An Integrated and Cost-Effective Simulation Tool for GNSS Space Receiver Algorithms Development

João S. Silva, Hugo D. Lopes, Tiago R. Peres, José M. Vasconcelos, Maria M. Coimbra, Pedro Freire, *DEIMOS Engenharia*
Pedro Palomo, Juan Pérez, José A. Pulido, *DEIMOS Space*
Alberto Garcia, Josep Roselló, *European Space Agency*

BIOGRAPHIES

João S. Silva received his Aerospace Engineering degree from Instituto Superior Técnico, Portugal, in 2004. He worked as a Research Engineer in Instituto de Telecomunicações from 2003 to 2005, when he joined DEIMOS Engenharia, where he is currently working as a Project Manager in the GNSS Technologies Division. His areas of expertise focus on GNSS, including signal processing algorithms, systems engineering, and simulation tools.

Hugo D. Lopes received his Double-Master degree in Aerospace Engineering from Delft University of Technology, The Netherlands, and Instituto Superior Técnico, Portugal, in 2011. He is currently a Project Engineer in the GNSS Technologies Division of DEIMOS Engenharia, where he started working in December of 2011.

Tiago R. Peres received his MSc degree in Aerospace Engineering from Instituto Superior Técnico, Portugal, in 2008. He joined Deimos Engenharia in 2008, where he is currently working as a Project Engineer in the GNSS Business Unit. Since then, he has been involved in GNSS Signal Tracking Algorithms, Multipath Mitigation Techniques, Software Simulation Tools, Digital Signal Processing Using FPGAs and GNSS-Reflectometry.

José M. Vasconcelos José F. Vasconcelos received the Licenciatura and the Ph.D. degrees in electrical engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 2003 and 2010, respectively. Since 2009, he has been a Project Engineer with DEIMOS Engenharia’s Flight Systems Business Unit. His research interests include nonlinear observers, inertial navigation systems, rendezvous, and robust control.

Maria M. Coimbra received her Aerospace Engineering degree from Instituto Superior Técnico, Portugal, in 2012. She is currently a Project Engineer in the GNSS Technologies Division of DEIMOS Engenharia, where she started working in September of 2012. She has been actively involved in EGNOS v3 and ionosphere modeling.

Pedro Freire received his Aerospace Engineering degree from Instituto Superior Técnico, Portugal, in 2001 and since then has been working in receiver and navigation related applications and technology. Since 2005, he has been working at DEIMOS Engenharia as responsible for the GNSS Technologies Division.

Pedro Palomo received his MSc in Computer Science by the UPM , Spain, in 2002. He is currently a Project Manager in the Embedded Systems Division of DEIMOS Space, where he started working in 2002. He is involved in activities related to safety‑critical SW for real-time systems and real-time test benches.

Juan Pérez received his Bachelor+Master degree in Electronics, Communications and Computer Engineering from University of Alcalá, Spain, in 2012. He is currently a Project Engineer in the Embedded Systems Division of DEIMOS Space, where he started working in 2010.

José Antonio Pulido was graded with a MsC and a PhD in Telecommunications Engineering by the UPM where he specialized in safety-critical software development. After this, he joined DEIMOS Space, where he develops algorithms in different GNSS navigation projects as well as safety-critical SW for real-time systems, such as Galileo Ground Mission Segment.

Alberto Garcia Alberto Garcia-Rodriguez works in the radio navigation section in ESA. He is involved in activities related to GNSS space receivers and applications. He worked previously at GMV (Spain) for several navigation projects. He has an MSc in Telecommunications Engineering from the T.U. of Madrid (1992).

Josep Roselló graduated as Telecommunications Engineer from Polytechnic University of Catalonia in Barcelona and, in 1998, completed an MBA in Rotterdam. He started as engineer at ESA ESTEC in 1993 in the Data System Division and, since 2007, has been with the Future Missions Division of the Earth Observation Directorate at ESTEC. His current responsibilities include internal research and technical management of industrial contracts, as well as inter-agency co-ordination.

ABSTRACT

AGGA-4 is a GNSS baseband signal processor for space applications, developed under ESA contract, currently under production.

In the scope of the GSTST (GNSS Dynamics Simulator and AGGA-4 Test and Simulation Tool) project (funded by ESA, contract number 16831/03/NL/FF), DEIMOS Engenharia is developing an integrated simulation and testing tool for the design and realistic analysis and test of GNSS signal processing and navigation algorithms for AGGA-4-based GNSS receivers. This tool will provide a relatively inexpensive solution (when compared to currently available hardware-based solutions) for the simulation of realistic GNSS observables and measurements (as the software part of the receivers would see them) as well as of the AGGA-4 programming registers, allowing AGGA-4-targeted software to be tested without the need for the AGGA-4 chip or complex and expensive hardware setups.

This paper describes the motivation, architecture, and functionalities of the GSTST and presents a couple of application examples (navigation for LEO missions and tracking loop closure).

INTRODUCTION

Background

Amongst the currently most used navigation systems are the Global Navigation Satellite Systems (GNSS) as the Global Positioning System (GPS) and the future Galileo system, currently under development by the European Union (EU) and the European Space Agency (ESA).

GNSS is already being used in space missions (e.g. GOCE, Swarm, Sentinel, and MetOp, among others), not only as a navigation sensor (either for orbit determination or relative navigation) but also as a science instrument (e.g. for altimetry, global geodesy, Radio Occultation and GNSS Reflectometry applications). The AGGA-4 (Advanced GPS and Galileo ASIC) is a baseband GNSS digital signal processor targeted for space applications, being developed under ESA contract, and represents an important step towards the miniaturisation of the next generation of GNSS space instruments [1]. Amongst its main constituents are a Digital Signal Processing (DSP) core (featuring very high-speed functionalities) and a LEON2-FT microprocessor.

AGGA-4 will support the processing of all current and future GPS civil signals and Galileo Open Service signals (also supporting legacy GLONASS signals and modernized GLONASS and Beidou signals) and is expected to have a very significant impact on a wide range of space applications.

Motivation

Space GNSS receivers typically operate under various and often complex operational scenarios, requiring trade-off analysis early in the design stage and also in later development stages, involving extensive test campaigns and considerable resource usage (both in terms of human resources and equipment).

Furthermore, advanced navigation system architectures and algorithms often require access to and control of signal processing stage parameters. Architectures in which the signal processing algorithms, the navigation functions, external sensor measurements and/or dynamics information are brought together (enabling the feedback of aiding signals to the GNSS receiver's tracking loops) require access to internal receiver signals and configuration parameters.

Although the AGGA-4 prototype is not expected to be available until 2014 its pinout is already known and the development activities of AGGA-4-based GNSS receivers can already start. To support this development – and as with any other development of complex GNSS architectures and algorithms – it is important to have access to Ground Support Equipment (GSE) capable of simulating realistic environments and generating representative multi-GNSS scenarios as the receiver would experience them. This is particularly critical when algorithms for the processing of currently scarce or unavailable signals (as modernized GPS signals and future Galileo signals) need to be simulated, and before key elements, as the AGGA-4, are integrated into receiver boards.

Powerful (highly realistic) hardware simulators exist (as Spirent's multi-GNSS simulation systems) that combine the dynamics of multiple GNSS satellites and the electrical characteristics of currently available and future GNSS signals to simulate the RF signal that would be receiver by a GNSS receiver's antenna. However, their cost and complex setup (in terms of both required equipment and experience) is a considerable limitation for its use. There are other simulation tools that enable the analysis and test of receiver architectures and navigation algorithms (as the GRANADA family tools [2]), facilitating and accelerating the development and validation process. However, due to the complexity and specificity of the AGGA-4, there is currently no commercially available tool which supports the modelling or simulation of the AGGA-4.

The availability of a tool that enables the analysis and test of Digital Signal Processing (DSP) and navigation algorithms for an AGGA-4-based receiver without the need for complex and expensive hardware setups (as GNSS constellation simulators and signal generators or AGGA-4 development boards) while still keeping a high level of realism would be useful for various applications, ranging from R&D to on-board SW development and verification. Such a tool would enable a user/researcher to develop and test navigation SW for the AGGA-4 without having to own an AGGA-4-based receiver board or feed it with RF or digitized signals. Such capability would be a very important asset given the coming multiple constellations and signals, allowing realistic, easy, and early testing and/or prototyping of new algorithms for GNSS receivers operating in complex space environments.

GSTST ARCHITECTURE AND FUNCTIONALITIES

In the scope of the GSTST (GNSS Dynamics Simulator and AGGA-4 Test and Simulation Tool) project (funded by ESA, contract number 4000105894/NL/FF/fk), DEIMOS Engenharia is developing an integrated simulation and testing tool for the design and realistic analysis and test of GNSS signal tracking and navigation algorithms for AGGA-4-based GNSS receivers. This tool provides a relatively inexpensive solution (when compared to currently available hardware-based solutions) for the realistic simulation of AGGA-4 observables and measurements (as the software part of the receivers would see them) as well as AGGA-4 programming registers (and their influence on the HW part of the receiver).

The GSTST supports all current and modernized GPS civil signals as well as all open service Galileo signals (as described further below) and is representative of a subset of relevant AGGA-4 features and functionalities (e.g. multi-antenna support, Aiding Unit, among others).

The GSTST consists of two main modules, as (illustrated in Figure 1): the Environment Simulator module, which simulates the test environment/scenario, and the GNSS Receiver Simulator module, which simulates relevant AGGA-4 hardware cores (Input Modules, GNSS Core and Processor), allowing receiver-targeted software to be tested.

The Environment Simulator module is further divided into the Reference Dynamics Simulator, which generates reference values for the GNSS–S/C geometry (position, velocity, attitude, and angular rates, as well as geometric ranges and range rates), supporting up to 4 antennas, and the Propagation Channel Model, which calculates C/N_0 at the receiver input (taking into account transmitter and receiver characteristics and geometric effects) and computes ionospheric and multipath error models.

The GNSS Receiver Simulator models relevant AGGA-4 DSP Cores, and allows user-developed software, the Software Under Test (SWUT), to be run in a realistic emulator of the AGGA-4's LEON-2 processor, the SWUT Test-Bed. To bridge the gap between the (higher-level) SWUT and the AGGA-4 DSP Cores model, the GNSS Receiver Simulator also includes a Lower-Level SW Simulator which implements/simulates the required GNSS DSP algorithms (as acquisition, tracking loops, bump-jumping, lock detection, and measurement generation) for the production of inputs for the SWUT.

The separation between the algorithms that are simulated in the Lower-Level SW Simulator and those that are implemented in the user-developed SWUT has a certain degree of flexibility, allowing both tracking and/or navigation algorithms to be run in the LEON-2 Processor Emulator. Additionally, the GSTST also supports feedback from the SWUT to the AGGA-4 DSP Cores Model, enabling the simulation of receiver aiding.

With the exception of the SWUT Test-Bed, which is based on a Commercial-Off-The-Shelf (COTS) processor emulator all modules are developed in MATLAB/Simulink® (based on the commercially available GRANADA GNSS Blockset, formerly GRANADA FCM Blockset [3], developed by DEIMOS Engenharia).

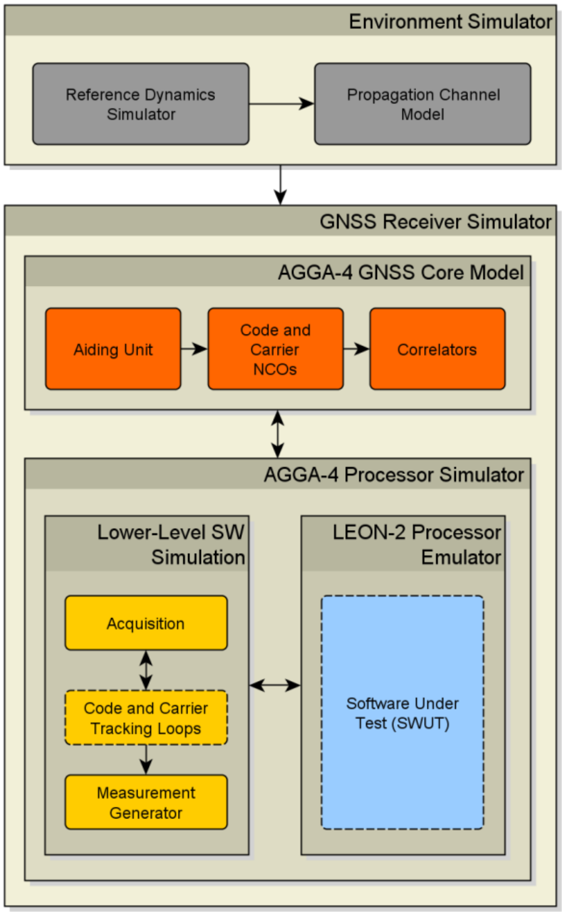


Figure 1: High-level GSTST Architecture

The main architectural blocks and functionalities of the GSTST are described in the next sections. For additional details, please refer to [11].

Reference Dynamics Simulator

The Reference Dynamics Simulator generates the GNSS satellites positions and velocities (based on an ephemeris file), the receiver's position, velocity (based on actual orbit propagation model), attitude and angular rates, as well as the corresponding relative ranges, range rates (based on the positions and velocities mentioned before), and relative orientation.

The Reference Dynamics Simulator module includes:

- A Receiver Dynamics module, which outputs spacecraft position, velocity, attitude and angular rate based on the propagated orbit and attitude law;
- An Antenna Dynamics module, which outputs the position and velocity of each receiver's antenna based on the position, velocity and attitude of the receiver and on the offset of the antennas relative to the center of gravity (COG) of the spacecraft;
- A GNSS Satellites Dynamics module, which outputs the GNSS satellite positions and velocities based on satellite ephemeris and receiver COG position;
- A Relative Dynamics module, which outputs satellites-receiver ranges and range rates, taking into account the GNSS satellites positions and velocities and the position and velocity of the receiver, as well as the relative orientations between the transmitter and receiver antennas.

The GNSS satellites' orbits are computed based on their orbital parameters (via an ephemeris file), while the S/C orbit is numerically propagated (starting from an initial state vector based on a Two-Line-Element file) taking into account gravity perturbations, Sun and Moon influence, solar radiation pressure and atmospheric drag models.

Propagation Channel Model

The Propagation Channel Model includes:

- A Geometric Visibility module, which determines the visibility of the GNSS satellites with respect to the receiver, along with ionosphere intersection points in the Line-Of-Sight (LOS) path, if applicable (as illustrated in Figure 2);
- A C/N0 Computation module, which calculates the C/N0 at the receiver input taking into account transmitter characteristics, propagation effects, geometric effects, and receiver characteristics;
- A Ionosphere Model, which computes the Total Electron Content (TEC) along the LOS path;
- A Multipath Model, which generates multipath induced perturbations to the code and carrier tracking loops based on statistical models.

For the C/N_0 computation, the propagation Channel Model takes into account the, the Equivalent Isotropically Radiated Power (EIRP) of the transmitted GNSS signals, the receiver antennas radiation pattern, and the transmitter-receiver geometry (visibility, range, and relative orientation).

The TEC computation is based on the NeQuick Ionosphere model [6] (thus taking into account the electron density profile of the Ionosphere, unlike 2-D Ionospheric models), on the receiver position (and intersection of the LOS path with the Ionosphere) and on the simulated date and time.

The multipath perturbations are generated based on independent stationary Gauss-Markov processes for the pseudo-range and carrier phase, defined by their steady-state standard deviations and autocorrelation times.

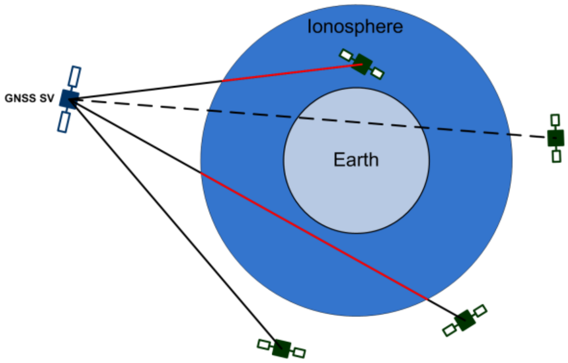


Figure 2: Different visibility and ionospheric intersection scenarios (scale exaggerated for illustration purposes)

AGGA-4 DSP Cores Model

The AGGA-4 DSP Cores Model includes:

- An Aiding Unit, in charge of integrating frequency and frequency rate estimates and adding them to the residual NCO rates (estimated by the tracking loops);
- Code and Carrier NCOs, which will integrate the rate estimates output by the Aiding Unit to produce code and carrier phase estimates;
- A Correlator Outputs Model, which models the AGGA-4 correlator outputs – real and imaginary Early-Early (EE), Early (E), Punctual (or Prompt, P), Late (L) and Late-Late (LL) correlator outputs – based on the correlator spacing, on the local carrier and code phase (output by the Code and Carrier NCOs), on the geometric and propagation characteristics of the signals (geometric ranges, visibility, ionospheric delay, C/N0, and multipath errors), and on GNSS satellites and receiver clock errors (for which models are also included).

The Correlator Outputs Model takes into account not only the characteristics of the AGGA 4's Correlator Unit and the influence of code and carrier estimation errors, thermal noise, and integration time on its outputs, but also various other effects, related with other AGGA-4 modules, external receiver components or environment characteristics (e.g. pre-correlation bandwidth, multipath, ionosphere, and GNSS satellite and receiver clocks). The Correlator Outputs Model is based on the FCM block of the GRANADA GNSS Blockset [3], which models the outputs of a GNSS receiver's correlators at the integration rate, precluding the need for simulation rates representative of the sampling frequency and thus considerably increasing simulation speed.

The Aiding Unit, Code and Carrier NCOs and Correlator Outputs Model represent the hardware part of the tracking loops, whose software part is implemented AGGA-4 Processor Simulator (either in the Lower-Level SW Simulator or in the SWUT).

The AGGA-4 Integration Epoch (IE), Long epoch (LE) and Measurement Epoch (ME) signals, which control the resetting of the integrators, the latching of observables, and the interruption of the processor, are also generated.

Lower-Level SW Simulator

The Lower-Level SW Simulator module includes the following sub-modules:

- An Acquisition Simulator, which simulates the behavior of the acquisition algorithm (providing Doppler and code phase estimates to initialize the tracking loops);
- Tracking Loops, which model the receiver code and carrier tracking algorithms, estimating code and carrier rate inputs to the NCOs based on the correlator outputs;
- A Measurement Generation module, which generates receiver measurements based on the receiver observables.

The Acquisition Simulator module simulates the behavior (in terms of acquisition time) of a Parallel Frequency Search (PFS) acquisition algorithm, which can be implemented in the AGGA-4 using its FFT Core.

The tracking loops include Code and Carrier Error Discriminators and Loop Filters, Lock Detectors (which monitor the tracking loops' lock status and decide what measurements may or may not be used and what GNSS satellites must be [re-]acquired, if visible, by the Acquisition Simulator) and a Bump-Jumping module, (which detects and corrects false locks for BOC modulated signals).

The Tracking Loops module of the Lower-Level SW Simulator may be replaced by software running on the SWUT Test-Bed. In such case, the remaining (Acquisition Simulator and Measurement Generator) modules are still implemented in the Lower-Level SW Simulator, although using the SWUT outputs instead.

The Measurement Generation module computes pseudorange, carrier phase and pseudorange rate measurements based on the input code phase, carrier phase and carrier rate estimates, respectively (from the tracking loop filters and NCOs). In addition, it also simulates integer ambiguity resolution and generates both ambiguous and unambiguous carrier phase measurements as well as differential phase measurements (in the case of multi-antenna simulations).

SWUT Test-Bed

The validation of user-developed software in flight-representative hardware is crucial for its reliable evaluation, allowing developers to accurately analyze and profile it in order to have a representative idea of its performance and resource requirements, hence allowing an enhanced implementation plan in a higher-level software project. The SWUT Test-Bed allows such user-developed software, the SWUT (which must be developed in C programming language and compiled for the LEON2 processor), to be run as if it was running on the AGGA-4's LEON2-FT processor, having access to AGGA-4 programming registers and observables, as well as measurements based on those observables.

The SWUT Test-Bed, on which the SWUT will run, implements a realistic and representative environment based on a COTS (Commercial Off-The-Shelf) processor emulator, the TSIM ERC32/LEON Simulator from Aeroflex [5], which is able to emulate the AGGA 4's LEON-2 processor on a standard PC.

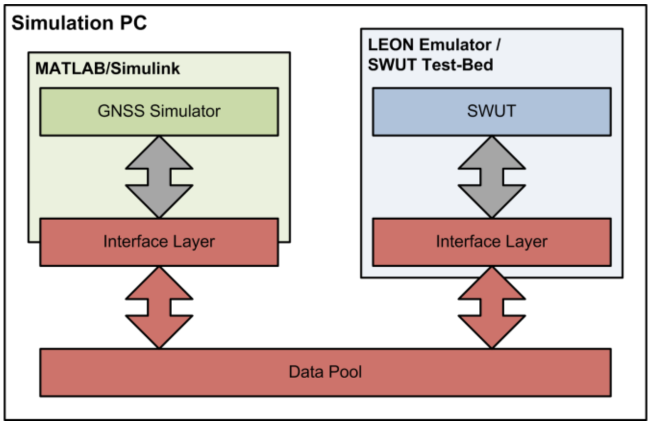


Figure 3: GSTST-SWUT interface and data pool

Figure 3 illustrates the environment on which the SWUT runs as well as its interface with the GSTST. The simulations run in a single PC, which hosts both the MATLAB/Simulink part and the LEON Emulator. This approach provides a realistic, low-cost, and highly portable solution that precludes the need for specific and complex hardware setups. Simulation data (organized into several data structures, as described further below) is exchanged between the GSTST and the SWUT via a Data Pool, which holds, among other data, reference values, ephemeris data, observables, measurements, and feedback signals.

Supported GNSS Signals

The GPS and Galileo systems transmit (or will transmit in the future) navigation signals over numerous bands. The modernized GPS shall transmit signals over the L1, L2 and L5 bands, while the Galileo system shall use the E1, E6, and E5 bands. All GPS and Galileo signals are Code Division Multiple Access (CDMA) spread spectrum signals resulting from the modulation of an RF carrier with a Pseudo-Random Noise (PRN) sequence (different for each satellite), a data bit stream, and, in some cases, a sub-carrier and/or a secondary code.

Hardware characteristics of the AGGA-4 limit the supported GNSS signals, which are listed in [1]. The GSTST will support all GPS and Galileo public signals (GPS civil signals/channels and Galileo Open Service signals/channels) supported by the AGGA-4, listed in Table 1. The characteristics of these bands and signals are provided in [7], [8], [9], and [10].

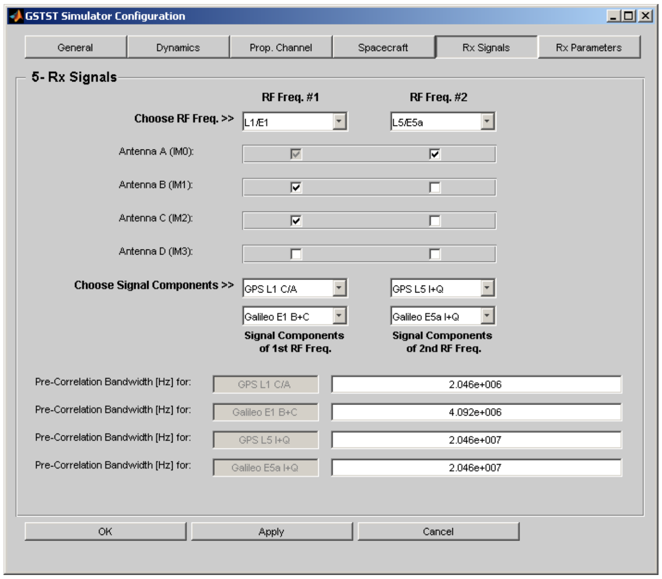
Table 1: Supported GNSS Signals

System	Band	Signals/Channel(s)
GPS	L1	C/A
		C (C_D, C_P)
	L2	C (CM,CL)
	L5	I, Q
Galileo	E1	B, C
	E5a	I, Q
	E5b	I, Q

GSTST OPERATION

Configuration

All the configuration parameters required to setup a simulation are defined in an XML Configuration file. A sub-set of configuration parameters is accessible through a Graphical User Interface (GUI), for convenience (see Figure 4). The remaining parameters can be modified by editing the XML configuration file (using a standard XML or text editor).



	RF Freq. #1	RF Freq. #2
Choose RF Freq. >>	L1/E1	L5/E5a
Antenna A (IM0):	☑	☑
Antenna B (IM1):	☑	☐
Antenna C (IM2):	☑	☐
Antenna D (IM3):	☐	☐
Choose Signal Components >>	GPS L1 C/A Galileo E1 B+C	GPS L5 I+Q Galileo E5a I+Q
	Signal Components of 1st RF Freq.	Signal Components of 2nd RF Freq.

Pre-Correlation Bandwidth [Hz] for: GPS L1 C/A 2.046e+006
Pre-Correlation Bandwidth [Hz] for: Galileo E1 B+C 4.092e+006
Pre-Correlation Bandwidth [Hz] for: GPS L5 I+Q 2.046e+007
Pre-Correlation Bandwidth [Hz] for: Galileo E5a I+Q 2.046e+007

Figure 4: GSTST's graphical user interface (signal selection tab)

The configuration parameters can be divided into four main groups: simulation setup parameters, environment characteristics, user spacecraft characteristics, and receiver configuration parameters, as described below.

The simulation setup parameters include general simulation parameters (such as the simulation duration, time step, and constant definitions) as well as the parameters related with the nature of the SWUT to be tested (allowing the user to enable/disable the SWUT or the Simulink tracking loops).

The environment characteristics (Dynamics and Prop. Channel tabs in Figure 4) include GNSS constellation(s) information (as the path to the ephemeris file containing their orbital parameters, satellite selection filter, and GNSS SV clock error parameters), propagation channel characteristics (as transmitter and receiver antenna patterns, RF front-end noise figure, Ionospheric model parameters, Satellite Clock errors, and multipath error model parameters), and allow to independently enable/disable receiver dynamics perturbations (as gravity perturbations, atmospheric drag, and solar radiation pressure).

The user spacecraft characteristics describe the S/C's physical characteristics (as mass, cross section, and drag coefficient), the different antenna locations and relative orientations (up to 4 independent antennas), and the S/C dynamics (initial orbital parameters and attitude law).

The receiver configuration parameters include the bands and signals to be processed and their assignment to each antenna, the RF front-end(s) pre-correlation bandwidths, as well as several HW and SW configuration parameters (as integration times, correlators' spacing, tracking loop discriminators, filter orders and bandwidths, lock detection thresholds, measurement period, clock error model parameters, satellite to AGGA-4 assignment criteria, and specific AGGA-4 HW programming parameters, among others).

Since the SWUT is user-dependent, the SWUT configuration (through a configuration file or any other applicable means) is left at the responsibility of the user (SWUT related configuration parameters are not available via the GUI, except for the enabling/disabling of the SWUT). However, GSTST configuration parameters may be passed to the SWUT via shared data structures in the Data Pool.

Operation Modes

The GSTST supports two main modes of simulation, enabling the test of two main types of SWUT:

- Simulink Tracking Loops mode: this mode (illustrated in Figure 5) targets SWUTs, which need only to concentrate on the measurements generated by the GSTST (simulated or post-processed ME observables, as code and carrier phases, raw and/or corrected pseudoranges, and unambiguous and/or differential carrier phases) and, optionally, provide feedback signals to aid the tracking loops (which are implemented in Simulink). These are usually SWUTs which implement navigation algorithms (as orbit determination filters or attitude determination filters);
- SWUT Tracking Loops mode: this mode (illustrated in Figure 6) targets SWUTs which gather observables (simulated IE observables as correlator outputs and code and carrier NCO rates and phases) and require control of the Aiding Unit and NCOs (by writing to the simulated programming registers). These are usually SWUTs which implement tracking loop closure algorithms, which replace (bypass) the tracking loops implemented in Simulink.

Additionally, other configurations are possible, for example:

- No SWUT: the GSTST is used only to produce observables and measurements:
- Tracking loops implemented in both Simulink and the SWUT: this allows early verification of the tracking algorithms in the SWUT by comparing their outputs with the ones being computed in the Simulink part;
- Tracking plus navigation in the SWUT: both tracking loop closure and navigation algorithms can be implemented in the SWUT at the same time, each running at the applicable event (IE or LE for the tracking loops and ME for the navigation part). In this mode, the navigation algorithms consume measurements produced by the Measurement Generator based on the outputs of the SWUT tracking loops.

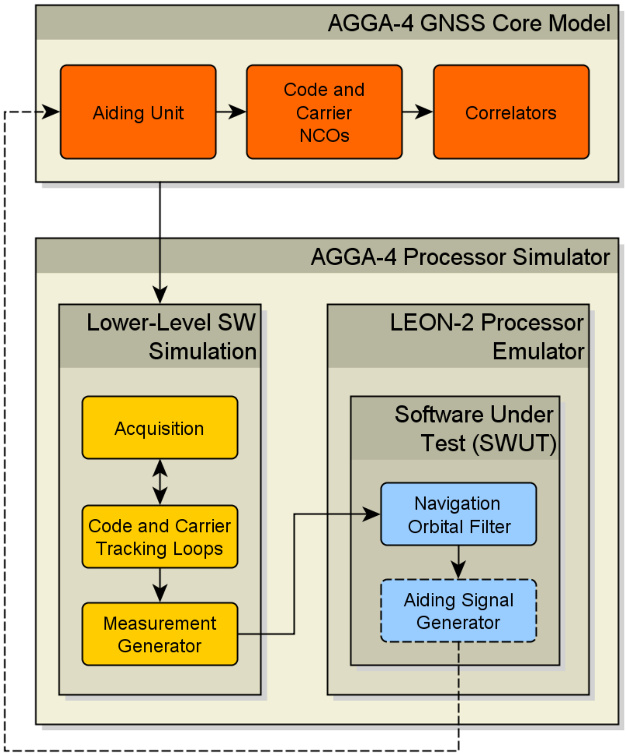


Figure 5: Simulink Tracking Loops mode

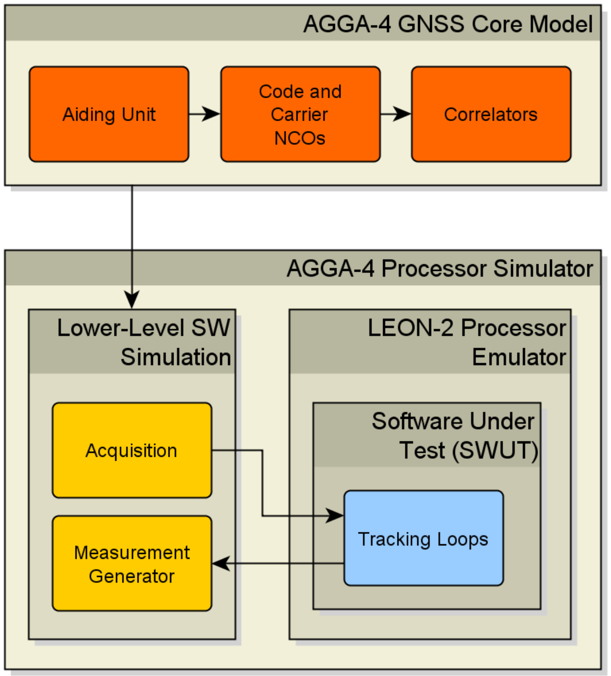


Figure 6: SWUT Tracking Loops mode

Upgradeability

In addition to the available operation modes, the MATLAB/Simulink environment allows the user to modify the simulator in order to easily adapt it to fit specific needs. For example, modifying the signals that are fed to the SWUT (and back) is a relatively simple task consisting on the selection of the relevant signals from already available buses.

In addition, the most relevant signals/buses can be saved to the MATLAB workspace by adding signal recording blocks available from the GSTST Simulink library. And if the available signals are not enough, adding test points is not much more complex than connecting a “Scope” or a “To Workspace” block to the relevant signals, as shown in Figure 7.

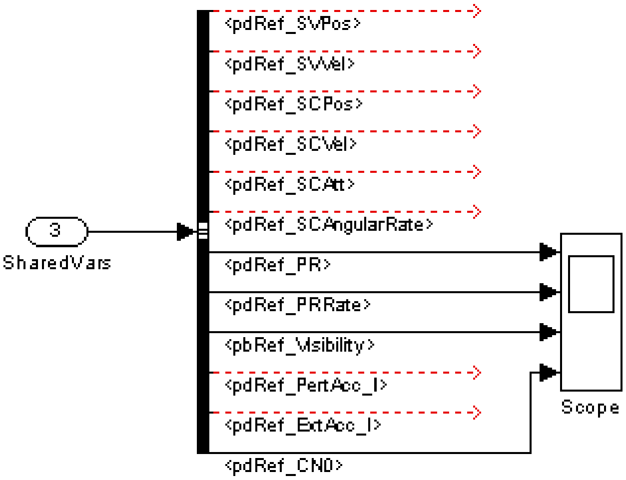


Figure 7: Example of adding a scope to GSTST to monitor selected signals

EXCHANGED DATA

The data exchanged between the GNSS Simulator and the SWUT Test-Bed can be sub-divided into the following categories:

- Hardware registers, which include AGGA-4 programming registers (listed in Table 2), IE and ME observables (listed in Table 3), and interrupt signals;

Shared GSTST variables, which include GNSS measurements (post-processed observables, listed in

- Table 4), ephemeris, status flags and other GSTST internal variables and outputs (including user-definable outputs);
- Shared SWUT variables, which include SWUT internal variables and outputs (including user-definable outputs), configuration parameters, control signals, and performance monitoring signals (for SWUT profiling);

Table 2: Accessible AGGA-4 programming registers

Name	Description
CarrSwFreq	Carrier NCO rate
CarrSwShift	Carrier NCO phase correction
CodeSwFreq	Code NCO Rate
CodeSwShift	Code NCO phase correction
NCOSettings[a]	Code and Carrier NCO settings related with the time application of the programmed rates and phase corrections
DelayLineCtrl[a]	Code Delay Line settings related with correlator spacing
CorrUnitCtrl[a]	Correlator Unit settings related with IE and ME observables type selection
IntCountCtrl	Integration Epoch and Long Epoch duration
AidingUnitCtrl[a]	Aiding Unit settings related with the enabling/disabling of the Code/Carrier Aiding Units and the aiding values application triggering
CarrAidFreq	Carrier NCO aiding rate
CarrAidAcc	Carrier NCO aiding acceleration
CodeAidFreq	Code NCO aiding rate
CodeAidAcc	Code NCO aiding acceleration
CarrSwFreq	Carrier NCO rate

[a] Partially simulated.

Table 3: Accessible AGGA-4 observables

Name	Description
IE Observables	
IE_IMT_LSW	The lower 32bits of the Instrument Measurement Time (IMT) Counter at IE.
IE_ValueEE_I, IE_ValueEE_Q	In-phase (I) and quadrature (Q) components of the EE, E, P, L, and LL correlator outputs at IE.
IE_ValueE_I, IE_ValueE_Q	
IE_ValueP_I, IE_ValueP_Q	
IE_ValueL_I, IE_ValueL_Q	
IE_ValueL_I, IE_ValueLL_Q	
IE_CodeFreq	Code NCO frequency value at IE.
IE_CarrFreq	Carrier NCO frequency value at IE.
IE_CarrObs/Phase	Carrier phase and carrier cycle count at IE.
IE_ContCount	Correlator Unit’s Continuous Counter value at IE.
IE_CodePhase	Code NCO phase at IE.
ME Observables	
ME_IMT_LSW	The lower 32bits of the Instrument Measurement Time (IMT) Counter at ME.
ME_CarrObs/Phase	Carrier phase and carrier cycle count at ME.
ME_IntCount	Correlator Unit’s Integration Counter value at ME.
ME_ContCount	Correlator Unit’s Continuous Counter value at ME.
ME_CodePhase	Code NCO phase at ME.

Table 4: Generated GNSS measurements

Name	Description
ME_PR	Pseudorange estimate at ME.
ME_PRCorrected	Corrected pseudorange estimate (without GNSS ephemeris, clock and ionospheric errors) at ME.
ME_CodeFreq	Code NCO frequency value at ME (pseudorange rate).
ME_CarrFreq	Carrier NCO frequency value at ME (doppler).
ME_CarrPhase_Amb	Ambiguous carrier phase estimate at ME.
ME_CarrPhase_Unamb	Unambiguous carrier phase valestimateue at ME.
ME_PR_sigma	Estimated pseudorange standard deviation.
ME_CarrFreq_sigma	Estimated carrier rate (Doppler) standard deviation.
ME_DiffCarrPhase_Amb	Ambiguous differential carrier phase estimate at ME.
ME_DiffCarrPhase_Unamb	Unambiguous carrier phase estimate at ME.
ME_CN0	C/N_0 estimate at ME.

APPLICATION EXAMPLES

The GSTST is currently being validated using navigation algorithms targeted for different space applications and environments (PVAT for LEO and PVT for GEO) as well as tracking algorithms, demonstrating its applicability. In the next sub-sections, some results of the applicability demonstration activities are presented.

Navigation

The functionality and performance of a Navigation Orbital Filter (NOF), developed for the purpose of GSTST verification (i.e. not meant for optimal performance) was verified for LEO and GEO scenarios by running the NOF in the GSTST's SWUT Test-Bed. The LEO scenario is more challenging in terms of receiver dynamics, while the GEO scenario is more challenging in terms of satellite visibility and received signal power.

The GOCE (Gravity Field and Steady-State Ocean Circulation Explorer) mission's orbit was used as reference LEO scenario (very low altitude, near-circular orbit with orbital height of approximately 270km). The position and velocity estimation errors are shown in Figure 8 and Figure 9, demonstrating the filter's correct functionality and performance. A GNSS outage was simulated for about 70 seconds, showing that the orbital propagator of the navigation algorithm is correctly estimating the position and velocity even in the absence of GNSS measurements (apart a slight drift over time due to unmodeled effects).

The position and velocity errors for the GEO scenario (for which NASA's GOES mission was used as reference orbit), are shown in Figure 10 and Figure 11.

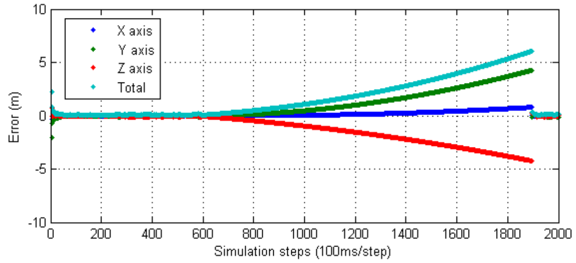


Figure 8: Position estimate error for LEO scenario

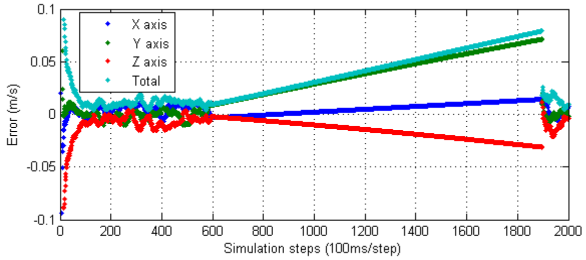


Figure 9: Velocity estimate error for LEO scenario

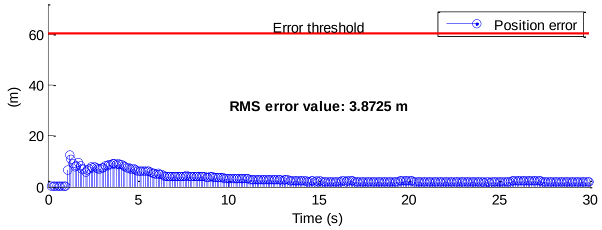


Figure 10: Position estimate error for GEO scenario

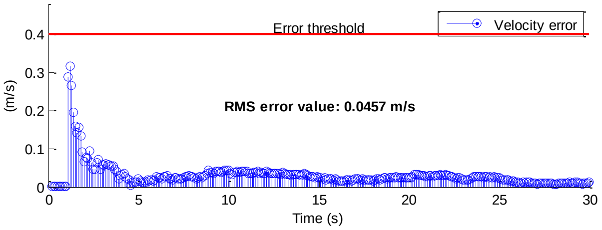


Figure 11: Velocity estimate error for GEO scenario

Tracking Loop Closure

The possibility of tracking loop closure via the SWUT (through access to AGGA-4 programming registers and observables) was verified by implementing and running tracking loop algorithms in the GSTST's SWUT Test-Bed. The same GEO scenario described above was used. The correlator outputs of channel #2 are shown in Figure 12 and the pseudorange, carrier phase and pseudorange rate measurements' errors for all used channels are shown in Figure 13, demonstrating the correct integration and functionality of the developed SW. The visible jumps in the errors for some of the channels are due to the loss of lock and re-acquisition phases, as can be inferred from Figure 14.

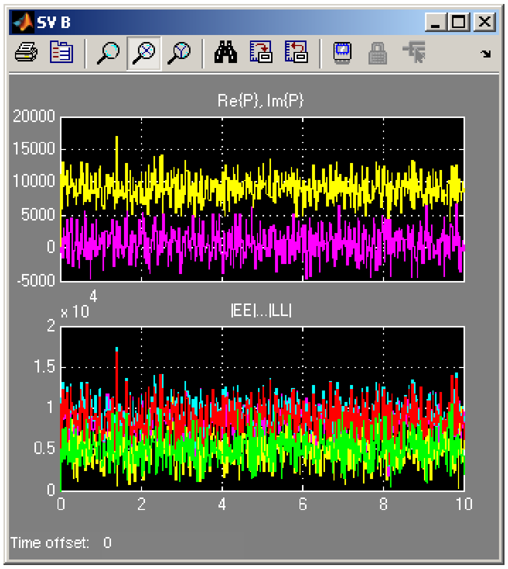


Figure 12: Correlator outputs for channel #2 (GEO scenario, C/N$_0$ of ~28.5dB)

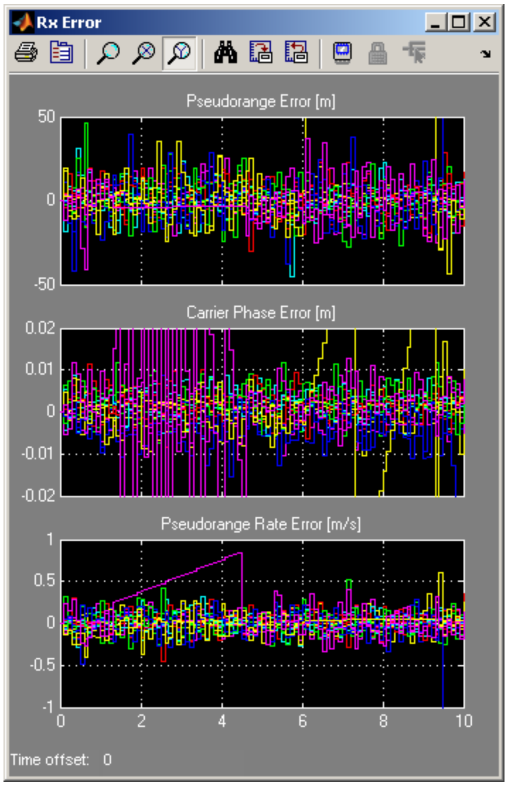


Figure 13: Measurement errors (GEO scenario, C/N$_0$ between 25dB-Hz and 41dB-Hz)

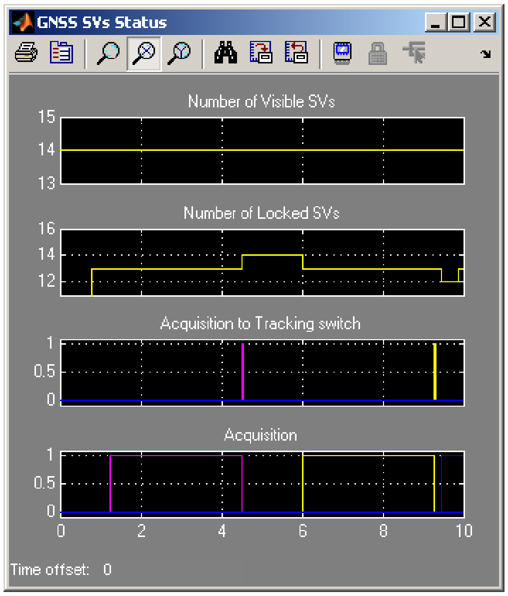


Figure 14: Visible and locked satellites and acquisition flags (GEO scenario)

Environment Manipulation and Fault Injection

An interesting feature (supported by the Simulink environment) is the ability of easily modify the nominal environment or forcing faults into the system under test. Figure 15 shows the correlator outputs for a particular AGGA-4 channel, which was processing Galileo E1 signal (assuming BOC(1,1) demodulation).

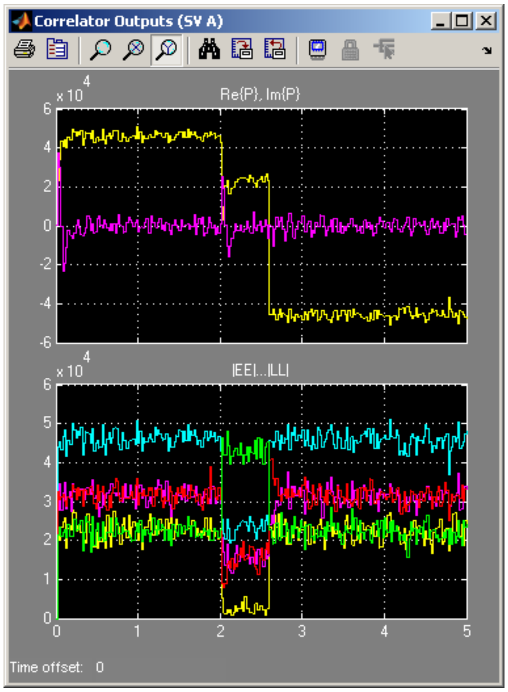


Figure 15: Correlator Outputs with forced false lock at t=2s (corrected approximately 0.5s after)

For this channel, a false lock (code lock at a secondary correlation peak) was forced after 2 seconds of simulation. It can be seen from the figure that the tracking loops detect the false lock and, after approximately 0.5s, correct it.

CONCLUSIONS

This paper describes the GNSS Dynamics Simulator and AGGA-4 Test and Simulation Tool (GSTST), an integrated cost-effective SW tool (being developed by DEIMOS Engenharia under ESA contract) which supports the development, analysis and test of navigation and DSP algorithms for space GNSS receivers based on the AGGA-4 (a baseband DSP core being developed under ESA contract). This tool allow a user/SW developer to test SW in a representative and realistic environment (as if it was running on the receiver's processor), taking into account HW functionalities and limitations as well as environment characteristics and without the need for complex and costly HW setups.

AKNOWLEDGEMENTS

Part of the GSTST work also built upon the outputs of the AGGA4VAL and GNSSGEO projects (also funded by ESA, contracts no. 22537/09/NL/AT and 21954/08/NL/ST, respectively), from which the acquired knowledge with respect to the AGGA-4 device and the GNSS simulation tools developed by DEIMOS Engenharia played an important role.

The authors also thank Antonio Latorre (DEIMOS Space) for his valuable inputs for architecture definition of the SWUT Test-Bed and Interface..

REFERENCES

[1] J. Roselló, P. Silvestrin, J. Heim, "*AGGA-4: Core Device for GNSS Space Receivers of This Decade*", NAVITEC 2010, December 2010, ESTEC, Noordwijk, The Netherlands

[2] DEIMOS Space, "The *GRANADA Software Suite*", retrieved 2013-09-11, from http:// http://granada.deimos-space.com

[3] João S. Silva, *et al.*, "*The GRANADA Factored Correlator Model Blockset: A Tool for Fast GNSS Receiver Signal Processing Simulations*", NAVITEC 2008, ESTEC, Noordwijk, the Netherlands, December, 2008

[4] Tiago Peres, *et al.*, "*GRANADA Factored Correlator Model Blockset Verification using an FPGA-Based GNSS Receiver*", NAVITEC 2010, ESTEC, Noordwijk, The Netherlands, December 2010

[5] Aeroflex-Gaisler, "*TSIM ERC32/LEON System Simulator Product Sheet*", February 2010

[6] Aragón-Ángel, *et al.*, *"Preliminary NeQuick Assessment for Future Single Frequency users of Galileo"*, In Proceedings of the 6th Geomatic Week, Barcelona, Spain, February 8-11, 2005.

[7] *"Navstar GPS Space Segment/Navigation User Segment Interfaces"*, IS-GPS-200F, 21/09/2011

[8] *"Navstar GPS Space Segment/User Segment L1C Interface"*, IS-GPS-800B, 21/09/2011

[9] *"Navstar GPS Space Segment/ User Segment L5 Interfaces"*, IS-GPS-705B, 21/09/2011

[10] *"Galileo Open Service Signal In Space Interface Control Document"*, OS SIS ICD, Issue 1.1, September 2010

[11] J. Silva *et al.*, *"Integrated and Cost-Effective Simulation Tool for GNSS Space Receiver Algorithms Development"*, NAVITEC 2012, December 2012, ESTEC, Noordwijk, The Netherlands