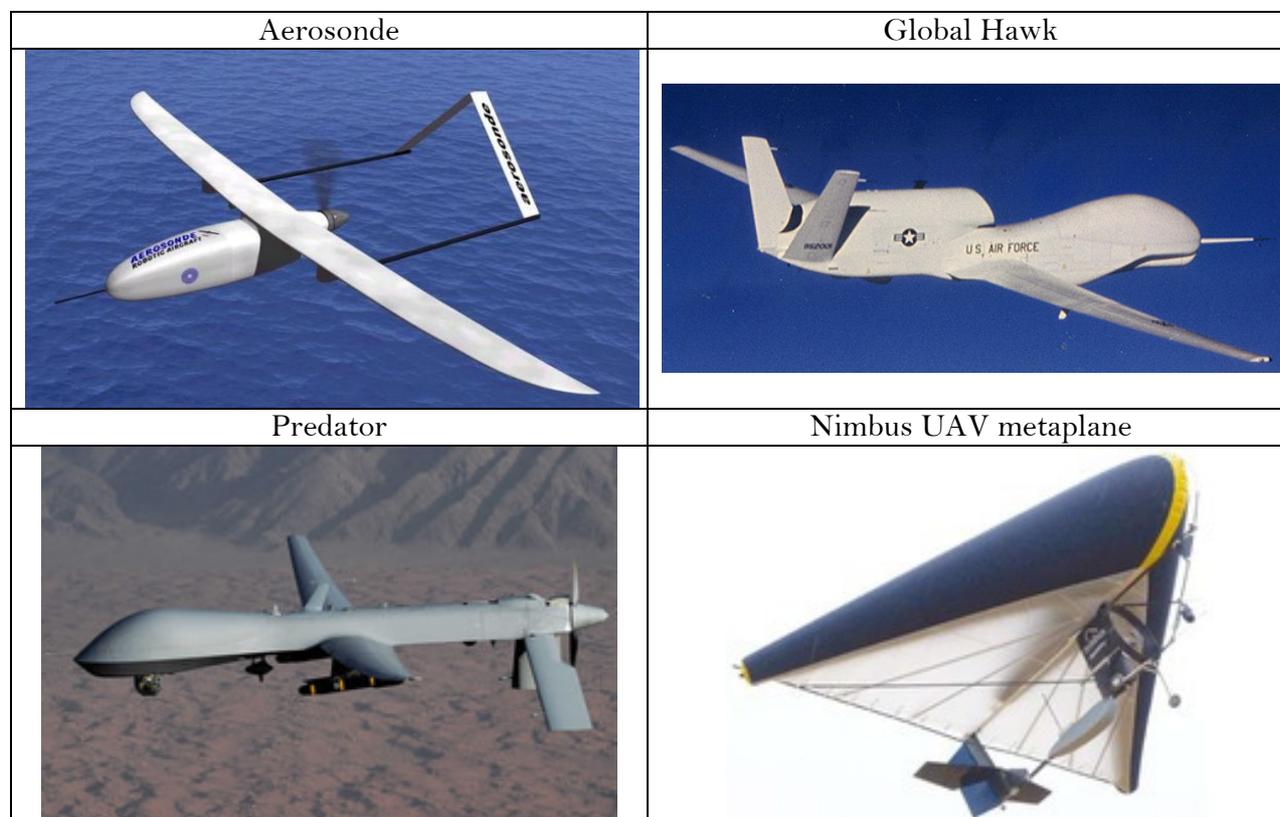


Figure 2-2 – Some of current UAV models

UAV remote sensing functions include electromagnetic spectrum sensors, gamma ray sensors, biological sensors, and chemical sensors. A UAV's electromagnetic sensors typically include visual spectrum, infrared, or near infrared cameras as well as radar systems. Biological sensors are sensors capable of detecting the airborne presence of various microorganisms and other biological factors. Chemical sensors use laser spectroscopy to analyse the concentrations of each element in the air.

Regarding the use of UAVs for not-military purposes some examples can be to use them as a tool for search and rescue, to find men missing in the desert, trapped in collapsed buildings, or lost in open sea. They can be used for surveillance of vast areas with low-cost systems, for example for the monitoring of numerous herds, for maintenance and security inspection of power lines, oil and gas pipelines or forest fire surveillance.

A growing use of UAVs has started also in civil application related to operation in areas too risky for manned aircraft: the NOAA (National Oceanic and Atmospheric Administration) began to use the unmanned system Aerosonde with the aim to hunt hurricanes. Aerosonde can fly into a hurricane and communicate near real-time data directly to the National hurricane Center in Florida, giving measurements very precise and detailed.

More recently, American UAV Global Hawk flew over the nuclear plant in Fukushima Dai-Chi in Japan, penetrating into the no-gone zone with the aim to monitor the reactors after the explosion caused by the famous earthquake this year and taking also photos with infrared sensors. The high radioactivity would have been made it impossible for human operators.

A last important use of UAVs is that related to the United States border control and the drug trafficking war. In 2011 it has begun a partnership between USA and Mexico to stem illegal immigration and drug trafficking across the border. The UAVs that will be used can fly at altitude of 18000 meters, are virtually invisible from ground and in a single day can minutely control an area of about 100000 square kilometers.

One of the key challenges that shall be faced by the research community relates to UAVs flying autonomously in an “Unstructured Environment” [B2]. “Autonomy” here refers to the absence of human intervention, and “unstructured environment” is associated with uncertainty both in the outside world (meteorological conditions, air traffic, fixed and moving obstacles) and in the vehicle subsystems (failures). Considering that the flight envelope of an airplane indicates the regions that an airplane can safely fly without any undue risk and with an acceptable margin of safety a main objective of aerospace research is, therefore, to employ many innovative techniques, such as enhanced vision systems, to expand it.



Figure 2-3 – Aircraft landing in presence of strong crosswind

Both for manned and unmanned vehicles, as it is demonstrated in some published statistics one of the most crucial phases of flight, considering the number of accidents and incidents the past decades, is the landing phase of flight. According to these statistics, close to 60% of the flight-mishaps occur in the landing phase [B8]. Causes for accidents could be divided into two main different categories. The first category is related to sensing errors, either human or sensor errors, such as altitude estimate error, runway conditions, and orientations. The second is due to sudden changes in atmospheric conditions. A famous example of an incident due to severe meteorological condition, shown in Figure 2-3, occurred at John F. Kennedy Airport in 1975 [B47]. Similar gust and wind shear conditions are responsible for a high number of hard landings and mishaps each year.

Traditionally, classic controllers, such as proportional–integral– derivative (PID) ones, need precise information about system dynamics and are sensitive to any changes in flight condition making it considerably hard to achieve an acceptable performance in a wide range of conditions. In fact, they have an effective capability just around the design point. Gain scheduling techniques have been used to solve this problem; however, it is necessary to implement an appropriate switching between gains, which is not possible in quick changing flight conditions such as landing in turbulent air.

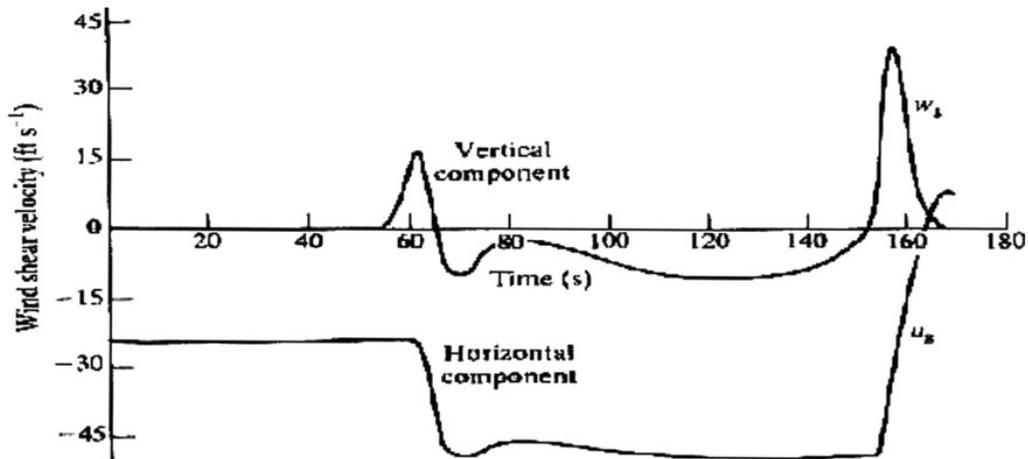


Figure 2-4 – JFK Airport downburst [B47]

Landing if compared with other flight phases implies considerably aerodynamics variation due to its vicinity to the ground, due to the existence of unknown patterns of wind and gust during the year, and finally due to natural obstacles and buildings and/or towers surrounding the airport. It is, therefore, desirable to develop a control system that can handle different climatic and situational conditions.

To design an autoland system implies considering a large number of aspects and integrate a large number of different algorithms. Such kinds of systems embrace all the disciplines typically faced by the Guidance, Navigation and Control systems developing area. In particular, in the current thesis will deal with the guidance problem of an autoland manoeuvre that has to take into account all the different flight phases and segments involving to manage very different flight envelopes and aerodynamic configurations.

One of the key technologies enabling UAVs crucial applications, as autoland, is *autonomous guidance and control*. Specifically, a feature of paramount importance is the capability of UAVs to autonomously generate a reference path taking the vehicle from a given point to a specified target. This should be accomplished by also accounting for several constraints arising from the vehicle limitations, i.e. maximum load factor, maximum roll angle, true air speed envelope. Path and mission constraints should be accounted for as well. This thesis deals with UAVs trajectory generation and tracking, both considering and not considering the presence of known no-fly zones.

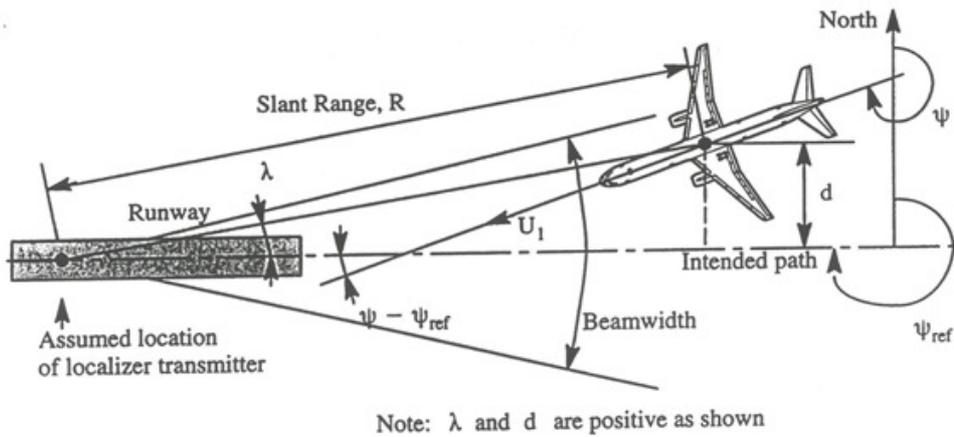


Figure 2-5 – Navigation sensors

Another GNC discipline faced in the current thesis is the *navigation sensor management*, intended to achieve with the required accuracy the knowledge on the state of the vehicle. The advent of new empowering stand-alone sensors, such as the laser range finder and GPS has encouraged the adoption of onboard positioning and enhanced the navigation systems. These systems are currently recognized as instrumental in bringing about all the capabilities of an UAV to perform high precision tasks in challenging and uncertain operation scenarios. Several different methods have been proposed and flight tested (as for example in [B44]) confirming the expected robustness and performance that can be achieved in the execution of specific maneuvers, such as landing or steering the vehicle to a desired target.

Nowadays airports are equipped with runway approaching systems that provide lateral and vertical guidance to aircrafts during the glideslope and final landing maneuver. A common mechanism is the so-called Instrument Landing System (ILS) [B45] that is composed of several radio beacons placed on the runway, allowing for vertical and lateral accurate guidance of the aircraft during the landing phase. Once the approaching maneuver starts, the ILS guides the vehicle to a certain height, referred to as the decision height, which depends on the airport's ILS category and on the ILS based guidance system available onboard the aircraft. Different ILS categories provide different levels of autonomy to the aircraft runway approach and landing system. The most advanced one, ILS Category IIIc, allows for the automatization of the entire maneuver including guidance along the runway. Despite the availability of those advanced landing systems, their complexity and the high cost involved on their implementation turn them into prohibitive solutions for small UAVs which should be able to land on any opportunity runway, grassy strip, or available road, resorting to low cost onboard navigation systems. These systems provide the vehicle's automatic landing guidance and control algorithms with the actual vehicle's linear and angular positions and velocities, and dedicated modules allow for the

integration of GPS/INS information with height data as acquired by Radar or Laser Altimeter mounted underneath the aircraft.



Note: γ as shown is negative.

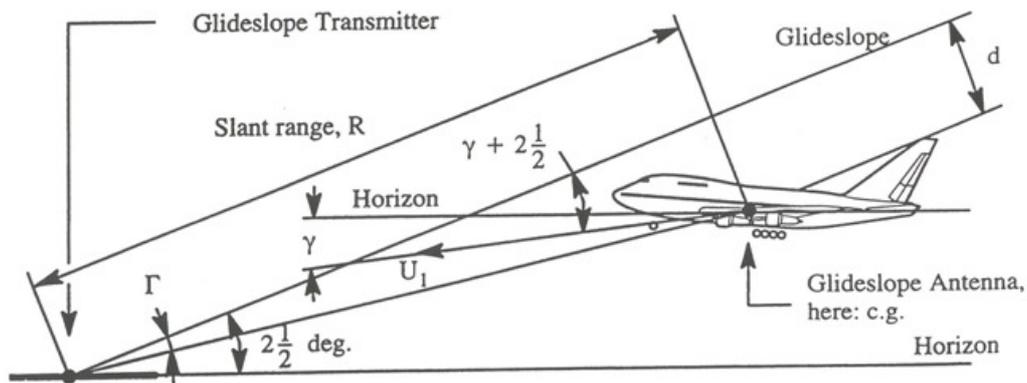


Figure 2-6 – ILS Localizer and Glideslope Tracking Geometry

Highly precise navigation is the core technology required also for many other applications, such as automated aerial refuelling (AAR), sea-based joint precision approach and landing systems (JPALS), station-keeping, unmanned aerial vehicles (UAV) swarming and formation flight and unmanned ground vehicles (UGV) convoys. Advances in the above mentioned technology are possible considering the future GNSS framework, given that adequate characterization of new GNSS devices are performed and that new algorithms are developed that fully exploits the functionalities made available by the future GNSS systems. In this thesis both aspects are considered, with specific reference to the use of GPS/EGNOS for reliable fixed wing aircraft automatic landing applications.

For what concern experimental characterization of the satellite based navigation system GPS/EGNOS, it would be interesting how exploit satellite data for computing and analysing the

performance in terms of Required Navigation Performance requirements RNP-RNAV [B32][B9].

The thesis is organized as follows. Chapter 3 provides an overview of the current state-of-the-art on all the types of algorithms involved in an autoland system. Chapter 4 includes a detailed description of the proposed innovative autoland algorithms. Chapter 5 describes the development and verification tools, including some specific models developed in the framework of the author Phd and integrated in the laboratory test rig of the CIRA flight demonstrator FLARE. Chapter 6 contains a summary of laboratory and flight results and an overview of the development process followed. Chapter 7 presents some concluding remarks.

3 SURVEY OF AUTONOMOUS APPROACH AND LANDING ALGORITHMS

As already mentioned in the previous chapter an autoland system is composed by several modules. The main sub-modules are the trajectory generator, the trajectory tracking or guidance module, the navigation sensor management module and the flight control law module. Coherently with this subdivision also the current and the following chapters will be subdivided in different paragraphs concerning the main sub-modules of an autoland system. In particular the first paragraph of the current chapter will deal with the state of art for complete autoland systems with focus on trajectory generation and tracking aspects for the final phases of the autoland manoeuvre. The following paragraphs of the current chapter will treat the trajectory or path generation in mid-air phase for reaching the way point aligned with the runway (typically known as terminal area operations phase in civil application procedures). The considered path generation algorithms survey will be further divided in algorithms considering or not considering the no-fly zones. Finally it will be described the state of art for sensor fusion algorithms related to optimal estimation of vehicle variables essential for autoland manoeuvre and it will be deepened the EGNOS (European Geostationary Navigation Overlay System) and the DEMs (Digital Elevation Maps).

3.1 Algorithms for Autonomous Landing Systems

Different approaches have been proposed to address the aircrafts automatic landing, based on the use of both modern intelligent control techniques and classical control theory [B35][B36][B37][B46], and aerospace companies and research centres have developed programs in the field of autonomy in UAVs, also covering the automatic landing issue [B38]. **Errore. L'origine riferimento non è stata trovata.**[B40].

All the main approaches available on the state of art are based on a pre-fixed trajectory in the space divided in several segments whose tracking is demanded to the control system. Typically, in all these cases, the control strategy has to efficiently reject all the disturbances in a very large envelope as that encountered during a complete approach and landing manoeuvre. In this sense the control strategy adopted has to manifest a certain degree of adaptivity to the external condition and to the different aerodynamic configurations experienced by the aircraft. In [B40] an un-powered automatic landing of a representative RLV (Reusable Launcher Vehicle) configuration has been considered. In this case the fixed trajectory is tracked by means of a control system H_{INF} -based.

Depth studies to design the autoland trajectory in relation to dynamics features of the vehicle and predicted whether condition on the runway have been carried out for the main re-entry NASA vehicles [B41][B42]. [B41] refers to the X-34 Mach 8 unpowered NASA vehicle and has the aim to develop an autoland trajectory enforcing physical constraints such as loads, vertical descent rate, continuity, and smoothness in to the design problem up to obtain a two-point boundary value problem with conditions on the initial and final dynamic pressure. Finding a solution, in this case has required developing trajectory simulation techniques that constrained the flight profile to a prescribed geometry. Also in [B42] is present a similar approach in this case the vehicle considered is the NASA shuttle and the approach to simulating trajectory is to use a guidance scheme (employing feedback control) to track the desired geometric profile. This is essentially the method used by the Autoland Shaping Processor that predicts touchdown conditions for the Shuttle. The main disadvantage of using this technique is an excessive complexity for both the design and the subsequent analysis. While the approach in [B41] for constructing a trajectory of well-defined geometric tends towards rapid trajectory design of an on-board reference for the flight guidance system. Anyway both methods above described for RPVs imply an iterative solution that could not be compliant with the computational capability of a MALE UAVs or of a GA A/C avionic system.

In [B38] is presented a standard trajectory generation for an autoland system developed for the IAI UAV. In this case the tolerance to faults and to unpredicted whether condition is achieved by means of a flight control module based on neural networks.

[B43] addresses an autoland control law synthesis problem deriving in a straightforward manner some H_2 controllers for the specific operating regions in which is possible divided the whole landing manoeuvre and using a Linear Matrix Inequalities (LMIs) approach based on the concept of quadratic stability for LPV (Linear Parameter Varying) systems. The idea is to exploit the LPV models representing a compromise between the global accuracy of nonlinear models and the straightforward controller synthesis techniques available for Linear Time Invariant (LTI) representations.

Also in **Errore. L'origine riferimento non è stata trovata.** is presented an autoland system based on a pre-fixed trajectory in the space. The authors recognize that a simple waypoint guidance is not sufficiently robust against the effects of wind and does not control height accurately enough to support an autoland function. Therefore a particular guidance algorithm is developed that uses the perpendicular distance and velocity from the demanded flight path to calculate demanded manoeuvre acceleration. The current vehicle speed and orientation is then

used to convert this acceleration vector to a demanded roll and pitch rate. The landing trajectory is defined by a straight line linking two waypoints, with the end waypoint located at the touchdown point and the vehicle is flown along the trajectory at the specified landing speed. During this phase of flight, the laser altimeter is used to correct height errors in the navigation solution which enables initiation of the flare manoeuvre at the correct height and minimises along track dispersion of the touchdown point. The flare manoeuvre begins when the vehicle altitude passes an altitude threshold that is set adaptively using estimated navigation errors and descent rate information.

Many research activities have been conducted on designing an automatic landing controller suitable for different classes of aircraft, especially heavy jet transports. For example, in [B47], an automatic landing system based on a human skill model is described. In [B48] a mixed H₂/H_{inf} control technique has been employed to develop controllers for the autoland of a commercial airplane.

It is known that the straight forward solution to the optimal control problem leads to Two Point Boundary Value Problem (TPBVP), defined by means of the Riccati equation, which is usually too complex in solution. In [B49], by means of optimal control theory, seven neural networks were trained to learn the costate variables of the system and estimate them in similar scenarios without using the final time value to avoid solving Riccati equation backward in time. The costate variables typically can be interpreted as Lagrange multipliers associated with the state equations of an optimization problem. The state equations represent constraints of the minimization problem, and the costate variables represent the marginal cost of violating those constraints; in economic terms the costate variables are the shadow prices.

However different case studies conducted by researchers have resulted in a point that, although it would not be necessary to solve the Riccati equation in the presence of gusts, the optimal-based neuro-controller does not normally have reasonable robustness. In [B50], five different types of neural network structures are used to design intelligent autoland controllers using linearized inverse dynamic models, in this work researchers tried to show how the type and complexity of a neural network is effective for landing flight phase. In [B51] an adaptive controller based on model reference adaptive system (MRAS) methodology has been designed for a flight vehicle that enables it to track a predetermined flight-path trajectory in the presence of strong wind shears.

[B35] focuses on developing a flexible human knowledge-based controller, which has the ability to adapt its performance based on changes in flight condition during landing. Its objective is to propose a basic controller design with performance and stability during landing phase of flight in the presence of strong wind shears. With this aim classical, neural-based, fuzzy-based, and adaptive controllers are compared to show the merits and weaknesses of each in the presence of different wind shears and very strong wind patterns. Simulation results show that both fuzzy-based and adaptive controllers meet the necessary performance requirements and have acceptable robustness; however, the adaptive controller due to its large number of fluctuations in control signal is hard to implement. The fuzzy-based controller satisfies all necessary conditions for the selected performance specification and is a good candidate for expanding the flight envelop of aircraft in the landing phase of flight.

However, also if the concept of learning capable control system stems deserves considerable research and attention, they are not still usable in certified equipment because the European authorities are at the moment far to accept as enough safe the available clearance methods on this kind of controllers.

3.2 Algorithms for Optimal Path Generation

3.2.1 Algorithms for Optimal Path Generation without No-Fly Zones

Considering trajectory generation and tracking without presence of known no-fly zones valuable literature exists that is mainly focused on path planning generation for robots in the presence of known obstacles (see [B20] and references therein).

In this thesis is treated an algorithm for the path generation that relies upon Dubins ideas to generate a suboptimal 2-D path satisfying initial and terminal conditions, specified in terms of position and heading angle. In the following is reported a survey on related work. The Dubins solution for the 2-D case was first proposed in [B3]; in [B11][B12] the authors base upon Dubins results to generate 2-D paths satisfying kinematic and tactical constraints; in [B13] an extension is proposed that guarantees the path passing through (or comes close) assigned waypoints.

Recent approaches for path generation and tracking have been proposed in [B14], where the authors introduce a method for finding fuel-optimal trajectories for spacecraft to requirements such as avoidance of collisions with obstacles or other vehicles, using a mixed-integer linear program (MILP) problem formulation; in [B15] where the author presents a novel framework for

safe online trajectory planning of unmanned vehicles through partially unknown environments; in [B16] where a technique is presented for creating continuously parameterized classes of feasible trajectories; in [B17] where a randomized path planning algorithm for autonomous vehicles in the presence of fixed and moving obstacles is presented; in [B18] where the authors propose a planning strategy that takes as input a 3-D sequence of way-points connected by straight flight trim conditions, and then smooths it in an optimal way.

3.2.2 Algorithms for Optimal Path Generation considering No-Fly Zones

Some of the emerging methods for optimal path generation considering no-fly zones are based on geometric approaches such as roadmap, cell decomposition and potential fields [B21][B22][B23][B24][B33]. Roadmap approaches are very common and include methods like Probabilistic roadmap [B25][B26] or more efficient techniques such as Rapidly Exploring Random Trees [B27][B28][B29]. The latter methods have the common feature of representing the search space of the solution through cells or nodes and, in particular environments, they may not perform satisfactorily unless the complexity is increased and the path planning becomes time consuming.

Indeed, the most important issue to be addressed is the computational load of path planning algorithm that must be compliant with the performances of flight control computer for real time applications during flight. The price one must pay for the improved accuracy is therefore the increased run time and memory consumption. This is the reason why some methods guaranteeing the optimality of the solution have a computational load which makes them not suited for real time applications (i.e. Visibility Line and similar approaches [B30][B31]). Generally speaking, it has been demonstrated that this class of problems is NP-complete [B32] and thus its complexity grows more than polynomially with respect to the number of assigned constraints, namely the number of no-fly zones. This makes the optimization problem very hard to solve, hence some simplifications and approximations are needed for developing an algorithm able to find at least a sub-optimal solution in real time.

Over the past years some efforts have been spent to simplify the problem of optimum path planning (also at the expense of the optimality) in order to allow the real time implementation of path planning capabilities. In fact, while an off-line approach can be useful to guide safely the UAV through the target point in nominal condition (i.e. when the size and position of no-fly zones is known before flight and it does not change during flight), it clearly lacks the necessary adaptivity in presence of forbidden areas and/or obstacles that may occur during flight.

Nevertheless, despite availability of several approaches for path planning, the problem of computational complexity is still an open issue.

3.3 Sensor Fusion Algorithms

Currently, no single sensor is capable of reliably realizing the required performance for autoland phase (RNP criteria) without relying on some ground measurement, hence UAVs navigation requirements can be fulfilled only by integration of measurements from multiple sensors. In particular, configurations that integrate inertial sensor measurements with GPS, altimeters, air data sensors, and magnetometers are very frequent ([B53][B54][B55]), and resulting performance and reliability depends on both sensors accuracy and adopted integration techniques.

The most common integrated navigation techniques make use of Kalman filtering ([B56][B57][B58]). The main drawback of these techniques is the necessity of an accurate sensor error model, so as when poor information are available about the sensors used it is very difficult to obtain an appropriate adjustment of the Kalman Filter.

	Advantages	Drawbacks
GPS (low frequency)	High accuracy in differential mode Long-term accuracy Deterministic solution to the fix position problem	Satellites visibility Jamming and/or cycle-slip Fast dynamics not captured Slow output data rate to determine a trajectory No attitude information Higher noise than INS
INS (high frequency)	No jamming High output data rate Works in every environment Fast dynamics captured Lower noise than GPS	No long-term accuracy Start-up calibration required

Table 3-1 - Main features of GPS and INS technologies

For what concerns, generally, a sensor fusion algorithm design, in a navigation system based only on GPS measures, typical error sources are:

- excessive noise,

- low updating frequency (generally up to 10 Hz), therefore for dynamics the faster measures turn out little reliable,
- satellite “loss”.

To overcome these limitations, a good solution consists in the integration of the GPS position and speed measures with the ones coming from other sensors: the auxiliary system most commonly used to such aim is an INS sensor. In Table 3-1, main features of both GPS and INS are shown and compared. In general it can be stated that the advantages arising by the INS/GPS integration are:

- advanced accuracy on position and speed,
- limitation of the INS errors by means of the GPS measures,
- possibility of using INS when the GPS is unusable,
- the velocity measure from the INS can help to eliminate GPS jamming problems,
- the position measure from the INS can help to reduce the acquisition time of the satellites (time to hot fix),
- the high INS short-term accuracy can eliminate the cycle-slip problem,
- thanks to GPS, INS can be economic and more compact.

As far as GPS sensor is concerned, the accuracy of the related measures can be improved considering receivers compliant with the recent satellite augmentation systems. In the following paragraph is presented a brief description of the EGNOS system with particular attention to the integrity concepts.

For satisfying the RNP criteria for autoland system is essential also achieving an optimal estimation of the “Above Ground Level” and the “Above Runway Level” of the vehicle. This kind of estimation necessitates the use of sensors of different nature and proper sensor fusion logics to enable efficient combination of their measures. In particular in this thesis an algorithm to fuse satellite, laser altimeter and DEMs (Digital Elevator Maps) measures is proposed. In this case there wasn't a valid alternative to the Kalman filtering to achieving optimal estimation of the vehicle position related to ground and runway. . In the paragraph §3.3.1 is presented an overview on the EGNOS system while in the paragraph §3.3.2 is presented the state of the art for DEMs devices.

3.3.1 EGNOS System Overview

EGNOS is the first European initiative in the satellite navigation field. EGNOS was mentioned for the first time in 1994, in a communication from the European Commission. This was

followed by the December 19, 1994, resolution by the Council of European Union to define the terms of the European Commission, European Space Agency, and EUROCONTROL.

The EGNOS program is an integral part of the European satellite radio-navigation policy. It is currently under the control of GALILEO Joint Undertaking (JU). The aim of EGNOS, like the other SBAS services, is to provide complementary information to the GPS and GLONASS signals to improve the RNP (Required Navigation Performance) parameters.

According to the integrated strategic vision for the provision of European GNSS, new services can be conceived as a result of combining GALILEO Satellite-Only Services (GSOSs) [B59][B60]. The latter provides ranging service, wide area differential corrections and integrity. The combination of EGNOS service with the GALILEO Service of Life (SoL) is of special interest. The combined services will provide independent and complementary integrity information on GALILEO and GPS constellations that may support, for instance, precision approach type operations in the aviation domain.

The overall system architecture is divided into three segments (Figure 3-1) [B61]: Space segment, Ground segment and User segment.

The space segment consists of three Geostationary Earth Orbit (GEO) satellites that provide triple coverage over Europe, the Mediterranean, and Africa.

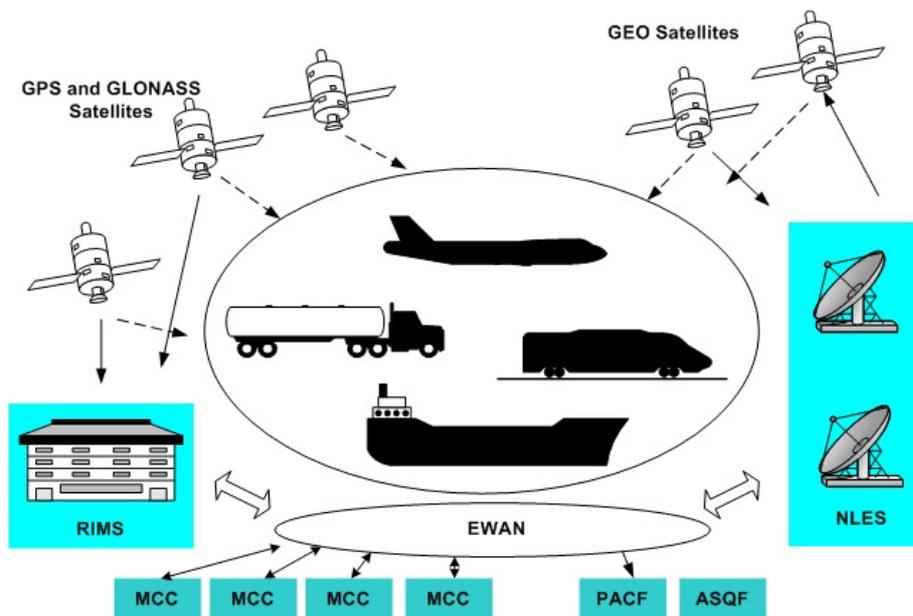


Figure 3-1: EGNOS overall system architecture

The ground segment includes the following elements:

- Four Mission Control Centres (MCC) that include Central Control Facility (CCF) and Central Processing Facility (CPF);
- Thirty-four ranging and Integrity Monitoring Stations (RIMS);
- Six Navigation Land Earth Stations (NLES);
- The Application Specific Qualification Facility (ASQF);
- The Performance Assessment and system Checkout Facility (PACF);
- The EGNOS Wide Area communication Network (EWAN)

These elements are distributed over the European territory and surrounding continents as shown in Figure 3-2. The RIMS measure satellite pseudoranges (code and phase) from GPS/GLONASS and SBAS GEO satellites signals. The raw measurements are transmitted to the CPF, which determines the wide area differential corrections and ensures the integrity of the EGNOS system for users.

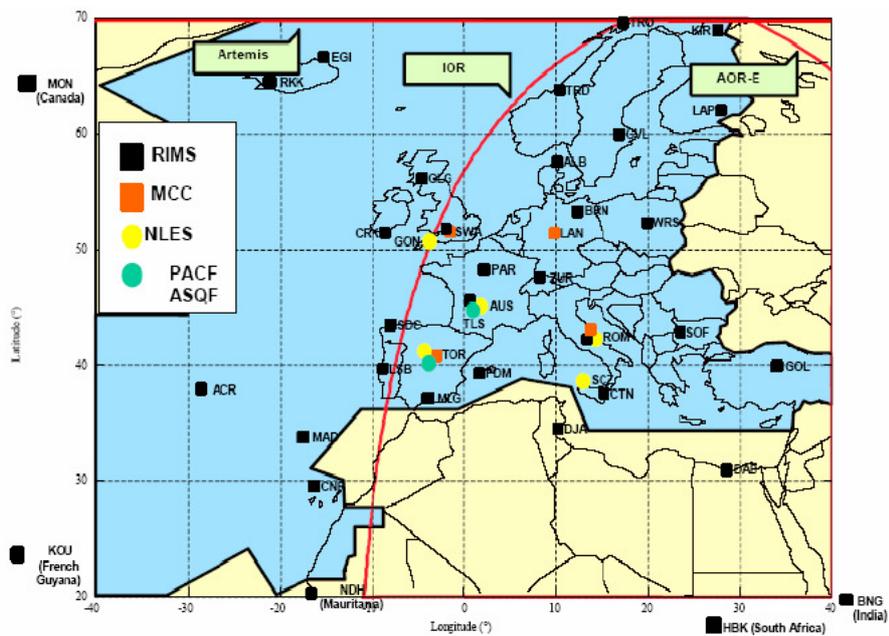


Figure 3-2: EGNOS Ground Segment Architecture

The EGNOS user segment is composed of a GPS and/or GLONASS receiver and EGNOS receiver. The two receivers are usually embedded in the same user terminal. The receiver can process the message that is scheduled in a 6-second duty cycle time. The EGNOS message includes more slowly changing errors, such as long-term satellite clock drift, long term orbital error correction, and ionosphere delay corrections and fast correction (rapidly changing errors, such as satellite clock errors) in the same frame, as showed in the Figure 3-3.

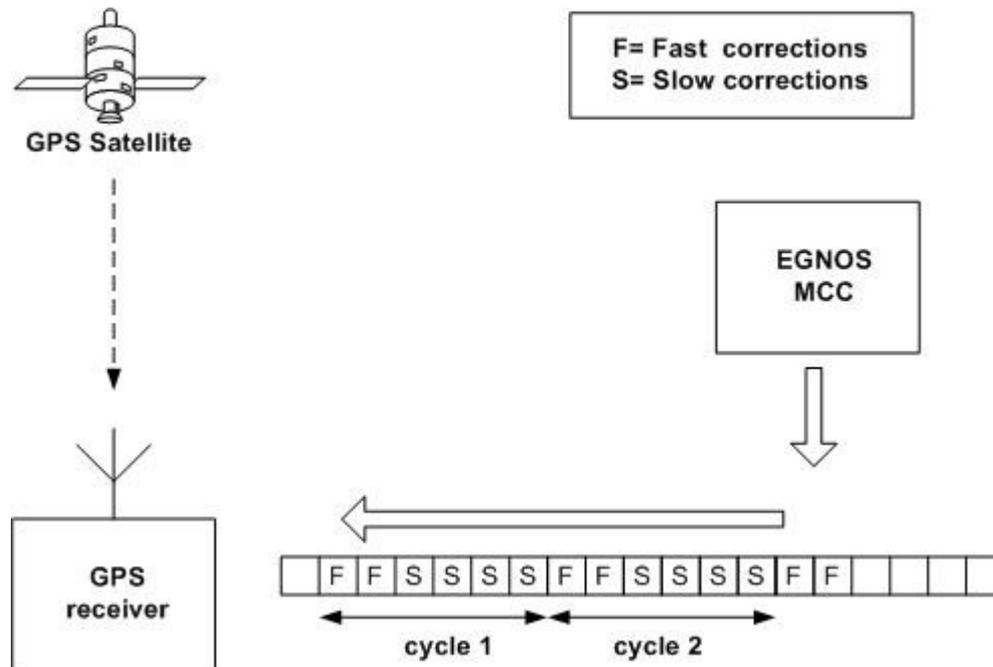


Figure 3-3 - EGNOS message composition

A given EGNOS GEO SATELLITE broadcasts either coarse integrity data or both such data and wide area corrections.

The coarse integrity data includes use/don't-use information on all satellites in view of the applicable region, including the GEOs. Correction data include estimates of the error after application of the corrections. The parameter, σ_{UDRE}^2 , is the variance of a normal distribution associated with the user differential range error for a satellite after application of fast corrections and long term corrections, excluding atmospheric effects. The parameter, σ_{GIVE}^2 , is the variance of a normal distribution associated with the residual ionospheric vertical error at an Ionospheric Grid Point (IGP).

Coherently with EGNOS system characterization the accuracy of a navigation system is defined in term of Total System Error TSE which is referenced to a required flight path defined for each phase of flight. To follow the required path, the aircraft navigation system estimates the aircraft's position and generates commands (either to a cockpit display or to the autopilot). Error in the estimation of the aircraft's position is referred to as Navigation System Error NSE which is the difference between the aircraft's true position and its displayed position (see Figure 3-4). The difference between the required flight path and the displayed position of the aircraft is called Flight Technical Error FTE and contains aircraft dynamics, turbulence effects, man-machine-interface problems, etc. The vector sum of the NSE and the FTE is the Total System Error. Since

the actual Navigation System Error cannot be observed without a high-precision reference system (the NSE is the difference between the actual position of an aircraft and its computed position), an approach has to be found with which an upper bound can be found for this error.

Horizontal Protection Level: The Horizontal Protection Level (HPL) is the radius of a circle in the horizontal plane (the plane tangent to the WGS-84 ellipsoid), with its center being at the true position, which describes the region which is assured to contain the indicated horizontal position. It is the horizontal region for which the missed alert requirement can be met. It is based upon the error estimates provided by EGNOS.

Vertical Protection Level: The Vertical Protection Level (VPL) is half the length of a segment on the vertical axis (perpendicular to the horizontal plane of WGS-84 ellipsoid), with its center being at the true position, which describes the region which is assured to contain the indicated vertical position. It defines the vertical region for which the missed alert requirement can be met. It is based upon the error estimates provided by EGNOS.

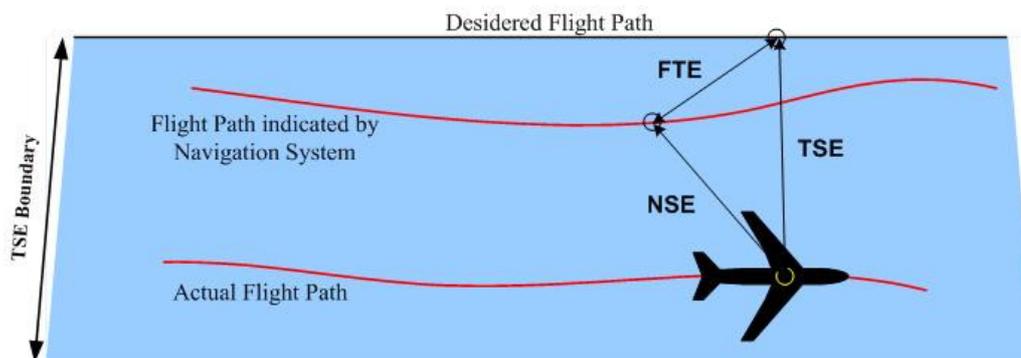


Figure 3-4 - Navigation System Error; Flight Technical Error and Total System Error

The computed protection levels [B62] must be compared to the required Alert Limits (AL) for the particular phase of flight. If the protection level is smaller than the required generic alert limit, then the phase of flight can be performed. However, if the protection level is greater than or equal to the required alert limit, then the integrity of the position solution cannot be guaranteed in the context of the requirements for that particular flight phase. If $XPL < XAL$ integrity can be assured, if $XPL \geq XAL$ integrity cannot be assured. Where with XPL we have denoted the Horizontal or Vertical Protection Level and with XAL we indicate the Horizontal or Vertical Alert Limit. The corresponding situation in the Horizontal plane is depicted in Figure 3-5.

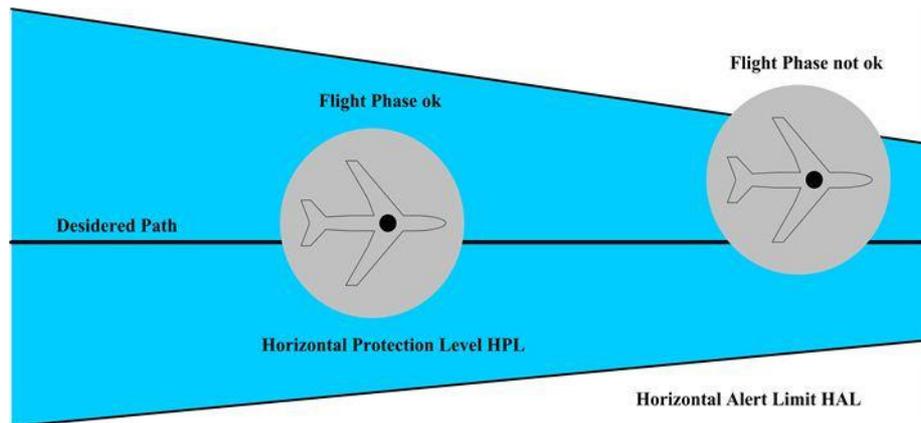


Figure 3-5 - Horizontal Protection Levels and Horizontal Alert Limits

It should be noted that the main significance using this approach is not the computation of the protection levels and their comparison with the corresponding alert limit.

The major interest should be considered to be on the assurance that the computed protection levels represent an upper bound on the NSE with a certain confidence. “Misleading Information” results only, if the NSE is greater than the alert limit and the protection level does not indicate this fact (for a more complete and detailed description of the “overabounding concept” and problems resulting of it, refer to [B63]).

As defined, an *Integrity event* is an epoch where the Position Error (PE) exceeds a maximum allowable limit, called the Alert Limit (AL) while no alert is generated within an allowable time period, called the Time to Alert (TTA).

Thus, every situation where the position error exceeds the protection level is reported. Hence the worst case scenario is considered and the analysis stays on the conservative side.

- *Misleading Information (MI)*: Misleading Information (MI) event is considered every epoch where $PE > PL$ which can be regarded as a system anomaly.
- *Hazardous Misleading Information (HMI)*: Hazardous Misleading Information (HMI) event is considered every epoch where $PE > AL$ and $AL > PL$ which can be regarded as a system anomaly that is hazardous for a specific user (note that AL can differ for different users/operations).
- *Near Misleading Information (Near MI)*: Near MI event is considered every epoch where $PE/PL > 0.75$.

- *Integrity pass criteria:* If one or more MI or HMI is present in a data set, the first glance test will be failed and an investigation into the causes should be performed.

From all valid samples all the Misleading Information (MI) events are determined based on samples with XPE>XPL. Horizontal and vertical events are counted separately and the total is determined by counting all events for which HPE>HPL OR VPE>VPL.

For each operation the same is done but now the Hazardous MI (HMI) are counted according to XPE>XAL>XPL.

3.3.2 Digital Elevator Map (DEMs) devices currently available

In this paragraph a study about currently available DEMs, techniques used to achieve them, characteristics of coverage, data sampling, accuracy, and related costs is carried out. In particular, the attention is focused on the three most used models, developed in the framework of SRTM ([B65][B66][B67]), ASTER ([B68], available for free download) and Intermap Star Technology ([B69]) programs. The results of some comparative experimental studies [B69][B70][B71] about the characteristics of accuracy of these DEMs also were examined, showing the significant improvement achieved by NEXTMAP data (produced by Intermap) with respect to their predecessors (against the cost of the product, not very cheap for the end user). These results are summarized in the following table.

	SRTM (NASA)	ASTER (METI)	NEXTMap (Intermap)
Collection Method	Interferometry	Photogrammetry	Interferometry
Platform	Shuttle	Satellite	Airplane
Ground Sampling Distance (m)	30 for USA 90 for other countries	30	5
Published Accuracy RMSE (m)	16	20	1
Cost	free	free	20 - 30 euro for square kilometer

Table 3-2 - Comparison of different Digital Elevation Models

4 PROPOSED AUTONOMOUS APPROACH AND LANDING ALGORITHMS

Based on the analysis of the state of the art on approach and landing algorithms described in the previous chapter in the following are reported some preliminary considerations used to design the proposed innovative algorithms.

As far as the guidance and control algorithms is concerned, it is clear that the greatest effort carried out by the aerospace research community regards innovative techniques for control system development as genetic algorithms, neural networks, fuzzy logic, optimal control, etc. The input references for all the possible developed flight controllers are generated by guidance modules using a prefixed nominal trajectory in the space. In this field the algorithms proposed aim for generating (and, if necessary, re-generating) in real time the trajectory during the mission being able, in this way, to consider and properly evaluate all the current vehicle and path constraints. In fact these constraints could vary during the mission and the initial assumptions on them could be not more valid. Moreover, nowadays, the innovative techniques for control system development above mentioned are not compliant with the certification guidelines imposed by the competent European authority (EASA) and, therefore, the related system would not be suitable for being commercialized as avionics products in the close future.

The feature above defined has been used for all the phases related to an autoland maneuver and in particular two algorithms have been developed to generate an optimal trajectory starting from the top of descent (final waypoint of the cruise phase) up to the final waypoint aligned with the runway (initial waypoint of the proper approach phase), a further algorithm has been developed for generating the flare trajectory up to the touchdown point.

Considering instead the navigation sensor management, and in particular the positioning estimation with accuracy defined by the aeronautical requirements [B9][B32][B10], the most common integrated navigation techniques make use of Kalman filtering. The main drawback of these techniques is the necessity of an accurate sensor error model, in fact when poor information is available about the sensors it is very difficult to obtain an appropriate adjustment of the Kalman Filter. With the aim of developing a low cost navigation sensor suite, the choice has been to use a simpler sensor fusion algorithm based on the concept of complementary filtering that will be described in the next.

Regarding the ARL (Above Runway Level e.g. the altitude of the vehicle above the runway) estimation the related state of the art doesn't present many works regarding sensor fusion of

more sources. More research activities have been carried out for developing DEMs (Digital Elevator Maps) as accurate as possible, altimeter more reliable and satellite data with high accuracy and integrity. The idea on which is based the proposed algorithm is to integrate these three kind of measures to optimal estimate the desired ARL with an accuracy better than that obtainable with the single sensors and with the further aim to make the system capable to tolerate a single failure of one of this sensors. For the ARL estimation has been selected the Kalman filtering as the optimal sensor fusion method due to a lack of valid alternatives for this particular purpose.

Summing up, the autonomous landing system here proposed has to be compliant with the following main features:

- capable to perform a fully autonomous landing starting from any point of three dimensional space typically representing the top of descent waypoint in the aeronautical procedures;
- fully adaptive during the flare phase, in the sense that it has to be able to generate on line, with a desired updating rate or in case of a pre-selected driven event, the nominal trajectory for the aircraft, based on the actual state of the vehicle and on the desired state at touch down point;
- capable to perform the automatic landing manoeuvre using a navigation system constituted by a DGPS, an AHRS (Attitude and Heading Reference Systems), an ADS (Air Data System), a Laser altimeter and a DEMs (Digital Map Elevator), in such a way to require only a weakly instrumented landing runway, which must be only equipped with the differential GPS rover station;
- capable to manage failures and/or adverse atmospheric conditions by means of decision-making logic evaluation for key-decisions regarding possible execution of altitude recovery manoeuvres based on the Differential GPS integrity signal and compatible with the functionalities made available by the future GNSS system.

With the general aims above described the author, during his Phd period, has contributed to the general work carried out at CIRA for developing an innovative GNC system with the capability of autonomous approach and landing. All the proposed algorithms have been integrated by the CIRA TECVOL team in a complete GNC SW, executed in a Flight Control Computer tested at the CIRA hardware-in-the-loop laboratory and, finally, integrated in the avionic setup of the CIRA Flying demonstrator FLARE. The laboratory and the flying demonstrator will be described in the chapter 5.

In the next paragraph will be presented the on-board functional architecture of the GNC SW above mentioned and in the following paragraphs will be described all the proposed algorithms.

4.1 On-board SW Functional Architecture

GNC architectures of current UAV mostly resemble a command and control avionic architecture of manned aircraft [B6][B7]. As already mentioned in the chapter 3, the current functional SW architectures support current UAV missions only with a limited degree of automation.

These architectures are mainly based on the data link presence whereby the on-ground remote pilot can directly command UAV actuators (like a manned aircraft), or can command an augmented aircraft (through the FCL module) as it is currently done in modern *Fly-By-Wire* aircraft (Airbus and modern Boeing commercial aircraft). Besides the remote pilot can use a virtual cockpit animated by a live video camera placed in a position inside the UAV to obtain same visual of an on-board pilot. Possibly, during the manual piloting, the remote pilot can also decide to use autopilot modes to easy some routine tasks or to execute portions of the flight activating FMS to follow a pre-determined (or changed on-line) flight plan.

The functional architecture proposed in the current thesis, and developed by the overall TECVOL project CIRA team, tends to add further degree of automation to the above one, allowing for automatic take-off and landing (that are normally performed by remote pilot in direct manual mode), for automatic reconfiguration of GNC functions upon some failure conditions, for automatically generating flight trajectories based on a given flight plan inclusive of both way points and no-fly zones and, finally, for managing selected emergency situations by performing direct abort missions or other automatic mission changes. In the following figure is reported a functional architecture that is able to perform all the above advanced features [AR9].

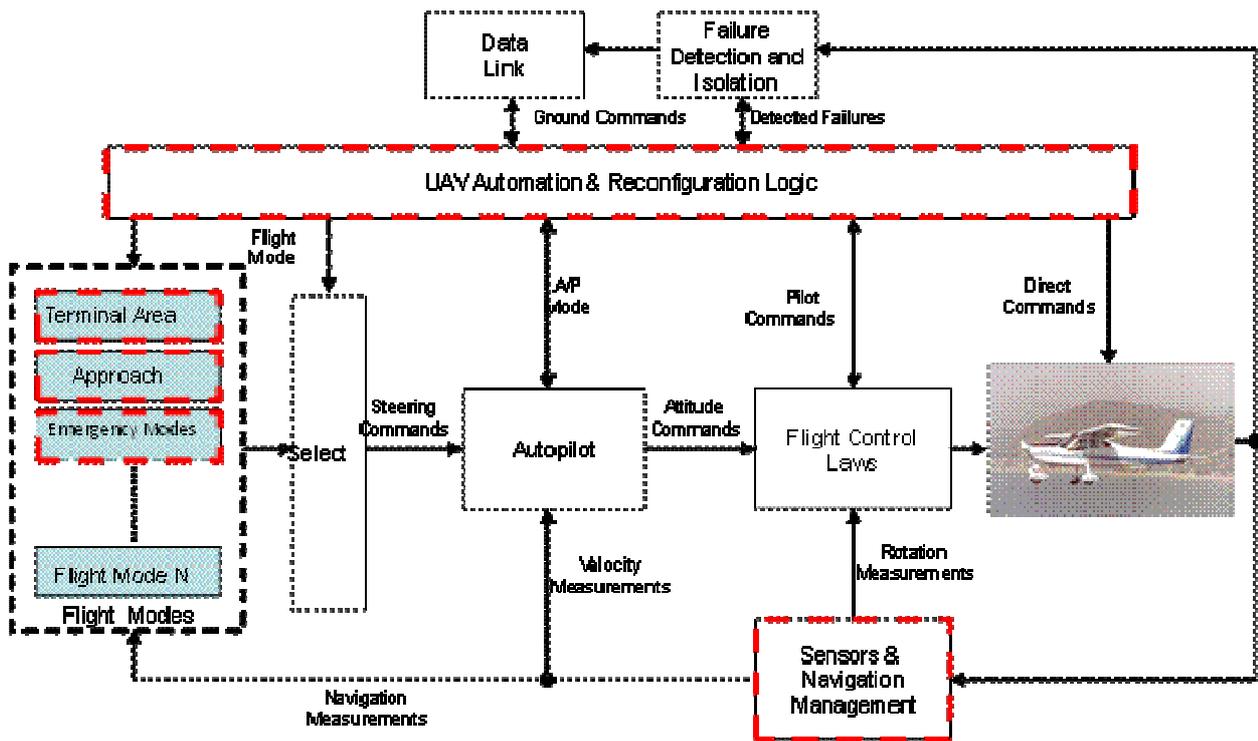


Figure 4-1 –GNC functional architecture of a UAV with advanced automatic features

In the above figure are outlined (with dotted red lines) the modules related to the author's Phd and described in the current thesis. In the following is reported a brief explanation of such modules.

- The *Sensor & Navigation System (SNS)* elaborates measurements coming from on-board sensors dedicated to GNC, such as inertial sensors, GPS, air data sensors, etc., to provide key feedback measures to all the above modules. Algorithms included in this module are linear and not linear filters, stochastic estimators (Kalman filters) and measurement acquisition and conversion procedures.
- the *UAV Automation & Reconfiguration Logic* that interprets remote pilot ground inputs enabling the appropriate path of commands for allowing the same manual or autopilot operation modes of architecture in Figure 4-1 or, in case of a fully automated flight, it execute an high level automatic sequence of operations through appropriate selection of a flight mode (among available ones) and reconfiguration of FCL and/or A/P, in case of detected failures;
- a *Flight Modes* module that actually includes several different trajectory generation and tracking algorithms developed for specific phases of flight (take-off [AR6], landing, mid-air flight, etc.) or failure conditions, appropriately selected by the automation logic. This module contains different sub-modules typically called with the aim of the particular phase that

manage and that are activated by the *UAV Automation & Reconfiguration Logic*. Each flight mode generates his output as references or for the Autopilot/Autothrottle [AR13] (typically altitude, vertical speed, lateral track, heading angle, TAS, etc.), or for the internal flight control law (typically attitude angles or angular velocities) or, finally, directly towards the actuators system (output of the whole GNC system expressed in terms of elevator, ailerons, rudder, throttle, flaps references). The particular kind of reference generated is communicated to the control modules by means of a specific configuration signal.

As said above, and with particular reference of the overall autonomous functionality for approach and landing, the work carried out during the doctorate have been addressed several modules of the architecture proposed in Figure 4-1, in particular the module Flight Modes (related to *Autonomous Mid Air Flight Execution*, *Autonomous Approach and Landing*, and *Emergency Modes*), the module of *UAV Automation & Reconfiguration Logic*, the module of *Sensor & Navigation System* (SNS). The *Mid Air Flight Execution* mode has been considered for managing the phase of terminal area operation preceding the proper phase of approach and landing.

4.2 UAV Automation and Reconfiguration Logic

The *Automation and Reconfiguration Logic* module was developed to accomplish the following two main tasks:

- to manage the sequencing of the different phases, segments (as depicted in Figure 4-2) and states involved in the autoland manoeuvre considering the proper transition events in case of nominal condition;
- to manage unpredicted events such as failure to critical and not-critical subsystems or lack of accuracy and integrity of the vehicle state estimation activating suitable emergency modes to make in safety the vehicle.

The proposed *Automation and Reconfiguration Logic* module, with reference to the *Autonomous Approach and Landing* functionality, in case of nominal condition (no failures to relevant subsystems occurred, correct and expected behavior of all the flight control line implemented and not presence of alarms due to inopportune altitudes above ground of the vehicle) evolves as showed in the logic diagram of Figure 4-3. The several states are defined in Table 4-1. The transition events among states are defined in Table 4-2.

In particular, as showed in Figure 4-2, the autonomous landing process is divided into four main phases, each corresponding to a specific state of the mission automation logic. These main phases are called Alignment, Approach, Flare and Pre-Touch Down.

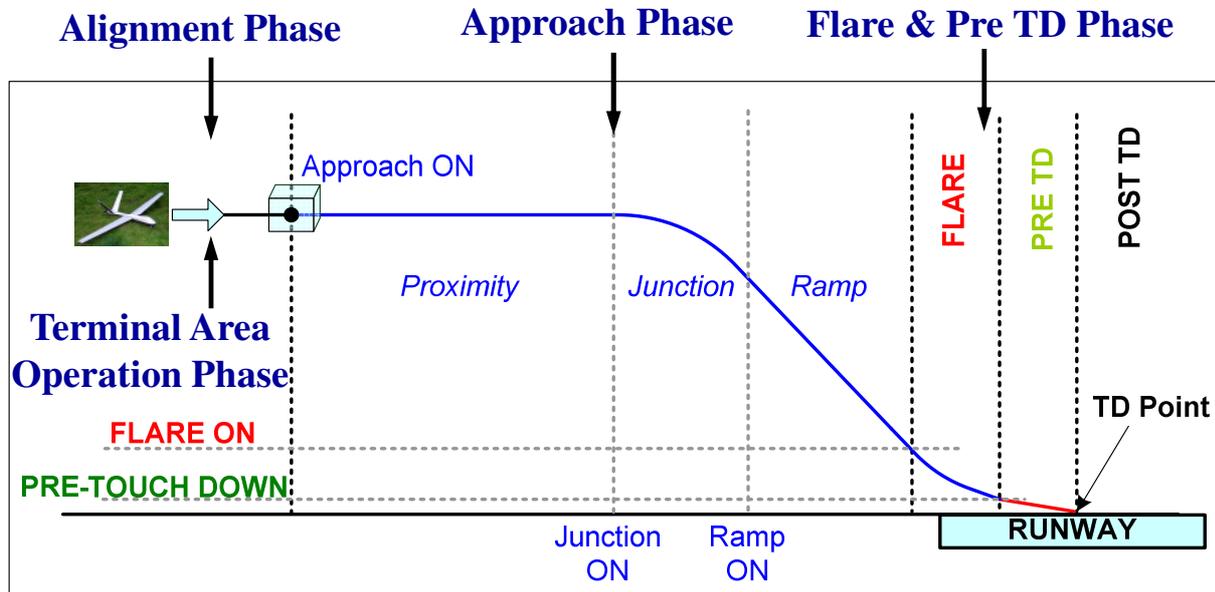


Figure 4-2 – Flight phases and segments involved in the autolanding manoeuvre

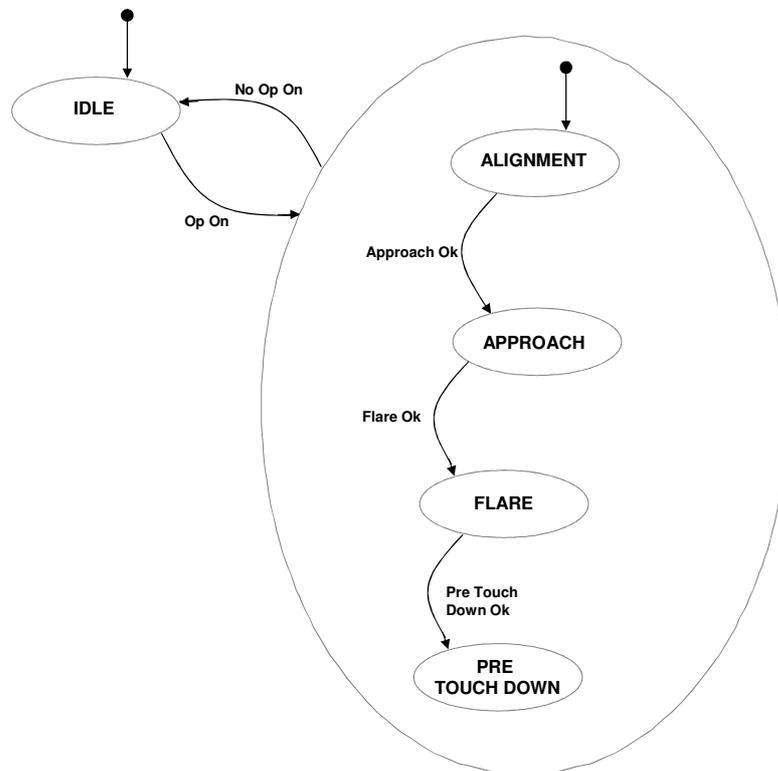


Figure 4-3 – Automation and Reconfiguration Logic state diagram in nominal condition

States	Description
Idle	Waiting state before autonomy activation
Alignment	State in which is managed the terminal operation phase (from top of descent up to the final waypoint aligned to runway)
Approach	State in which is managed the approach phase (proximity, junction and ramp segments as shown in Figure 4-2)
Flare	State in which is managed flare phase
Pre-Touch Down	State in which is managed final part of the flare phase just before touch down event

Table 4-1 - Automation and Reconfiguration Logic states definition

Events	Definitions
Op On	Autoland activation
Approach ok	Way point aligned to runway properly captured
Flare ok	Above Runway Level lower or equal than 15 m
Touch Down ok	Above Runway Level lower or equal than 2 m

Table 4-2 - Automation and Reconfiguration Logic transition events among states definition

The *Alignment phase* is intended to move the vehicle from its generic initial state (in terms of position and velocity), typically representing the top of descent waypoint in the aeronautical procedures, to a specified state, in which the vehicle is near the runway and aligned with the centreline. Also the final waypoint aligned to the runway is specified in terms of three-dimensional position and velocity vector of the vehicle.

In order to connect the initial position with the final waypoint, a 3D trajectory, constituted at the most by two circular arcs and one straight line, is generated on-line. This trajectory is sub-optimal, in the sense that it is the minimum length trajectory if the vehicle moves only in the horizontal plane but not necessarily it is the minimum length trajectory in the 3D space. The nominal trajectory is generated by solving on-line a constrained optimization problem, in which suitable constraints on the flight path angle and on the minimum turn radius are considered. The restriction on the minimum turn radius is derived from a proper constraint on the roll angle and from the inertial speed reference imposed to the vehicle. Since the autothrottle is designed in order to track the TAS instead of the inertial speed, in the Alignment phase the inertial speed considered for the trajectory generation is set to a proper safety value. Both autothrottle and autopilot systems are inside the functional module Autopilot showed in Figure 4-1. The detailed description of the trajectory generation and tracking algorithms for the Alignment phase is described in the paragraph 4.3. The Alignment subsystem provides the autopilot with suitable

references in terms of altitude, heading angle and lateral displacement and furnishes also the autothrottle with the TAS reference. These references are fixed with regard to a target point, located on the nominal 3D trajectory, opportunely further on with respect to the current aircraft position¹⁰. In this phase the flaps are not extended.

Once completed the Alignment phase, the vehicle height above the runway is suitable for the Approach phase and the aircraft is aligned with the centreline. This is assured by the proper setting of the final Alignment waypoint but, considering the presence of possible atmospheric disturbances, it is advisable to formulate the switch condition from the Alignment to the Approach in terms of a three-dimensional window to be crossed by the vehicle, with a track angle limited inside a specified range. If the vehicle, at the end of the Alignment, is inside the specified 3D window and his track angle is inside the fixed range, the Approach phase can start, otherwise the mission automation logic activates once again the alignment phase.

The Approach phase is divided into three segments: Proximity, Ramp and Junction. These segments, each corresponding to a specified state, are managed by low level phase automation logic. The Proximity segment aims to smoothly reduce the aircraft TAS from the value suitable for the previous Alignment phase to a proper level for the descent towards the runway. The nominal trajectory for this segment is a horizontal straight line, so the longitudinal (height) reference is constant. The lateral reference is the runway centreline and the flaps reference is the maximum extension. It must be noted that, even if the flap extension is commanded when the Proximity segment starts, it is really actuated only if the true airspeed (TAS) is lower than a specified value (with hysteretic threshold), in order to avoid structural damages. The TAS reference is a two steps reference: as the Proximity segment starts, a first speed reduction is commanded, then, as the flaps are extended at 10%, a second speed reduction is commanded, down to the proper value for the descent.

When the vehicle arrives at a specified longitudinal distance from the runway threshold, the low level phase automation logic activates the Junction segment. This segment aims to smoothly connect the horizontal straight line reference of the Proximity segment with the glide slope reference for the next Ramp segment. The longitudinal (height) reference is parabolic and tangential to the previous and the next trajectory references. The lateral reference is the runway centreline, the flap and TAS references are the same of the Proximity segment.

Once terminated the Junction segment, the Ramp segment is activated. This segment aims to move the vehicle down to the runway, following a GPS-based glide slope reference. The height

reference is therefore a ramp with negative and properly fixed slope. The lateral reference is the runway centreline, the TAS reference is the proper value for the descent and the flap reference is the full extension value. When the height above the runway reaches a specified threshold, the mission automation logic passes to the Flare phase.

The Flare phase aims to reduce the vertical speed of the vehicle to a value suitable for the touchdown and to increase the pitch attitude. During this phase an adaptive algorithm is adopted, called *vzlin*, for the generation of vertical speed and TAS references. It can be periodically executed with a pre-fixed rate for generating the nominal trajectory, into the longitudinal plane, in terms of $V_z(X)$ (vertical speed function of X-Runway axis) profile and $TAS(X)$ (true air speed function of X-Runway axis) profile. This algorithm will be detailed explained in the paragraph 4.4.

When the vehicle height above the runway crosses a specified threshold, the mission automation logic passes into the *Pre-Touch Down phase*, which aims to continue the flare manoeuvre guarantying the proper vehicle attitude at the ground contact. If there aren't critical failures, the nominal Pre-Touch Down mode is activated by the logic. In this case, the longitudinal and TAS references are still generated likewise the previous phase but activating an envelope protection subsystem that aims to avoid the ground contact of the aircraft tail or nose. The lateral reference is the runway centreline but with limited Autopilot inner roll reference, while the rudder control line is used to perform a decrab manoeuvre. The flap extension, finally, is held.

As the weight on wheels (WoW) output signal is on, the Post Touch Down phase is activated and all the references are direct link commands to elevator, ailerons, rudder, throttle and flaps.

In order to describe the activation logic of the recovery modes considered, we first need define some relevant concepts:

- Safe Altitude - It is intended as an altitude which allows the safe execution of any recovery manoeuvre.
- Ground Proximity Altitude - It is intended as a specified height above the runway which is considered too low with respect to the current flight phase.
- Middle Altitude – It is intended as an altitude which allows, in case of necessity, the execution of a normal pull up manoeuvre.
- Critical Altitude – It is intended as a height above the runway which requires, in non-nominal conditions, the execution of a fast pull up manoeuvre.

- No-Return Altitude – It is intended as a height above the runway below which is not possible to perform any pull up manoeuvre.
- Critical Failure – It is intended as a subsystem failure such that the Mission Automation Logic cannot activate any flight-law control algorithm, being possible only the direct link command on a specific control surface actuator.

In general: (Safe Altitude) > (Ground Proximity Altitude) > (Middle Altitude) > (Critical Altitude) > (No-Return Altitude).

Firstly note that, in the case in which a failure occurs when the vehicle height is below the No-Return Altitude, the vehicle landing cannot be disabled and the touch down will be executed in nominal mode or, if it is necessary, in a particular recovery mode, as will be described in the following. The recovery modes will be described in the next.

The Hard Emergency mode can be activated starting from any flight phase, except Pre-Touch Down. In the case a Critical Failure occurs and persists beyond a time threshold, despite the vehicle height and failure duration, the Mission Automation Logic activates this recovery mode. In this recovery mode, the system applies a direct link control on all the surfaces/throttle actuators involved with the critical failure setting specified values to retain. The flaps position is sampled and held and a critical alarm signal appears on the pilot cockpit in order to allow the return to manual control.

The Soft Emergency activation condition occurs when the positioning module cannot guarantee the required accuracy for the autonomous flight during the Alignment phase and the vehicle height is higher than the Safe Altitude. In this recovery mode, which aims to set the vehicle in safe condition without activating control modules involving positioning estimation, the control system activates the pitch control on the longitudinal channel and moves, with a specified linear rate, the aircraft pitch to a pre-defined value. Furthermore, on the lateral channel the actual heading is held and on the directional channel the yaw damper is activated. Assuming to not be in presence of an ADU (Air Data Unit) failure the autothrottle functionality is activated and, finally, flaps position is also sampled and held. If the failure condition ends, the Automation Logic returns in nominal conditions, starting newly with the Alignment phase.

The Altitude Recovery mode can be activated when the Automation Logic is in the Alignment phase and the vehicle height is lower than the Ground Proximity Altitude or when the Mission Automation Logic is in the Approach phase. This activation can be a consequence of an

appropriate failure or of excessive performance deterioration, so the independent activation conditions for this recovery mode are:

- it occurs, for a specified duration, a failure of a sensor which is considered relevant in order to allow the required performances;
- the current performances of the tracking system are deteriorated in such a way as they cannot be considered admissible with respect to the ones required in the nominal trajectory tracking.

Once activated, this recovery mode, which aims to pull up the vehicle, can act in three different manners (sub-modes), depending on the vehicle height and current subsystem failures. If the height is lower than the Critical Altitude, the altitude recovery sub-mode is Critical Altitude Recovery, while, if the height is higher than this level, the altitude recovery sub-mode is Normal Altitude Recovery in the case in which there isn't a GPS failure or is Safe Altitude Recovery if there is a GPS failure.

In the Critical Altitude Recovery sub-mode, which aims to implement a fast vehicle climb, the control system activates the pitch control on the longitudinal channel and provides a linear pitch reference up to a specified nose up value. On the lateral channel, the control system activates the roll angle control and provides a linear roll reference up to the wing levelled attitude. The yaw damper is activated, the throttle is set to 100% and the flaps position is sampled and held.

Once the vehicle returns over the Middle Altitude, the suitable sub-mode (Safe Altitude Recovery or Normal Altitude Recovery) is activated, depending on GPS availability, as previously described. In the Safe Altitude Recovery sub-mode, the control system activates the pitch control on the longitudinal channel and provides a linear nose up pitch reference, with rate lower than in the Critical Altitude Recovery, up to a specified value, lower than in the Critical Altitude Recovery. The controls on lateral and directional channels and on throttle are the same as in the Critical Altitude Recovery sub-mode.

In the Normal Altitude Recovery sub-mode, the control system activates the vertical speed control on the longitudinal channel, providing a linear vertical speed reference up to a specified positive value. On the lateral channel, the track control is activated and the current track is sampled and held. The yaw damper is active, the TAS reference is a linear reference up to a specified value and, finally, the flaps position is sampled and held.

When the vehicle returns over the Safe Altitude, in case of persistent GPS failure the Soft Safe recovery mode is activated otherwise a new Alignment phase starts.

The *Pre Touch Down Emergency* mode is activated when a Critical Failure occurs under the Pre-Touch Down Mode activation height, the Pre-Touch Down Mode is set to the Safe TD sub-mode. In this case, the control system activates on the longitudinal channel the direct link elevator control, providing it with a linear reference up to a fixed nose-up elevator value. On the lateral and directional channels, a direct link ailerons and rudder control is respectively activated, with fixed reference values. The throttle is commanded in direct link with a linear reference down to idle value and the flaps position is sampled and held.

4.3 Path Generation and Tracking for En-Route and Terminal Area Operations

As already mentioned above a crucial aspect of an autoland system design is to face the problems regarding the generation of the path during the Alignment phase. Hence, the path to be generated is limited to the one going from the present position to the fixed position. In view of the fact that the generation of the Alignment path has to be performed online by the Flight Control Computer, together with all the other navigation and control features, the problem has been further simplified introducing the following assumptions:

- a) the vehicle is a rigid body and has a constant mass and inertia;
- b) the Earth is flat and not rotating;
- c) it is assumed that the envelope and structural limitations of the vehicle (static constraints) can be taken into account by simple geometric constraints on the path, namely constraints on curvature radii and flight path angle.

Anyway for an exhaustive examination of the problem of trajectory planning with a more formal mathematical definition including insights into the constraints of the optimization problem for aerospace applications and giving an overview of models and equations used for representing mathematically these constraints see [B1].

In a NED coordinate frame (x_{NED} , y_{NED} , z_{NED}) fixed with respect to the earth, the path can be represented by the following equations:

$$\begin{bmatrix} \dot{x}_{NED} \\ \dot{y}_{NED} \\ \dot{z}_{NED} \\ \dot{\chi} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} V_0 \cos \chi \cos \gamma \\ V_0 \sin \chi \cos \gamma \\ -V_0 \sin \gamma \\ \eta \\ \mu \end{bmatrix}$$

Eq. 1 $\mathbf{X}(t_0) = [x_0 \quad y_0 \quad z_0 \quad \chi_0 \quad \gamma_0]^T = \mathbf{X}_0$
 $\mathbf{X}(t_f) = [x_f \quad y_f \quad z_f \quad \chi_f \quad \gamma_f]^T = \mathbf{X}_f$

where $\mathbf{X} = [x_{NED} \quad y_{NED} \quad z_{NED} \quad \psi \quad \gamma]^T$ is the vehicle state, x_{NED} , y_{NED} and z_{NED} are the north, the east and the down positions in NED respectively, γ is the flight path angle, ψ is the heading angle, η and μ are the control inputs.

Let introduce the following definitions of Horizontal and Vertical Curvatures.

$$R^H = \frac{V \cos \gamma}{\dot{\chi}}$$

Eq. 2 $R^V = \frac{V}{\dot{\gamma}}$

The first of Eq. 2 defines the radius of a suitable circle tangent to the horizontal trajectory and centered in a point of the xy plane, while the second expresses the same thing for vertical trajectory (see figure below).

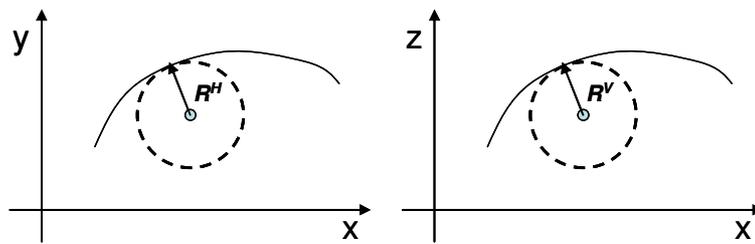


Figure 4-4 - Horizontal and Vertical Trajectory Curvatures

Moreover assuming that initial and final conditions of our trajectory generation problem do not specify different velocity modules V (see [B1] for a detailed discussion on this argument) and allowing that this (constant) variable is within its requested limits of the vehicle we can consider the further assumption

- d) the scalar velocity V is constant during the path

In this case the problem of the path generation can hence be formulated as an open-loop control problem where the control function has to be determined in such a way to guarantee the satisfaction of the constraints

$$\text{Eq. 3} \quad \begin{aligned} \frac{1}{R^H} &\equiv \frac{\eta}{V \cos \gamma} \leq \frac{1}{R_{\min}^H} \\ \frac{1}{R^V} &\equiv \frac{\mu}{V} \leq \frac{1}{R_{\min}^V} \\ \gamma_{\min} &\leq \gamma \leq \gamma_{\max} \end{aligned}$$

while minimizing the length of the path

$$\text{Eq. 4} \quad \min_{\eta(t), \mu(t)} J = \int_{t_0}^{t_f} dt$$

The assumption c) is justified as it can be easily shown (see [B1]) that, under certain hypotheses, constraints like load factors, angle of attack, bank angle and engine thrust limitations can be mapped into constraints of Eq. 3. This allows simplifying the trajectory generation problem, as geometric constraints only depend on input variables η, μ and on the state variable γ .

4.3.1 Proposed Algorithm for Optimal Path Generation without No-Fly Zones

Let consider a further assumption respect to that defined above:

- e) it is assumed that there are no obstacle to be avoided;

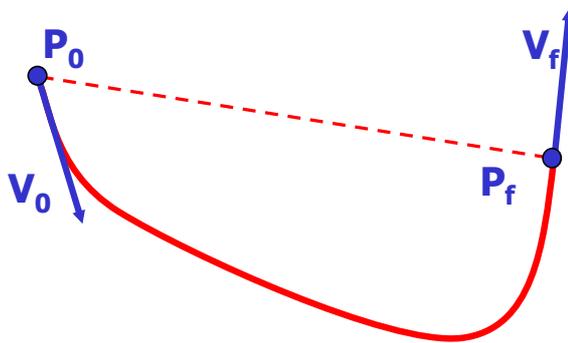


Figure 4-5 – Flight Planning without no-fly zones

Unfortunately the optimal control problem expressed by Eq. 1,Eq. 2,Eq. 3,Eq. 4 does not admit an analytical solution, and it is too demanding to be solved on-line using a numerical approach (it is still a non-convex problem). For this reason it was decided to resort to a suboptimal

approach which is viable from a computational point of view. The method chosen takes advantage of an empirical approach based on the optimality of the two-dimensional trajectory made up of straight lines and arcs, according to Dubins theory [B3]. In fact it has been shown in [B3] that, in case of a two-dimensional trajectory, the shortest path is composed by the union of an arc of circumference, a segment and again an arc of circumference. Hence, the 2D path generation problem is obtained by letting $z_f = z_0 = 0$ and $\gamma_f = \gamma_0 = 0$ (trajectory contained in x-y plane) and both the arcs of circumference have the minimum radius $R_H \min$ (see Figure 4-6).

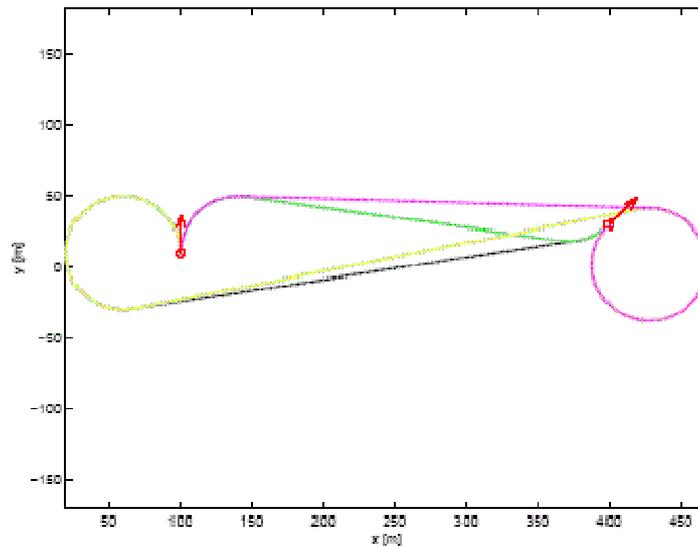


Figure 4-6 - Dubins circle algorithm. The optimal path is the green one

For what concerns the construction of 3D trajectory, the flight path angle is kept constant during the flight, and it is computed as follows:

$$\text{Eq. 5} \quad \gamma_m = \arctan\left(\frac{z_f - z_0}{d_m}\right)$$

where d_m is the length of the computed horizontal trajectory. In case flight path angle γ_m is not compliant with flight path angle constraints, a simple method based on an iterative procedure is used to enlarge the horizontal curvature radius until the flight path angle is between the maximum and minimum value of Eq. 3. Further details can be found in [AR1] and [B1].

Clearly, the obtained reference γ_m value may be different from both the initial and final flight path angles γ_f and γ_0 ; this discontinuity in the flight path angle is actually managed by the tracking algorithm, that will be described in §4.3.3.

4.3.2 Proposed Algorithm for Optimal Path Generation considering No-Fly Zones

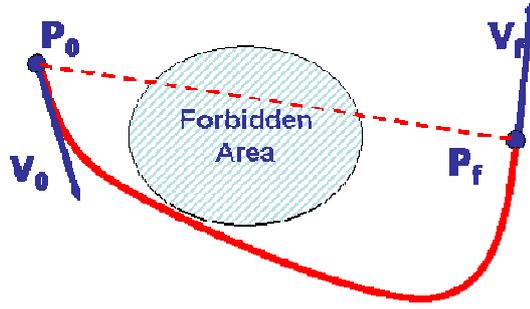


Figure 4-7– Problem Formulation in case of No-Fly Zones

The free flight planning algorithm, described in the previous paragraph, provides an optimal 3D trajectory without considering any constraints on forbidden zones or fixed obstacles. This trajectory is used as starting point for the trajectory generation in the presence of such constraints. So the assumption e) is not more considered and the optimization problem now is

$$\begin{aligned}
 \min_{\eta(t), \mu(t)} J &= \int_{t_0}^{t_f} dt \\
 \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\chi} \\ \dot{\gamma} \end{bmatrix} &= \begin{bmatrix} V_0 \cos \chi \cos \gamma \\ V_0 \sin \chi \cos \gamma \\ -V_0 \sin \gamma \\ \eta \\ \mu \end{bmatrix} \\
 \mathbf{X}(t_0) &= [x_0 \quad y_0 \quad z_0 \quad \chi_0 \quad \gamma_0]^T = \mathbf{X}_0 \\
 \mathbf{X}(t_f) &= [x_f \quad y_f \quad z_f \quad \chi_f \quad \gamma_f]^T = \mathbf{X}_f \\
 \text{Eq. 6} \quad \frac{1}{R^H} &\equiv \frac{\eta}{V \cos \gamma} \leq \frac{1}{R_{\min}^H} \\
 \frac{1}{R^V} &\equiv \frac{\mu}{V} \leq \frac{1}{R_{\min}^V} \\
 \gamma_{\min} &\leq \gamma \leq \gamma_{\max} \\
 r_j^2 - (\Delta x_j)^2 - (\Delta y_j)^2 &\leq 0, \quad j = 1, 2, \dots, M \\
 \Delta x_j &= x - x_j, \Delta y_j = y - y_j
 \end{aligned}$$

where each no-fly zone j is defined as a circular region with a time invariant radius r_j

This problem is solved by a local optimization procedure aimed to select the optimum trajectory (the shortest one) between two waypoints in the presence of no-fly zones to avoid. The optimization procedure is based on the following considerations:

- The trajectory generated by the ‘free flight’ 2-D path planning is the shortest path between two waypoints with given limitations on horizontal trajectory radius (see §4.3.1).
- If such free flight trajectory violates the constraints of Eq. 6 (i.e. it intercepts one or more no-fly zones), a revised (suboptimal) trajectory can be found by minimizing the length difference from the free flight path, while accounting for constraints of Eq. 6.
- A sub-optimal path shall be found considering the trajectories composed by a sequence of arcs and straight lines that are tangent to one or more forbidden zones suitably chosen among all the specified no-fly zones.

Starting from the above considerations, an optimization procedure is hereinafter proposed.

After the 2D free flight path has been generated between a starting waypoint WP_A and a target waypoint WP_B , the compliance to constraints of Eq. 6 of the trajectory straight segment, that is, the line connecting the intermediate points A and B (see Figure 4-8), is checked. The subset Π' of forbidden zones actually crossed by the trajectory (i.e. the red ones in the figure) is then considered, while the remaining ones (green zones) are discarded.

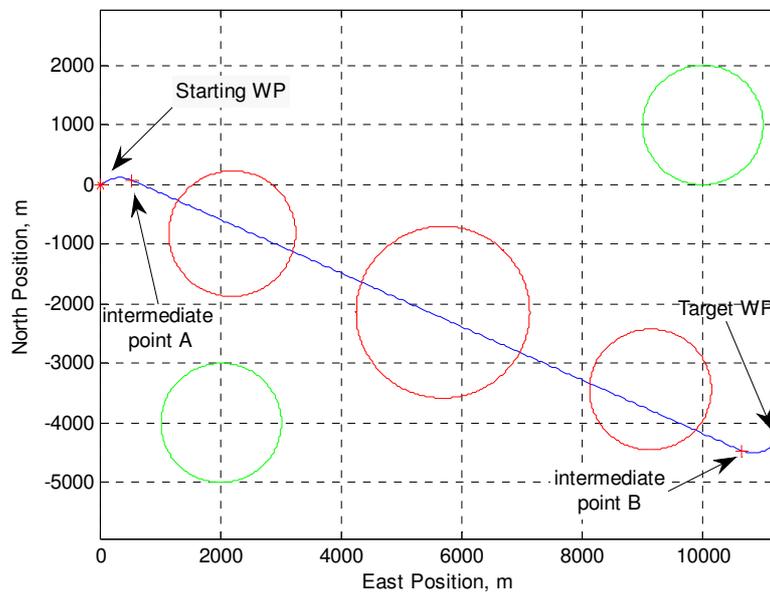


Figure 4-8– Free Flight trajectory Generation between WP_A and WP_B

Generally speaking, the proposed algorithm efficiently finds a trajectory that: a) it is tangent to one or more forbidden zones actually crossed by the free flight path (the subset Π'), b) it does not cross any other no-fly zones of subset Π' , c) it has as low as possible length increment with respect to the free flight path.

To this end, in order to avoid considering all possible combinations of trajectories tangent to the zones of subset $\Pi' = (D'_1, \dots, D'_n)$, this subset is sorted in ascending order considering the distance between the centre of each zone and the target waypoint WP_B and the following procedure is executed (see from Figure 4-9 to Figure 4-13):

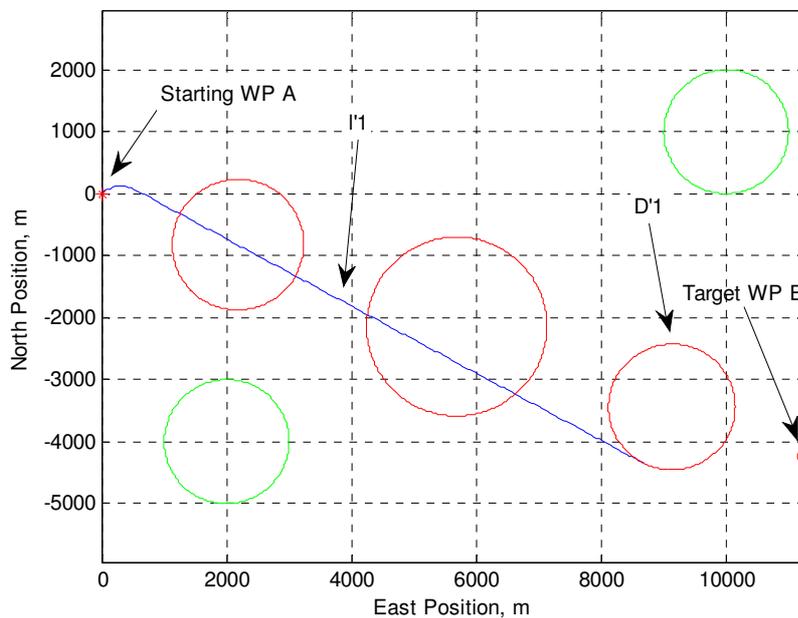


Figure 4-9– Computation of the line tangent to the zone nearest to the target WP-step1

- 1) Starting from the point A, the straight line tangent to the first element D'_1 of Π' is generated. It is important to remark that even if there are two possible tangents to this circular zone, the algorithm selects the one with the minimum displacement from the unconstrained trajectory in terms of track angle deviation. This trajectory has the minimum distance from the optimal unconstrained trajectory in the tangency point. In other words, this is the trajectory having the minimum increment of total length w.r.t. the optimal unconstrained trajectory, while satisfying the constraint of avoiding the considered no-fly zone. The resulting straight line l'_1 can be feasible or unfeasible depending on whether it crosses any other no-fly zones or not.

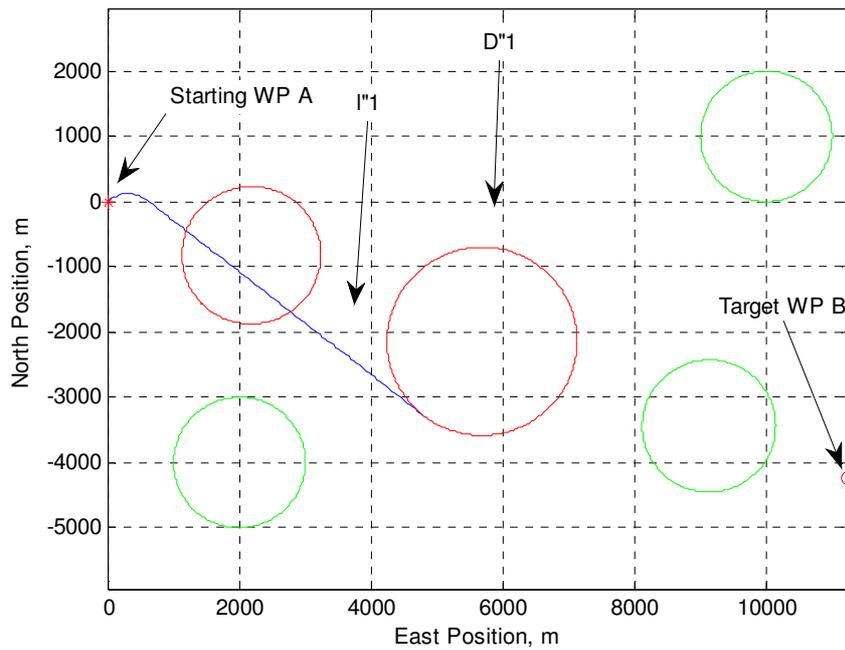


Figure 4-10– Computation of the line tangent to the zone nearest to the target WP-step2

- 2) The feasibility of l_1' is checked. If it is feasible, D_1' is considered the optimal no-fly zone that allow complying with constraints of Eq. 6 while minimizing path length w.r.t. free flight trajectory and the step 4 below is executed.
- 3) Otherwise, a new subset $\Pi'' = (D_1'', \dots, D_m'')$ of zones crossed by l_1' is generated and sorted with the same criteria of Π' . Then, step 1 is repeated considering D_1'' , instead of D_1' .
- 4) After selecting the optimal no-fly zone using above steps, the straight line starting from A and tangent to this zone is considered as a segment of the global path. The procedure described in the above steps is then repeated, providing that the initial waypoint WP_A is replaced by the point A_1 , which is the tangency point of this trajectory segment with the considered optimal no-fly zone (see Figure 4-12).

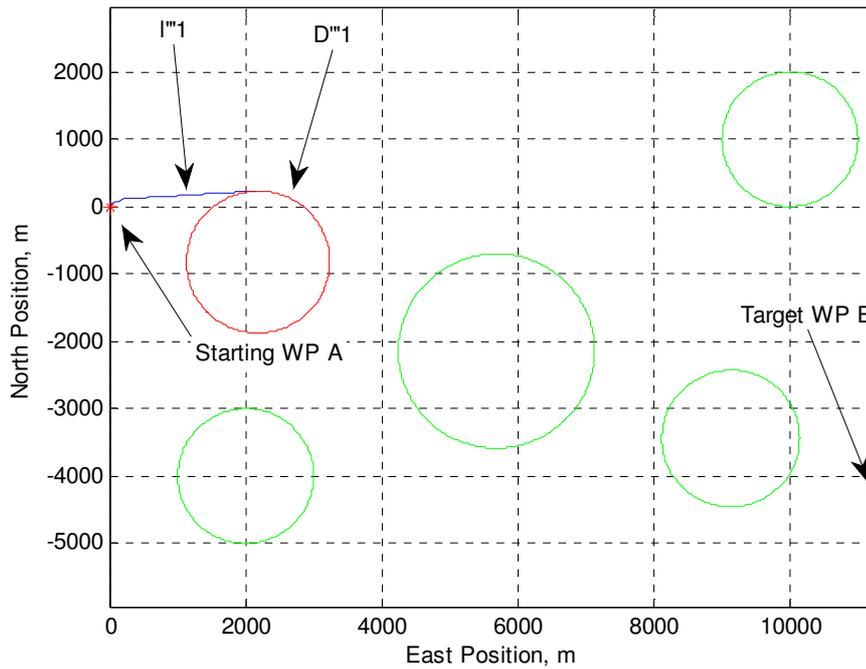


Figure 4-11– Computation of the line tangent to the zone nearest to the target WP-step3

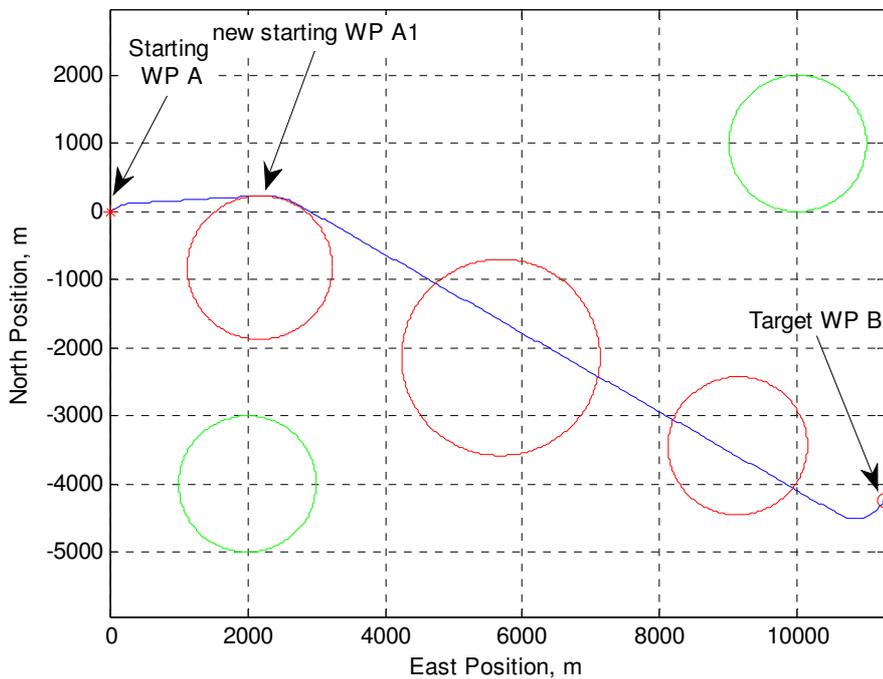


Figure 4-12– Generation of the Unconstrained Trajectory from the new starting WP

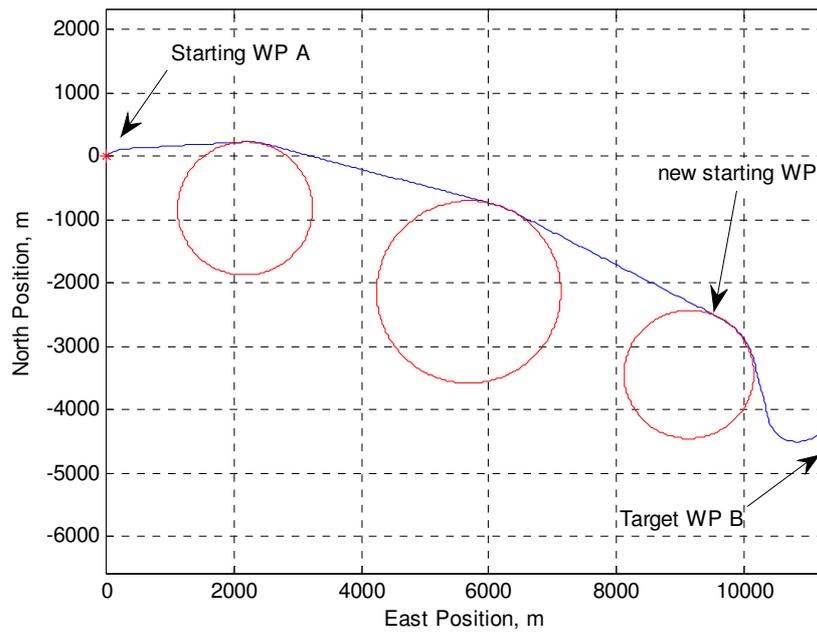


Figure 4-13– Generation of the final trajectory

The final result for the considered example is shown in Figure 4-13. In the flow chart of Figure 4-14, the optimization algorithm is graphically represented.

It is worth noting that, each iteration of steps 1 to 3, the candidate optimal zone is closer to the starting WP than the one obtained at the previous iteration. Obviously, in the best case only one iteration is needed, while in the worst case, the number of iterations is equal to the total number M of no-fly zones known during flight. On the other hand, each time a new starting point is chosen as described above in step 4, the remaining no-fly zones to be avoided are reduced of one element, at least. Thus, every time the procedure is repeated, the maximum number of iterations is equal to the number of no-fly zones between the current point and the target WP. This number cannot be greater than $M-i$ where $i \geq 0$ are the forbidden zones already avoided and M is the total number of zones.

Since the path planning algorithm stops when the unconstrained trajectory reaches the target WP without crossing any forbidden zones, the number of trajectory computations needed for the generation of a feasible trajectory is at most:

$$\text{Eq. 7} \quad I_{wc} = \sum_{i=0}^M (M+1-i) = (M+1)^2 - \sum_{i=1}^M i = \frac{(M+1)(M+2)}{2}$$

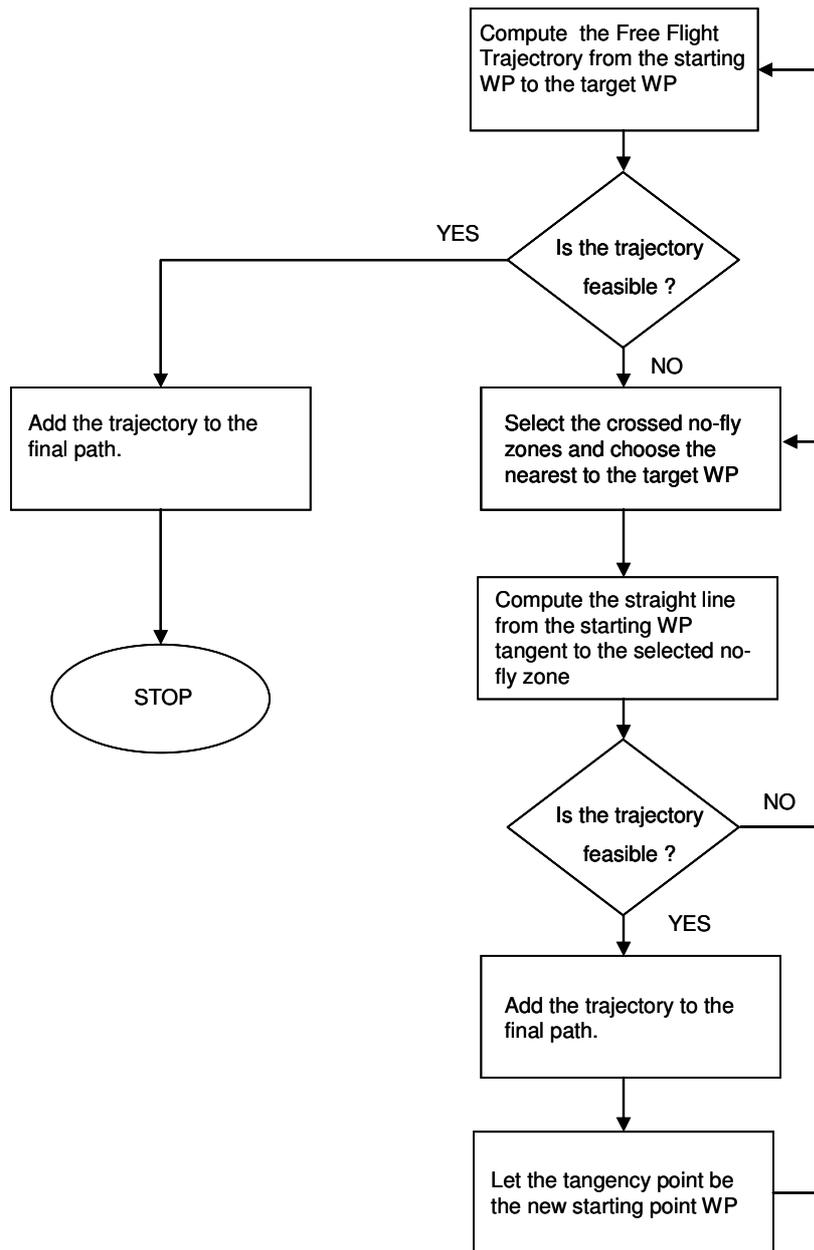


Figure 4-14– Flow Chart of Path Planner Algorithm

From this equation, it is easy to recognize that the computational complexity of the above described approach is quadratic in the worst case. In other terms, NP-complexity of the problem formulated in [B34] has been reduced to polynomial complexity at the expense of solution sub-optimality. Furthermore the algorithm can be implemented in real time, as the computation time is deterministic once that the maximum number of no-fly zones has been fixed. In this way, trajectory generation can be repeated anytime during flight, thus accounting for both variations of vehicle constraints or for changes of position/size of no-fly zones as well as for adding new no-fly zones.

For a more detailed treatment on the optimality aspects of the algorithm versus his computational complexity and on algorithm limitation and possible improvements see [AR2][AR10].

4.3.3 Trajectory tracking

In this paragraph, an algorithm for UAV trajectory tracking is described. The algorithm is used for tracking of the reference trajectory generated by using the method described in the previous paragraphs. Furthermore many tracking algorithms exist in literature that could be used for these purposes; in this thesis, an approach is described that resembles the *line-of-sight guidance* [B19][B34].

The algorithm is based only on the kinematic equations of motion of Eq. 1.

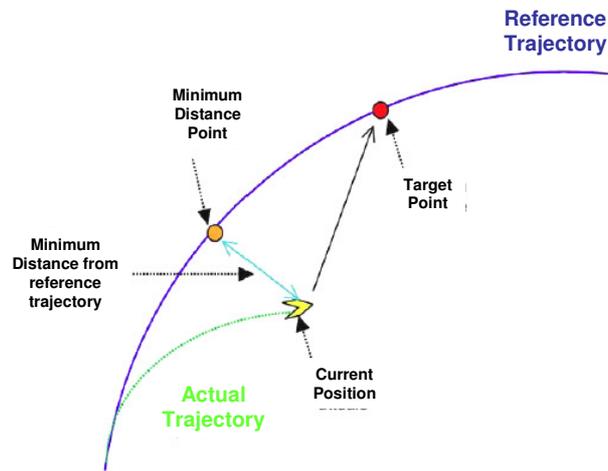


Figure 4-15- Line of Sight Guidance Concept

Hereafter χ and γ will play the role of control inputs and s is the curved abscissa

$$\text{Eq. 8} \quad s(t) = \int_{t_0}^t v d\tau,$$

where τ is simply an integration variable.

It can be shown that the control law expressed by Eq. 9

$$\begin{aligned}
 \cos \psi &:= \frac{\tilde{e}_x}{\sqrt{\tilde{e}_x^2 + \tilde{e}_y^2}}; \\
 \sin \psi &:= \frac{\tilde{e}_y}{\sqrt{\tilde{e}_x^2 + \tilde{e}_y^2}}; \\
 \cos \gamma &:= \frac{\sqrt{\tilde{e}_x^2 + \tilde{e}_y^2}}{\tilde{e}} \\
 \sin \gamma &:= \frac{\tilde{e}_z}{\tilde{e}}
 \end{aligned}
 \tag{Eq. 9}$$

where the following definition last

$$\begin{aligned}
 \tilde{e}_x &:= x_1(s) \Big|_{s=1/K_p} - x = e_x + \frac{1}{K_p} \cos \gamma_r^* \cos \psi_r^*; \\
 \tilde{e}_y &:= y_1(s) \Big|_{s=1/K_p} - y = e_y + \frac{1}{K_p} \cos \gamma_r^* \sin \psi_r^*; \\
 \tilde{e}_z &:= z_1(s) \Big|_{s=1/K_p} - z = e_z + \frac{1}{K_p} \sin \gamma_r^*; \\
 \tilde{e} &:= \sqrt{\tilde{e}_x^2 + \tilde{e}_y^2 + \tilde{e}_z^2} = \sqrt{e^2 + \frac{1}{K_p^2}}
 \end{aligned}
 \tag{Eq. 10}$$

$$\begin{aligned}
 e_x &:= x_r(s^*) - x; \\
 e_y &:= y_r(s^*) - y; \\
 e_z &:= z_r(s^*) - z; \\
 e &:= \sqrt{e_x^2 + e_y^2 + e_z^2}; \\
 \gamma_r^* &:= \gamma_r(s^*); \\
 \psi_r^* &:= \psi_r(s^*);
 \end{aligned}
 \tag{Eq. 11}$$

guarantees asymptotic convergence to the reference trajectory [AR1], i.e. $e(t) \rightarrow 0$ asymptotically. The value of K_p is related to the *look-ahead distance* of the classical line-of-sight algorithms.

It is worth noting that control law of Eq. 9 is ideal and takes into account neither curvature nor flight path angle allowable ranges, i.e. the constraints defined in Eq. 3. This limitation is then overcome by using a rate limiter on χ and γ , at the expense of formal proof of asymptotic convergence to the reference path.

4.4 Path Generation for Flare phase

In this paragraph is described the adaptive algorithm adopted during the flare phase, called *vzlin*, used for the generation of vertical speed and TAS references. It can be periodically executed with a pre-fixed rate for generating the nominal trajectory, into the longitudinal plane, in terms of $V_Z(X)$ (vertical speed function of x-Runway axis) profile and $TAS(X)$ (true air speed function of X-Runway axis) profile. For each single iteration, the algorithm has two objectives. The first is to achieve a profile of desired vertical speed and ground speed proportional with the longitudinal position along the X- Runway axis according with the following formulae:

$$\text{Eq. 12} \quad \begin{aligned} V_Z(X) &= a(X - X_F) + b \\ V_X(X) &= c(X - X_F) + d \end{aligned}$$

The second objective is to have, at the touch down event, a desired kinematical state in terms of velocity vector and position. With regard to the starting point of the Flare phase, let be:

- H_0 the initial position along Z-Runway axis;
- X_0 the initial position along X-Runway axis;
- V_{X0} the initial inertial velocity profile along X-Runway axis.

With regard to the desired state at the touch down, let be:

- H_F the desired final position along Z-Runway axis;
- X_F the desired final position along X-Runway axis;
- V_{XF} the desired final inertial velocity profile along X-Runway axis;
- V_{ZF} the desired final inertial velocity profile along Z-Runway axis.

The problem solved by the algorithm *vzlin* is the calculation of the coefficients a , b , c and d of Eq. 12 such that the following constraints are satisfied

$$\text{Eq. 13} \quad \begin{aligned} V_Z(X_F) &= V_{ZF} \\ \int_{X_0}^{X_F} V_Z(X) dX &= H_F - H_0 \\ V_X(X_F) &= \dot{X}(X_F) = V_{XF} \\ V_X(X_0) &= \dot{X}(X_0) = V_{X0} \end{aligned}$$

The solution is expressed in the following equations

$$\text{Eq. 14} \quad a = \frac{V_{ZF}(X_0 - X_F) \ln\left(\frac{V_{XF}}{V_{X0}}\right) - (H_F - H_0)(V_{X0} - V_{XF})}{(X_0 - X_F)^2 \left[1 + \frac{V_{XF}}{(V_{X0} - V_{XF})} \ln\left(\frac{V_{XF}}{V_{X0}}\right) \right]}$$

$$b = V_{ZF}$$

$$\text{Eq. 15} \quad c = \frac{V_{X0} - V_{XF}}{X_0 - X_F}$$

$$d = V_{XF}$$

The final aim is to generate the vertical speed and TAS references for the Autopilot. For what concerns the TAS reference generation, the strategy chosen foresees to generate a predefined profile of desired TAS_D , as in the following expression:

$$\text{Eq. 16} \quad TAS_D(X) = c_1(X - X_F) + d_1$$

with

$$\text{Eq. 17} \quad c_1 = \frac{TAS_0 - TAS_F}{X_0 - X_F}$$

$$d_1 = TAS_F$$

where TAS_F is the desired TAS at touch down (notice that this value has to be well evaluated because it directly influences the pitch angle at touch down), TAS_0 is the TAS reference when the flare phase starts (e.g. the TAS reference during the ramp segment of the approach phase).

Under the following assumptions

$$\text{Eq. 18} \quad \text{Constant } \vec{V}_{WIND} \text{ and } V_x \gg V_z$$

the TAS profile shown in Eq. 16 implies an inertial velocity along X-Runway axis profile proportional to X, confirming the second objective of the algorithm. The coefficients c and d of Eq. 15 are easily deducible from Eq. 15, Eq. 16, Eq. 17 and from the relation

$$\text{Eq. 19} \quad \vec{V}_{INER} = \vec{TAS} + \vec{V}_{WIND}$$

Where $\vec{V}_{INER}, \vec{TAS}, \vec{V}_{WIND}$ are respectively the inertial velocity vector, the true air speed vector and the wind velocity vector in the NED reference frame. The vertical speed profile defined by Eq. 12 and Eq. 14 can be directly used as vertical speed reference to the Autopilot.

In spite of the assumptions expressed in Eq. 18 the effectiveness of the algorithm is assured by the capability to re-generate the longitudinal trajectory with a proper rate. Anyway, as it will be shown in chapter 6, even just two iterations of the algorithm at the start of the flare and of the pre-touch down phase can be very convenient with respect to a strategy using a pre-fixed flare trajectory [AR14][AR15].

In brief, during the flare phase the TAS reference is a ramp, function of X, between the constant TAS reference imposed during the ramp segment of the approach phase and the final desired TAS value at touch down; the longitudinal reference is deducted by means of a single or multiple iteration of the *vzlin* algorithm. In chapter 6 the results of two possible strategies will be shown: one based on a step-by-step iteration of *vzlin* generating a TAS and vertical speed reference, and the other based on a double iteration of *vzlin*, generating a TAS and an altitude reference, at the begin of the Flare and Pre-Touch Down phases. The altitude reference can be calculated by integration of the vertical speed profile:

$$\text{Eq. 20} \quad H(X) = H_0 + \frac{a}{c}(X - X_0) + \frac{(bc - ad)}{c^2} \ln\left(\frac{c(X - X_F) + d}{c(X_0 - X_F) + d}\right)$$

with a, b, c and d defined in Eq. 14 and Eq. 15.

4.5 Sensor Fusion Algorithms

4.5.1 Positioning Estimation

As already mentioned in the preamble of this chapter the proposed algorithm for positioning estimation is based on complementary filtering inertial and satellite navigation measures [AR3].

The complementary filter proposed, described in the next, is a method for integrating position and speed measures (coming from GPS) with accelerations, attitude and orientation measures (coming from an AHRS - Attitude and Heading Reference Systems). In this way it is not necessary the use of a sophisticated and expensive INS (Inertial Navigation System) with its algorithms for estimating, independently from the GPS, position and speed of the vehicle.

This filter aims to determine in the best way the aircraft position and speed, in the NEU reference system, by using both the raw measures from the inertial sensors and the measures supplied by the GPS.

The general concept of the complementary filter is the integration of acceleration measures supplied by the AHRS, in order to obtain position and speed measures affected by lower noise and with a larger band in comparison with GPS measures. However, even if the AHRS measures are little noisy, they are affected from remarkable bias errors, so speed and position calculated only by integration of the accelerations can quickly diverge from the real values. In order to limit the effects due to the bias, therefore, it can be thought to integrate the accelerations and to process them through a high-pass filter, obtaining the medium-high frequency component of the considered signals. The low frequency components can be obtained by a filtering stage of the GPS measures through a low-pass filter. The final estimate of position and speed is equal to the sum of the two components above mentioned.

The resulting architecture of the complementary filter we developed is, therefore, the one shown in the schematic representation of Figure 4-16.

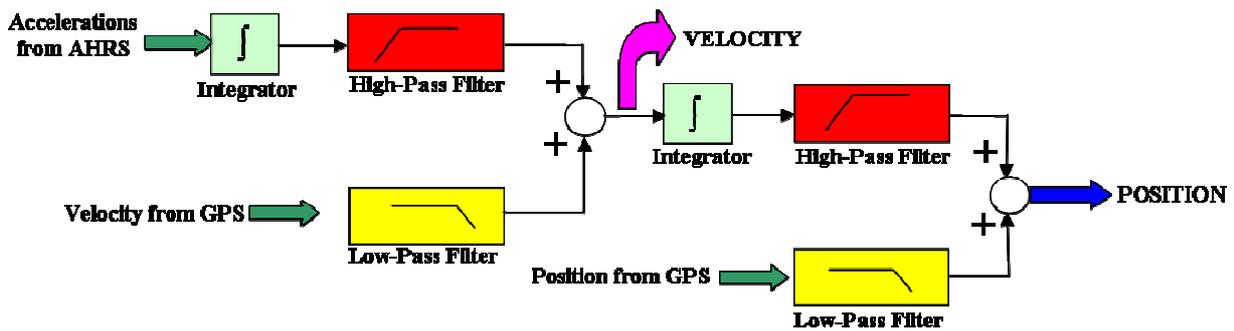


Figure 4-16 - Conceptual representation of the complementary filter

It is important to emphasize that, in both velocity and position measures estimation, the high-pass filter applied to AHRS measures and the low-pass filter applied to GPS measures must be “complementary”, in the sense that the sum of the transfer functions of the two filters must be equal to one. This is the reason why the navigation measures integration method here proposed is defined “complementary filter”.

The specific cut-off frequencies used in the filters shown in Figure 4-16 have to be chosen to reach the following two contrasting aims: minimizing the noise power due to the GPS and avoiding the error arising from the integration of the AHRS accelerometers bias.

The method above described applies in normal no-failure conditions, where INS and GPS sensors correctly work. However, also in the case of GPS failure it is necessary obtain

estimation, even if not optimal, of vehicle navigation data. The strategy adopted in this situation is described in the next. In the case of GPS failure, the basic idea is to replace the GPS measures with the ones provided by a sensor characterized by the same characteristics, even if with lower precisions: in this case ADS, with an appropriate offset adjustment, represents a good solution. Pressure altitude (PALT) is used regarding the vertical position measure, while for the vertical speed is used the PALT RATE measure.

Regarding position and velocity in the horizontal plane, instead, ADS does not directly supply such measures, but they can be opportunely obtained. In particular, for the velocity in the horizontal plane estimation the procedure described in the next is used. As long as GPS correctly works, it is continuously performed wind estimation, based on the relation already expressed in the previous Eq. 19 and isolating the wind velocity vector at the first member of the equation.

When a GPS failure is detected, this wind estimation is frozen $\vec{V}_{WIND-FR}$ and constant wind is considered, so from the TAS estimation derived from ADS measures it is possible to approximately estimate the inertial speed velocity vector from

$$\text{Eq. 21} \quad \vec{V}_{INER} = \vec{TAS} + \vec{V}_{WIND-FR}$$

In this way it is possible, in case of GPS failure, approximately estimate the inertial speed components in the horizontal plan. Such components are used in place of GPS velocity measures as inputs in the complementary filter, which supplies in output velocity and position estimation.

This idea correctly works when the aircraft is following a trajectory in a mid-air flight mission. In the case of GPS failure during landing, to obtain a better estimation of the measures of interest, it is also possible to use laser altimeter measures. During landing phase, therefore, PALT and PALT RATE measures from ADS are replaced by altitude and vertical speed estimations derived from laser altimeter measures. In this case too, of course, the cut-off filtering frequencies applied on the laser are specifically optimized.

For what concerns the use of the navigation measures integration method here proposed in the future Global Navigation Satellite Systems (GNSS) framework, furthermore, it is very relevant to emphasize that the described sensor fusion algorithm can be used in this framework too, by simply replacing the GPS receiver with one able to receive EGNOS (European Geostationary Navigation Overlay Service) and GALILEO signals.

Moreover, in the future GNSS framework it will be possible to improve the proposed algorithm, by including new safety features. In particular, the basic idea consists in using the EGNOS performance information (in terms of accuracy, integrity, continuity and availability) to improve the sensor fusion algorithm efficiency and to add an integrated diagnostic function for detecting system failures. Based on this integrated diagnostic function, it will be possible to switch, in case of failure, in an appropriate degraded navigation mode.

This will constitute a very relevant enhancement of the proposed navigation system, considering that integrity issues, important in general for many applications, are particularly critical in the aviation field, where vehicles can travel at high speed and can quickly deviate from the flight path.

4.5.2 Above Runway Level Estimation

As already noted in §3.3 for satisfying the RNP criteria for autoland system is essential also achieving an optimal estimation of the “Above Ground Level” and the “Above Runway Level” of the vehicle. This kind of estimation necessitates the use of sensors of different nature and proper sensor fusion logics to enable efficient combination of their measures.

With this aim a sensor fusion algorithm has been developed for an optimal estimation of altimetry of an aircraft during low altitude flight, by the combined use of Laser Altimeter, GPS and with the innovative idea to use also digital elevation models (DEMs).

In this case the sensor fusion algorithm was designed using a Kalman filter and combining the measures of altitude and vertical speed performed by GPS, range as measured by the laser altimeter, and terrain elevation provided by the DEM, in order to best estimate the Above Runway Level of the vehicle and with the aim to identify the accuracy that the DEM should have in order to significantly improve the estimate obtained without it.

The algorithm was validated by means of numeric simulation and in-flight data collection. The validation phase required the development of a model of orographic profile, the use of a developed DEM error model, and the appropriate modification to the model of the laser altimeter previously used in TECVOL project for taking into account the established true terrain elevation. The models developed will be showed in the paragraph 5.1.1 and the validation results in the paragraph 6.2.2.

4.5.2.1 Overview on discrete Kalman filtering implementation

For an overview on the general Kalman filtering theory refers to [B52]. In the following is reported the proposed implementation of a Kalman filter for observing a discrete non-linear stochastic process.

The state and output equations are

$$\text{Eq. 22} \quad \begin{cases} \underline{x}_k = f(\underline{x}_{k-1}, \underline{u}_k) + \underline{w}_{k-1} \\ \underline{y}_k = g(\underline{x}_k) + \underline{v}_k \end{cases}$$

With \underline{x}_k state vector at the generic step k (k), \underline{u}_k input vector (k), f and g nonlinear functions representing the system, \underline{y}_k the output vector (k), \underline{w}_k is the noise vector associated model/input error, \underline{v}_k is the noise vector associated with the measurement noise.

considering $\hat{\underline{x}}_k^-$ as an a-priori estimation, based on the knowledge of the current input \underline{u}_k and the state vector in $k-1$ we can write

$$\text{Eq. 23} \quad \hat{\underline{x}}_k^- = f(\hat{\underline{x}}_{k-1}, \underline{u}_k)$$

Let be the following definition

- $\hat{\underline{x}}_k$ a-posteriori state estimation using the knowledge of \underline{y}_k ;
- $\underline{e}_k^- = \underline{x}_k - \hat{\underline{x}}_k^-$ a-priori estimation error;
- $\underline{e}_k = \underline{x}_k - \hat{\underline{x}}_k$ a-posteriori estimation error;
- $P_k^- = E[\underline{e}_k^- (\underline{e}_k^-)^T]$ a-priori error covariance matrix;
- $P_k = E[\underline{e}_k (\underline{e}_k)^T]$ a-posteriori error covariance matrix;
- R_k covariance matrix associated with the measurement noise
- Q_k covariance matrix associated with the dynamic disturbance or model/input noise
- $A_k = \frac{\partial f(\hat{\underline{x}}_{k-1}, \underline{u}_k)}{\partial \underline{x}}$ linearized system matrix
- $C_k = \frac{\partial g(\hat{\underline{x}}_k)}{\partial \underline{x}}$ linearized output matrix

the a-posteriori state estimation can be calculated by means of

$$\text{Eq. 24} \quad \hat{\underline{x}}_k = \hat{\underline{x}}_k^- + K_k (\underline{y}_k - \underline{g}(\hat{\underline{x}}_k^-))$$

where K_k is the Kalman gain matrix and $g(\hat{x}_k^-)$ is the a-priori estimated measures. K_k has to be calculated step by step for minimizing P_k and the used method to do this is

$$\text{Eq. 25} \quad K_k = P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1}$$

where for estimating the error covariance matrixes the following equations hold

$$\text{Eq. 26} \quad P_k^- = A_k P_{k-1} A_k^T + Q_k$$

$$\text{Eq. 27} \quad P_k = (I - K_k C_k) P_k^-$$

The characteristic equation of the filter can be divided in two groups, time update equation and measurement update equation. The first ones are used for the a-priori estimations and the second ones use the a-priori estimations and the measurements for the a-posteriori estimations. This kind of subdivision can be interpreted as an algorithm predictor-corrector typically used for resolving numeric problems.

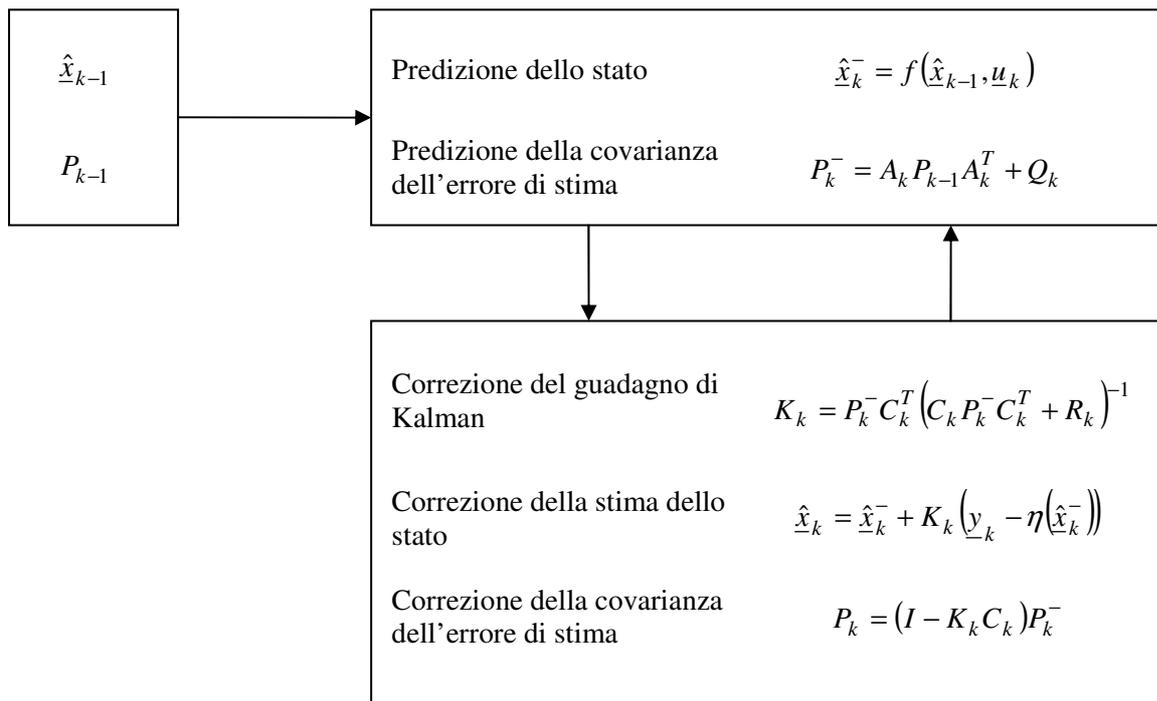


Figure 4-17 – Flow diagram of the Kalman filter operations executed each time step

For a more detailed description of the particular Kalman filtering implementation adopted see [B52] and [B64].

4.5.2.2 Laser altimeter data conditioning

The proper measure of the laser altimeter sensor is the *laser range*. It represents the length of the laser beam between the laser beam generator and the point touched by it on the terrain (current laser footprint). In the LSR coordinate system the vector expressing this measure is

$$\text{Eq. 28} \quad \underline{p}_R^{LSR} = \begin{bmatrix} 0 \\ 0 \\ r_m \end{bmatrix}$$

The aim of this paragraph is to calculate the altitude of the vehicle center of gravity with respect to the footprint of the laser beam q_m . For this aim we have to express the vector \underline{p}_R^{BFR} in NED selecting as NED origin the vehicle center of gravity. The rotation matrix between LSR and BFR is

$$\text{Eq. 29} \quad M_{L2B} = \begin{bmatrix} \cos \beta_L & 0 & -\sin \beta_L \\ \sin \beta_L \sin \alpha_L & \cos \alpha_L & \cos \beta_L \sin \alpha_L \\ \sin \beta_L \cos \alpha_L & -\sin \alpha_L & \cos \beta_L \cos \alpha_L \end{bmatrix}$$

the laser measured vector expressed in BFR is

$$\text{Eq. 30} \quad \underline{p}_R^{BFR} = M_{L2B} \underline{p}_R^{LSR} + \underline{t}_L^{BFR}$$

where \underline{t}_L^{BFR} is the LSR origin expressed in BFR (lever arm effect). The vector now can be expressed in NED by means the following transformation

$$\text{Eq. 31} \quad \underline{p}_R^{NED} = M_{B2N} \underline{p}_R^{BFR}$$

where considering $c(\bullet) = \cos(\bullet)$ and $s(\bullet) = \sin(\bullet)$ the matrix M_{B2N} is defined as

$$\text{Eq. 32} \quad M_{B2N} = \begin{bmatrix} c(\psi)c(\vartheta) & -s(\psi)c(\varphi) + c(\psi)s(\vartheta)s(\varphi) & s(\psi)s(\varphi) + c(\psi)s(\vartheta)c(\varphi) \\ s(\psi)c(\vartheta) & c(\psi)c(\varphi) + s(\psi)s(\vartheta)s(\varphi) & -c(\psi)s(\varphi) + s(\psi)s(\vartheta)c(\varphi) \\ -s(\vartheta) & c(\vartheta)s(\varphi) & c(\vartheta)c(\varphi) \end{bmatrix}$$

Finally selecting the third component of the vector defined in Eq. 31 we have the formula for q_m

$$\text{Eq. 33} \quad q_m = r_m (\sin \beta_L \sin \theta + \cos \beta_L \sin \alpha_L \cos \theta \sin \varphi + \cos \beta_L \cos \alpha_L \cos \theta \cos \varphi) + M_{B2N} \underline{t}_L^{BFR}$$

4.5.2.3 Kalman filtering for ARL estimation

The designed Kalman filter integrates the altitude of the vehicle center of gravity with respect to the footprint of the laser beam q_m , the vertical speed given by GPS (expressed in NED), the altitude of the vehicle center of gravity with respect to the reference geode (WGS84) by GPS, and the profile of the terrain elevation with respect to the reference geode (WGS84) by the DEM (note that the DEM requires as input the GPS geodetic horizontal position measure of the aircraft). Let be

- h the above runway level of the vehicle;
- ΔR the terrain elevation above the runway correspondently to the geodetic coordinates of the vehicle center of gravity;
- V_{Z_m} vertical velocity measured by GPS;
- ΔR_m the terrain elevation (as indicated by the DEM used) above the reference geode (WGS84) correspondently to the geodetic coordinates of the vehicle center of gravity;
- q_m the laser altitude elaborated using the measured laser range r_m (as explained in the previous paragraph);
- h_m the vehicle center of gravity elevation above the reference geode (WGS84) as measured by the GPS;
- h_0 the runway elevation above the reference geode (WGS84) (hypothesized known)

In the following are reported the process equations of the dynamic model to observe by the Kalman filter where the utilized symbols are clarified into the following and in paragraph 1.3. Eq. 34 is the state equation while Eq. 35 is the measures equation.

$$\text{Eq. 34} \quad \begin{cases} \dot{h} = V_{GPS} + w_1 \\ \Delta \dot{R} = -G\Delta R + (\Delta R_m - h_0) + w_2 \end{cases}$$

$$\text{Eq. 35} \quad \begin{cases} q_m = h - \Delta R + v_1 \\ h_m - h_0 = h + v_2 \end{cases}$$

The above runway level of the aircraft and the elevation of the terrain (the point of the terrain on the vertical axes under the center of gravity of the aircraft) are considered as state variables, the laser range and the altitude on the reference geoid of the aircraft (provided by the GPS receiver) as measures and, finally, the vertical speed (provided by the GPS receiver) and the elevation of

the terrain (under the center of gravity of the airplane) on the geoid (provided by the DEM) as inputs. Regarding the main design parameters of the filter, it is worth noting that:

- they should be chosen depending on the speed of the aircraft and the orographic trend of the terrain (an average speed was assumed together with standard profiles of terrain, such as sine waves). In fact the second of Eq. 34 can be interpreted as a linear filter for ΔR forced by ΔR_m , so the cut-off frequency of this “filter” has to be chosen in order to follow the “signal” of elevation, whose frequency content has not to be cut. This aim can be achieved by appropriately choosing the parameter G .
- standard deviations for the process noise were chosen from typical values for the vertical velocity measurements by the GPS receiver (w_1), and from the elevation error at 90% of SRTM (e.g. the selected DEM) data (w_2), slightly increased to take into account the approximations of the model;
- standard deviations for the measurement noise were chosen from the typical error for altitude measurements of laser (v_1) and GPS (v_2).

During the flare phase and in particular when the laser beam footprint enters into the known runway the algorithm substitutes the DEM measure with the known elevation of the runway (e.g. null elevation respect h_0) and adapts the covariance matrixes of the Kalman filter to this new scenario where terrain elevation under the vehicle is perfectly known and laser altimeter measures are more accurate and reliable.

4.6 Configuration and Design Constraints

The development of the autonomous landing system previously described has been carried out taking into account also suitable performance constraints referring to the nominal trajectory tracking and to the performances at touch down point.

The Table 4-3 shows the desired values of the main parameters at the touch down event. The performance constraints referred to trajectory tracking require that vehicle flies into a specified three-dimensional volume around the nominal trajectory, while the performance constraints at touch down point are the ones shown in Table 4-3.

PARAMETER	VALUE
XRW touch down [m]	100
YRW touch down [m]	0
Climb rate [m/s]	-0.5
Pitch Angle [deg]	5
Roll Angle [deg]	0
Heading Angle [deg]	-120
TAS [m/s]	23

Table 4-3 - Touch down desired condition

PARAMETER	VALUE	PROBABILITY
Longitudinal dispersion range	[-60,60] m	95%
Lateral dispersion range	[-6,6] m	95%
Maximum vertical load factor	1.7 g	N/A
Maximum inertial speed	30 m/s	N/A
Minimum TAS	(flaps 35 deg)	20.8 m/s
	(flaps 15 deg)	22.5 m/s
	(without flaps)	24 m/s
Inertial vertical speed range	[-1,-0.1] m/s	N/A
Maximum inertial lateral speed	0.9 m/s	N/A
Bank angle range	[-5,5] deg	95%
Pitch angle range	[3,10] deg	95%
Heading angle range	[-5,5] deg	95%

Table 4-4 - Performance constraints at touch down point

Furthermore, in system development and software implementation other dynamic and structural limits of the vehicle have been considered, as reported in Table 4-5

PARAMETER	VALUE
Maximum bank angle [deg]	30
Maximum flight path angle [deg]	5
Minimum flight path angle [deg]	-5
Maximum TAS without flaps [m/s]	50
Maximum TAS with flaps [m/s]	35
Stall TAS without flaps [m/s]	21
Stall TAS with flaps [m/s]	18

Table 4-5 – Assumed vehicle constraints

Finally, also environmental disturbances limits have been considered in system development and software validation. The maximum value for the wind module (considering both average and

gust components) for each direction is reported in Table 4-6. Also turbulence was taken into account by means of Dryden based modelling.

FLIGHT PHASE	GUST DIRECTION	LIMIT (M/S)
Alignment	All	10
Approach and Touch Down	Lateral	5
	Nose	10
	Tail	2

Table 4-6 - Environmental disturbances condition

5 DEVELOPMENT OF VERIFICATION TOOLS

All the algorithms related to the CIRA GNC System and, in particular, that described in this dissertation has been developed following a well-defined development process cycle deepened in the following capitol (§6.1). The methodology adopted is the model-based design and the development of models to support the verification activities during the different phases of the process is crucial in the same way as the proper algorithms development. The several environments used for the verification stages are briefly described in the following paragraph and was developed by CIRA with the support of the author during the last ten years. The only two models detailed in specific paragraphs (e.g. §5.1.1 and §0) is the Laser Altimeter model and DEM model, that were developed in the framework of the author Phd and have been included in the publication [AR8].

5.1 Numeric Simulation Environment

The numeric simulation environment refers to a complete detailed model of the aeronautical flying demonstrator FLARE (see paragraph 5.1.1). This simulation model has been implemented in Matlab/Simulink, as shown in Figure 5-1 and can be considered representative of a UAV of medium altitude unmanned aerial vehicle.

The numeric simulation environment also integrates the model of all the GNC algorithms (including trajectory planning algorithms for en-route, terminal area, approach and landing operations) and in general all the application SW downloaded in the Flight Control Computer used in the flight demonstrator. Moreover the numeric simulation environment also takes into account some peculiarities of the laboratory test rig such as HW signals filtering and serialization. In Figure 5-1 are highlighted in yellow that modules related to the Flight Control Computer while highlighted in orange that ones used also in the laboratory test rig environment. Below a short description of main simulation modules is reported:

AC_SIM - This module includes the simulation model of FLARE. It includes: the 6DoF model of the aircraft, engine model, servo-actuation models, landing gear detailed model [AR7], and external environment (atmosphere) including a Von Karman or Dryden model of turbulence. This module is configurable to set the turbulence level, type of servos (position or velocity controlled), injecting fixed wind disturbances, etc. Aerodynamics has been tuned using parameter identification from flight data, while mass and inertia data has been derived from internal avionic configuration and constructor data.

GNC2SIM - this module simply adapts output format of GNC algorithm to the module of AC_SIM.

OFFL_NAV_Sensors - This module contains the off-line models of on-board sensors used for both rotation and navigation that entails: GPS, AHRS, ADS, Laser Altimeter, and Radar Altimeter. This models includes dynamics, latency (where applicable) and measurement error models that can be configured before a simulation session.

OFFL_AC_Sensors - This module includes the off-line models of on-board sensors used by auxiliary aircraft systems. It entails: aero surface position sensors, engine sensors (rpm, temperature, etc.) landing gear sensors.

Cmd_Gen - This is used to generate all commands (for configuring and operating a simulation run) to GNC and AC_SIM.

SIM2GNC - This module simply adapts output format of AC_SIM to GNC input format.

ODID SENORS - This module contains the off-line models of on-board sensors used for the functionalities of Obstacle Detection and Identification. This block is used for simulating the algorithms related to the function of Autonomous Collision Avoidance implemented in the CIRA GNC but not threatred in this thesis.

ODID VIRTUAL - This block replaces the same on-board module for virtual emulating during flight testing the presence of an intruder (a potential flying obstacle).

INTRUDER MODEL - This module includes the simulation model of the vehicle Intruder. It is based on a cinematic model with the same dynamic performances of the real intruder used for in-flight testing of the Autonomous Collision Avoidance Functionality.

The environment used for simulation includes some tools for result visualization and other tools for performing Montecarlo analysis by varying uncertainties that are included in both AC_SIM for each aircraft component (aerodynamics, engines, mass, inertia, etc.) and sensors for errors and dynamics. Moreover, several environmental disturbances can be introduced (wind gust and turbulence) and failures on the sensors can be also reproduced.

5.1.1 Laser Altimeter Model

With the aim of properly verifying the algorithm regarding the above runway level estimation (as described in §4.5.2) it has been necessary to deeply modify and improve the *Navigation Sensors Model* of the *TECVOL Numeric Simulation Environment* for what concerns the laser altimeter model.

The laser range r_m is the distance between the laser's firing point and the laser beam footprint on the terrain. The altimeter model previously used in the *TECVOL* framework is based on the assumption of flat earth under the aircraft and calculates the laser range by using the simulated true attitude of the aircraft Euler angles ϑ_t and ϕ_t), the position of the on board altimeter with respect to the center of gravity in NED coordinates ($M_{B2N} \underline{t}_L^{BFR}$ as already described in §4.5.2.2), its orientation in a body reference frame (angles α_L β_L), and the simulated true altitude q_{mt} . For the true (e.g. not still corrupted by typical laser altimeter errors) laser range r_{mt} calculation is used the inverse formula of Eq. 33:

$$\text{Eq. 36} \quad r_{mt} = \frac{q_{mt} - M_{B2N} \underline{t}_L^{BFR}}{(\sin \beta_L \sin \theta_t + \cos \beta_L \sin \alpha_L \cos \theta_t \sin \varphi_t + \cos \beta_L \cos \alpha_L \cos \theta_t \cos \varphi_t)}$$

The simulated measure of the laser altimeter r_m is then calculated corrupting r_{mt} with the typical laser altimeter errors (like for example bias, scale factor, discrete resolution, etc.).

The aim of the proposed altimeter model is taking into account the real orographic trend of the terrain under the aircraft and the real elevation on the reference geoid of the laser beam footprint. Since the laser beam footprint elevation can be determined only by identifying the intersection between the laser beam and the ground, and since the ground elevation in this intersection can be known only using a map of the orographic trend which requires its coordinates on the reference geoid (unknown *a priori*), it was necessary to implement an iterative process for the calculation of the laser range at each time. The designed iterative process is described in the next.

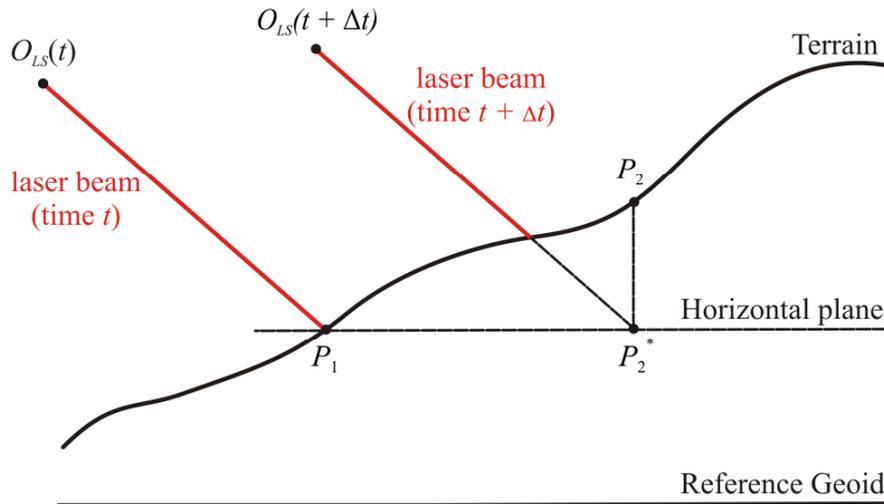


Figure 5-2 – Iterative process scenario in xz plane

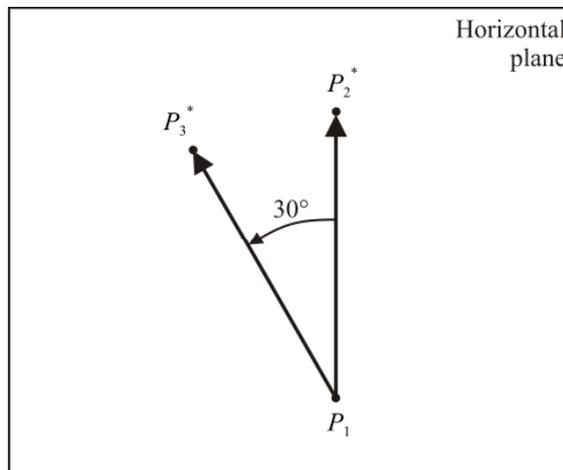


Figure 5-3 – Iterative process scenario in xz plane

With reference to Figure 5-2, Figure 5-3 and Figure 5-4, at each generic time $t+T$, an estimation P^* of P (the true position on the ground of the footprint of the laser) is made as the intersection between the laser beam and the plane Γ passing through the following three points:

- P_1 that is the footprint of the laser beam at the immediately previous time t (see Figure 5-2);
- P_2 point on the terrain profile whose horizontal geodetic coordinates (longitude and latitude) coincide with the horizontal coordinates of P_2^* , that is the intersection of the laser beam at the current time and a horizontal plane passing for P_1 (see Figure 5-2);
- P_3 point on the terrain profile whose horizontal geodetic coordinates (longitude and latitude) coincide with the horizontal coordinates of P_3^* , that is determined by rotating 30° around the vertical direction the vector $(P_2^* - P_1)$ and by summing this rotated vector to P_1 (see Figure 5-3).

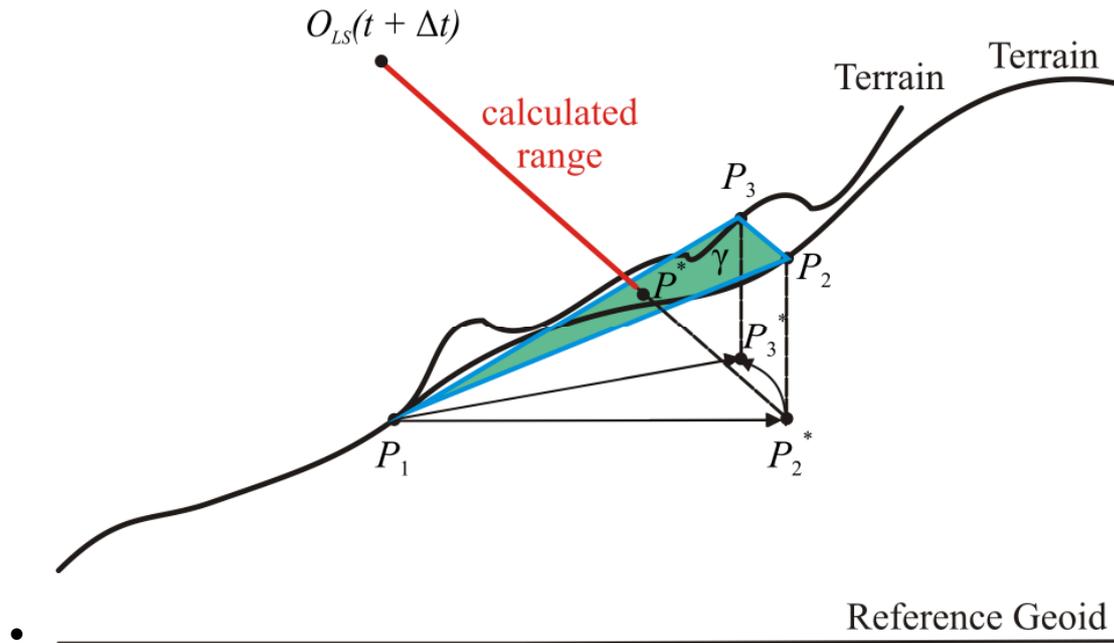


Figure 5-4 – Iterative process scenario in 3D vista

So the geodetic coordinates (longitude, latitude and elevation) of the point P^* and the laser range corresponding to it (distance between P^* and the laser's firing point) are calculated (see Figure 5-4). A specific check was developed for verifying the reliability of the estimation, considering unreliable when the plane Γ was parallel to the laser beam or when P_1 had an elevation greater than P_2 but smaller than P^* .

The process described above is repeated iteratively, considering, for each step, the point on the terrain profile whose horizontal geodetic coordinates coincide with those of P^* , as the “new” P_1 ; it is found by using the simulated map of real orographic trend. This process terminates when the established maximum number for the iterations (consistent with the requirements of the real time simulation) is exceeded or whether the differences in range and calculated position for the footprint of the laser beam (in NED axis) between successive iterations are both smaller than a given threshold. Then everything is repeated for the following considered instant.

The validation phase of the model was carried considering as test scenario an autolanding maneuverer and a not flat terrain elevation profile. In Figure 5-5 is shown the behavior of the model proposed comparing the true simulated above ground level with the above ground level calculated by using the laser range r_m outputted by the model. The error due to the iterative process is always confined under 5 cm and is lower than the typical error of laser altimeters used for these applications.

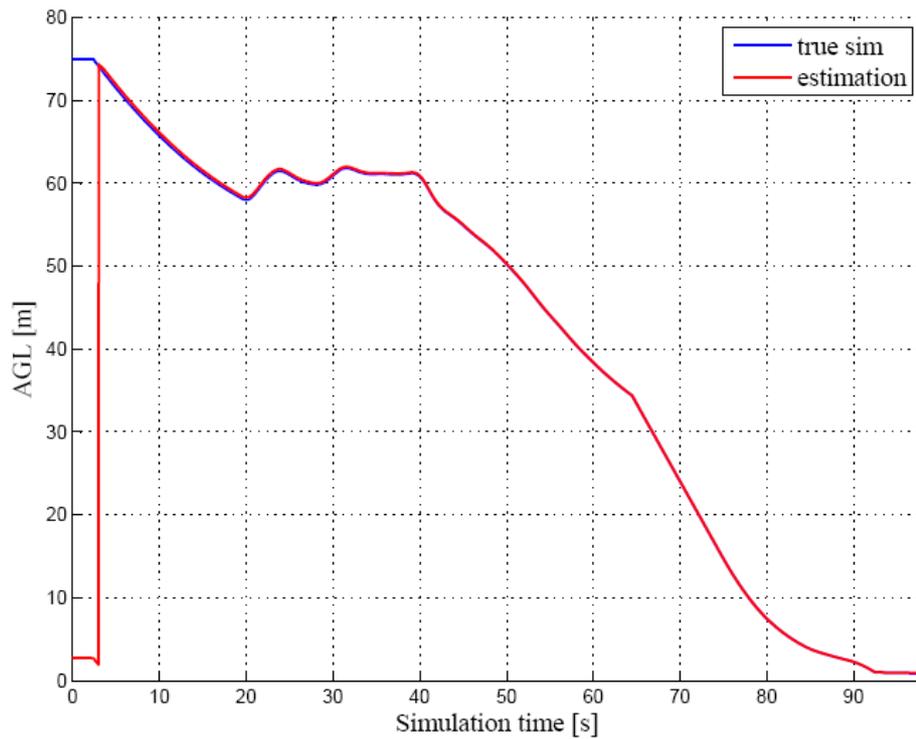


Figure 5-5 – Laser altimeter model validation results

5.1.2 DEM Error Model Development

A DEM model was developed as an additional sensor model. The DEM model developed requires the coordinates of the center of gravity of the aircraft as input and provides the elevation of the point on the earth surface under these coordinates as output. This elevation “measure” is obviously affected by a modeled error. In particular, the error model of the “map sensor” was developed considering as references the available literature researches about the errors of the SRTM data (see paragraph 3.3.2).

The error components identified are described in the following.

- a sampling error (in a digital elevation map every element is referred to a certain area and represents an average elevation of it);
- geo-location (random) errors in latitude and longitude;
- a very long wavelength error caused mainly by the Shuttle’s attitude manoeuvres during the data collection for the DEM. This type of error is negligible if the platform used for the data collection is an airplane, such as for NEXTMAP data;
- a short wavelength random error due to several causes with a different nature (for more detailed information see [B67]), not negligible also if the platform used for the data collection is an airplane, but smaller for NEXTMAP data compared to SRTM data [B69];

- a quantization error.

These error components added to the “true” elevation of an ideal map of the orographic terrain trend give a measure of elevation. The sampling error, the variance of geo-location and random errors, the amplitude and spatial frequency of the long wavelength error have been set according to some research projects conducted about SRTM data [B67].

The designed digital map sensor error model can be used in combination with any real orographic trend.

5.2 Laboratory Test Rig

In the framework of the projects connected with the UAV CIRA program, with the aim of performing a real-time ground validation of the on board segment SW, a test rig that simulates the on board system and the on ground system has been done.

The laboratory experimental testing is needed to check correctness of software implementation, to solve software to hardware integration issues, to verify real time execution of algorithms, and to perform a final functional assessment of the overall system before going in flight.

The figure below depicts a typical architecture for a test rig performing such laboratory testing [AR4].

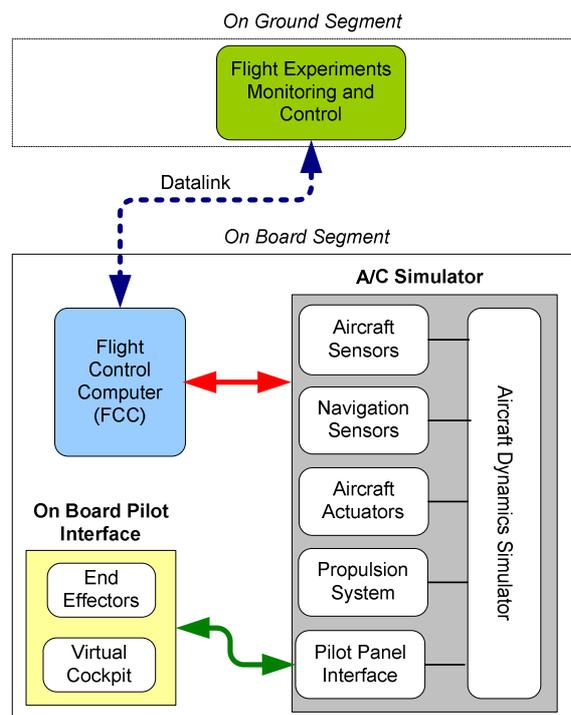


Figure 5-6 - On Ground Validation Test Rig

The FCC of the test rig is the same Flight Control Computer used for the on board segment. It is connected to the vehicle simulation model described in §5.1. The rig contains also modules for the simulation of other relevant subsystems (such as data link), a ground control station (that might be a functional emulator or the actual ground control station) and a simulator workbench that is used for monitoring and configuring the real time simulator and the laboratory test execution. In particular the Pilot Panel Interfaces allows executing pilot-in-the-loop simulation whereas the pilot has the task of on-board safety pilot as in the real in-flight tests with FLARE.

5.3 Flying Demonstrator

The CIRA aeronautical experimental platform FLARE (Flying Laboratory for Aerospace Researches) is a piloted flying test-bed, whose on board avionic system has been designed and integrated by CIRA. This platform is able to perform flight testing of automatic take-off and landing, mission automation and Detect, Sense & Avoid systems. The flying platform has a wing span of about 9 m, a maximum take-off weight of 450 Kg, a maximum speed s/l about 218 km/h and a cruising speed about 190 km/h.

The on-board pilot has the task to put the aircraft in the condition foreseen for the flight test and to monitor the flight test overriding (with suitable available switches) in case of safety problems.



Figure 5-7 – CIRA’s Aeronautical Flying Platform FLARE

This platform includes a Ground Control Station (GCS) installed in a big shelter fixed on the ground near the runway (see Figure 5-8). GCS is designed to manage telemetry data, present them to the flight test engineers through dedicated human-machine interfaces and for remote reconfiguration of the on board avionics system. GCS entails: a bidirectional datalink with an operating range of about 6 km, a meteorological station, a GPS base station, used to send differential correction to the on board GPS sensor, a virtual Cockpit HMI for presenting

information in a pilot-like cockpit, an engineering HMI for presenting information in an engineering-like way, autonomous mid-air flight, automatic landing and automatic collision avoidance HMI for controlling flight experiments of such different algorithms.



Figure 5-8 – CIRA Aeronautical Ground Control Station for FLARE

The on board Avionics System of FLARE includes all devices needed to perform the in-flight experimental validation of advanced guidance, navigation and control functionalities. A brief description of such devices, integrated using Commercial-Off-The-Shelf (COTS) components, is below reported.

- A Flight Control Computer (FCC) based on a PowerPC processor integrated in modular HW architecture. The SW development environment for the FCC allows for automatic real-time coding directly from Simulink diagrams.
- A navigation sensor suite including a two DGPS-RTK L1/L2 system capable to provide position measurements with an accuracy of few centimetres, a solid state Attitude Heading Reference System (AHRS) with MEMS sensor technology and two dedicated sensors for distance to ground measurements respectively using radar and laser technologies. The altimeter sensors can be alternatively mounted because of weight limitations.
- A radar device installed on the top of the plane for obstacle detection.
- Digital electromechanical servos to command both aerodynamic surfaces and throttle, driven by the FCC via PWM signals.
- A digital bidirectional data link system able to exchange data between on board FCC and the Ground Control Station with a maximum bit-rate of 9.600 bit/sec in uplink and 115.200 bit/sec in downlink.

6 NUMERICAL AND EXPERIMENTAL ASSESSMENTS

This chapters presents some examples of numerical verification and ground validation phases results and, moreover, also relevant flight demonstrations will be also described that will assess effectiveness of the proposed system. All the tests were performed by CIRA in the framework of the aeronautical project TECVOL.

6.1 Development and Implementation Process

The design and test process adopted for the algorithms described in this thesis are the same ones adopted in the TECVOL project. The development phases have been performed by using a top-down process, usually well known as V-cycle (Figure 6-1). Actually the whole design process can be divided into three main phases: *requirement definition & system design*, *implementation & on-ground testing* and *integration & flight validation*. In the following paragraph will be briefly described each phases. The development cycle includes all the phases of a GN&C system realization from the definition of sub system requirements to HW/SW prototyping up to the flight tests and post flight analysis activities. This approach has remarkable advantages, especially on the quality of the final product, including:

- close correlation between control system specifications, SW implementation and related documentation;
- reduction of the code generation time;
- “strong” control over implementation and/or specification errors.

6.1.1 Requirement definition and System Design

During this phase, system and user requirements are defined, system architecture is delineated and the design of the control system is carried out through the use of simulation models (*Model Based Design*). The simulation models are developed in Matlab/Simulink/Stateflow environment. All requirements (from GN&C system requirements to equipment specifications) produced during the design activities are defined with unique identification tags, traced in both directions of development stages (downward and upward) and maintained during all the project life. All test reports are traced with respect to one or more requirements. Justification and compliance matrixes are provided at each stage of development giving evidence about analysis and tests performed to define each requirement or to verify design compliance. Each subsystem is validated through robustness and performance off-line analysis.

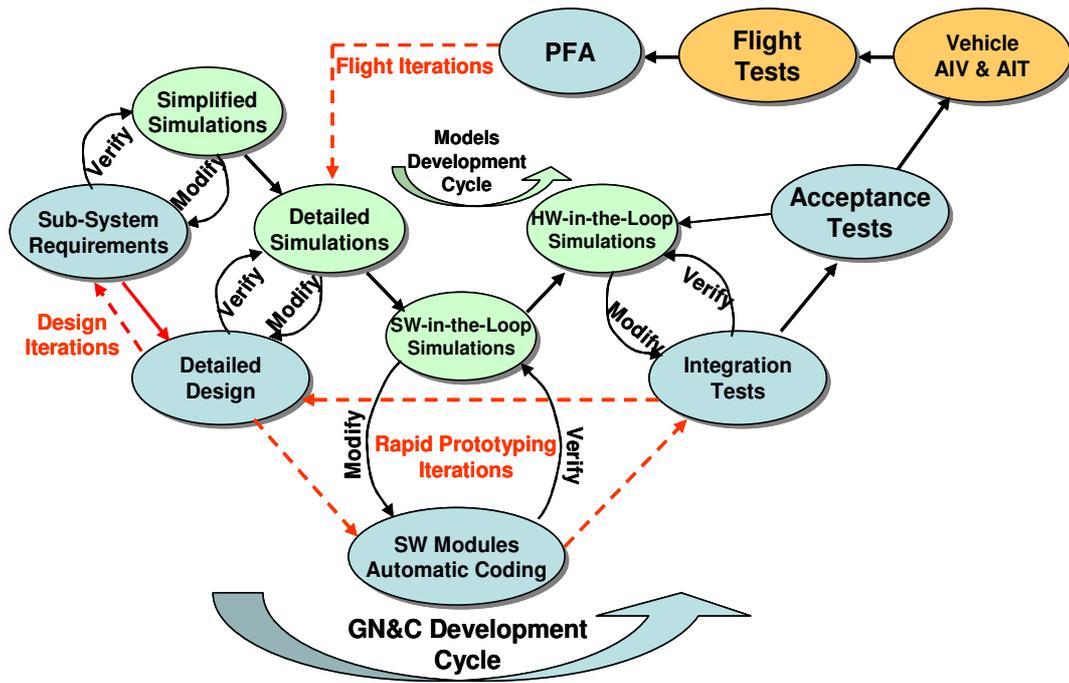


Figure 6-1 – GN&C Algorithm Development Cycle at CIRA

6.1.2 Implementation & On-Ground Testing

The GNC modules are designed directly by means of a Matlab/Simulink environment endowed with a specific tool (Real-Time Workshop) that allows the Automatic Program Building for a specific target machine. This approach has two main advantages:

- automation logics and control algorithms can be developed using a high level programming language;
- debugging can be easily done during both preliminary simulations, while defining the control strategy, and validation of the control system using HIL simulations.

During this phase the high level code is integrated with C/C++ code and downloaded to the target machine, which manages the resulting application according to its micro-kernel's primitives. Always during this phase, HIL simulations are performed, allowing validation and testing of the control system, interacting with both the simulated environment and the real instrumentation (feedback sensors, real Human Machine Interfaces, Airborne Virtual Cockpits, and Ground Control Stations). An interesting feature of this technique is the capability to monitor and/or to modify (using SW tools) the system parameters and the control strategy. In this way, the validation process and the control system fine tuning become easy and quick. This will allow executing rapid iterations among SW low level specification (Detailed Design), SW unit tests and integrated real time tests phases (*Rapid Prototyping Iterations*).

6.1.3 Integration and Flight Validation

This is the final development phase. The GNC HW equipment is going to be integrated and GNC SW is going to be targeted and deployed in the host flight control computer. The GNC HW/SW equipment, finally, is going to be integrated in the aircraft and the system accepted after successful acceptance test sessions. During the flight tests analysis of data and events, comparison with expected results and parameter model identification are used after a test flight for performing a validation and a refinement of simulation models increasing their prediction accuracy and/or reducing their level of uncertainty and for identifying possible GN&C algorithm enhancements or needed modifications (due to, for example, a not satisfactory behaviour during flight). This further design iteration is executed in such projects where multiple missions are planned with possibly increasing level of mission difficulty or risk. Finally, this iteration is used to gather the maximum possible value added from the execution of a flight test in order to perform next missions with lower risks or with an higher level of difficulty and to finally increase know-how and experience for future projects.

With reference to the above development cycle, the activities described in this dissertation have been mainly focused on the design of algorithms related to the autonomous take-off and landing. Anyway, CIRA, supported by the author, has performed all other activities of above cycle to finally execute some flight tests with the above described flight demonstrator FLARE. The following paragraphs report the key results of these activities that demonstrate effectiveness of the proposed algorithms.

6.2 Numerical and Laboratory Assessments

In the following paragraphs is reported the numeric and laboratory real time assessment as is expected from the process cycle described above.

6.2.1 Numerical Assessments

Numerical assessment of the proposed algorithms was carried out by means of the numeric simulator described in §5.1.

Firstly, hereafter it is considered the free flight path generation and line-of-sight tracking algorithms described in §4.3 [AR1]. Numerical test has been performed using Montecarlo analysis for randomly varying initial position and velocities with the same final point. In any of the runs both algorithms performed as expected, bringing the vehicle to the final conditions

(within some specified accuracy) and using continuous commands that satisfy vehicle constraints. Just for example, below we show the trajectory generation and tracking results obtained with reference to initial and terminal conditions specified in the following table, under strong wind disturbance. The UAV nominal inertial velocity is $V= 25$ m/s.

	Initial conditions	Terminal conditions
x [m]	1450	1900
y [m]	1000	0
z [m]	200	75
γ [deg]	0	0
χ [deg]	0	0

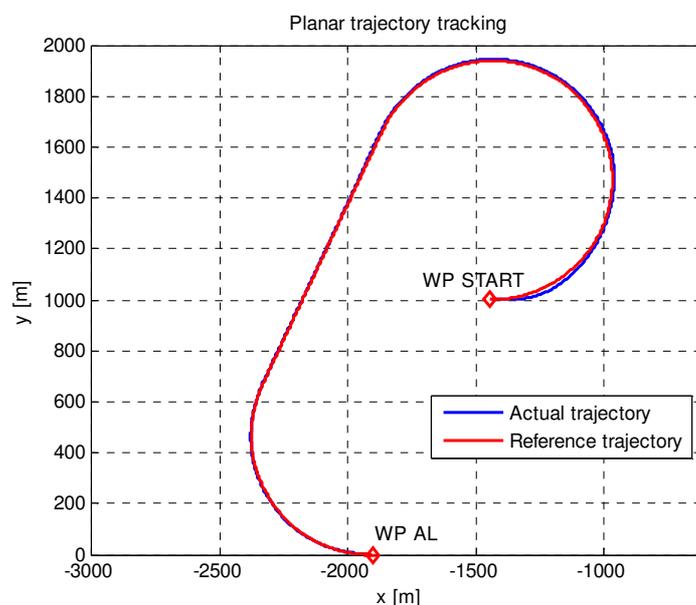


Figure 6-2 – Horizontal Free Flight Trajectory Generation and Tracking Simulation

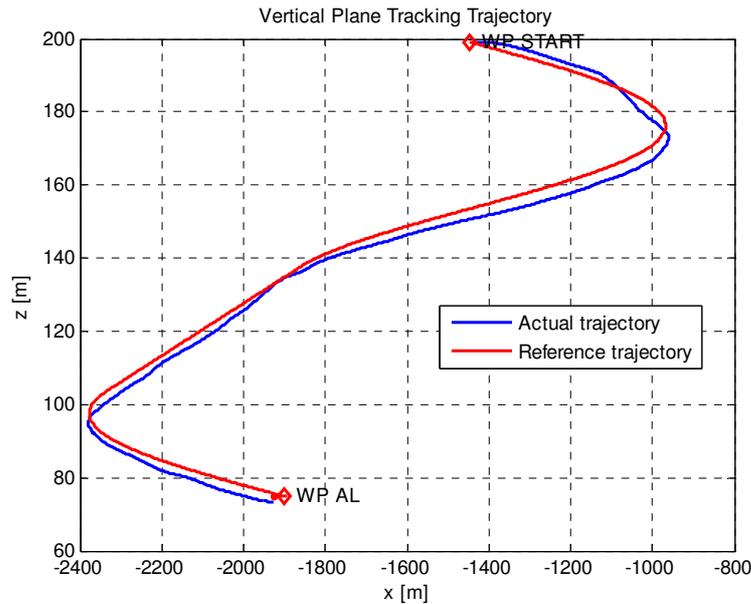


Figure 6-3 – Vertical Free Flight Trajectory Generation and Tracking Simulation

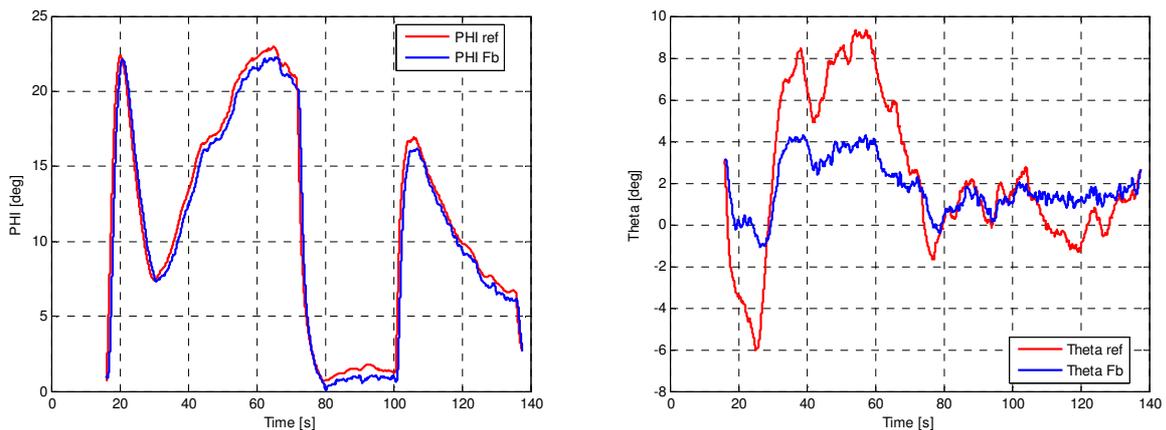


Figure 6-4 – Reference and actual values of commands in free flight trajectory planning

In Figure 6-2 and Figure 6-3 are showed the results of the trajectory generation and related horizontal and vertical tracking, while Figure 6-4 shows the references in attitude generated by the line-of sight tracking algorithm and the related actual values as obtained by means of the flight control system module described in §4.3.3.

A simulation example is hereafter shown where the algorithms have been executed in a case with fixed no-fly zones. The proposed scenarios include a list of waypoints to be followed by the aircraft, reported in the table below.

WP n.	x [m]	y [m]	z [m]	χ [deg]	γ [deg]	V [m/s]
1	-2500	0	150	56	0	35
2	-1500	1500	150	346	0	35
3	500	1000	120	0	0	35

In the following table is reported the specification of the list of no-fly zones, whose positions (coordinates x - y of the centre and radius) are known before flight.

Area n.	x_c [m]	y_c [m]	R [m]
1	-500	1300	500
2	-3000	5000	1500
3	-3000	8000	1000
4	-2000	11000	1000

Figure below shows behaviour of the algorithm in this case.

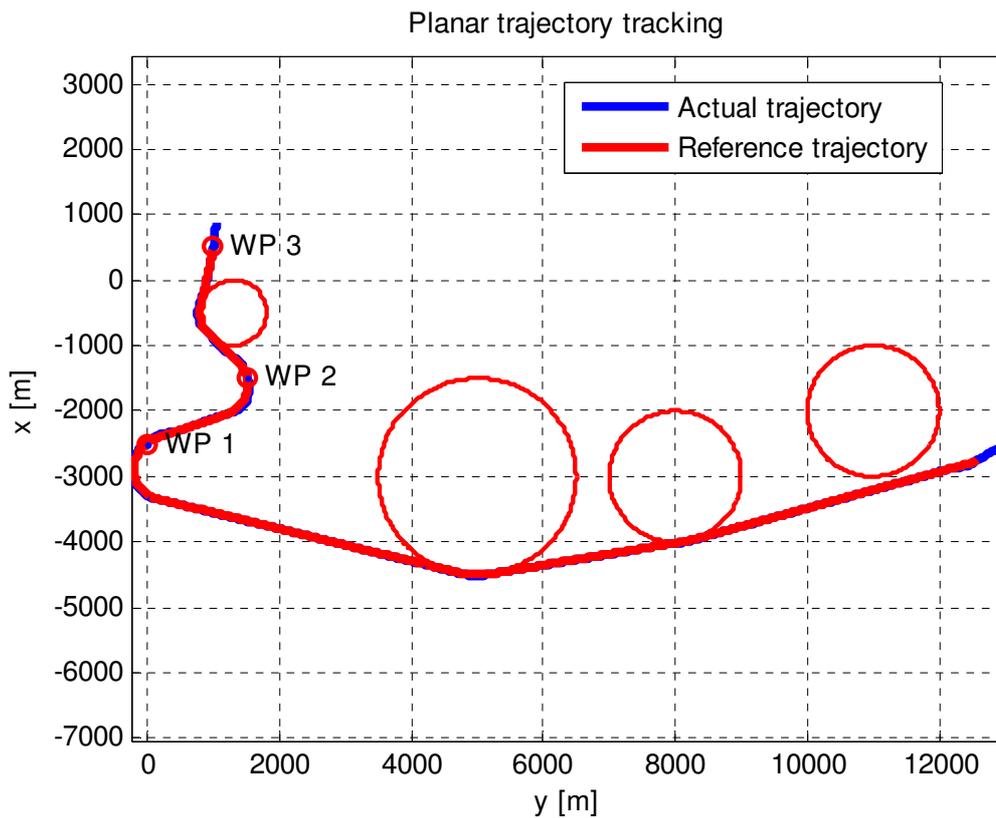


Figure 6-5 – Flight Planning with (fixed) no-fly zones

In order to show the adaptive capabilities of the proposed algorithm in the presence of changing conditions during flight, some scenarios in which the environment conditions abruptly change are considered. In particular, these scenarios account for the presence of new no-fly zones to

avoid (not known before flight) due, for instance, to either changed weather conditions or upcoming threats to face. On the other hand, a no-fly zone known before flight may become “permitted” during flight as a consequence of some changed environment conditions. These scenarios clearly require the adaptation of the trajectory planner algorithm that must on-line compute a new trajectory compliant with the updated constraints, provided that these are available to the trajectory planner. To this end, the trajectory planner is supposed to be provided with the updated list of forbidden zones during flight. In the next figures, the horizontal and vertical trajectories are shown for the following scenarios:

- a no-fly zone known before flight becomes “permitted” during flight;
- the position of a no-fly zone is changed during flight;
- a new no-fly zone comes up during flight.

In the first scenario (see Figure 6-6), at a certain moment during flight, the trajectory planner is informed that a given zone (the green one in the figure) is no-longer forbidden, so the trajectory planner is provided with an updated list of the forbidden zones and it readily computes a new reference trajectory accounting for the current list of no-fly zones.

In the second scenario (see Figure 6-7), the trajectory planner is informed that the position of a given no-fly zone is changed (the new zone is depicted in black), thus it regenerates the reference trajectory according to the updated list of no-fly zones.

Finally in the last scenario Figure 6-8, a new forbidden zone (black zone in the figure) not known before flight is supposed to come up during flight. Also in this case, the trajectory planner is able to adapt itself to this changing scenario and to on-line compute a new feasible trajectory, that is, a trajectory which does not cross any forbidden zone.

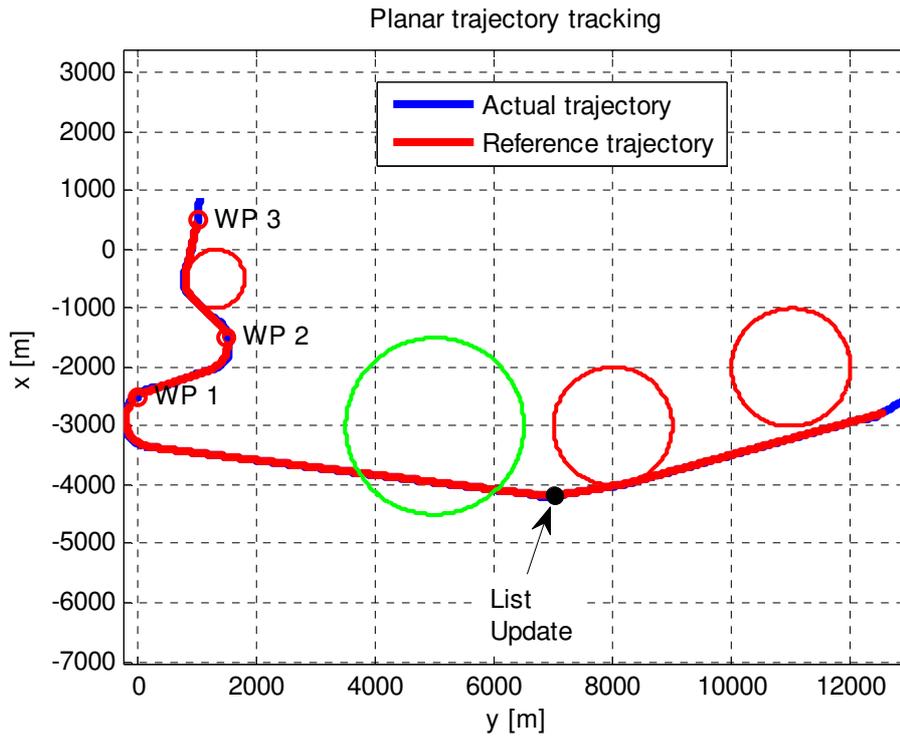


Figure 6-6 – A no-fly zone is no more forbidden

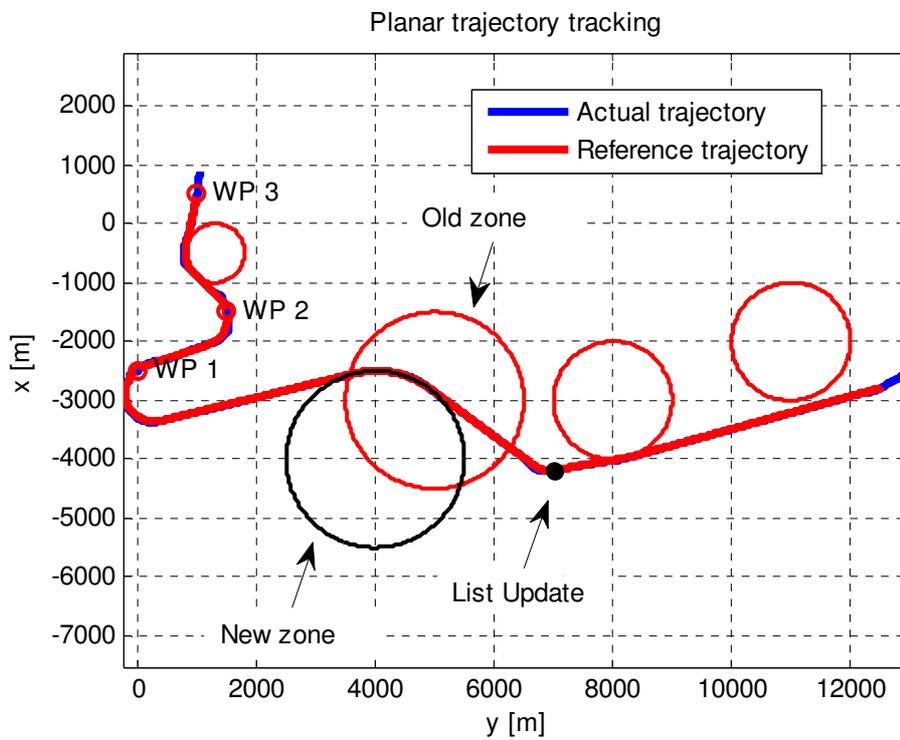


Figure 6-7 – The position of a no-fly zone changes during flight

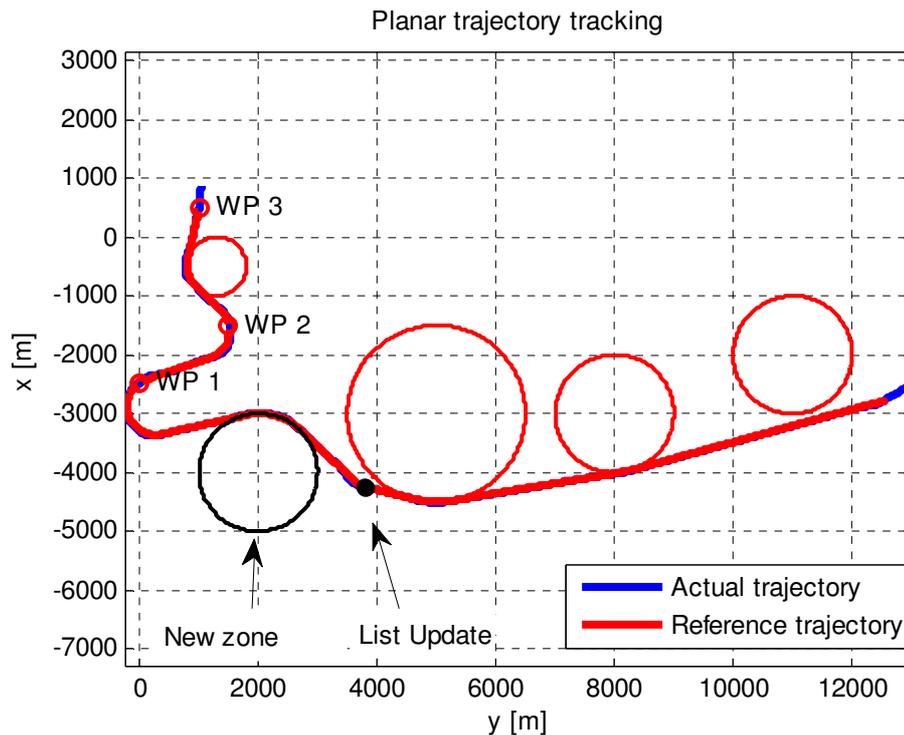


Figure 6-8 – A new zone becomes forbidden during flight

6.2.2 Laboratory Assessments

All the described algorithms has been implemented in real time using the process described in §6.1 and then they have been extensively tested in real time using dedicated HW-in-the-Loop laboratory test rigs, before being used in actual flight demonstrations.

As first example of laboratory assessment the behavior of two different algorithms related to approach and landing autonomous execution are presented, based on an identical external environment. The case A regards the algorithm using a pre-fixed nominal trajectory for the flight control law references generation called *PreFix* algorithm [AR14][AR15], while the case B concerns the algorithm described in §4.4 and [AR5], executed step by step from the start of the flare phase up to the Touch Down event, called *Vzlin* Algorithm.

These cases refer to an automatic landing maneuver, without failures during the flight, in presence of a persistent wind which is tail oriented during approach and touch down phases and a wind gust in opposite direction (head wind gust) injected before the end of the approach phase. The system is commanded to perform an automatic landing maneuver starting from an arbitrary position and the simulation is stopped 1.5 seconds after the contact of the rear landing gear (touch down event). The complete test conditions, including initial position and speed of the vehicle, waypoint to be reached in the alignment phase, environmental conditions and so on are reported in the below table.

Runway	Orientation (NEU reference) [deg]	-120
Sensors	Noise and Errors	Yes
	Serialization	Yes
Vehicle initial position and inertial speed	$x_{0\ RW}$ [m]	-2600
	$y_{0\ RW}$ [m]	0
	$z_{0\ RW}$ [m]	75
	$V_{0\ inertial}$ [m/s]	35
	χ_0 (NEU reference) [deg]	-120
	γ_0 [deg]	0
Atmospheric disturbances	Permanent Wind magnitude [m/s]	5.5
	Permanent Wind direction (NEU reference) [deg]	-120
	Wind Gust magnitude [m/s]	4
	Wind Gust direction (NEU reference) [deg]	60
	Turbulence	Yes
Alignment WP	x_{WP} [m]	-1900
	y_{WP} [m]	0
	z_{WP} [m]	75
	χ_{WP} (NEU reference) [deg]	-120
	γ_{WP} [deg]	0

Table 6-1 – Autoland test conditions

The Figure 6-9 shows the profile of the altitude in dependence of the longitudinal runway axis X for both the *PreFix* and *Vzlin* algorithms. The red line represents the nominal altitude reference trend, the blue line represents the vehicle altitude trend using the *PreFix* algorithm, and the black line represents the vehicle altitude using the *Vzlin* algorithm. The figure shows a better altitude reference tracking using the *Vzlin* algorithm respect the *PreFix* algorithm. A zoom of the figure during the flare phase is presented in Figure 6-10.

In particular Figure 6-10 demonstrates the better behaviour of the *Vzlin* algorithm in response to the wind gust disturbance injected during the flare phase. In fact the longitudinal touch down point using the *Vzlin* algorithm is closer at the desired value defined in Table 4-3 and reported below for reader convenience, with respect to the correspondent result of the other algorithm.

PARAMETER	VALUE
XRW touch down [m]	100
YRW touch down [m]	0
Climb rate [m/s]	-0.5
Pitch Angle [deg]	5
Roll Angle [deg]	0
Heading Angle [deg]	-120
TAS [m/s]	23

The Figure 6-11 shows the profile of the vertical speed (feedback and its related control reference) during flare and pre-touch down phases in dependence of the longitudinal runway axis X for both the algorithms.

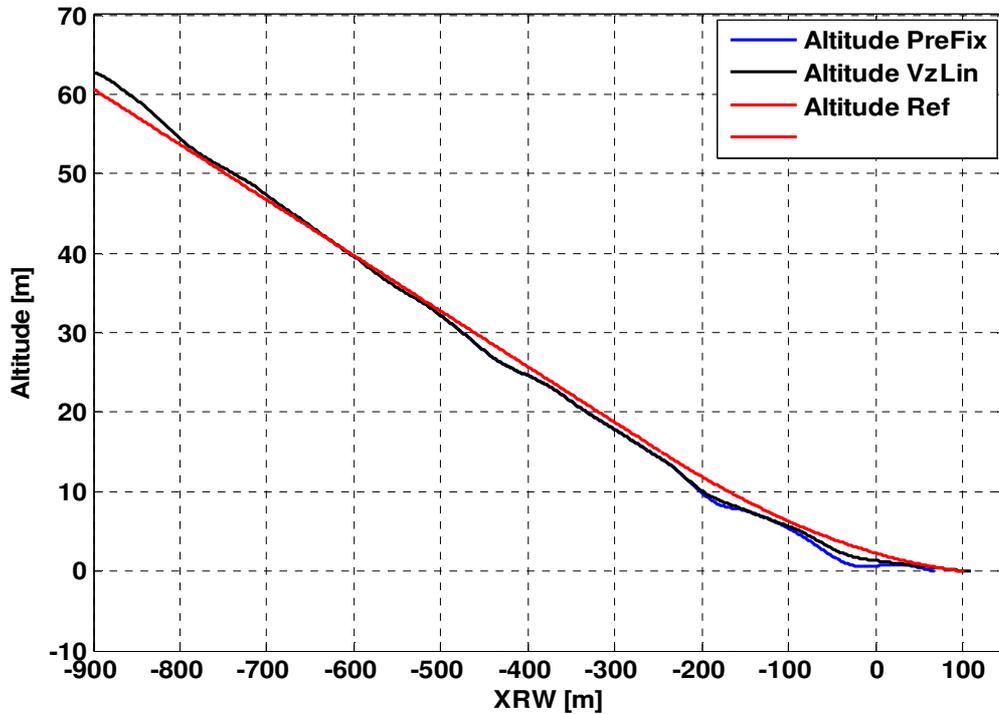


Figure 6-9 – Altitude profile using both the AL algorithms

Also in this case the above figure demonstrates the better behaviour of the *Vzlin* algorithm and considering the root means square of the control error between the nominal vertical speed reference and the vehicle altitude and considering the desired vertical speed at the touch down event.

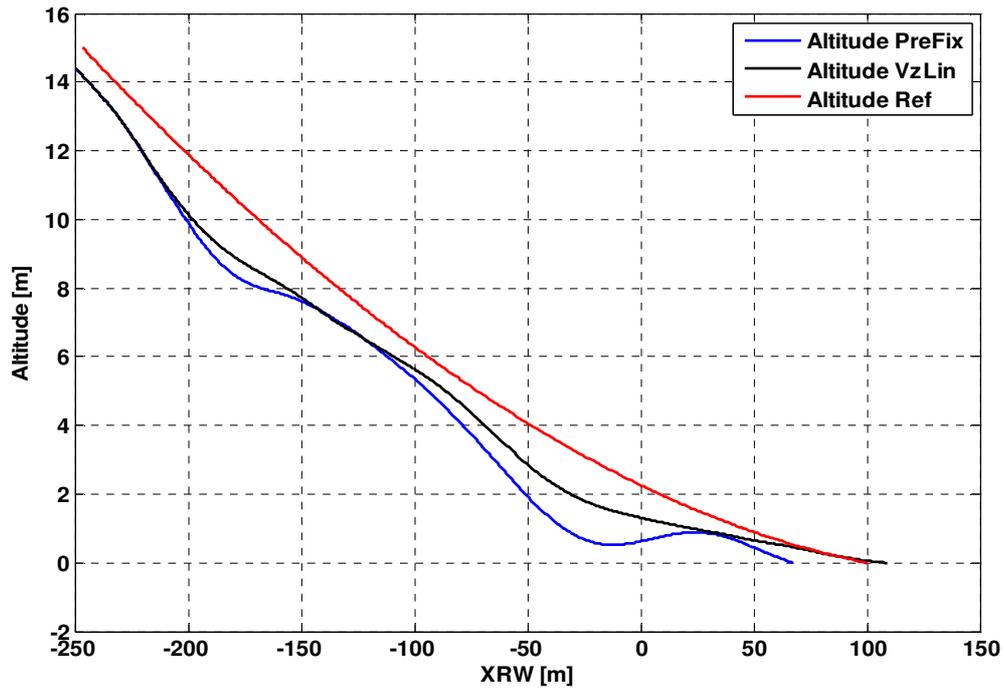


Figure 6-10 – Altitude profile using both the AL algorithms during the flare phase

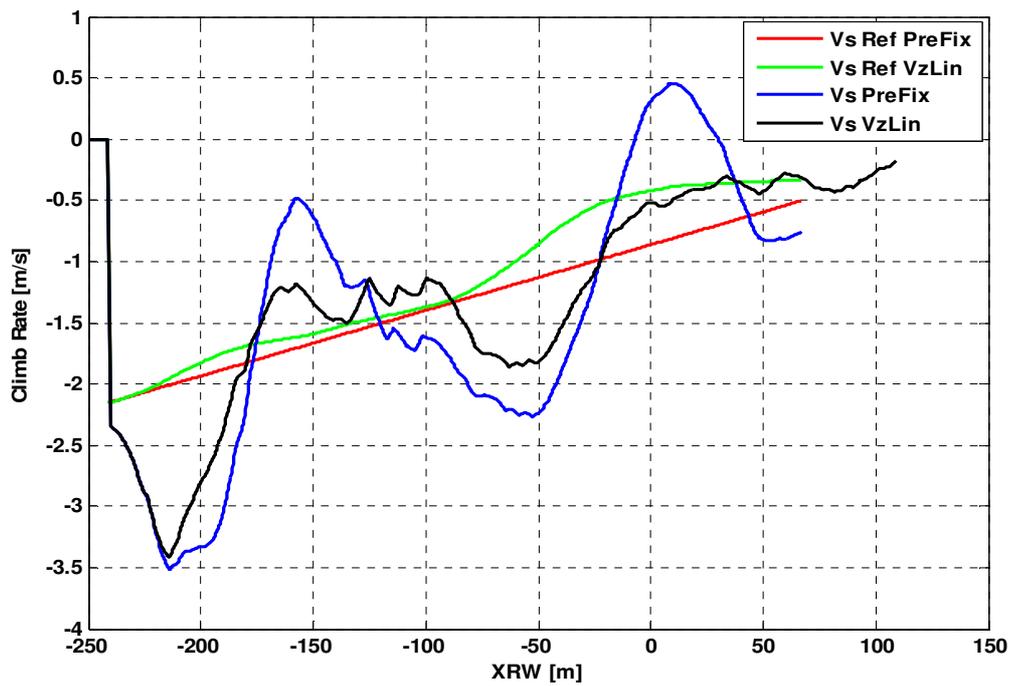


Figure 6-11 – Vertical speed tracking during Flare and Pre-Touch Down phases

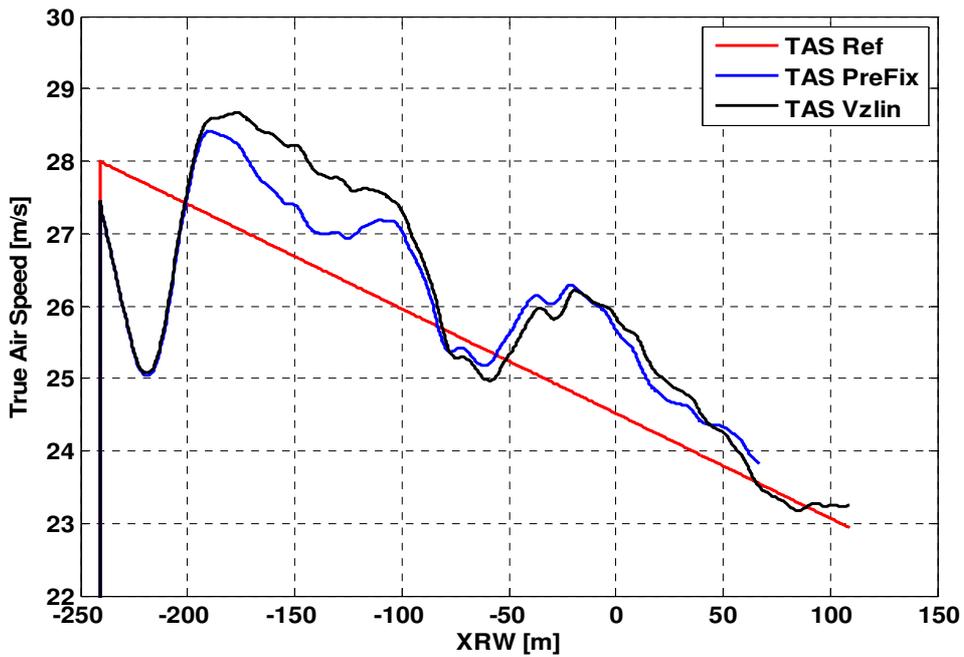


Figure 6-12 – TAS profile during Flare and Pre-Touch Down phases

Figure 6-12 shows the profile of the TAS in dependence of the longitudinal runway axis X for both algorithms.

In Figure 6-13 are shown the results of the positioning estimation (described in §4.5.1) laboratory validation. The test scenario is the flare phase and the touch-down event. In particular is shown the vertical speed estimation (red line) compared with the true simulated vertical speed (blue line) and the vertical speed estimated by simple linear filtering of the GPS measure (black line).

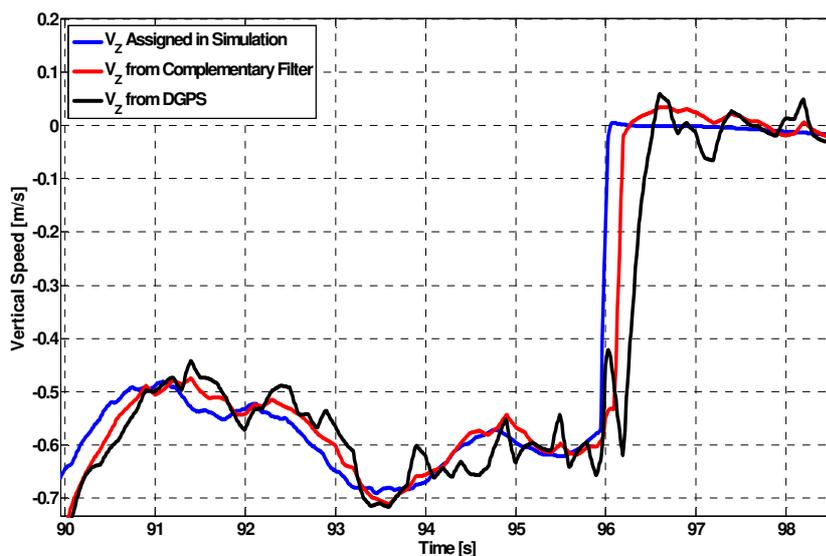


Figure 6-13 – Complementary filter validation with a test scenario with flare phase and touch-down event

The figure shows as the complementary filter has the double advantage with respect to a linear filter of reducing noise and showing a larger band.

The Above Runway Level estimation algorithm described in §4.5.2 was validated using several test scenarios. Below are reported some results related to a case where before the runway is placed a little hill of 20 meter of maximum elevation and sinusoidal profile. The case consider to have available a GPS only in standalone configuration (e.g. no differential correction available) and a not so accurate DEM with the same features of the SRTM DEM cited in §3.3.2. Figure 6-14 compares the following four signals

- true simulated ARL (label true sim and black line)
- Kalman filter ARL estimation (label KF and red line)
- ARL calculated only using GPS altitude measure and the known runway elevation (label GPS and blue line)
- ARL calculated by using laser range and DEM elevation data (label Laser+DEM and green line)

In Figure 6-15 compares the related errors of the three ARLs calculated with the above explained methods and the true simulated ARL.

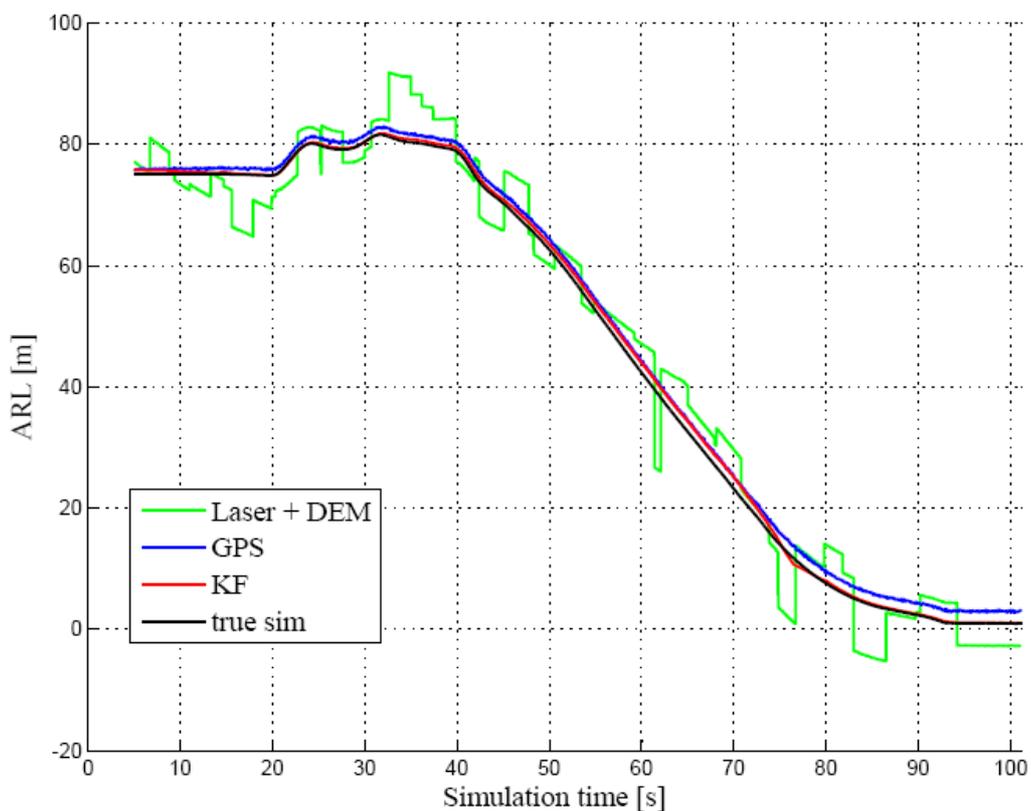


Figure 6-14 – ARL different estimation results

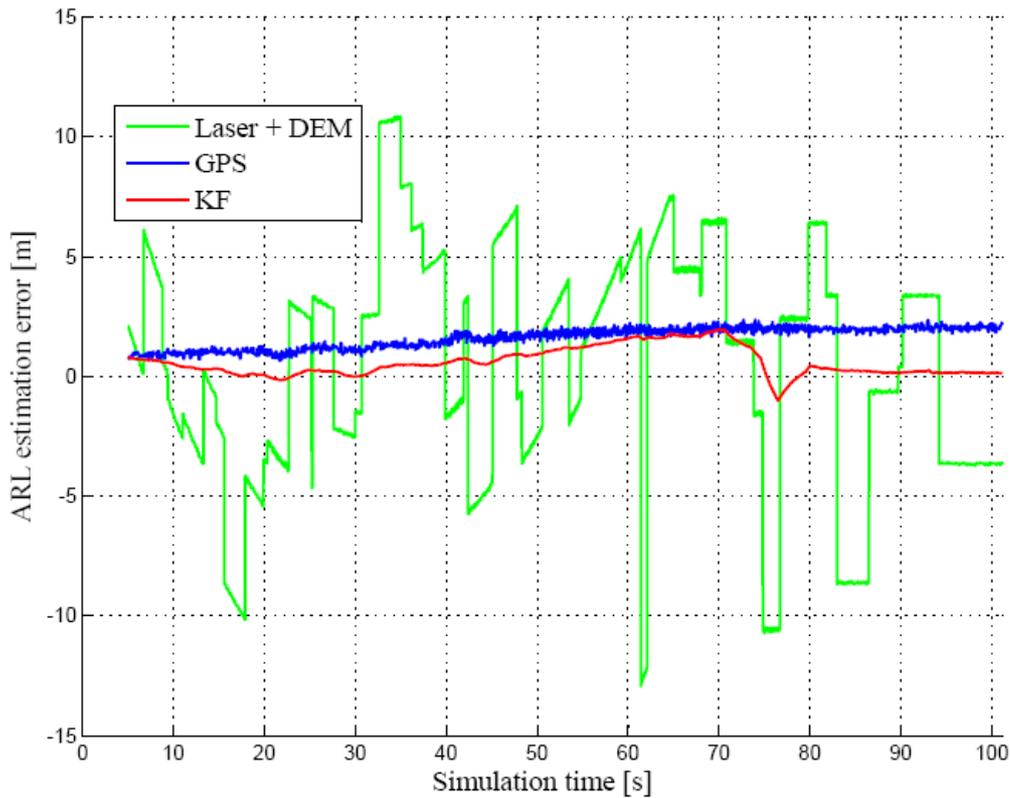


Figure 6-15 – ARL different estimation errors with respect to true simulated ARL

All the algorithms under real-time testing have behaved similarly to numerical simulation, that demonstrates correctness of software implementation and that above described algorithms can be actually executed in a hard real time environment, so enabling flight testing.

6.3 Flight Demonstrations

In this paragraph are presented some results of the real in-flight demonstrations carried by means of the prototypal flight test bed named FLARE (FLYing platform for Aeronautical REsearch) already described in §5.3. These flight tests and related developments have been performed by CIRA in the national funded UAV-TECVOL (Technologies for Autonomous Flight) project. In this framework, CIRA developed and tested in flight a complete autonomous mid-air flight, collision avoidance, take-off and landing system for fixed wing aircrafts. This overall system improves the results and developments of a previous CIRA project ATOL (Automatic Take-Off and Landing), successfully completed in 2004. Autonomous mid-air flight and, partially, autonomous landing capabilities of this system have been already tested, up to experimental flight validation [AR11][AR12].

Firstly is presented a flight demonstration of the free flight trajectory generation algorithm described in §3.2. The line-of-sight tracking algorithm is implemented in cascade to the trajectory generation algorithm so to send track and vertical velocity to FLARE autopilot. The algorithm is used together with other purely geometrical trajectory generation algorithms and it is employed for targeting the first way-point starting from an arbitrary position or to re-target the next way-point after some failures that can result in big trajectory displacements.

The flight tests are all performed in the following way. After a manual take-off, the on board safety pilot switches aircraft control to FCS that executes a pre-programmed flight plan (a sequence of way-points). After reaching the final way-point the safety pilot take over aircraft control, execute some manoeuvres and the test can be repeated in different initial conditions. In some tests, some way-points can be changed or added with respect to the predefined flight plan. The proposed algorithm for free flight generation and line-of-sight tracking are only executing for reaching the first way point of the sequence, starting from an arbitrary point in the air. Below some results for two of such flight tests are reported.

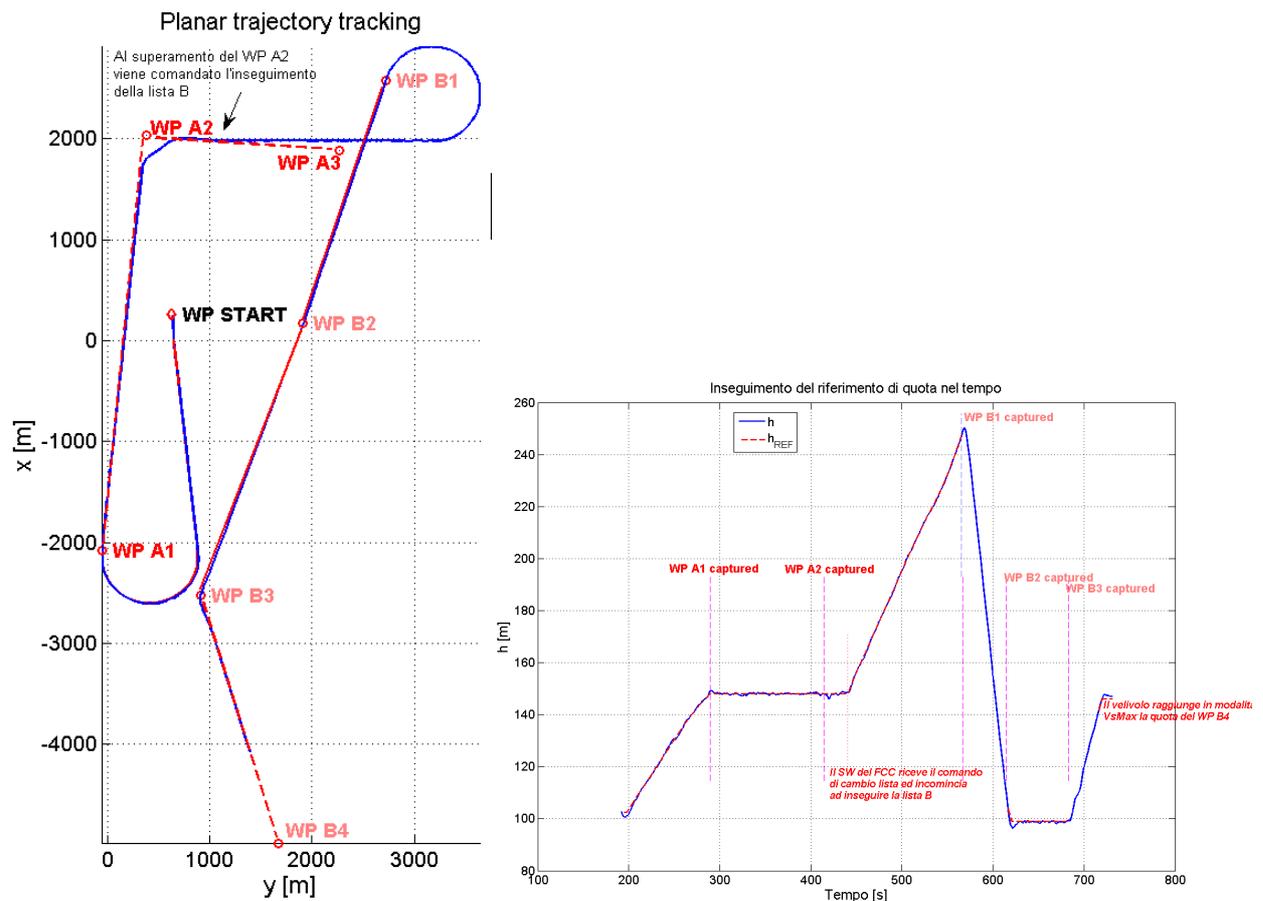


Figure 6-16 – Flight Test of Free Flight Planning algorithm (trajectory segment from WP START to WP A1)

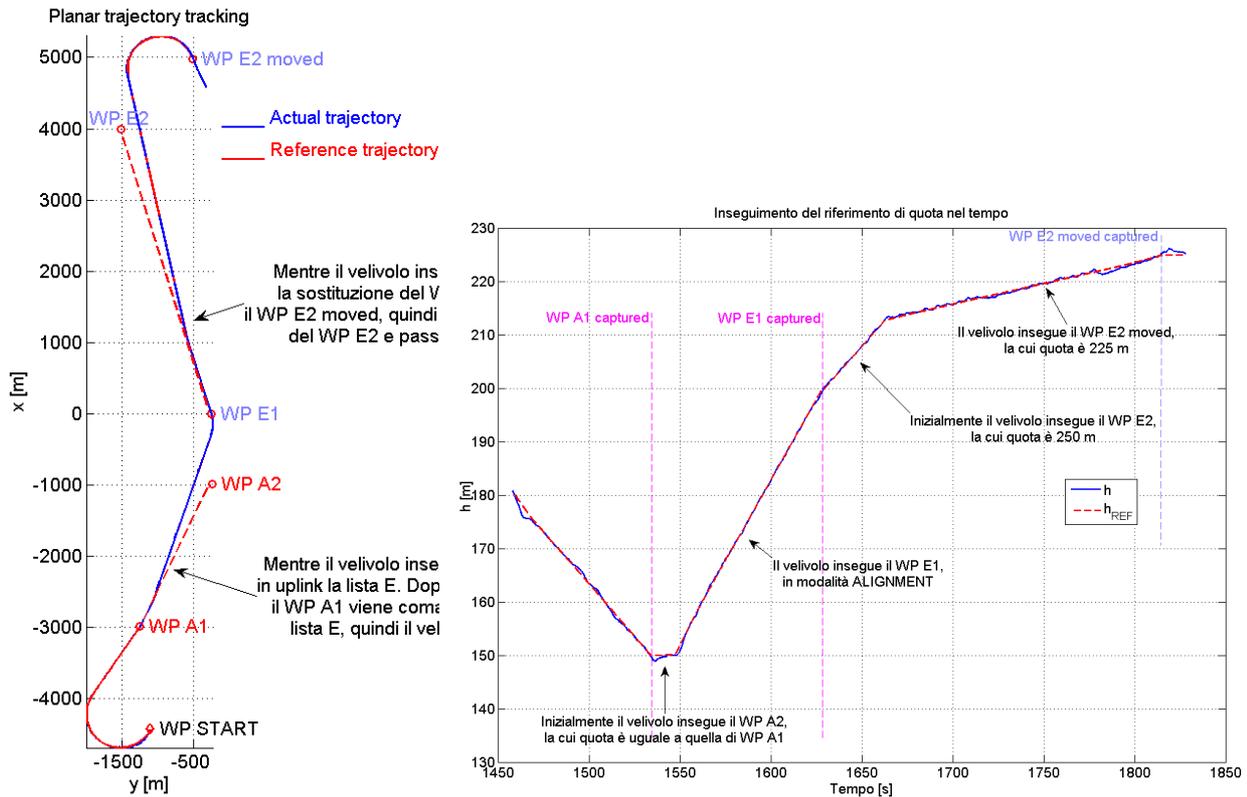


Figure 6-17 – Flight Test of Free Flight Planning algorithm (trajectory segment from WP START to WP E1)

The real in-flight validation of the automatic approach and landing system has been planned in three steps, as described below (the first two steps have successfully been performed, while the third step is already planned for the current year):

- Step 1: validation of the algorithm using a pre-fixed nominal trajectory for the flight control law references generation. The related results have already been described in [AR14][AR15].
- Step 2: validation of the algorithm $Vzlin$, already described in §4.3.3, executed twice at the start of the Flare phase and at the start of the pre-touch down phase. The related results will be shown in the following of the current of this chapter and will be subject of a publication of the next year [AR5].
- Step 3: validation of the algorithm $Vzlin$, already described in §4.3.3, executed step by step from the start of the flare phase up to the touch down event. The related Laboratory Testing results have been shown in §6.2.2, while the real in-flight testing will be performed in the next future.

The results of the second step emphasize that the proposed system satisfies all the run time constraints, demonstrating its effectiveness in performing the autoland manoeuvre.

In the following figures are presented several variables related to the autoland manoeuvres with focus on the Flare and Pre-Touch down phases, in order to emphasize the performances obtained by the proposed algorithm in a relevant environment. The runway orientation is equal to -124 deg with respect to Nord and the nominal touch down point is $X_{RW} = 75$ m, $Y_{RW} = 0$ m.

Figure 8 shows the vehicle trajectory in the lateral plan during the Flare and Touch down phases, the desired value is the centre of runway (e.g. $Y=0$) and the Y profile is shown even for some seconds after the ground contact.

Figure 6-18 shows the vehicle trajectory in the lateral plan during the Flare and Touch down phases, the desired value is the centre of runway (e.g. $Y=0$) and the Y profile is shown even for some seconds after the ground contact.

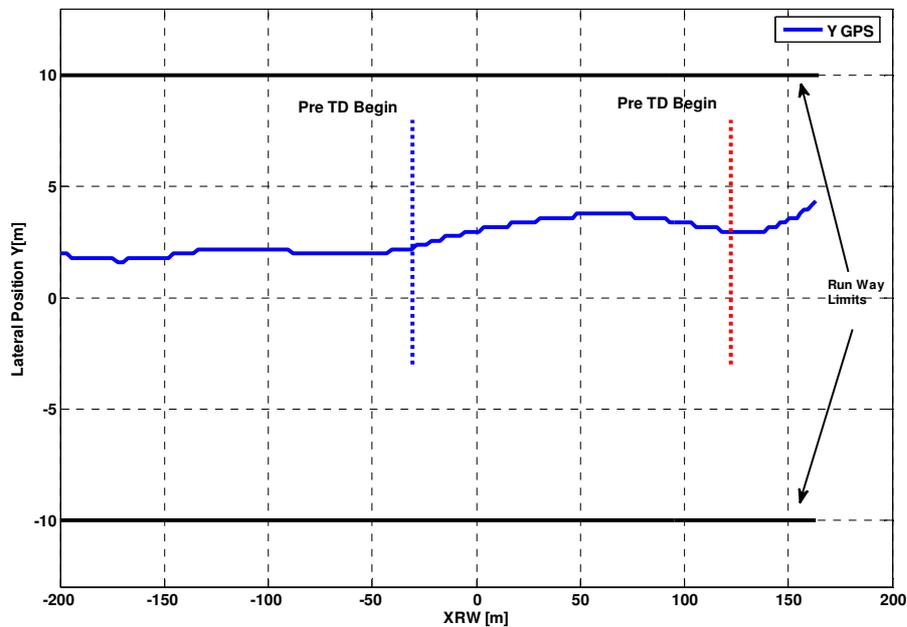


Figure 6-18 – Planar position during Flare and Pre-Touch Down phases

As shown in the figure, during the Pre Touch Down phase, low Y values have been hold in spite of the contemporaneous execution of a Decrab manoeuvre using the rudder control line to achieve a correct heading angle at the touch down. The planar position performances have been satisfied even if the flight control law inner reference for the bank angle ϕ has been limited to 4 deg in magnitude to avoid exceeding the maximum ϕ angle allowed at the touch down (as already described in §4.3.3).

During the Flare phase the *vzlin* algorithm has been executed only once at the begin of the phase and the altitude controller has selected as the most appropriate during this phase. The applied altitude reference is the profile calculated in §4.3.3. Figure 6-19 shows the trend of the altitude

(obtained recording the on-line estimation of the vehicle altitude on the runway) during the last part of the approach up to some seconds after the touch down event.

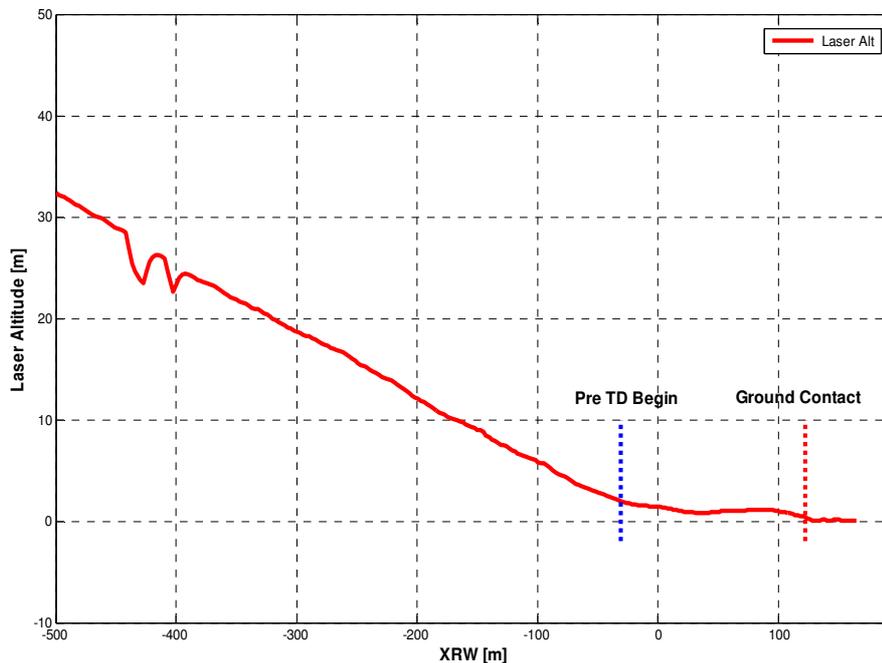


Figure 6-19 – Altitude during Flare and Pre-Touch Down phases

During the Pre-Touch Down phase the *vzlin* algorithm has been executed once more and the vertical speed controller has been selected as the most appropriate during this phase with the aim to reach the desired touch down performances not only in terms of position and velocity but also in terms of proper attitude and orientation.

Figure 6-20 shows the profile of the vertical speed during the Flare and the tracking of the vertical speed (feedback and its related control reference) during the Pre-Touch Down phase. The flight test, in this case, was performed in windy condition with a sever level of turbulence that caused the showed oscillations along the nominal manoeuvre profile for the vertical speed. Anyway the touch down value of the vertical speed was in the allowed performance range.

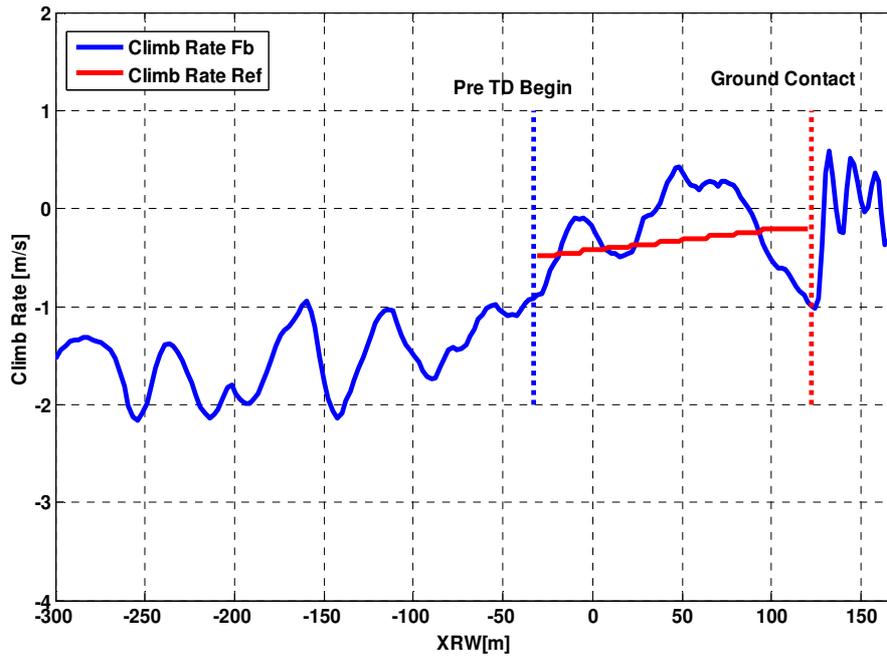


Figure 6-20 – Vertical speed during Flare and Pre-Touch Down phases

Finally for what concerns the autoland trajectory generation and tracking, Figure 6-21 shows the TAS (feedback and its related control reference) and inertial velocity (green curve) of the vehicle during the Flare and Pre-Touch Down phases.

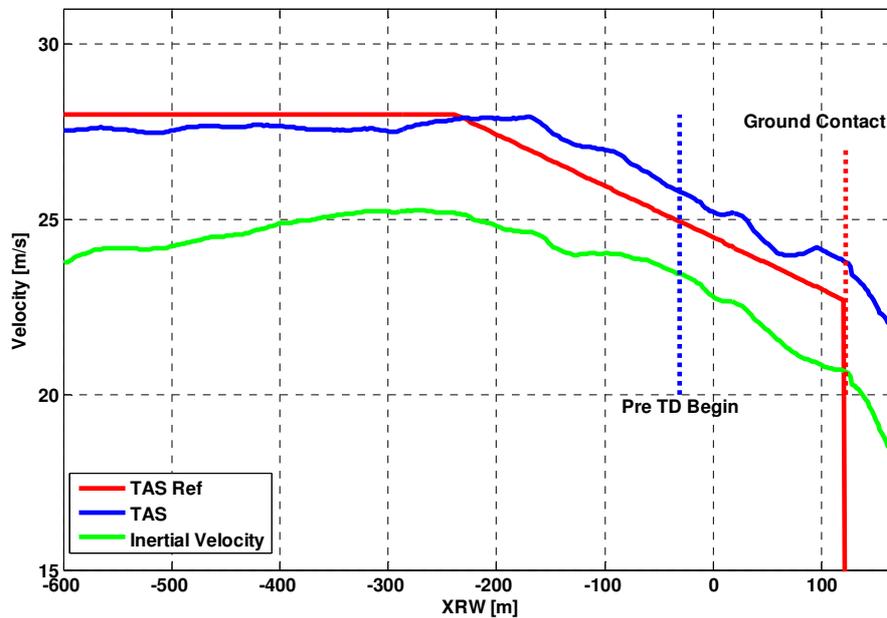


Figure 6-21 – TAS and total inertial speed during Flare and Pre-Touch Down phases

Regarding the positioning estimation algorithm (described in §4.5.1) the complementary filter developed is able to delete the frequency content due to a sudden GPS precision loss. This is shown in Figure 6-22 which refers to a GPS precision loss case experienced during real flights (as confirmed by increasing value of GPS error estimation).

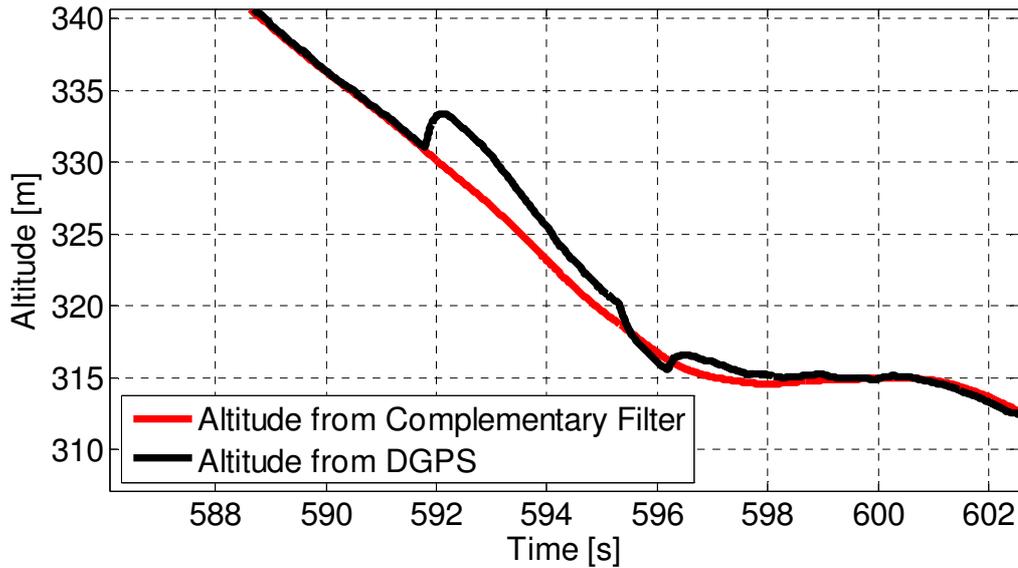


Figure 6-22 – Comparison among altitude measures obtained by complimentary filter and GPS during real flight in case of GPS precision loss

Finally as flight testing results of emergency procedures and algorithms (described in §4.2) Figure 6-23 shows a test of an approach and landing manoeuvre executed with a virtual touch down point objective placed at a height of 100 m above the runway and with a simulated vertical integrity event during the ramp phase.

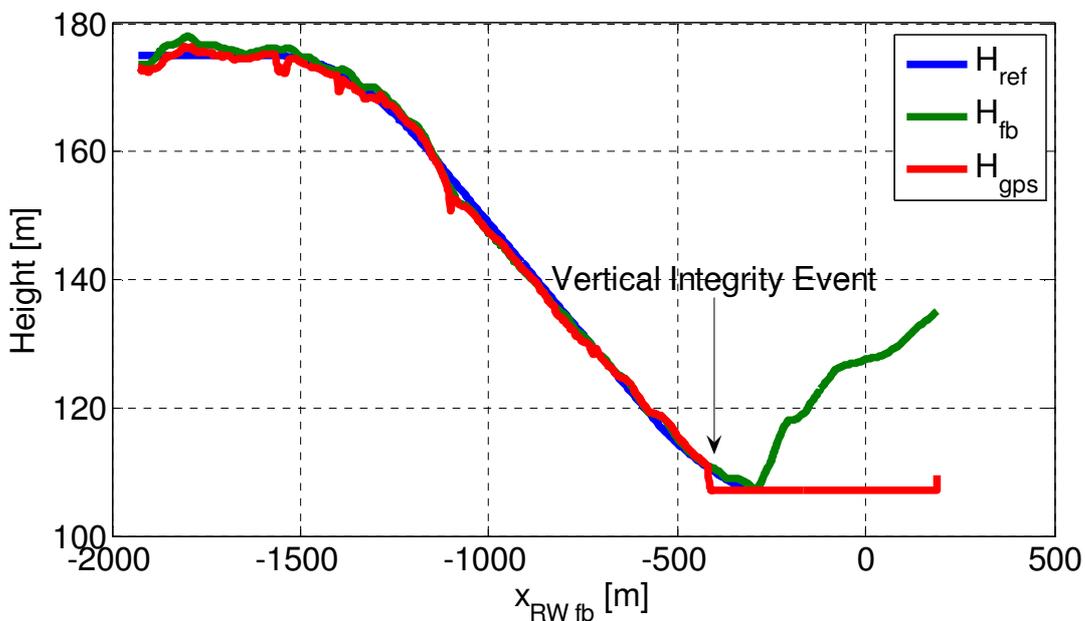


Figure 6-23 – Altitude recovery manoeuvre caused by a Vertical Integrity Event (VPL)

7 CONCLUSIONS

In the research framework related to expand autonomy for the UAVs (Unmanned Aerial Vehicles) with the aim to face unstructured environment this thesis presented some innovative algorithms related to the autonomous approach and landing system. Here “autonomy” refers to the absence of human intervention, and “unstructured environment” is associated with uncertainty both in the outside world (meteorological conditions, air traffic, fixed and moving obstacles) and in the vehicle subsystems (failures).

In the development of the proposed autoland system several algorithms have been integrated. In fact developing such kinds of systems with a desired level of reliability, accuracy and safety involves an evolution of all the subsystems related to the guide, navigation and control disciplines.

In particular, the thesis has dealt with the guidance problem of an autoland manoeuvre taking into account all the different flight phases and segments involving the management of very different flight envelopes and aerodynamic configurations during the same mission. Two algorithms were proposed to generate an optimal trajectory starting from the top of descent (final waypoint of the cruise phase) towards the way point aligned with the runway (initial way point of the proper approach phase) and a further algorithm was presented with capability of on-line generating the flare trajectory up to the touch down point adapting it to the actual state of the vehicle. The benefits of the proposed system with respect to the state of the art concern the adaptivity of the trajectory generating and tracking process to unpredicted external events, such as varied environmental conditions and unexpected threats to avoid, achieved without involving technologies implying missed compliance with the general guidelines imposed by certification authorities.

Moreover in the thesis two sensor fusion algorithms for inertial positioning and above runway level (e.g. the altitude of the vehicle above the runway) estimation was presented exploiting a low cost navigation sensor suite but satisfying the strict performance and reliability requirements imposed for landing manoeuvres by both civil and military competent authorities. The algorithm aiming inertial position and velocity estimation was based on the complementary filter technique preferred to a Kalman filtering approach due to its simplicity and independency by specific sensors used. Regarding the above runway level estimation an innovative idea was proposed developing an algorithm capable to fuse satellite, inertial and DEMs measures achieving accuracy better than that obtainable with the single sensors and with the further aim to make the

system capable to tolerate a single failure to one of these sensors. For the ARL estimation the Kalman filtering has been selected as the optimal sensor fusion method due to a lack of valid alternatives for this particular purpose.

Finally, effectiveness of each of the proposed algorithms has been singularly demonstrated using numerical simulations, HW-in-the-loop real time simulations and flight testing in relevant scenarios. All the algorithms were developed and validated at CIRA (Italian Research Aerospace Center) in the framework of TECVOL project by the Guide, Navigation and Control Laboratory team with the support of the author belonging to the same laboratory during his PhD period.

Further development of the proposed guidance algorithm during flare phase regards the application also to this segment of an idea already developed for other mid-air phase applications ([B1]) and consisting of divide on-line trajectory generation in long and short term paths. The long term trajectory would satisfy mission requirements and path constraints (no-fly zones), while considering only approximately vehicle and static constraints. The short term trajectory would minimize displacements to the long term one keeping into account exactly vehicle dynamic and static constraints.

Finally, even if much work is still to be performed to finalize the proposed autoland system as a future commercial product applicable to the emerging and growing UAVs market, the presented results let us conclude that work performed for this thesis can be considered a promising improvement towards increased UAVs autonomy and safer and reliable autonomous approach and landing operations.

8 AUTHOR REFERENCES

- [AR1] M. Ariola, F. Corraro, E. De Lellis, G. Ambrosino, et alii, *Path Generation and Tracking in 3D for UAVs*, IEEE Transactions on Control Systems Technology, vol. 17, no. 4, July 2009
- [AR2] E. De Lellis, F. Corraro, G. Morani, V. Di Vito, *On-Line Trajectory Generation for Autonomous Unmanned Vehicles in the presence of No-Fly Zones*, accepted for publication on Journal of Aerospace Engineering (2012)
- [AR3] F. Corraro, L. Garbarino, V. Di Vito, E. De Lellis, C. Marrone, *A Sensor Architecture for High Precision UAS Navigation in the Future GNSS Framework*, Journal Coordinates Magazine, Volume 5, Issue 4, pagg.14-17, April 2009
- [AR4] De Lellis E., Corraro F., Garbarino L., Canzolino P., Ciniglio U., *An EGNOS Based Navigation System for Highly Reliable Aircraft Automatic Landing*, ENC-GNCSS 09 Conference, Naples, 4 May 2009
- [AR5] E. De Lellis, F. Corraro, V. Di Vito, C. Marrone, U. Ciniglio, *Flight Testing of a Fully Adaptive Algorithm for Autonomous Fixed Wing Aircrafts Landing*, submitted to AIAA Infotech@Aerospace Conference (I@A), California, June 2012
- [AR6] N. Genito, C. Marrone, E. De Lellis, L. Garbarino, *Autonomous Take Off System: Development and Experimental Validation*, XXI Congresso Nazionale AIDAA – III CEAS Air&Space Conference, Venezia, Ottobre 2011
- [AR7] N. Genito, C. Marrone, E. De Lellis, *A Real-Time Landing Gear Simulation Model for a Very Light Aircraft: Development and Experimental Validation*, AIDAA, Milano, Luglio 2009
- [AR8] E. De Lellis, D. Accardo, F. Corraro, G. Greco, et alii, *Exploiting forward looking radar measurements and digital map data sensor fusion for altimetry estimation during low-altitude flight*, AIAA Infotech@Aerospace Conference (I@A), St. Louis, Missouri, March 2011
- [AR9] De Lellis E., Marrone C., Di Vito V.; *TECVOL: Sviluppo e Testing Real Time del Software Free Path DGPS/AHRS Autolanding*, CIRA-CF-06-1213, CIRA Internal Technical Report

- [AR10] Morani G., De Lellis E., Di Vito V., *Progetto AFMS: Descrizione degli algoritmi Enhanced di pianificazione della traiettoria*, CIRA-CF-09-1300, CIRA Internal Technical Report
- [AR11] V. Di Vito, E. De Lellis, *TECVOL: Validazione in volo su FLARE del SW di AMF per l'inseguimento di una lista di way-point 3D aggiornabili on-line*, CIRA-CF-09-1567, CIRA Internal Technical Report
- [AR12] E. De Lellis, *TECVOL: Validazione in volo su FLARE del Modulo SW Free Path DGPS/AHRS Autolanding*, CIRA-CF-07-1206, CIRA Internal Technical Report
- [AR13] C. Marrone, E. De Lellis, *TECVOL: Sviluppo e Testing real Time del SW SCAS/Autopilot Multivariabile*, CIRA-CF-08-1353, CIRA Internal Technical Report
- [AR14] V. Di Vito, E. De Lellis, C. Marrone, U. Ciniglio, F. Corraro, *UAV Free Path Safe DGPS/AHRS Approach and Landing System with Dynamic and Performance Constraints*, UAV Systems 2007 International Conference & Exhibition, Paris, France, 12-14 June 2007
- [AR15] V. Di Vito, E. De Lellis, C. Marrone, Genito N., U. Ciniglio, F. Corraro, *UAV Free Path Safe DGPS/AHRS Autolanding: algorithms and flight tests*, UAS 2008 International Conference & Exhibition, Paris, France, 10-12 June 2008

9 BIBLIOGRAPHY

General References

- [B1] F. Corraro, “On-Line Trajectory Planning Algorithms for Unmanned Aerial Vehicles” , Phd Thesis, University Parthenope of Naples, 2010
- [B2] M. Patcher, P.R. Chandler, *Challenges of Autonomous Control*, IEEE Control System Journal, August 1998
- [B3] L. E. Dubins, “*On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents*” Amer. J. Math., vol. 79, no. 3, pp. 497–516, Jul. 1957
- [B4] Office of Secretary of Defense, *Unmanned Aerial Vehicle Roadmap 2002-2027*, Dicembre 2002
- [B5] B. Etkin, *Dynamics of Atmospheric Flight*, John Wiley & Sons, Inc.
- [B6] R. P. G. Collinson, *Introduction To Avionics*, Chapman & Hall
- [B7] Cary R. Spitzer, *Digital Avionic Systems: Principles and Practices*, McGraw-Hill, inc.
- [B8] National Transportation Safety Board, “*Aircraft Accident Report*”, 1997
- [B9] Eurocontrol Navigation Sub-Group, *Navigation Application & NAVAID Infrastructure Strategy for the ECAC Area up to 2020*, Ed.2, may 2008.
- [B10] SESAR Definition Phase, Deliverable 3, *The ATM Target Concept*, DLM-0612-001-02-00a, September 2007

Path Optimization Algorithms References

- [B11] G. Yang and V. Kapila, “*Optimal path planning for unmanned air vehicles with kinematic and tactical constraints*” in *Proc. 41st IEEE Conf. Decision Control*, Las Vegas, NV, 2002, pp. 1301–1306.
- [B12] H.Wong, V. Kapila, and R.Vaidyanathan, “*UAV optimal path planning using C-C-C class paths for target touring*” in *Proc. 43rd IEEE Conf. Decision Control*, Bahamas, 2004, pp. 1105–1110.

- [B13] E. P. Anderson and R. W. Beard, “An algorithmic implementation of constrained extremal control for UAVs” presented at the AIAA Guid., Nav., Control Conf. Exhibit, Monterey, California, 2002.
- [B14] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, “Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming” *AIAA J. Guid., Control, Dyn.*, vol. 25, no. 4, pp. 755–764, 2002.
- [B15] T. Schouwenaars, “Safe trajectory planning of autonomous vehicles” Ph.D. dissertation, Dept. Aeronautics Astronautics, Massachusetts Inst. Technol., Boston, 2006.
- [B16] C. Dever, B. Mettler, E. Feron, J. Popovic’, and M. McConley, “Nonlinear trajectory generation for autonomous vehicles via parameterized maneuver classes” *AIAA J. Guid., Control, Dyn.*, vol. 29, no. 2, pp. 289–302, 2006.
- [B17] E. Frazzoli, M. A. Dahleh, and E. Feron, “Real-Time motion planning for agile autonomous vehicles” *AIAA J. Guid., Control, Dyn.*, vol. 25, no. 1, pp. 116–129, 2002.
- [B18] C. L. Bottasso, D. Leonello, and B. Savini, “Path planning for autonomous vehicles by trajectory smoothing using motion primitives”, *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1152–1168, Nov. 2008.
- [B19] T. I. Fossen, M. Breivik, and R. Skjetne, “Line-of-Sight path following of underactuated marine craft” in *Proc. 6th IFAC MCMC*, Girona, Spain, 2003, pp. 244–249.
- [B20] Latombe, J. C. “*Robot Motion Planning*”, Kluwer Academic Press, 1990
- [B21] Gu, D. W., Postlethwaite, I. and Kim, Y., “A Comprehensive Study on Flight Path Selection Algorithms”, IEE Seminar on Target Tracking: Algorithms and Applications, Birmingham, UK, 7-8 March 2006
- [B22] LaValle, S. M., “*Planning Algorithms*”, Cambridge University Press, 2006.
- [B23] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. E. and Thrun, S., “*Principles of Robot Motion: Theory, Algorithms, and Implementations*”, Cambridge, MA: MIT Press, 2005

- [B24] Hwang, Y. K., and Ahuja, N., “*A Potential Field Approach to Path Planning*” IEEE Transactions on Robotics and Automation, Vol. 8, No. 1, 1992, pp. 23–32, doi:10.1109/70.127236
- [B25] Kavralu, L. E., Svestka, P., Latombe, J., and Overmars, M. H., “*Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces*”, IEEE Transactions on Robotics and Automation, vol. 12, no. 4, August 1996
- [B26] Belghith, K., Kabanza, F., Hartman, L., and Nkambou, R., “*Anytime Dynamic Path-Planning with Flexible Probabilistic Roadmaps*”, Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida - May 2006
- [B27] Kothari, M., Postlethwaite, I., and Gu, D., “*Multi-UAV Path Planning in Obstacle Rich Environments Using Rapidly-exploring Random Trees*”, IEEE Conference on Decision and Control 2009
- [B28] Amin, J. N., Boskovic, J. D., Mehra, R. K., “*A Fast and Efficient Approach to Path Planning for Unmanned Vehicles*” Proceedings of the AIAA Guidance, Navigation and Control Conference, Keystone, CO, August 2006
- [B29] Pettersson, P.O. and Doherty, P., “*Probabilistic Roadmap Based Path Planning for an Autonomous Unmanned Aerial Vehicle*”, Proceedings of the Workshop on Connecting Planning and Theory with Practice, ICAPS 2004.
- [B30] Omar, R. and Gu, D.-W., “*Visibility line based methods for UAV path planning*”, In ICROS-SICE International Joint Conference. ICROS-SICE 2009
- [B31] Huang, H. and Chung, S. (2004), “*Dynamic visibility graph for path planning*”, IEEE Proceeding of International Conference on Intelligent Robots and Systems, 2004
- [B32] Reif, J. H., “*Complexity of the Mover’s Problem and Generalizations*”, 20th IEEE Symposium on the Foundations of Computer Science, Inst. of Electrical and Electronics Engineers, New York, 1979, pp. 421–427
- [B33] Jorris, T. R., and Cobb, R. G., “*Multiple Method 2-D Trajectory Optimization Satisfying Waypoints and No-Fly Zone Constraints*”, AIAA Journal of Guidance, Control and Dynamics, Vol. 31, No. 3, May-june 2008

- [B34] Luongo, S., Corraro, F., Ciniglio, U., and Di Vito, V., “*A Novel 3D Analytical Algorithm for Autonomous Collision Avoidance considering Cylindrical Safety Bubble*”, IEEE Aerospace Conference 2010, March 6-13, 2010, Big Sky, Montana, USA

Automatic Landing Systems

- [B35] S. M. B. Malaek, Hojjat A. Izadi, Mehrdad Pakmehr, “*Flight Envelope Expansion in Landing Phase Using Classic, Intelligent, and Adaptive Controllers*”, Journal Of Aircraft, Vol. 43, No. 1, January–February 2006
- [B36] Juang J.-G., Lin B.-S., Chin K.-C., “*Automatic Landing Control Using Particle Swarm Optimization*”, Proceedings of the 2005 IEEE International Conference on Mechatronics, Taipei, Taiwan, July 10-12, 2005
- [B37] Juang J.-G., Chin K.-C., Chio J.-Z., “*Intelligent Automatic Landing System Using Fuzzy Neural Networks and Genetic Algorithm*”, Proceeding of the 2004 American Control Conference, Boston, Massachusetts, June 30 – July 2, 2004
- [B38] Attar M., Wahnou E., Chaimovitz D., “*Advanced Flight Control Technologies for UAVs*”, 2nd AIAA “Unmanned Unlimited” Systems, Technologies And Operations, 15-18 September 2003, San Diego, California, AIAA 2003-6537
- [B39] Riseborough, P., BAE Systems, “*Automatic Take-Off and Landing Control for Small UAV's*”, 5th Asian Control Conference, 2004, Vol.2, pagg. 754 – 762, DOI: 10.1109/ASCC.2004.184961
- [B40] Jategaonkar R., Behr R., Gockel W., Zorn C., “*Data Analysis of Phoenix RLV Demonstrator Flight Test*”, AIAA Atmospheric Flight Mechanics Conference and Exhibition, San Francisco, California, 15-18 August 2005
- [B41] Gregg H. Barton, Steven G. Tragesser, “*Autoland Trajectory Design for the X-34*”, AIAA Atmospheric Flight Mechanics Conference and Exhibit, Portland, Oregon, August 9-11, 1999
- [B42] Walyus, K.D. and C. Dalton, “*Approach and Landing Simulator for Space Shuttle Orbiter Touchdown Conditions*”, Journal of Spacecraft and Rockets, Vol. 28, No. 4, 1991, pp. 478-485.

- [B43] Paulo Rosa, Carlos Silvestre et alii (Instituto Superior Tecnico), “*Autolanding controller for fixed wing UAV*”, AIAA Guidance, Navigation and Control Conference and Exhibit, 20 - 23 August 2007, South Carolina
- [B44] Shakernia, O., Vidal, R., Sharpy, C. S., Ma, Y., and Sastry, S., “*Multiple View Motion Estimation and Control for Landing an Unmanned Aerial Vehicle*”, ICAR 2002, USA.
- [B45] ICAO Annex 10 Volume 1, Radio Navigation Aids, Fifth Edition – July 1996.
- [B46] Fang Liao, Jian Liang Wang, Eng Kee Poh, Dong Li, “*Fault-Tolerant Robust Automatic Landing Control Design*”, Journal Of Guidance, Control, and Dynamics Vol. 28, No. 5, September–October 2005
- [B47] Mclean D., “*Automatic Flight Control Systems*”, Prentice–Hall, Englewood Cliffs, NJ, 1990.
- [B48] Shue, S. P., and Agrawal, R. K., “*Design of Automatic Landing Using Mixed H_2/H_∞ Control*”, Journal of Guidance, Control, and Dynamics, Vol. 22, No. 1, 1999, pp. 103–114.
- [B49] Izadi, H. A., Pakmehr, M., and Sadati, N., “*Optimal Neuro-Controller in Longitudinal Autolanding of a Commercial Jet Transport*”, Proceedings of the 2003 IEEE Conference on Control Applications (CCA03), Vol. 1, 2003, pp. 492–497.
- [B50] Juang, J. G., Chang, H. H., and Cheng, K. C., “*Intelligent Landing Control Using Linearized Inverse Aircraft Model*”, Proceedings of the 2002 American Control Conference, Vol. 4, 2002, pp. 3269–3274.
- [B51] Malaek, S. M. B., Izadi, H. A., and Pakmehr, M., “*Autolanding Controller Design Using Adaptive Control*”, Advances in Dynamics, Instrumentation and Control, 1st ed., edited by C. Y. Su, S. Rakheja, E. Wang, and R. B. Bhat, World Scientific, Nanjing, PRC, 2004, pp. 188–199.

Sensor Fusion Algorithms

- [B52] Greg Welch, Gary Bishop, “*An introduction to the Kalman Filter*”, Department of Computer Science, University of North Caroline at Chapel Hill, Chapel Hill, NC.

- [B53] Kaminer, I., Yakimenko, O., Dobrokhodov, V., and Jones, K., “*Rapid Flight Test Prototyping System and the Fleet of UAV’s and MAVs at the Naval Postgraduate School*”, AIAA-2004-6491, AIAA, 3rd “Unmanned Unlimited” Conference, Chicago IL, 2004.
- [B54] Kingston, D., and Beard, D., “*Real-Time Attitude and Position Estimation for Small UAVs Using Low-Cost Sensors*”, AIAA-2004-6488, AIAA, 3rd “Unmanned Unlimited” Conference, Chicago IL, 2004.
- [B55] Johnson, E., Schrage, D., Prasad, J., and Vachtsevanos, G., “*UAV Flight Test Programs at Georgia Tech*”, AIAA-2004-6492, 3rd “Unmanned Unlimited” Conference, Chicago IL, 2004.
- [B56] Savage, P., “*Strapdown Analytics*”, Strapdown Associates Inc., Minneapolis MN, 2002, Chap. 15.
- [B57] Farrell, J.A., and Barth, M., “*The Global Positioning System & Inertial Navigation*”, McGraw Hill Professional, New York NY, 1998, pp 135-139, pp 241-257.
- [B58] Grewal, M.S., Weill, L.R., and Andrews, A.P., “*Global Positioning System, Inertial Navigation and Integration*”, John Wiley & Sons, New York NY, 2002, pp 14-37, pp 103-130.
- [B59] *GALILEO – Integration of EGNOS Council Conclusion*, n. 9698/03 (Press 146), 5.6.2003.
- [B60] Integration of EGNOS programme in the GALILEO Programme, Communication from the Commission to the European Parliament and the Council, COM (2003) 123 Final, March 19, 2003.
- [B61] George C. Clark and J. Bibb Cain, *Error Correction Coding for Digital Communications*, Plenum Press, New York, 1981.
- [B62] RTCA Special Committee SC-159, *Recommended Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment*, RTCA/DO-229C, Washington, D.C.: RTCA, November 28, 2001.
- [B63] Pieter Bastiaan, *SBAS Integrity Concept*, Eurocontrol, Issue 1.0, Reelektronika b.v., June 2001

- [B64] G. Greco, “*Progetto di un algoritmo di sensor fusion per la stima della quota su pista di un velivolo*”, Master Degree Thesis, University of Naples Federico II, 2011

DEMs state-of-the-art

- [B65] Jung Hum Yu, Linlin G, Chris Rizos, “*Digital elevation model generation from interferometric synthetic aperture radar using multi – scale method*”, Jung School of Surveying and Spatial Information Systems, University of the New South Wales, Australia.
- [B66] Farr et alii, “*The Shuttle Radar Topography Mission*”
- [B67] E. Rodriguez, C. S. Morris, J. E. Belz, E. C. Chapin, j. M. Martin, W. Daffer, S. Hensley, “*An assessment of the SRTM topographic products*”, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, NASA.
- [B68] Michael Abrams and Simon Hook, Bhaskar Ramachandran, “*ASTER User Handbook Version 2*”, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, EROS Data Center, Sioux Falls, SD.
- [B69] Bryan Mercer, “*Countrywide Coverage of RADAR DTMs – The Intermap Approach*”, <http://www.intermap.com/>.
- [B70] M. Lorraine Tighe and Drew Chamberlain, Doug King, Keiko Balzter, Heather McNairn, “*Accuracy Comparison of the SRTM, ASTER, NED, NEXTMAP USA Digital Terrain Model over several USA study sites*”, Intermap Technologies.
- [B71] M. Lorraine Tighe, “*Comparison of canopy height models derived from SRTM/NED and NEXTMAP USA elevation data*”