UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II"

FACOLTÀ DI INGEGNERIA

Dottorato di ricerca in "Ingegneria Aerospaziale, Navale e della Qualità" – XX Ciclo

Settore scientifico disciplinare ING-IND/05:
Impianti e Sistemi Aerospaziali

Multisensor based Fully Autonomous Non-Cooperative Collision Avoidance System for UAVs

---

Ing. Giancarmine Fasano

Tutor:

Ch.mo Prof. Ing. Antonio Moccia

Coordinatore:

Ch.mo Prof. Ing. Antonio Moccia

"The pursuit of truth and beauty is a
sphere of activity in which we are
permitted to remain children all our lives"
Albert Einstein

*(*La ricerca della verità e della bellezza
rappresenta una sfera di attività in cui ci è
permesso di rimanere bambini per tutta la vita*)*

Ai miei genitori, con affetto

Abstract

Unmanned Aerial Vehicles will have a safe access to the Civil Airspace only when they will be able to avoid collisions even with non cooperative flying obstacles. Thus, they need to replace the capability of human eye to detect potential mid-air collisions with other airframes and the pilot experience to find an adequate avoidance trajectory.

This thesis deals with development and test of a fully autonomous system devoted to avoidance of non cooperative intruders. In particular, it focuses on sensors, and processing logics and hardware, required on the unmanned system to acquire situational awareness. The study was carried out in collaboration with the Italian Aerospace Research Center within a research project named TECVOL, funded in the frame of National Aerospace Research Program. The performed activities covered all the steps in the development process from the analysis of requirements deriving from the application, to the real time implementation of designed logics.

Designed prototype system is based on a multi-sensor architecture with a Ka-band pulsed radar as the main sensor, and four electro-optical cameras as aiding sensors. Proper logics and algorithms for real time sensor fusion have been developed, tested in off-line simulations, and later implemented on embedded systems to enable technology flight demonstration.

Numerical results and flight data have shown the potential of the developed system. Also on the basis of the international scenario, this technology demonstration has gained a significant scientific value.

KEYWORDS: Unmanned Aerial Vehicles, Collision Avoidance, Sensor Fusion, Multiple Sensor Tracking, Flight Demonstration

Acknowledgements

Here is my opportunity to thank some people who helped me during this adventure…
First of all, thanks to all the Aerospace Systems Research Group at the university of Naples. For me, it has been an honor and a pleasure to be part of this team during my Ph.D. course.
In particular, thanks to prof. Antonio Moccia, my tutor, for giving me the possibility to work on the sense and avoid subject, and for his genuine care for research and for young people who love research.
Thanks to Dr. Domenico Accardo, who has worked with me on project TECVOL, for his fruitful collaboration and precious help.
Thanks to prof. Marco D'Errico, who gave me the possibility to follow also another line of research. But above all, thanks for being a guide and a real friend since I graduated.
Thanks to prof. Michele Grassi for showing me how to be a good researcher and a genuine person.
Thanks to Dr. Giancarlo Rufino, for his friendship and support.
Thanks to Dr. Carlo Del Vecchio Blanco and Dr. Francesco Esposito. I shared with them all the time spent in laboratory, which was great because we also share the passion for aerospace research… and for jokes. Staying in lab with them was not work (mostly)!
Thanks to my parents and my brother for their love and encouragement during these years, and special thanks to my grandmother, for making me understand that picking grapes and olives is fun!
Thanks to all my friends in Battipaglia, who are for me an extended family. They deserve a medal for tolerating me for the last three years… .
Thanks to Alessandro, to whom I paid a breakfast when I was admitted to the Ph.D. course. It was simply great to share the Fasano house in Naples!
Thanks to Irma, who made this journey special.

Finally, thanks to my grandfather, whose life has been the best example for me. Sempre in gamba, nonno.

Contents

Introduction

I.1. Unmanned Aerial Vehicles in the Civil Airspace: the Sense and Avoid problem

In recent years, Unmanned Aerial Vehicles (UAVs) have experimented a great worldwide diffusion. Several institutions and manufacturers are developing or operating unmanned vehicles which cover a large range of dimensions, performance, and payload mass: from university aero-models to large aircrafts such as the Northrop Grumman™ RQ-4B Global Hawk™ (figure 1) and the General Atomics™ Predator-B™ (figure 2).

Figure 1 – Northrop Grumman RQ-4B Global Hawk

Figure 2 – General Atomics Predator-B

While the current use of these systems is mainly in the military field, there has also been a growing interest in using UAVs for civil purposes. UAVs are in fact perfectly suited for many applications: their potential uses include from border and coastal surveillance, to drug interdiction, to checking the status of oil pipelines and power systems, to environmental surveillance and support during emergencies, crop assessment (farming), automotive traffic surveillance, and sky based communication networks.

While autonomous control and payload technologies can be considered mature, one basic problem hinders the use of UAVs in civil scenarios: it is represented by flight safety in terms of collision risk with respect to other aircrafts.

The need to overcome this problem led to a remarkable increase in worldwide research for integration of UAVs in the Civil Airspace [1-4]. The most important organizations for standard development were involved in writing the rules to allow UAVs for a safe access to flight [5, 6]. This effort was sponsored by FAA and was developed by important regulatory agencies such as ASTM and RTCA. Also the European Union with project USICO [2], Japan, and Australia were involved in this field of research.

One of the basic principles of these analyses was that introduction of UAVs does not have to modify airspace regulations which already apply to manned aircrafts: otherwise, the cost to be paid (also in economic terms) in order to allow manned and unmanned aircrafts to coexist would be too high. In a few words, UAVs have to coexist with manned aircrafts, and not vice-versa.

As a consequence of this approach, Federal Aviation Administration (FAA) 7610.4 regulation [7], stated that UAV flight in the Civil Airspace is allowed only if it guarantees "…*an equivalent level of safety, comparable to see-and-avoid requirements for manned aircraft*", both in the controlled and in the uncontrolled airspace.

What is meant by equivalent level of safety will be clarified in the following. However, it is important to note since now that in order to satisfy the requirement, UAVs have to be endowed with two capabilities: a situational awareness capability on the one hand (sense), a decision making capability on the other hand (avoid).

The solution to the "sense and avoid" problem can be considered today as the real key to enable massive usage of unmanned aircrafts for civil applications, thus allowing for a revolution in unmanned aviation (figure 3).

Figure 3 – Aerospace America, June 2006

This thesis will focus on the situational awareness aspect of the problem.

I.2. The CIRA TECVOL project

All the study reported in this thesis has been conducted in the framework of project TECVOL (Technologies for Autonomous Flight) carried out by the Italian Aerospace Research Center (CIRA).

CIRA holds the Italian Aerospace Research Program (PRORA) funded by the Italian Government. This program involves the realization of a High-Altitude Long-Endurance (HALE) UAV. Within this program, the TECVOL project aims at the development and flight demonstration of the technologies needed to support the HALE UAV flight autonomy, and will realize a hardware/software prototype that integrates the following functions:

- Autonomous Flight Path Execution;
- Autonomous Approach and Landing;
- Obstacle Detect See & Avoid (DS&A);
- Autonomous Runway Search and Lock;
- Enhanced Remote Piloting.

Regarding the sense and avoid function, the Department of Aerospace Engineering (DIAS) of the university of Naples “Federico II” has been in charge of developing and testing the anti-collision sensing system and logics.

The system prototype will be initially installed onboard a manned laboratory aircraft equipped for automatic control so that flight tests will verify the adequacy of attained performances for supporting fully autonomous flight. The optionally piloted laboratory aircraft is a Very Light Aircraft (a TECNAM P-92) and has been named FLARE, which means Flying Laboratory for Aeronautical Research. It is shown in figure 4.

Figure 4 – FLARE at landing

Airframe technical specifications are reported in table 1:

TECHNICAL SPECIFICATIONS	
Wing span	8,70 m
Length	6,40 m
Height	2,50 m
Wing Area	12 m2
Engine Maximum Power at sea level	73,5 kW
Propeller	GT - 172/164
MTOW	450 Kg

Table 1 – TECNAM P-92 specifications

FLARE is equipped with a general hardware configuration that includes:

- A set of navigation sensors (Attitude and Heading Reference System, Laser Altimeter, Standalone GPS, and Air Data Sensor);
- Sensors and Processing Units for Obstacle DS&A;
- Electro-optical sensors for enhanced Remote Piloting;
- A Flight Control computer.

Some of these systems are shown in figure 5.

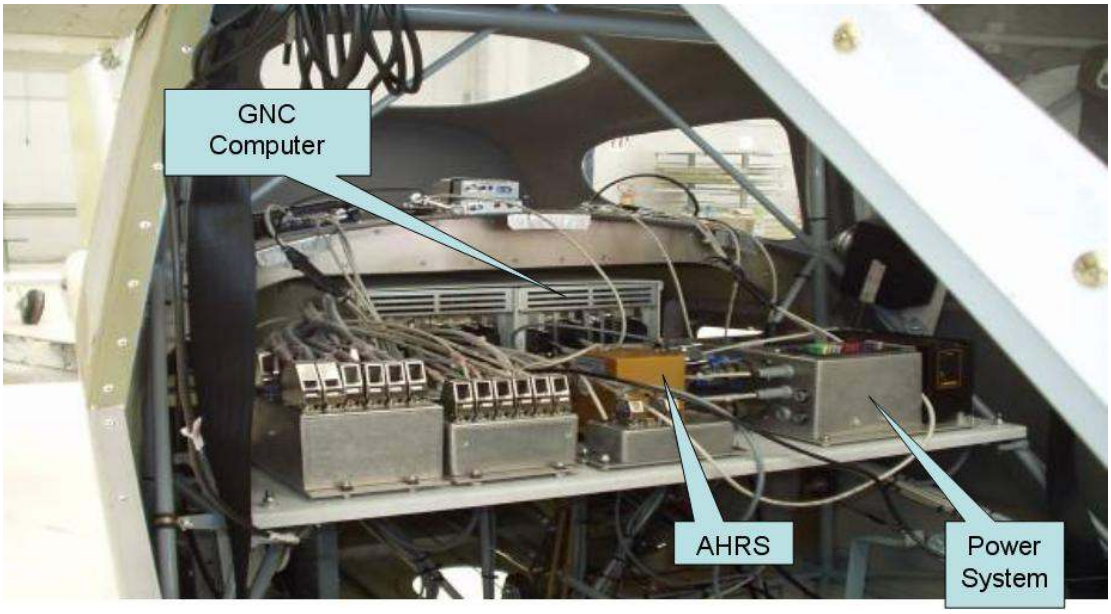

Figure 5 – FLARE internal view

Sense and avoid planned flight tests will include sessions where a single intruder will enter the Field of Regard (FOR) of obstacle detection sensors. Initial tests will verify the capability of the designed system to detect and track the intruder, in different approaching geometries and weather/illumination conditions. Subsequently, real Collision Avoidance tests will be performed.

I.3. Thesis objectives and outline

The main objective of this thesis is to develop and test a prototype of a fully autonomous system devoted to avoidance of non cooperative flying obstacles. This requires the identification of a logical/hardware architecture to satisfy sensing requirements, the development of proper algorithms for real time fusion of the measurements from all the sensors needed to reach situational awareness, and the implementation of these algorithms on embedded systems to enable technology flight demonstration. All these activities are detailed in the thesis, on the basis of the following outline.

Chapter 1 clarifies in more detail the sense and avoid problem by pointing out the consequences of the concept of "equivalent level of safety". Then, advantages and drawbacks of different possible sensor choices and architectures are pointed out, and a brief survey

covers methods and results of sense and avoid experiences performed in the United States. The last section of the chapter is dedicated to a study on detection algorithms based on electro-optical sensors only. This study was conducted to explore potential and limitations of this approach to the sense and avoid problem.

The designed anti-collision system is treated in chapter 2. In particular, starting from requirements, selected sensors are described. Then, different possible data fusion techniques are analyzed in view of the constraints deriving from the application, and the logical architecture of the system is clarified, together with the hardware system that implements these logics.

Data fusion algorithms are detailed in chapter 3 with particular regard to the tracking module, which can be considered as the core of the whole anti-collision system. First, basics of multi-target tracking techniques are described. Then, the analysis is focused on state estimation algorithms to be used for obstacle relative position and velocity. A numerical analysis allows to point out main differences of different algorithm choices in terms of accuracy and reliability for collision avoidance.

Chapter 4 reports a detailed numerical survey which was performed to estimate system performance in realistic collision scenarios. A proper software environment for off-line simulation was built to enable these analyses. The different scenarios are briefly described, then numerical results are reported which allow to estimate system accuracy in different phases, and to understand the main effects of navigation errors on tracking.

The second part of the thesis is devoted to real time implementation. In particular, ground and on-board software architecture and design is clarified in chapter 5. Different functions are implemented by different operating systems and programming languages. Hardware in-the-loop tests are described which regard on-board data handling and real time performance of developed system.

Collision avoidance flight tests have not been performed yet, due to project delays related to other functions. However, some experimental activities have already been carried out. In

particular, a procedure has been designed to align the electro-optical sensors with the inertial unit. It is described in chapter 6, together with the results obtained in test campaigns.

Preliminary flights for functional verification have also been performed. Relevant results and flight images from these tests are reported in chapter 7.

Finally, conclusions and further research are pointed out in the last chapter of the thesis.

Chapter 1

Sense and Avoid

1.1. Autonomous obstacle detection as a prerequisite for integration of UAVs in the Civil Airspace

FAA requires that unmanned aerial systems guarantee an equivalent level of safety with respect to manned aircrafts, both in the controlled and in the uncontrolled airspace.
In the former, safety is guaranteed at a higher level by procedures and air traffic control systems. Moreover, every airliner has a TCAS (Traffic Alert and Collision Avoidance System) [8], which provides a highly reliable detection sensor for cooperative threats. TCAS is based on a transponder and transmits to the other aircrafts own-ship position. Moreover, it has a high grade of autonomy, since it is able to perform conflict detection and resolution by assigning proper avoidance maneuvers in the vertical plane. It is worth noting that the choice of vertical avoidance maneuvers is strictly dependant on the fact that TCAS range and altitude precision is relatively high, while there is a large bearing angle noise from directional antenna construction.

TCAS is required for all the airliners and it is generally installed on-board general aviation aircrafts. Of course, TCAS can not guarantee collision avoidance with respect to aircraft which do not carry it (which happens if uncontrolled airspace is considered), or in case of system failures. Thus, the last level of safety against collision is the human pilot capability to "see and avoid" other aircrafts, or more generally other flying objects (figure 1). On the basis of this line of reasoning, requiring an equivalent level of safety for UAVs means that an onboard system for autonomous obstacle Detect, Sense and Avoid (DS&A) has to be considered as mandatory. Of course, the non cooperative "see & avoid" function is complementary to collaborative anti-collision systems such as TCAS or ADS-B [9], also in view of an application of autonomous obstacle detection systems as auxiliary systems for manned aircrafts. In fact, research efforts are being carried out also on the integration between cooperative and non cooperative systems [10]. For example, this integration can give better bearing accuracy with respect to TCAS, thus allowing for different choices in avoidance maneuvers.

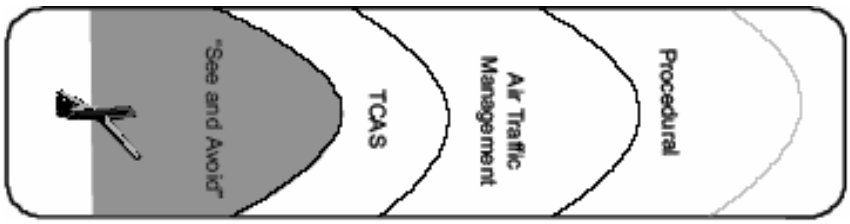

Figure 1 – Flight safety levels

At present, UAV autonomous anti-collision systems are at experimental level [11-15] and research studies are still being carried out to find out system requirements and sensing solutions. The trend is to consider different requirements and to develop a customized Obstacle DS&A system for each UAV category [16].

1.2. Sensing solutions and architectures for DS&A

In general, different sensing architectures have been selected to be adopted onboard manned and unmanned aerial vehicles for implementing autonomous collision avoidance capabilities. In this framework, multi-sensor fusion is considered as a valuable tool.
Basically, two sensor types are investigated:

A. Electro-Optical (EO) Sensors in the Visible (VIS) or Infrared (IR) bands [17,18];
B. Microwave (MW) sensors [13,19] from L-band to Millimeter Wavelenghts;

EO sensors have some peculiarities, such as:

a) Fast scan rate (10 Hz or more);
b) Fine Angular resolution < 0.1°;
c) Low cost;
d) Small size and weight;
e) Low electric power consumption. They are passive sensors;
f) Reduced sensitivity to rain.

However, EO sensors also have also several disadvantages with respect to radars:

a) No range-to-obstacle information is provided;
b) Infrared sensors must be installed to guarantee night flight coverage;
c) Detection range can be much shorter than radars. The requirements on time-to-collision can not be met;
d) Target detection performance is strongly dependant on background;
e) Large sensitivity to fog;
f) The resulting sense and avoid system is intrinsically non all time all weather.

Some additional considerations are needed on EO sensors drawbacks. As for point a), range is clearly unobservable on the basis of angular-only measurements. However, range to obstacle can be estimated on the basis of a maneuver performed by the UAV and the consequent navigation measurements. The technique is known in literature as passive ranging collision avoidance [20] and is based on an Extended Kalman Filter as state estimation algorithm. However, it has some relevant issues which are connected to the time needed for the filter to give a reasonably accurate range estimation. Moreover, the maneuver requested for range estimation could last too much because of the aerodynamic characteristics of the UAV. For example, the Global Hawk has a configuration which does not allow for agile maneuvers.[20] Moreover, it must considered that for increasing resolution and thus pixel number, the computational load of detection algorithms increases.

A configuration composed by sole EO sensors (in one or more bandwidth) can be considered adequate when limited economical and/or dimensional resources are available, for example in the case of mini-UAVs. In the most compact configuration, it is composed by one or more cameras in the visible waveband and a fast Image Processing CPU that operates a search in the whole Field-Of-View (FOV) to extract elevation and azimuth angles of candidate obstacles and relevant angular rates.

In these cases, the detection process can be carried out in different ways, according to typical operational conditions: in case of uniform or quasi uniform background (sky, large clouds, sea, etc.), which mainly occurs in the high-medium altitude region, or in very large and clean areas (flight over sea or over extended plains), in theory see-and-avoid algorithms can operate on single frames, adopting thresholding criteria. Typical techniques are global threshold, local adaptive threshold, segmentation by hue [21-24]. Instead, in case of non uniform or cluttered background, which mainly occurs in the low-medium altitude region, or during nap-of the-earth flight over woody or bushy landscapes, these algorithms must operate on sequences of frames, and according to contrast or better correlation criteria, or to Optical Flow methodology [25-28].

As shown in the following section, up to now systems based on this technology have suffered from high levels of false or missed detections [18,29]. This is basically due to the fact that though optical flow is a well-assessed technique for image tracking, it has challenging detection issues to be solved, in order to be applied in the considered application. They concern the type of background of images that include the obstacles to be avoided. In particular, the Optical Flow algorithm is affected by the following background effects:

- Sensitivity to aggressive attitude maneuvers;
- Sensitivity to scene illumination conditions, even for obstacles that have the sky on their background;
- Sensitivity to weather conditions;
- Sensitivity to ground when the obstacle flies at an altitude that is lower then the one of the aircraft that is equipped with the cameras.

Another important point of discussion related to optical flow-based detection algorithms is that frame rate has to be very high for reliable detection, which can pose hard requirements on the processing system. Moreover, the heavy computational weight of these techniques

hinders their implementation in real time. These are some of the reasons why a purely EO approach to sense and avoid has been discarded in TECVOL. However, additional studies have been carried out on detection algorithms for an EO-only sensor system [30], and will be described in the last section of this chapter.

In many senses radars are complementary sensors with respect to EO cameras. In fact, they provide direct range estimate, and also range rate if Doppler processing is used. Moreover, they can guarantee larger detection range, low levels of missed or false detection, and are much less affected by weather conditions: the all-time all-weather requirement can be satisfied. On the other hand, radars can hardly provide an angular resolution comparable to EO cameras: finer resolution essentially implies larger antenna dimensions. Moreover, in general radars are demanding in terms of cost, size, weight and required electric power.

Other possible solutions are laser radar active systems, which provide at the same time range information and fine spatial accuracy. However, these systems are very limited by fog and if excessive power requirements have to be avoided, unacceptably low revisit rates are obtained.

Considering radar and EO cameras, the shortcomings of the two sensors can be compensated by combining their data, so as to obtain a hybrid multi-sensor configuration. Of course, this configuration is more demanding in terms of onboard and economic resources than a purely EO sensor suite, but in theory it allows for all-weather and all-time coverage, low levels of missed or false detections, large detection range, high revisit rate and fine angular resolution. Moreover, the resulting system is intrinsically very reliable, because the different sensor answers are related to completely different physical phenomena [31].
The two configurations (purely EO, integrated EO/radar) are illustrated conceptually in figure 2. Both require a tracking software to be implemented thus allowing for proper obstacle dynamics estimation. This software should be capable to manage multiple obstacles in the system Field-Of-Regard (FOR). The tracking algorithm in the second architecture should be capable to handle sensor data fusion by properly using measurements from integrated sensors. In this case, two approaches can be pursued:

- *High-level sensor fusion.* The two systems generates estimates of obstacle dynamics independently from each other. Subsequently, a proper real-time algorithm operates high-level measurements fusion;

– *Hierarchical architecture*. The system is composed by a "chief" sensor, i.e. the radar, and by the EO sensors that are "aiding" sensors. In this case, the "chief" sensor directs the search in the "aiding" sensors FOV and their measurement are used to refine obstacle estimates.

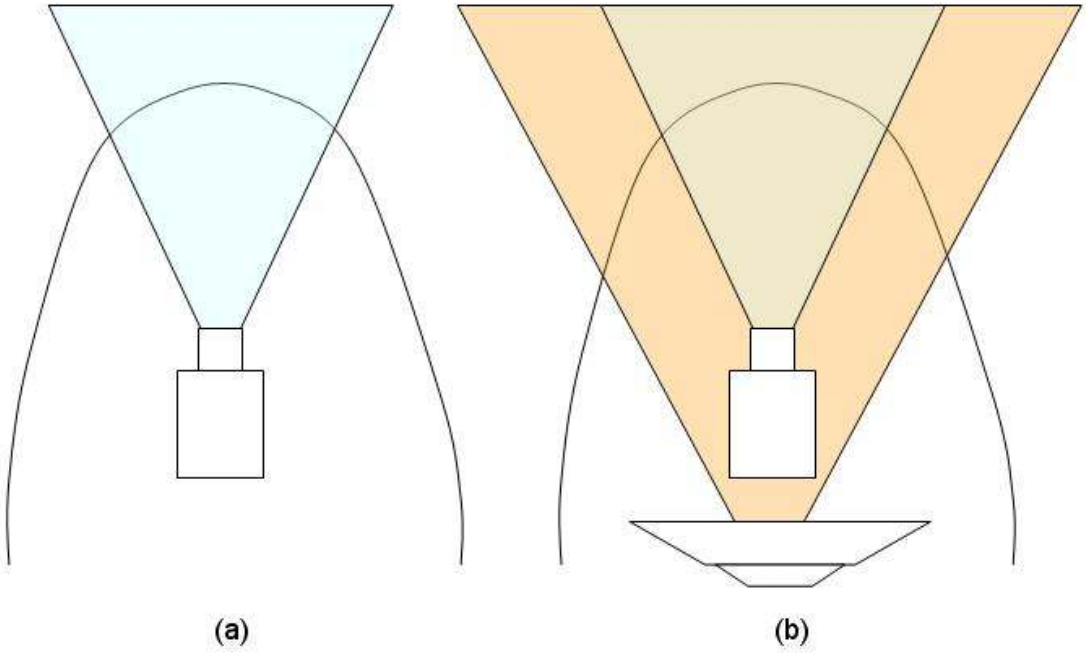

Figure 2 - Baseline sensing configurations for DS&A systems: (a) sole EO and (b) integrated EO and radar

1.3. International experiences and research

Sense and Avoid is currently a very active area of research, both from theoretical and experimental point of view, as it is also demonstrated by the large number of papers and conference sessions on this theme [10-12, 15-18, 20, 32-38]. In particular, in the United States there is a flurry of research carried out by different industries and institutions. Experimental campaigns have been carried out by NASA Dryden Research Center, in the framework of ERAST project [13,14] and by Air Force Research Lab, in collaboration with Defense Research Associates™ (DRA) [15,18] and Northrop Grumman™ [10,11,20]. It is interesting to point out the main results of these campaigns also to have a reference point for comparing the designed anti-collision system.

In the NASA-ERAST project, a nose Ka-band pulsed radar was mounted onboard Proteus, an optionally piloted aircraft realized by Scaled Composites™ (figure 3)

Figura 3 – Nose radar on-board Proteus

Besides the radar sensor, a cooperative system was installed on the Proteus and on some of the intruders. Several collision geometries were tested with different types of intruders, ranging from gliders to interceptors. In all these tests, collisions were avoided by maneuvers performed by the pilot on-board or the remote pilot in the ground station. In other words, obstacle detection sensors were used within a man-in-the-loop architecture. Basically, these tests were dedicated to experimental estimation of obstacle detection range. Radar detection range was demonstrated to be adequate for avoidance maneuvers except in the case of very fast intruder, i.e. F-18 Hornet.

As for AFRL-DRA tests, they were based on a purely EO sensor system, made by three visible cameras in order to cover a 220° FOV in azimuth. The cameras were first mounted on the nose of a manned aircraft used as a surrogate UAV (figure 4). Detection was performed by optical flow-based methods: thus, a complex processing system had to be used to sustain the high frame rate and computational weight of detection algorithms. The basic result of the flight tests was that while missed detections were avoided, multiple false tracks were generated. In particular, the algorithm suffered because of small bright clouds (figure 5).

Anyway, detection performances were estimated as adequate for collision avoidance requirements [18].

Figure 4 – DRA experimental setup (from [18]

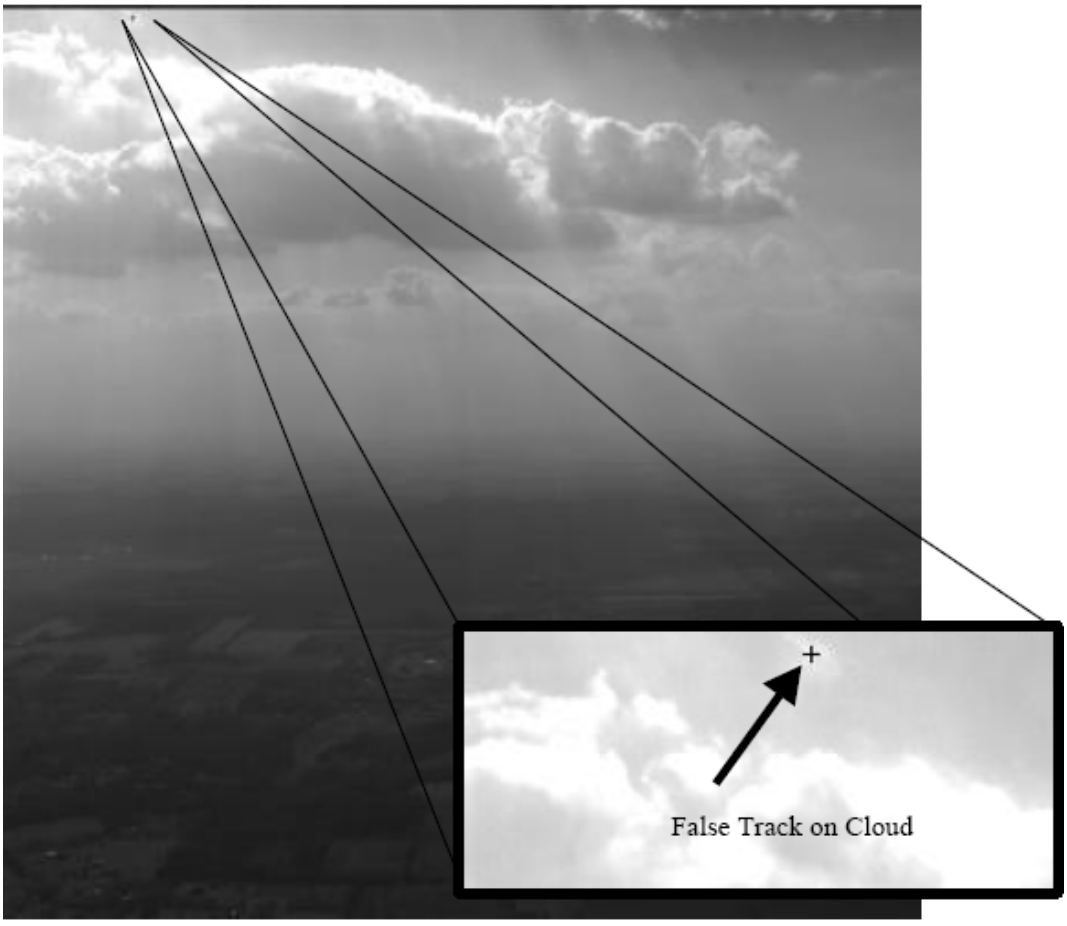

Figure 5 – Cloud phenomenology inducing false tracks (from [18])

Subsequent tests were carried out on Predator UAV. Excessive false alarms were reported which caused "the acknowledged failure" [29] of the system. Infact, "…Intended as a near-term sense-and-avoid solution for USAF MQ-1 Predator UAVs, the system was, says Colonel Larry Felder, commander of the USAF UAV Battlelab, a failure. It did not work. We had multiple false targets." [29] .

Integrated sense and avoid flight tests activities are now being carried out on the Aerostar UAV. Nevertheless, the final goal seems now to be an integrated radar/EO solution [39].

Finally, Northrop Grumman studies have been directed towards multiple sensor configurations and data fusion. In particular, most recent studies and experiments focused on the issues in the integration of TCAS and EO sensors [10,11].

1.4. Study on detection algorithms for an Electro-Optical based sense and avoid system

Though, as it will be shown in the following chapter, a radar/EO multisensor architecture was selected in TECVOL for installation on FLARE, a study was carried out on obstacle detection techniques for EO sensors, in order to better understand potential and limitations of this approach to sense and avoid [30]. In particular, the analysis dealt with optical flow-based methods, which can be used for detection both in the case of uniform and non-uniform background. The performed study is detailed in this section.

Optical flow is the 2-d motion field, which is the perspective projection onto the image plane of the true 3-D velocity field of moving surfaces in space [25], arising from the relative motion of objects and the viewer. Moving objects can be separated from the background by analyzing this motion [26]. It is worth noting that Optical flow only requires electro-optical sensors measurement to evaluate kinematical behavior of objects (with respect to the background), without any need of navigation system measures.

In particular, two techniques were implemented and tested in this study, which are the most used according to literature [27]: the Horn and Schunck's (HS) [28] and the Lucas and Kanade's (LK) algorithms [27]. Both of them are based on differential solving schemes. In other words, they compute image velocity from numerical evaluation of spatiotemporal derivatives of image intensities. The image domain is consequently assumed to be differentiable in space and time. The basic assumption in measuring image motion is that the

intensity structures of local time-varying image regions are approximately constant for, at least, a short time duration. This assumption brings to the following condition, known as the "Optical Flow Constraint Equation":

$$\frac{\partial E}{\partial x}u + \frac{\partial E}{\partial y}v + \frac{\partial E}{\partial t} = 0 \tag{1}$$

where E represents intensity, x and y the two spatial coordinates in the image, u and v the corresponding apparent velocity components, and t is time.

This is an under-constrained equation, since only the motion component in the direction of the local gradient of the image intensity function may be estimated: this is known as "aperture problem". Therefore, one more assumption is necessary.

Horn and Schunck's method assumes that the motion field is smooth over the entire image domain. Thus, it computes an estimation of the velocity field [u, v] that minimizes both the sum of the errors for the rate of change of image brightness in eq. (1), and the measure of the departure from smoothness in the velocity flow [28]. Iterative equations are used to minimize equation (1) and thus to obtain velocity field:

$$u^{n+1} = \bar{u}^{n} - \frac{E_x \cdot \left(E_x \bar{u}^{n} + E_y \bar{v}^{n} + E_t\right)}{\alpha^2 + {E_x}^2 + {E_y}^2} \tag{2}$$

$$v^{n+1} = \bar{v}^{n} - \frac{E_y \cdot \left(E_x \bar{u}^{n} + E_y \bar{v}^{n} + E_t\right)}{\alpha^2 + {E_x}^2 + {E_y}^2} \tag{3}$$

where superscripts refer to the iteration number, subscripts refer to derivation, and α is a positive constant known as smoothness factor.

Instead, Lucas and Kanade's method divides the original image into smaller sections, Ω, assuming a constant velocity in each section. Then, it performs a weighted least-square fit of the optical flow constraint equation, to a constant model for [u, v] in each section, by minimizing the following equation:

$$\sum_{x \in \Omega} W^2(\underline{x}) \cdot [E_x(\underline{x},t)u + E_y(\underline{x},t)v + E_t(\underline{x},t)]^2 \tag{4}$$

where W is a weights function which basically gives more importance to the equations written near the center of the considered window Ω.

The developed algorithm consists of mainly two components: an Optical Flow Solver, based on the algorithms described above, and a Target Detector, that selects point objects with a statistically different motion with respect to background. For each pair of adjacent frames, the system computes the global motion, then it extracts potential object pixels, computing their kinematical features. Schematically, autonomous target detection is performed in the following steps:

1. Image pre-processing. Pre-filtering with a spatiotemporal Gaussian filter, it is possible to extract signal structure of interest and to enhance the signal-to-noise ratio. This attenuates temporal aliasing and quantization effects [26]. Temporal smoothing is achieved through a temporal 1-d Gaussian filter, g(t), centered in the considered frame, with standard deviation of 1.5. Then, a 2-d Gaussian filter, G(x,y), performs spatial smoothing. G(x,y) is defined into a domain of 3x3 pixels, with a standard deviation of 1.5.
2. Image pyramid representation. HS and LK differential techniques require image intensity to be nearly linear, with velocities less than about 1 pixel/frame [40]. For this reason, images are decomposed in different scales of resolution, in the form of Gaussian pyramids. The higher levels of the pyramid contain filtered and sub-sampled versions of the original image. Hence, velocity estimate is first produced at lower resolution scale, with velocities less than 1 pixel/frame (that is the so called “Coarse-to-fine strategy”). In this way, the optical flow constraint equation becomes applicable also in the case of large image motions. Unfortunately, the main drawback of this step is its high computational cost.
3. Extraction of spatiotemporal derivatives. Temporal derivatives are computed with a backward scheme. The algorithm computes spatial derivatives using a 5-points finite-difference central scheme for differentiation.
4. Motion field resolution. Depending on the chosen technique, the solver integrates the above equations (2,3) or (4), to produce a 2-d motion field [u, v] (figure 6).

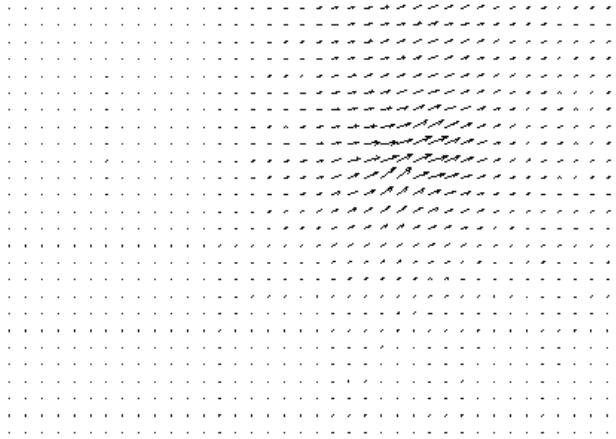

Figure 6 – Motion field resolution

5. Decision step. After solving the 2-d motion field, the algorithm detects object pixels as those with a statistically different velocity from the background. The detection threshold $T_{\underline{V}}$ is chosen dynamically, on the basis of a statistical mean:

$$T_{\underline{V}} = |\underline{V}|_{mean} \pm K\,\sigma \qquad (5)$$

where $|\underline{V}|_{mean}$ is the mean value of velocity distribution, σ is the standard deviation, K is a non-negative coefficient. Velocity of pixels potentially belonging to obstacles are given in output as in figure 7:

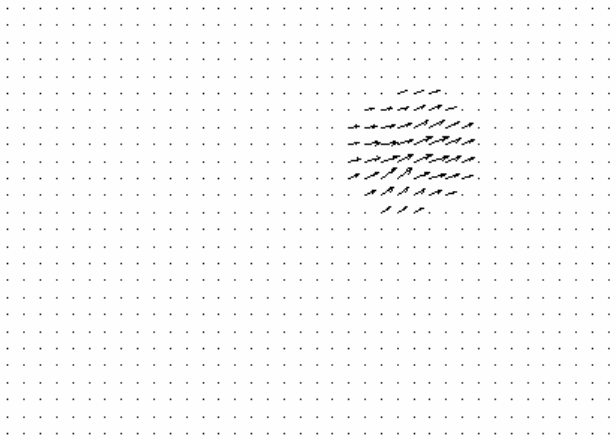

Figure 7 - Decision Step (Threshold Criteria)

6. Observation merging and centroids determination. Nearby object pixels are grouped. Then, for each group, the algorithm calculates centroid coordinates and mean velocity (figure 8).

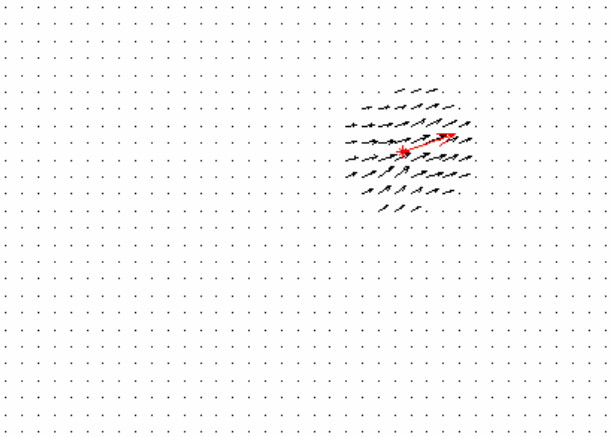

Figure 8 - Object grouping and centroid calculation

Flow diagram of the overall algorithm is shown in figure 9.

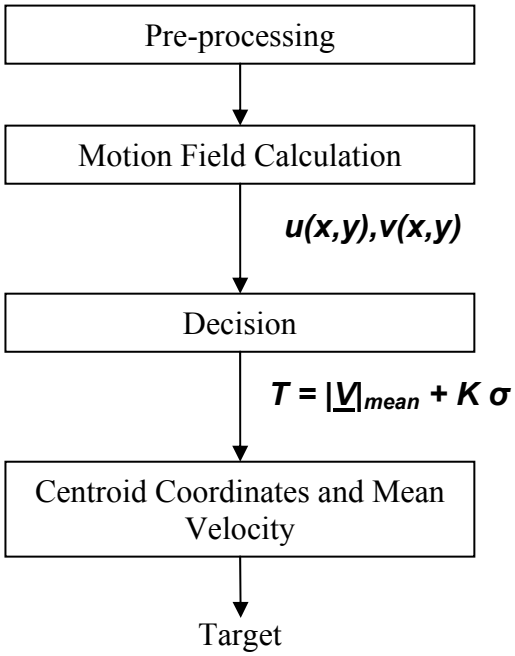

Figure 9 – Detection algorithm flow diagram

In order to have a realistic test environment, developed algorithms were applied on real video sequences of flying aircrafts, acquired in test campaigns which were conducted in Naples,

imaging civil aircrafts approaching Capodichino airport, and in Castel Volturno (CE).The cameras used in these tests are the same that have been later mounted on FLARE for obstacle detection.

The LK technique guaranteed better performances in performed tests. In fact, although it produced a less dense motion field, velocity estimation were, on average, more accurate. Furthermore, it required a constant number of computations for a given sequence, depending only on images dimensions in pixels. Thus, it resulted to be more suitable for real-time implementation, unlike the HS method which is based on an iterative scheme. In this second case it is not possible to a-priori estimate the required number of iterations. In fact, applying HS algorithm to different sequences the number of iterations needed for convergence changes randomly. These results agree with literature suggestions [27]. For this reason it was decided to adopt the LK algorithm for all the other computations.

In the second step of the analysis, the reliability level of the whole algorithm was evaluated, in terms of miss-detection (MD) and false alarm (FA) rate. FA rate is the ratio of the total number of false alarms throughout the sequence to the number of image frames in the sequence. MD rate is the ratio of the number of frames in which the target was missed, to the total number of frames. FA rate depends on the amount and motion of clutter in the images, whereas MD rate depends on the target size and contrast, and therefore, it generally increases with the target distance [41]. Another parameter which was evaluated is the number of object points, N, erroneously extracted by the algorithm (step 5), as a fraction of the number of image frames in the sequence: a high value of this parameter is connected to FA rate and moreover reduces the efficiency of pixel objects grouping (step 6), since the latter requires more calculations.

Considering our tests in Naples, they were performed with good weather and illumination, and in two different conditions: the first one is with fixed background, which was the real experimental situation (figure 10). Then, we simulated (off-line) a motion of the optical axis, in order to reproduce camera wobbling effects.

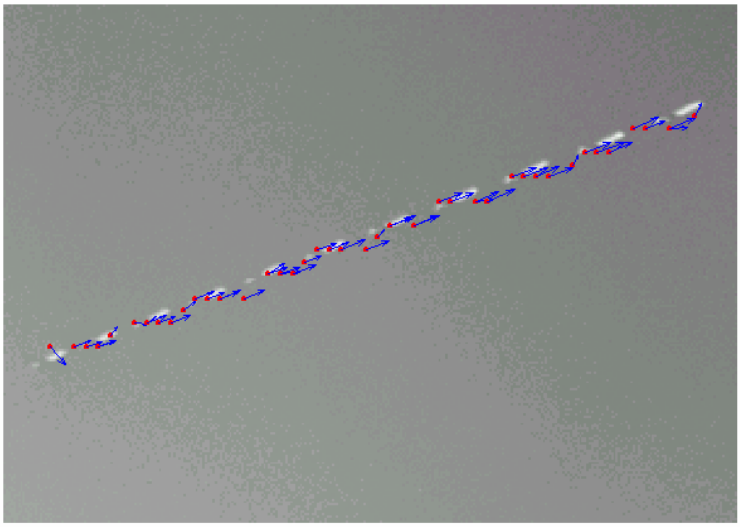

Figure 10- Target Detection on fixed background

Table 1 reports the performance of the detection algorithm on a number of 8 image sequences (1100 frames in total), in the case of fixed background. If a suitable value is chosen for K (equation 5), no MD and FA can be noticed.

MD	FA	N	K
0	3.5	6.6	1.5
0	2.14	3.6	2.5
0	0.76	1.5	5
0	0.34	0.96	7.5
0	0.18	0.4	10
0	0.04	0.08	15
0	0.02	0.02	20
0	0.02	0.02	25
0	0	0	30

Table 1 - Target Detection Algorithm: Miss-Detection Rate (MD), False Alarm Rate (FA), Number of erroneously extracted Object Points (N).

Instead, in the case of moving optical axis, the parameter K was selected in order to keep MD as low as possible, and to reduce the FA rate. Figure 11 shows that increasing K, FA rate is reduced, while MD rate is increased. For K=1.5, zero MD were measured, while, on average, 4.4 FA/frame were observed.

However, it is worth noting that this is not necessarily a suitable choice for every case. In fact, according to [41] characteristic parameters of the algorithm should be adjusted according to factors like the clutter level, or the scene lightening conditions. Then, the effects of increasing oscillation velocity on the algorithm performances were analyzed. As shown in figure 12, it maintains a good level of reliability for an oscillation velocity of 5 pixel/frame: MD is less than 2.5, while FA is less than 4.5.

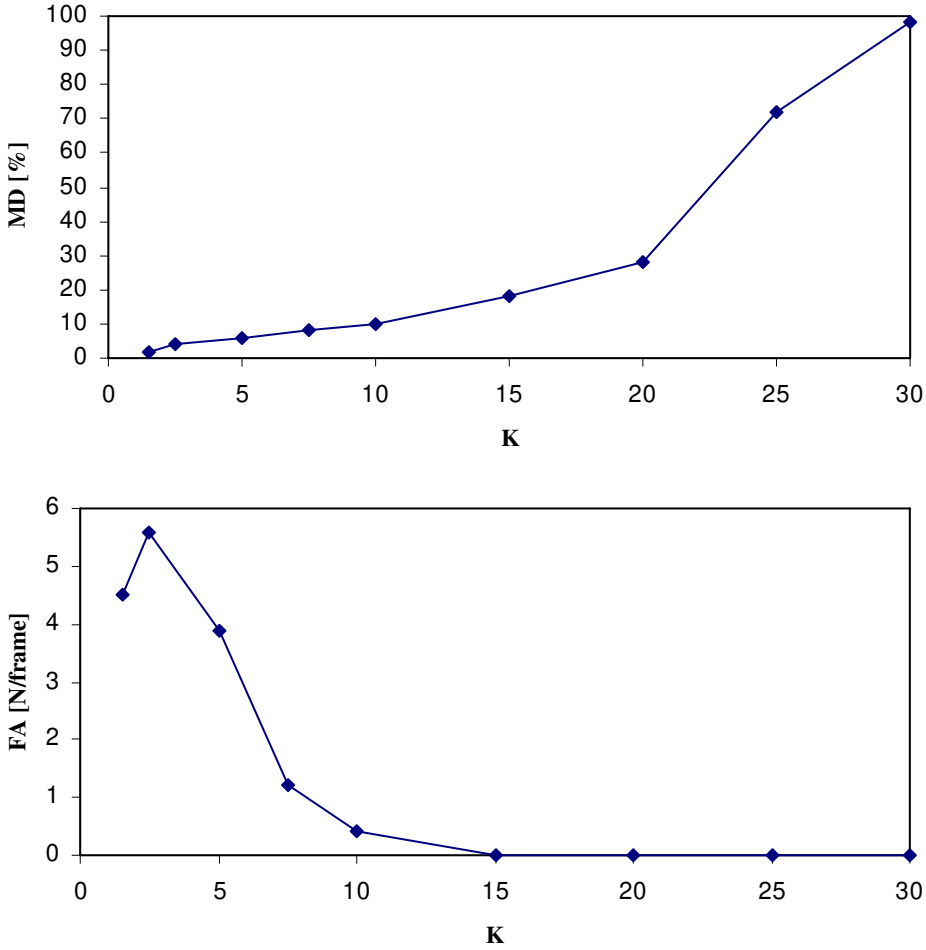

Figure 11 - Optical Axis Oscillation of 1 pixel/frame: FA and MD profiles with K.

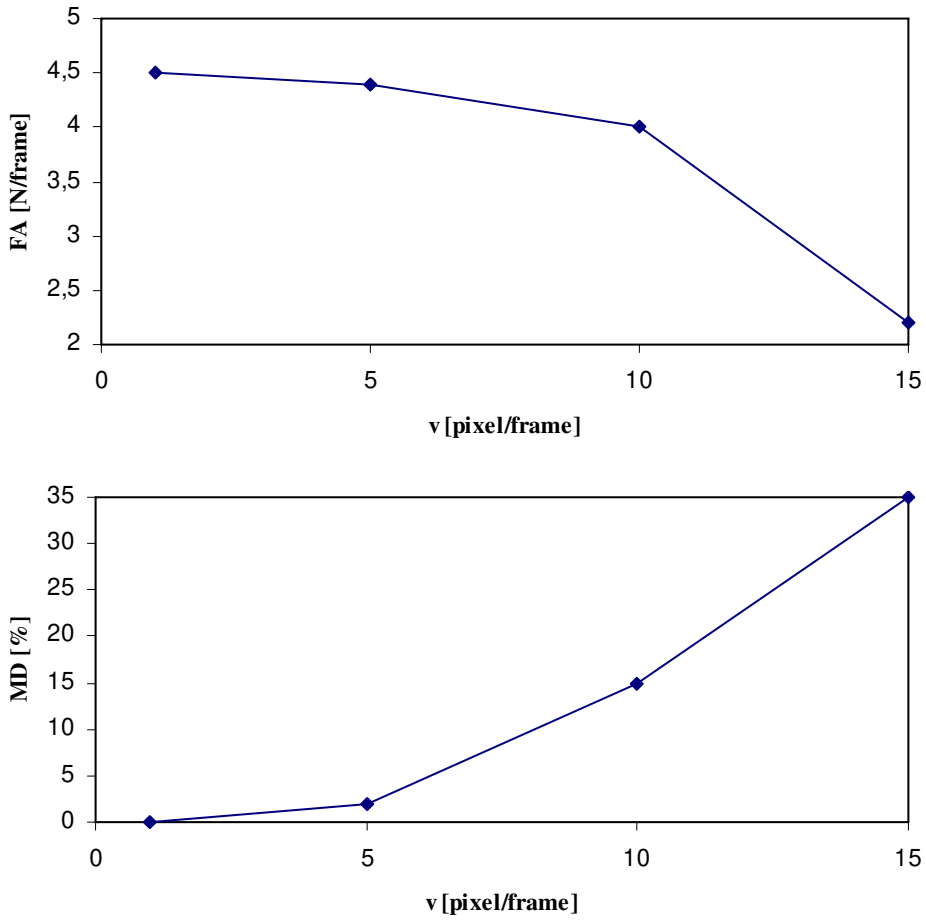

Figure 12 - Optical Axis Oscillation: FA/MD as a function of optical axis oscillation velocity (K=1.5)

In summary, the algorithm exhibited a good reliability level in conditions of nice weather, good illumination, and fixed background. However, imposing a motion of the optical axis in order to simulate camera wobbling effects, false detections and miss-detections were observed.

Subsequent tests in Castel Volturno were carried out in more critical weather and illuminations conditions. In this case, the false alarm problem was much more dramatic, leading to a really hard trade-off in detection algorithm between sensitivity and reliability. An example is reported in figure 13. It is worth noting that these results are in agree with flight tests performed in USA.

Figure 13 – Multiple false alarms in critical illumination conditions

From a practical point of view, the computational cost of the algorithm is high. Using a standard desktop computer with a 3GHz processor, on average 1 sec/frame is needed to realize velocity field estimation and target detection. Much time is required to produce velocity estimates at lower resolution scale, which is needed by the two differential techniques. These computations could be avoided if images were taken at very high frame rate in order to reduce targets apparent velocity, but this would also be very demanding for the point of view of processing resources.

In summary, the limitations of an Optical Flow-only based approach made it not adequate to replace the main system architecture of the anti-collision system that includes a radar.

Chapter 2

Integrated multi-sensor system architecture

2.1. The Autonomous Collision Avoidance module

In order to perform autonomous collision avoidance, as already anticipated an UAV has to be endowed with different capabilities: the capability to develop situational awareness and identify potential threats for flight safety ("sense") and the capability to perform evasive maneuvers and then recover its nominal trajectory ("avoid"). From an engineering point of view, both capabilities require that own-ship dynamics is known in real time: navigation system plays a key role. Then, the autonomous collision avoidance module is connected to the flight control system: in case of predicted collision, it has to ask for trajectory modification and then nominal mission recovery. These considerations lead to the conceptual architecture which is reported in figure 1. It is clear that though the sense and the avoid function are conceptually independent, they deeply influence each other. This Ph.D. thesis is focused on the sense function: all the sensors, techniques, algorithms, and hardware needed to this aim, are treated.

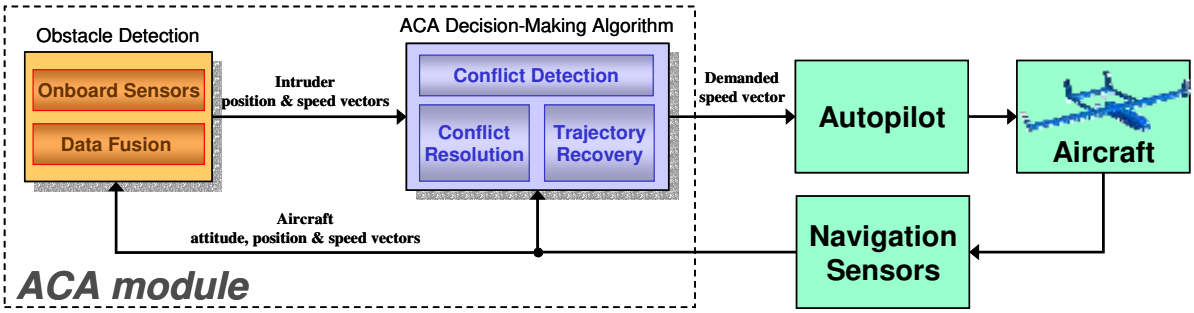

Figure 1 – Autonomous collision avoidance module architecture

2.2. Sensing Requirements

Sensing requirements for a DS&A System are connected to achievable field of regard, to range and angular resolution, to the detection range (which is connected to the time-to-collision by means of approaching speed) and to the system data rate.

The process of understanding the requirements was carried out in previous studies and is illustrated in [42]. The focus was on mid-air flight, thus maximum considered approaching speed was 500 kts. The all-time all-weather requirement was also considered.

In its initial configuration, TECVOL aims at demonstrating the capability to avoid one non cooperative flying obstacle in the search volume. In the final configuration, the system should be able to avoid up to four obstacles.

Table 1 reports the assigned sensing requirements. They were derived mainly from flight mechanics considerations relative to the execution of avoidance maneuvers. It is worth noting that they are compliant with international standards [5,6], as it also results from table 2, considering field of regard requirements.

Parameter	*Value*
Field of regard extent in azimuth	*-110°/+110°*
Field of regard extent in elevation	*-15°/+15°*
Range resolution	*20 m*
Azimuth resolution	*0.27°*
Elevation resolution	*0.27°*
Minimum allowed time-to collision	*19 s*
Data rate	*10 Hz*

Table 1 – Sensing system requirements

Source	***Azimuth***	***Elevation***
FAA P-8740-51: How to Avoid a Mid-Air Collision	*+/- 60 degrees*	*+/- 10 degrees*
International Standards, Rules of the Air, Section 3.2 (ICAO)	*+/- 110 degrees*	*No guidance (ASTM reports +/- 15°)*
FAA Advisory Circular 25.773-1 (Transport Aircraft Design)	*+/- 120 degrees*	*Variable: +37 and -25 degrees (varies with azimuth)*

Table 2 – International references for field of regard requirements

The first row of table 2 can be considered as a near term requirement, whereas the last line is the estimated reference for the far term. As it will be shown in the following, the near term requirement has been assumed as reference for flight demonstration in TECVOL project.

2.3. Obstacle Detection Sensors

One key point of sensor selection for autonomous collision avoidance is that no single sensor was found that is capable to fulfill all the requirements of table 1.

For example, radars guarantee adequate detection range and all-time all-weather performance, but angular accuracy is unsatisfying and data rate is of the order of 1 Hz.

Thus, a multiple sensor approach was selected by using sensors based on different technologies (i.e. active microwave, passive infrared, and visible cameras) in order to compensate the lack of performance of single sensors.

Considering the radar, in the choice of wavelength, maximizing detection range, minimizing sensor dimensions to enable installation on-board a lightweight aircraft, and improving as much as possible angular resolution are contradicting requirements.

In fact, radars operating at low frequencies are relatively unaffected by atmosphere, but are large in size and unable to provide required spatial resolution, due to main lobe width, which is directly proportional to operating wavelength, as shown in equation (1).

$$\vartheta_{3dB} = K\frac{\lambda}{l} \qquad (1)$$

where *K* is a coefficient whose value depends on the considered aperture and feeding, and *l* is the antenna length in the considered direction.

A higher frequency radar, instead, is smaller in size and provides better resolution for given aperture size, but is more susceptible to atmospheric and weather effects [31], and in particular to rain, as it results if we consider atmospheric attenuation produced by fog and rain (figure 2).

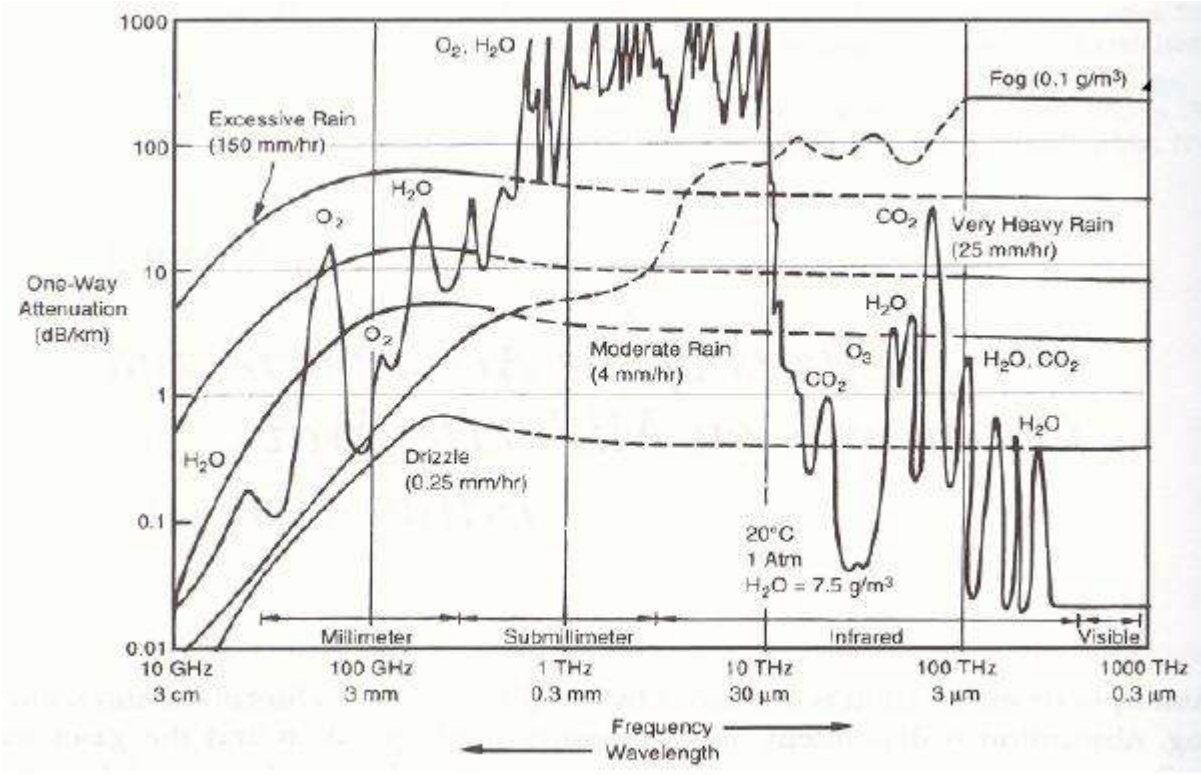

Figure 2 – Atmospheric attenuation for different weather conditions

A Ka-band radar (frequency 35 GHz, wavelength 0.9 cm) can be considered as a good compromise between the contradicting requirements. Thus, the selected radar was the AI-130^{TM} OASys^{TM} (Obstacle Awareness System) model produced by Amphitech^{TM}. It is a pulsed radar operating with a carrier at 35 GHz. It has been already used for UAV anti-collision flight tests in NASA-ERAST flight tests [13,14]. As it was already reported in the previous chapter, OASys performance in terms of detection range was found to be adequate for collision avoidance requirements except in the cases of very fast intruders.

Figure 3 – Amphitech OASys Radar

In the designed system, the radar is the main sensor and provides the all-time all-weather capability. It allows for demonstration of the sense and avoid capability in a field of regard of 120° in azimuth.

Its detection range as a function of target mean radar cross section (RCS) can be evaluated on the basis of radar equation [42] and is reported in figure 4:

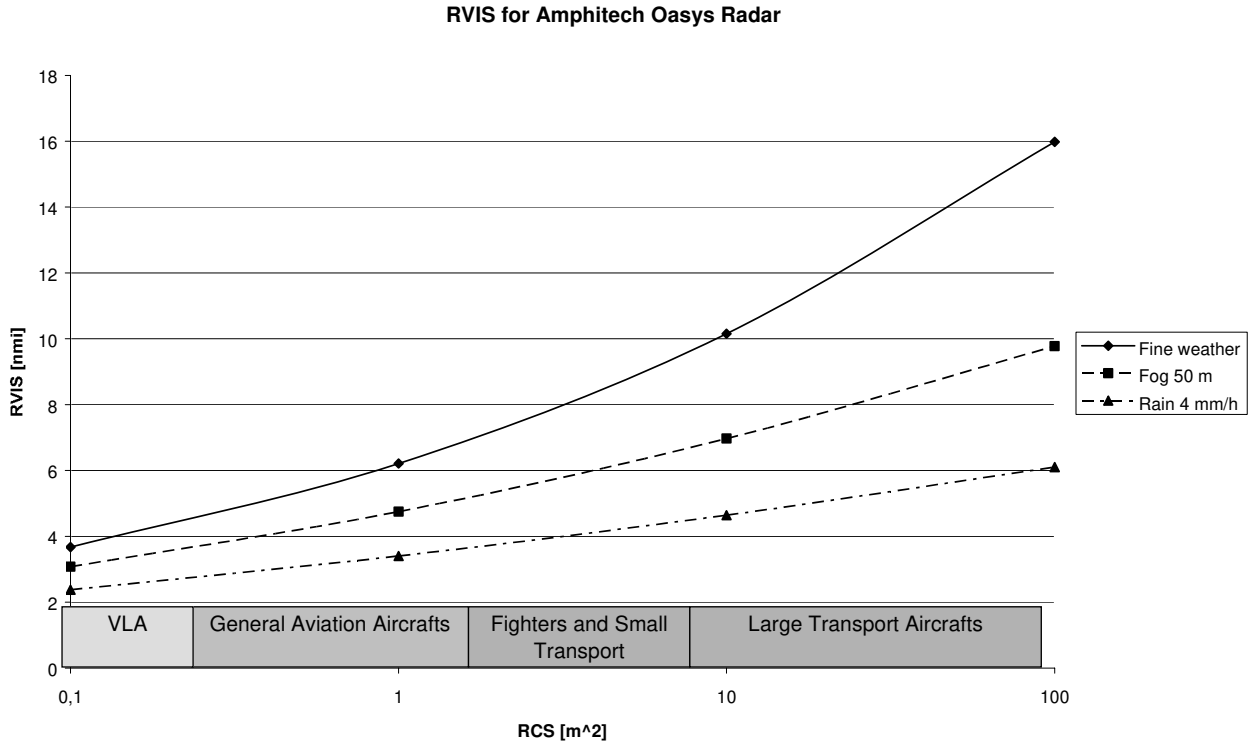

Figure 4 – Estimated detection range for OASys radar (from [42])

In order to improve angular accuracy and data rate, auxiliary electro-optical sensors have been considered, and in particular:

- 2 visible cameras, one panchromatic, one color (figure 5);
- 2 infrared cameras (figure 6);

Figure 5 – Visibile cameras

Figure 6 – Infrared cameras

In particular, the visible cameras are the Marlin™ F145B2™ and F145C2™, produced by Allied Vision Technologies GMBH™ (AVT). The former is a panchromatic camera, whereas

the latter is a color (Bayer tiled) camera. Both communicate via an IEEE1394 IIDC interface and are capable of producing color/panchromatic images up to 1392x1040 pixels. Furthermore, they can acquire images up to 10 Hz at full scale, with a resolution depth of 8/10 bits per pixel. They were equipped with MV618T^{TM} optics realized by AVT, with focal length of 6.5 mm and thus a field of view (FOV) of approximately 52.9° x 40.8°.

The two visible cameras are installed parallel to the aircraft longitudinal axis to get simultaneously a high resolution panchromatic image and a color one of the same region. While the panchromatic camera was chosen basically for fusion with radar data, the color camera was considered to add a feature information useful for obstacle recognition.

The infrared (IR) cameras are two Thermocam A40V^{TM} produced by FLIR^{TM}. They have a detector with 320 X 240 pixels, can acquire images up to 50 Hz, and are equipped with optics with focal length 35 mm, so that their FOV is 24 ° X 18 ° and the IFOV is of 0.075°. Thermal resolution is 0,08 °C at full frequency. The sensor is a Focal Plane Array (FPA), that is a non-cooled micro-bolometer. The imaged spectrum is in the thermal infrared field: from 7.5 μm to 13 μm. It is worth noting that the IR wavelength choice is due to the fact that spectral radiance has a broad peak in this region for temperatures near 300 K. In any case, the camera can cover a temperature field from -40 °C to 500 °C. Due to their limited angular aperture, the IR cameras are pointed slightly eccentric to get a field of view comparable to VIS cameras one.

An approximate estimate of IR sensors detection range can be based on geometrical considerations only. In particular, it is connected to the linear dimension which corresponds to the Effective Instantaneous Field of View (EIFOV). EIFOV can be evaluated as EIFOV=1.5 * IFOV [43].

Following this line of reasoning, figure 7 is obtained.

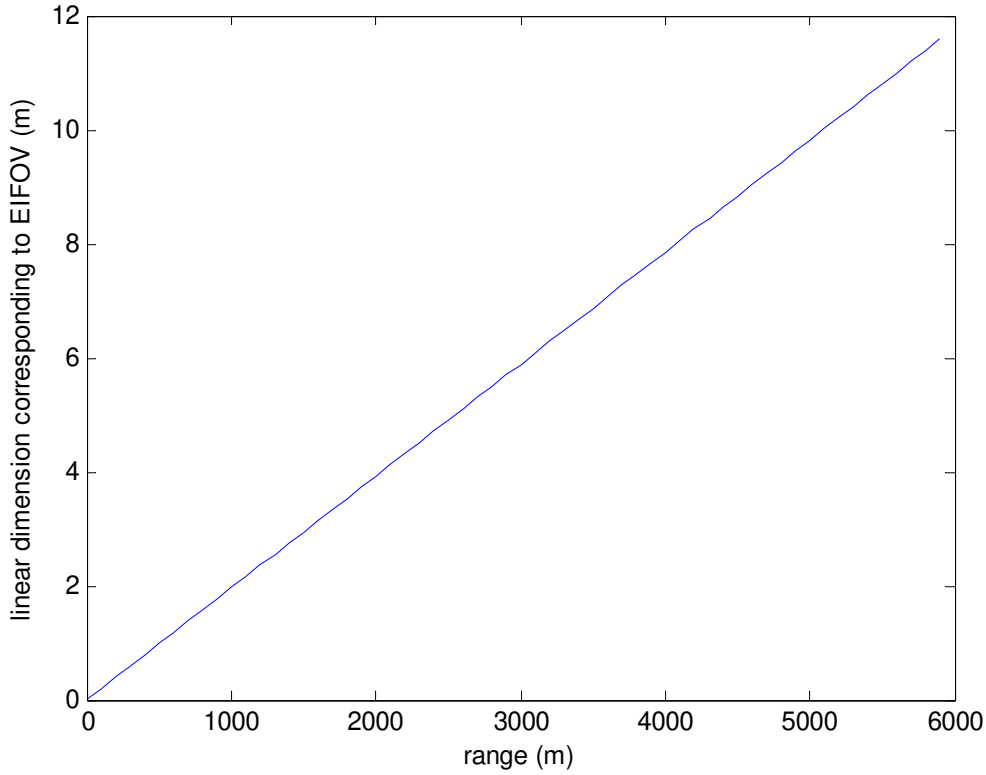

Figure 7 – Infrared sensors EIFOV in meters as a function of range

Assuming an obstacle reference linear dimension of 5 m, one gets a detection range of approximately 3 km. It is worth noting that the same calculation can be applied to visible cameras, and it would lead to a larger (about double) detection range. However, it must be taken into account that while in the IR case it can be foreseen that a sufficient contrast is generated between intruder and the blue sky background, this does not hold in general for visible sensors. Detection range for these sensors is very much dependant on illumination conditions. In any case, it will be possible to estimate detection range for EO sensors in the planned flight tests.

Sensors' layout is shown in figure 8. It was designed to allow for a compact installation of the sensor system, and in order to minimize vibration effects. The radar is mounted in central position with respect to the EO cameras. All the sensors pack is installed on the top of the aircraft wing (figure 9).

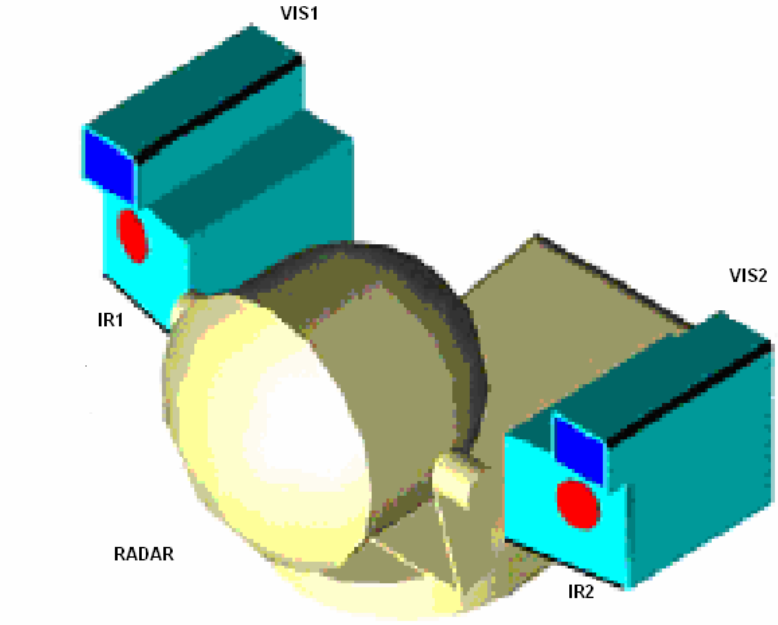

Figure 8 - Sensors' layout

Figure 9 - Obstacle detection sensors mounted on-board FLARE

2.4. Sensor and data fusion architectures

A multi-sensor configuration, such as the one selected for autonomous collision avoidance, allows for integration of different, complementary measurements and information. The resulting system is intrinsically more reliable with respect to a single sensor configuration: the different sensors derive object signatures from different physical processes, and so they generally do not cause false alarms on the same artifacts.

However, combining sensors' information in an effective way, and respecting the strict real time requirements of the system, is not an easy task.

Generally speaking, data fusion architectures can be classified in different ways [31], depending on the amount of data processing that occurs at sensors level, and the location of fusion processes. Basically, three architectures can be identified: sensor-level fusion, central-level fusion, hybrid fusion.

In sensor-level fusion, each sensor detects and estimates tracks of targets before data entry into the fusion processor. This means that computational power has to be available at sensor level. Moreover, from target kinematic state estimation point of view, sensor-level tracks, and not measurements, are to be fused. Track-to-track fusion is a complex task and requires a certain amount of computational expense [44,45]. Of course, sensor-level fusion reduces the exchange of data among nodes, paying the cost of a larger computational load.

Central-level fusion is based on a completely different logic. In fact, each sensor provides to the fusion processor minimally processed data, which are combined to obtain a single set of tracks. Globally, the computational load is less than in sensor-level architecture, however generally more data have to be exchanged among network nodes.

Regarding the accuracy of the tracking function, central-level fusion is proved to provide the best results [46]. Central-level fusion major drawback is that if a sensor measurement is degraded, it affects the entire estimation process. This degradation can be due to problems in sensors' hardware, but above all to electronic counter-measures that affect, for example, radar range measurements. In the framework of TECVOL, both cases are not considered. In particular, it was not supposed to build a one-failure-tolerant architecture. Furthermore, diagnostic systems at software level have been designed to detect hardware problems at sensor level. Then, operating in a civil collision avoidance scenario electronic countermeasures have not been taken into account.

Hybrid fusion tries to combine the advantages of sensor and central level fusion. In fact, it is supposed that both raw sensor data and sensor-level tracks or high level information can be combined in the fusion processor. However, the disadvantage is in terms of increased processing complexity and possibly increased data transmission rates.
As it will be clarified further in the next section, a central-level fusion architecture has been considered in this project. The main reason was to reduce the overall computational cost in order to respect the real time requirements of the system. The problem of raw data exchange was solved by decentralizing the detection function while keeping an unique central-level tracking module.

2.5. Developed sensor fusion logical architecture

In the first part of TECVOL experimentation it is foreseen to demonstrate the autonomous collision avoidance capability with a single obstacle. To reach this milestone, multi-sensor detection and tracking are needed. The following steps are related to the implementation of autonomous obstacle identification and collision avoidance with multiple obstacles. Thus, the identification function will be developed in detail in the near future. Anyway, at architectural level identification has already been considered.
The sensor fusion logical architecture is depicted in figure 10.
In particular, detection, tracking and identification modules are identified. Though identification logic has not been developed yet, it is foreseen that a two-way information exchange exists between the latter and the tracking module: while the ID information can be used to adjust the tracking filter bandwidth to obstacle typical dynamics, track behavior can be used as an additional identification feature.
Basically, as anticipated the sensor fusion architecture is central-level: in fact, tracks are not produced at sensor level and minimally processed data are combined in an unique Kalman filter-based tracking algorithm. However, detection function is in some way decentralized, and a hierarchical sensor architecture is considered.

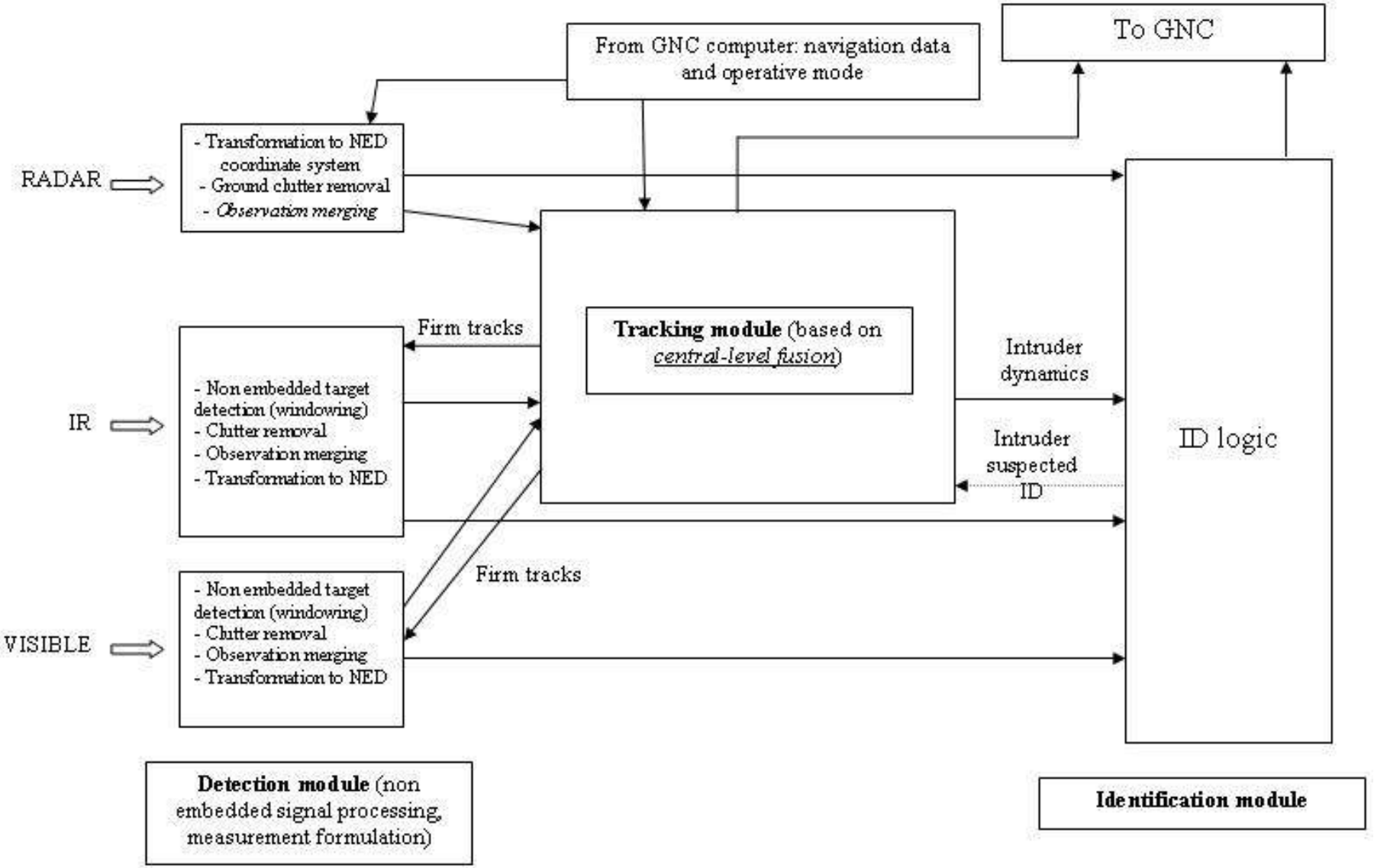

Figure 10 – Sensor fusion logical architecture

This choice derives from the consideration that radar measurements are less sensitive to atmospheric effects and typically offer larger detection range with respect to EO cameras. Thus, the radar is considered as the main sensor, and the EO (both visible and infrared) cameras work as aiding sensors.

This means that while target detection by radar is performed autonomously with a search in the whole sensor field of view, the object detection process is carried out in EO images on the basis of cues by the tracking module, considering a search window centered in the foreseen obstacle position. In particular, only firm tracks (generated on the basis of radar measurements) are used for EO object detection, in order to keep a reasonably low false alarm rate.

Thus, the first step in obstacle detection by EO cameras is the selection of an area of interest in the latest acquired image on the basis of tracking predictions. An important point is the determination of the optimal area of interest dimensions. If the area is too small and does not take into account relevant uncertainties, then it is possible that the obstacle is found outside the area. On the other hand, if the area is too large, there is a negative effect on computational time for detection, but also reliability of the detection process can be reduced.

In fact, the basic methods for obstacle detection in the area of interest are threshold criteria, which operate on the single frame and detect object pixels as those with a statistically different intensity from the background. For example, in global threshold techniques, as in the optical flow-based algorithm illustrated in the previous chapter, the detection threshold is usually chosen dynamically, on the basis of a statistical mean:

$$T = \mu \pm K \sigma \qquad (1)$$

where μ is the mean value of intensity distribution in the area of interest, σ is the standard deviation, K is a non-negative coefficient which is chosen on the basis of requested false alarm rate and probability of detection.

These techniques are based on the assumption of an uniform background. Enlarging the area of interest implies that a more heterogeneous area is imaged with a negative impact on detection performance. In the considered case, the area of interest is selected taking into account radar angular accuracy, tracker predicted accuracy (which is estimated in the tracks covariance matrices, as it will be clarified in the following), EO cameras alignment uncertainty, predicted aircraft attitude angles error noise.

After the pixel detection step, nearby object pixels are grouped and the algorithm calculates centre coordinates, which represent the synthetic estimate to be sent to the tracking module. In this phase, a control is applied on the number of object pixels: if it is too large, the measurement is discarded and treated as unreliable. Since only target estimates are sent to the tracking module, there is a really low data rate which is another positive aspect in view of real time implementation. The EO detection process is schematically described in figure 11.

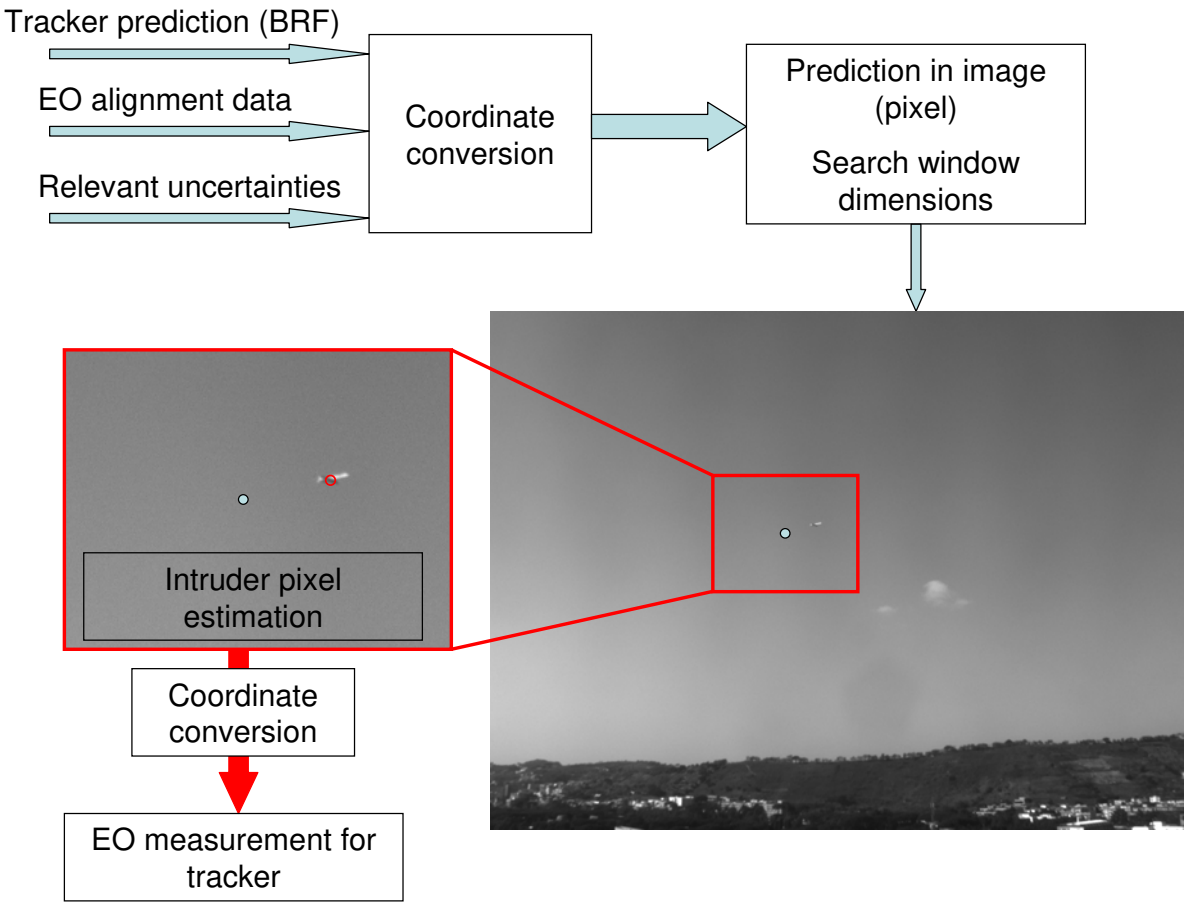

Figure 11 – Logic of intruder detection by electro-optical sensors

Figure 10 also reports, in the measurement formulation phase, the transformation of sensor data to a North-East-Down reference frame. This transformation depends on the tracking algorithm structure and has important consequences on it, as it will be shown in the next chapter.

Tracking module constitutes the core of the entire system, and it is strongly related to the Guidance, Navigation and Control System. In fact, on the one hand it needs navigation data in real time to properly follow obstacle dynamics. On the other hand it has to transmit intruder

estimated to the flight control system, which has to decide eventual collision avoidance maneuvers in case of predicted collision.

Tracking algorithm will be described in detail in the following. However, it is important to point out that EO sensors can be considered as auxiliary sensors also because their estimates do not influence tracks' status, which is based on radar measurements only. This means that the system can work in case of a failure to the EO system, while it is strongly dependant on radar. In case of radar failure, the detect and sense module is useless. Of course, a partially radar failure tolerant system is considered among further developments.

2.6. Sensor fusion hardware architecture

The whole DS&A System hardware architecture (including the flight control computer with sensors and actuators) is reported in figure 12. Considering the sensing system, which is the subject of this thesis, it is made up by two separate processing units, which implement different functions with different operating systems (OSs). The Real Time Computer is based on a deterministic OS. It is directly connected with the radar sensor via an Ethernet link and the TCP/IP protocol, performs tracking (and identification in the future), and exchanges data with the Guidance, Navigation, and Control (GNC) system by a deterministic data bus, which is the Controller Area Network (CAN) bus. CAN bus is finding a growing diffusion in aerospace systems thanks to its simplicity, reliability, flexibility and low cost [47-49]. The EO sensors are connected via a Firewire link to the Image Processing Computer, based on a conventional OS, that has to process the visible and infrared images to find estimates of intruders' position and shape (in view of the identification function). The two computers exchange data by an Ethernet connection and the User Datagram Protocol (UDP) [50]. In particular, firm tracks are sent in one direction, target estimates (in case of reliable detection) are sent in the other. The separation of the two processing units allows to reduce the computational load on both systems. All the architecture and the on-board software will be described in more detail in the following.

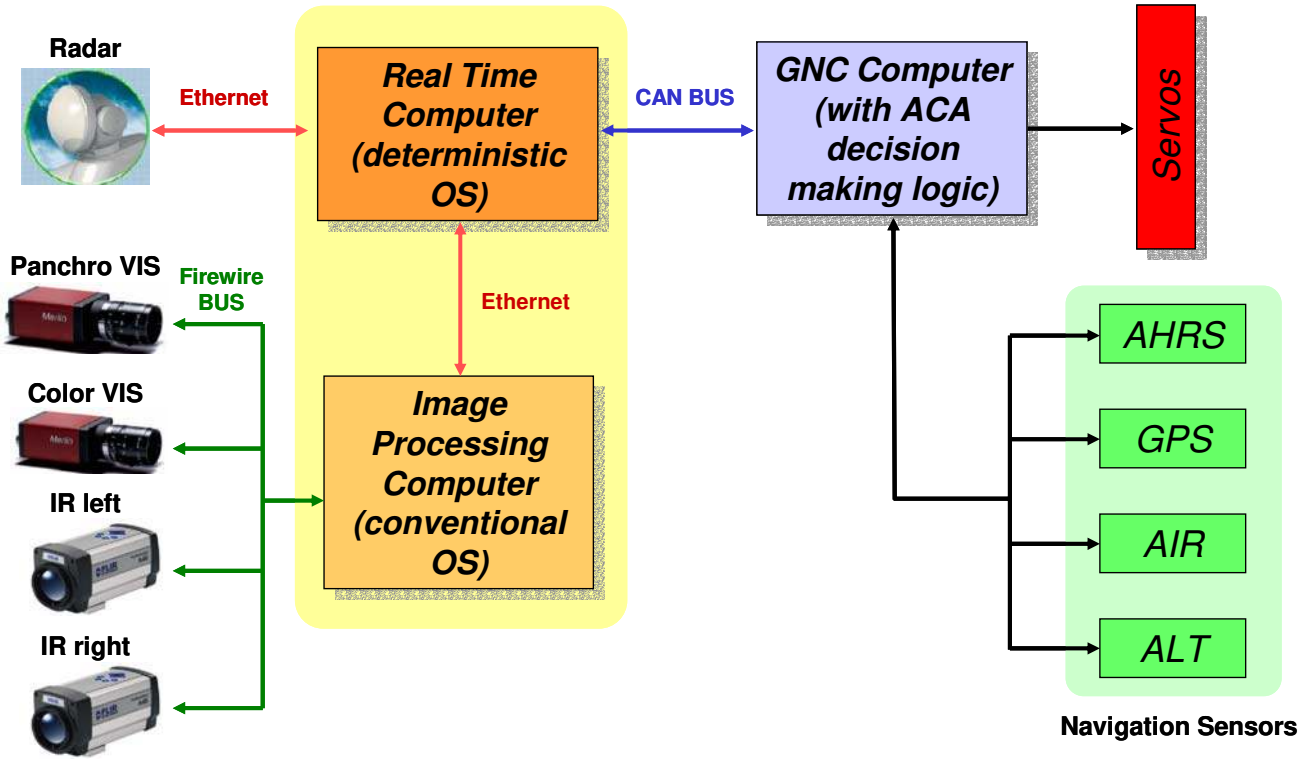

Figure 12 – DS&A System hardware architecture

Chapter 3

Airborne multi-sensor tracking algorithm design

3.1. Introduction

Tracking module and its algorithm constitute the core of the whole DS&A system. In fact, tracking is based on fusion of information from the different sensors in order to autonomously reach required situational awareness for the UAV. Basically, tracking is needed:

- to associate at the same intruder measurements gathered in different scans/frames;
- to measure obstacle kinematics in order to identify a potential collision;
- to increase measurement rate with respect to sensor raw data;
- to eliminate false alarms and clutter returns.

It is worth noting that, since the system is completely autonomous, it is mandatory to have reliable estimates not only of intruders' positions, but also of their motion, since this latter information is needed by the collision avoidance logic to decide whether or not it is necessary to perform an evasive maneuver.

This chapter illustrates in some detail how does the tracking algorithm work, and the relevant design choices which derived from the considered application. Among these, one of the design drivers was to allow for real time implementation with standard computational resources. On the other hand, civil airspace features had to be taken into account.
In the following sections, after an overview of multi-sensor multi-target tracking basics, the design process for the collision avoidance application is detailed. In particular, first of all relevant aspects deriving from operative scenario are clarified. Then, the mathematical structure is detailed by analyzing Kalman filtering, choice of dynamic models and tracking coordinates, inclusion of navigation system uncertainties. Finally, a numerical analysis is presented that puts into evidence the differences among different design choices.

3.2. Basics of multi-sensor multi-target tracking

The basic elements of a multi-target tracking system are shown in figure 1.

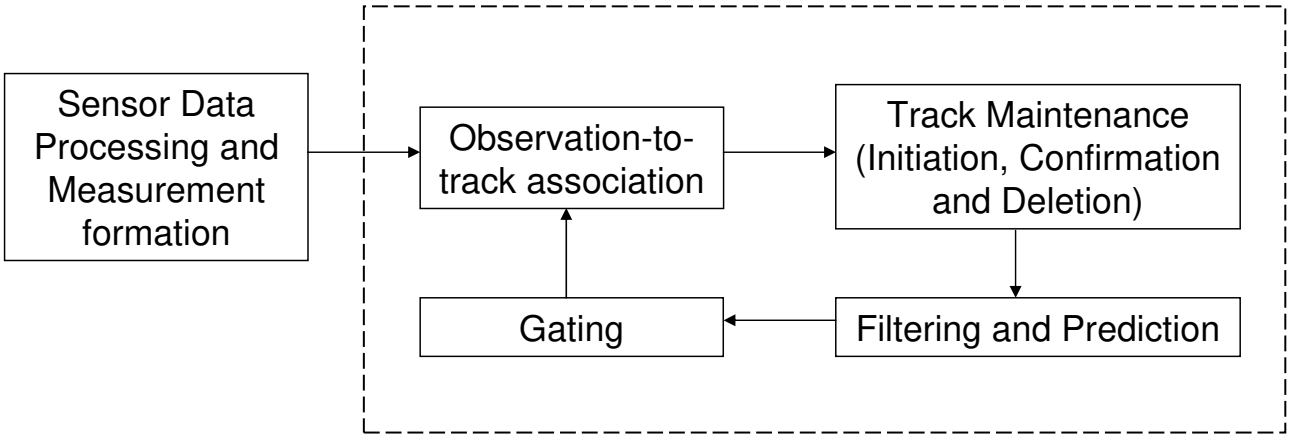

Figure 1 – Basic elements of a multi-target tracking system [46]

Generally speaking, tracking foresees 4 main phases. The input to the algorithm are sensor measurements, which represent in general objects of interest, false alarms, and clutter. Of course, objects of interest and clutter depend on the considered application. In our case, most of clutter returns will likely come from ground echoes. The tracks can be defined as hypothetical constructs which represent in substance estimates of position and velocity (and acceleration), given a time-ordered sequence of measurements. Assuming that a set of tracks

is present in the system memory at a given time, incoming input measurements are first considered for the update of existing tracks.

It is important to point out that a measurement formation phase has in general to be foreseen before data integration in the tracking algorithm. This is not only due to the necessity of registration in the case of multiple sensors, but also to the fact that tracking design is greatly simplified assuming that the sensors do not provide multiple simultaneous observations from the same source. In the developed anti-collision system (see previous chapter) this is obtained in the case of electro-optical sensors by mean of a centroiding operation, while observation merging does not seem to be necessary for the radar, due to its rough angular accuracy.

Gating operation allows to determine "reasonable" observation-to-track pairings. It corresponds to build a volume in the state space around the predicted estimate, where valid observations are likely to fall. The logic is to compare predicted estimates with measured quantities by calculating the distance between them. If the latter is less than a maximum established limit, then the observation satisfies the gate and is a valid candidate for updating the track. Of course, gate dimensions constitute a key point. On the one hand, gate has to be large in order to reduce the risk that a valid measurement is rejected. On the other hand, it has to be small enough to avoid that undesired returns are considered for association to track, which can degrade a lot tracking performance.

Usually, the distance between observations and predictions is evaluated in a statistical way, in the sense that it is normalized to take into account prediction and measurement accuracy. It is worth noting that this requires that both accuracies are to be known. Then, the maximum allowable error is usually set on the basis of statistical considerations.

Gating allows to reduce subsequent computations. However, in general a further step is necessary in order to determine which observation-to-track assignments are actually made. This is usually named as association.

Association takes the observation-to-track pairings that satisfied gating and solves the assignment problem. This is usually performed with the aim of minimizing a cost function, which can be for example the total summed distance. In mathematical terms, the problem can be formulated as the classical assignment problem for which optimal (Munkres algorithm [51]) and sub-optimal solutions [52-54] exist.

Some simple solutions are represented by the global nearest neighbor (GNN), the nearest neighbor (NN), the greedy assignment. GNN minimizes the total summed distance under the constraint that at most one observation can be used to update a single track, and vice versa.

NN associates each track with the nearest observation, so that an observation can be used to update more than one track. The greedy assignment is the simplest technique, it foresees that the best possible assignments are made sequentially and observation and tracks are removed as they are assigned, so that an observation/track can not be used with more than one track/observation.

It is worth noting that association needs to be increasingly more complex (and computationally expensive) if a more and more dense target environment is considered. In case of very dense multi-target environments, recent techniques such as multiple hypothesis tracking have to be used [55]. Indeed, the weight to be assigned to data association algorithms can be based on the relation between obstacle detection sensors angular accuracy, and typical obstacle separation. In fact, it comes out that miscorrelation (i.e., wrong observation-to-track association) frequently occurs when targets' separation is less than five times the angular measurement-error standard deviation [56]. Moreover, miscorrelation in presence of a single target can occur because of false alarms or clutter.

After the association phase, track status has to be properly handled. In particular, tracks are usually divided in the following categories: one-plot (single observation not associated to any existing track), tentative (at least two observations, but confirmation logic still required), firm (confirmed track). The transition tentative-firm can be based on different techniques without a great computational load. Tracks are deleted if they are not updated within a reasonable interval.

The filtering and prediction phase allows for combination of track predictions and sensor measurements, and produces new track predictions. It is usually performed using Kalman filtering. One of the advantages in using Kalman filtering is that prediction uncertainty can be estimated from the covariance matrix, which is useful in the gating process. Of course, this implies that covariance estimate consistency must be properly assured.

This section offered a brief overview of multi-target tracking algorithms. In the real world practice, tracker design must be tailored to the particular application. This influences the computational weight to be assigned to the different parts, and the mathematical models to be applied.

3.3. Tracking for collision avoidance in civil airspace

In general, tracker design is the result of a trade-off between conflicting requirements: for example, the accuracy of tracks for non-maneuvering aircrafts versus maneuver detection and track continuity through maneuvers. Furthermore, in the case of civil airspace scenarios, there are some particular features that influence tracking logic. First of all, a dense multi-target environment has very low probability. Of course, depending on the sensors' angular accuracy and detection range, it is possible that different flying obstacles produce miscorrelation though respecting the so-called bubble distance, established by Federal Aviation Administration (FAA) regulations (165 m) [6], but this can happen at high distances, when avoidance maneuvers are not initiated, considering a nuisance-free approach.

This fact greatly influences allocation of computational resources: the association phase can be kept at a relatively simple level, while a greater effort can be devoted to improve filtering and prediction performance by means of accurate dynamic modeling and optimal choices of tracking coordinates. This conclusion makes a big difference with respect to other scenarios (the military ones for example) where association is the most difficult task and filtering schemes are often kept simple to save computational resources [56].

Another aspect to be taken into account is that in the considered scenario the most important quality parameter of the tracker is its reliability at short distances, rather than its absolute accuracy, since this is the most critical situation for the entire system. Finally, it is worth noting that, regardless of DS&A sensors' resolution, navigation data are corrupted by their own errors. Since it is not possible to eliminate them, estimating navigation errors' effects on the tracking accuracy and on the behavior of the entire DS&A system is certainly a very important subject. Of course, the latter analysis doesn't make sense if navigation sensors are much more accurate than obstacle detection sensors (which is not our case if radar/EO fusion is considered).

3.4. Intruder state estimation: filtering and prediction

3.4.1. Overview of Kalman filtering

Tracking requires that track predictions and correlated measurements are properly combined in order to find an updated estimate. The simplest filters use fixed pre-computed gains for this combination; the alpha-beta and alpha-beta-gamma filters are typical examples in this category [57]. The computational and memory advantage of these filtering schemes has a cost in terms of accuracy, reliability (track continuity through maneuvers is hard to obtain) and flexibility (the filter gains do no adapt to detection history). Also because of the progress in computer systems, most trackers today rely on Kalman filtering based algorithms.

Kalman filter is a common algorithm for state estimation on the basis of sensor measurements [58-60]. It was originally developed for tracking [61,62], and since then it has found usage in several fields of applications. Kalman filter works on the basis of a prediction/correction scheme, as it is shown in figure 2 considering its discrete form.

It is in general based on a few assumptions: the system state evolves according to a known linear equation driven by a known input and an additive process noise, which is zero-mean white (uncorrelated in time) with known covariance matrix Q(k). Moreover, measurements are a known function of the state with an additive measurement noise, which is again zero-mean white with known covariance R(k). Required initial parameters are the initial state with its uncertainty (that is, its covariance). System and measurement noise are assumed to be uncorrelated.

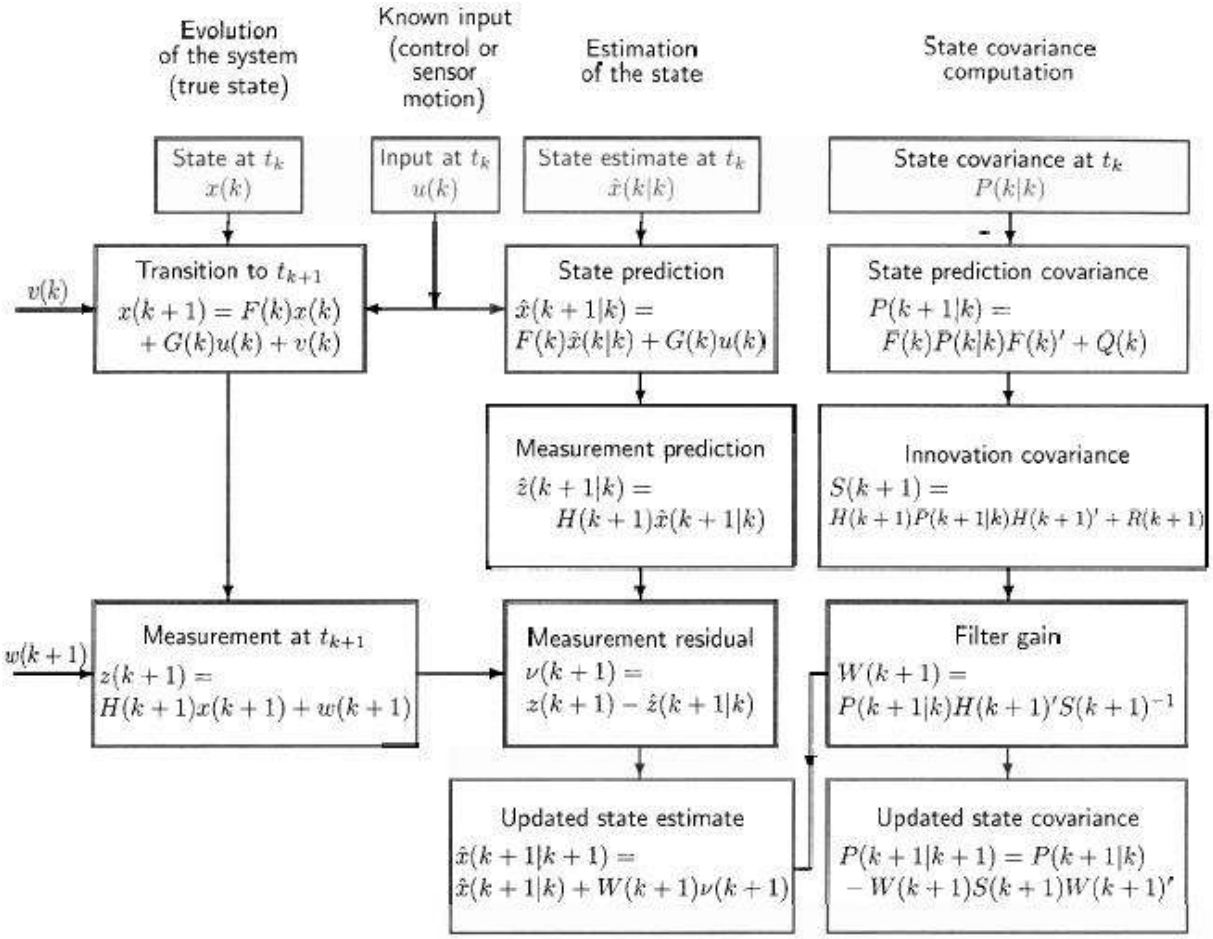

Figure 2 – One cycle in the state estimation of a linear system by Kalman filtering (from [44])

It can be demonstrated [57,58] that under the Gaussian assumption for all the errors and noises the Kalman filter is the minimum mean square error state estimator. Without the Gaussian assumption, the Kalman filter is the best within the class of linear estimators.

Besides giving better accuracy in state estimates with respect to fixed gain filters, as anticipated above the use of Kalman filtering is particularly suited for tracking since it self estimates its accuracy through the covariance matrix. The accuracy estimation is required for the gating operation. Moreover, Kalman filter flexibility is very useful in the considered case, where multiple sensor measurements at different data rates have to be combined: different sensors can be accommodated using different measurement equations and covariance matrices, while the filter automatically adapts to any radar/EO detection history, and to variable sampling times.

In practice, the Kalman filter produces accurate estimates if both sensors and obstacle dynamics are accurately modeled. Accurate sensor modeling is connected to a correct choice

of their covariance matrix R. As for target motion, this is obviously unknown and thus has to be modeled taking into account typical obstacle dynamics. Some options on the choice of dynamical models are described in the following section.

3.4.2. Obstacle dynamics: the Singer model

The problem of modeling aircraft motion has been widely studied from the origin of tracking systems, in view for example of application in the air traffic control systems. In our case, the choice of dynamic model was not only very important for the accuracy of tracking in different phases, but also had to take into account the real time requirement of the anti-collision system, with a 10 Hz data rate: thus, excessively complex dynamic models were not suitable for the application.

A comprehensive survey of dynamic models for tracking is reported in [63].

The most used models for random target motion are based on target acceleration. One of the most common among them is the Singer acceleration model [64]. It assumes that the target acceleration evolves in terms of a correlated noise process, so that, considering a one-dimensional case with a scalar coordinate x, acceleration auto-correlation function is given by

$$r_{\ddot{x}}(\tau) = E\{\ddot{x}(t)\ddot{x}(t+\tau)\} = \sigma_m^2 e^{-(\mu\tau)} \tag{1}$$

where

σ_m^2 is the acceleration instantaneous variance;

μ is the inverse of target acceleration time constant.

σ_m and μ represent the input parameters for the model: σ_m is connected to the range of accelerations that can be foreseen for the target and can be determined for example on the basis of a ternary-uniform mixture [63], while μ is connected to how fast target dynamics changes, so that the time constant can be considered as a maneuver time duration. In continuous terms, the dynamic model is in the form

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\mu \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) = A \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + Bw(t) \tag{2}$$

where w(t) is a white zero-mean gaussian process with variance σ_m^2. In discrete terms (considering a sampling interval equal to T), target acceleration is a first order Markov process of the form

$$a(k+1) = \rho_m a(k) + \sqrt{1-\rho_m^2}\,\sigma_m r(k) \tag{3}$$

where $\rho_m = \exp(-\mu T)$ and r(k) is a zero-mean unit-standard deviation Gaussian random variable. The resulting transition matrix is

$$\Phi(k+1 \mid k) = \begin{bmatrix} 1 & T & \dfrac{\left(e^{-\mu T} - 1 + \mu T\right)}{\mu^2} \\ 0 & 1 & \dfrac{\left(1 - e^{-\mu T}\right)}{\mu} \\ 0 & 0 & e^{-\mu T} \end{bmatrix} \tag{4}$$

while the exact solution for the system noise covariance matrix Q is given by

$$Q = 2\sigma_m^2 \mu \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \tag{5}$$

where

$$
\begin{aligned}
q_{11} &= \frac{1}{2\mu^5}\left[1 - e^{-2\mu T} + 2\mu T + \frac{2\mu^3 T^3}{3} - 2\mu^2 T^2 - 4\mu T e^{-\mu T}\right] \\
q_{12} &= \frac{1}{2\mu^4}\left[e^{-2\mu T} + 1 - 2e^{-\mu T} + 2\mu T e^{-\mu T} - 2\mu T + \mu^2 T^2\right] \\
q_{13} &= \frac{1}{2\mu^3}\left[1 - e^{-2\mu T} - 2\mu T e^{-\mu T}\right] \\
q_{22} &= \frac{1}{2\mu^3}\left[4e^{-\mu T} - 3 - e^{-2\mu T} - 2\mu T\right] \\
q_{23} &= \frac{1}{2\mu^2}\left[e^{-2\mu T} + 1 - 2e^{-\mu T}\right] \\
q_{33} &= \frac{1}{2\mu}\left[1 - 2e^{-2\mu T}\right]
\end{aligned}
\tag{6}
$$

As it results from previous equations, the model is flexible in that it allows to take into account different dynamic environments by properly adjusting σ_m and μ. Regarding these parameters, and the resulting Q matrix, in practice a conservative approach has to be chosen. In fact, a relatively large Q allows to bound the Kalman gains from below, preventing the tracking filter from generating biases which can finally result in track loss [65]. Moreover, in the tracking practice it is often convenient to have a process noise that accounts for potential miscorrelation effects or for obstacle maneuvers [46]. In a few words, Q matrix results from a trade-off between tracking reliability (especially during intruder maneuvers) and track accuracy during non maneuvering phases.

Singer acceleration model can assume some forms of interest on the basis of the ratio between sampling time and maneuver time constant. In particular, if sampling interval is much less that maneuver time, then acceleration is essentially seen as a constant with its derivative being a white noise process. This is known as constant acceleration model [66] and Q matrix takes the limit form

$$\lim_{\mu T \to 0} Q = 2\sigma_m^2 \mu \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6} \\ \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^3}{6} & \frac{T^2}{2} & T \end{bmatrix} \tag{7}$$

When sampling interval is much larger that maneuver time, the acceleration is seen itself as white noise: the resulting model is known as constant velocity model [66] and its Q matrix is

$$\lim_{\mu T \to \infty} Q = 2\sigma_m^2 \mu \begin{bmatrix} \frac{2T^3}{3\mu} & \frac{T^2}{\mu} & \frac{1}{\mu^2} \\ \frac{T^2}{\mu} & \frac{2T}{\mu} & \frac{1}{\mu} \\ \frac{1}{\mu^2} & \frac{1}{\mu} & 1 \end{bmatrix} \tag{8}$$

These limiting cases put into evidence how Singer model allows to cover a wide range of dynamic environments, as a function of design choices. On the other hand, model success relies on the accuracy of these choices.

The Singer model was detailed in this section in relation with a single state variable x. In the next section, modeling is extended in order to derive the real equations to be used on-board the UAV for real time obstacle motion estimation.

3.4.3. Tracking coordinates and filtering schemes in airborne multiple sensor tracking

The choice of tracking coordinate system and state components in airborne tracking is a non trivial problem [46, 56, 57, 63, 67].

Before addressing in detail the algorithms' choice and evaluation, it is important to underline that, on the basis of discussions with the developers of collision avoidance logic and software, it was decided that tracker outputs consist of estimated range, azimuth in the Body Reference Frame (BRF), elevation in the BRF, their first order time derivatives, and the same estimates

in a locally level reference frame, which is the North-East-Down (NED). The BRF can be assumed with axes along longitudinal, lateral and vertical aircraft axes and origin in the aircraft centre of mass, or in the inertial navigation unit if more convenient. The NED has the same origin of BRF, while its axes can be considered non rotating for aircraft platforms. It is worth noting that, due to the sensors' layout on FLARE, there is a vertical separation of the order of 1 m between obstacle detection sensors and aircraft AHRS (which is located in close proximity to aircraft center of mass). This generates a parallax effect in tracking estimates both in BRF and in NED. This effect can certainly be neglected in tracking algorithm development and simulation, however it can give some additional angular track error in the real time functioning when the intruder is in close proximity. Thus, the last step in real time track prediction on-board the UAV will be a conversion to account for this parallax.

In the considered case, tracking could be performed directly in the BRF, or in NED. The advantage of tracking in body coordinates is that the measurement conversion in NED is avoided, thus eliminating the effects of errors in attitude angles measurements. However, navigation errors would impact the tracker in any case since inertial measurements with their errors (accelerations and angular velocities) should be used also in this case. Moreover, the real drawback is that relative motion in BRF includes attitude dynamics, which makes it more difficult to track, unlike its projection in NED, which only depends on relative position dynamics. The difference between filtering in BRF or NED is somewhat similar to what happens in navigation, regarding the choice of stable platforms or strapdown systems. Furthermore, from a practical point of view, tracking in BRF can be heavily impacted by latencies in navigation data acquisition and reception by the tracker. Thus, in view of the considered application the NED was considered as the tracking reference system, while both body and stabilized estimates were considered to evaluate system performance and the impact on it of navigation errors. This comparative analysis is especially detailed in the following chapter.

In general, in tracking applications, several applicable models exist [63], which differ first of all in the state variables' choice [56,57]. In fact, it is possible to operate in Cartesian or in spherical coordinates.

The two choices are in substance complementary in terms of advantages and drawbacks.

Cartesian coordinates are useful in the prediction phase since relative dynamics is linear with good approximation in these coordinates in absence of intruder aggressive maneuvers. A common choice consists in considering as state variables the three NED coordinates of the

intruder, together with their first order time derivatives (which are the components of relative velocity), and possibly also the second and third order derivatives. Of course, tracking computational load increases as a function of the number of state vector components.

On the other hand, cartesian coordinates have the inconvenient that measurement equation is non linear, since both radar and EO sensors provide measurements in spherical coordinates. The problem can be faced in different ways.

A very efficient solution consists in considering in the measurements' vector the sensor measurements converted in Cartesian coordinates [68]. The transformation that is performed must be debiased and consistent [69-71]. However, this transformation requires the existence of a range measure, which is not the case of videocameras' measurements. It could be possible to derive the Cartesian "pseudo-measurements" by using the range prediction, but this introduces a potential for instability in the system, directly merging prediction and measures. From the computational point of view, in any case this procedure requires that measurement covariance matrix is calculated in real time.

The alternative is to implement an Extended Kalman Filter (EKF), where the relation state vector-measurements is linearized at each time step on the basis of the latest state estimate. Of course, it requires on-line calculation of measurement matrix H through derivation.

In detailed terms, a linear Kalman filter based on converted Cartesian coordinates can be developed as follows. Let us consider that the state vector is made up by 9 components, which are the obstacle NED coordinates with their first and second time derivatives.

Thus, $\underline{x} = \left[x_{NED} \; \dot{x}_{NED} \; \ddot{x}_{NED} \; y_{NED} \; \dot{y}_{NED} \; \ddot{y}_{NED} \; z_{NED} \; \dot{z}_{NED} \; \ddot{z}_{NED} \right]^{|}$

A classical Singer model can be assumed for the three target acceleration components. Thus, system dynamics is described by the equation

$$\underline{x}(k+1) = \Phi(k+1 \mid k)\underline{x}(k) + \underline{u}(k) + \underline{v}(k) \tag{9}$$

where the transition matrix is a 9 X 9 block diagonal matrix:

$$\Phi = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & f \end{bmatrix}_{9X9} \tag{10}$$

with the three *f* blocks in the form of equation (4). Each of them refers to a coordinate, so that different input parameters can be defined for the three components of target random acceleration.

In the same way, Q matrix is a 9 X 9 block diagonal matrix

$$Q = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{bmatrix} \tag{11}$$

and each of the *q* blocks takes the form of equation (5).

The deterministic input vector $\underline{u}_k$ contains the UAV acceleration components (projected in NED) and thus allows for own-ship motion compensation:

$$\underline{u}_k = \begin{bmatrix} -\frac{1}{2}T^2 a_N \\ -Ta_N \\ 0 \\ -\frac{1}{2}T^2 a_E \\ -Ta_E \\ 0 \\ -\frac{1}{2}T^2 a_D \\ -Ta_D \\ 0 \end{bmatrix} \tag{12}$$

The measurement equation (considering radar measurements) is in the standard linear form:

$$\underline{y}_k = H\,\underline{x}_k + \underline{w}_k \tag{13}$$

with

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{14}$$

Converted measurements can be debiased with proper equations [68]. This procedure influences measurement covariance matrix, which is a full matrix in the form

$$R = \begin{bmatrix} R_a^{xx} & R_a^{xy} & R_a^{xz} \\ R_a^{xy} & R_a^{yy} & R_a^{yz} \\ R_a^{xz} & R_a^{yz} & R_a^{zz} \end{bmatrix} \tag{15}$$

Explicit expressions for entries are provided in [71]. It is important to point out that they depend on measurement values and so R has to be calculated in real time. It is also important to point out that Cartesian structure generates a coupling effect in measurement errors, even if it is supposed that original radar range and angular errors are uncorrelated.

When only electro-optical measures are available, as already said the only way to obtain converted "pseudo-measurements" is to combine range prediction and angular measurements. As obvious, in the latter case the EO angular accuracy must be accounted for in the evaluation of R matrix. At the same time, range estimate accuracy can be taken from filter covariance matrix.

When both radar and EO measurements are available, R becomes a 6 X 9 matrix with the entries evaluated as in the case of radar only/EO only updates.

Initial state can be estimated on the basis of (radar) measurements and UAV velocity and acceleration as provided by the GPS system and by the inertial unit:

$$\underline{x}(0) = \begin{bmatrix} x_{NED\,0} \\ -v_N \\ -a_N \\ y_{NED\,0} \\ -v_E \\ -a_E \\ z_{NED\,0} \\ -v_D \\ -a_D \end{bmatrix} \tag{16}$$

Initial covariance matrix P can be estimated based on R entries for obstacle position, while uncertainties on relative velocity and acceleration can be initially estimated on the basis of typical obstacle dynamic behavior. It makes sense to choose a large initial covariance matrix in order to give a large weight to radar measurements in the track initialization phase.

An alternative approach for the introduction of angular-only estimates from EO sensors in a Cartesian model consists in adopting an Extended Kalman Filter. In this case, non linearity in measurement equation is handled by linearization around the latest state estimate. In detail, the EKF can be based on the same dynamic model as the previous filter. The measurement vector in case of radar detections is instead

$$\underline{y} = \begin{bmatrix} R & \varphi_{NED} & \vartheta_{NED} \end{bmatrix} \tag{17}$$

where φ_{NED} and θ_{NED} represent, respectively, target azimuth and elevation as computed in NED, usually named “stabilized azimuth” and “stabilized elevation”.

The measurement vector reduces to the second and the third coordinate of equation (17) in case of EO detections, while synchronous detection by radar and EO sensor can be accounted for by means of a five components vector. Obviously, in case of combined radar/EO detection, the filter will weight EO angular estimates more than the less accurate radar ones.

The non-linear measurement equation (considering the radar) is

$$\underline{y} = \begin{bmatrix} \sqrt{x_{NED}^{\ 2} + y_{NED}^{\ 2} + z_{NED}^{\ 2}} \\ \arctan\left(\dfrac{y_{NED}}{x_{NED}}\right) \\ -\arcsin\left(\dfrac{z_{NED}}{\sqrt{x_{NED}^{\ 2} + y_{NED}^{\ 2} + z_{NED}^{\ 2}}}\right) \end{bmatrix} + \underline{w} \qquad (18)$$

and can be linearized around the latest state prediction. Thus, in the EKF case H has to be evaluated in real time, while R is constant and assigned off-line on the basis of sensors specifications.

Linearization can be performed on the basis of predicted state also in case of EO angular-only measurements: thus, introduction of EO sensors is made in EKF more simply.

A totally different choice is to operate in spherical coordinates. It is worth noting that these coordinates are not the sensors' coordinates, since it is supposed that tracking is performed in NED in any case. Nevertheless, the relevant advantage is in the linear measurement equation, both for radar and EO sensor measurements. However, since relative dynamics, even in non-maneuvering phases, is characterized by strong non linearities in spherical coordinates (the so-called "pseudo-accelerations", [57]), estimates of high order derivatives of the angles would be required. Following Blackman [56], it is possible to derive spherical filters that attenuate these effects, but it is worth noting that they grow for decreasing obstacle distance, since angular variations are very fast when intruder distance is, for example, less than 1 km. Usual implementations foresee the use of spherical coordinates for large target distances [65].

The most important feature of spherical filters is that, under reasonable assumptions (essentially, elevation angle must be sufficiently small) it is possible to decouple the filters for range, azimuth and elevation, obtaining a lighter algorithm from the computational point of view. In any case, range and angle filters have to exchange their estimates for range and angle extrapolation.

Detailed derivation of spherical filters for airborne tracking is rather complex and is reported in [56]. The range filter uses range, range rate, and range acceleration as states, and the Singer model can be used for range acceleration. Own-ship acceleration is included in the model by projection in the RHV reference frame. The latter is a target-dependant reference frame with

one axis along the range vector, and the other two axes perpendicular to the range vector and in the horizontal and vertical plane, respectively (figure 3).

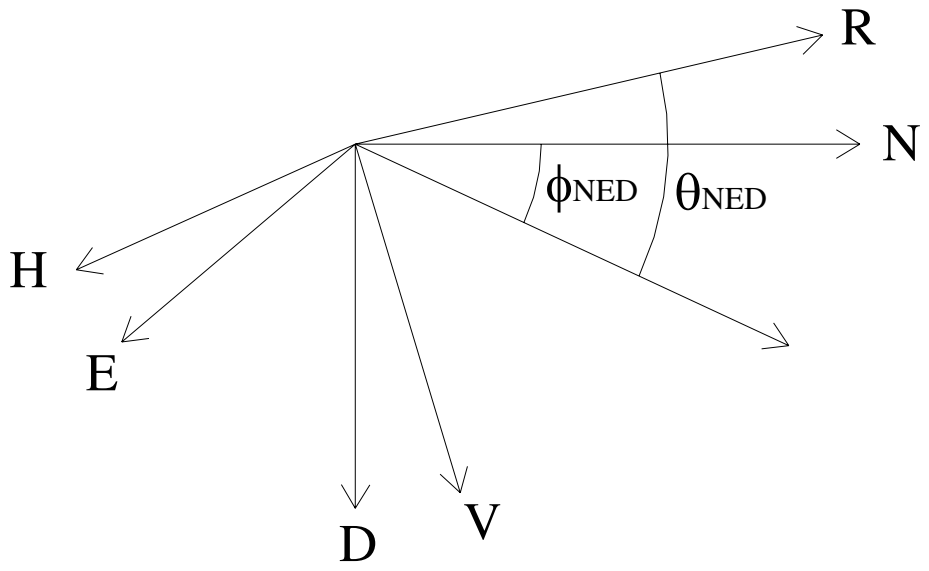

Figure 3 – Definition of RHV reference frame

RHV reference frame also allows to define the angle filters. In fact, the state components are not the stabilized angles with their time derivatives, but, in the case of stabilized azimuth for example

$$\underline{x}_{\varphi} = \begin{bmatrix} \varphi \\ v_H \\ a_H \end{bmatrix} \tag{19}$$

where v_H and a_H are, respectively, the relative velocity component perpendicular to the line of sight along the H axis of the RHV reference frame, and the horizontal relative acceleration. This choice allows for reduction of pseudo-acceleration effects.

Analyzing the structure of spherical filters, it comes out that some information must be exchanged among range and angle filters in order to propagate system dynamics. In fact, as shown in equation (20), range filter transition matrix contains target angular rate perpendicular to its line of sight (ω_p), which can be estimated on the basis of angle filters.

$$\Phi_R = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ \omega_p{}^2 T & 1 + \frac{\omega_P{}^2 T^2}{2} & T\left(1 - \frac{\mu_R T}{2}\right) \\ 0 & 0 & \rho_{aR} \end{bmatrix} \tag{20}$$

In equation (20), $\rho_{aR} = \exp(-\mu_R T)$ where μ_R is the inverse of maneuver time constant for range dynamics.

On the other hand, propagation of angle filters requires (again in the transition matrix) estimates of range and range rate, which are provided by the range filter. For example, transition matrix for the azimuth filter is reported in equation (21)

$$\Phi_H = \begin{bmatrix} 1 & \frac{T}{R_H} C_R & \frac{T^2}{2R_H} \\ 0 & \rho_R & T\left(C_R - \frac{T\mu_H}{2}\right) \\ 0 & 0 & \rho_H \end{bmatrix} \tag{21}$$

where

$$\begin{aligned} C_R &= 1 - \frac{v_R T}{2R}, R_H = R\cos\vartheta_{NED} \\ \rho_R &= \exp\left(-\frac{v_R T}{R}\right), \rho_H = \exp(-\mu_H T) \end{aligned} \tag{22}$$

Moreover, Q matrices for azimuth and range have to be amplified for decreasing range, in order to take into account faster angular dynamics.

For each of the three filters, the measurement equation is linear with

$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{22}$$

and the definition of the three R matrices is straightforward.

The computational advantage of the spherical filters relies on the fact that the three filters require handling and inversion of 3 X 3 matrices instead of a 9 X 9 matrix as before. However, unlike Cartesian filters these schemes require real time adjusting of the dynamic model, in terms of transition and noise matrices.

On the other hand, it is worth noting that filter decoupling in the Cartesian coordinates case can be obtained only paying the cost of degraded performance, which is essentially due to the fact that the measurement errors in Cartesian coordinates are coupled even though radar errors on range and angles are independent.

These filtering/prediction schemes have been tested in order to understand their advantages and drawbacks in view of the considered application. The results are reported in what follows.

3.4.4. Inclusion of navigation uncertainties in track updating and gating

Navigation estimates are in general very important for tracking, since they allow for own-ship motion compensation. Considering the mathematical structure of the filters introduced above, in particular, acceleration estimates are used for deterministic input, velocity estimates are used in track initialization phase, while angular velocity estimates are not used: their only function is to provide azimuth and elevation rate estimates in Body Reference Frame, starting from predicted angular rates in NED. Though attitude angles do not appear explicitly in the algorithms, they are the most important navigation parameters: in fact, as already said they are used for transformation of sensor raw measurements in NED. Attitude angles estimation is basically affected by a slow time-varying bias, and a noise. Of course, biases can not be removed and will affect tracking accuracy in NED. As for noise, it is important to point out that it must be accounted for in the tracking algorithm since it affects measurement covariance matrix. Otherwise, a too optimistic R estimate can result in a too optimistic P estimate, with the result of generation of biases in tracks. Moreover, the gating process can be negatively affected by these errors: the gate can become too small preventing from correct association and track updating.

Estimation of angular errors in NED based on uncertainty in azimuth and elevation in BRF, and on uncertainty in heading, pitch and yaw, can be performed analytically by first order approximations.

In fact, transformation from BRF to NED is performed in the following steps:

- ϕ_{BRF} and θ_{BRF} are used to evaluate components of target unit vector in BRF

$$\begin{cases} \Lambda_1 = \cos\vartheta_{BRF}\cos\varphi_{BRF} \\ \Lambda_2 = \cos\vartheta_{BRF}\sin\varphi_{BRF} \\ \Lambda_3 = -\sin\vartheta_{BRF} \end{cases} \tag{23}$$

- the vector is converted in NED through the attitude matrix

$$\begin{bmatrix} \Lambda_N \\ \Lambda_E \\ \Lambda_D \end{bmatrix} = T_{BRF\to NED} \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{bmatrix} \tag{24}$$

- the new unit vector components are used to calculate ϕ_{NED} and θ_{NED}

$$\begin{cases} \varphi_{NED} = \arctan\left(\dfrac{\Lambda_E}{\Lambda_N}\right) \\ \vartheta_{NED} = -\arcsin\Lambda_D \end{cases} \tag{25}$$

Analytical computation of angular errors in NED is based on these relations and for the sake of brevity is not reported; what is important is that errors depend not only on the uncertainties, but also on the target position and aircraft attitude.

In theory, this would require that measurement uncertainties are computed in real time during track propagation, whatever filtering scheme is used. However, uncertainty variation is rather small, so that the approach for implementation has been to amplify R entries by fixed coefficients, which were evaluated off-line by means of extensive Monte Carlo simulations.

3.5. Numerical analysis of different intruder state estimation algorithms

A numerical analysis has been carried out in order to point out the main differences in state estimation algorithms performance [72]. In particular, the three filters described above have been tested in a simplified dynamic environment. Since the primary interest was to establish the differences between different choices of tracking filters, no maneuvering handle logic (like Interacting Multiple Models, for example [46]) has been applied. Moreover, this choice is related to the necessity of keeping a light computational load, as it will be further explained in the following.

The following simplified dynamic scenario has been chosen: UAV moves towards North with constant speed (40 m/s) and incidence angle (2°), and unperturbed attitude, whereas a flying obstacle is moving with almost the same speed (35 m/s) southward. There is no altitude separation between the two airplanes, and, in East direction, there is a constant offset equal to 165 m, that is the previously recalled bubble distance. This can be considered as a near collision scenario, since the minimum distance between airplanes equals the bubble distance. The geometry is illustrated in figure 4.

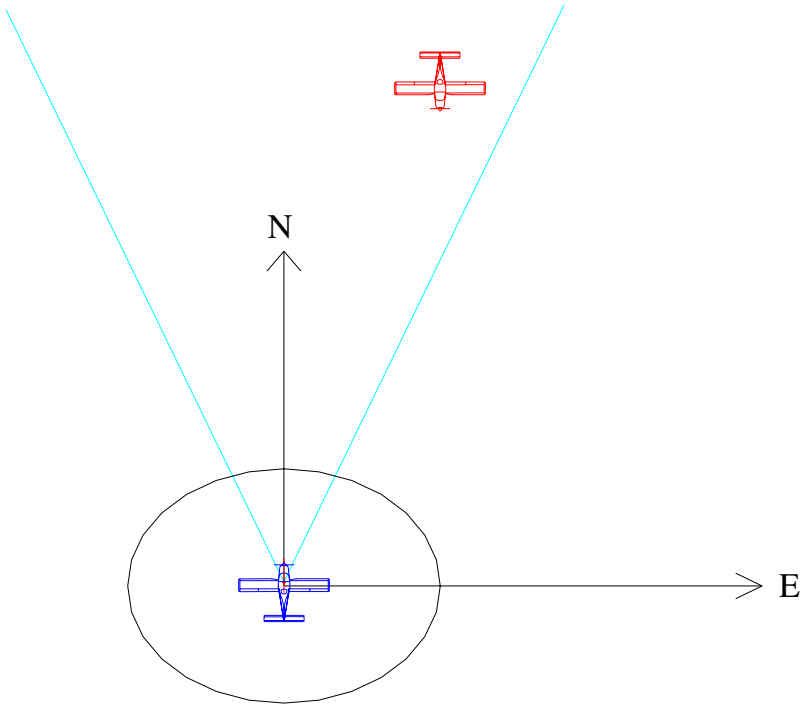

Figure 4 – Simulated near collision scenario

Both the UAV and the intruder aircraft are considered as Very Light Aircrafts (VLA), with a mean Radar Cross Section of 1 m^2, and they are both Swerling 2 targets (propeller-driven airplanes) [73]. Swerling obstacle categories are further clarified in the following when describing the radar detection process simulator.

As for navigation data, measures have been simulated by corrupting real values with a constant bias and a white zero mean Gaussian random noise. These parameters are reported in Table 1:

Parameter	Bias	Noise standard deviation
Heading angle error	1°	0.15°
Pitch angle error	1°	0.15°
Roll angle error	1°	0.15°
Acceleration components' error (BRF)	0.1 m/s^2	0.1 m/s^2
Angular velocity components' error (BRF)	$1 \cdot 10^{-3}$ rad/s	$5 \cdot 10^{-3}$ rad/s

Table 1 - Biases and noises of navigation errors

The tracking filters are all designed to produce estimates at a frequency of 10 Hz, consistently with system requirements for the anti-collision system. As for obstacle detection sensors, in this numerical analysis it is supposed to have a pulse radar (range rate not available), plus an EO system. Radar range accuracy is 15 m, and its angular accuracy is 2° both in elevation and in azimuth, data rate is 1 Hz. These data are consistent with Amphitech™ OaSYS™ radar specifications. The radar detection process is simulated according to references [74, 75] through a radar detection process simulator which is briefly described in the next chapter. The radar Field Of View (FOV) dimensions are 120° in azimuth and 20° in elevation, whereas the EO FOV is supposed to be 50° in azimuth and 40° in elevation.

Due to the difficulty in generating simulated images of the scenario as seen by the EO sensors, the latter have been simulated in a simplified way. In fact, it is supposed that the EO system gives angular measures with an accuracy of 0.1° (in the BRF) at a frequency of 2 Hz, when intruder distance is less than 3 km. It is worth pointing out that electro-optical sensors usually work at higher frequencies, nevertheless a low frequency was chosen to account for the time necessary for image processing, and to show the large improvement of tracking performance that is obtained when accurate angular measures are achievable, even at this

frequency. Sensors' mounting misalignment error is neglected. Details about obstacle detection sensors simulation are summarized in table 2.

DS&A Sensors	*Detection range*	*σ range (m)*	*σ angles (BRF) (°)*	*FOV azimuth (°)*	*FOV elevation (°)*	*Data rate (Hz)*
radar	>3 km	15	2	120	20	1
EO	3 km	/	0.1	50	40	2

Table 2 – Obstacle detection sensors simulation parameters

The selected scenario allows to test the tracking filters in different phases. At first, the intruder is detected by the radar only, without detection misses, thus the system fast switches to firm tracking. After about 26 s, due to decreasing range (figure 5) the obstacle is detected by both sensors in some instants, by the EO system only in others. Intruder lies in the radar FOV for 65 s, then it is no more detectable because azimuth angle becomes larger than 60° (figure 6).

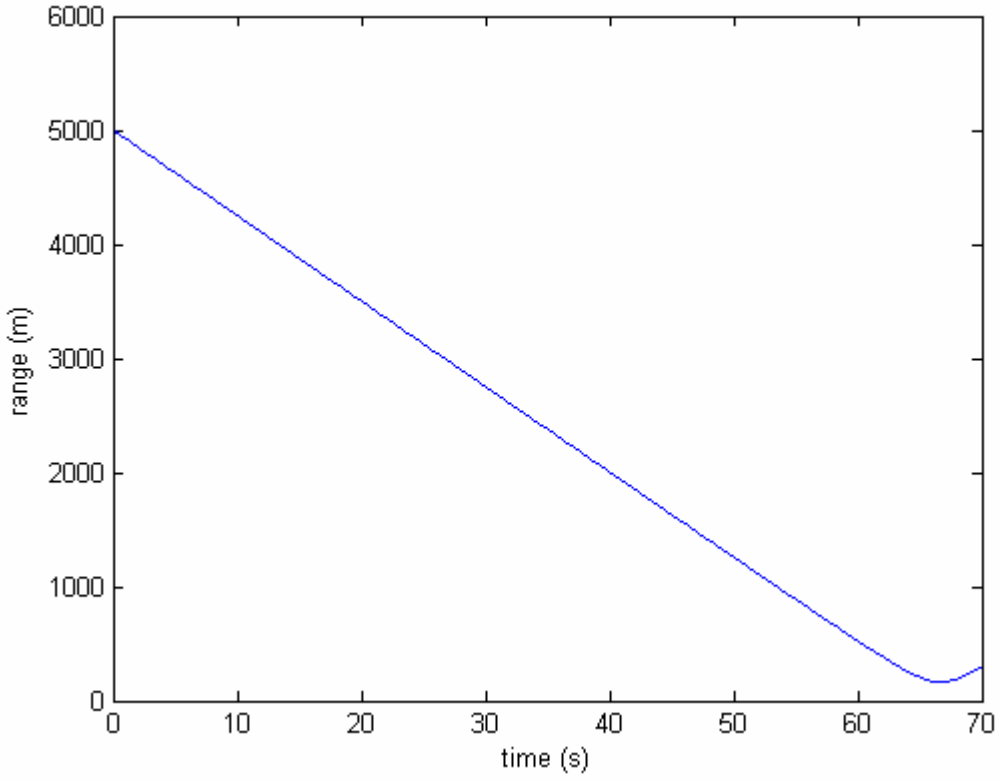

Figure 5 – Range as a function of time in the considered scenario

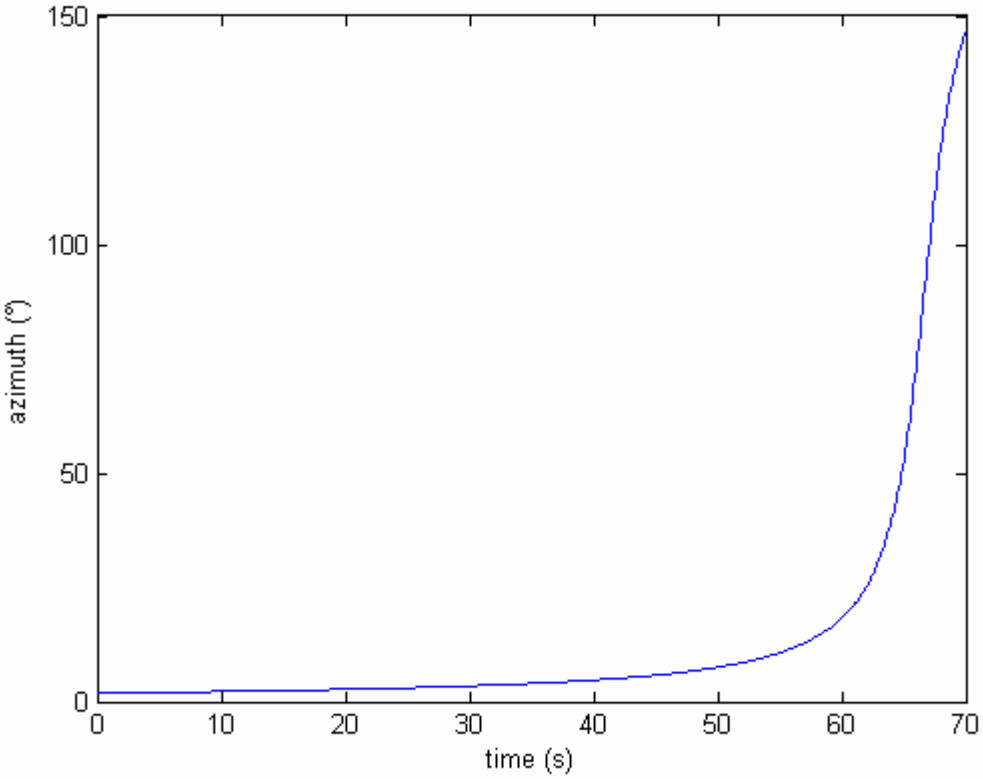

Figure 6 – Azimuth angle as a function of time in the considered scenario

Due to the non deterministic nature of sensors detection, both for radar and EO cameras, a statistical approach is needed to assess algorithm performance. Thus, for this scenario 100 Monte Carlo simulations have been performed regarding the generation of radar, EO and navigation measures, and then tracking filters have been applied to their output.

Ellipsoidal gating (see next section) was taken into account in these simulations: that is, track was updated only if measurement satisfied gate. It is worth noting that no gating problems were observed, which is indirectly a confirmation of prediction and measurement covariance matrices consistency.

Next figures report estimation errors for the three filters, considering both mean and standard deviation values, as a function of time to collision. Collision is considered to happen when intruder distance becomes equal to the bubble distance. In particular, range error is reported in figures 7-9, azimuth error is reported in figures 10-12, elevation error is reported in figures 13-15.

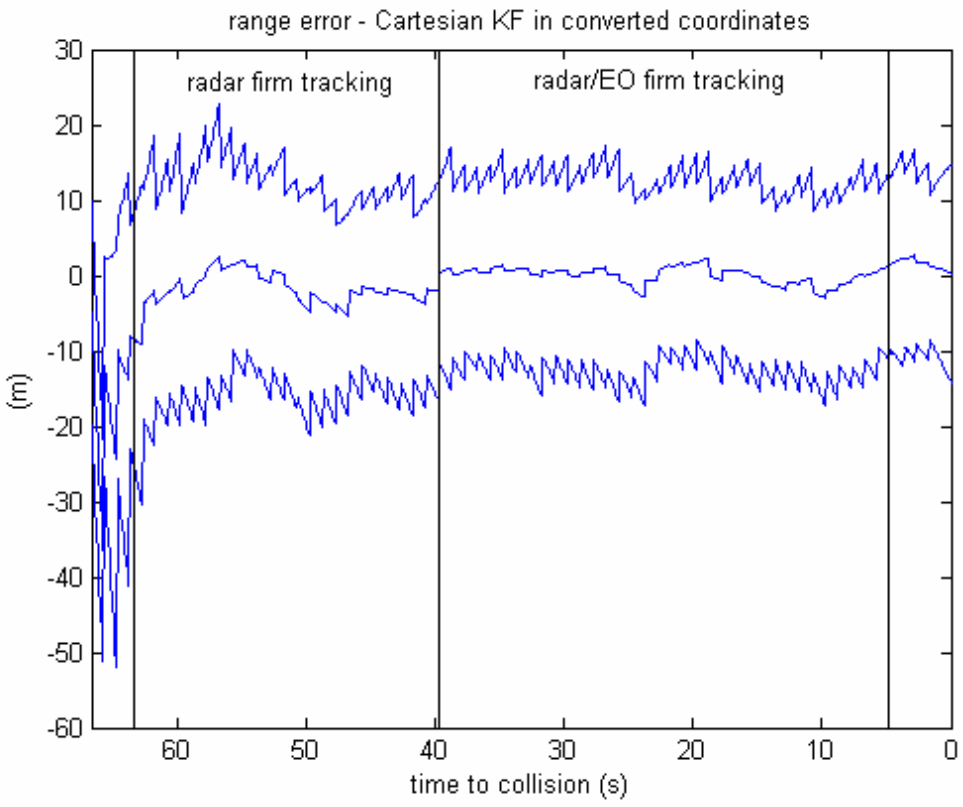

Figure 7 – Error in range estimate for Cartesian filter, in terms of mean and standard deviation

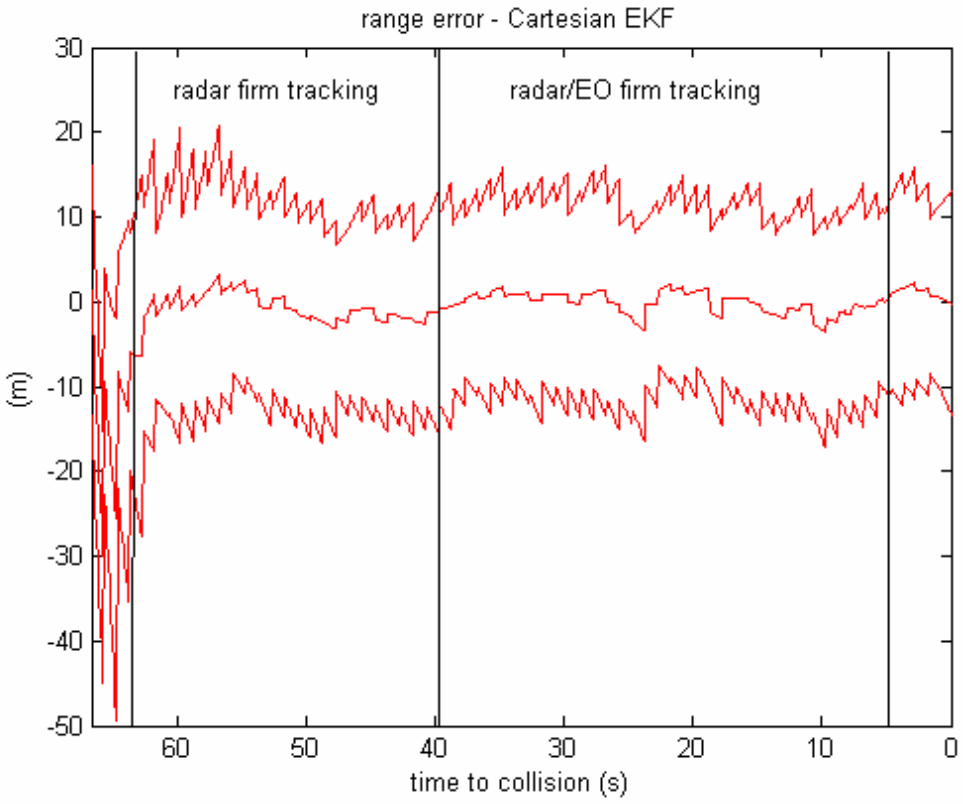

Figure 8 - Error in range estimate for Cartesian EKF, in terms of mean and standard deviation

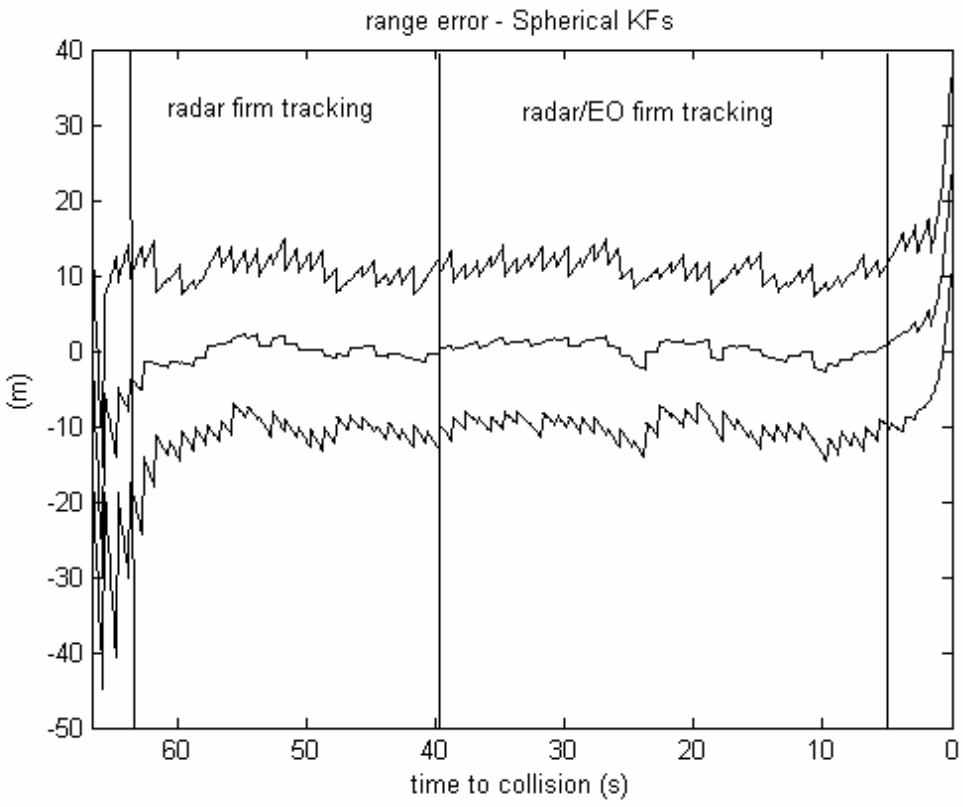

Figure 9 - Error in range estimate for spherical filters, in terms of mean and standard deviation

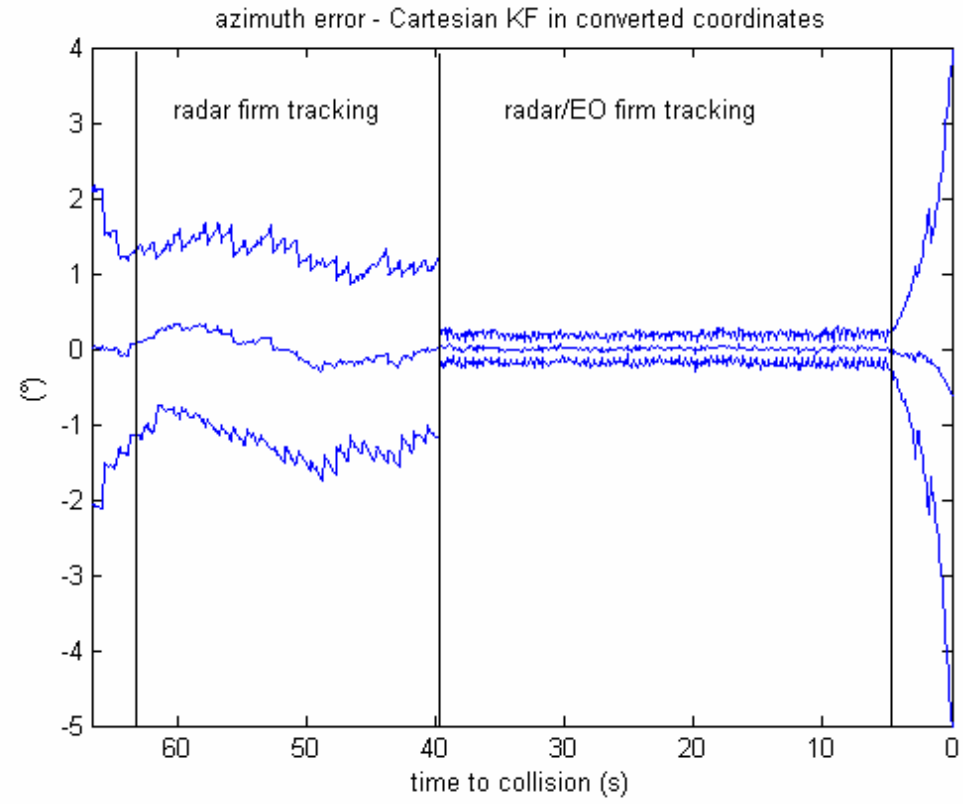

Figure 10 - Error in azimuth estimate for Cartesian filter, in terms of mean and standard deviation

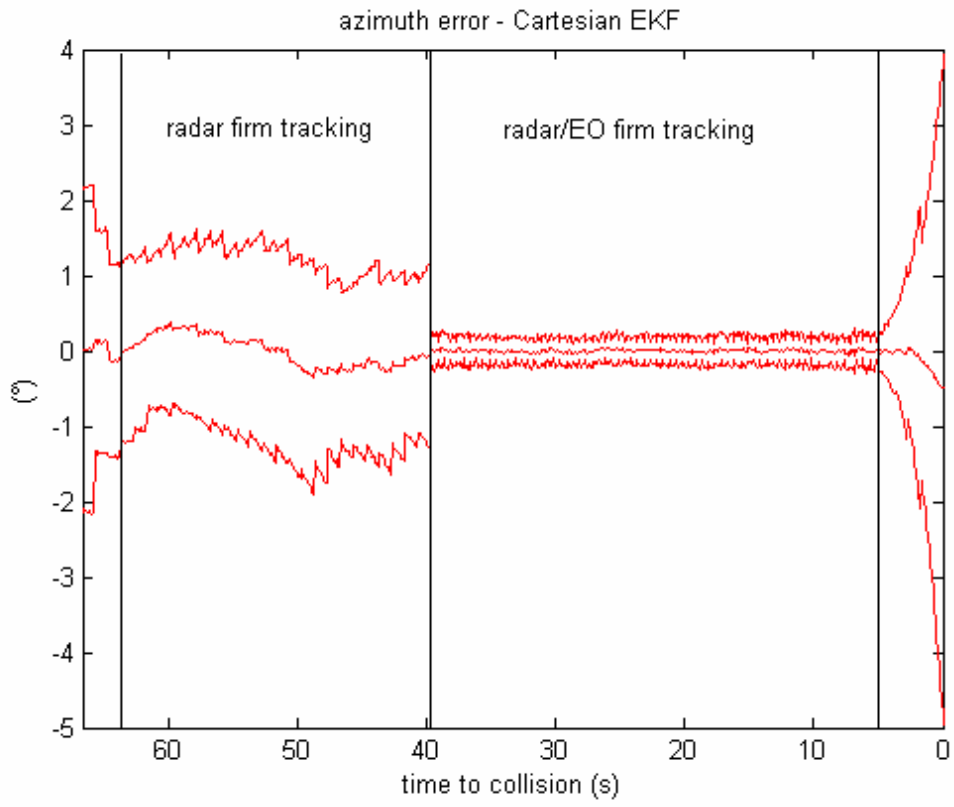

Figure 11 - Error in azimuth estimate for Cartesian EKF, in terms of mean and standard deviation

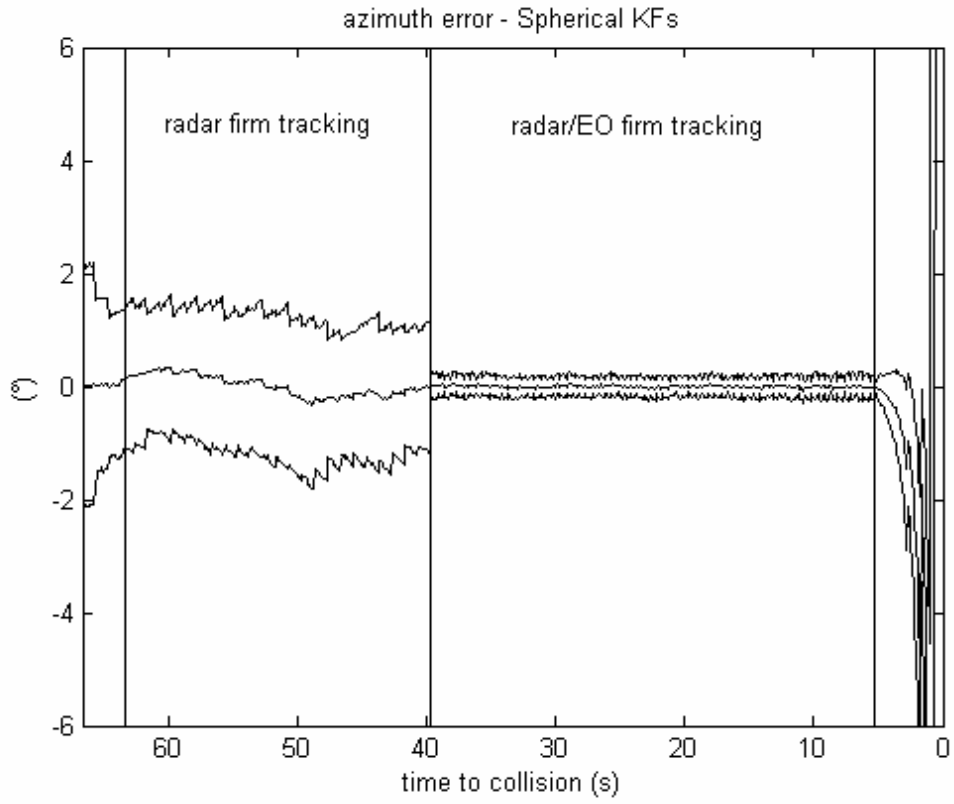

Figure 12 - Error in azimuth estimate for spherical filters, in terms of mean and standard deviation

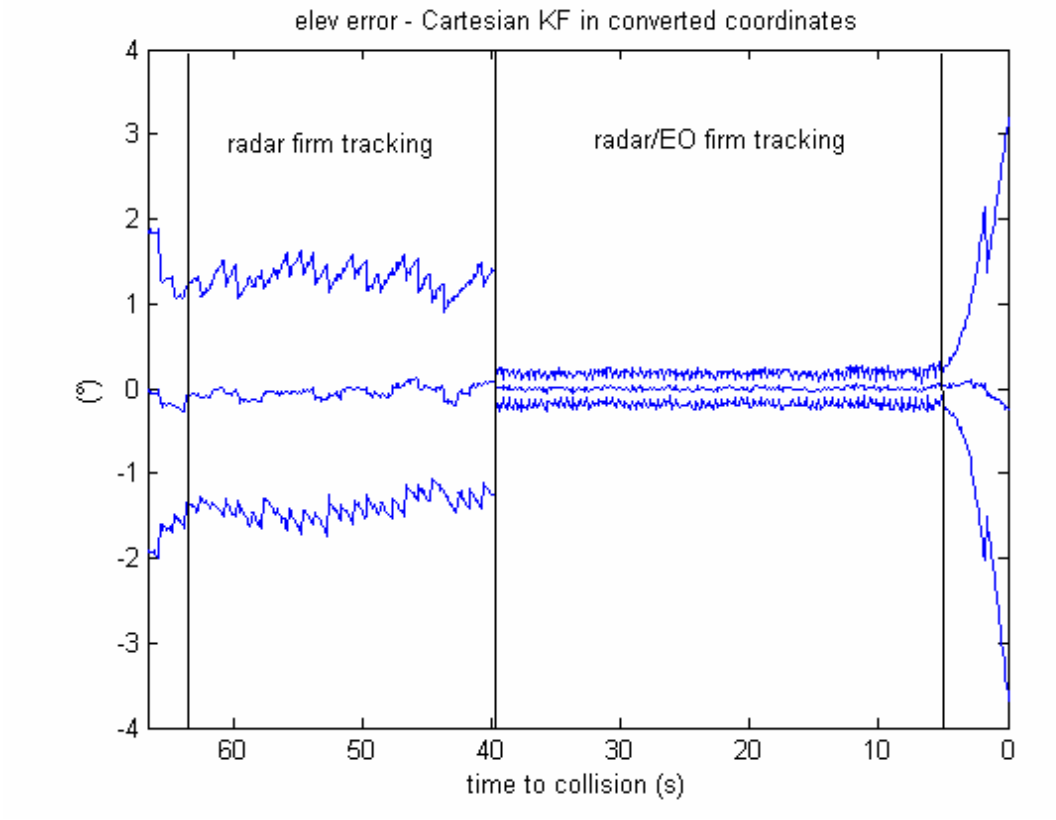

Figure 13 - Error in elevation estimate for Cartesian filter, in terms of mean and standard deviation

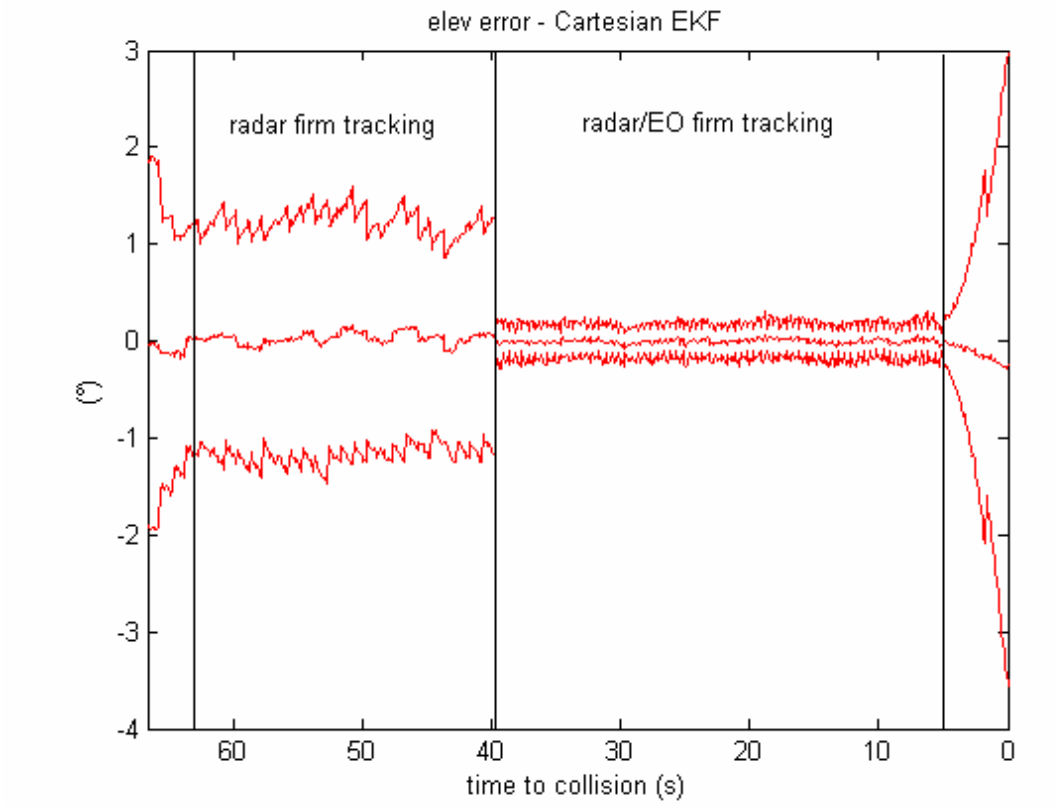

Figure 14 - Error in elevation estimate for Cartesian EKF, in terms of mean and standard deviation

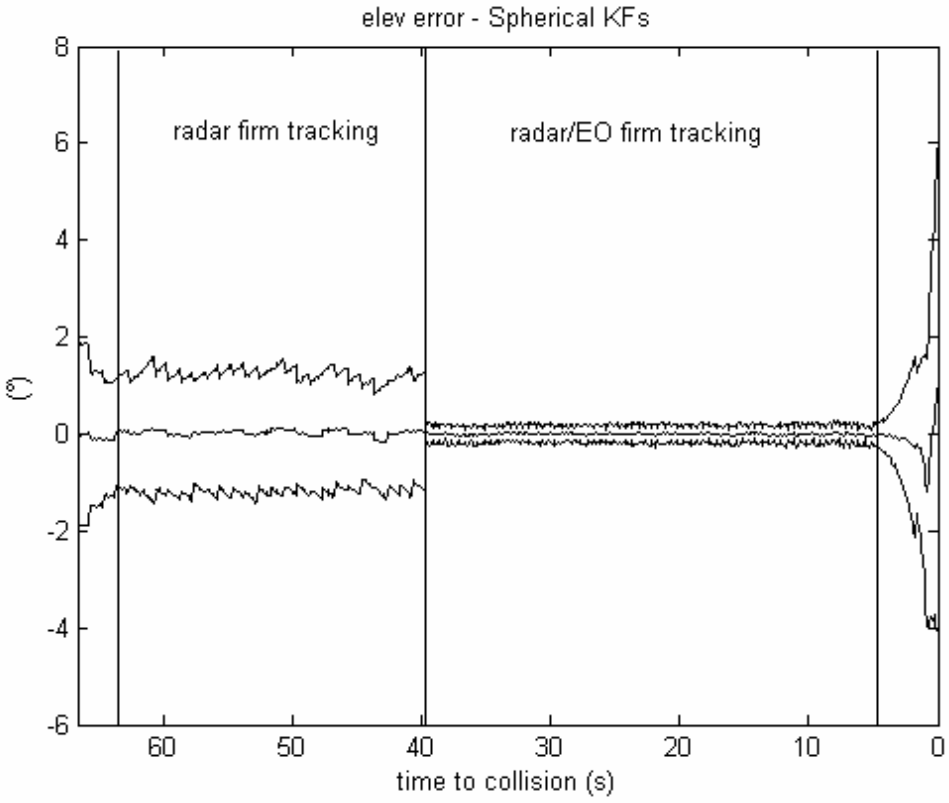

Figure 15 - Error in elevation estimate for spherical filters, in terms of mean and standard deviation

The analysis of these figures allows to appreciate that the three filtering schemes exhibit similar performances in terms of global accuracy. The different modes of the tracking algorithm are put into evidence by vertical separation lines. When firm radar tracking is activated, the error decreases for decreasing intruder distance, which is due to the fact that angular errors are stable. However a great drop of the positioning error is achieved only when the fusion radar/EO is activated. This happens whatever filtering scheme is used, and is due to the possibility to use accurate angular measures, even at relatively low frequencies. In fact, it is worth underlining that angular estimates are produced at a frequency of 10 Hz, while EO measurements are simulated at a frequency of 2 Hz. On the other hand, range error is not influenced so much by EO activation.

Though some differences in tracking accuracy can be noted in firm radar and firm radar/EO phases, the difference between the filters appears only when intruder distance is really low. In fact, at this point the high angular rates (especially in azimuth), combined to the fact that spherical filters tend to be less precise in absence of frequent updates, produce a greater error

for the latter ones. In particular, figure 12 puts into evidence unstable behavior in azimuth estimation process.

Instability in azimuth estimates is reflected also in the range and elevation filters (figures 9 and 15), because of the exchange of information on which state prediction is based in spherical filters.

It is worth noting that the considered situation is a limit case, with a time to collision of a few seconds and no avoidance maneuvers. However, it is evident that spherical filters are less reliable at very low distances, or in case of detection misses.

A different behavior is shown in both Cartesian filters. In fact, an increase in error standard deviation is observed, which is due to the lack of measurements by the sensors (the obstacle goes out of their FOV). However, in these cases, average range and angular errors are very stable around zero even in the last seconds before collision. This result is due to the fact that the fast angular dynamics due to relative motion does not affect at all these algorithms, since the state vector is expressed in terms of relative coordinates and not angles.

Considering the case of the Cartesian filters with de-biased converted measurements, no unstable phenomena were observed when updating tracks only by EO measures. It is useful to point out that, to be more conservative, the accuracy estimate for predicted range was taken to be constant and equal to 12 m, that is of the order of the estimated standard deviation of range error in the one sensor configuration. However, as already said the reliability of this procedure, which directly merges prediction and measure, still needs to be verified.

In summary, the best compromise between reliability, computational load and accuracy seems to be the Extended Kalman Filter, as it results considering mean obstacle positioning error, which is reported in figure 16 for the three filters. Thus, the EKF in Cartesian coordinates has been selected for real time implementation and is considered in the following.

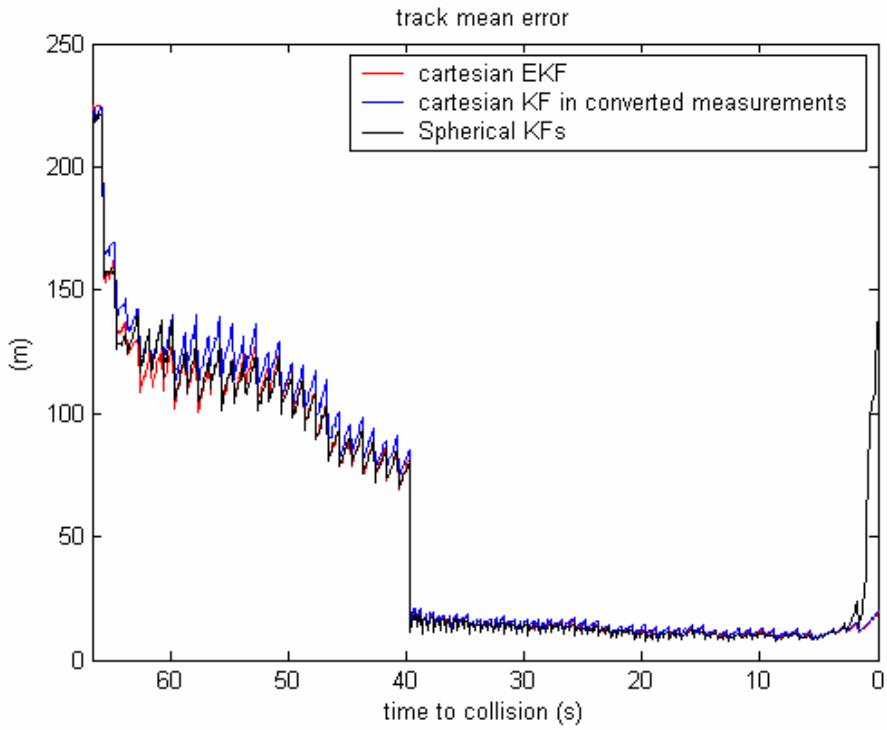

Figure 16 – Positioning mean error as a function of time for the different considered filters

3.6. Further elements and considerations about tracking algorithm

The previous study allowed to identify potential choices for the obstacle kinematic state estimation algorithm. This section reports design choices relevant to the other tracking algorithm modules.

First of all, ellipsoidal gating has been considered for the anti-collision system. This means that, in order to perform measurement-to-track association, the following decision statistics is considered:

$$\xi = \left[\underline{y} - \hat{y}\right]^{|} \left[R + HPH^{|}\right]^{-1} \left[\underline{y} - \hat{y}\right] \tag{26}$$

where

$\underline{x}$ is the track state prediction at time of detection, P is the predicted state estimation covariance matrix at time of detection, and $\hat{y}$ is predicted measurement at time of detection on the basis of predicted state and measurement equation.

This is a normalized distance between measurement and prediction. Normalization is made on the basis of measurement and prediction uncertainties (that is, covariances). This answers to the very intuitive principle that the more the track and the measurement are believed to be accurate, the more the measurement-to-track distance will be expected to be small. This also implies that an excessively optimistic covariance estimate (both for state and measurement) can be dangerous and can definitely lead to track deletion. The statistic ξ is distributed as a chi-squared normal variable with a number of degrees of freedom that equals the dimension of the measurement vector. Thus, in the considered system (radar measurement) n=3. The upper limit for the statistic can be defined to yield a given (low) probability of rejecting the correct measurement when present. Assuming for example this probability equal to 0.01, from chi-square tables the gate limit is 11.34. This limit establishes in the state space an ellipsoidal region, hence the definition of ellipsoidal gating.

As for association, given the civil scenario considerations (section 2) and the sense and avoid flight tests planning (first experiments with a single intruder in UAV field of regard), it is believed that it can be kept at a very simple level in the on-board software first version. In fact, it will be based on nearest neighbor logic.

Another point to be analyzed is the possibility to use more complex dynamic models, not only based on *a priori* information. Singer model, as already stated, has been considered as a suitable choice for the application since it covers a wide range of obstacle dynamics. Of course, as already stated before its accuracy depends on input parameters for the accelerations, which basically establish the filter bandwidth. This is clearly a non-optimal choice if obstacle behavior changes a lot with time. In this case, different approaches can be followed, such as state augmentation or system noise adaptation [56], but the most effective solution is to use a multiple model approach, in particular the Interacting Multiple Model (IMM) estimator [44].

IMM is based on a bank of Kalman filters running in parallel. Intruder state estimation can be based on a weighted average of estimates from the different models, where the weights are

evaluated on-line on the basis of different modes probabilities (i.e., on the basis of sensor measurements). In IMM implementation, it makes sense to use dynamics models which are sufficiently different from each other, so that it is easier to establish which one is better related to measurements. The obvious drawback of IMM, compared to classical single Kalman filter implementation, is in the computational weight.

Since all the study described in this thesis was focused on real time implementation and flight demonstration, IMM has not been implemented on the on-board unit. Nevertheless, a research effort to establish which advantage can derive in the collision avoidance application from IMM usage is considered among future developments.

Chapter 4

Tracking system performance assessment

4.1. Introduction

The analysis proposed in the previous chapter allowed to identify candidate solutions for multiple sensor integration in the tracking module. In particular, the Extended Kalman filter in Cartesian NED coordinates was found to be a reasonable compromise in view of the collision avoidance application. In this chapter, a detailed analysis about EKF behavior in real collision avoidance scenarios will be proposed [76].

This numerical study was carried out in collaboration with the Flight Systems Department at CIRA, which is responsible for development of collision avoidance logic and software. Thus, CIRA provided the collision scenarios to be given as input to the tracking algorithm.

Among the scopes of this activity, one was to establish tracking sensitivity to different avoidance maneuver choices. In fact, obstacle avoidance can be performed in different ways, depending on how aggressive is the UAV maneuver. Aggressive maneuvers are basically characterized by a larger attitude and position dynamics, which directly impacts the sensor system which is, of course, body fixed.

On the other hand, avoidance can be obtained by gradual small changes in trajectory (and attitude). Moreover, maneuvers can be performed in the horizontal plane, in the vertical plane, or in a combination of the two. The performed numerical analysis allowed to point out the main differences among the different maneuvers from the tracking point of view.

Another goal of the study was to provide a realistic assessment of tracking system performance, to be also provided as input for Monte Carlo simulations dealing with avoidance maneuvers execution. What has been obtained can be considered as a realistic estimate of tracking system accuracy in absence of real time implementation issues such as latencies in navigation data exchange and sensor measurements transmission. Nevertheless, as it is explained in the following chapter, the on-board system has been designed to minimize these effects.

4.2. Simulation software environment

Obstacle detection and tracking performance was tested following the line of reasoning shown in figure 1. It also reports the software tools which were used.

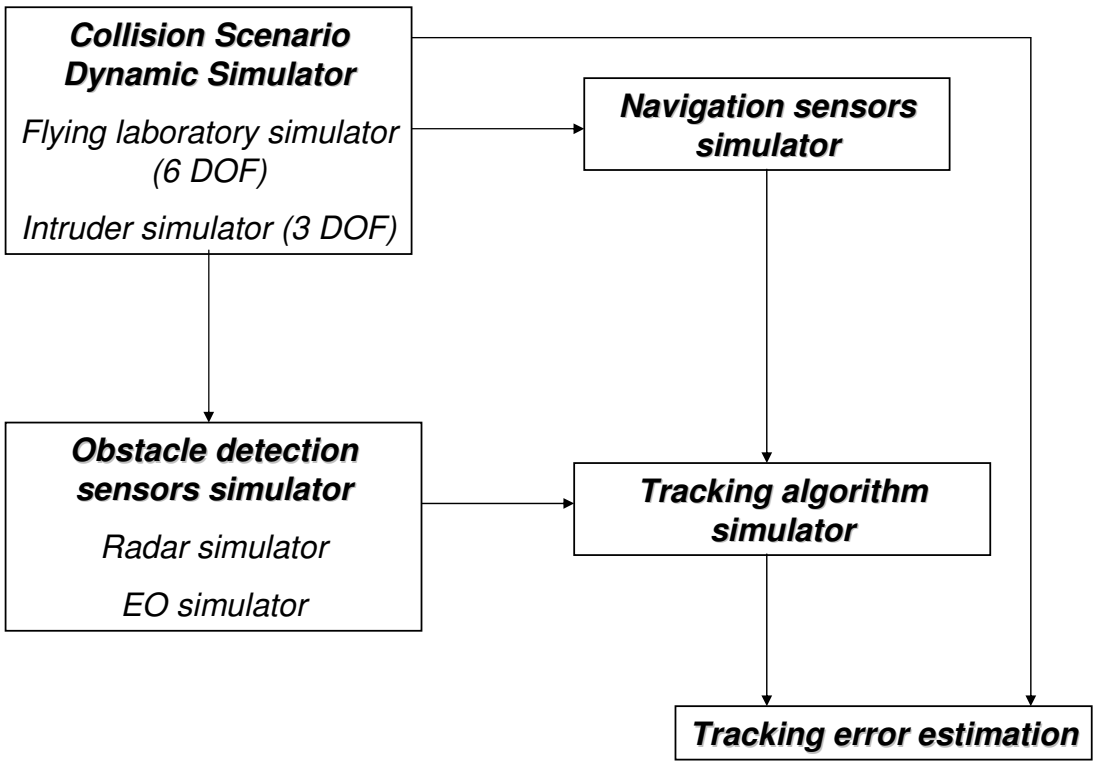

Figure 1 – Logic of tracking performance estimation

Given the off-line nature of the simulation, all the tools were developed in the Matlab™/Simulink™ environment. The collision scenario dynamic simulator includes the flying laboratory simulator and the intruder simulator. The first is based on a 6 Degrees of Freedom (DOF) dynamic model with related aerodynamic and inertial uncertainties, and the outside environment with Von-Karman turbulence and wind-shear models. The intruder dynamics is based on a 3 DOF model which however takes into account some dynamic limitations.

Navigation sensors have been simulated on the basis of referenced models [77]. In particular, AHRS and GPS measurements have been simulated by considering the lever arm effect and sensors' biases and noises. The latter have been assigned based on datasheets for the sensors installed on-board FLARE [78, 79].

Regarding obstacle detection sensors, as already stated in the previous chapter the EO system has been simulated in a simplified way by assuming a given angular accuracy and detection range. The radar sensor, instead, has been simulated by computing probability of detection for the target as a function of its range from the UAV. Of course, since the detection process is intrinsically non deterministic, a Monte Carlo approach is required for radar simulation.

Radar simulator computes obstacle probability of detection if the intruder lies in the radar Field of Regard. In this case, first of all radar equation allows to evaluate one pulse signal to noise ratio (SNR).

Radar equation can be expressed as follows:

$$\left(\frac{S}{N}\right) = \frac{P_t \cdot G_t \cdot G_r \cdot \tau \cdot \lambda^2 \cdot \sigma}{1.2 \cdot (4\pi)^3 \cdot k \cdot T \cdot F_n \cdot L \cdot R^4} \tag{1}$$

where the following parameters depend on the sensor:

- P_t: peak transmitted power;
- G_t, G_r: antenna gain in transmission/reception;
- λ: wavelength;
- τ: impulse duration;
- F_n: receiver noise figure.

L represents atmospheric losses, k is the Boltzmann constant, R represents range and σ is the obstacle radar cross section (RCS).

Regarding the latter, it is important to point out that the value in (1) is, in statistical terms, the RCS mean value: RCS, in fact, is an aleatory variable. RCS statistical properties do not influence one pulse SNR, but do influence probability of detection.

Radar targets can be classified on the basis of the classical Swerling classification [73], according to which they can be divided in five categories, as reported in table 1:

Swerling case	**Scan to scan fluctuations**	**Pulse to pulse fluctuations**	**Example**
0	No	No	Sphere
1	Yes-Complex	No	Jet Aircraft
2	Yes-Complex	Yes	Propelled Aircraft
3	Yes-Simple	No	Small missile
4	Yes-Simple	Yes	Small prop missile

Table 1 – Swerling classification

Swerling type influences probability of detection since typically radars can integrate consecutive impulses in order to increase this probability: thus, RCS autocorrelation properties influence the probability density for the receiver output, which is made by signal plus noise. Probability of detection can be evaluating integrating the (signal+noise) probability density function from the threshold to infinite. Threshold value depends on the assumed false alarm probability. When probability of detection is computed, detection is simulated by extracting a random number from the [0,1] uniform distribution. In case of detection, measurement is generated by corrupting true range and angles with an additive white Gaussian noise with zero mean and a given standard deviation. In particular, in the considered case, range standard deviation is considered to be 15 m, while angular accuracy is 2°. It is worth noting that more accurate models of radar measurement generation exist [65,80], where measurement accuracy is expressed as a function of SNR. However, the adopted choice can be considered conservative (also in view of the collision avoidance application) and in fact it is typically adopted in simulation for tracking systems. Figure 2 shows the flux diagram for the radar simulator.

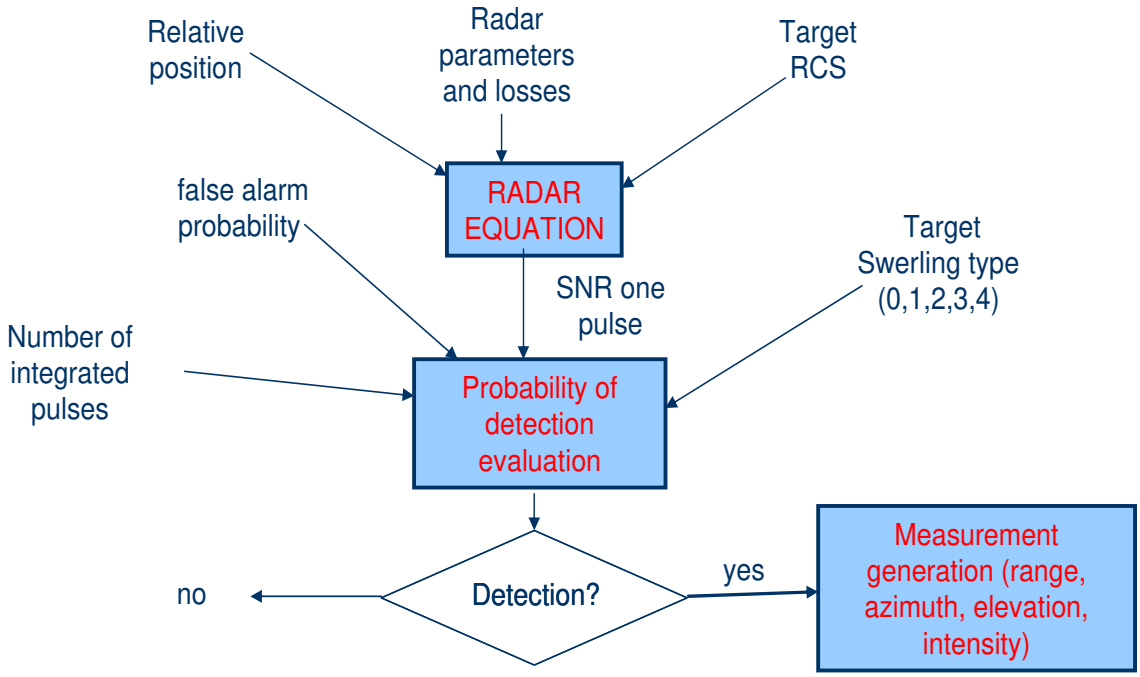

Figure 2 – Flux diagram for the radar simulator

Key parameters for sensors simulation have already been reported in chapter 3 (table 2). It is worth recalling that, in view of planned flight tests, the intruder has been simulated as a Swerling 2 target (propeller-driven airplane) with a mean Radar Cross Section of 1 m^2.

Both for radar and EO cameras, sensors' mounting misalignment error has been neglected. However, a calibration phase is foreseen before of the first collision avoidance flight tests and has already been positively tested, showing the capability to estimate cameras' alignment with respect to the AHRS with a residual error of the order of 0.3°. This alignment technique is detailed in the following.

Also, cameras' vibration has been neglected, due to the fact that EO images taken during first preliminary flights devoted to system identification and EO sensors functional verification have shown that image quality is not affected by these phenomena.

4.3. Collision scenarios

Ten collision scenarios have been generated as follows:

1. The two airplanes fly at an altitude of 300 m, initial distance among them is 3.5 nm and they are moving towards each other. The UAV avoids collision by means of a maneuver in horizontal plane near the safety bubble (figure 3).

2. The two airplanes fly at an altitude of 300 m, initial distance among them is 3.5 nm and they are moving towards each other. The UAV avoids collision by means of an aggressive maneuver in horizontal plane with maximum roll angle (40°), changing track angle of 90° with respect to nominal trajectory.

3. The two airplanes fly at an altitude of 300 m, initial distance among them is 3.5 nm and they are moving towards each other. The UAV avoids collision by means of a maneuver in vertical plane near the safety bubble.

4. The two airplanes fly at an altitude of 300 m, initial distance among them is 3.5 nm and they are moving towards each other. The UAV avoids collision by means of an aggressive maneuver in vertical plane, changing slope angle of 8° with respect to nominal trajectory.

5. The two airplanes fly at an altitude of 300 m, initial distance among them is 3.5 nm and they are initially moving on parallel trajectories. After 28.4 s the intruder performs a maneuver in the horizontal plane, changing its direction of 20 ° towards the UAV. The latter avoids collision by means of a maneuver in the horizontal plane near the safety bubble (figure 4).

6. The two airplanes fly at an altitude of 300 m, initial distance among them is 3.5 nm and they are initially moving on parallel trajectories. After 28.4 s the intruder performs a maneuver in the horizontal plane, changing its direction of 20 ° towards the UAV. The latter avoids collision by means of an aggressive maneuver in the horizontal plane. In fact, UAV

performs a turn with maximum roll angle (40°) changing its direction of 90° with respect to nominal trajectory.

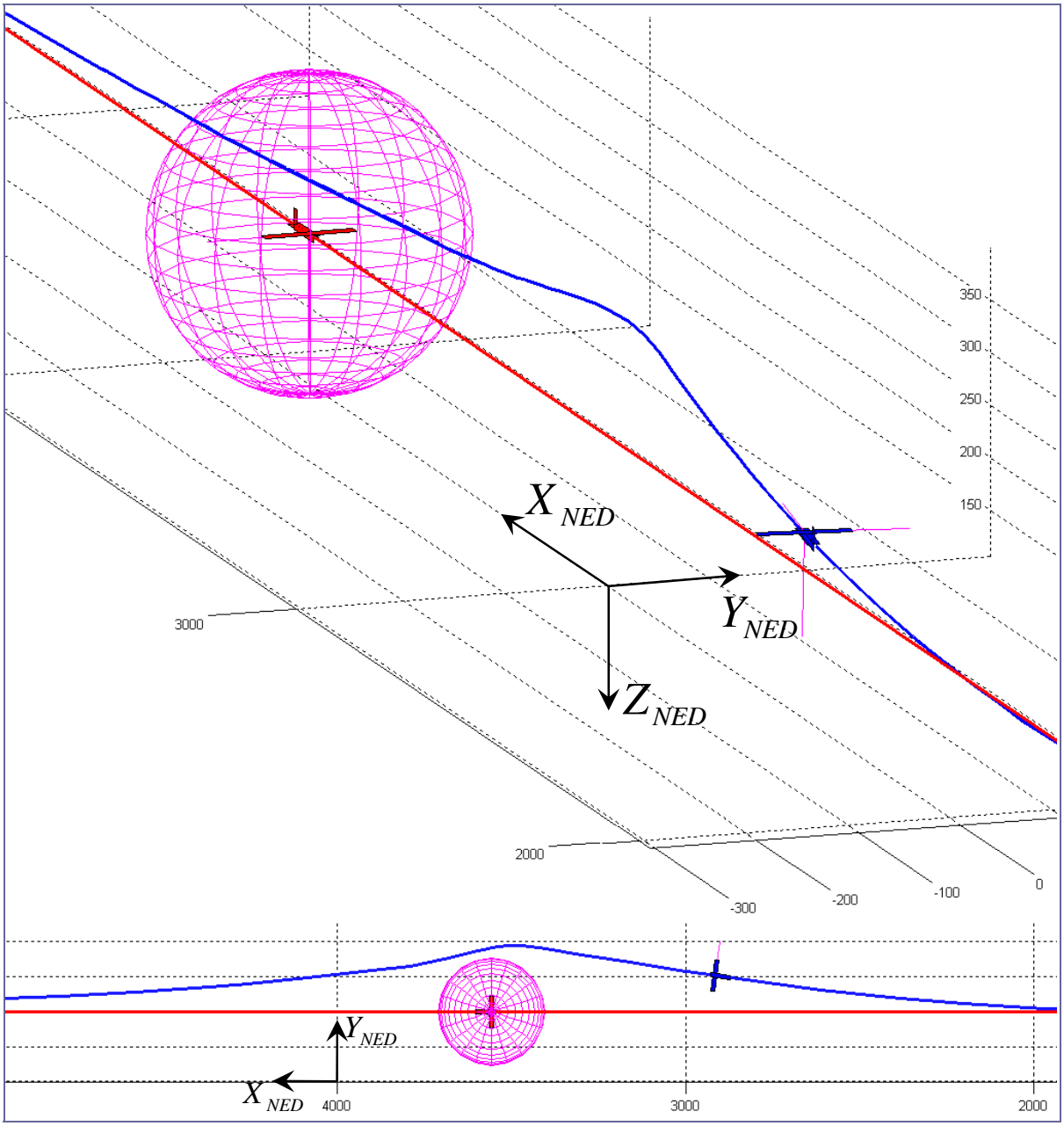

Figure 3 – Graphical description of scenario no. 1

7. The UAV is initially flying at an altitude of 300 m, while the intruder is flying at an altitude of 463 m, initial distance among them is 3.5 nm and they are initially moving on parallel trajectories. After 30.78 s the intruder performs a maneuver in the vertical plane,

changing its direction of 5° in slope towards the UAV. The latter avoids collision by means of a maneuver in the vertical plane near the safety bubble.

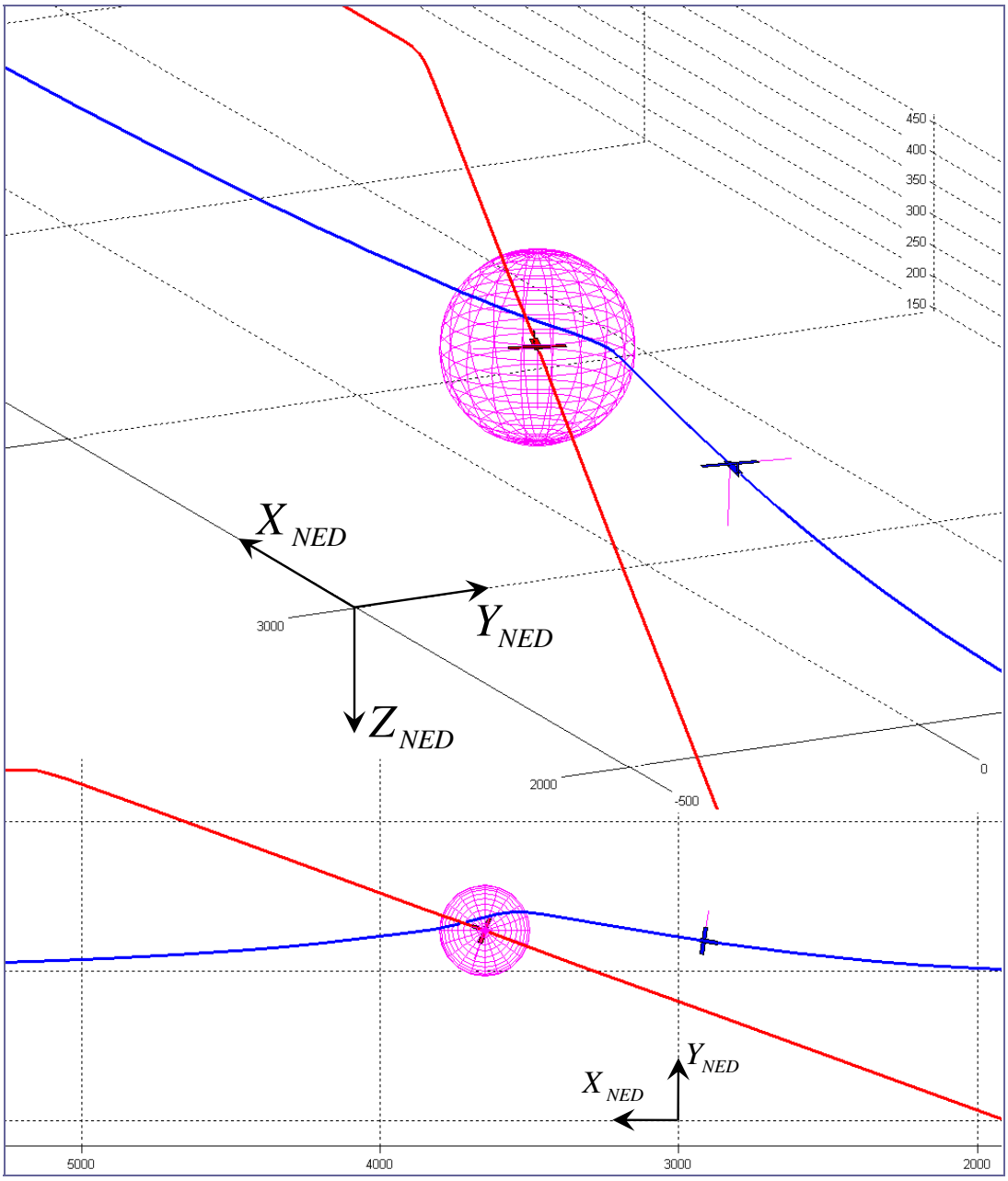

Figure 4 - Graphical description of scenario no. 5

8. The UAV is initially flying at an altitude of 300 m, while the intruder is flying at an altitude of 463 m, initial distance among them is 3.5 nm and they are initially moving on parallel trajectories. After 30.78 s the intruder performs a maneuver in the vertical plane, changing its direction of 5° in slope towards the UAV. The latter avoids collision by means of

an aggressive maneuver in the vertical plane, changing its slope angle of 8° with respect to nominal trajectory.

9. The UAV is initially flying at an altitude of 300 m, while the intruder is flying at an altitude of 137 m, initial distance among them is 3.5 nm and they are initially moving on parallel trajectories. After 30.78 s the intruder performs a maneuver in the vertical plane, changing its direction of 5° in slope towards the UAV. The latter avoids collision by means of a maneuver in the vertical plane near the safety bubble.

10. The UAV is initially flying at an altitude of 300 m, while the intruder is flying at an altitude of 137 m, initial distance among them is 3.5 nm and they are initially moving on parallel trajectories. After 30.78 s the intruder performs a maneuver in the vertical plane, changing its direction of 5° in slope towards the UAV. The latter avoids collision by means of an aggressive maneuver in the vertical plane, changing its slope angle of 8° with respect to nominal trajectory.

4.4. Numerical results

As in the analysis proposed in the previous chapter, also in this case for each scenario 100 Monte Carlo simulations have been performed regarding the generation of radar and EO measures, and then tracking filters have been applied to their output. In summary, a set of 1000 complete detection/tracking simulations have been performed.

In all the considered scenarios, initial range is of about 6 km. As a consequence, for the given radar parameters, probability of detection is high and there are very few detection misses. Thus, after the first intruder detection, the system fast switches from tentative radar tracking to firm radar tracking. On the basis of how EO sensors have been simulated, this tracking phase lasts until the obstacle goes out of the sensors FOV or the distance between the UAV and the intruder becomes smaller than 3 km. If this is the case, the system switches to combined radar/EO tracking with a dramatic improvement in angular accuracy, as it has already been shown in the simplified scenario of chapter 3 and it is confirmed in the following. In all cases tracking ends when the intruder is no more detectable by radar/EO sensors because it is out of their FOV. When this happens, after a few seconds the track is

lost. It is worth pointing out that since tracking filter is based on a linear dynamic model in Cartesian NED coordinates, and the intruder follows a straight trajectory, even when the obstacle is no more detectable the accuracy of obstacle positioning is very stable for many seconds. This behavior can be considered as another advantage in terms of reliability with respect to spherical filters implementation. Of course, from the point of view of autonomous collision avoidance module operation it must be considered that these estimates are affected by a large uncertainty.

In the following, for the sake of brevity only one collision scenario will be examined in detail. Then, statistic data of the simulations performed on all the scenarios will be provided. The considered scenario for detailed analysis is the scenario 1, which is depicted in figure 3.

Exact values for range, azimuth and elevation with respect to the radar sensor (i.e., in Body Reference Frame) are provided in figures 5, 6 and 7. It can be seen that the minimum distance between the airplanes is of about 165 m , and that the intruder goes out of the radar FOV after a time of approximately 70 s.

It is worth recalling that the radar is the main sensor and it is the only sensor used for association and track status definition. This also means that EO detection does not influence track status but only track update in terms of state and covariance estimate. Moreover, EO measurements are not used in the data association process.

Figure 7 shows that elevation dynamics in body is very perturbed following in substance UAV pitch dynamics. In fact, relative dynamics in NED is much less perturbed, as shown by the trend of stabilized elevation, reported in figure 8.

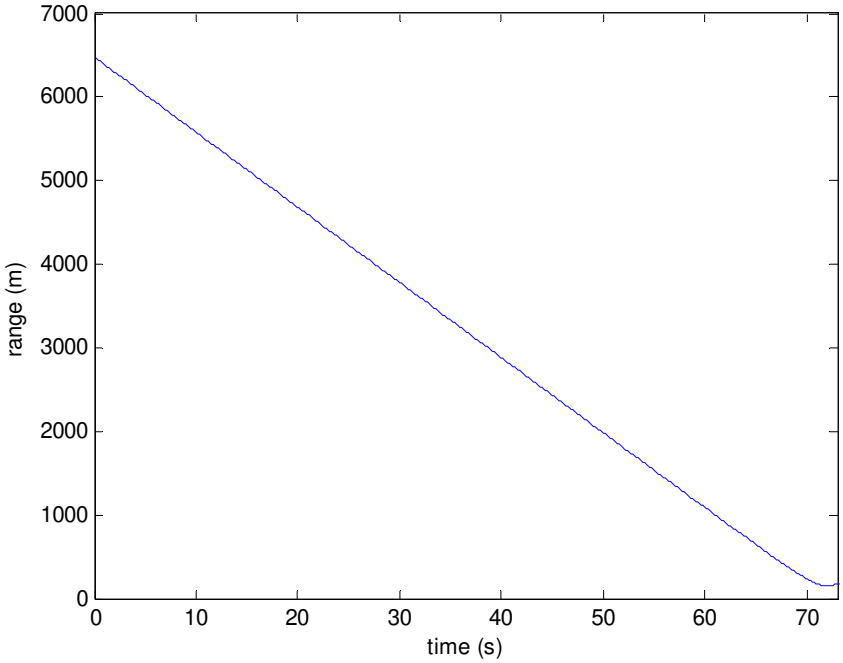

Figure 5 – Range in the considered scenario as function of time

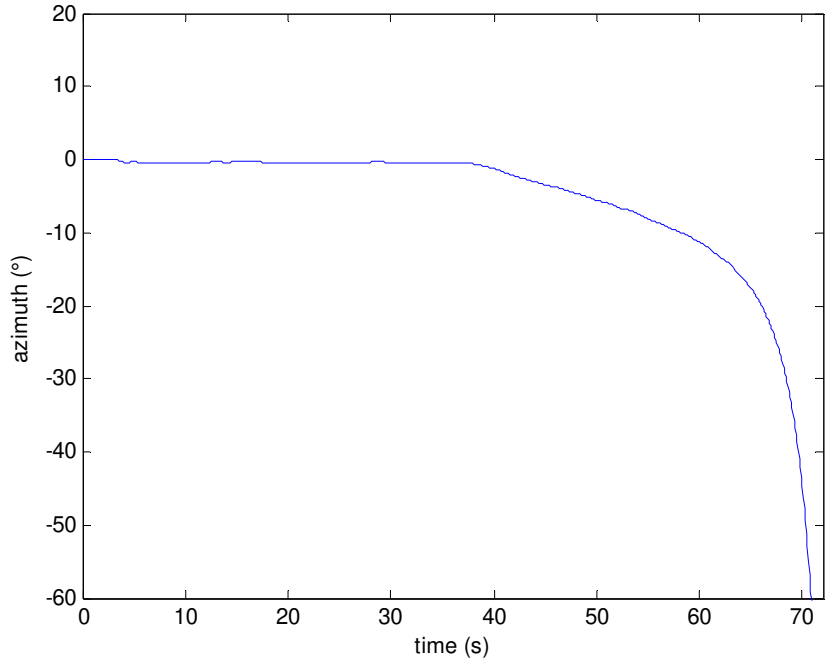

Figure 6 - Azimuth (in BRF) in the considered scenario as a function of time

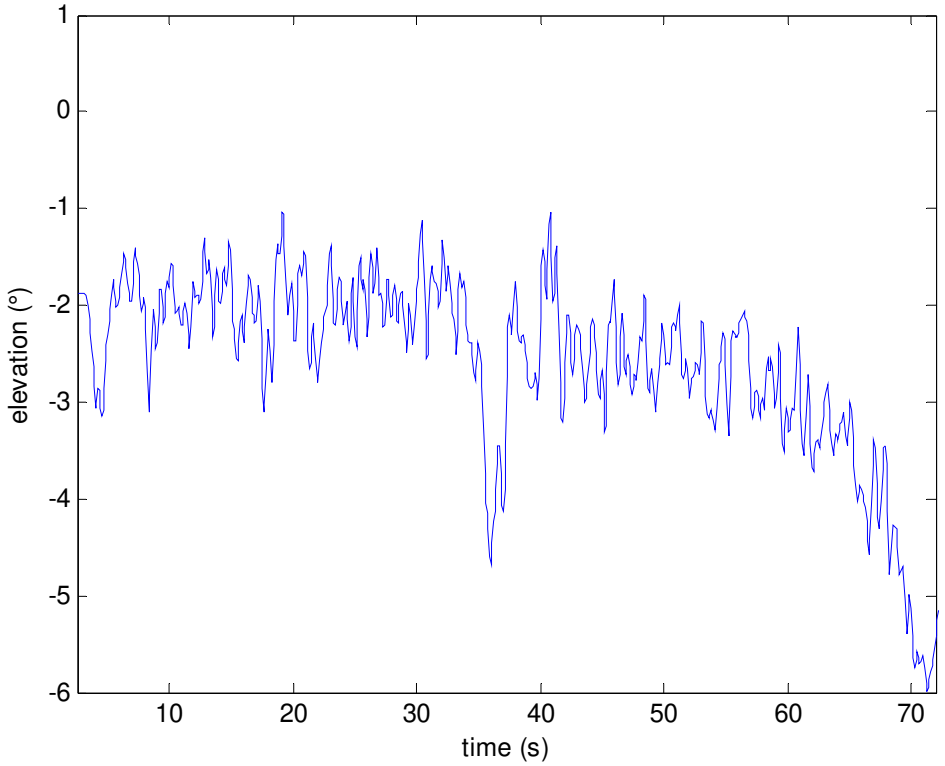

Figure 7 - Elevation (in BRF) in the considered scenario as a function of time

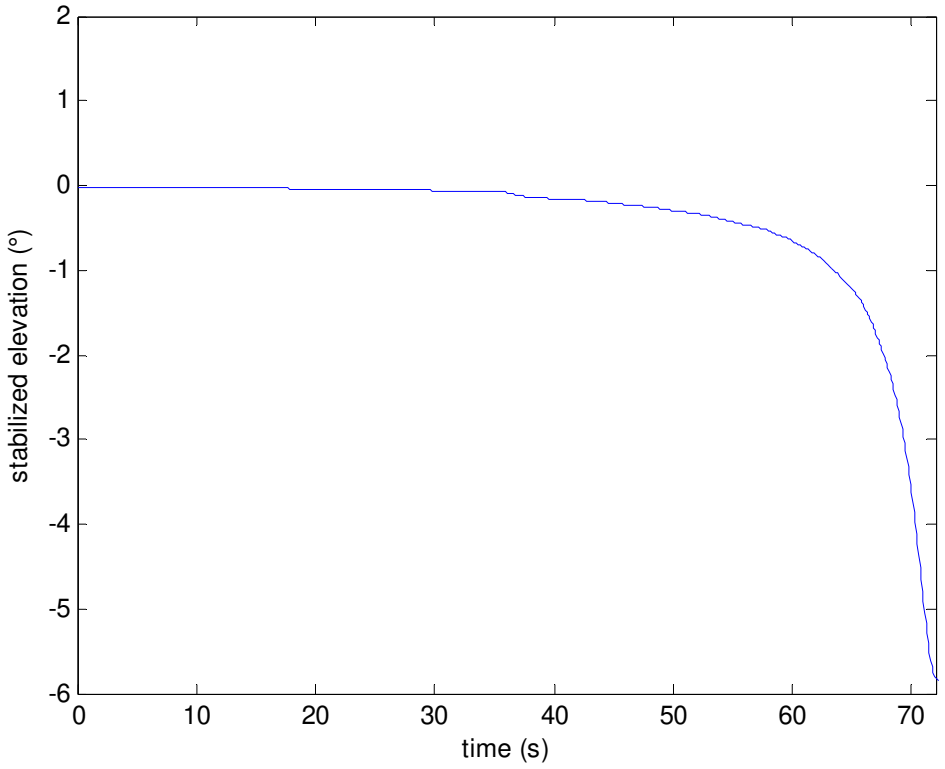

Figure 8 - Stabilized elevation in the considered scenario as a function of time

Focusing now on tracking estimates, let us consider first of all tracking performance in a single simulation. Figures 9, 10 and 11 report errors with respect to exact values in range, azimuth and elevation (in BRF). Tracking estimation error (sampled at 10 Hz) is reported in blue, whereas radar measurement error (sampled at 1 Hz) is reported in red.

It can be seen how the algorithm filters sensor noise though preserving its reliability, in fact no biases are produced. EO sensors detection corresponds in the simulated scenario to a time of about 39s. Since cameras provide only angular measures, their effect on range error is not so much evident, though an improvement can be observed and is due to the coupling between range and angles produced by the Cartesian EKF structure. EO effect can be better appreciated in figure 10 and figure 11, with a large improvement in tracking accuracy. It is worth pointing out that the more "perturbed" trend of azimuth and elevation estimates with respect to the range ones is due to the UAV attitude dynamics more than on the tracker (which operates in NED, as stated before).

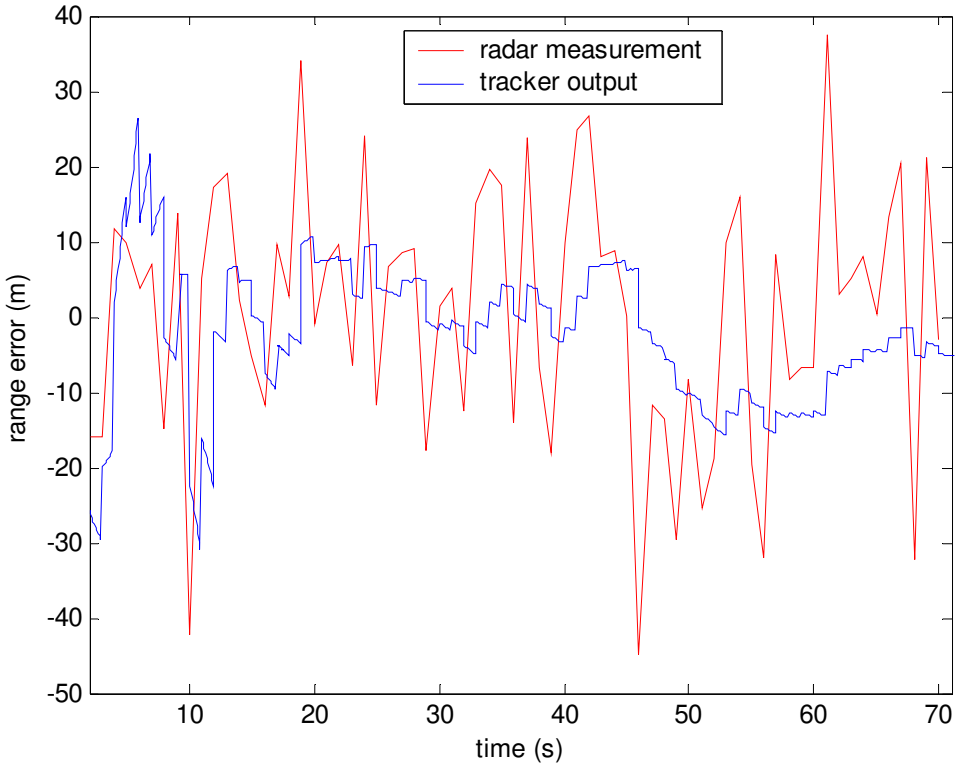

Figure 9 - Radar and track range error as a function of time

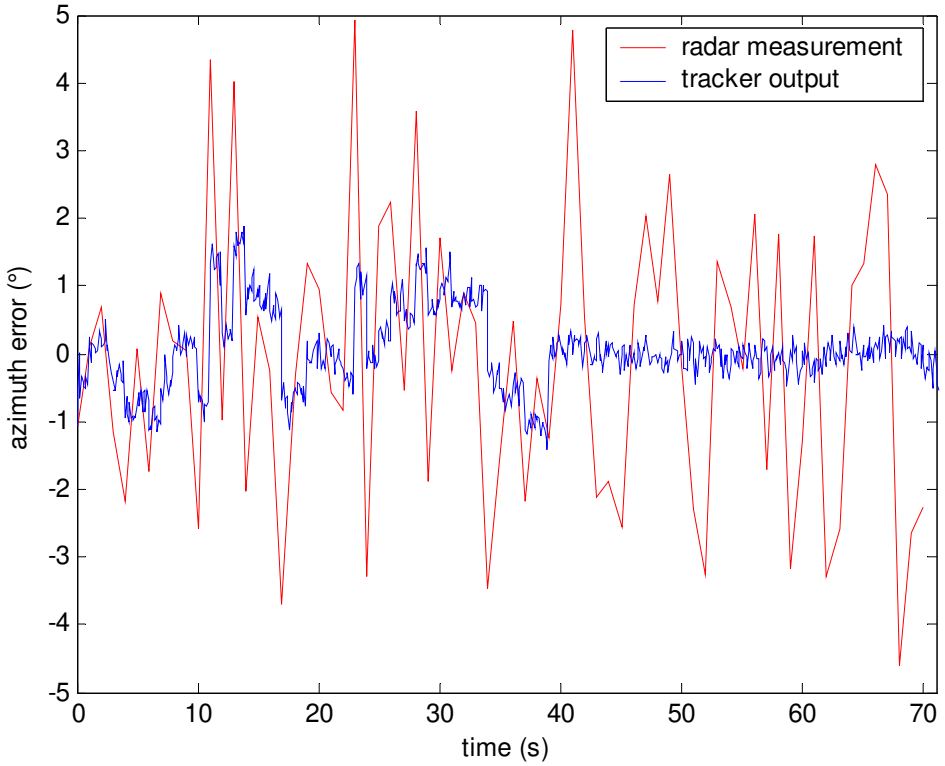

Figure 10 - Radar and track azimuth error as a function of time

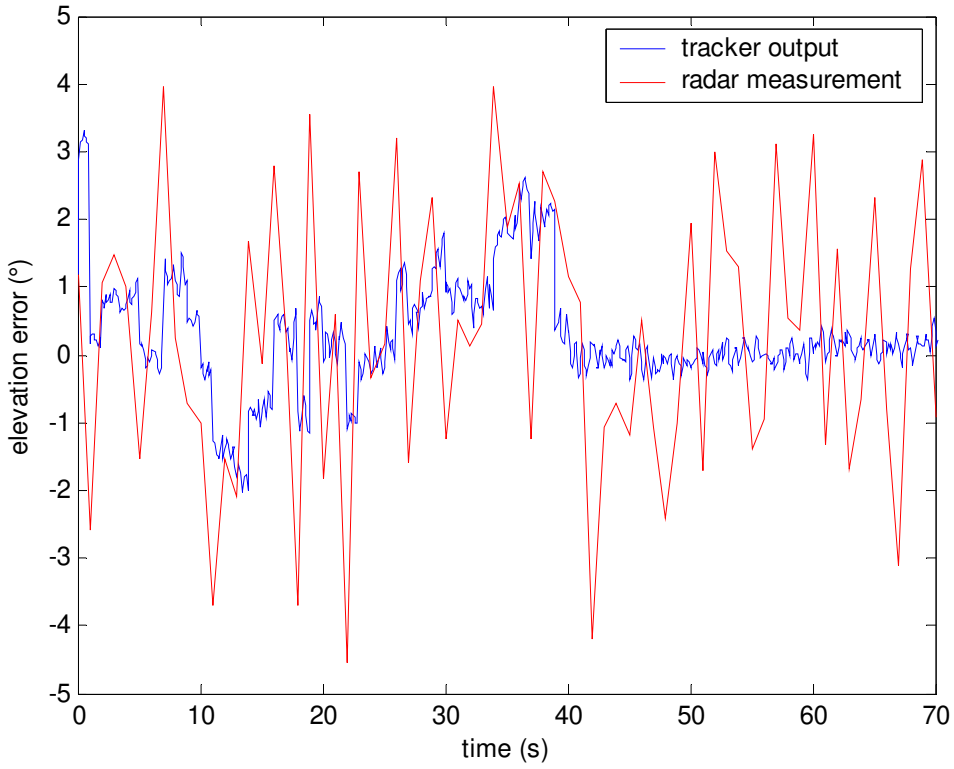

Figure 11 - Radar and track elevation error as a function of time

Figure 12 reports range rate tracker estimation compared with true range rate from the scenario. Of course, since the radar gives no Doppler information, tracker estimates can not be compared with measurements. This estimate is important since it allows to evaluate obstacle velocity: before the maneuver, relative velocity has components only in the range direction. It can be appreciated how the filter estimate converges towards the true value. In figure 13, considered time interval lasts more than 70 s (which corresponds to sensors visibility time in this scenario). This allows to appreciate the filter estimate stability without updates: this is connected to the good approximation of the dynamic model with respect to the simulated scenario.

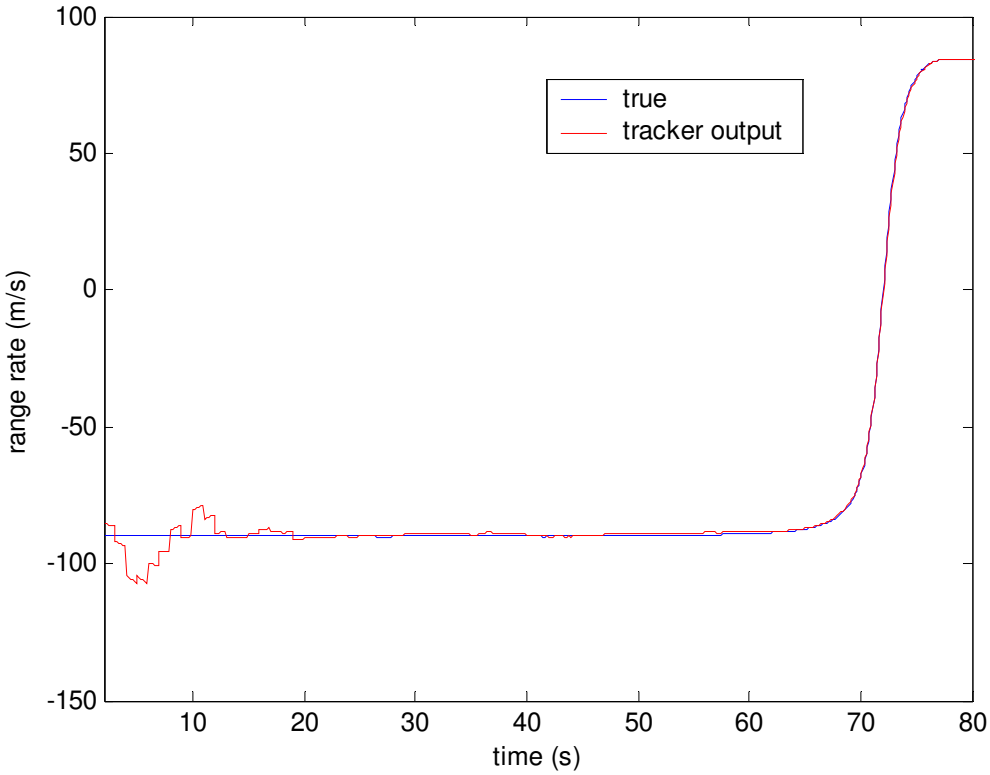

Figure 12 – Tracking range rate estimate compared with true range rate in the considered scenario

Following figures report range and angular errors for the considered scenario as a function of time, considering all the 100 Monte Carlo simulations and the statistical mean and standard deviation calculated on them. The different tracking phases (firm radar and firm radar/EO) are explicitly indicated. During firm radar tracking phase, the tracker follows intruder trajectory with a positioning accuracy which improves in time due in substance to the reducing distance between the airplanes. EO detection is not so much visible in range error diagram (figure 13),

whereas it reduces so much standard deviation of errors on the angles that the three diagrams (mean, mean plus standard deviation, mean minus standard deviation) are hard to distinguish (figures 14 and 15). If we consider NED estimates (figures 16 and 17), then a less perturbed trend is obtained (no effects of attitude dynamics) and it is easier to estimate reduction of standard deviation in angular errors.

Furthermore, it is interesting to observe that biases are produce both in stabilized azimuth and in stabilized elevation. This is closely connected to the AHRS errors in estimating attitude angles. In fact, stabilized azimuth error mean equals in substance the bias in heading estimate, whereas a smaller error mean is observed on stabilized elevation, which is of the same order of magnitude of biases in estimates of roll and pitch angles. These relations are due to the fact that an almost non maneuvering flight segment is considered. In a few words, biases in navigation errors are almost completely filtered in body angular estimates, and thus corrupt only NED estimates.

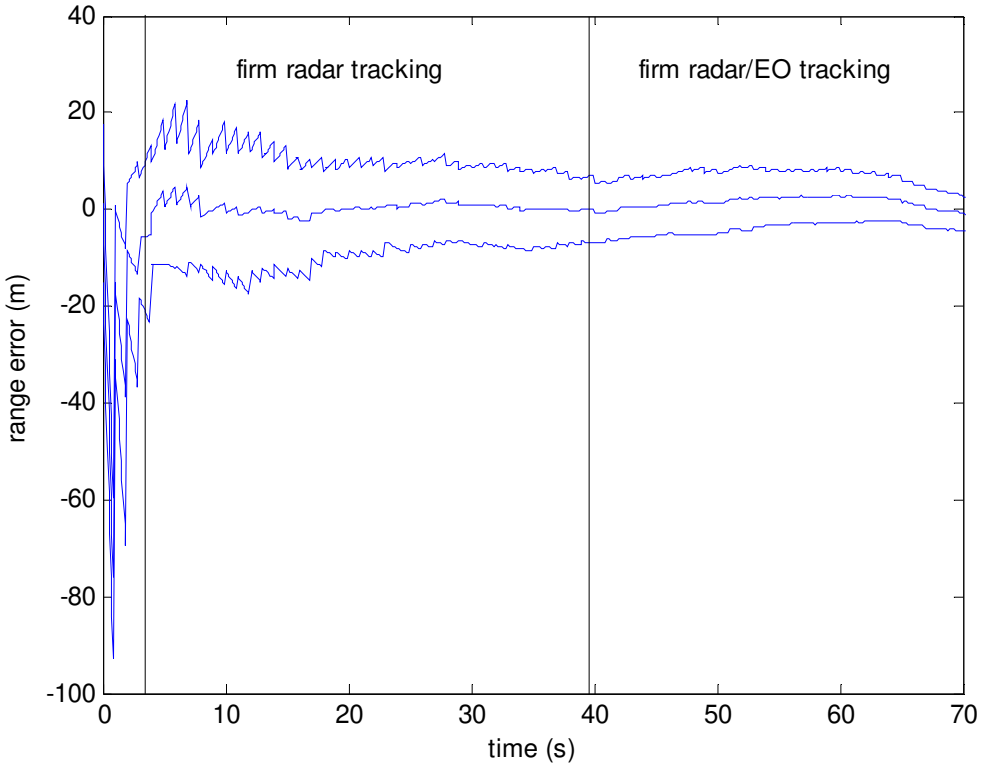

Figure 13 - Error in range estimate, in terms of mean and standard deviation

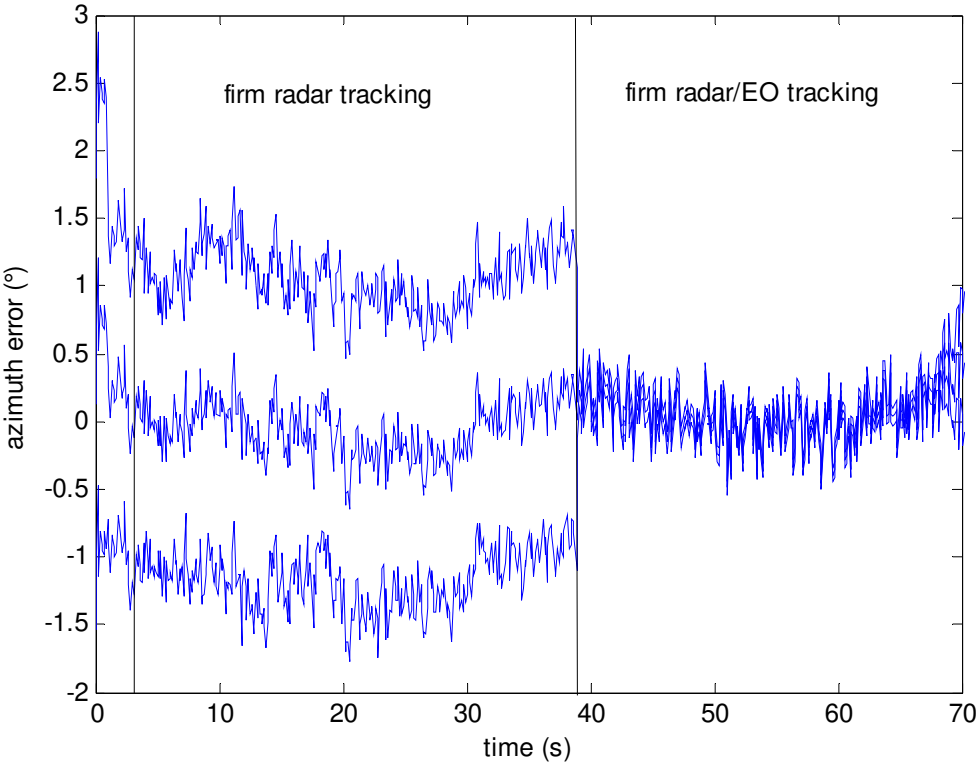

Figure 14 - Error in azimuth estimate, in terms of mean and standard deviation

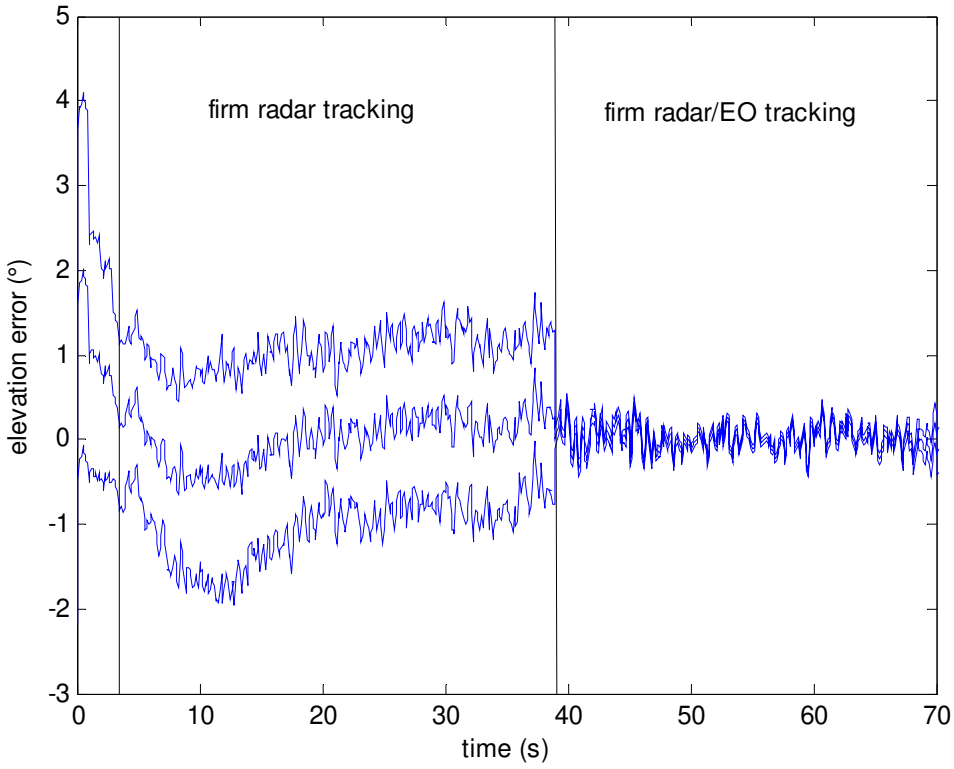

Figure 15 - Error in elevation estimate, in terms of mean and standard deviation

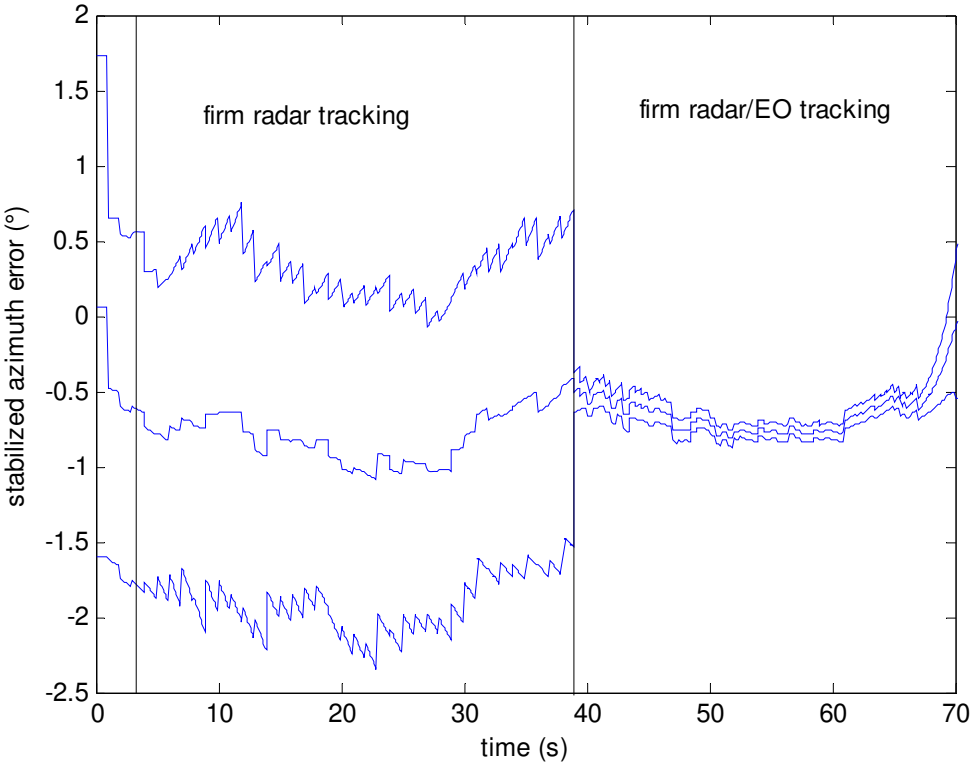

Figure 16 - Error in stabilized azimuth estimate, in terms of mean and standard deviation

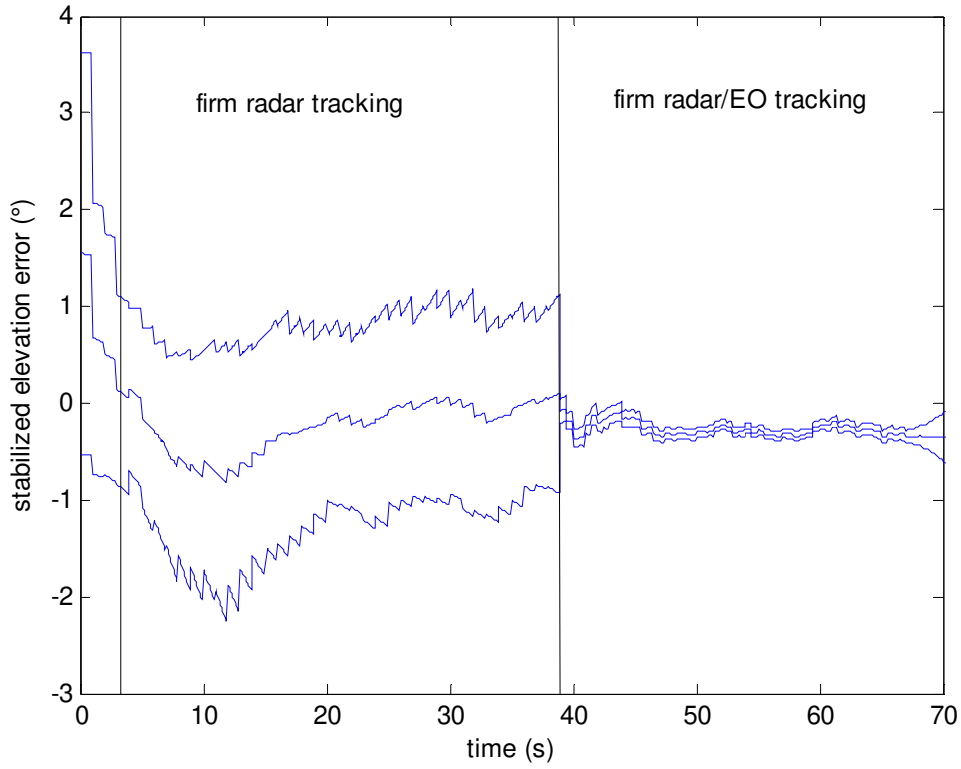

Figure 17 - Error in stabilized elevation estimate, in terms of mean and standard deviation

Rate estimates allow for similar considerations. Range rate estimates are very stable and increasingly more and more accurate for decreasing distance (figure 18). Body angular rate estimates (figures 19 and 20) are influenced by attitude dynamics, especially regarding elevation.

Stabilized angular rate estimates (figures 21 and 22) allow to appreciate the reduction in standard deviation due to radar/EO fusion

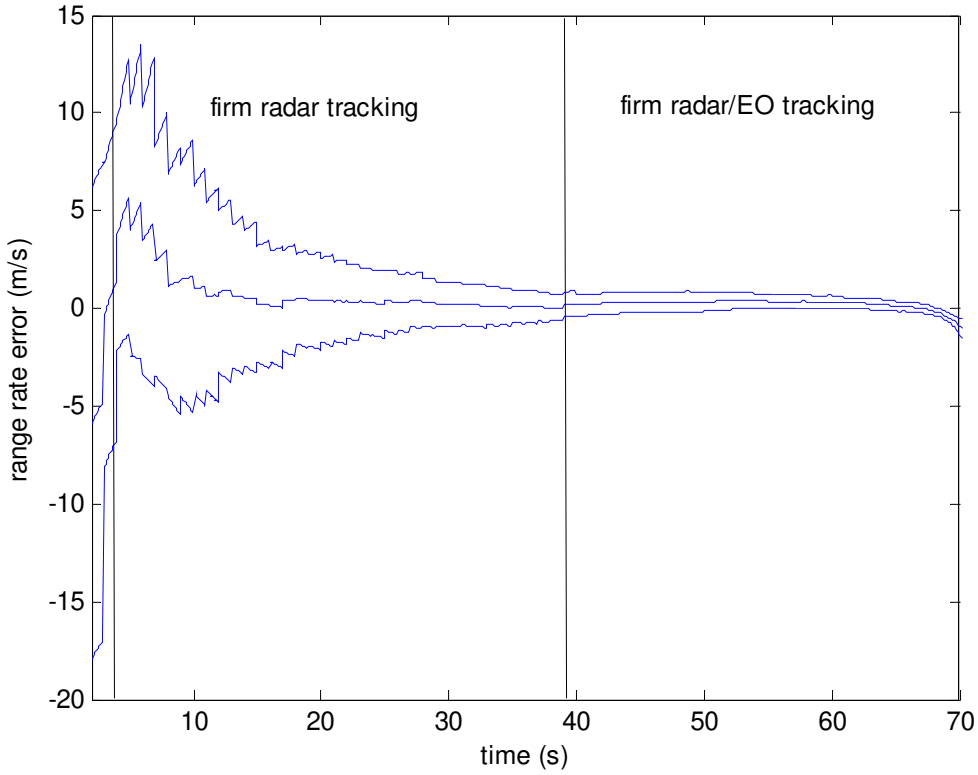

Figure 18 - Error in range rate estimate, in terms of mean and standard deviation

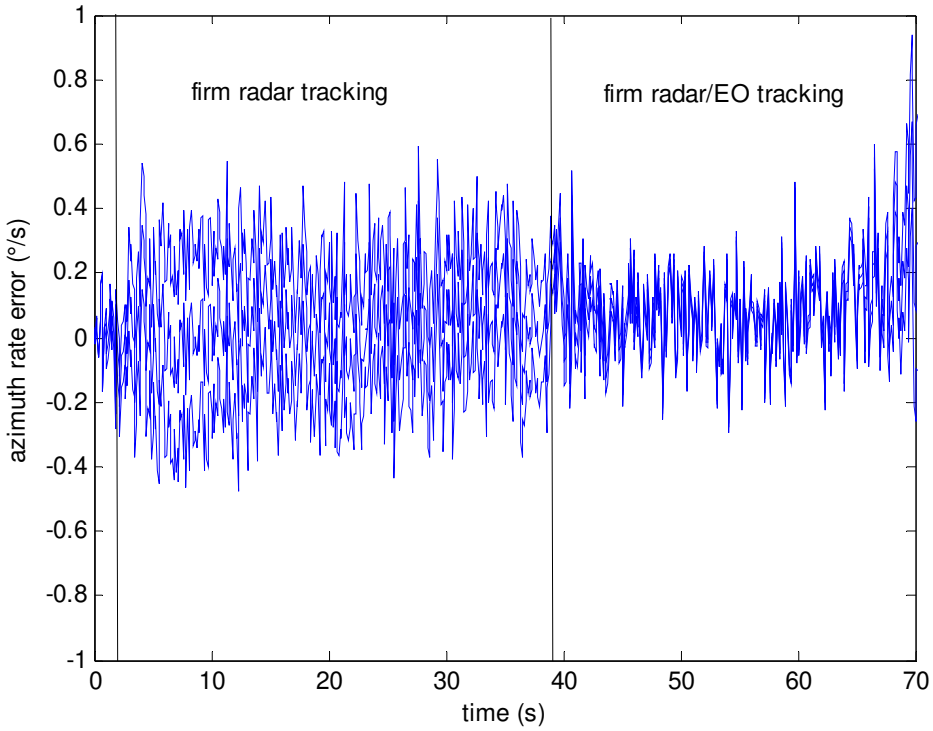

Figure 19 - Error in azimuth rate estimate, in terms of mean and standard deviation

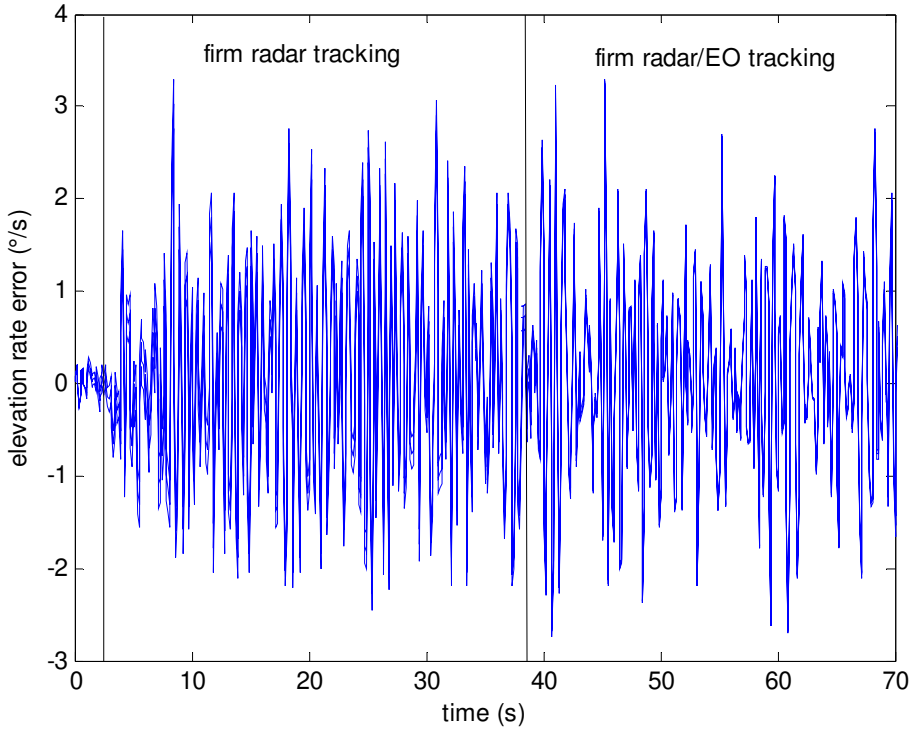

Figure 20 - Error in elevation rate estimate, in terms of mean and standard deviation

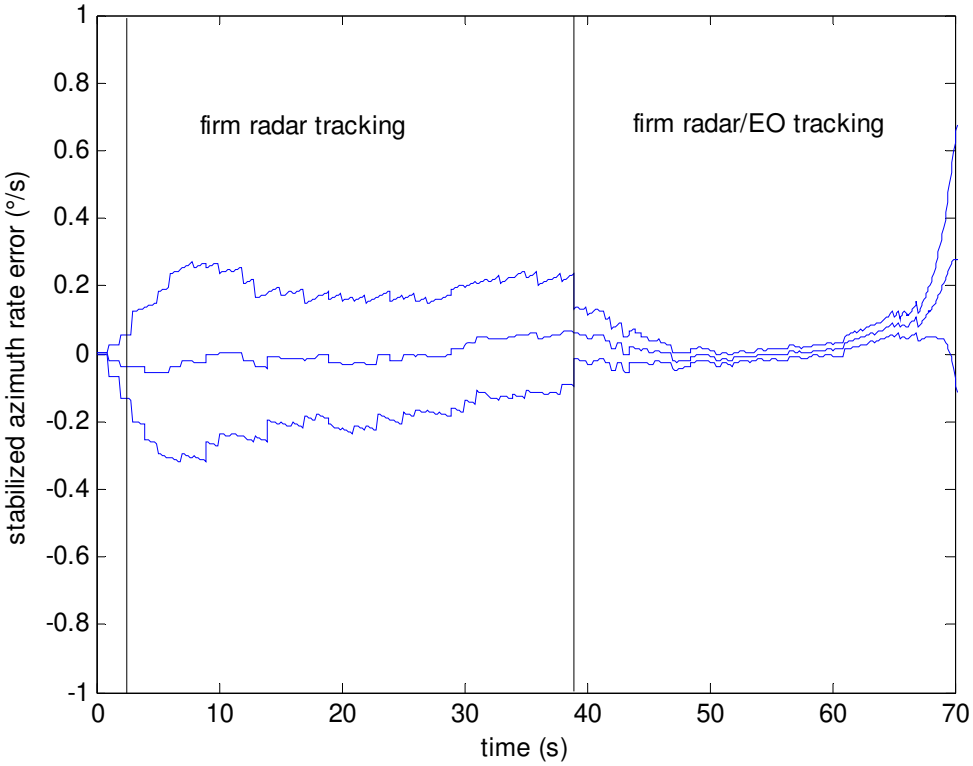

Figure 21 - Error in stabilized azimuth rate estimate, in terms of mean and standard deviation

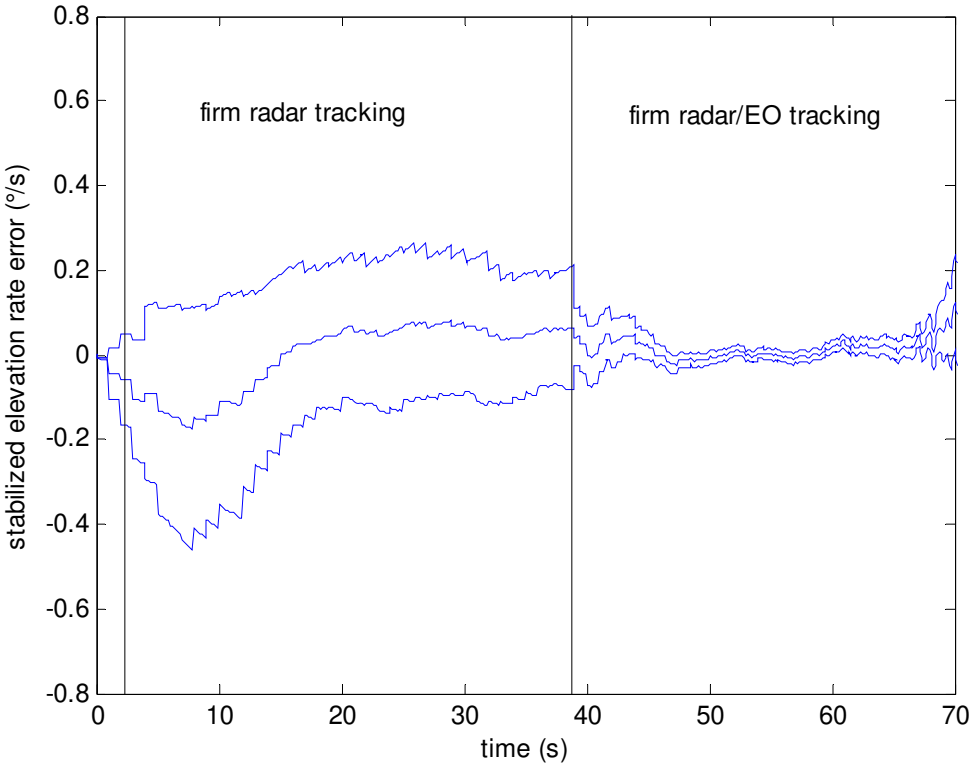

Figure 22 - Error in stabilized elevation rate estimate, in terms of mean and standard deviation

It is also interesting to point out that, regarding the accuracy of gating operation, as in the simplified scenario presented in previous chapter, no track loss phenomena have been observed in all the performed simulations.

Considering numerical results for all the 10 scenarios, it is worth noting that performed simulations for the different avoidance maneuvers held very similar results in tracking performance, showing that tracking algorithm is robust with respect to UAV flight dynamics and collision avoidance maneuver choice. Furthermore, this is also due to the fact that the obstacle is tracked mostly before the avoidance maneuver and in the first few seconds after maneuver initiation. Then, the execution of avoidance maneuver, both because of trajectory and attitude modification, brings the obstacle outside of the FOV of the sensors, which are of course body fixed.

However, visibility time for both radar and EO sensors does depend on the avoidance strategy, and is longer for minimum deviation maneuvers: this constitutes a key point for collision avoidance maneuver planning.

Synthetic results are reported in table 2, table 3, table 4, and table 5. In particular, Table 2 reports time averaged values of statistical mean errors for all the scenarios and the different estimates related to obstacle position, whereas Table 3 reports time averaged values of statistical standard deviations for these estimates. Table 4 and table 5 report time averaged values of errors' statistical mean and standard deviation, considering estimates related to obstacle velocity.

Mean errors	Range (m)	Azimuth BODY (°)	Azimuth NED (°)	Elevation BODY (°)	Elevation NED (°)
Firm radar tracking	-0.24	-0.02	-0.75	0.03	-0.21
Firm radar/EO tracking	1.54	0.07	-0.64	0.02	-0.23

Table 2 - Mean (position) errors for the available tracking modes

Standard deviation of errors	Range (m)	Azimuth BODY (°)	Azimuth NED (°)	Elevation BODY (°)	Elevation NED (°)
Firm radar tracking	10.30	1.11	1.10	1.09	1.09
Firm radar/EO tracking	5.90	0.085	0.085	0.071	0.071

Table 3 - Standard deviation of (position) errors for the available tracking modes

Mean errors	Range rate (m/s)	Azimuth rate BODY (°/s)	Azimuth rate NED (°/s)	Elevation rate BODY (°/s)	Elevation rate NED (°/s)
Firm radar tracking	0.76	0.022	-0.012	0.036	-0.0050
Firm radar/EO tracking	0.35	0.077	0.026	0.066	0.013

Table 4 - Mean (velocity) errors for the available tracking modes

Standard deviation of errors	Range rate (m/s)	Azimuth rate BODY (°/s)	Azimuth rate NED (°/s)	Elevation rate BODY (°/s)	Elevation rate NED (°/s)
Firm radar tracking	3.03	0.21	0.20	0.19	0.19
Firm radar/EO tracking	0.47	0.044	0.044	0.031	0.031

Table 5 - Standard deviation of (velocity) errors for the available tracking modes

Synthetic data for all the simulations confirm some results which have already been pointed out for scenario 1. Considering mean errors (table 2), body angular estimates are in substance zero-mean whereas biases in attitude angles estimation are reflected in non-zero mean errors for NED angular estimates. It can be observed that a small bias in range error is generated in

firm radar/EO tracking. This is due to the fact that because of track update with EO cameras measurements, track covariance reduces and so also the weight of radar range measurements does. Nevertheless, as it appears analyzing also table 3, RMS error reduces in firm radar/EO phase also in range estimate.
Finally, it is worth noting that standard deviations in angular/angular rate errors do not depend on the considered reference frame, because the origin of the variability is in both cases in the radar/EO detection process. This fact results also in the previously presented diagrams.

In summary, it can be concluded that on the basis of off-line simulations, developed hardware/software for obstacle detection and tracking is compliant with the requirements indicated in the preliminary studies of TECVOL project.

Chapter 5

System development and real time issues

5.1. Introduction

The previous chapters dealt with anti-collision system architecture and algorithms. Since the goal of performed activities is sense and avoid flight demonstration, real time implementation of developed logics has been considered as a very important subject. It is detailed in this chapter.

For the sake of clarity, however, first it is necessary to clarify how the ground and the flight segments are connected, in particular for what regards the sense and avoid function. Ground control station is made up by a number of workstations which are connected on an Ethernet bus and communicate by means of the User Datagram Protocol [50]. In particular, it is foreseen that a workstation is dedicated to obstacle detection and tracking experiments monitoring. The obstacle detection ground station sends commands and receives data in real time from a computer which acts as the ground communication controller. The latter is devoted to communication with FLARE.

In particular, it communicates with an on-board communication controller by means of a full duplex Radio Frequency (RF) data link. This latter computer communicates with the flight control computer and the real time computer by means of the CAN bus. This architecture is shown in figure 1.

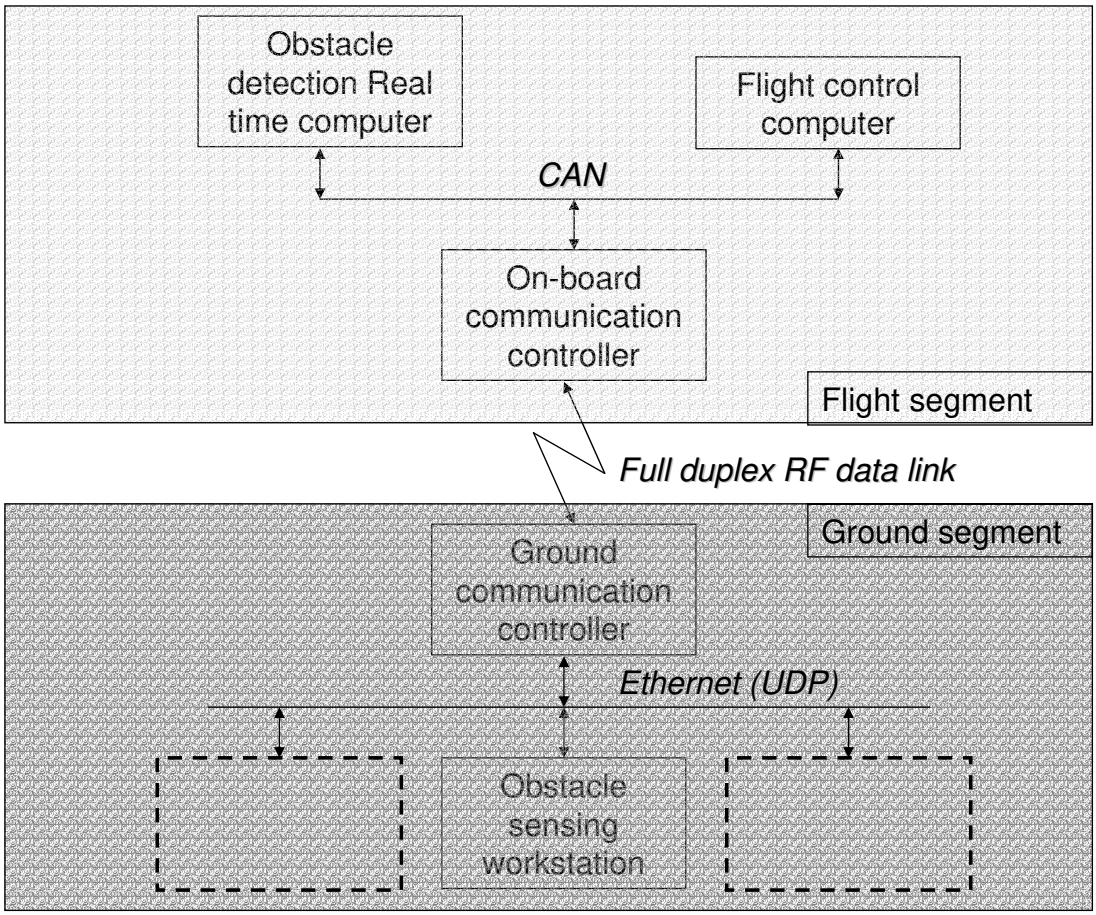

Figure 1 – Physical architecture relevant to obstacle detection function

The software development process was performed with different approaches and languages for the real time computer and the image processing computer [81]. The two units are described sequentially in the following sections, together with the ground station software.

Then, communication with Amphitech radar is described. It is based on a customized protocol, and allows to remotely control the radar from the ground station and to choose among different operative modes. Hardware-in-the-loop tests have been performed to test reliability and latency of communications. The relevant results are pointed out.

Then, in the last section an analysis is presented dealing with real time tracking issues. In fact, on the one hand it must be considered that radar measurements are received with a time delay, so that tracks have likely been predicted for a later instant. On the other hand, a pre-

processing phase is necessary for eliminating ground echoes. In particular, this allows to identify an uncertainty region where tentative tracking is required for echo rejection.

5.2. Real time computer and software

The real time unit is based on a single board PCI computer , the Emcore i-6419^{TM} (figure 2). The board is equipped with a Celeron M 1.5 GHz processor. It has been assembled in a compact configuration including an Ethernet card for debugging function and a 2-port CAN interface card (figure 3).

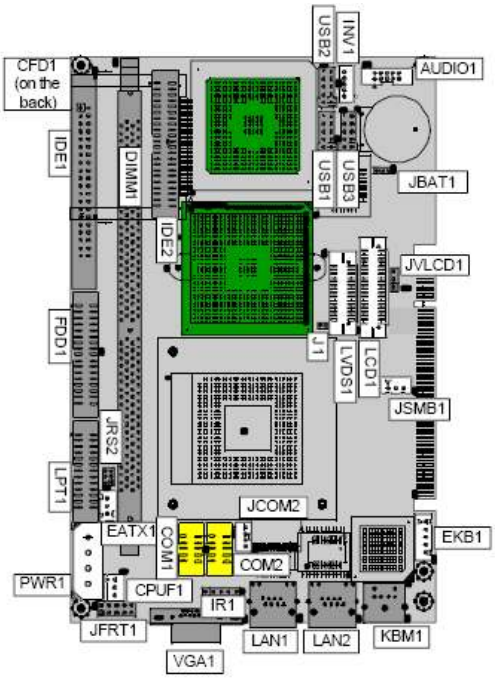

Figure 2 – Board layout for real time computer

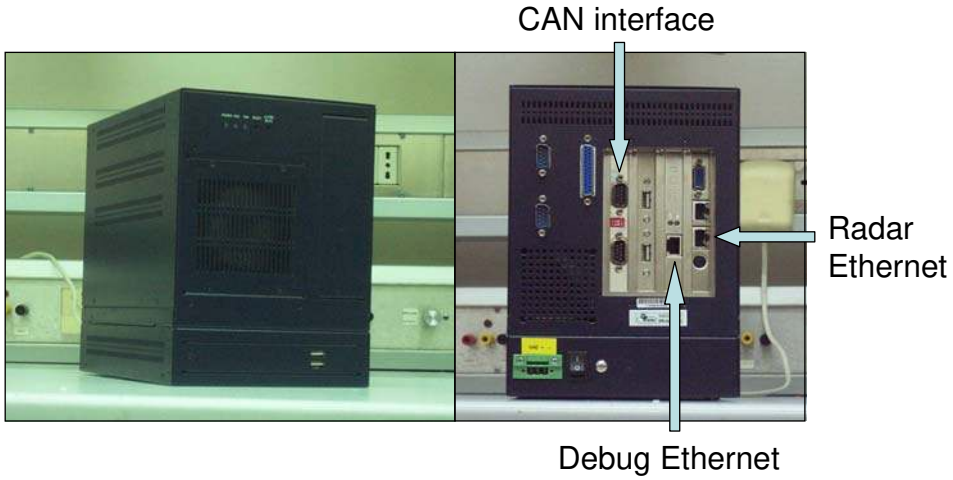

Figure 3 – Real time computer: front and rear view

The real time computer operating system (OS) is a customized OS based on Microsoft™ Windows CE 5.0™, which is a 32 bit hard real time OS with a very small footprint [82].

Real time system is programmed in target-host configuration. In this configuration, a desktop computer is used as development station, and an Ethernet connection (on the debug Ethernet card) allows to download the operating system and the applications to the target unit, and to control it. The operating system has been customized by means of the Platform Builder™ software (figure 4), which allows to choose which modules are to be added to the target operating system. Then, it performs operating system compiling and target control. Customized OS footprint is of the order of 6 MB.

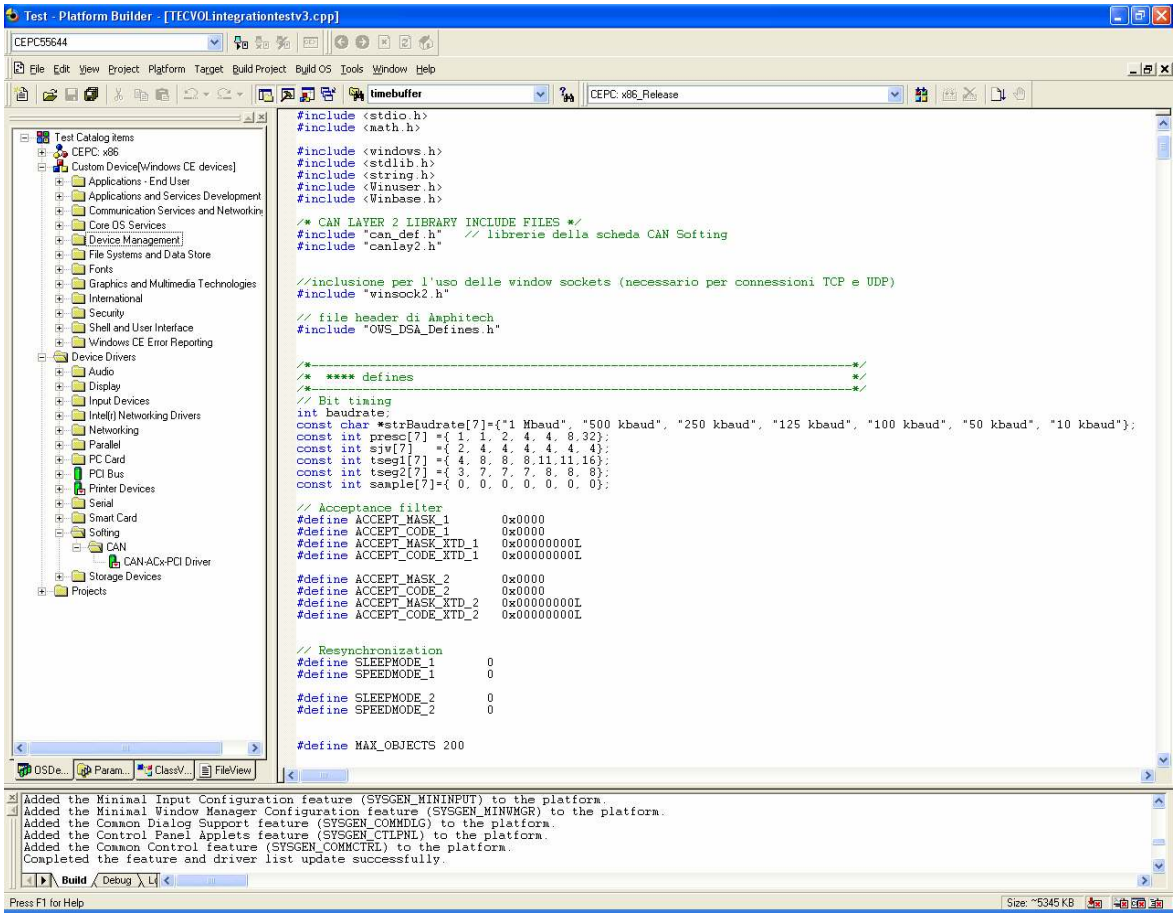

Figure 4 – Example from development workstation software

The on board software for the real time computer was developed to be remotely controlled from the ground station while keeping high autonomy in operations and accurate real time performances. Basically, it is made up by different modules whose activation/deactivation is performed on the basis of timing and received commands/data. The software has to interface

with the image processing system, the radar and the CAN bus, from which it receives both navigation sensors data (generated by the GNC computer) and commands from ground (sent by the on-board communication controller). On the other hand, it provides tracking estimates, diagnostic messages, and raw targets data from radar. The entire architecture is depicted in figure 5.

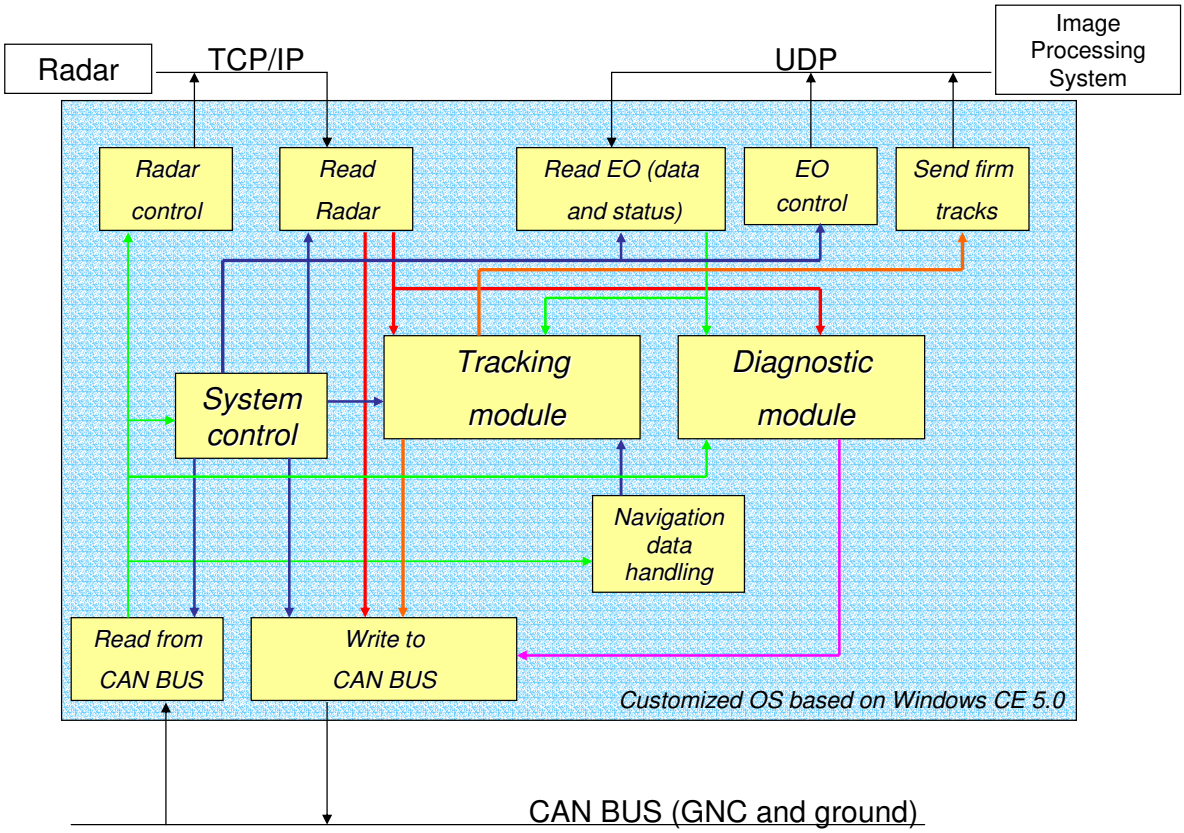

Figure 5 - Real Time software architecture

A comprehensive description of CAN bus can be found in [47]. CAN handles real time requirements by assigning different priorities to the messages. In particular, each message is characterized by an integer identifier (ID) which determines priority (higher identifier means lower priority). In case of collision between two packets, the highest priority message is transmitted first. Each computer on the bus can filter the messages on the basis of their ID, so that only data of interest are received and treated at application level. Among the other features, CAN is very flexible: if a node is added or subtracted from the network, no changes are required to the other nodes, except if other messages are to be received. In TECVOL, priorities were assigned to the messages on the basis of impact of latencies on system performance and safety.

As it can be seen in figure 5, the system control module commands the activation/deactivation of the tracking system, based on commands from ground, and of all the other modules, based on the CPU clock. This ensures that the highest priority task is always performed with minimum uncompensated latencies.

Communication with radar is performed by means of an Ethernet link and the TCP/IP protocol [83], and will be detailed further in a following section. The radar control module is activated only when configuration changes are commanded from the ground station. It allows to activate/deactivate RF emission by the radar and to change field of regard (and consequently the scan time), max unambiguous range, and center of scan volume with respect to aircraft body axes. The read radar module is activated depending on system clock and provides raw data for the tracking algorithm and for downlink.

Communication between real time and image processing computer is based on the UDP protocol. Since this is a point-to-point communication with very low data rate, no loss of data are caused by using this connectionless protocol. On the other hand, UDP allows to minimize latencies and simplify communication. From application programming point of view, communication is performed by means of sockets [84], which are classical low-level communication interfaces. Messages from real time computer to image processing computer regard cameras control parameters and the estimates of firm tracks, to be used for windowing by the image processing software. All these information are contained with a fixed order in a single UDP packet, so that message identifiers are not necessary.

A navigation data handling module has been introduced to store navigation data (which are received from flight control computer at a frequency of 10 Hz) as a function of elapsed time, because radar scan rate is of the order of 1 Hz, and tracking algorithm requires track initialization and coordinate transformation of raw data (from body reference frame to NED reference frame) to be performed with the best possible accuracy. Of course, it makes sense to store navigation data which only refer to radar scan time.

Diagnostic module is a low priority module used to evaluate in real time the correct functioning of the system. In particular, it evaluates latencies of the different communication channels, verifying that no anomalous values are generated. This allows to evaluate the correct functioning of the radar, of the image processing system with all the cameras, and the correct reception of navigation data. For example, in case of CAN data the interface card autonomously stores time of reception of latest packets, so that communication latency can be

easily kept under control. All the software for the real time unit has been programmed in C language.

5.3. Image processing computer and software

From the hardware point of view, the image processing computer is very similar to the other unit, except that it is equipped with a Pentium M™ processor, and the CAN interface card is replaced by a Firewire 3-port card. Regarding the software, it was developed in a conventional OS, namely Windows XP Embedded™ sp2, in order to interface to Firewire cameras. XP Embedded is a componentized version of Windows XP™ which allows to reduce footprint by maintaining only those modules which are required by the selected hardware and application [85]. Of course, resulting OS is much more time-efficient than Windows XP. In fact, resulting footprint is typically of the order of 250 MB. The customized version of the OS can be developed and built in the Target Designer™ environment (figure 6). The process is highly automated thanks to a software utility (Target Analyzer, [86]) which allows to identify all the OS modules required for the considered hardware. Then, software components needed for the application can be added, and Target Designer builds the customized OS. It is worth noting that, thanks to the choice of the OS, image processing system development has been carried out in a standard Windows XP environment.

Image Processing software architecture is shown in figure 7. Each camera is managed by means of an ActiveX Control. The system is commanded by the real time computer by means of the UDP link. When firm tracks are present, intruder state predictions are sent to the image processing software. After initialization phase, which foresees connection to the cameras, the image processing system works trying to answer the requests from the real time computer. The UDP read module commands the cameras to provide their latest images to the Object Detection module, which tries to detect obstacles by applying threshold criteria to a windows centered on predicted obstacle position. From programming point of view, the Object detection module operates on the memory buffers containing the latest images. This detection logic allows to reduce processing time and save the real time requirements of the system. Of course, the same logic could not be applied with success and in a reasonably short time to the whole images produced by the cameras. Thus, as already pointed out before, radar detection is essential also in view of EO detection. Diagnostic module periodically checks cameras connection status. The software has been programmed in the Visual Basic.Net environment.

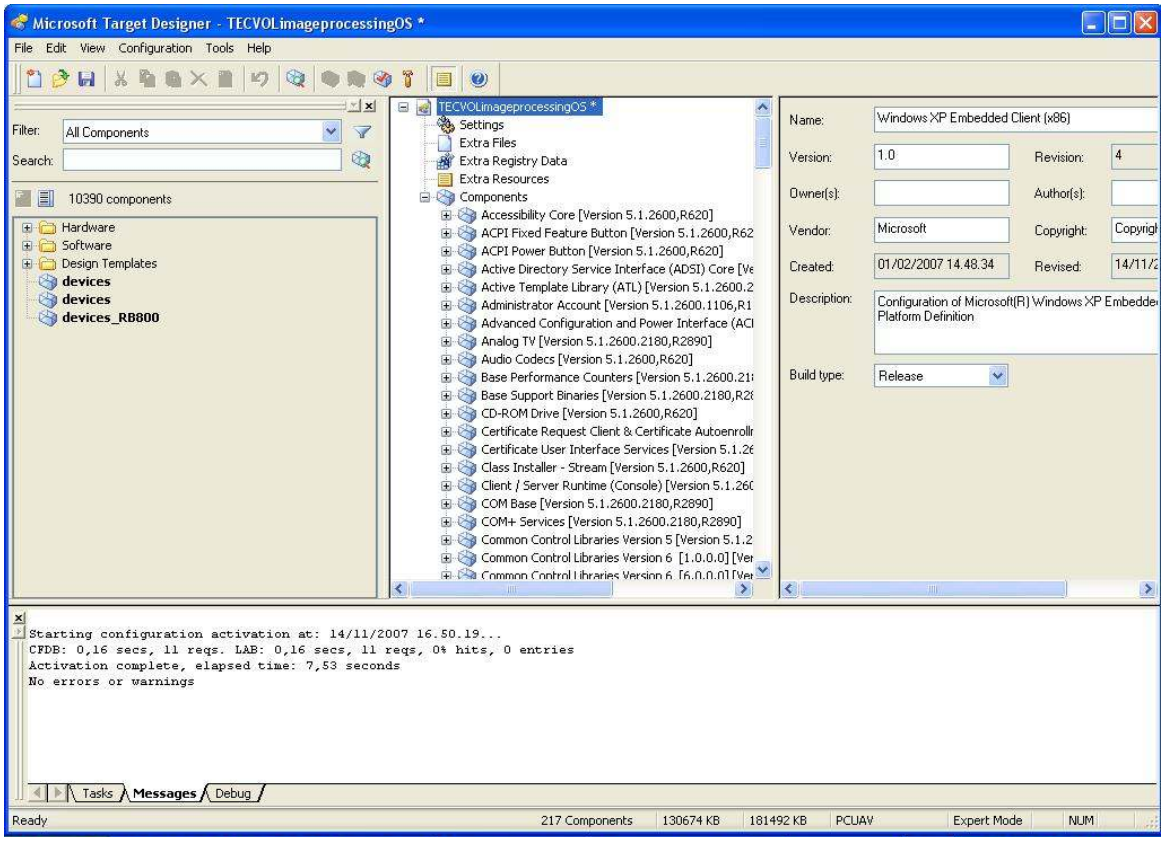

Figure 6 – Target designer graphical interface

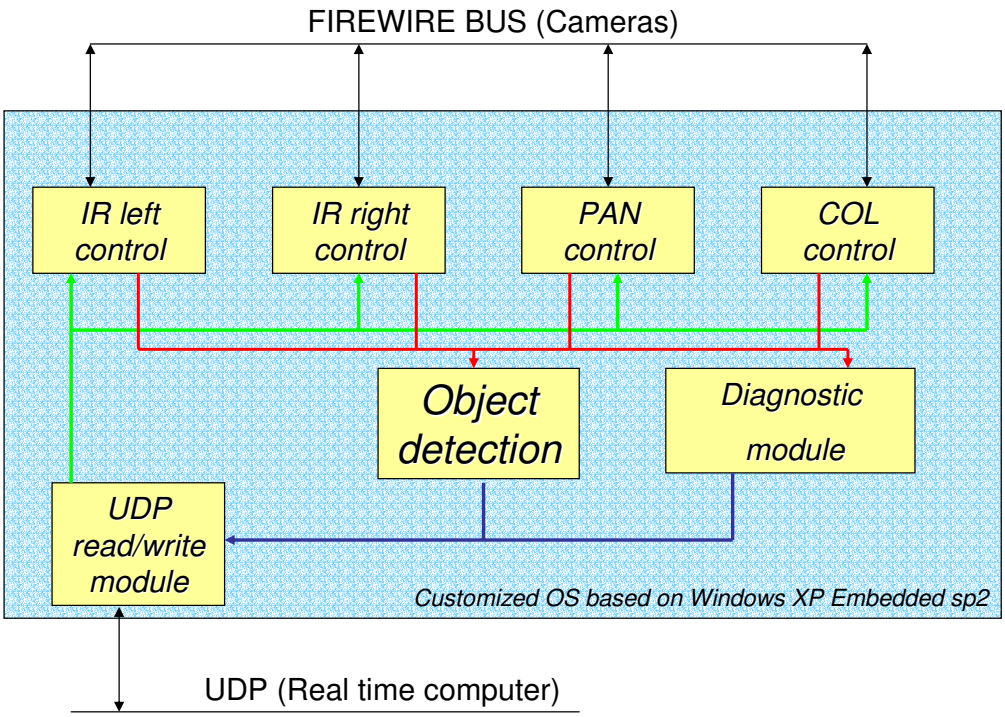

Figure 7 - Image processing software architecture

5.4. Ground station software

Ground station software is used to monitor test status by means of information reported in graphical displays and numerical output, and to send commands. Data are transferred by means of the radio downlink and the ground Communication Controller CPU. A preliminary version of graphical interface is depicted in figure 8.

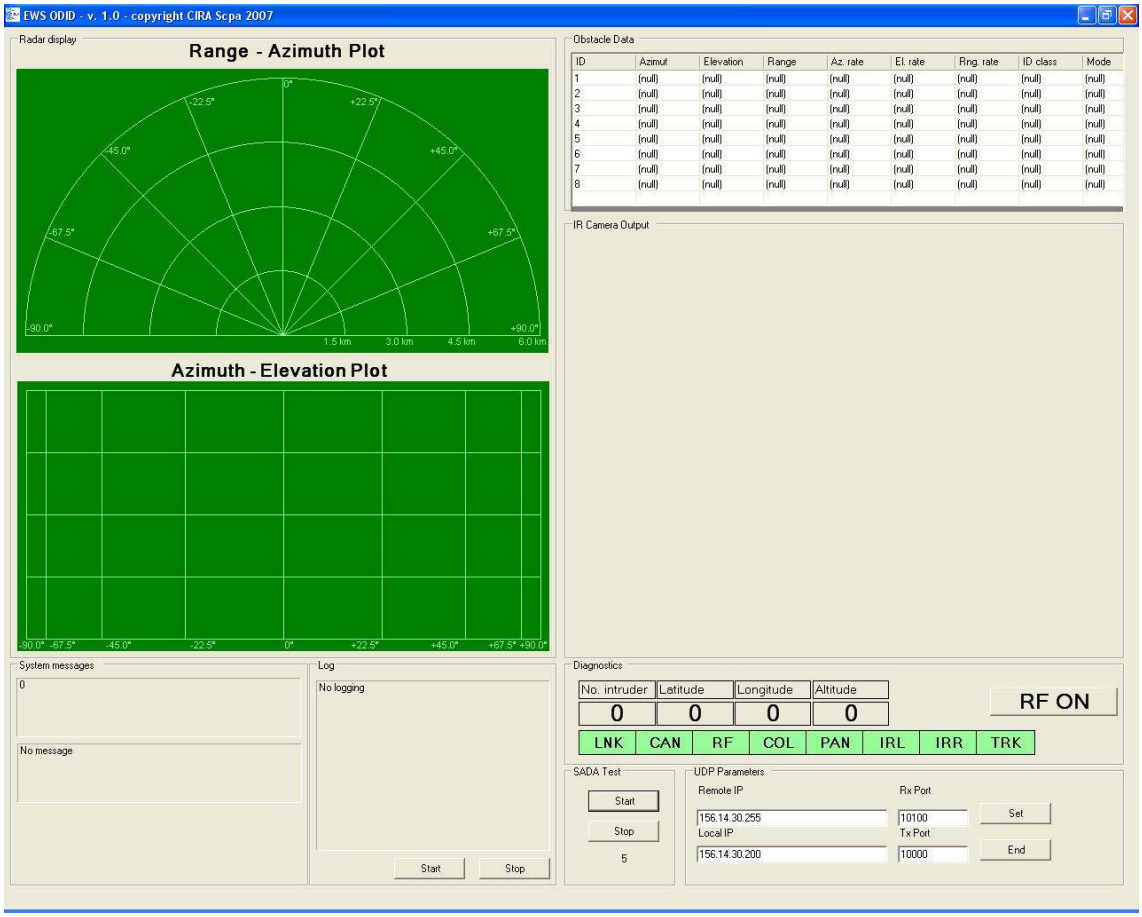

Figure 8 - Typical Ground Station output

5.5. Communication with Amphitech OASys radar

Communication with radar is performed by the Real Time computer by means of an Ethernet connection and the TCP/IP protocol. This is a connection-oriented protocol [83] and thus assures the required reliability to this critical connection.

Indeed, from the software point of view, the developed system interfaces with an Amphitech proprietary software, the OWSInterface application (figure 9). The latter receives raw data from the radar and sends relevant information to the real time system. This choice allowed to develop a customized information protocol without impacting the radar hardware unit. From programming point of view, OWSInterface acts as a TCP/IP server, while the real time software acts as a TCP/IP client, asking for connection in system initialization phase.

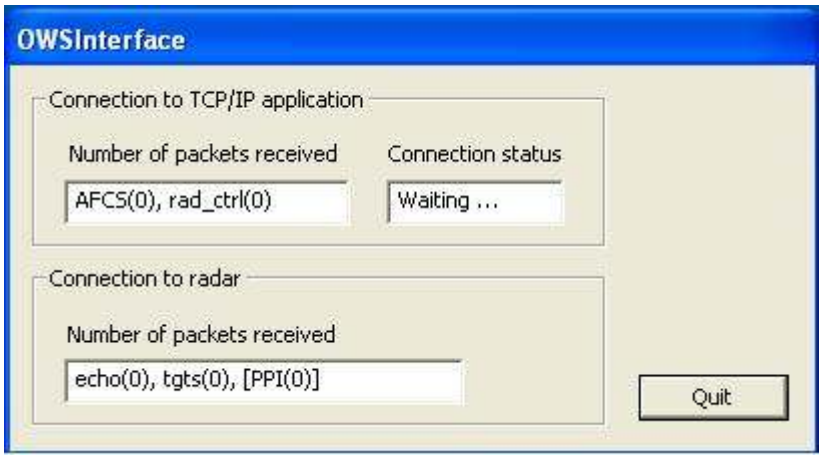

Figure 9 – OWSInterface graphical output

The high level protocol was defined on the basis of several interactions with the radar manufacturer. It allows the send the following commands to the radar:

- required operating mode for radar output;
- unambiguous range;
- scan height and width;
- azimuth and elevation of the center of scan volume with respect to aircraft body axes;

Of course, scan time is affected by changing scan volume parameters.

As for the radar output, three outputs can be received independently from each other:

- targets data;
- raw sectors;
- radar images.

Targets data are used for the tracking filter. They derive from an observation merging operation that is performed at hardware level. Targets data include range, azimuth, elevation, intensity, and delay of data transmission with respect to detection. This latter parameter is very important for tracking accuracy since it allows for correct time referencing of measures. Besides being processed in the tracking module, targets data are sent to ground through CAN and the radio data link.

Raw sectors constitute the non processed radar output. They represent a huge amount of data, and will be recorded to be examined off-line. The same operation will be performed on radar images. Both raw sectors and radar images will not be used for real time tracking.

Communication with radar was tested by means of a hardware unit, the OWS simulator (figure 10).

Figure 10 – OWS simulator

The OWS simulator is a low level simulator which can work with pre-loaded scenarios, as the one showed in figure 11.

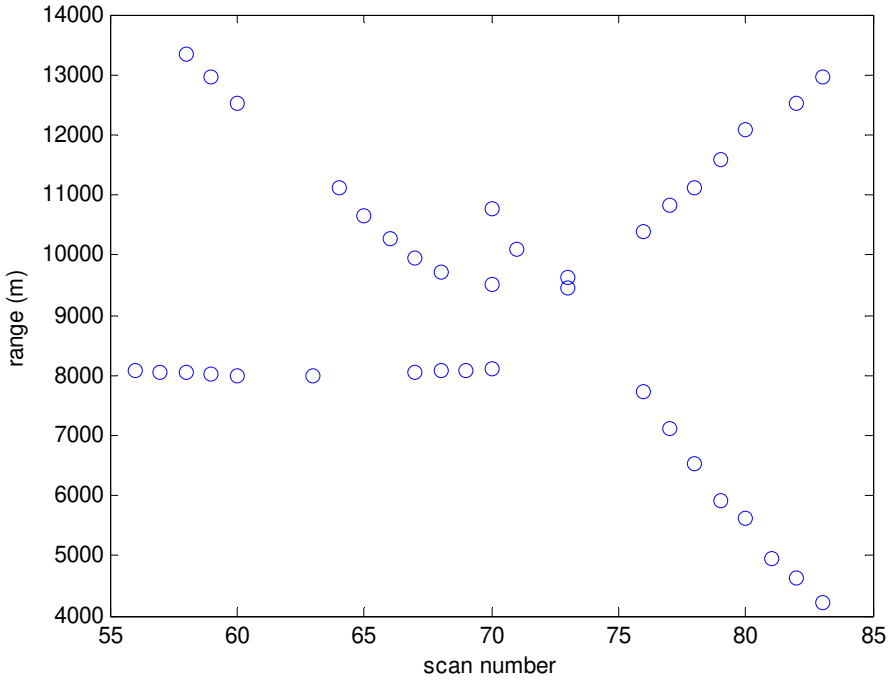

Figure 11 – Example of OWS simulator output

5.6. Hardware-in-the-loop tests

Laboratory tests have been performed to verify hardware real time capability and communication among the different computing units. Tests were performed in collaboration with the CIRA Flight Systems Department (which is responsible for the GNC computer) and the CIRA Test Systems Department (which is responsible for the Communication Controllers). In particular, first tests have been performed to verify CAN bus reliability and performance, since latency in data exchange between the flight control computer and the real time computer plays a key role for the sense and avoid function. Though the average data rate is of the order of only one tenth of CAN bus throughput (1 Mbps), data are transmitted in burst mode at a frequency of 10 Hz and the bus has to solve collision phenomena between packets. Tests were performed to analyze communication links and loops as reported for example in figure 12.

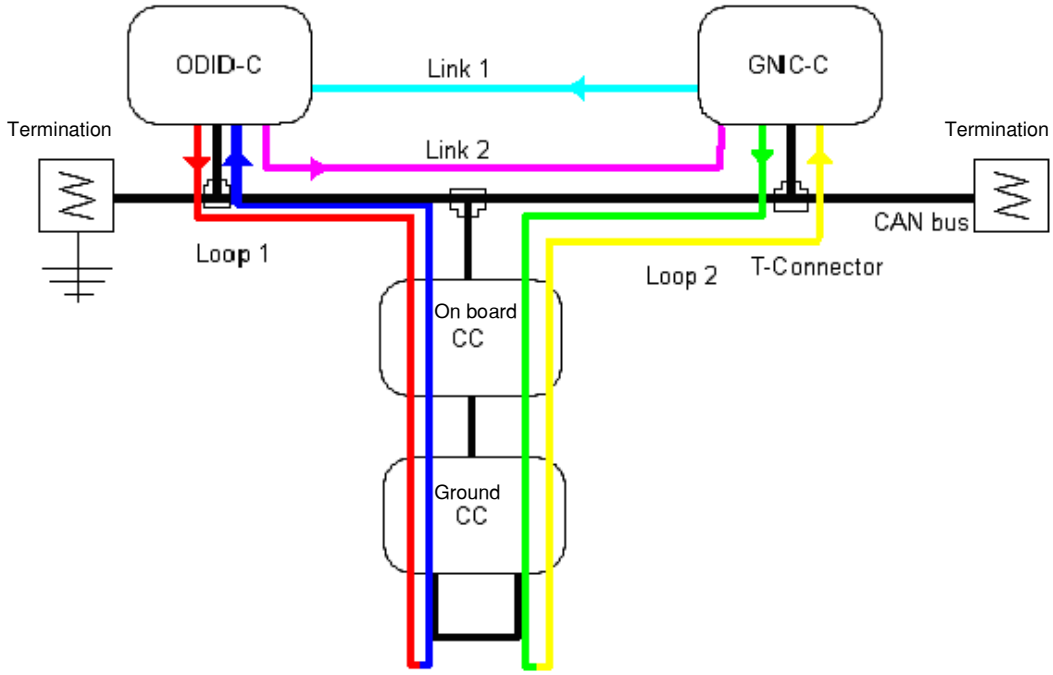

Figure 12 – On board data handling tests

A conservative approach was followed, in order to estimate latency and eventual loss of messages in a worst case situation. For example, the real time obstacle detection computer sent data as in the case of four intruders and eight radar targets detected in each scan. Moreover, loops were structured so as to quantitatively evaluate priority effects on latencies. Regarding the red-blue loop of figure 12, figure 13 reports the measured delay times. On

average, a two-way latency of 20 ms was estimated. Figure 14 points out the effects of messages priority: the different colors refer to different ID, in particular blue, green, black and red curves refer to increasing ID (which means decreasing priority). As expected loop time increases for decreasing priority.

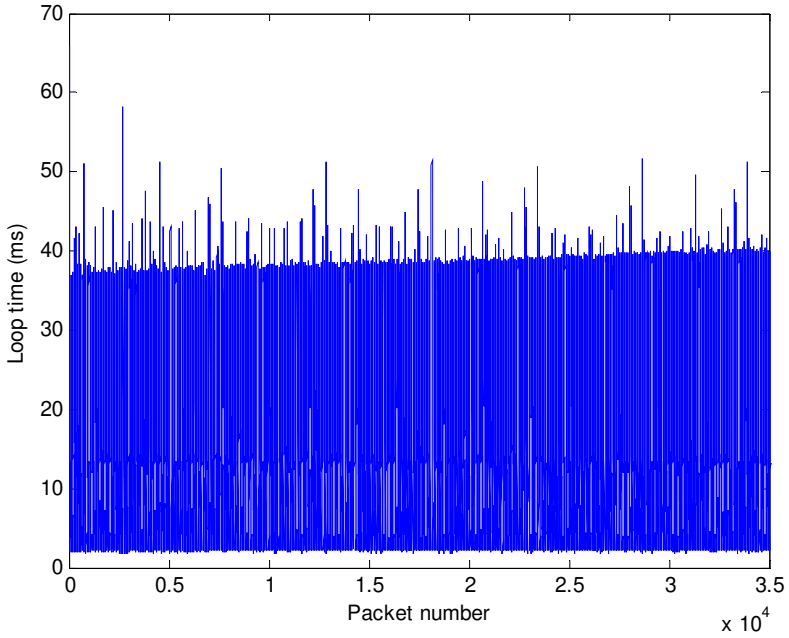

Figure 13 – Loop time for all the CAN packets

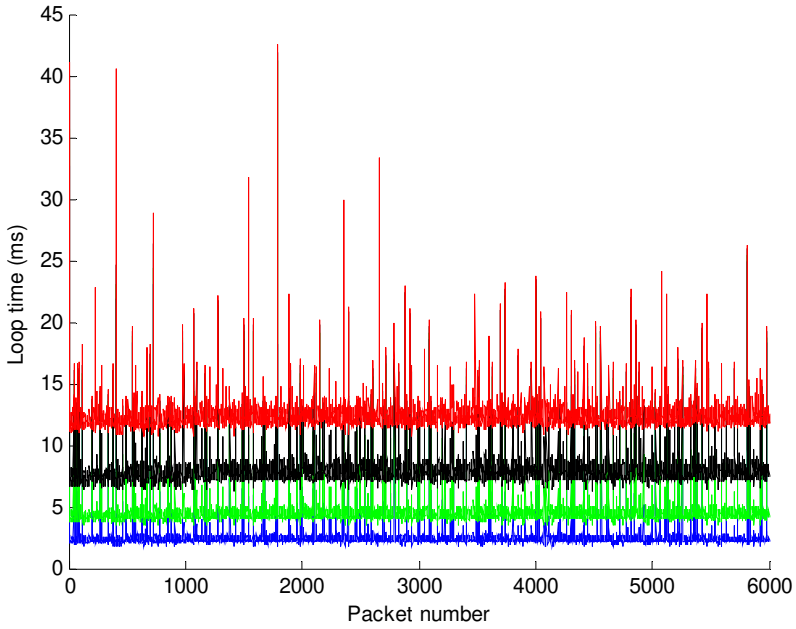

Figure 14 – Loop time for messages with different priority

In order to reduce communication latencies due to burst transmission, an inter-message time of 280 μs has been later introduced. As shown in figure 15 for all the messages, and in figure 16 for 4 identifiers, this slight transmission delay (which does not impact significantly the time required for sending all the messages) allows to reduce collision phenomena between packets, thus reducing latencies.

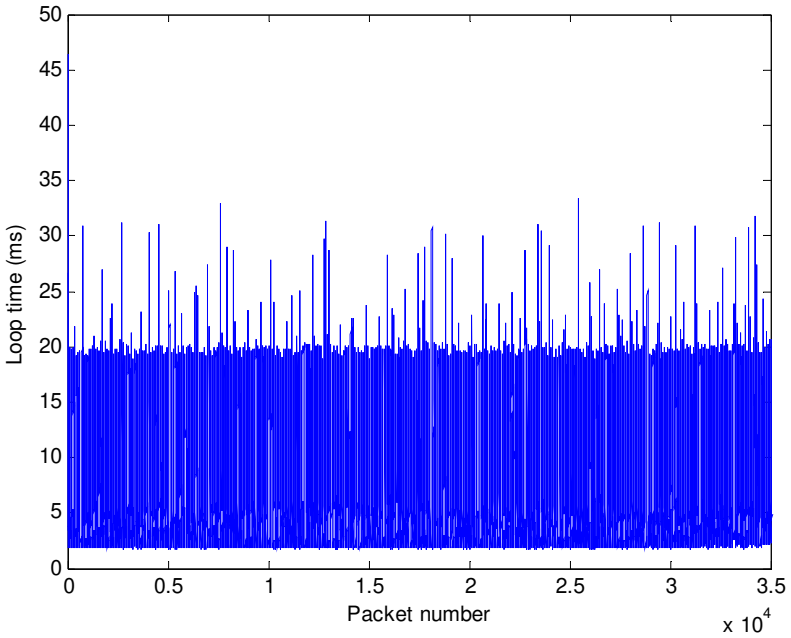

Figure 15 – Loop time for all the CAN packets with introduction of an inter-message time of 280 μs

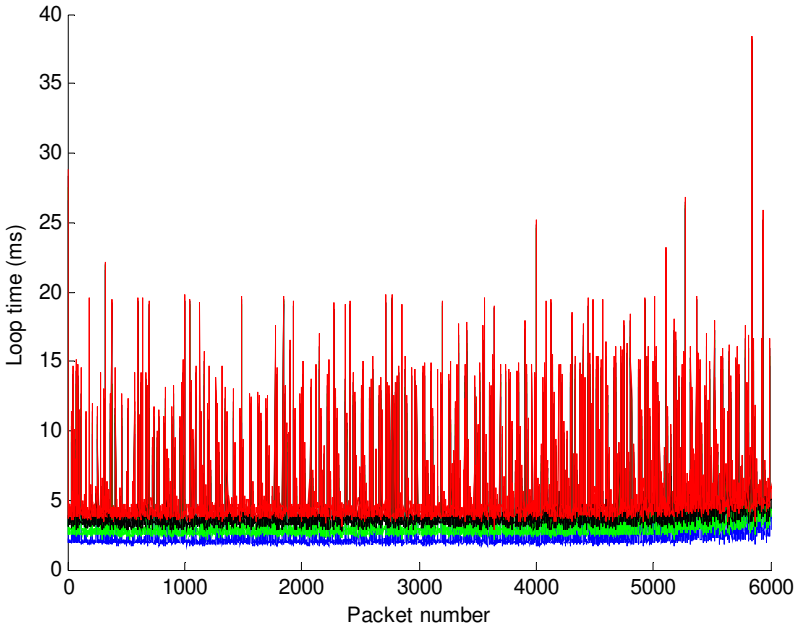

Figure 16 – Loop time for messages with different priority with introduction of an inter-message time of 280 μs

Similarly, a latency of a few ms was estimated in the communication between the real time and the flight control computer. Moreover, a small drift was observed in latencies which is due to non perfect synchronization between the clocks of the two computers. As a systematic effect, it can be easily removed. Anyway, the estimated drift was of the order of 15 ms for a 30 minutes test.

Complete hardware-in-the-loop tests are currently in progress. As shown in figure 17, they are performed to verify the entire flight and ground architecture, except for the radio data link.

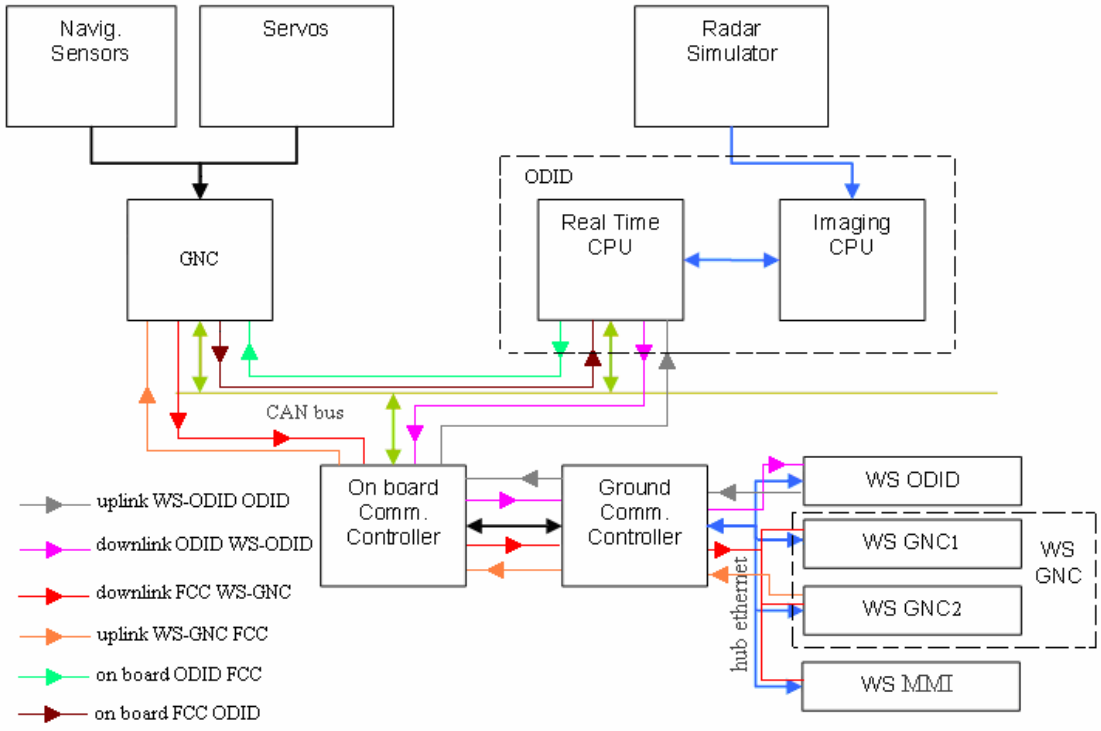

Figure 17 – Architecture for complete hardware-in-the-loop tests

First tests, including the OWS simulator and a dummy processing of its data, have pointed out that real time performance is achieved. For example, figure 18 reports time between consecutive sendings of a CAN packet which represents one of the tracker outputs.

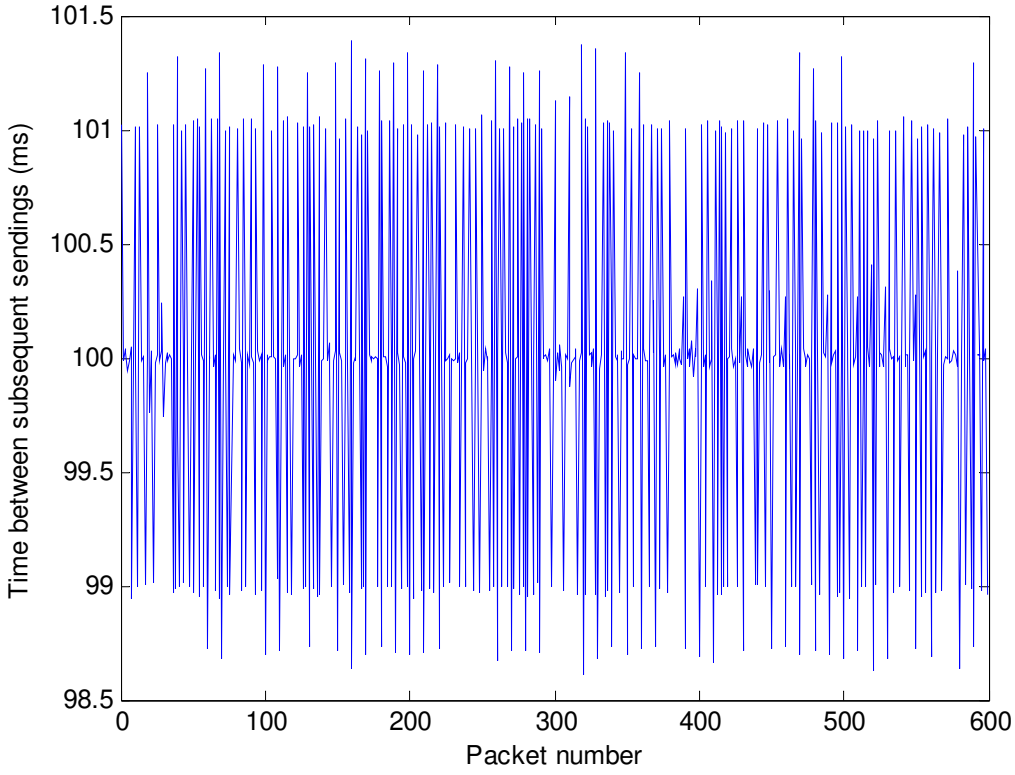

Figure 18 – Time difference between subsequent sendings of a CAN message representing a tracker output parameter

5.7. Real time implementation issues

This section details some real time implementation issues for tracking. The first is related to the fact that the radar system sends its measures with a delay with respect to data acquisition. Since the tracking algorithm generates estimates at 10 Hz frequency, and this delay is likely larger than 0.1 s (it can be of the order of 1 s), this means that state and covariance have to be updated with a measurement at time t_l after they have been propagated to a later time t_n. Also, gating and association have to be performed with the values of state and covariance at time t_l. It is important to note that, even though collision avoidance logic did not request it, in any case the system should work at 10 Hz frequency in order to properly follow UAV dynamics through the deterministic input (see chapter 3).

In order to understand how is the problem treated, first of all it is important to point out that in the developed system a measurement is assigned a time stamp which is the nearest time on the tracker 0.1 s scale.

To perform gating, association and track updating, the solution is to keep a sliding window where navigation and track data are stored. The considered time window dimensions correspond to the largest possible radar data latency. However, it is important to evaluate which navigation and track parameters have to be stored. Thus, let us consider a starting time t_1 when state is x_1 and covariance is P_1. State and covariance are propagated in time on the basis of transition matrix, process noise matrix and deterministic input (that is, own-ship accelerations converted in NED). After n tenths of seconds (predicted state is x_n and predicted covariance is P_n), a radar measurement is sent to the tracker, which refers to time t_1. The problem is how to produce an updated state and covariance estimate at time t_n given the measurement at time t_1. First of all, in order to perform gating and association, state and covariance at time t_1 have to be known: thus, they are supposed to have been stored. Moreover, attitude angles must be stored to perform measurement conversion in NED. Then, updated estimates at time t_1 can be calculated on the basis of standard Kalman filter equations. Let us define as x_i and P_i the state and covariance estimates at time t_i before measurement inclusion, $\overline{x_i}$ and $\overline{P_i}$ the filtered estimates at time t_i, and $\delta x_i = \overline{x_i} - x_i$, $\delta P_i = \overline{P_i} - P_i$.

If u_i represents deterministic input at time t_i, following equations describe state prediction:

$$x_2 = \Phi x_1 + u_1 \tag{1}$$

$$x_3 = \Phi x_2 + u_2 = \Phi^2 x_1 + \Phi u_1 + u_2 \tag{2}$$

…

…

$$x_n = \Phi^{n-1} x_1 + \Phi^{n-2} u_1 + \Phi^{n-3} u_2 + ... + \Phi u_{n-2} + u_{n-1} \tag{3}$$

In the same way, covariance prediction is described by

$$P_n = (\Phi)^{n-1} P_1 \left(\Phi^{|}\right)^{n-1} + (\Phi)^{n-2} Q^{n-2} \left(\Phi^{|}\right)^{n-2} + ... + Q \tag{4}$$

State and covariance predictions are linear functions of x_1 and P_1, respectively. Thus, $\overline{x_n}$ and $\overline{P_n}$ can be evaluated as follows:

$$\overline{x_n} = x_n + \Phi^{n-1} \delta x_1 \tag{5}$$

$$\overline{P_n} = P_n + (\Phi)^{n-1} \delta P_1 \left(\Phi^{|}\right)^{n-1} \tag{6}$$

This implies that acceleration data must not be stored to provide filtered estimates of a track with delayed measurements, since x_n already contains this information. At the same time, since covariance propagation depends only on the transition matrix and the process noise matrix, supposing that a non adaptive model is used (both matrices are constant) it is possible for each track to keep in memory only the current covariance. In this case, P has to be extrapolated back in time when necessary, using a-priori known matrices.
However, it is worth noting that acceleration and velocity estimates are needed for initialization of one-plot tracks. Thus, it can be concluded that they do not need to be stored if a firm track is already present, in collision avoidance flight tests with a single intruder.

Another relevant problem for real time implementation is connected to ground echoes removal. In fact, due to the types of aircrafts that are used in tests (very light aircrafts) flight altitude can be very low, so that ground metal objects can give, also depending on relative geometry, big signal reflections. Ground objects can be identified on the basis of their velocity; however, since the radar gives no Doppler information, this requires a tentative tracking phase, which means that at least two subsequent echoes from ground have to be associated with negative consequences on computational time. In order to avoid wasting time with useless calculations, single scan criteria are also required to separate ground objects.
Of course, the most immediate criteria can be based on apparent intruder altitude. It is easily calculated on the basis of radar measurement, UAV altitude and attitude angles. A simple logic is to eliminate a return if altitude is below a given threshold. For the sake of concreteness, let us consider that this threshold is 100 m. Unluckily, this solution is too simple and unreliable, since it does not take into account the uncertainties in intruder altitude estimation: a valid intruder echo can be eliminated.
A more appropriate solution can be to eliminate a return only if it is, with a high confidence level, a ground object. In case of uncertainty, tentative tracking has to be performed and combined altitude/velocity estimation can be used.
A simplified approach can be followed if only pitch angle is considered for the UAV. In this case, apparent altitude of a radar detected object is given by

$$h_{object} = h_{UAV} + R \sin \vartheta_{NED} \tag{7}$$

where R is the measured range, and θ_{NED} is the stabilized elevation, which includes pitch and object elevation with respect to the radar.

Uncertainty on relative altitude measurement can be evaluated by the following formula:

$$\sigma^2 \left(R \sin \vartheta_{NED} \right) = \sin^2 \vartheta_{NED} \sigma^2 \left(R \right) + R^2 \cos^2 \vartheta_{NED} \sigma^2 \left(\vartheta_{NED} \right) \tag{8}$$

while global uncertainty on object altitude is given by

$$\sigma^2 \left(h_{object} \right) = \sigma^2 \left(h_{UAV} \right) + \sin^2 \vartheta_{NED} \sigma^2 \left(R \right) + R^2 \cos^2 \vartheta_{NED} \sigma^2 \left(\vartheta_{NED} \right) \tag{9}$$

The impact of the second term can be likely neglected, due to the fact that elevation is typically small and the radar is very accurate in range. Moreover, $\cos \vartheta_{NED}$ can be approximated to 1. Thus,

$$\sigma \left(h_{object} \right) \approx \sqrt{\sigma^2 \left(h_{UAV} \right) + R^2 \sigma^2 \left(\vartheta_{NED} \right)} \tag{10}$$

The most important of the two terms is the second contribution, which combines uncertainty on object elevation, radar alignment, and pitch estimation.

Since $\sigma \left(h_{object} \right)$ depends on range, a single scan criterium can be based on object apparent position in the range-altitude plane. In particular, if $h > h_{threshold} + \sigma \left(h_{object} \right)$, then the object is with high confidence a flying object. Inversely, if $h < h_{threshold} - \sigma \left(h_{object} \right)$, the object is on ground. In other cases, uncertainty remains and tentative tracking is required.

Criterium in the range-altitude plane is shown in figure 19. For the sake of concreteness, $\sigma \left(\vartheta_{NED} \right)$ has been assumed equal to 2.5°, and values of 10 m (standalone GPS) and 1 m (differential GPS) have been considered for $\sigma \left(h_{UAV} \right)$.

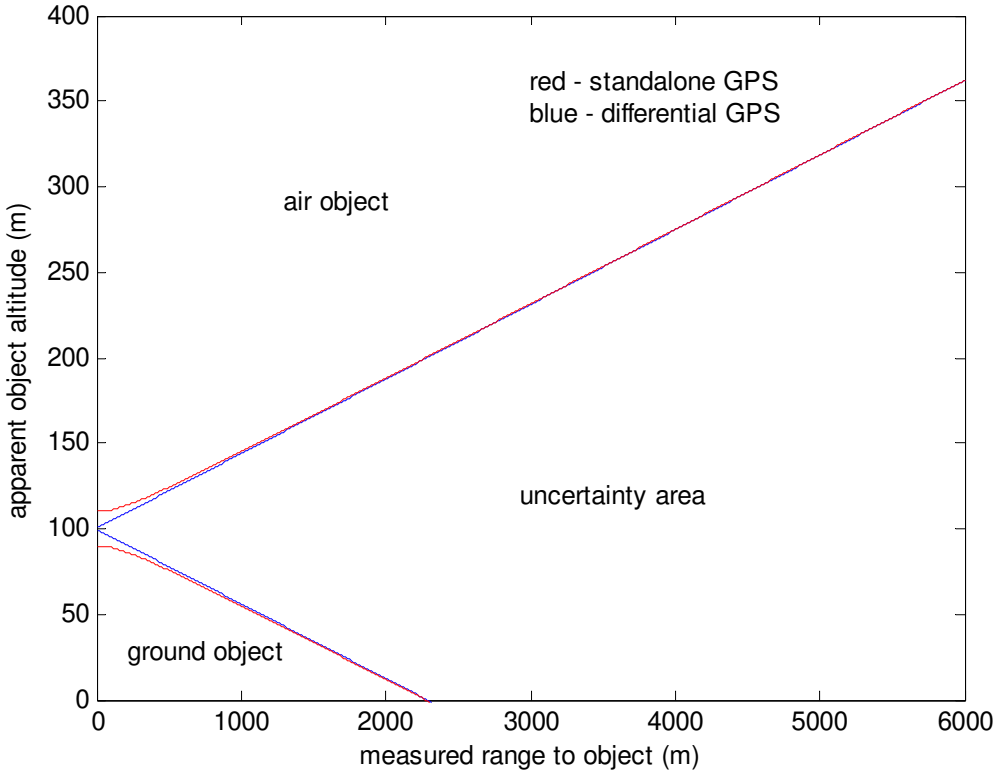

Figure 19 – Range-altitude plane for ground echoes rejection

Since the most important contribution to altitude uncertainty comes from stabilized elevation accuracy, the difference between standalone and differential GPS appears only al very low ranges. It is also worth noting that low altitude returns at large distances in substance can not be filtered because of the effects of angular uncertainty. However, probability of ground object detection reduces for increasing distance. On the other hand, in the range-altitude plane it has to be supposed that an upper threshold is considered on range axis, which corresponds to maximum estimated detection range for the largest possible intruder. All echoes with larger measured range are automatically eliminated.

This logic can also used to understand the effects of UAV altitude for sense and avoid flight tests. For increasing UAV altitude, minimum slant range to ground obviously increases, thus reducing ground clutter effects. In particular, figure 20 refers to a 10° (negative) stabilized elevation angle.

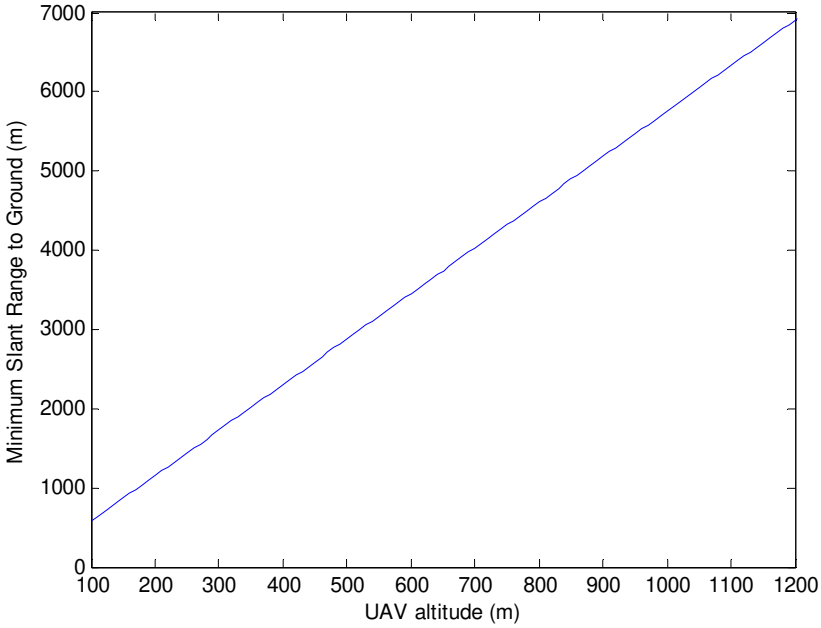

Figure 20 – Minimum slant range to ground as a function of UAV altitude (-10° elevation angle)

Considering for example a maximum range for detection of 8 km, and the lower range threshold from figure 19 (2.2 km), it is then possible to calculate which part of the field of regard in elevation can give "ambiguous" ground returns, that is echoes that can not be eliminated with the simple range-altitude logic. It is depicted in gray in figure 21.

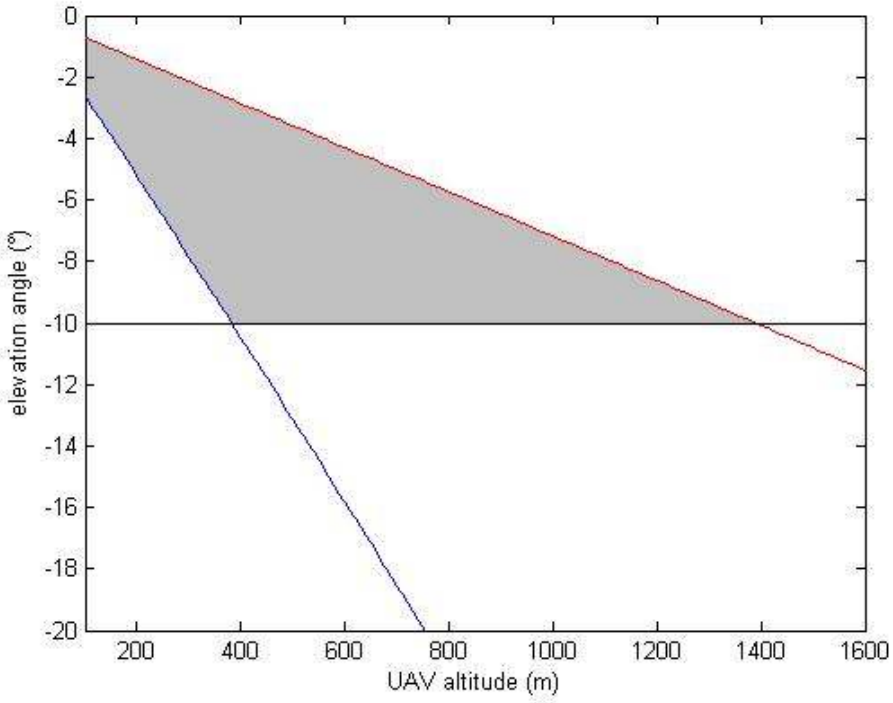

Figure 21 – Range of angles in elevation that correspond to ambiguous ground echoes

It is worth noting that the ambiguous region at first increases, due to the fact that at low altitude distance from ground is less than 2.2 km in a large part of the radar FOR, and so ground echoes can be eliminated on the range-altitude plane. Then, after a maximum, it decreases for increasing altitude.

Nevertheless, it can be estimated that due to probability of detection effect, computational load for ground echoes removal decreases for increasing altitude in any condition. In any case, the threshold parameters for ground echoes rejection will be set after the first flight tests with the radar, on the basis of sensor returns.

Chapter 6

Accurate EO sensors alignment by means of DGPS, AHRS and image analysis

6.1. Introduction

As demonstrated in the previous chapters, one of the most important features of the electro-optical sensors is the high accuracy angular information they can provide thanks to their high resolution in pixels.

Of course, besides being dependant on the sensor resolution, overall angular error is also due to sensors mounting errors which introduce biases. It is worth noting since now that conceptually, in this application alignment error refers to the error computed with respect to the body reference frame as individuated by the attitude heading reference system (AHRS). In fact, navigation data as provided by the AHRS are used for autonomous flight by the flight control computer.

Indeed, obstacle detection for collision avoidance is not the only application that requires this kind of alignment. For example, an other case where it is very important to accurately estimate EO sensors orientation with respect to the AHRS defined body reference frame is in real time surveillance, for example for natural disasters monitoring [87].

In these cases, navigation sensors information is essential for correct geo-registration of acquired multispectral/thermal/hyperspectral images, and the error in geo-location depends on navigation, image processing and EO sensors alignment errors. Moreover, it is worth noting that in these applications it is unlikely to have the possibility to use ground control points for geo-registration and geo-location.

In general, several techniques can be used for EO sensors alignment which are based on interferometric, mechanical, or image processing techniques [88]. For example, a stereoscopic couple can be calibrated on the basis of analysis of a sample pattern, and relative translation and rotation between cameras are some of the extrinsic parameters that are calculated [89]. However, traditional methods or algorithms are hard to use to provide calibration directly with respect to the AHRS. In the considered case, instead, a low-cost and relatively fast procedure is needed, also to evaluate cameras' angular displacement caused by successive flights.

In this chapter, a procedure is illustrated to provide fast and accurate geometric calibration of EO sensors taking advantage of the navigation instrumentation, namely AHRS and GPS, which are very accurate in static mode and in carrier phase differential Real Time Kinematic mode, respectively.

The chapter is organized as follows. The following section describes the hardware setup (navigation) which was used to test the alignment technique. In the third section, the calibration procedure is illustrated in detail, pointing out both theoretical and practical issues. Then, in the forth section results achieved during different calibration sessions are illustrated and reported in detail. Potential and limitations of the designed procedure, as well as the lessons learned, are finally analyzed.

6.2. Navigation hardware

Electro-optical sensors have been described in chapter 2. As for navigation sensors, the central unit is the AHRS400CC™ manufactured by Crossbow™. The AHRS400CC is a high-performance solid-state attitude and heading reference system. In static mode, by averaging sensors output for some seconds (data rate is 100 Hz), it is possible to reach an accuracy of the order of 0.1°.

The ground GPS antenna is the LegAnt™ manufactured by Topcon™, whereas other two GPS antennas are located on the aircraft wings.

The measurement technique is the Real Time Kinematic (RTK) carrier-phase differential mode. This is the highest accuracy mode available with GPS signal [90,91]. In particular, accuracy can be within 1 cm. RTK is a process where GPS signal corrections are transmitted in real time from a reference receiver at a known location to another receiver. The use of an RTK capable GPS system can compensate for atmospheric delay, orbital errors and other variables in GPS geometry, increasing positioning accuracy. Using the code phase of GPS signals, as well as the carrier phase, which delivers the most accurate GPS information, RTK provides differential corrections to produce the most precise GPS positioning.

Navigation hardware is reported in table 1.

AHRS	**GPS ground antenna**
	

Table 1 – Navigation hardware for alignment procedure

6.3. Procedure description

The developed procedure allows to calibrate all the EO sensors together, at the same time. In fact, it needs in substance that a number of images of a target (at least 2) are acquired by all the cameras, while at the same time aircraft attitude is measured by the AHRS and target position is measured by CDGPS in RTK mode. At the end of the acquisitions, cameras' positions must be measured by CDGPS with the same level of accuracy. A phase of the measurement procedure is illustrated in figure 1.

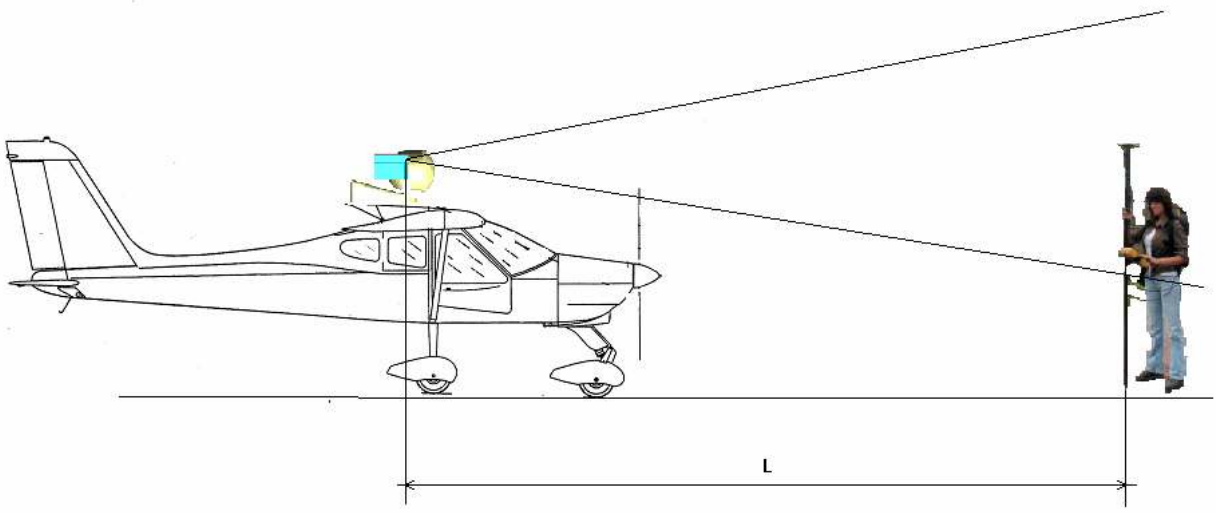

Figure 1 - Measurement procedure

In order to establish how many targets positions are to be measured, and at what distance the target must be placed, both CDGPS accuracy and sensors IFOV must be taken into account.
In fact, in theory the best solution would be to place the target as far as possible from the sensor, so that the GPS error falls below single pixel angular dimensions. However, this makes target positioning harder to realize. In fact, in order to have a globally accurate alignment, the test points should be selected uniformly in the cameras field of view. In the considered case, the relevant accuracies are shown in table 2.

Measurement Precision of GPS TOPCON Legacy-H in carrier-phase mode	3 mm + 1 ppm
Visibile cameras Instantaneous Field Of View (IFOV)	0.041°
IR cameras Instantaneous Field Of View (IFOV)	0.074°
VIS cameras Field Of View (FOV) (at res. 1280 X 960)	48.6° (H) x 37.6° (V)
IR cameras Field Of View (FOV) (for each camera)	24° (H) x 18° (V)

Table 2 - Relevant accuracies and fields of view

By a simple geometric relationship, it is possible to determine at what distance the GPS precision equals the linear dimension which corresponds to the cameras IFOV. Some numerical data for the considered case are shown in table 3.

Distance	[m]	2	4	5	10	20
Width FOV VIS	[m]	1,805	3,610	4,513	9,025	18,050
Height FOV VIS	[m]	1,361	2,722	3,402	6,805	13,610
Width FOV IR	[m]	0,850	1,700	2,124	4,249	8,498
Height FOV IR	[m]	0,633	1,266	1,583	3,166	6,332
Length IFOV VIS	[m]	0,001	0,003	0,004	0,007	0,014
Length IFOV IR	[m]	0,003	0,005	0,006	0,013	0,026

Table 3 - Linear dimensions corresponding to FOV and IFOV at different distances

From table 3, it can be concluded that in the considered case the procedure can be implemented by locating the target at a distance of about 4 meters from the focal plane of the sensors and moving it in a rectangle of about 4 m X 3 m. Thus, this distance was selected in the performed calibration tests. A picture describing a phase of a calibration session is shown in figure 4.

Figure 2 - Calibration procedure: acquisition of target images and position

In order to determine the rotation matrix between sensors' reference frames and aircraft body reference frame (X-nose, Y-right wing, Z-down), a classical least squares technique was adopted (the Wahba method), which estimates the transformation matrix on the basis of a series of vectorial observations of the same points in the two reference frames [92]. In what follows, body reference frame will be considered as a synonym of AHRS-defined reference frame.

The basic assumption of Whaba method is that the main component of the error of the single observations is random, thus it is supposed that the camera is perfectly calibrated. This means, among other things, that optical distortion is neglected. It can be stated that this assumption can be considered consistent with the scopes of this application, also because of the narrow field of view of the cameras. On the other hand, the validity of this assumption can be verified by analyzing the residual errors after camera alignment, which is reported in the following.

Given the target and the camera position in the Earth Centered Earth Fixed (ECEF) Reference Frame, it is possible to evaluate target position with respect to the East North Up (ENU) reference frame with origin in the camera, through an exact transformation, if the Earth geoid model is known [93].

Then, the target position in camera-based ENU $\underline{r}_{iENU}$ can be transformed in the Body reference frame (again, with origin in the considered camera), on the basis of the attitude angles measured by the AHRS, by equation 1:

$$\underline{r}_{iBRF} = M_{321}(\gamma, \beta, \alpha)\underline{r}_{iENU} \tag{1}$$

where γ, β and α are, respectively, the yaw, pitch and roll angle, and the matrix M_{321} is obtained as follows:

$$M_{321} = \begin{bmatrix} \cos\beta\cos\alpha & \cos\beta\sin\alpha & -\sin\beta \\ -\cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha & \cos\gamma\cos\alpha + \sin\gamma\sin\beta\sin\alpha & \sin\gamma\cos\beta \\ \sin\gamma\sin\alpha + \cos\gamma\sin\beta\cos\alpha & -\sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha & \cos\gamma\cos\beta \end{bmatrix} \tag{2}$$

By dividing $\underline{r}_{BRFi}$ by its modulus, it is possible to evaluate cosine directors of target in BRF with origin in a given camera. Let us call $\hat{\underline{r}}_{iBRF}$ the computed unit vector.

On the other hand, image analysis allows to find target centre pixel and thus its cosine directors in the camera reference frame. In order to derive cosine directors, a classical pinhole camera model can be assumed [94]. In particular, pixel coordinates are converted in linear coordinates on the image plane on the basis of pixel dimensions, then target direction is computed through the focal distance. Considered geometry is depicted in figure 3, where the image plane is represented in front of the projection center, in order to avoid sign inversion.

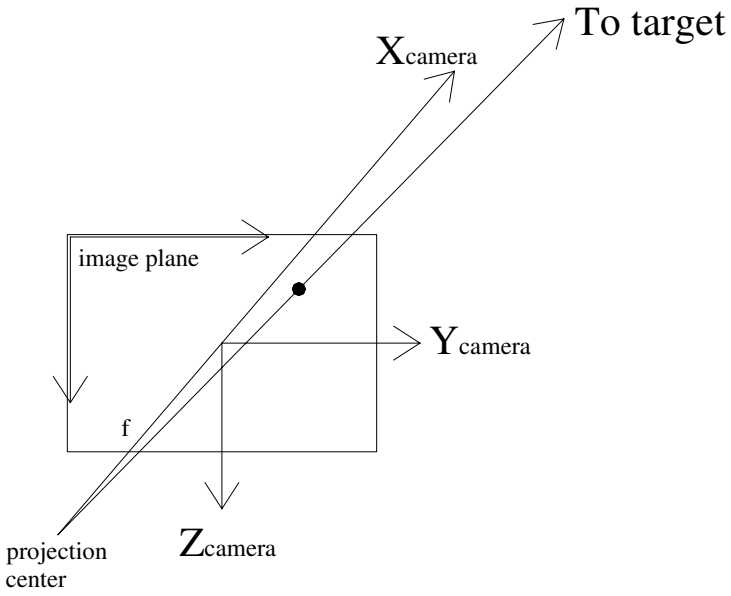

Figure 3 – Pinhole camera model

Let us call $\hat{\underline{r}}_{iSENS}$ the unit vector representing target direction in the camera based reference frame. It is worth noting that camera reference frames have been chosen with the same convention of BRF (X nose-Y right-Z down).

It is now possible to define a loss function

$$J(M_{CAM}) = \sum_{i=1}^{n} w_i \left| \hat{\underline{r}}_{iSENS} - M_{CAM} \hat{\underline{r}}_{iBRF} \right|^2 \qquad (3)$$

where n is the number of collected images/positions, w_i is the weight of measurement i, M_{CAM} is the attitude matrix of the considered camera with respect to the aircraft, and as stated before $\hat{\underline{r}}_{iSENS}$ is the unit vector of the target in image i with respect to the considered sensor and $\hat{\underline{r}}_{iBRF}$ is the unit vector of target in image i with respect to the BRF with origin in the sensor position, which is computed by means of GPS/AHRS measurements. The camera attitude matrix M_{CAM} can be computed to minimize the loss function by means of the elegant algorithm developed by Wahba [92]. In the considered case, all the measurements have the same weight.

It is worth noting that since the considered procedure extracts only the rotation matrices between the cameras and the AHRS reference frame, it does not consider the so called parallax error, due to the distance among EO sensors and inertial unit. The parallax effect depends on the distance of a given target: thus, when using EO data for real time tracking, this error has to be corrected on the basis of range measurement.

6.4. Test results

Two different calibration campaigns were conducted in order to evaluate developed procedure accuracy and also to verify stability of cameras' orientation after flight operations. In fact, the two calibration sessions were performed before and after some FLARE flights devoted to other TECVOL functions.

Next figures report some examples of images taken during calibration sessions. In particular, figures 4, 5, 6, and 7 report images taken by the panchromatic camera, the color camera, the left IR camera, and the right IR camera, respectively.

Figure 4 – Example of a color image taken during a calibration session

Figure 5 - Example of a panchromatic image taken during a calibration session

Figure 6 – Example of an IR image taken during a calibration session (left camera)

Figure 7 – Example of an IR image taken during a calibration session (right camera)

6.4.1. First calibration session

Next figure reports the time variation of panchromatic camera position ECEF components. It can be seen that these variations are of the order of a few cm. They have been averaged to increase position measurements accuracy.

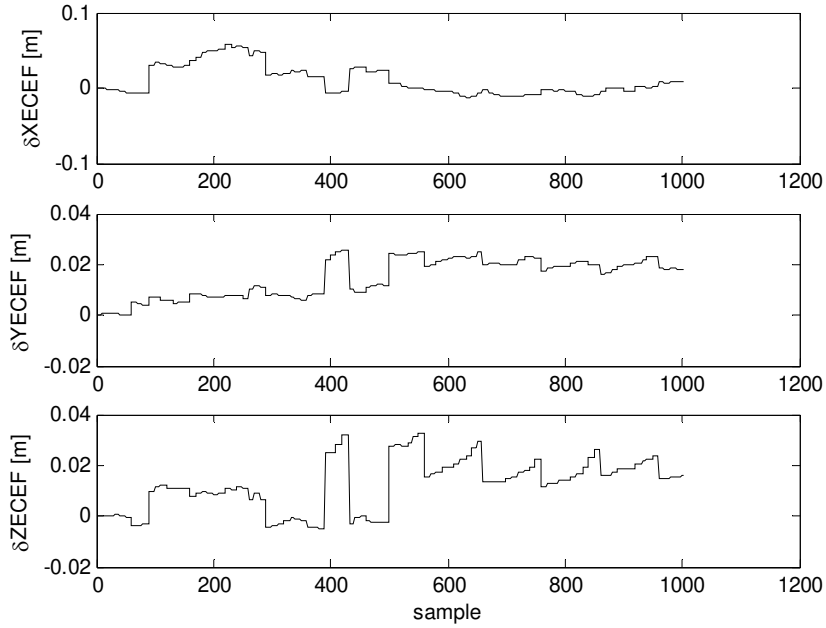

Figure 8 – Variation of measured ECEF components during acquisition of panchromatic camera position

On the basis of estimated attitude matrices and the Whaba algorithm, the Euler angles of cameras with respect to the body reference frame as given by AHRS have been estimated and are given by table 4. Only one of the two IR cameras is considered because the other camera was not installed at the time of this calibration session.

Estimated Angle (°)	Panchromatic	Color	IR
Roll	8.393249	5.310281	-0.380102
Pitch	1.078569	0.461661	-0.151451
Yaw	-1.211358	4.311618	-1.589749

Table 4 - Estimated cameras attitude angles (1st session)

Next figures report targets positions in panchromatic (figure 7), color (figure 8) and IR (figure 9) images for the first acquisition session as measured by GPS/AHRS and converted through the calculated rotation matrix (in red) and as obtained by image analysis (in blue):

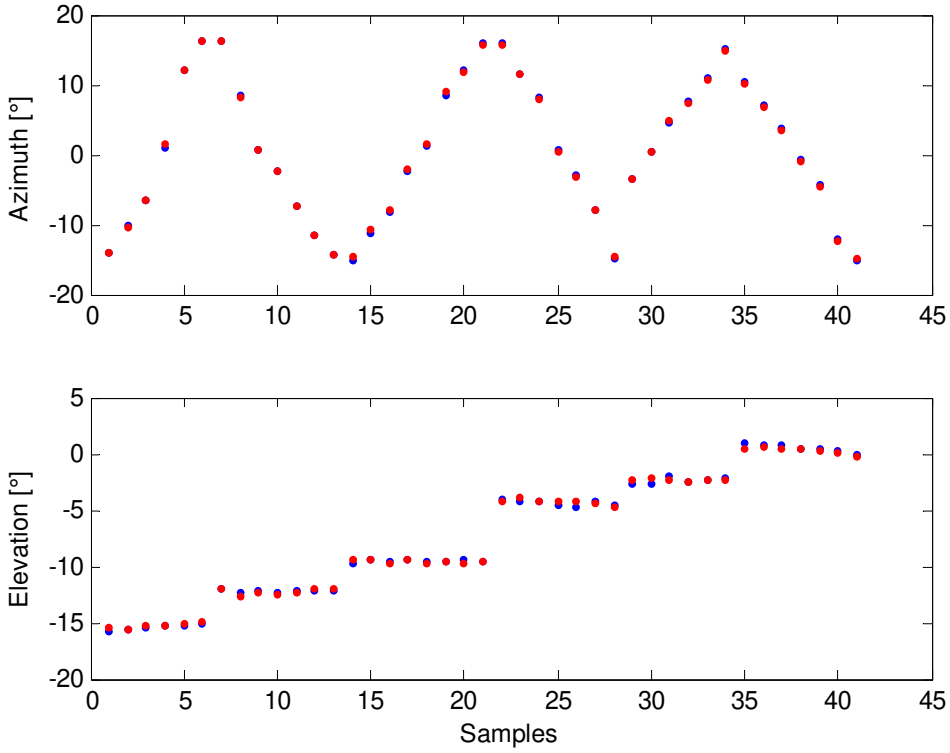

Figure 9 - Target positions as extracted from images (blue) and from GPS/AHRS measurements and computed rotation matrix (red) : panchromatic camera, 1st calibration session

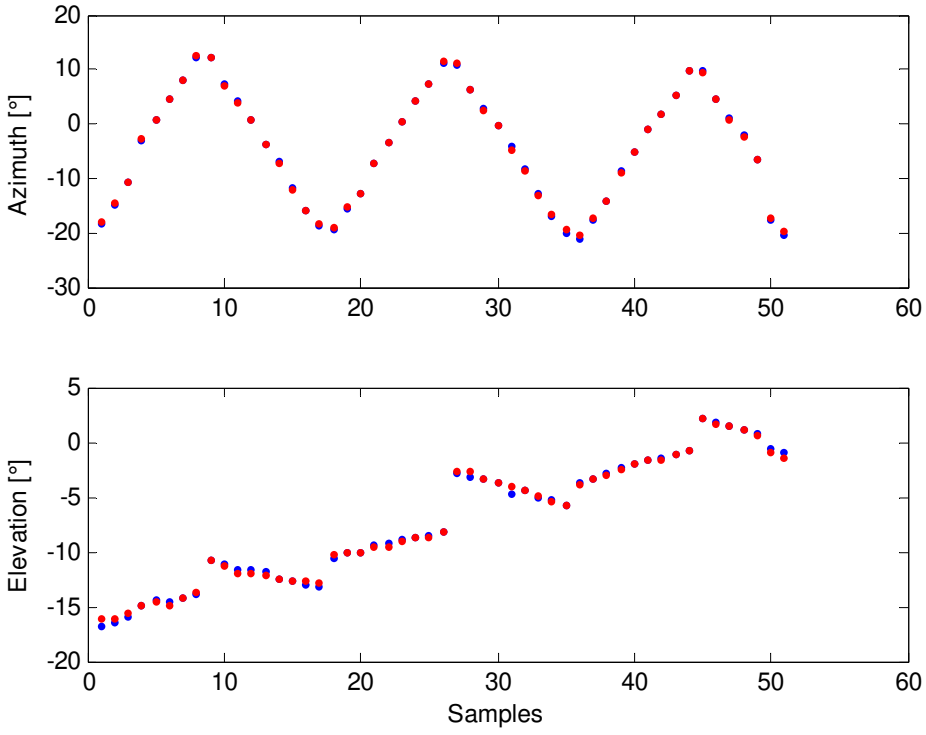

Figure 10 - Target positions as extracted from images (blue) and from GPS/AHRS measurements and computed rotation matrix (red) : color camera, 1st calibration session

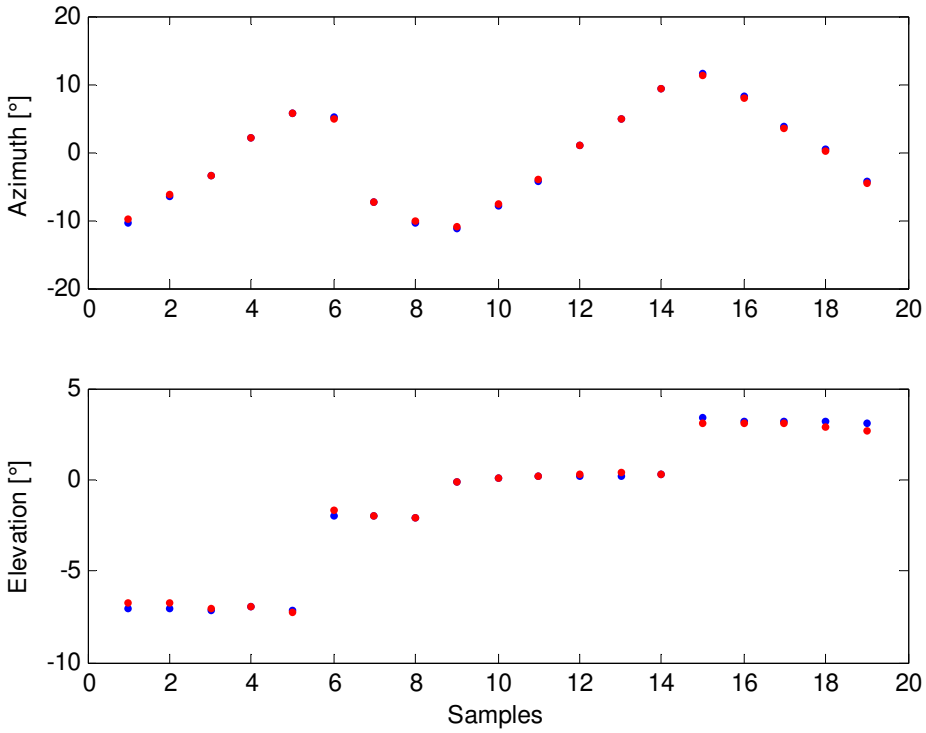

Figure 11 - Target positions as extracted from images (blue) and from GPS/AHRS measurements and computed rotation matrix (red) : infrared camera, 1st calibration session

Residual angular errors on test points are reported in the following. For the sake of brevity, only the panchromatic camera is considered. Angular errors are reported in figure 12. Vector representation of angular errors on the azimuth-elevation plane shows no significant systematic effects (figure 13). On the other hand, the normality of the distribution can be verified by applying a normal probability plot to the angular residuals, as shown in figure 14.

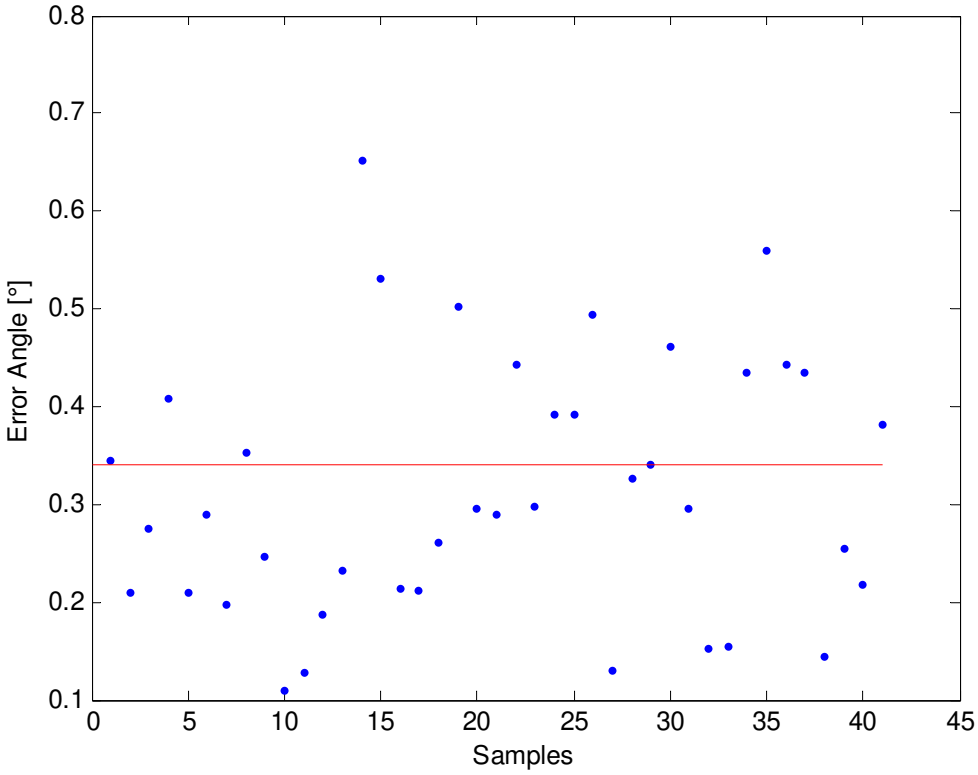

Figure 12 - Residual angular errors for panchromatic camera (1st session)

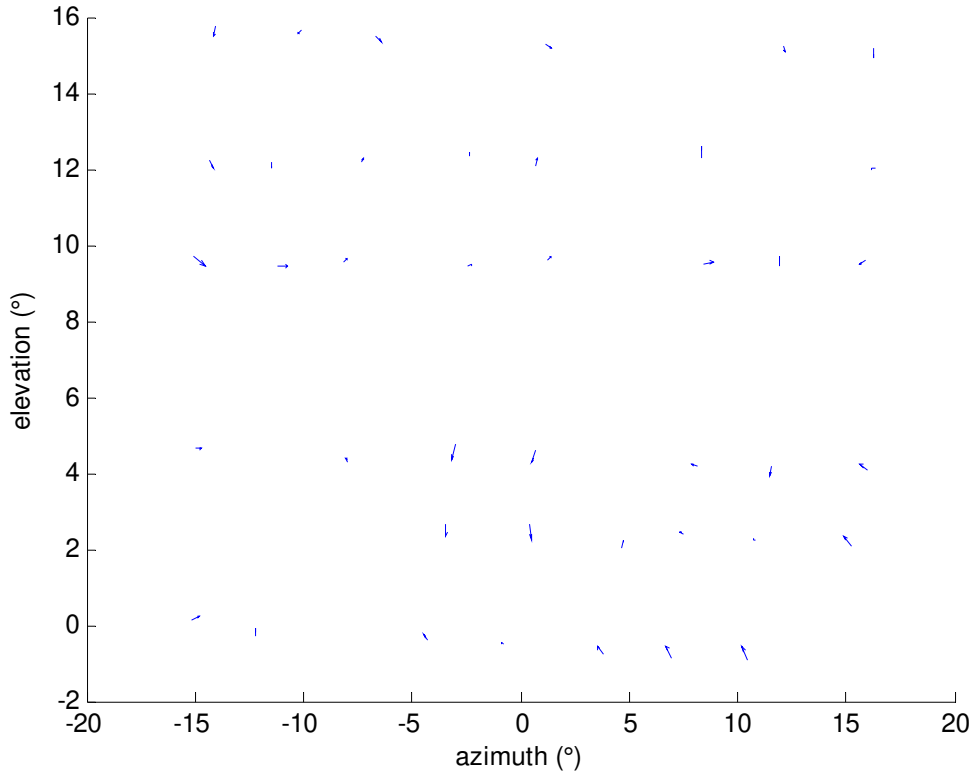

Figure 13 – Vector representation of azimuth and elevation differences (panchromatic camera, 1st session)

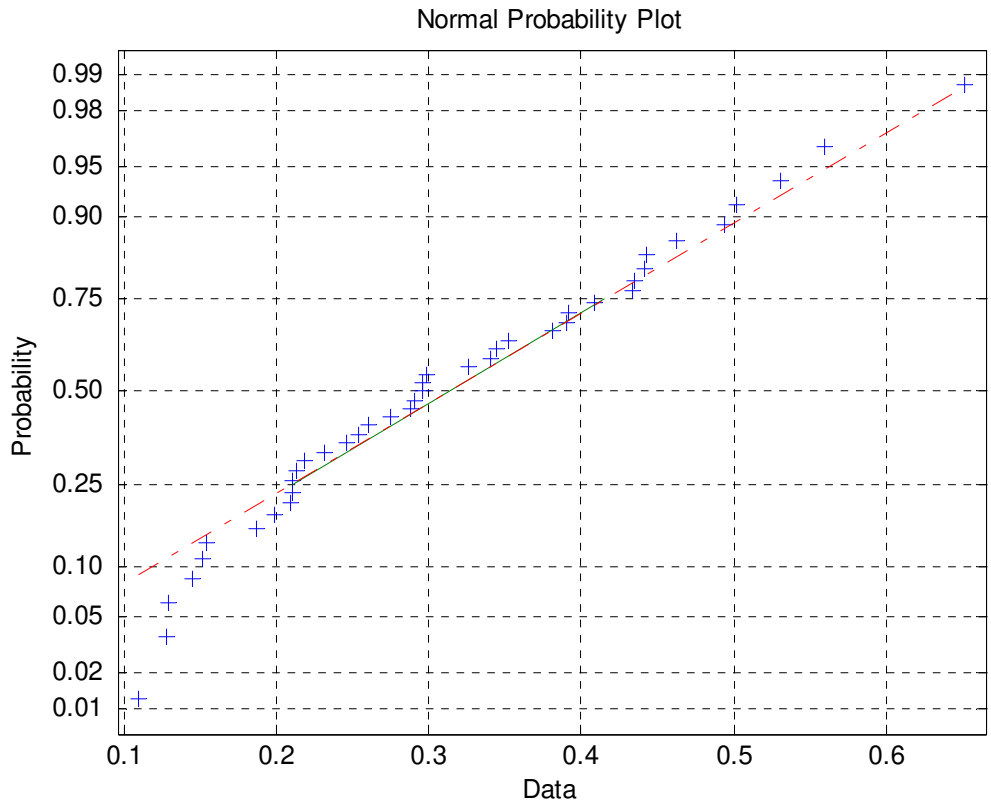

Figure 14 - Normal probability plot for angular residuals: panchromatic camera, 1st session

It can be seen that the error is of the order of 0.3° RMS. This value can be attributed to the low contrast in some images and can reasonably be reduced if better quality images are recorded. Due to the normality of the errors distribution, higher order calibration laws don't seem to be necessary.

6.4.2. Second calibration session

The second calibration session was conducted after flight activities. The procedure was carried out exactly in the same way as in the first session.

AHRS (averaged) attitude estimation as a function of sample number is reported in figure 15. It can be appreciated that maximum variation is of the order of 0.1°.

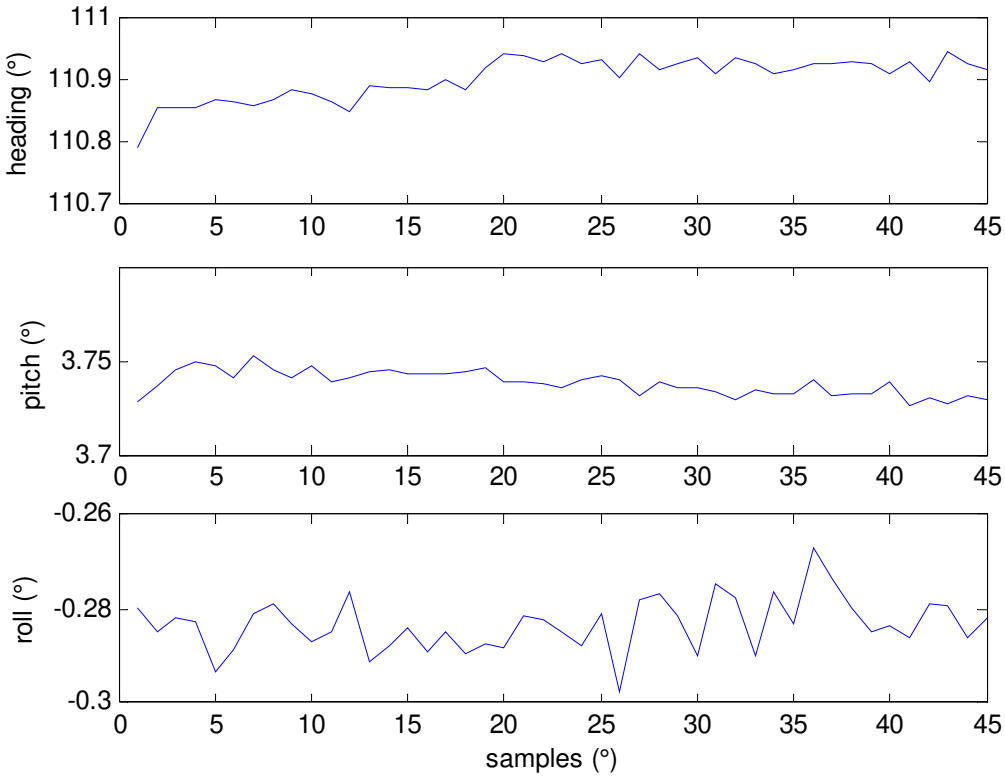

Figure 15 – Estimated aircraft attitude angles as a function of sample number

In this case, the Euler angles of cameras with respect to body reference frame as given by AHRS are reported in table 5 :

Estimated Angle (°)	Panchromatic	Color	IR
Roll	8.752124	4.717738	0.031195
Pitch	1.146022	0.780936	-0.109817
Yaw	-1.220175	3.982433	-1.723170

Table 5 – Estimated cameras attitude angles (2nd session)

Next figures report targets positions in panchromatic (figure 16), color (figure 17) and IR (figure 18) images for the second acquisition session as measured by GPS/AHRS and converted through the calculated rotation matrix (in red) and as obtained by image analysis (in blue):

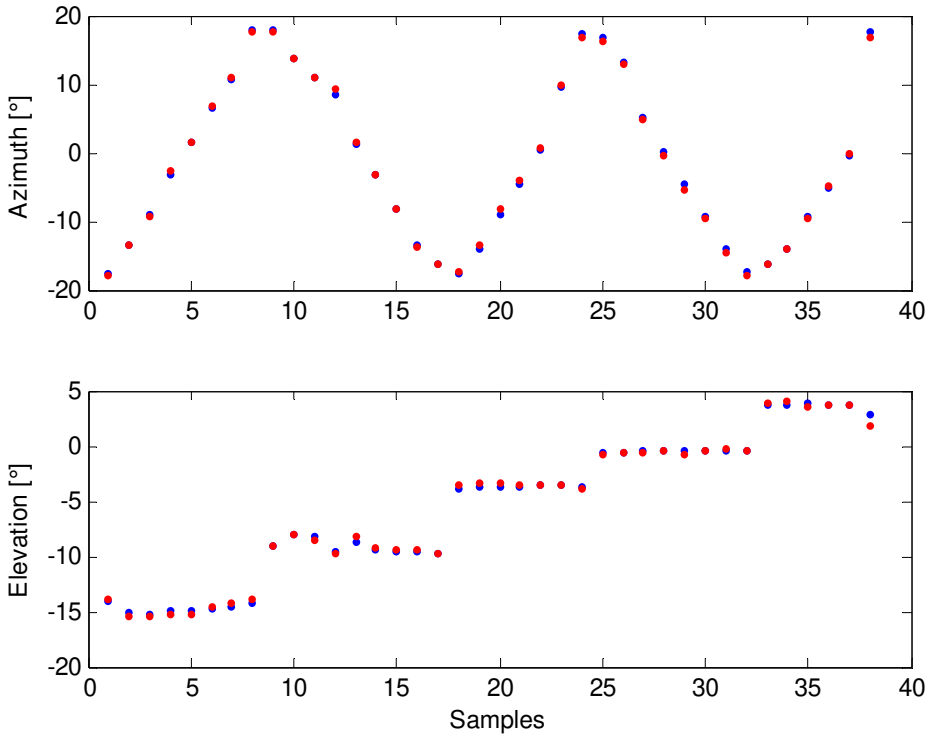

Figure 16 - Target positions as extracted from images (blue) and from GPS/AHRS measurements and computed rotation matrix (red) : panchromatic camera, 2nd calibration session

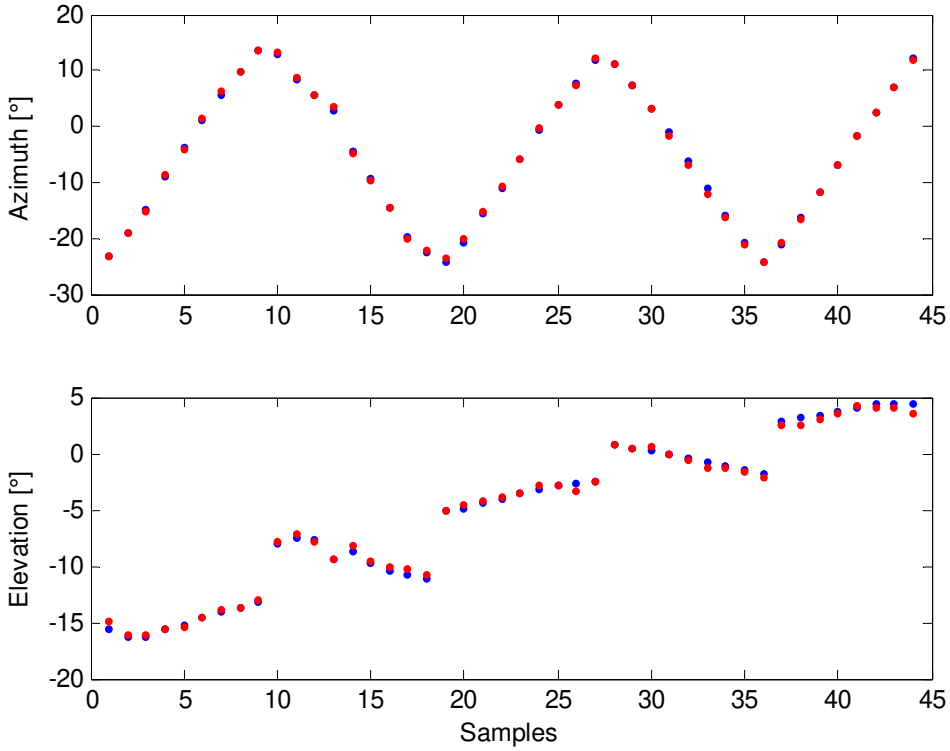

Figure 17 - Target positions as extracted from images (blue) and from GPS/AHRS measurements and computed rotation matrix (red) : color camera, 2nd calibration session

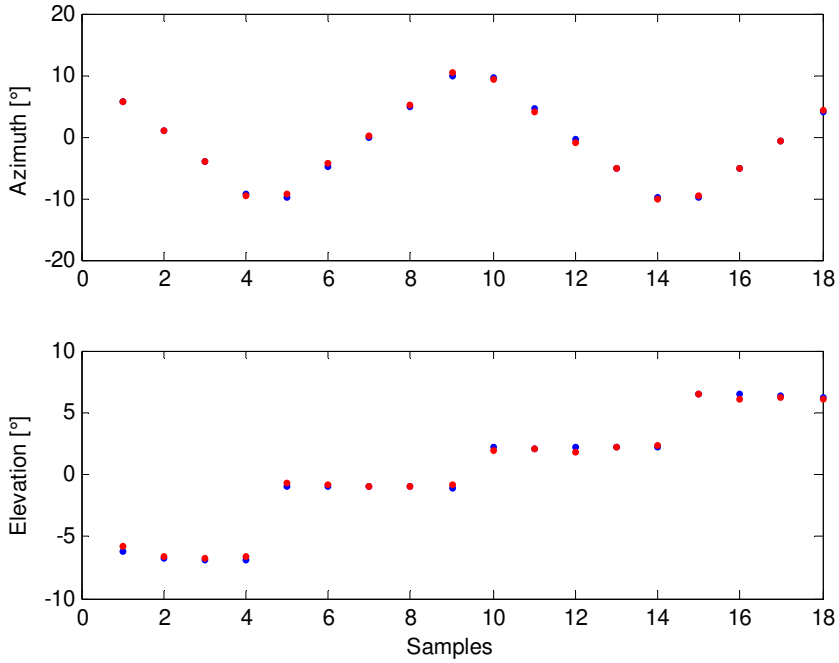

Figure 18 - Target positions as extracted from images (blue) and from GPS/AHRS measurements and computed rotation matrix (red) : IR camera, 2nd calibration session

Residual angular errors on test points are reported in the following, considering for the sake of brevity only the color camera (figure 19). Also in this case no systematic effects appear on the azimuth-elevation plane (figure 20) and the normality of the residuals can be simply verified.

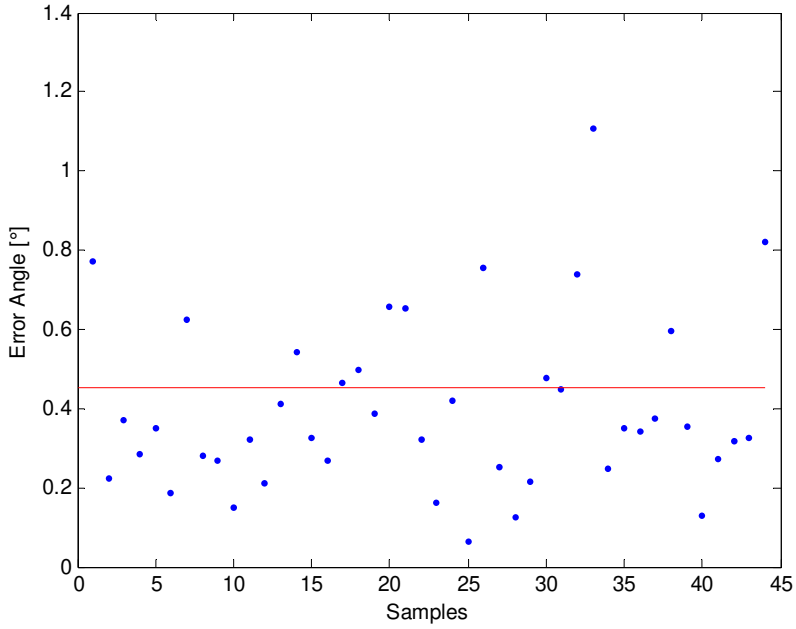

Figure 19 - Residual angular errors for color camera (2nd session)

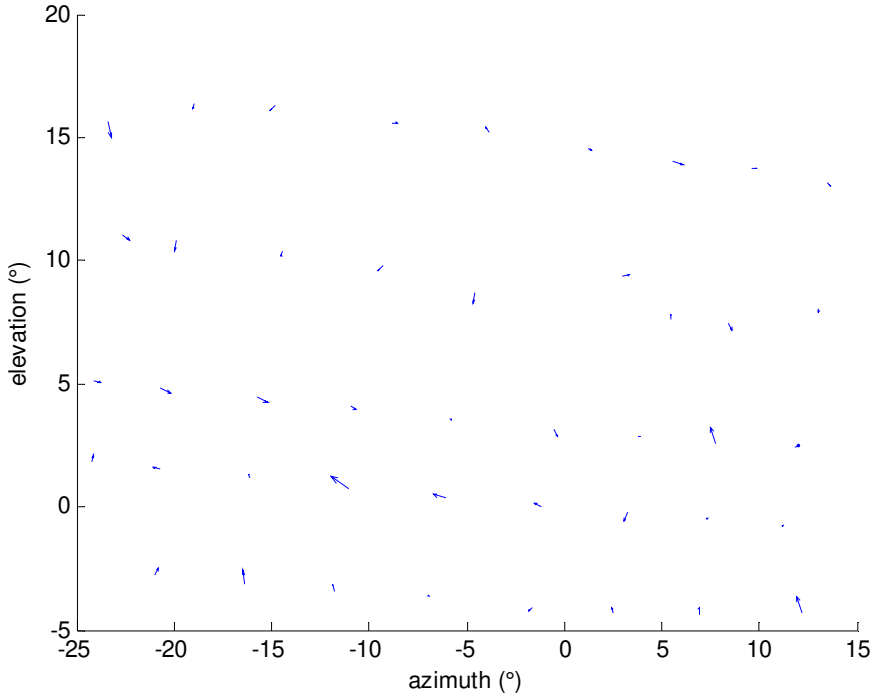

Figure 20 - Vector representation of azimuth and elevation differences (color camera, 2nd session)

Synthetic results for the two calibration sessions are reported in table 6.

Calibration session	Camera	RMS Error
1st	Panchromatic	0,34°
1st	Color	0,38°
1st	Infrared	0,27°
2nd	Panchromatic	0,48°
2nd	Color	0,45°
2nd	Infrared	0,42°

Table 6 – RMS error for the different cameras in the two calibration sessions

The second calibration session showed a slightly large error which is believed to be due to weather conditions (wind which produced system and target oscillation).

Future alignment sessions will be performed before of the sense and avoid flights. It is believed that alignment can be performed with better accuracy with respect to the data reported in this chapter, because a number of improvements will be applied, such as better contrast of the target with respect to the background, and strict control on GPS accuracy both for the target and the cameras.

The angular displacements between cameras axes considering 1st and 2nd session are reported in table 7. It can be seen that the rotations fall in the measurements error range, thus flights impact on calibration can be considered as negligible.

Camera	X Axis [°]	Y Axis [°]	Z Axis [°]
Color	0.36	0.36	0.07
Panchromatic	0.66	0.69	0.46
Infrared	0.41	0.43	0.14

Table 7 - Angular displacements of cameras' axes between the two calibration sessions

Chapter 7

Preliminary flight data and collision avoidance tests planning

7.1. Introduction

Due to project delays non dependant on the anti-collision system, sense and avoid flight tests have not been performed yet. However, flight activities have been carried out, in parallel with other functions developed in TECVOL project, in order to provide functional verification of EO sensors. In particular, the interest was in evaluating the impact on image quality of vibrations and in general of all the disturbances coming from the aircraft. For the infrared cameras, there was a particular interest in evaluating the thermal influence of aircraft engine, and the image quality in day hours.

Visible and infrared cameras flight images, and the relevant considerations in view of future sense and avoid tests, are reported in the following sections.

Furthermore, a schedule for sense and avoid flights has been defined and it is briefly described in the last section.

7.2. Flight tests: visible cameras

Tests for functional verification of EO sensors were performed with a different hardware configuration with respect to the architecture described in chapter 2. In fact, the software developed for cameras acquisition was installed on a laptop which was connected to the cameras, with no interfaces with the on-board flight control system. FLARE trajectory during one of the flight tests is reported on the map in figure 1. Images acquired by the visible cameras in the three positions signaled by the circles are shown in the following. In particular, point 1 is in correspondence of a left turn; color and panchromatic images are reported in figure 2 and figure 3, respectively. Propeller appears as a still object in figure 3, due to very low shutter time. In the same conditions, it "spreads" a little when it appears in color images due to Bayes filtering and consequent longer shutter time required for a given level of absorbed electromagnetic energy. It is worth noting that quality in panchromatic images was affected in these tests by non correct installation, regarding in particular non precise assembly of optics. This is the reason why panchromatic images show a little defocusing effect.

For the sake of brevity, only color images are shown for the other reference points on figure 1. Figure 4 shown an image acquired during a right turn (point 2), while figure 5 reports the runway as seen in point 3 with the sun in the back. It is worth noting that the grass runway is undetectable from the rest of the image, on the basis of the intensity information, in color (and panchromatic) images, even with a favorable geometry with respect to sun.

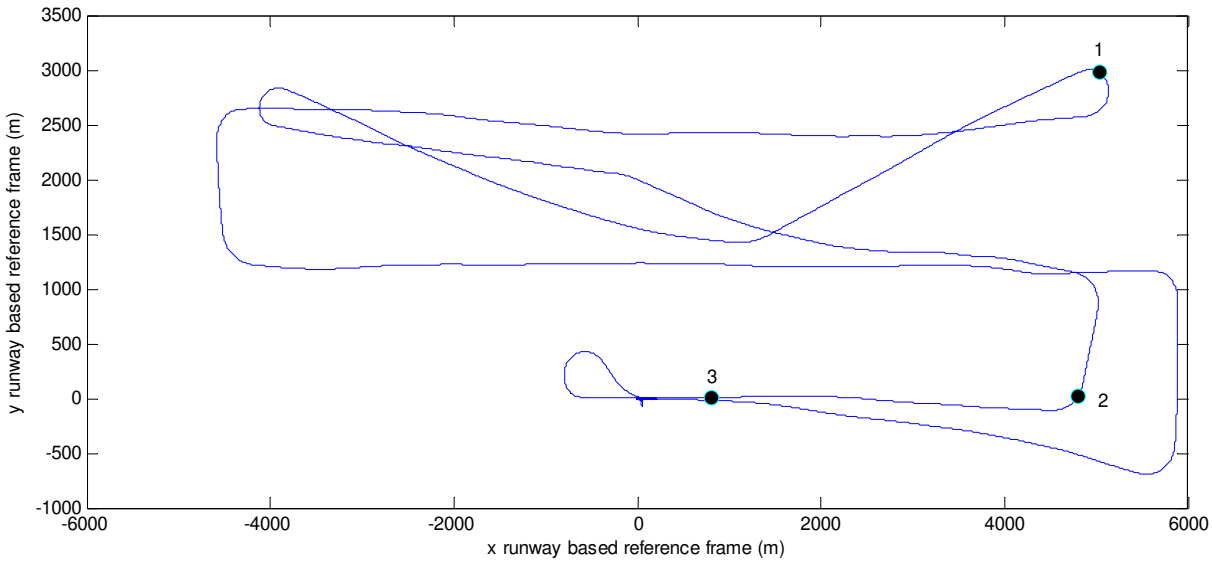

Figure 1 – FLARE trajectory during one of the performed flight tests.

Figure 2 – Image taken by the color camera during a left turn (point 1 in figure 1)

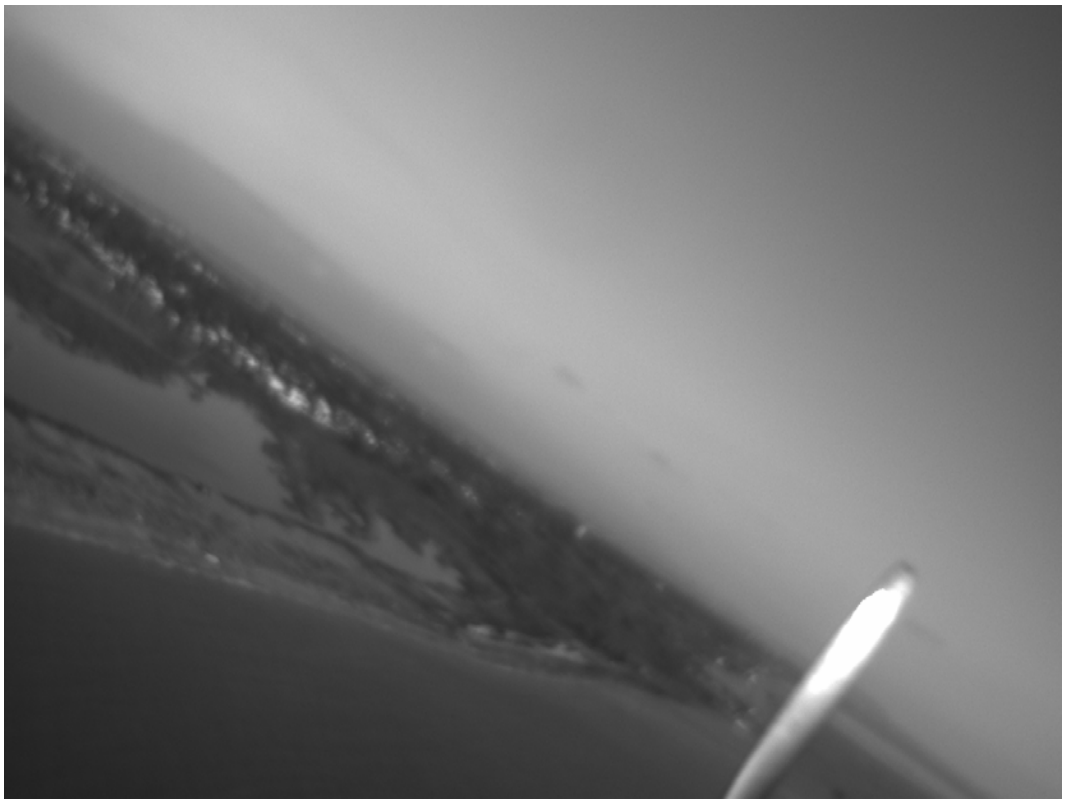

Figure 3 - Image taken by the panchromatic camera during a left turn (point 1 in figure 1)

Figure 4 – Image taken by the color camera during a right turn (point 2 in figure 1)

Figure 5 – Runway as seen by the color camera (point 3 in figure 1)

All the previous images show the high quality of images, with negligible effects of vibrations. They all refer to geometries with the sun in back or in lateral position. An example of images with the sun in front in reported in figure 6. It can be appreciated that only about 30% of the images is completely saturated and thus is useless for obstacle detection. This is the minimum possible saturation with the sun in the FOV since the image has been taken with the minimum aperture. Detection range as a function of visible sensors orientation with respect to sun will be evaluated in flight tests with an intruder aircraft.

Figure 6 – Effect on image of sun presence in FOV

7.3. Flight tests: infrared cameras

Flight tests with infrared cameras were devoted to estimating IR image quality in daytime conditions in different orientations with respect to sun, and eventual thermal disturbance by the aircraft engine. All the gathered images showed high quality and contrast in daytime hours and in any orientation with respect to sun. Some images are reported in the following. In all the images, the small rectangle on the left part of the figure represents mean image intensity. Figure 7 is a low altitude image that allows to point out the detail and radiometric resolution of the sensor: differences in ground temperature due to shadows are detectable. Figure 8 is

taken with the sun in front (though not in the FOV which is narrow if compared to visible cameras): contrast is not affected by sun and hot objects (for example, the road) are well detectable with respect to the rest of the image. Figure 9 confirms high sensitivity of the sensor, and a river (at lower temperature than land) is clearly detectable.

Figure 7 – Low altitude image taken by the Infrared camera

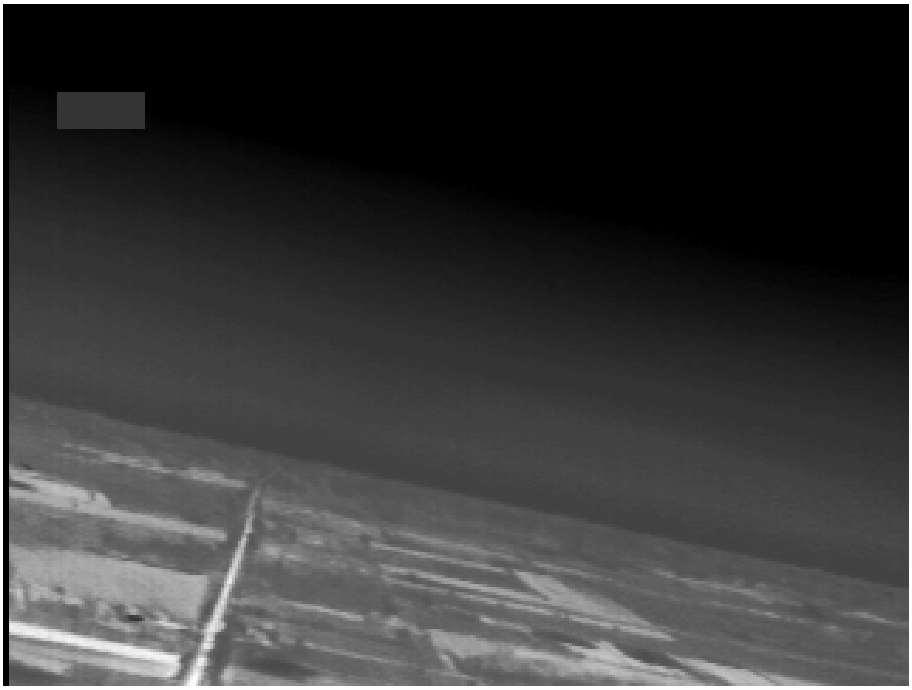

Figure 8 – Image taken by the Infrared camera with the sun in front

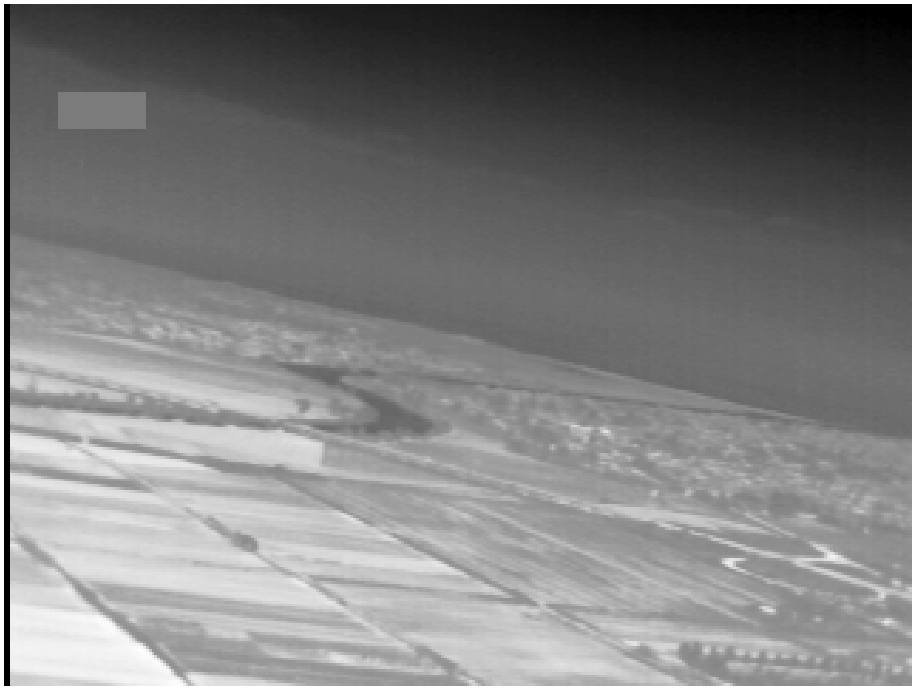

Figure 9 – High contrast image taken by the infrared camera

Figure 10 (again, taken with the sun in front) shows that in infrared images the grass runway is detectable in the image. Because of this result, it is now foreseen to use IR cameras in TECVOL also for the runway detection function.

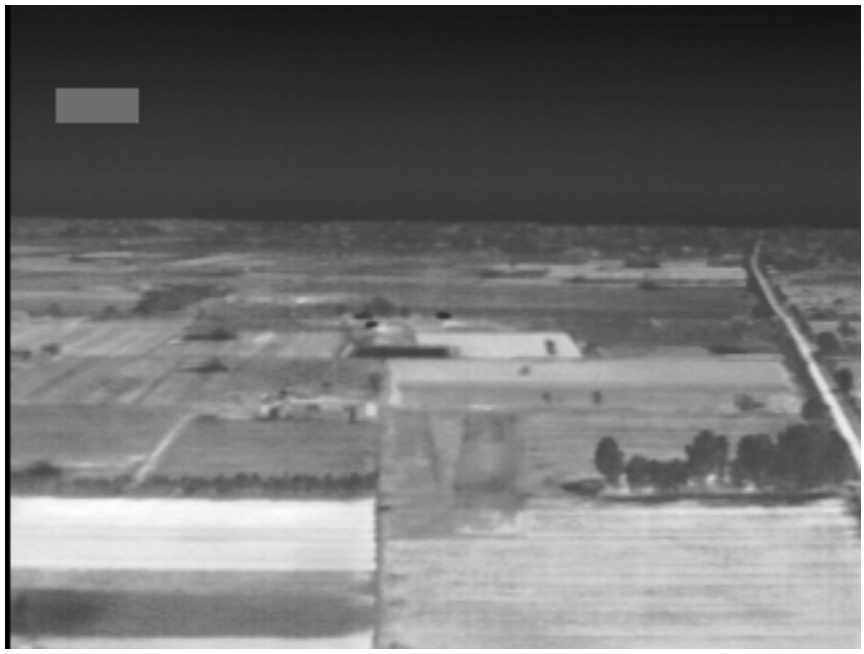

Figure 10 – Runway as seen by the infrared camera

Figure 11 shows the propeller effect in IR images. Because of the shutter time required by the sensor, the propeller spreads in the image. Thus, the main effect for obstacle detection is a reduction of signal-to-noise ratio in a part of the image. Nevertheless, it must be considered that in preliminary flights the IR cameras were mounted forward-looking without eccentricity with respect to aircraft longitudinal axis. Applying the required divergent angle, propeller impacts only a very small part of IR sensors FOV.

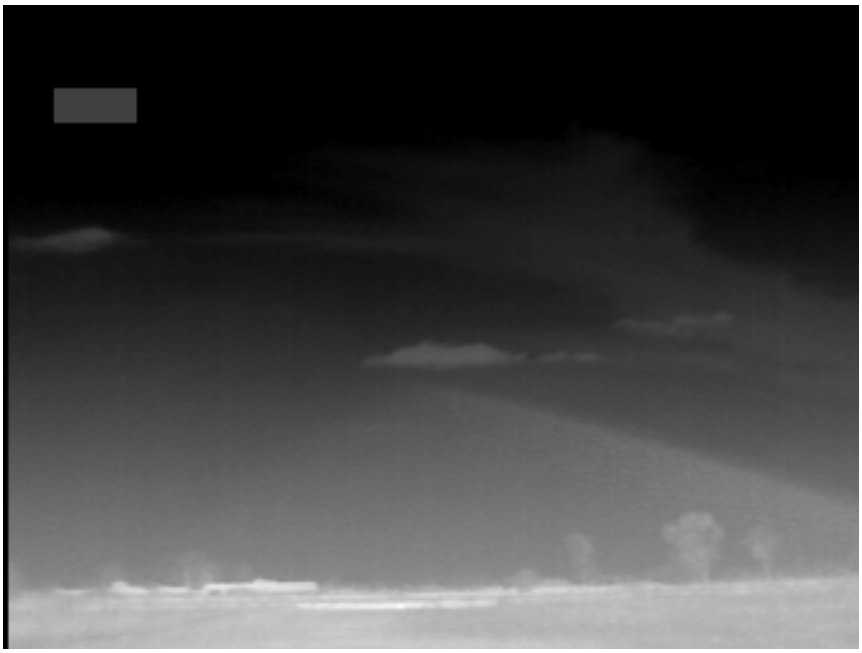

Figure 11 – Image taken at take-off showing propeller effect

The following two figures are very important in view of the collision avoidance application. In fact, as already said detection range for EO sensors could not be evaluated experimentally because of absence of intruders. However, some images of other small aircrafts have been taken by the IR cameras when FLARE was on ground and image recording had already been activated. Figure 12 shows a TECNAM P-2002 during take-off. The engine area is very bright compared to the rest of the aircraft. It is believed that engine temperature will be very important for intruder detection in near collision geometries. The same result is confirmed in figure 13, where a P-92 is imaged during landing. The engine temperature makes it detectable even on a cluttered background. Detection of an intruder flying over the horizon is a more favorable situation since, as observed in all the images, the sky appears as a dark background in any geometry with respect to sun. In summary, preliminary flights gave high confidence on

the possibility to use EO sensors information for obstacle detection and tracking in the designed radar-driven anti-collision system.

Figure 12 – A TECNAM P-2002 during take-off in an image taken by the IR camera

Figure 13 - A TECNAM P-92 during landing in an image taken by the IR camera

7.4. Sense and Avoid flight tests

Sense and avoid flight tests will be performed in the next few months with the following logic. First of all, flights for radar acceptance will be performed. These tests will require that the entire architecture described in this thesis is properly installed on board FLARE, except for the EO sensors. The radar will be remotely commanded from the ground station, by means of the radio link, the CAN bus and the real time computer. A proper ground/on board software has been developed for these flights. Different tests will allow to validate the different operating modes for the radar, ant to evaluate detection range for a VLA intruder.

As for sense and avoid flight tests, they are divided in two main categories: obstacle detection and tracking flight performance assessment, combined autonomous collision avoidance.

In the first category automatic control will not be activated. In particular, in the first tests the flying laboratory will follow an intruder aircraft at higher altitude to keep it in its field of regard for a long time without safety problems (Fig. 14-1). This will enable to verify tracking capabilities with different sets of sensors activated. Intruder at higher altitude will reduce problems due to land clutter. Then, the flying laboratory will fly several "near collision" "quasi frontal" trajectories with an intruder again at higher altitude (Fig. 14-2). Finally, the same tests will be conducted with the intruder at lower altitude, which is the hardest situation for the necessity to remove land clutter (Fig. 14-3).

Tests will be repeated for several weather and illumination conditions, and flying altitude will be chosen on the basis of the considerations reported in chapter 5 and of experimental radar data acquired in acceptance flights.

In a subsequent phase, collision avoidance tests will be performed. Preliminary collision avoidance tests will deal with simulated fixed obstacles to verify correct engagement of autonomous collision avoidance system. Thus, the sensing system will be bypassed.

Then, real collision avoidance tests will take place. In these tests, the flying laboratory, with all systems on, will fly several near collision trajectories with a single intruder in its field of regard. On the basis of detection and tracking system estimates, flight control computer will generate and follow in real time a proper escape trajectory in case of predicted collision. A flight plan developed for these tests is shown in figure 15 that depicts the area selected for flights.

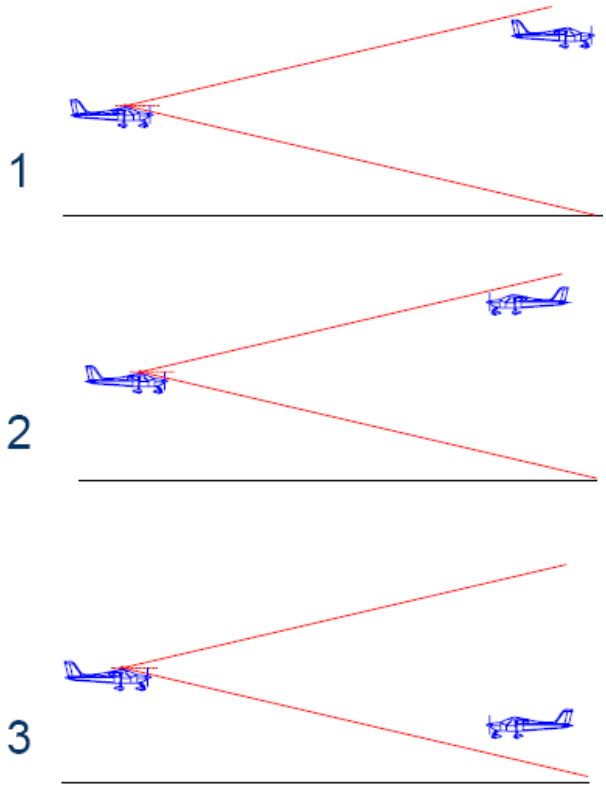

Figure 14 - Obstacle detection and tracking flight tests

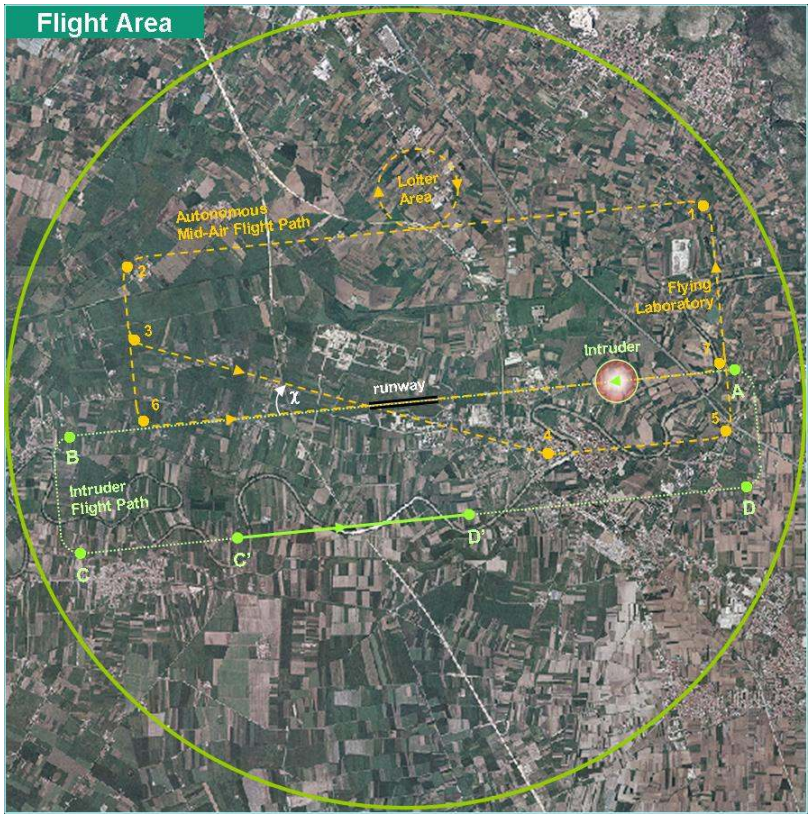

Figure 15 - Flight area for Collision Avoidance tests

Conclusions and further research

This thesis was devoted to the development of an integrated multi-sensor based non cooperative anti-collision system for UAVs. The performed activities covered all the steps in the development process from the analysis of requirements deriving from the application, to the real time implementation of designed logics.

Advantages and drawbacks of different sensors were analyzed. The all-time all-weather requirement, together with theoretical and experimental analyses of limitations of a purely electro-optical approach to the collision avoidance problem, led to a multiple sensor architecture where a Ka-band pulsed radar acts as the main sensor. Four electro-optical cameras (two in the visible band, two in the infrared band) were selected as aiding sensors.

A logical/hardware architecture was designed in order to perform real time sensor fusion and object tracking within the sensors field of view. It is based on the use of two processing units and on communication via CAN bus with the flight control computer.

Models and algorithms for airborne multiple sensor tracking were analyzed in detail by means of numerical simulation of collision scenarios dynamics, and of navigation and obstacle detection sensors output.

A tracking algorithm based on Extended Kalman filtering in Cartesian North-East-Down coordinates, and Singer model for acceleration components, was estimated to be a good compromise between tracking accuracy and reliability at very short distances, which is considered to be the worst case condition for the anti-collision system.

Radar/electro-optical fusion was demonstrated to hold a dramatic improvement in tracking accuracy, besides augmenting system reliability thanks to the fact that detection for the different sensors is based on completely different physical phenomena.

Extensive off-line numerical simulations were carried out considering realistic collision scenarios and the execution of different avoidance maneuvers. It was demonstrated that tracking performance is in substance unaffected by the choice of collision avoidance logic. A residual rms error of about 10.5 m in range and 1.1° in (body-referenced) azimuth and elevation resulted for intruders detected only by the radar. Instead, rms errors for range and angles when performing radar/EO fusion were estimated to be of the order of 6.1 m and 0.11°, respectively. Navigation system effects on tracking accuracy were pointed out. In particular, biases in attitude angles errors are filtered in body referenced angular estimates and their effect can be seen only on stabilized (that is, NED) angular estimates. Attitude angles errors' noise influences also body estimates.

All the developed logics and methods were implemented on the on board processing units for flight technology demonstration. The two computers are based on a deterministic (Microsoft Windows CE) and a conventional (Microsoft Windows XP Embedded) operating system, and were programmed in different languages (Embedded C and Visual Basic.Net). Both the two developed software packages are based on modules that perform the different communication and processing functions. Communication with the radar system for data acquisition and sensor command is based on TCP/IP and a customized interface protocol that allows to choose among different operating modes. Extensive hardware-in-the-loop tests were performed to estimate latency in data exchange and accuracy of real time performance. In particular, latency in the communication via CAN bus between the obstacle detection real time unit and the flight control computer was estimated to be of the order of a few ms. It is largely compatible with application requirements.

A procedure for electro-optical sensors alignment with respect to the navigation unit was designed and tested. It is based on use of carrier-phase differential GPS in Real Time Kinematic mode, accuracy of the AHRS in static mode, and analysis of sample images with target pixel extraction. First calibration sessions demonstrated an alignment accuracy of the

order of 0.3° rms, while the effect of flights on alignment results was found to be of the same order of measurement accuracy. It is believed that calibration accuracy can be improved in future tests thanks to higher target contrast with respect to the background, and to more stable GPS estimates of targets and cameras' positions.

Preliminary flights were performed for electro-optical sensors functional verification. Acquired images show that disturbances induced by the platform are negligible, in terms of vibrations, propeller impact, and thermal noise for the infrared sensors. Visible images taken with the sun in the back or in lateral position show high detail and contrast. Instead, the effect of sun presence in the field of view makes completely useless for obstacle detection about 30% of the image. Infrared sensors performance in any geometry with respect to sun are largely satisfying in terms of contrast and radiometric resolution. Thus, the flight data confirmed the good potential of electro-optical sensors for target detection, especially for intruders flying over the horizon.

Due to project delays, sense and avoid flight tests have not been performed yet. These tests can be considered both as the appropriate conclusion of all the activities performed in this study, and as a starting point for further research and development. Flight tests with real time integration of microwave and electro-optical sensors, and with contemporary object detection and tracking on the one hand, and collision avoidance maneuvers execution on the other hand, within a fully autonomous architecture, have never been performed until today. Thus, the scientific value of this technology demonstration would be significant.

Moreover, flight tests will be very important in view of future evolution of sense and avoid studies and systems. Experimental evaluation of radar performance will allow to understand the critical improvements needed to design a new generation microwave sensor especially suited for sense and avoid. Demonstration of collision avoidance capability with a single intruder will be the starting point for collision avoidance with more than one intruder. It is worth noting that from tracking point of view, considering multiple intruders will have an impact more in terms of computational cost than system approach.

The availability of multiple sensor flight data will be the key for other future developments and researches. In fact, first of all they will be exploited for design and implementation of intruder classification algorithms. Moreover, electro-optical flight images will be used for further analysis and development of image based techniques for detection and tracking. In this framework, real time integration of images and navigation data seems to be the roadmap for a significant performance improvement.

References

[1] Walker, J., “Safe Integration of Unmanned Aircraft Systems – UAVs into Civil Managed Airspace,” *Air Traffic Control Association Congress*, Forth Worth TX, USA, November 2nd 2005.

[2] Okrent, M., “Civil Uav Activity within The Framework of European Commission Research,” *Proc. of AIAA 3rd Unmanned Unlimited Technical Conference*, AIAA 2004-6329, Chicago, IL, 2004, pp. 1-12.

[3] “Airspace Integration Plan for Unmanned Aviation”, Office of the Secretary of Defense, Washington, DC, 2004.

[4] Wegerbauer, C., “The Access 5 Project - Enabling routine UAS operations in the U.S. National Airspace System,” *Unmanned Vehicles Magazine*, May/June 2005.

[5] Walker, J., and Kenagy, R., *UAV-Related Standards* - The RTCA Special Committee 203, 2004.

[6] ASTM Designation: F 2411 – 04, *Standard Specification for Design and Performance of an Airborne Sense-and-Avoid System*, ASTM International, West Conshohocken, PA, 2004;

[7] FAA, FAA Order 7610.4, available from www.faa.gov, 2006

[8] FAA, "*Introduction to TCAS Version 7*", US Department of Transportation, Federal Aviation Administration, 2000.

[9] RTCA, "Minimum Aviation System Performance Standards For Automatic Dependent Surveillance Broadcast (ADS-B)", *RTCA/DO-242A*, Washington DC, 2002

[10] Portilla E., Fung A., Chen W.Z., Shakernia O., and Molnar T., Sense And Avoid (SAA) & Traffic Alert and Collision Avoidance System (TCAS) Integration for Unmanned Aerial Systems (UAS), *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[11] Shakernia, O., Chen, W.Z., Graham, S., Zvanya, J., White, A., Weingarten, N., Raska, V.M. "Sense and Avoid (SAA) Flight Test and Lessons Learned," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[12] Gibbs, D.G, "Sense and Avoid Flight Demonstration," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[13] Wolfe, R. C., *NASA ERAST Non-Cooperative Detect, See, & Avoid (DSA) Sensor Study*, Modern Technology Solutions Incorporated, Alexandria, VA, 2002

[14] Wolfe, R.C., *NASA ERAST Non-Cooperative DSA Flight Test*, Modern Technology Solutions Inc., Alexandria VA, USA, 2003

[15] Utt, J., McCalmont, J., and Deschenes, M., "Test and Integration of a Detect and Avoid System," *Proc. of AIAA 3rd Unmanned Unlimited Technical Conference*, AIAA 2004-6329, Chicago, IL, 2004, pp. 1-10

[16] Maroney, D.R., Bolling, R.H., Athale, R., and Christiansen, A.D., "Experimentally Scoping the Range of UAS Sense and Avoid Capability," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[17] Carnie R., Walker R. and Corke P., "Image Processing Algorithms for UAV 'Sense and Avoid' ", *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida, May 2006

[18] Utt, J., McCalmont, J., and Deschenes, M., Development of a Sense and Avoid System, *1st AIAA Infotech@Aerospace Conference*, Washington DC, USA, September 2005

[19] Rees F.R., "Collision Avoidance System for Use in Aircraft", *Patent no. US RE39,053 E*, Flight Safety Technologies In.c, Mystic CT, USA, 10 February 2003.

[20] Shakernia O., Chen W.Z., and Raska V.M., "Passive Ranging for UAV Sense and Avoid Applications", *1st AIAA Infotech@Aerospace Conference*, Washington DC, USA, September 2005

[21] Doehler, H-U, Löving, K., Oyzel, A., Roviaro, E., See & Avoid Technologies,– Unmanned Aerial Vehicle Safety Issues for Civil Applications, *project USICO - final report,* 2004

[22] Lie, W.N., Automatic Target Segmentation by Locally Adaptive Image Thresholding, *IEEE Transactions on Image Processing*, Vol. 4(7), pp:1036-1041, IEEE, 1995

[23] Rydergård, S., "Obstacle Detection in a See-and-Avoid System for Unmanned Aerial Vehicle", *Master's Thesis in Computer Science*, Royal Institute of Technology Stockholm, Sweden, 2004

[24] Zhang, X-P, Desai, M.D., "Segmentation of Bright Targets Using Wavelets and Adaptive Thresholding", *IEEE Transactions on Image Processing*, Vol. 10(7), pp:1020-1030, IEEE, 2001

[25] Girosi F., Verri A., and Torre V., "Constraints for the Computation of Optical Flow", *IEEE Workshop on Visual Motion*, Proceedings, pp:116-124, IEEE, 1989.

[26] Beauchemin S.S. and Barron J.L., "The Computation of Optical Flow", *ACM Computing Surveys*, Vol. 27, No. 3, pp: 433-461, 1995.

[27] Barron J.L., Fleet D.J. and Beauchemin S.S., "Performance of Optical Flow Techniques", *International Journal of Computer Vision*, Vol. 12(1), pp. 43-77, 1993.

[28] Horn B.K.P and Schunck B.G., "Determining Optical Flow", *Artificial Intelligence*, Vol. 17, pp. 185-203, 1981.

[29] La Franchi P., "See the future"-Flight International, September 2005

[30] Recchia, G., Fasano, G., Accardo, D., Moccia, A., and Paparone, L., "An Optical Flow Based Electro-Optical See-and-Avoid System for UAVs", *Proceedings of IEEE 2007 Aerospace Conference*, Big Sky, MT, USA, March 2007

[31] Klein, L.A., *Sensor and Data Fusion – A Tool for Information Assessment and Decision Making*, Spie Press, Bellingham WA, USA, 2004

[32] Marshall, D.M., Trapnell, B.M., Mendez, J.E., Berseth, B.L, Schultz, R.R., and Semke, W.H., "Regulatory and Technology Survey of Sense-and-Avoid for UAS," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[33] Coulter, D.M., and Wolfe, R.C., "Encounter Analysis to Support Detect, Sense, and Avoid (DSA) Elevation Field of Regard (FOR) Requirements," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[34] Szu1, H.H., and Krapels, K., "Autonomous Collision Avoidance of flying Vehicles," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[35] Joulia, A., and Le Tallec, J.C., "UAVs: What and how to sense? To avoid what? ," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007;

[36] Gibbs, D.G, "Sense and Avoid Flight Demonstration," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[37] Zeitlin, A.D, "Technology Milestones – Detect, Sense &Avoid for Unmanned Aircraft Systems," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[38] Duke E.L., Vanderpool C.C., and Duke W.C. "Turning PINOCCHIO Into A Real Boy: Satisfying A Turing Test For UA Operating In The NAS", *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007

[39] Rosemberg B., "Avoidance Systems Aim to Make UAVs Safe", DefenseNews.com, http:// www.defensenews.com/story.php?F=3081163&C=airwar

[40] Ye M., "Robust Visual Motion Analysis : Piecewise-smooth Optical Flow and Motion-Based Detection and Tracking", *Doctoral Degree's Thesis in Electrical Engineering* - University of Washington, USA, 2002

[41] Gandhi T.L., Yang M.T., Kasturi R., Camps O., Coraor L., McCandless J., "Detection of Obstacles in the Flight Path of an Aircraft", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 39(1), pp. 176-191, 2003.

[42] Accardo, D., Moccia, A., Cimmino, G., and Paparone, L., "Performance Analysis and Design of an Obstacle Detection and Identification System," *Proc. of AIAA 1st Infotech@Aerospace Technical Conference*, Arlington VA, USA, 26-29 September 2005

[43] Holst, G.C., *Electro-optical imaging system performance*, SPIE Press & JCD Publishing, Bellingham WA, USA, 2000.

[44] Bar-Shalom Y., Kirubarajan T., *Tutorial course on Multitarget Tracking and Multisensor Fusion*, IEEE International Conference on Information Fusion, Florence, July 2006.

[45] Chen H., Kirubarajan T., Bar-Shalom Y., "Performance limits of track-to-track fusion versus centralized estimation: theory and application", *IEEE Transactions on Aerospace and Electronic Systems*, Volume: 39 , Issue: 2 pages: 386 – 400, April 2003

[46] Blackman, S.S., and Popoli, R.F., *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999

[47] Bosch GMBH document: http://www.semiconductors.bosch.de /pdf/can2spec.pdf

[48] Prasad S., . "Experiences of CAN in Space" *1st CAN in Space Workshop (Tutorial Day)*.5-6 December 2002 ESTEC, Noordwijk, The Netherlands

[49] "Controller Area Network (CAN) Overview" http://zone.ni.com/devz one/cda/tut/p/id/2732#toc2

[50] http://www-net.cs.umass.edu/kurose/transport/UDP.html

[51] Munkres J., "An Algorithm for the Assignment and Transportation Problems", *SIAM Journal*, No. 5, 1957, pp. 32-38.

[52] Jonker, R. and Volgenant, A., "A Shortest Augmenting Path for Dense and Sparse Linear Assignment Problems", *Journal of Computing*, vol. 38, pp. 325-340, 1987.

[53] Castanon D.A., "New assignment algorithms for Data Association", *SPIE*, Vol. 1698, pp. 313-323.

[54] Malkoff, Donald B., "Evaluation of the Jonker-Volgenant-Castanon (JVC) assignment algorithm for track association", *SPIE* Vol. 3068, 1997.

[55] Blackman S.S., "Multiple hypothesis tracking for multiple target tracking", *IEEE Aerospace and Electronic Systems Magazine*, Volume: 19 , Issue: 1 , Part 2, pp-5-18, Jan 2004

[56] Blackman, S.S., *Multiple-Target Tracking with Radar Applications*, Artech House, Dedham MA, USA, 1986.

[57] Brookner, E., *Tracking and Kalman Filtering Made Easy*, John Wiley and Sons, New York NY, USA, 1998

[58] Gelb A., *Applied Optimal Estimation*, MIT Press, Massachusetts, 1974

[59] Grewal M.S., Andrews A.P., *Kalman Filtering: Theory and Practice Using Matlab*, Second Edition, John Wiley and Sons, 2001.

[60] Haykin S., *Kalman Filtering and Neural Networks*, John Wiley and Sons, 2001.

[61] Kalman, R. E., ''A New Approach to Linear Filtering and Prediction Problems", *Journal of Basic Engineering*, *Trans. of ASME*, Ser. D, Vol. 82, No. 1, March 1960, pp. 35–45.

[62] Kalman, R. E. and R. S. Bucy, ''New Results in Linear Filtering and Prediction Theory'', *Transactions of the ASME, Journal of Basic Engineering*, Vol. 83, No. 3, December 1961, pp. 95–107.

[63] Li, X.R., Jilkov, V.P., A Survey of Maneuvering Target Tracking: Dynamic Models, *SPIE Conference on Signal and Data Processing of Small Targets*, Orlando, FL, USA, April 2000.

[64] Singer, R.A., Sea, R.G., and Housewright, K.B., "Derivation and Evaluation of Improved Tracking Filter for Use in Dense Multitarget Environments", *IEEE Transactions on Information Theory*, vol. 20, no. 4, IEEE, 1974

[65] Dana, M.P., *Introduction to Multi-sensor Multi-target Data Fusion Techniques for Detection, Identification and Tracking,* OEI Course, April 2005

[66] Bar Shalom, Y., and Li, X.-R., *Estimation and Tracking: Principles, Techniques and Software*, Artech House, 1993.

[67] Zollo, S., Ristic, B., *On the Choice of the Coordinate System and Tracking Filter for the Track-while-scan Mode of an Airborne Pulse Doppler Radar*, DSTO-TR-0926, December 1999.

[68] Lerro, D., Bar-Shalom, Y., "Tracking with Debiased Consistent Converted Measurements Versus EKF", *IEEE Trans. On Aerospace and Electronic Systems*, vol. 29, no. 3, July 1993

[69] Mo., L., et alii, "Unbiased Converted Measurements for Tracking", *IEEE Trans. On Aerospace and Electronic Systems*, vol. 34, no. 3, July 1998

[70] Duan, Z., et alii, Comments on "Unbiased Converted Measurements for Tracking", *IEEE Trans. On Aerospace and Electronic Systems*, vol. 40, no. 4, October 2004

[71] Suchomski, P., "Explicit expressions for Debiased Statistics of 3D Converted Measurements", *IEEE Trans. On Aerospace and Electronic Systems*, vol. 35, no. 3, January 1999

[72] Fasano, G., Accardo, D., Moccia, A., and Paparone, L., "Airborne multisensor tracking for Autonomous Collision Avoidance" *Proc. IEEE Fusion 2006 Congress*, Florence, ITA, paper no. 311, July 10-14 2006

[73] Swerling, P., *Probability of detection for fluctuating targets*, RM-1217, 1954, also IEEE Transactions on Information Theory, April 1960

[74] Skolnik M.I., ed. by, *Radar Handbook*, McGraw Hil Professional, New York NY, USA, 1990.

[75] Mahafza B.R., *Radar Systems Analysis and Design Using MATLAB*, Chapman & Hall/CRC, Boca Raton FL, USA, 2000.

[76] Fasano, G., Accardo, D., Moccia, A., Carbone, C., Luongo, S., Ciniglio, U., and Corraro, F., "Multisensor based Fully Autonomous Non-Cooperative Collision Avoidance System for UAVs," *2nd AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, USA, May 2007. Selected for publication on the *AIAA Journal on Aerospace Computing, Information, and Communication* and currently under final review.

[77] Grewal M. S., Weill L. R., and Andrews A. P., *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley Publishing, Copyright, 2001

[78] http://www.xbow.com/Products/Product_pdf_files/Inertial_pdf/AHRS400CD_Datasheet.pdf

[79] http://www.geotop.it/gps_GNSS_topo_geo_legacy_H.htm

[80] Lank, G.W., Radar measurement accuracy for fluctuating targets, *IEEE Transactions on Aerospace and Electronic Systems*, November 1991

[81] Fasano, G., Moccia, A., Accardo, D., Paparone, L., "Sensors and Algorithms for UAV Non Cooperative Collision Avoidance", *XIX AIDAA National Congress*, Forlì, September 2007.

[82] http://msdn2.microsoft.com/en-us/library/ms905511.aspx

[83] http://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf

[84] http://www.tenouk.com/cnwinsock2tutorials.html

[85] http://msdn2.microsoft.com/it-it/embedded/aa731409.aspx

[86] http://msdn2.microsoft.com/en-us/library/ms940856.aspx

[87] Esposito F., Rufino G., Moccia A., Donnarumma P. Esposito M., Magliulo V., "An Integrated Electro-Optical Payload System for Forest Fires Monitoring from Airborne Platform", *2007 IEEE Aerospace Conference*, 3-10 March 2007

[88] Frahm J.M., Koser K., and Koch R., "Pose Estimation for Multi-Camera Systems", *Proceedings of DAGM 2004*, Tuebingen, Germany, August 2004

[89] http://www.vision.caltech.edu/bouguetj/calib_doc/

[90] Ford T.J., and Neumann J., "NovAtel's RT-20 - A Real Time Floating Ambiguity Positioning System", *ION GPS-94*. Salt Lake Citv, September 20-23, 1994

[91] Neumann J.B., Manz A., Ford T.J., and Mulyk O., "Test Results from a New 2 cm Real Time Kinematic GPS Positioning System", *ION GPS-96*. Kansas City, September 17-20. 1996

[92] Wertz, J.R. (editor), *Spacecraft Attitude Determination and Control*, Reidel Publishing Company, 1978

[93] Farrell, J. and Barth, M., *The Global Positioning System and Inertial Navigation*, McGraw Hill, 1999.

[94] Jain R.; Kasturi R.; Schunk B., *Machine Vision*, Computer Science Series, 1995 McGraw-Hill International Editions.