World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:11, No:10, 2017

A Real-Time Simulation Environment for Avionics Software Development and Qualification

U. Tancredi, D. Accardo, M. Grassi, G. Fasano, A. E. Tirri, A. Vitale, N. Genito, F. Montemari, L. Garbarino

***Abstract*—**The development of guidance, navigation and control algorithms and avionic procedures requires the disposability of suitable analysis and verification tools, such as simulation environments, which support the design process and allow detecting potential problems prior to the flight test, in order to make new technologies available at reduced cost, time and risk. This paper presents a simulation environment for avionic software development and qualification, especially aimed at equipment for general aviation aircrafts and unmanned aerial systems. The simulation environment includes models for short and medium-range radio-navigation aids, flight assistance systems, and ground control stations. All the software modules are able to simulate the modeled systems both in fast-time and real-time tests, and were implemented following component oriented modeling techniques and requirement based approach. The paper describes the specific models features, the architectures of the implemented software systems and its validation process. Performed validation tests highlighted the capability of the simulation environment to guarantee in real-time the required functionalities and performance of the simulated avionics systems, as well as to reproduce the interaction between these systems, thus permitting a realistic and reliable simulation of a complete mission scenario.

***Keywords*—**ADS-B, avionics, NAVAIDs, real time simulation, TCAS, UAS ground control station.

I. Introduction

OVER the past years the Italian Aerospace Research Centre (CIRA) has invested significant resources in the development of innovative algorithms, implemented into avionic software, for the guidance, navigation and control of the aircraft, aimed at increasing the autonomy of Unmanned Aerial Vehicles (UAVs) in all flight conditions, and improving the aircraft's capabilities of flight assistance for pilots of General Aviation (GA) vehicles. CIRA is also investigating avionic technologies and procedures for the integration of UAVs and Remotely Piloted Aircraft System (RPAS) within the future civil ATM system [1].

In order to develop and test the above-mentioned algorithms, technologies and procedures, suitable analysis and verification tools should be available. Consequently, both the tools and the development process affect the correctness of the software. Concerning verification, several steps are foreseen during the development cycle of new concepts. They are based on the use of functional and detailed simulation environments, for on ground tests, and on actual experimental flying facilities, for the final in flight demonstration [2]. In particular, the simulation environments allow performing fast time simulation and real-time tests with software and/or hardware and/or human in the loop. Simulation facility provides ways to emulate realistic environment to access un-measurable flight variables, assess unforeseen human behavior, and conduct sensitivity analysis on the performance with respect to flight and scenario parameters [3]. They also allow detecting and preventing unexpected hardware malfunctions as well as errors in the real-time simulation codes or other potential problems (for example due to sensor noise and actuator lag) prior to the flight test. In conclusion, these tests permit dramatically reducing costs, time and risks of the development process. Of course, the implementation of a detailed simulation environment running in real-time has to face many challenges, whose solution may introduce a high complexity level in each simulation module composing the environment [4]. These challenges mainly concern:

- the number of systems and actors to be modeled;
- the necessity to make their simulated behavior as close as possible to the reality.

The real-time simulation framework has been widespread used by industry and academia for several applications in different fields. In the aerospace context, as said above, real-time simulations are possibly carried out before performing the inflight demonstration.

In the framework of the Italian national project MISE, financed by the Italian Ministry of Economic Development, CIRA developed and tuned a detailed simulation environment for high fidelity modeling of ATM scenario, UAV and GA vehicles, on-board and ground based avionic systems dynamics [4]. Within this project, CIRA started a collaboration with the Universities "Parthenope" and "Federico II" for modeling and

A. Vitale is with the On-Board Systems and ATM at CIRA (Italian Aerospace Research Centre), Capua, Italy (e-mail: a.vitale@cira.it).

U. Tancredi was with Department of Engineering at University of Naples "Parthenope", Italy. He is now with European Patent Office (e-mail: urbano.tancredi@uniparthenope.it).

D. Accardo, M. Grassi, G. Fasano, and A. E. Tirri are with the Department of Industrial Engineering at University of Naples "Federico II", Italy (e-mail: domenico.accardo@unina.it, michele.grassi@unina.it, giancarmine.fasano@unina.it, annaelena.tirri@unina.it).

N. Genito, F. Montemari, and L. Garbarino are with the On-Board Systems and ATM at CIRA, Capua, Italy (e-mail: n.genito@cira.it, f.montemari@cira.it, l.garbarino@cira.it).

software implementation of radio-navigation systems, flight assistance systems and Unmanned Aerial System (UAS) ground station. The present paper describes these models (which have been integrated into the CIRA high fidelity real time simulation environment) and their software implementation. These models are denoted as software modules in the following of the paper.

The Flight Assistance System software module is composed of Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Alert and Collision Avoidance System (TCAS) systems. The Radio-Navigation Systems (NAVAIDs) software module provides simulation for the most used radio navigation aids available for the civil aviation. More specifically this system is composed of four different devices: the Very High Frequency (VHF) Omni Directional Radio Range (VOR), the Non-Directional radio Beacon (NDB) and Automatic Direction Finder (ADF) system, the Distance Measuring Equipment (DME), and the Instrument Landing System (ILS). Two modules are used to model each Radio-Navigation system: the first module simulates the ground station equipment, while the second one simulates the on-board receiver (also called "Airborne Equipment"). In addition to the simulation software, a database of all the NAVAIDs available in the Italian airspace was created. The Ground Control Station (GCS) software module simulates the data link connection between the aircraft and the Ground Station and dynamics, delays and errors due to the interaction between GCS interfaces and human operators. It also includes some simplified logics, which model the behavior of the pilot and the mission operator in predefined mission scenarios and events.

The software modules, implemented in MATLAB/Simulink environment, can be used for both fast-time and real-time simulations. The life cycle of the software modules starts with the definition of their tool operational requirements (TORs). Then the tests to be performed to verify the TORs are designed. The TORs are needed to identify the requirements of the software modules from the user's point of view. Next, the software modules are developed in two phases: first, the high-level requirements are generated by the analysis of the TORs, then the detail design is performed, producing the Simulink models. Finally, the Simulink models are subjected to the test phase, in order to assess the verification of the TOR.

The settings of MATLAB/Simulink environment for the software modules development were configured to constrain the model developer to follow some tightening modeling rules, which avoid obtaining inaccurate or inefficient simulation of the system that the model represents. Such rules are derived from safety consideration and algorithm accuracy objectives from the RTCA-DO178C [5] and its accompanying supplement DO-330 [6]. According to these standards, CIRA framework environment is compliant with the second criteria, that is, "a verification tool that could fail to detect an error, and is used to reduce other development or verification activities" [6], with a tool qualification level 5. Indeed, since the software modules will be integrated into the CIRA simulation environment, and used to verify prototypical on board software that could be safety critical and subject to a certification process, these modules (together with the whole CIRA simulation environment) could be qualified. If the verification tool is qualified, when prototypical software is assessed using the tool, it is easy to generate realistic test vectors based on the designed scenarios.

The described development process allows increasing the reliability of the simulation environment to be used as verification tool and obtaining the C-code of the simulation environment through automatic code generation tools, such as Automatic Program Building (APB) [2] provided by the MathWorks. This C-code does not violate RTCA-DO178-C modeling standards or guidelines, and can be deployed on a target hardware for real-time simulations.

II. Flight Assistance Systems

The following two subsections report a detailed description of Flight Assistance Simulink™ models developed for the project framework. Two types of systems are modeled for the mentioned application, such as:

- ADS-B that is a system that provides airborne surveillance capabilities by broadcasting ownship GPS data and other navigation data through 1090MHz extended squitter or 978MHz Universal Access Transmitter (UAT) radios. The same data are received from local traffic to support airborne surveillance;
- TCAS that is a system that provides autonomous Collision Avoidance capabilities to aircraft equipped with mode-S transponder. It is capable to autonomously detect a collision threat on the basis of transponder output and airborne radar input. This system can negotiate proper autonomous avoidance maneuvers.

Extended details on the operating capabilities of both systems are reported in [7]. The models aimed at simulating the output of the above reported systems as it is received by on board data buses, such as MIL 1553 and ARINCTM 629 buses. Depending on the format that is used to produce the data, two output modes are considered for each system, such as:

- Single Report (SR), i.e., each time an input is received from an aircraft included in the local traffic it is formatted and transmitted to the bus with minimum latency. This model generates a continuous data stream, if enough traffic is present;
- Multiple Report (MR), i.e. all inputs received from the local traffic in a selected timeframe, e.g. 1 second, are grouped and transmitted to the bus simultaneously in the form of matrix. The stream generated by this model presents data bursts.

A. ADS-B

The ADS-B Simulink model is reported in Fig. 1 with its input and output lines. The same layout is provided for both MR buffered configuration and SR output configuration. The sole difference is that the first one generate a n x m x p matrix as output, whereas the second one generates an n x p vector at higher data rate, where n is the number of message parameters, i.e. 67, m is the index of a single intruder in the report matrix ranging from 1 to 100, and p is the index of sample that depends on the temporal extent of simulation. The input signals for the model are derived from standard ADS-B configuration [8], [9]. The model produces two different signals:

- ADS-B In signal that reports navigation and situational information about traffic to the onboard data bus;
- ADS-B Out signal that reports navigation and situational information about own-ship to the onboard data bus. Indeed, this is the data that are prepared to be transmitter to traffic in order to be received ad ADS-B In.

The protocol is different for 1090 MHz extended squitter Mode-S transponders and 978 MHz UAT radios. In the first case, due to reduced bandwidth, the whole message is split into two reports, i.e. State Vector Report and Mode Status Report, which contain both duplicated critical information, such as GPS time of arrival and Participant Address, plus non-duplicated non-critical information.

ADS-B Out message is formed by merging data from its primary sources, i.e. Air Data Computer - ADC, Flight Management System - FMS, Satellite Navigation Receiver - GPS, Inertial Navigation System – INS, and Transponder – TRANS. Internally, all the operations executed by the ADS-B model are depicted in the first layer of the model, see Fig. 2. First, messages from different sources are grouped to form ADS-B Out message. If no failure is generated for ADS-B Out function, the ADS-B Out block produces the report after an assigned time delay has passed. Subsequently, a proper block verifies if the relative range and altitude between the own aircraft and intruder are less than a threshold value in order to output the intruder state vector. Otherwise, the model outputs a null vector. Moreover, a proper layer verifies that no failure has affected the receivers. Then, a further layer accounts for random effects that prevent from receiving messages, such as multiple messages collisions and temporary signal losses. A stated probability threshold is assigned to determine the generation of failures. Finally, the output generation module creates the matrix output for all messages received within the time interval of 1s by assigning at each column the intruder ADS-B mode S protocol. This block is not provided in the single report model.

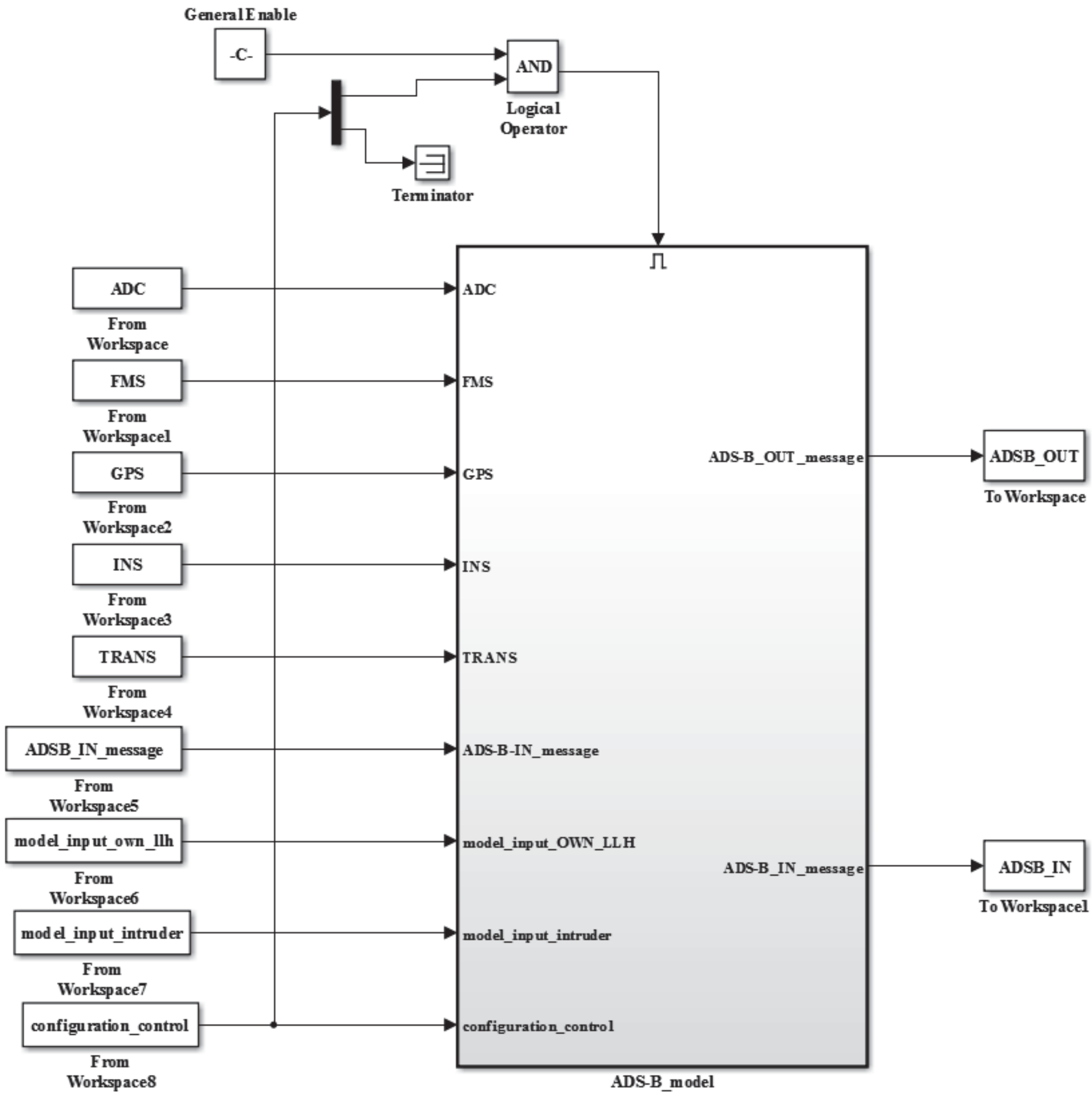


Fig. 1 ADS-B Simulink model external layer layout

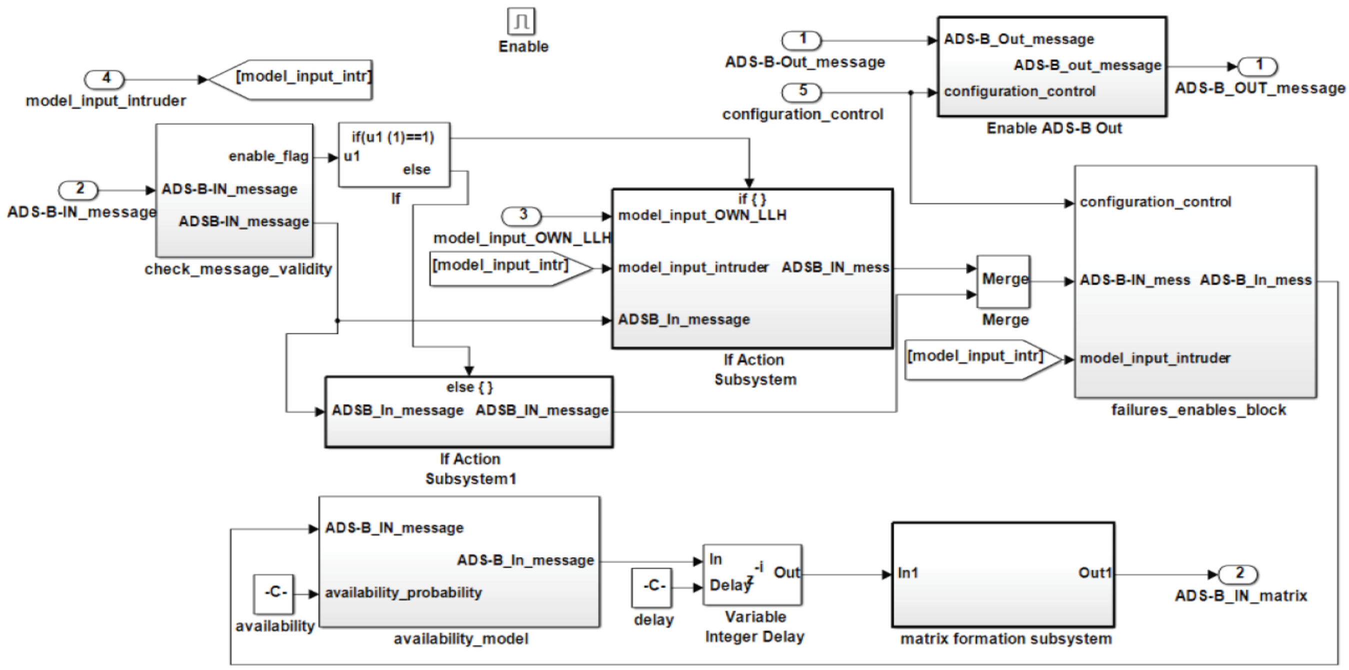


Fig. 2 ADS-B Simulink model internal layer layout

B. TCAS

Fig. 3 shows TCAS Simulink model with the inputs and outputs terms. The same layout is provided for both Multiple Intruder buffered configuration and Single Intruder output configuration. The sole difference is that the first one generates an n x m x p matrix as output whereas the second one generates an n x p vector at higher data rate (where n, m and p are defined as for the ADS-B model). It is worth noting that the system generates only the segment of TCAS data that are provided to estimate relative traffic and ownship position and velocity and not the segment that includes information about the Collision Avoidance logic.

The internal layer of TCAS model is reported in Fig. 4. In this layer, there are five blocks: TCAS_Own_message formation block, If Action Subsystem, TCAS_interrogation_mode_subsystem, a delay, and Output_generation subsystem. The second and third block are enabled by a flag (intruder_flag_selection) defined by the operator in the input file. This flag allows selecting the intruders for which the output is desired among those in the airspace volume (defined in the Num_intruder block).

The TCAS_interrogation_mode_subsystem is comprised of three blocks, such as: range and altitude compatibility block, TCAS intruder message formation module, and TCAS_mode_selection. The first one checks for compatibility of output in terms of range and relative altitude. The second one calculates the intruders' relative position. The last model provides the output depending on the interrogation mode (TCAS_mode input) only if range between the own aircraft and intruder is less than a threshold value in order to create a generic volume of TCAS-equipped aircraft. The output generation module creates the matrix output assigning at each column the intruders variables, which depend on TCAS interrogation flag. If the flag is null, the output is a null vector, i.e. there are no aircraft selected by the operator in the input file; otherwise the system outputs the intruder state vector containing the information defined above. In the single report mode, the matrix is not generated, and each intruder is output as soon as it is made available and the time delay is passed.

III. Short and Medium-Range NAVAIDs

The following sections are concerned with the most used short and medium-range radio navigation systems for the civil aviation, NAVAIDs for short. Description of their operating principle and features is widely available [7], [10] and is not carried out herein.

A. Model Features

The primary focus of the developed models is to reproduce the navigation functionality of the NAVAID, that is, the capability of the NAVAID to provide a measurement of (part of) the aircraft CoM position. The variable of interest is thus the output at the airborne side of the system, denoted as the *measured output* in the following. The measured output is the only output that is considered interesting for modeling the navigation function. *Auxiliary outputs*, i.e. those that can be derived by the primary one, such as the Course Deviation in VOR receivers, the Time To Station (TTS) in DME interrogators and so on, are also foreseen. The measured and auxiliary outputs of the four NAVAIDs are listed in the following table. Fig. 5 aids in interpreting the relevant variables. All the NAVAIDs, being based on a radio signal transmission, have a limited coverage. The effective coverage depends on numerous factors, including the sensitivity of the airborne equipage, propagation losses, atmospheric radio noise, etc.

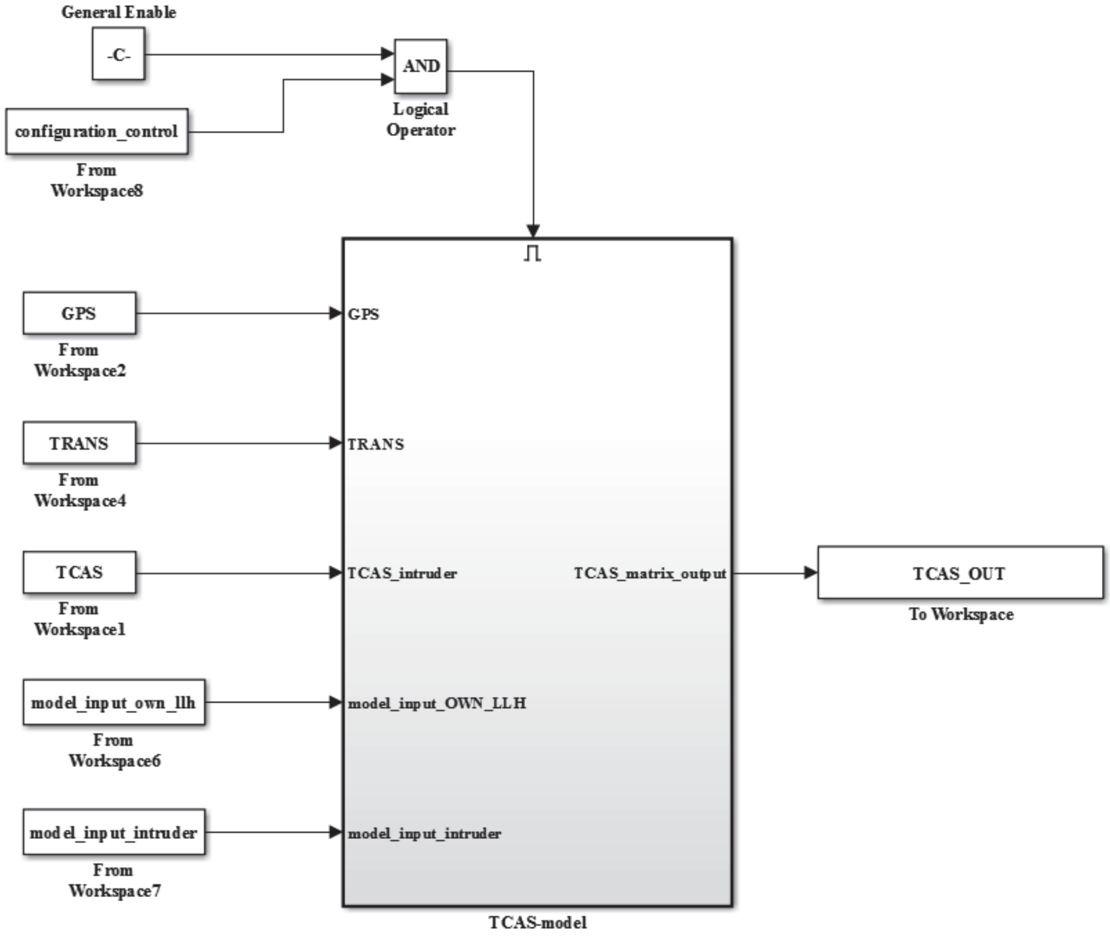


Fig. 3 TCAS Simulink model external layer layout

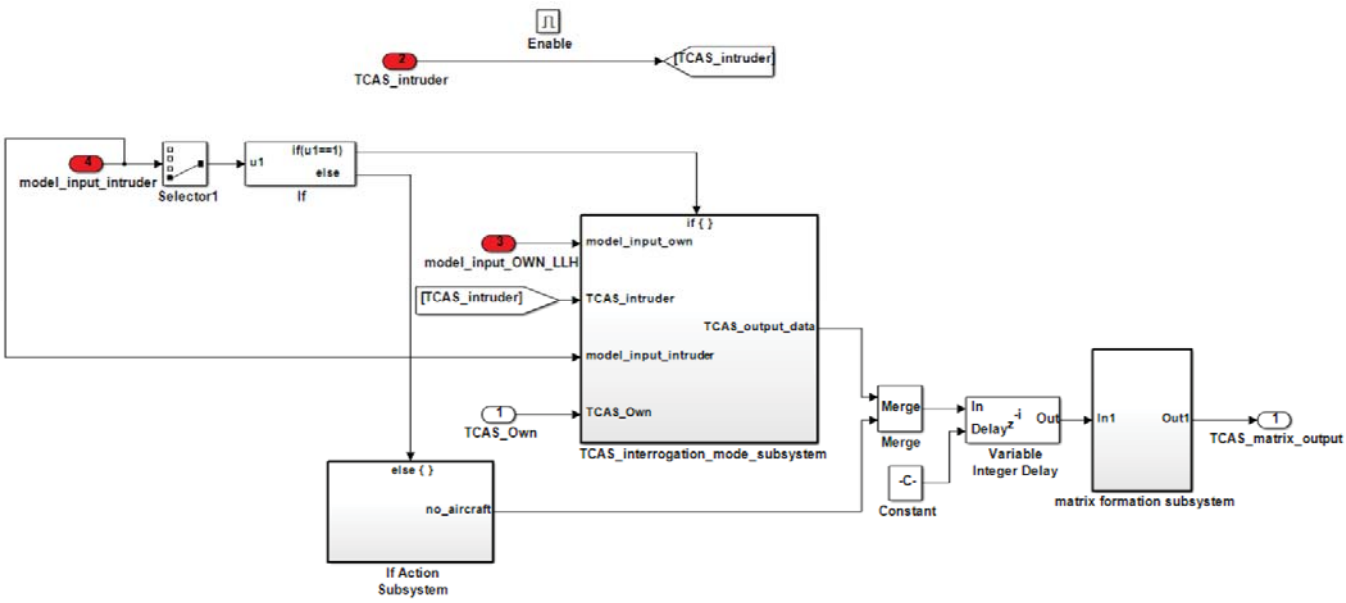
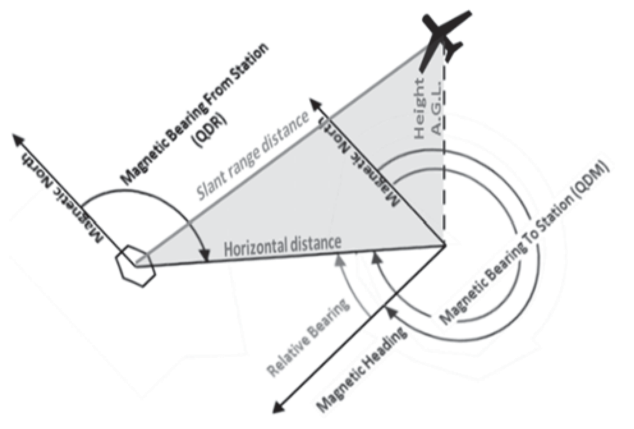


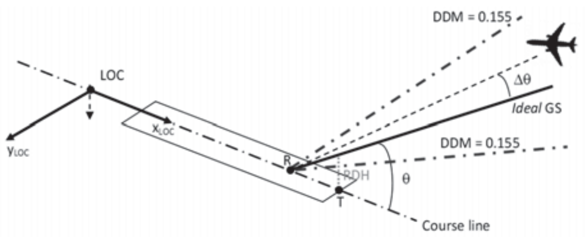
Fig. 4 TCAS Simulink model internal layer layout

For establishing unambiguously, the space in which the SiS shall allow a measurement complying with the international standards, the reference concepts of Service Volume (SV) and rated coverage have been introduced in [11], [12]. Because of the focus on the NAVAIDs navigation function, the Standard Service Volumes [12] are used in the models. Fig. 6 shows the

SSV implemented for a High-altitude VOR (HVOR) ground station.



(a)



(b)

Fig. 5 (a) VOR, NDB/ADF, DME measured output; (b) ILS measured output

TABLE I
NAVAIDs MODELED OUTPUT

NAVAID	Measured Output	Auxiliary Output
ILS	Deviation from course line Deviation from glide slope Marker beacons reception	Ground Station (GSt) id code Integrity Flag
VOR	Magnetic bearing to station (QDM)	GSt id code Course Deviation TO/FROM Integrity Flag
NDB/ADF	Relative bearing to station	GSt id code
DME	Slant range distance	GSt id code Groundspeed Time-To-Station

B. NAVAIDs Error Models

The NAVAIDs measured output is corrupted by typical errors, accounting for typical error sources and dynamics. Even though error features are specific of each NAVAID, a common structure can be discussed. The error is apportioned into three contributions:

- Ground Station (GSt) error, including all errors arising within the ground station;
- Airborne Equipage (AEq) error, including all errors arising within the airborne equipage (e.g. receiver, antenna and wiring);
- Propagation error, comprising errors due to non-ideal propagation of radio signals (e.g. site effects).

NAVAIDs error models are based on requirements laid out in [11], and by adaptation of the models proposed in [13], [14]. The NAVAIDs error *e* is split in two aliquots: a constant in time systematic bias β (i.e. a random constant) and a zero-mean, time-varying, random component, γ. The constant bias comprises, in general, station and receivers effects, and is modeled as a sum of independent random constants, with zero-mean Gaussian distribution. The zero-mean random component γ, including propagation effects, is modeled as a first order Gauss Markov (GM) process of variance σ_γ^2.

$$e = \gamma + \beta_t = \gamma + \beta_{GSt} + \beta_{AEq} \quad (1)$$

The variable error is assumed to be a non-zero mean, Gaussian, exponentially correlated stationary stochastic variable, with cut-off frequency ω_2. The error is realized by a first-order shaping filter, which is used to color a white noise w(t) having PSD defined by $\Phi_{ww}(j\omega)=\sigma_\gamma^2$.

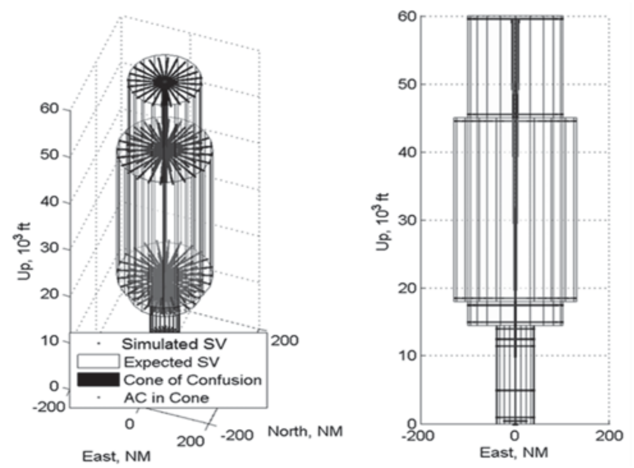


Fig. 6 High-altitude VOR (HVOR) Service Volume (SV)

The overall structure of the shaping filter is represented in (2) and Fig. 7 [14].

$$H(s) = \frac{\sqrt{2\omega_2}}{\omega_2 + s} \quad (2)$$

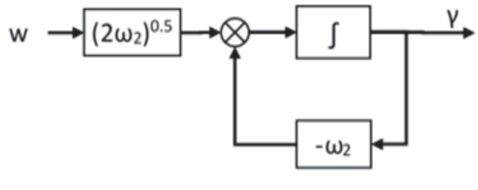


Fig. 7 Variable error block diagram

The cut-off frequency of the first-order GM process varies for each NAVAID, depending on its errors dynamics. Because the modeled variable error frequency distribution below ω_2 is basically flat, its value determines the maximum frequency of admissible errors. This is equivalent to assume that reflections from the terrain give rise to errors with no preferred frequency due to irregularity of the terrain, but only up to a certain frequency. Because these errors are due to terrain reflections, the cut-off frequency shall depend on the aircraft ground speed, usually in a linear fashion [13], [14].

Concerning the error magnitude, typical performances of NAVAIDs have been found to a limited extent only for US ground stations [15], whereas the airborne equipage typical performances are not readily available. However, international standards are available which dictate the performance that a NAVAID shall meet [16], [17]. Since these standards are expected to be met by any NAVAID, they are all-embracing generalizations of a typical system and their prescribed values may not reproduce with high fidelity any true-world system. Thus, they have been integrated with data available in the open literature [13]-[15], as far as practical.

Error budgets have been defined for all NAVAIDs following the previous approach. Results show, for instance, that the magnetic bearing error value for a VOR shall be between 4 deg. and 5 deg. (95th percentile) and the polarization error shall be within ± 2.0 deg. for roll angles ranging in ± 30 deg. The NDB/ADF relative bearing error absolute value 95th percentile, inclusive of all error sources, shall be 5 deg. when using a Compass Locator and 10 deg. when using other NDBs. The error budget for the DME NAVAID is more involved, because of the different accuracy standards for DME/N systems for en-route aids, for landing aids, or DME/P ones. Depending on the system components, the slant range distance error 95th percentile ranges between 90 meters in final landing segments with a full DME/P system, to over 350 m for a legacy DME/N-DME/N system. ILS error budgets are also complicated by the several possible combinations of GSt and AEq categories. Standard error budgets have been assumed. The reader is referred to [13] for a detailed overview.

True-world spatial and temporal correlation among measurements of different receivers locked onto the same GSt has also been reproduced. Clearly, only part of the measurement error experiences such correlation (e.g. terrain reflection effects). More precisely, the correlation between relevant errors is full for two AEq locked onto the same GSt in the same location at the same time, and decreases as the relative distance increases. The correlation becomes null at a certain correlation scale distance. To simplify the error correlation model, a realistic error correlation is sought only for distances smaller than the correlation scale distance.

Error correlation is enforced for all applicable ground station and propagation errors in the horizontal plane, both in the radial and in the tangential direction. This error correlation is simulated by specifying a deterministic variability both with horizontal distance d and with magnetic bearing QDR. For instance, referring to the variable signal γ(t), independent signals with the same variance of γ (i.e. γ1,…,γ4) are generated and mixed by applying sine and cosine coefficients depending on d and QDR.

$$\gamma(QDR) = \gamma_1 \sin(n_c \cdot QDR) + \gamma_2 \cos(n_c \cdot QDR) \quad (3a)$$

$$\gamma(d) = \gamma_3 \sin\left(\frac{\pi}{2}\frac{d}{L}\right) + \gamma_4 \cos\left(\frac{\pi}{2}\frac{d}{L}\right) \quad (3b)$$

The correlation scale distance is modeled by the n_c and L terms. The n_c term controls the correlation scale distance for aircraft flying on different radials, whereas L stands for the horizontal distance at which errors become orthogonal. Suggested values are 5 cycles per 360 deg. QDR shift for n_c, and 10 n.mi. for L.

C. Models Architecture

Each NAVAID model is divided into two independent modules: a GSt and an AEq. Separation of the two modules allows locating the ground station on an independent PC, serving as a scenario simulator for avionics. The two models interact via a Signal in Space (SiS), which mimics the navigations functionality of the signal radiated by the GSt without modeling its radio-electric features. This signal allows simulation of the measured output in the airborne equipage model, including realistic measurement error. As such, it conveys the same information contained within the true-world SiS, but not coded into an electro-magnetic wave, such as the GSt id code, the signal's frequency, the GSt position, the ground station measurement error, coverage, rated accuracy, and all other GSt attributes needed to compute the measured output.

The architecture of the GSt module is shown in Fig. 8. The GSt is in charge of emulating the generation of the SiS. This is done, in practice, by selecting the desired GSt within a GSt database and loading its relevant attributes. A measurement error block is also provided for realizing the "seeds" necessary to the airborne error model for realistically reproducing spatial and temporal correlations among different receivers (see, e.g., γ1 and γ2 in (3)). A total loss failure mode is also realized for performing risk analyses. A model configuration control signal enables and disables the various parts of the model.

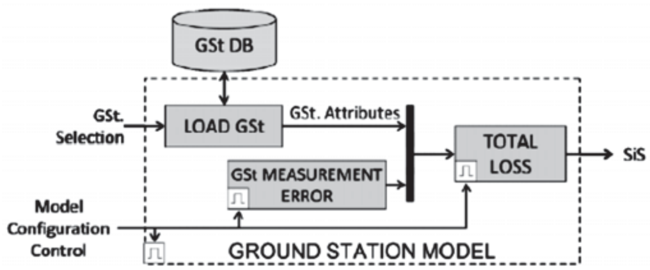


Fig. 8 Architecture of the typical Ground Station model

The GSt database has been created enclosing all Italian NAVAIDs. The official data source on the status and properties of the NAVAIDs under the Italian jurisdiction is ENAV's AIP Italy [18]. A total of 388 entries relevant to 148 unique GSt id codes are present into the database, which encloses 64 VOR, 98 DME, 73 NDB and 52 ILS ground stations. Fig. 9 portrays the geographical distribution of these NAVAIDs.

The architecture of the AEq module is shown in Fig. 10. The AEq is in charge of emulating the generation of the measured output on board the aircraft, as well as other auxiliary outputs. The AEq can be tuned on any GSt via the GSt selection signal (which is completely equivalent to the GSt selection for the ground station module). For mimicking true-world operation, an ether input port is provided, which accepts a stack of whatever finite dimension of SiS. If the AEq finds the SiS corresponding to the desired GSt, then it starts running the ideal measurement block shown in Fig. 10. Other inputs that control the operation of the AEq, such as the OBS in the VOR receiver, are also inputs to the model, and denoted as control inputs. Clearly, the model must also foresee all information necessary to model the ideal measurements as well as relevant errors, which are denoted as model inputs.

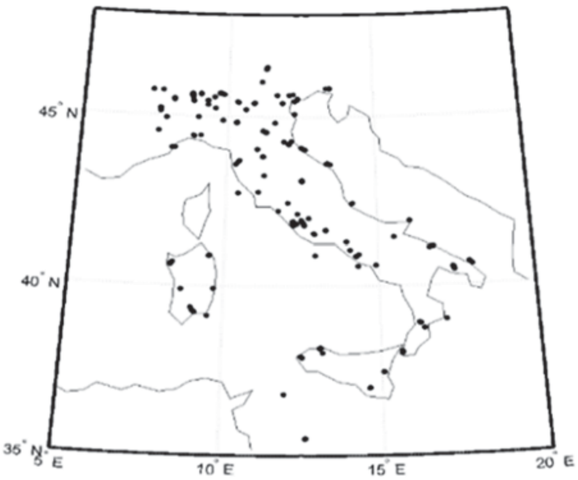


Fig. 9 Italian NAVAIDs geographical distribution

The ideal measurement, which is the value appearing in the measured output in ideal conditions, without errors, is computed taking into account the GSt and AEq positions, provided that the AEq is within the GSt coverage. As discussed before, because of the focus on modeling the NAVAIDs navigation function, we chose to provide an on-off signal acquisition depending on the NAVAID SV. The AEq thus measures the output if and only if the aircraft is inside the GSt SV. A measurement error block is also provided for realizing the error's time-history. As previously discussed, realization of the measurement errors is performed taking into account possible error correlation among different AEq locked onto the same GSt. This involves generating random errors within the AEq, but also processing the "seeds" generated within the GSt and available to the AEq via the SiS. As for the GSt module, a total loss failure mode and a model configuration control signal are foreseen.

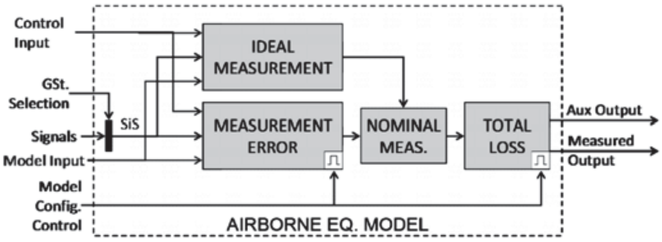


Fig. 10 Architecture of the typical Airborne Equipage model

IV. UAS Ground Control Station and Data Link

Control stations (mostly based on ground) and communication links are a key component of every class of UAS [19], [20], and they heavily affect the level of situational awareness that the remote pilot can achieve. While the GCS is in general the man-machine interface with the UAV, the maintenance of the communications is of paramount importance in UAS operations, and data link characteristics are a primary driver for guidance and control architecture. In general, GCS and data link components, functions and dimensions are closely related to UAS class and its mission. In particular, an important architecture driver is whether the system can operate only line-of-sight (LOS) or if it is capable of satellite-based beyond-line-of-sight (BLOS) operations. Communication latency is usually increased remarkably in BLOS operations.

The developed model is aimed at a realistic simulation of main GCS and data link components, and of their impact on the overall system. Simulated GCS includes the two most common personnel roles, i.e. the remote pilot and the mission operator. Moreover, ground components such as visualization interfaces and ground computers are taken into account in the developed software. The simulation model architecture (Fig. 11) comprises the following modules: Payload Data Link, Command and Control (C2) Data Link, Human Machine Interface Module (HMIM) for pilot and mission operator, On Ground Computer Module (OGCM) for pilot and mission operator, Pilot Module (PM), and Mission Operator Module (MOM). Output signals comprise uplink data that are generated by the pilot or by the mission operator, whilst input signals comprise:

- Payload data, i.e. data from mission sensors onboard the UAV;
- C2 downlink data for pilot/mission operator, i.e., the set of downlink data that are of interest for remote pilot/mission operator;
- Model input: variables relevant to the current UAV position and the current signal absorption of the payload and the C2 link (in dB/km). An auxiliary Simulink model has been implemented for calculating atmospheric losses as a function of wavelength and weather conditions;
- Model configuration control: inputs that can be used to command failures of single sub-modules. In particular, failures of ground computers and/or communication links can be commanded during a simulation run;
- Commands for pilot/mission operator module, i.e., commands from the simulation environment that activate proper pilot/mission operator events.

Considering the main features of the simulation model architecture, it is worth underlining that the two data link modules are simulated independently. This is consistent with the fact that C2 and payload links usually have to fulfill different requirements and are often designed using different frequency bands. In fact, C2 link is flight critical when the unmanned aircraft is incapable of landing without positive control, is often secured against jamming, and is generally low bandwidth thus being compatible with frequencies in the vhf and uhf spectra. On the other hand, payload link is mission critical and is often characterized by large data rates which require higher frequencies. In the developed software, operation of each data link module is influenced by model input and configuration control commands. GCS constant parameters are defined in an external script. They comprise all the parameters associated to link budgets, GCS characteristics, communication and operation latencies.

In general, loss of communication during operations may result from data link failures, loss of LOS conditions (when BLOS capabilities are not implemented), weakening of received power due to the distance from the GCS, and intentional or inadvertent jamming of the signals [19]. These events are simulated in the developed software. In fact, while the configuration control signal can be customized to simulate data link failures or jamming, payload and C2 data link modules verify line-of-sight coverage by taking link budget and geometry into account. They also define the correct delays to be applied to communication messages or disable the communication link if LOS is not achievable and a LOS-only GCS is simulated. The core of these modules is represented by a block estimating current data link conditions and delay. The required signal-to-noise ratio (SNR) that guarantees an acceptable signal quality (for example in terms of bit error rate) is set in pre-processing phase. During simulation, maximum range that allows having the required SNR is computed by an embedded Matlab function. In fact, if atmospheric absorption is included in the link budget, a nonlinear equation in range results, which has to be solved by a numerical procedure. In the considered case, Newton-Raphson method is adopted, and the first tentative solution is the maximum range calculated without considering atmospheric losses. LOS conditions are verified if the actual range is smaller than the minimum between range thresholds calculated from link budget and geometry. Total signal losses in dB/km can be controlled in real time thus simulating variable weather conditions. To this aim, an auxiliary Simulink model has been built which calculates losses by a decibel sum of atmospheric absorption (basically due to water vapour and biatomic oxygen) and absorption losses due to precipitation (due to clouds/fog and/or rain) [21]-[23].

Other key components of the developed model are the Pilot Module and the Mission Operator Module. In this case, development has been aimed at enabling full control of these modules from the simulation environment (in view of man-in-the-loop simulations), while also implementing some "autonomous" operations which correspond to realistic behaviors in terms of decisions and latencies. In fact, generation of uplink commands can be based on inputs from the simulation environment (man-in-the-loop simulations), it can derive from downlink events on the basis of proper logics (e.g., request of manual command following onboard anomalies), or it can depend on the interaction between pilot and mission operator (e.g., activation of target following function by the pilot after identification of a region of interest by the mission operator). In the latter two cases, proper delays are applied which can be set in the GCS configuration file.

In fact, there is not a well assessed literature analyzing typical times needed by UAS operators to carry out mission-related operations: first research results have just been published [24] regarding typical times of interaction of UAS pilots with ATC. On the other hand, applying the same approaches envisaged for Air Traffic Controllers (i.e., setting proper delays for given operations) seems to be a reasonable choice. It is worth underlining that the developed architecture is fully flexible: for example, a real pilot-in-the-loop can operate in the simulation environment together with a simulated mission operator, and vice versa.

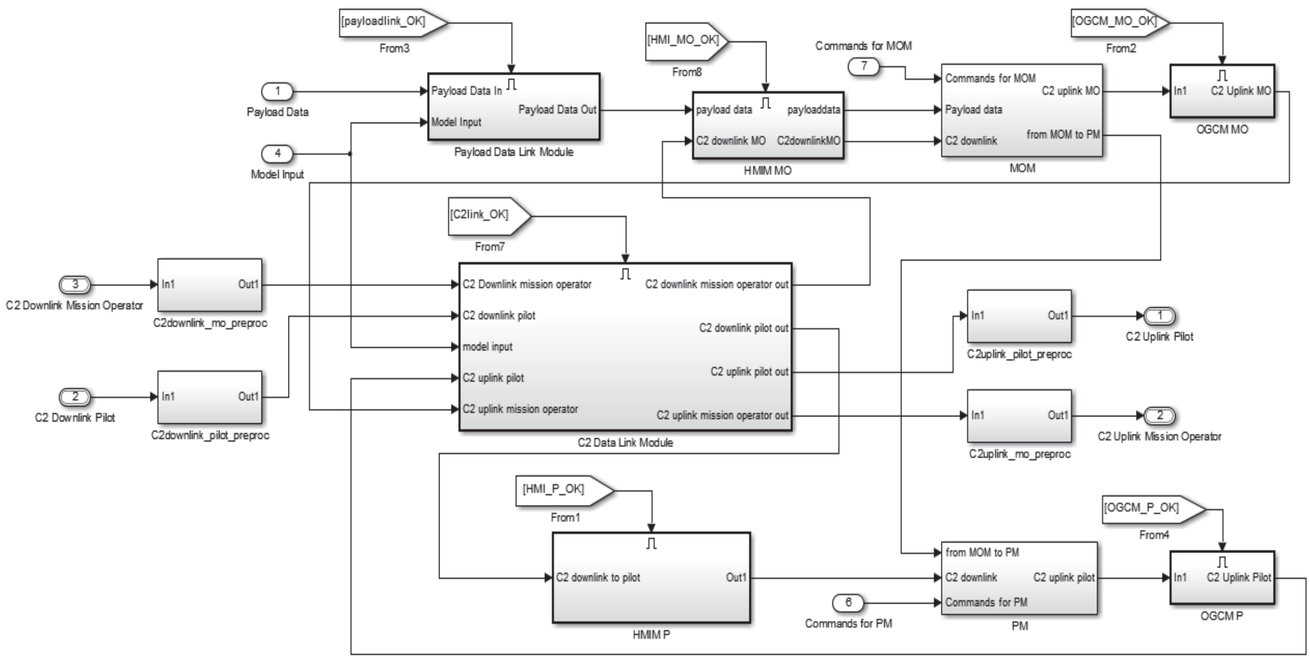


Fig. 11 Screenshot from the GCS & Data Link simulation model

Finally, OGCM and HMIM are modeled in terms of very short delays (both) and rate transitions (HMI modules only), which are introduced to take into account typical data visualization frequencies. The main interest here is in the possibility of introducing temporary failures of ground components. As a matter of fact, some UAS accidents resulted from ground failures and improper ground personnel reactions [25].

V. Simulation Models Unit Tests

An extended set of unit tests has been carried out; these tests were aimed at verifying each single software module as well as their interaction. The tests, described in the present paragraph, have demonstrated the capability of the simulation module of reproducing functionalities and performance of the target avionic systems.

The executed tests are based on a set of requirements previously established for the simulation environment (not reported in this paper for the sake of brevity). For the whole avionics software environment, 46 requirements has been defined. They need to be satisfied by tests, mainly regarding the desired behaviour of each avionic item in terms of functionality and expected performance. Globally, 160 tests has been executed to verify the conformity of the simulation environment to the requirements. A series of benchmark models have been designed to execute one or more test; each one is dedicated to a single avionic item. The benchmarks are implemented in Matlab/Simulink environment. They basically consist of three different elements:

- the avionic item under test,
- a subsystem that generates input for the avionic model and suitable for the test's objectives (which generate not only the input for the avionic model but also give the user the expected output, if needed for each test)
- a subsystem that process the output of avionic item for checking that test's criteria are satisfied.

This benchmark structure is common for all the executed tests, while each benchmark possess unique and distinctive features needed for tests of different avionic items. Each test is based upon verifying a certain number of Pass\Fail criteria; one benchmark can be used for performing more than one test. The requirements verification tests are designed for testing a single requirement on a specific avionic model. All tests are marked with a unique identifier, which closely resembles the ID of the requirement under test to ease readability of the results.

As an example, here is reported the test executed on the NDB Airborne Equipment avionic module and the relative benchmark model (Fig 12); the test objective is to verify the expected equipment coverage. Fig. 13 summarizes the result of this test; the measured output status of the avionic module (dashed line) is coherent with the expected one (continuous line), giving hence a positive test result.

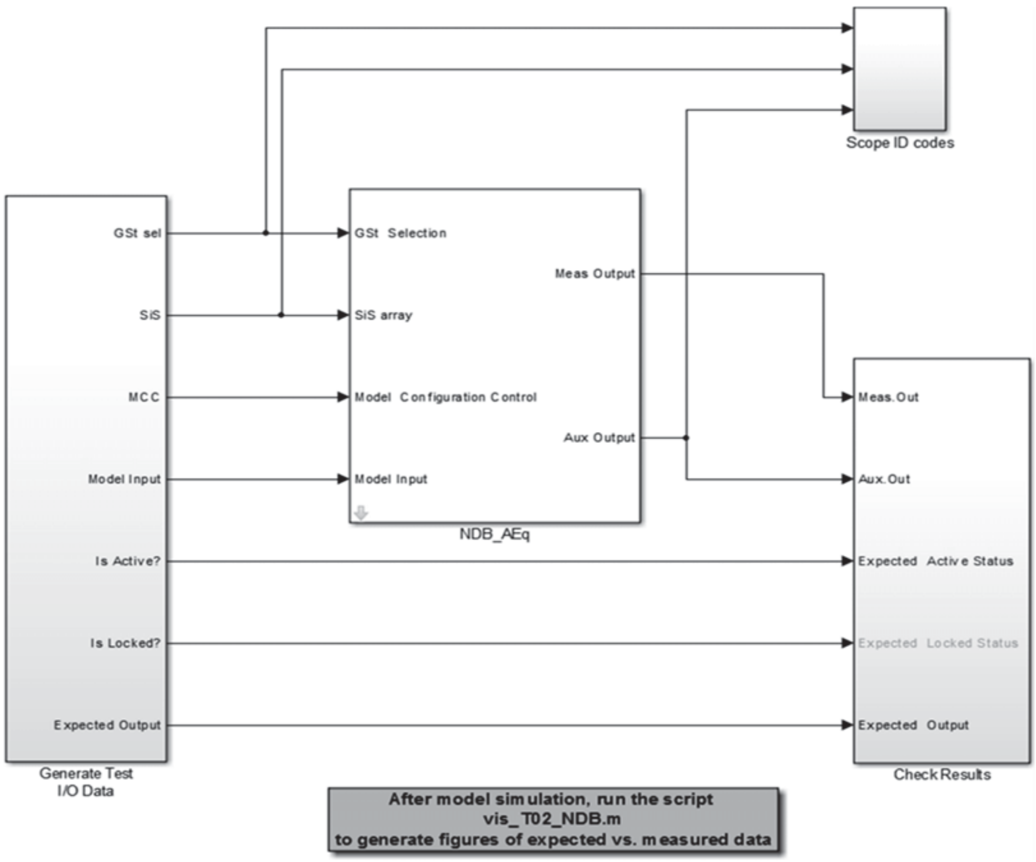
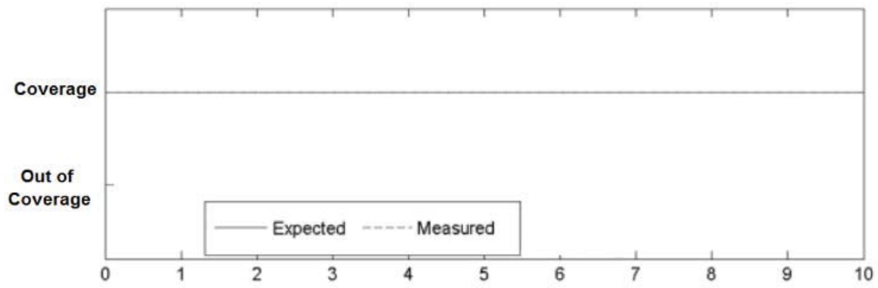
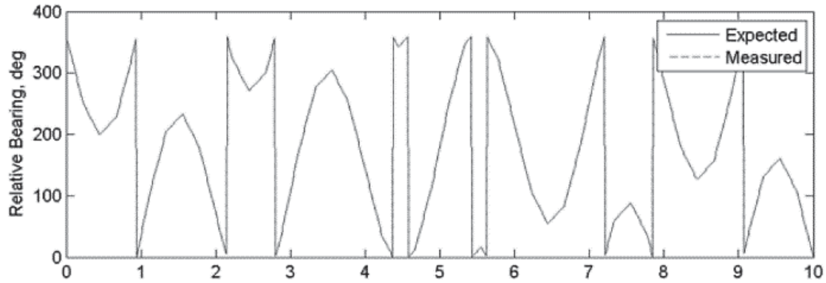


Fig. 12 NDB Airborne Equipment benchmark model



(a)



(b)

Fig. 13 NDB Airborne Equipment unit test: (a) Comparison between expected and measured; (b) Comparison between expected and measured relative bearing

VI. Real-Time simulation application

In addition to a functional and performance well defined behaviour, each Matlab/Simulink avionic module is expected to be complaint with a real-time simulation environment. To ensure this objective, each avionic module is designed following a set of rules derived from DO-178C [26]-[28]. The result is a Simulink model ready for being translated in C-code by the automatic code generator tool for dSpace Hard Real-time environment. In fact, today a Model-Based Design that use rapid prototyping tool chain is a common approach, highly reliable and flexible, allowing the development and deployment of new software compliant with real time environment.

The definition of a two step-approach is necessary for verifying each module. The first step is based on Simulink "Model Advisor" tool, used to check the compatibility of each avionic module with the DO-178C standards. The Simulink model is also used to perform an offline simulation, whose input and output are recorded. In the second step, for each module, the associated C-code is generated and compiled for the real-time dSpace platform [3]. Each module verification is performed on the running real-time platform using the input recorded in the first step; the offline and real-time outputs are compared, and if they coincide, the test is passed. Moreover, the computational effort was evaluated. This real-time test procedure verifies the capability of each module to be correctly converted into C-code useful for real time simulations.

A first example of real-time application of these modules concerns the ADS-B system. The ASD-B module has been integrated into the Integrated Simulation Facility (ISF) [4] developed by CIRA, able to reproduce manned and unmanned aircraft flight simulation in a real-time environment and capable to emulate Air Traffic Management and Air Traffic Control operations. Through this real-time simulation facility, the avionic ADS-B module has been widely used in various CIRA research projects allowing the researcher to execute laboratory tests of algorithms for autonomous separation and collision avoidance manoeuvres, based on ADS-B cooperative data exchange between aircrafts [1], [29].

VII. Conclusions

A real-time simulation environment for avionics software development and qualification for GA aircraft and UAV has been presented. The simulation environment includes models for medium-range radio navigation aids, for flight assistance systems, and for ground control stations. The simulation environment incorporates a database of the Italian radio-navigation aids, but, of course, it can be expanded with a small effort.

The proposed model architecture provides an effective reproduction of the considered avionics systems, as reported in the reference literature including aeronautical standards. This architecture allows operating and interfacing the several modules, corresponding to the various avionic systems, in an easily and effective way, thus permitting a realistic simulation of a complete mission scenario, from take-off to landing.

A wide set of tests were carried out on the presented simulation models. The tests demonstrated the capability of the models to guarantee in real-time the required functionalities and performance of the simulated avionics systems, as well as the capability to correctly interact between them.

REFERENCES

[1] E. Filippone, V. Di Vito, G. Torrano, D. Taurino, A. Ferreira, D. Zammit-Mangion, J. Gauci, G. Gargiulo, “RPAS – ATM integration demonstration – real time simulation results”, *15th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA AVIATION Forum, (AIAA 2015-3407).

[2] E. De Lellis, V. Di Vito, L. Garbarino, C. Lai, F. Corraro, “Design process and real-time validation of an innovative autonomous mid-air flight and landing system”, *World Academy of Science, Engineering and Technology* N.55 , pp. 174 (2011).

[3] A. Fedele, N. Genito, L. Garbarino, G. L. Di Capua, V. Baraniello, “A pilot-in-the-loop facility for avionic concept development”, *Journal of Automation and Control Engineering* (Vol. 4, No. 4, August 2016).

[4] N. Genito, F. Montemari, G. Corraro, D. Rispo, R. Palumbo, “Integrated simulation facility for interoperability operation”, *Lecture Notes in Computer Science* LNCS-8906 (2014).

[5] J. Kim, S. Lee, K. Ryu, “Development of avionics hot bench for avionics system simulation and validation”, *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA 2007-6362.

[6] *Software Tool Qualification Considerations*, RTCA, December 2011. DO-330.

[7] A. Helfrick, L. Buckwalter, *Principles of avionics*, 8th ed., Avionics Communications Inc., Leesburg, VA, USA (2013).

[8] *Minimum Operational Performance Standards for Universal Access Transceiver (UAT) Automatic Dependent Surveillance — Broadcast*, RTCA, Inc. December, 2009. DO-282B.

[9] *Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance — Broadcast (ADS-B) and Traffic Information Services — Broadcast (TIS-B). RTCA*, Inc.. December, 2011. DO-260B.

[10] M. Kayton, W. Fried, *Avionics Navigation Systems*, J. Wiley & Sons. Inc., London (UK), 1997.

[11] *ICAO ANNEX 10 Aeronautical Telecommunications Volume I, Radio Navigation Aids*, 6th Edition, 2006.

[12] *U.S. Department of Transportation, Federal Aviation Administration, Aeronautical Information Manual*, Feb.2012.

[13] G. B. Litchford, “Analysis of cumulative errors of cat. II, III operations with requirements for additional research”, NASA CR-1188, 1968.

[14] H. Johansen, “A survey of general coverage NAVAIDS for V/STOL aircraft: a VOR/DME error model”, MIT, NASA-CR-1588, 1970.

[15] *2010 Federal Radionavigation Plan, Department of Defense, Department of Homeland Security, and Department of Transportation*, Apr.2011.

[16] *ICAO, Manual on Testing of Radio Navigation Aids*, Volume I, Doc 8071, Fourth Edition, 2000.

[17] *U.S. Department of Transportation, Federal Aviation Administration, U.S. National Aviation Handbook for the VOR/DME/TACAN Systems*, FAA Order 9840.1, 1982.

[18] ENAV, AIP Italia, Jan. 2014.

[19] R. Austin, *Unmanned aircraft systems: UAVs design, development and deployment*, Wiley, 2010.

[20] J. Gundlach, *Designing unmanned aircraft systems: a comprehensive approach*, AIAA Education Series, 2012.

[21] Recommendation ITU-R P.676-9 (02/2012), "Attenuation by atmospheric gases".

[22] Recommendation ITU-R P.840-5 (02/2012), "Attenuation due to clouds and fog".

[23] R. E. Collin, *Antennas and Radiowave Propagation*, Mc Graw Hill, 1985.

[24] J. Ziccardi et al., “Measuring UAS pilot responses to common air traffic clearances”, *Lecture Notes in Computer Science* Volume 8017, 2013, pp 606-612.

[25] G. Carrigan, D. Long, M. L. Cummings, J. Duffner, “Human factors analysis of predator B crash”, AUVSI 2008.

[26] *Software Considerations in Airborne Systems and Equipment Certification*, RTCA, December 2011. DO-178C.

[27] B. Potter, “Complying with DO-178C and DO-331 using Model-Based Design”, 12AEAS-0090 (2012).

[28] R. G. Estrada, E. Dillaber, G. Sasaki, *Best practices for developing DO-178 compliant software using Model-Based Design*, 2013.

[29] M. Orefice, V. Di Vito, L. Garbarino, F. Corraro, G. Fasano, D. Accardo, “Real-time validation of an ADS-B based aircraft conflict detection system”, *AIAA Infotech @ Aerospace*, AIAA SciTech Forum, (AIAA 2015-0486).