SIMULATED REALTIME MULTI-CHANNEL IQ DATA GENERATOR FOR AN AESA RADAR

R. Gordini[(a)], M. Giorgiutti[(b)]

[(a)] SELEX Galileo S.p.A., Ronchi dei Legionari, Italy
[(b)] SELEX Galileo S.p.A., Ronchi dei Legionari, Italy

[(a)] romolo.gordini@selexgalileo.com, [(b)] michele.giorgiutti@selexgalileo.com

ABSTRACT

Modern airborne radar systems require a complex digital processing for target detection and tracking. An increasing challenge on testing is required to validate these algorithms before flight trials. Hence the need for a development rig, capable of simulating the sensor in a realistic and dynamic environment.

The research highlights potentialities of "HW in the loop" for the radar simulation, describing a simulated real-time multi-channel IQ data generator to be used by AESA radar manufacturer to test critical modes in the radar RIG.

The system provides at runtime the IQ data for the 4 complex channels, with different PRFs scheduling, in A/A modes.

The sensitive point of the work is the matching of realtime performance requirements w.r.t. band requirements. In fact, the IQ generator should guarantee an output bit-rate of 400 Mbytes/sec, scheduling its processing, triggered by an external sync signal, at 400 Hz, with reaction times of fractions of ms.

Keywords: Radar Simulator, IQ channels, Active Electronically Scanned Array (AESA) radar, realtime performances

1. INTRODUCTION

This industrial research is aimed to provide a real-time multi-channel IQ Data Generator as produced by the SELEX Galileo radar simulation toolkit for a modern Active Electronically Scanned Array (AESA) radar prototype test RIG.

The simulated IQ Data Generator allows the Signal Processing and Radar Tracking functions of the AESA Radar to be tested against simulated real time targets in a Lab Environment.

The radar simulator provides, for each radar burst, the IQ Data (number of coherent integrated pulses × number of non-ambiguous range samples) to the Radar Signal processor for the Sum Channel Digital Data, for the Difference Azimuth Channel Digital Data, for the Difference Elevation Channel Digital Data and for the Guard Channel Digital Data for the radar modes MPRF air-to-air, HPRF air-to-air, LPRF air-to-air and Calibration modes. The System consists of 4 independent commercial servers. Each server produces the IQ data for one radar channel. The IQ Data Generator receives real-time input data via an high speed serial link at 400 Hz for navigation data, and via the 1Gbit Ethernet link at 400 Hz for radar burst data. It sends out the generated IQ data via four 10 Gbit Ethernet optical links.

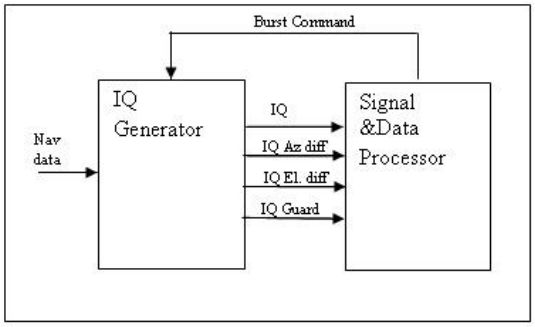


Figure 1: Architectural schema

The configuration of the test-rig is depicted in Figure 1. All elements perform in real-time. Input data is supplied to the IQ data generator from the Scenario Generator and the Navigation Simulator. Radar operational mode selection is provided by the Radar Control Panel Sim. According to the required operational mode, the Radar Data Processor (RDP) selects the radar waveform and creates a series of commands to generate the waveform and control the AESA radar. Data from these command is selected to stimulate the IQ data generator. The IQ data generator creates data representative of the target configuration at the time the command is received and sends the simulation results to the IQ output buffer. The external hardware combines the IQ data with the other parameters required from the RDP command message to mimic inputs to the processor. This complete IQ simulated data, plus header data, is sent to the processor over optical links (one link per channel).

For each channel the following radar effects are simulated:

ISBN 978-88-903724-3-8

1. Thermal noise.
2. Jet Engine Modulation (JEM).
3. Antenna Effects of electronic steering in azimuth and elevation.
4. Antenna Transmit pattern for Sum.
5. Antenna Receive pattern for Sum, Guard and difference channels.
6. Presence of multiple different targets into the scenario. A target signal accounts for the effects of:
   - Range
   - Range rate
   - Angular position within antenna pattern (including monopulse in difference channel(s))
   - Radar Cross Section (RCS)
   - Fluctuation models (Swerling 0,1 & 2)

The Radar Data Processor (RDP) generates new burst demands at rates up to 400Hz. A burst demand contains waveform and antenna command information. Each new burst demand is generated and sent whilst the current burst is in the progress of being generated. Whenever a new burst identifier is received, the real-time data generator provides the simulated IQ data for the new burst after the current burst has been concluded. The gap time between bursts has to be minimised (about 100 microseconds). Most demanding burst durations are for Air Combat MPRF and HPRF.

Considering the great amount of data to be processed and the communication rate required to the IQ Data Generator, the implemented solution consists of multiple independent workstations within a rack, basically one per channel to carry out the simulation. Each work-station shall be equipped with at least 8 cores (physical cores). The HW platform shall be identical for all the 4 radar channels. This solution not only distributes the computational load but the required band as well. Indeed a point to point communication topology is established to prevent any data collision, given the huge amount of data flowing. The chosen operating system is QNX 6.5 Neutrino (QNX).

Quantitative results of the implemented solution are presented and analyzed in details.

2. DESIGN CONSIDERATIONS

The main goal of this project is to reduce the design costs for an advanced airborne coherent radar, minimizing the need of flight trials to collect performance data. As customary in modern design processes, the development must be as standard as possible in order to make possible the reuse of the design for other radar installations. Considering these facts, the natural choice is to orient the implementation of a flexible radar test rig towards a software solution, which can be easily customized. The radar simulation toolkit is a multiplatform software that can run on the most widespread operating systems/processors, so that the choice of machine/OS is a degree of freedom for the designer (Windows, Linux, QNX, Unix and others).

A constraint of the system is that it must be hard realtime, so it has to respond within predictable time frames. This leads to the choice of QNX 6.5 Neutrino, a consolidated OS for embedded applications. The architecture chosen is the 64 bit Intel platform, given its high performance/cost ratio.

The other main constraint is the huge amount of data that must flow from the IQ generator to the signal processor. The theoretical figure is 400 Mbytes per second, in the worst case. Rather than centralizing all the simulation into a high performance machine, it is better to distribute the computational load, so that the required bandwidth for each single machine is a fraction of the centralized solution, and there is guaranteed parallelism between the machines. Even a mainframe with four powerful network adapters may not guarantee the necessary bandwidth all the times, depending on the implementation of that particular machine. This does not meet the objective of flexibility of such a system, since it binds the performance of the system to a peculiar machine, rather than to a machine class.

It can be argued that this implementation does not strictly guarantee realtime. In a sense it is true, because the rig relies on network technologies, but given the architecture, the probability of an overrun it isn't likely to occur. And if on overrun occurs it is detected, an exception is thrown. An overrun is not hazardous, since this is not a safety critical application, so just the reporting of the overrun it is enough. The exception will invalidate the exercise results, so that they won't be used in the radar processor development.

3. DATA SYNCHRONIZATION

Since a distributed computation scheme has been chosen for the simulation, a synchronization mechanism is necessary to have a consistent simulation. In particular NAV data and target data must be the same for each of the workstations performing the processing. To achieve input data alignment, an exercise based implementation of the targets has been devised. An exercise is a trajectory in time and space which is assigned to each of the simulated targets. Trajectories are defined as a sequence of segments, and each segment must have the following parameters specified:
Segment start and stop times (t_0, t_1)
Segment velocity as

$$\begin{cases} V_x(t) = V_{x,0} + V_{x,C}\ \cos\left(\omega_a\left(t-t_0\right)\right)\ \cos\left(\omega_b\left(t-t_0\right)\right) \\ V_y(t) = V_{y,0} + V_{y,C}\ \sin\left(\omega_a\left(t-t_0\right)\right)\ \cos\left(\omega_b\left(t-t_0\right)\right) \\ V_z(t) = V_{z,0} + V_{z,C}\ \sin\left(\omega_b\left(t-t_0\right)\right) \end{cases}$$

Segment linear acceleration (constant vector).
All these vectors are referred to a North oriented fixed Cartesian reference system (scene centre origin). Angular velocities Ω_a, Ω_b represent respectively the heading rate and the pitch rate of target. Before starting the realtime, hardware in the loop simulation, exercises are sent to the Synthetic Environment. The SE

ISBN 978-88-903724-3-8

computes target trajectories as define above, and sends the time tagged information to all processing machines. In this way all processing machines can implement the same identical interpolation of the data between two consecutives simulation frames.

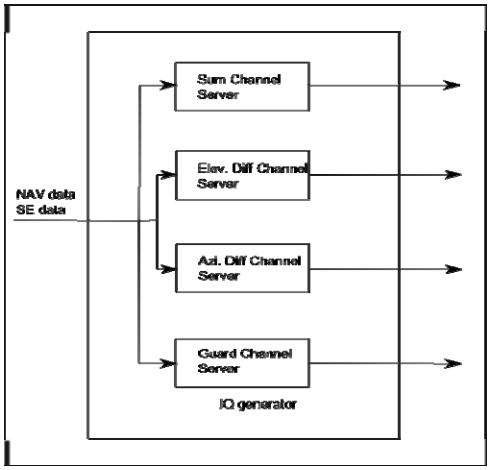


Figure 2: IQ generator block diagram

If any problem occurs, or the processing requires more time than prescribed, the output buffer is marked as not valid before is sent to the signal processor.

4. IQ GENERATOR PROCESSING OVERVIEW

Burst operation is often implemented in coherent radar modes. During burst operation the radar transmits packets of pulses, called bursts, and each pulse is separated from the previous one by dozens of μs. The radar return from each pulse of the burst is elaborated, and amplitude and phase of the signal are extrapolated for each range bin, representing the radar response in the range-time domain. This information is more suitable for digital processing in Cartesian representation, so amplitude and phase are converted into in phase and in quadrature components. Of course this gives the name to the IQ generator. Data to be processed for each burst can be stored in a matrix of complex numbers: the rows are ordered by pulse, hence time, the columns by range-bin, hence distance. Indeed the time range matrix represents the output of the IQ generator.

Table 1: IQ matrices

$I_{11} + j\,Q_{11}$	$I_{12} + j\,Q_{12}$	…	…	$I_{1N} + j\,Q_{1N}$
…	…	…	…	…
$I_{M1} + j\,Q_{M1}$	…	…	…	$I_{MN} + j\,Q_{MN}$

For non coherent radar operation the usual radar equation is employed (Picardi), hence only the received power is known, there is no way of estimating the relative phase of consecutive pulses. The relative phase of the signal can be computed considering the estimated wave vector $\vec{k}$ as follows: $\alpha = \vec{k} \cdot \vec{r}$, where α is the phase and $\vec{r}$ is the ray vector that connects the antenna bore-sight to the simulated target. From this it is clear that it is mandatory to perform accurate target slant range measures. Indeed the meter to be used is the wavelength at working the frequency, so the slant range resolution Δ must be less than $\lambda / 30$.

This implies that the simulation of targets, along with the navigation data, must be as accurate and as synchronized as possible. Standard Synthetic Environments do not meet the latter constraint so that a custom SE based exercise has been designed, as previously discussed. Between two consecutive frames an interpolation is applied, to limit side effects due to space quantization.

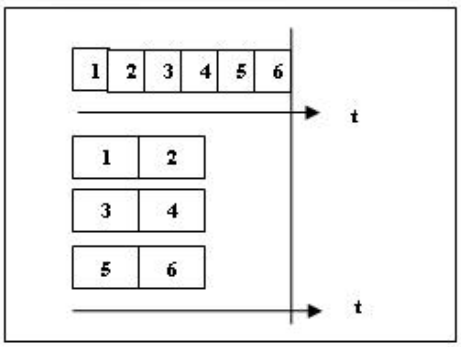
Additive white noise is inserted to model the receiver noise figure NF, the mean power of the random noise been computed as: $Pn = K_b T B N F$, where K_b is Boltzmann constant, T is the absolute temperature and B is the receiver's bandwidth.

5. PARALLELISM SCHEME

Given the stringent time constraint, it is essential to perform the processing as fast as possible. In order to achieve this, a two level parallelism is employed, at channel level and at pulse level. Data to be filled in each channel is of course not dependant from data of other channels. Indeed there is one IQ matrix per channel. Each machine, there is a server per channel as shown in Figure 2, implements a parallel computing scheme based on pulses. In fact, given the limited computational power, the time simulation of a single pulse is not feasible, because the pulse lasts few μs. For this reason the burst β to be simulated is partitioned into sets of pulses:

$$\beta = \{\{\rho_1, \rho_2, ..., \rho_{N1}\}, \{\rho_{N1+1}, ..., \rho_{N2}\}, ..., \{\rho_{Nk-1,...,} \rho_{Nk}\}\}.$$

The cardinality of β is equal to the number of processors available, and each element of $\{N_1, N_2, ..., N_k\}$ represents the number of pulses in the burst. The simulation for each of the pulses subsets starts simultaneously on different CPUs, and this the key factor that allows the burst simulation on time.



Example 1: Pulse Parallelization

ISBN 978-88-903724-3-8

It is evident from Example 1 that even though the single pulse time is grater than the actual radar pulse time, exploiting more CPUs makes possible to simulate the burst in realtime.

6. ANTENNA SIMULATION

The implemented antenna is a four lobe electronic scan monopulse antenna (Sletten). All monopulse channels are considered: sum channel, elevation channel and azimuth difference channel. AESA radars can form multiple beams to scan the volume without mechanical steering. The gain patterns of the antenna have been sampled from the actual radar system, but a parametric gain pattern customization is possible as well. Given the azimuth and elevation pointing angles the antenna gain patters are extracted. In the former case, sampled gain patterns, it is just matter to retrieve table indexes and perform an interpolation if required. In the latter case, parametric gain pattern, there is the need to call a designed mathematical function that implements the gain pattern.

Of course sampled gain patters yield a more reliable antenna simulation, but they require a large amount of system memory to store runtime lookup tables. Parametric gain patterns are useful solely when a standard "data-package" of the radar is available to the simulation, and hence only global figures are known.

7. PERFORMANCE ANALYSIS (1 CHANNEL MOCK-UP)

7.1. SYSTEM OVERVIEW

A prototype of the test rig has been developed to investigate the feasibility of the project. The test setup is depicted in Figure 3. Workstation 1 implements the sum channel, while an emulator of the radar signal (SIP) and data processors (RDP) mimics the functionality of the devices under test. Having the control of the emulated radar processors enables to effectively measure the overall time performance of the system, including data transfer time, which is not negligible.

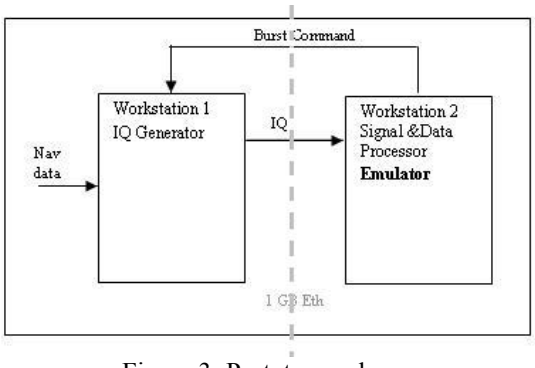


Figure 3: Prototype schema

The communication is carried out through an 1 GB Ethernet link (both directions of Figure 3). Final radar rig has got, as previously detailed, 10 Gbit optical links, i.e. the measured output throughput of the mock-up has got more band limitations than the target system.

At application level, a basic communication protocol has been devised:

- The IQ generator starts and waits for a burst command from the emulator.
- The emulator starts and sends a burst command to the IQ generator. The emulator waits for an acknowledge.
- The IQ generator receives the command and begins the processing, while sending the acknowledge
- If the emulator doesn't receive the acknowledge before the time out (half the simulation period), it throws an exception, otherwise it puts itself in an idle state, waiting for the IQ generator response.
- While the processing is over, the IQ generator sends the time-range matrix to the emulator that collects the data. If no exceptions occurred, a new cycle can start.

The most demanding modes of operation are Medium PRF and High PRF, the latter for the amount of data to be processed, the former for the data rates required.

Table 2: Performance estimation

1 CHANNEL	HPRF	MPRF
Data	175 kB	50 kB
Burst Time	10 ms	2.5 ms
Predicted proc. time	5 ms	2 ms
Comm. time	5 ms	0.5 ms
Data rate	35 MB/s	100 MB/s

To obtain a quantitative benchmark of the system, it is necessary to measure the global time frame, which includes data IO for the stimuli, processing time and data IO for the result. This global time must be less than the prescribed frame, with a reasonable safety margin. Time measures of the pure processing time are also recommended, since this gives a metric for the computing efficiency and the input output time performance. This information can be used to assess the level of parallelism reached for each machine, changing the number of concurrent threads. Increasing the number of parallel threads reduces the overall computing time.

7.2. HARDWARE

The mock-up of the test rig is composed of two servers, one running the IQ sum channel simulation software, the other the radar processors Emulation software. The characteristics of these machines are briefed in Table 3.

Table 3: Server characteristics

ISBN 978-88-903724-3-8

Processor	X 2	Intel Xeon processor E55303, 2.40 GHz, 8 MB cache, 1066 MHz memory, Quad-Core
RAM	X 2	8GByte of DDR3 RAM
Network adapter	X 2	NetXtreme BCM5764M Gigabit Ethernet PCIe1 GBit
Operating system	X 1 X 1	QNX 6.5 Neutrino SUSE Linux Enterprise Server 11, RT

Standard serial ATA hard drives are installed into the machines for a total storage capacity of 250GB for each channel.

7.3. HPRF

HPRF modes provide long range air-to-air detection of closing targets w.r.t. the radar. Target signals are well separated from surface clutter returns so that target detection is noise-limited. In this mode the radar has got a pulse repetition frequency so high that the non ambiguous range is very short. This means that the number of range bins is limited in number, while there are many pulses in every burst. It follows that the time-range matrix has got more columns than rows. Using figures of Table 2, it is clear that this mode has the greatest number of elements of the matrix, and consequently the longest processing time, but it has also a large time frame to achieve its task.

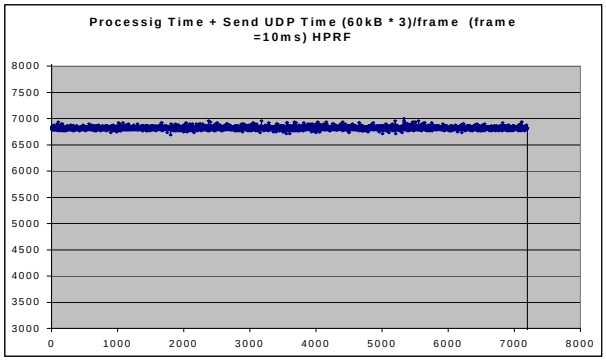


Figure 4: Global time frame measurements vs. burst number, HPRF

Figure 4 shows that, on the average, the global time frame is less than 7 ms, and hence yields a 30% margin. An interesting remark can be made comparing the above figure with the results figure 5. Indeed it shows the processing time. It takes 4ms, hence the input output operations occupy 3ms, which is equal to the 75% of the active modeling. The IO operation are expected to be faster on a 10 Gbit optical link, so there is further room for improvement of the bearable time frame.

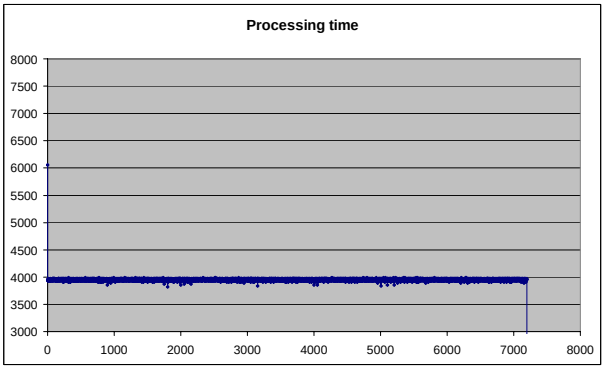


Figure 5: Processing measurements versus burst number, HPRF

7.4. MPRF COMBAT

This is an air-to-air detection mode designed to rapidly acquire targets at close range before transition from TWS to STT operation. The measurements for MPRF COMBAT shows that the global I/O time is about 1 ms, which is more than one third of the prescribed time frame of 2.5 ms. The processing itself for the IQ data generation is about 1.6 ms.

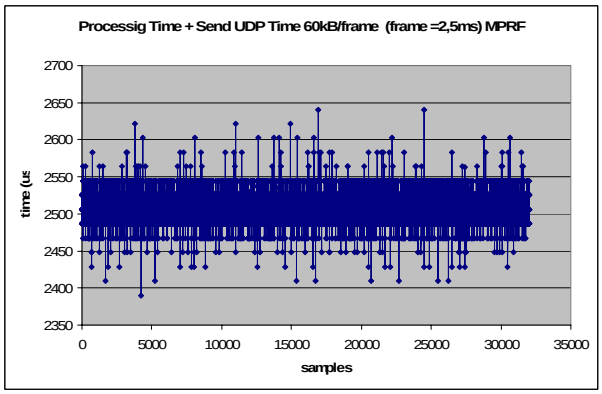


Figure 6: Global time frame measurements vs. burst number, MPRF

From the empirical evidence, there is an additional delay (over the 2.5 ms). It must be considered though that these tests were performed with a 1GBit Ethernet (125 Mbyte maximum theoretical rate), and that the time required for the IO amounts to 40% of the whole time frame. With a 10 GB (1.25 Gbyte maximum theoretical rate) this aspect will improve more than significantly.

For example if the band doubles, the weight of the transmission time will be 20 % (500 us) possibly yielding a cycle time of 2 ms.

ISBN 978-88-903724-3-8

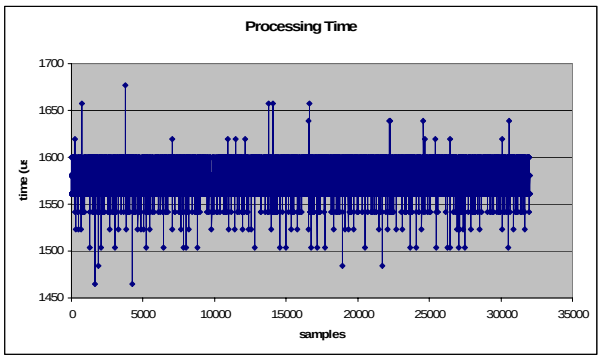


Figure 7: Processing measurements versus burst number, MPRF

8. CONCLUSIONS

Performed tests lead to the conclusion that HPRF bursts (the worst case for data I/O rate) can be successfully handled by the realtime IQ Data Generator with the identified solution. Having theoretical dwell periods of 10 ms, 4 ms shall be used for IQ data processing, 2,9 shall be used for data I/O. The idle time is 3.1 ms (31 %). The usage of 10 Gbit Ethernet boards will improve more and more these numbers.

The AESA RDP Emulator did not registered a single overrun, proving that the comprehensive I/O mechanism (data exchange protocol and synchronization) is finely working, as per hypothesis.

Tests run so far lead to the conclusion that MPRF COMBAT bursts (the worst case for computation rate) can be successfully computed by the realtime IQ Data Generator with the identified solution. It remains an uncertainty on the time needed for data I/O. Having theoretical dwell periods of 2.5 ms, 1.6 ms shall be used for IQ data processing, 1 shall be used for data I/O. Spurious overruns of 0.5 - 1.0 ms have been measured both on the IQ Data Generator and on the AESA RDP Emulator. It must be considered though that these tests were performed with a 1 Gbit Ethernet, and that the time required for the IO amounts to 40% of the whole time frame. With a 10 Gbit this aspect will improve more than significantly.

Using a machine with more that 8 cores and 10 Gbit boards, it is quite likely to be able to handle also this case as per requirements.

The usage of the simulated IQ Data Generator allows the Signal Processing and Radar Tracking functions of the real AESA Radar to be tested against simulated real time targets in a Lab Environment in a consistent and repeatable manner. Very often such evaluation with flight trials are very complicated, expensive and difficult to be repeated.

REFERENCES

Picardi, G., 1988. *Elaborazione del segnale radar, metodologie ed applicazioni*, 31-41

Asoke K. Bhattacharyya D.L. Sengupta: Radar Cross Section Analysis & Control

Carlyle J. Sletten, 2010: Reflector and Lens Antennas Analysis and design using personal computers

QNX Software System Ltd, 2004 : QNX Neutrino RTOS V6.3 system architecture

AUTHORS BIOGRAPHY

Romolo Gordini graduated in Business Engineering at the University of Udine in 1997. He joined SELEX Galileo in 2001 where he has been working in the Radar Simulation, that he led as a Project Leader since 2004. In 2005 he has been assigned as Head of Sensors Simulation Department in Ronchi dei Legionari, with design and integration responsibility of sensors simulation for all the simulation programmes of the Line of Business, such as the CAPTOR radar on the Eurofighter Typhoon, the Tornado Nose radar and the APS-784 radar on the AW EH101 helicopter.

He is one of the 2 authors of the realtime radar simulation toolkit MARS (Multimode Airborne Radar Simulator).

Michele Giorgiutti graduated in Electronic Engineering at the University of Udine in 2009, with the thesis “Discrete geometrical formulations for the solution of 2D and 3D electromagnetic problems”. He also worked as collaborator for the DIEGM department of the University of Udine. in the same year. During that period he developed C/C++ and Fortran software for the numerical solution of three dimensional eddy currents problems employing a geometrical method called “Cell method”. Since June 2009 he has been working for SELEX Galileo, in Ronchi dei Legionari, Italy, where he is a System Engineer for the Sensor simulation & Database Department. He contributed to the analysis and design of realtime interfaces between avionic equipments and sensor simulators, running on commercial servers.

ISBN 978-88-903724-3-8