

DEIMOS Space Develops a Bit-True Software Receiver for the Galileo Positioning System with MathWorks Tools

When the European Union's Galileo satellite navigation system becomes operational in 2010, it will serve as an independent alternative to the U.S. Global Positioning System (GPS). Galileo's 30 satellites will provide high precision and improved coverage of satellite signals at high latitudes.

To expedite development, the Galileo Joint Undertaking (GJU) funded the Galileo Receiver Analysis and Design Application (GRANADA), a configurable tool that serves as a platform for integrating and evaluating receiver technologies.

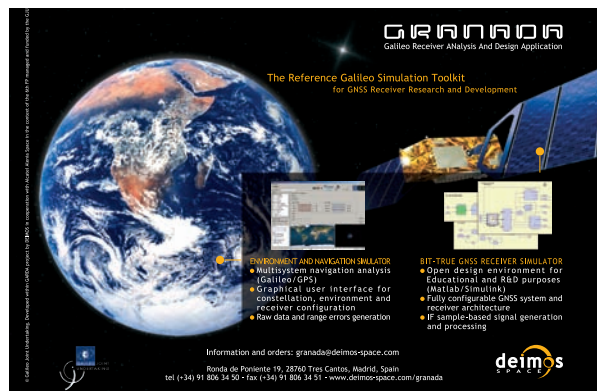
The Spanish company DEIMOS Space developed a bit-true software receiver for GRANADA entirely in MATLAB® and Simulink®, using a sampled-based simulation approach.

"MathWorks tools provided a complete design and simulation environment for modular receiver development," says José Diez, lead engineer on the GRANADA project for DEIMOS Space.

THE CHALLENGE

DEIMOS Space needed to build an end-to-end simulation of GRANADA components, including signal generation, transmission through the atmosphere, and processing by the complete receiver chain. GRANADA needed to be modular and flexible enough to enable application developers to replace the standard receiver architecture with their own algorithms and code.

The signal processing chain has a sampling frequency of approximately 200 MHz from the transmitter to the receiver. Most simulators of this kind require additional CPUs, but DEIMOS Space needed to achieve bit-true simulation of this chain on a single PC.



The GRANADA application.

THE SOLUTION

DEIMOS Space engineers used MATLAB, Simulink, and Signal Processing Blockset to develop the bit-true software receiver.

They followed European Cooperation for Space Standardization (ECSS) E40B standards for software development throughout the project, relying on MathWorks tools in the architecture design and code implementation phases.

"The GJU validated that we were following the ECSS-E40B standard throughout software development with MATLAB and Simulink. No additional investment was needed to turn a prototype into a finished product," says Augusto Caramagno, Head of the Advanced Projects Division.

The team developed a Simulink model that generated Galileo satellite signals. DEIMOS Space needed to simulate very different sampling frequencies in the same model, enabling the user to configure each frequency independently. They did this using the variable step feature in Simulink. "Variable steps were one of the key features of Simulink for us," Caramagno says. "We wouldn't have been able to offer users full flexibility in the receiver frequency plan without them."

THE CHALLENGE

To develop an end-to-end simulation of the European global positioning system, including a bit-true software receiver

THE SOLUTION

Use MathWorks tools to simulate signal generation and transmission, design the software receiver algorithms, and implement the receiver architecture

THE RESULTS

- Development time and cost reduced by 40%
- Application developed under ESA SW engineering standards
- Efficient code generated



We have seen a number of project teams at DEIMOS Space migrate from Fortran or C programming to Simulink and Real-Time Workshop because they realize a 40% improvement in development time and the resultant code is just as fast.



José Díez, DEIMOS Space

Using Simulink and Signal Processing Blockset, they then created an initial version of the receiver, incorporating basic data acquisition, tracking loops, and data demodulation capabilities.

“We used Signal Processing Blockset extensively because all the simulations required discrete signals, and the blockset enabled us to define the sample frequencies very easily,” says Díez.

In Simulink, DEIMOS Space ran simulations that concurrently generated signals, which they processed in the receiver chain. After debugging the receiver, they created a transmission channel model and introduced noise and perturbations, such as multipath delay and GPS interference, and dynamics, such as Doppler shift.

For each transmission effect that was added, the team retested and refined the receiver model, adding more complex tracking loops, filters, and other enhancements.

With MATLAB they created an interface that enables application developers to configure GRANADA by selecting parameters before automatically generating C code from the model using Real-Time Workshop®.

GRANADA was completed and delivered on time as part of the first call of the GJU’s Galileo Research and Development Activities program.

GJU has authorized DEIMOS Space to commercialize GRANADA and make it available to a wider range of GNSS Receiver Manufacturer and Navigation Application developers.

THE RESULTS

▪ **Development time and cost reduced by 40%.** “If we had to develop GRANADA in Fortran or C, we estimate it would have taken 40% longer,” says Antonio Fernández, Head of GNSS Technologies in the Advanced Projects Division at DEIMOS Space. “In addition, MATLAB and Simulink enabled us to reduce design and coding costs by 40% to 50%.”

▪ **Application developed under ESA SW engineering standards.** “Reviews by ESA and GJU experts validated that we followed ESA SW standards throughout development,” says Augusto Caramagno, Head of the Advanced Projects Division at DEIMOS Space. “GRANADA is not a prototype; it is fully certified under the ECSS-E40B standards.”

▪ **Efficient code generated.** “With Simulink we optimized the computational load of the simulation and ran bit-true simulations with high sampling frequencies on a single PC,” says Caramagno. “We had one engineer reproduce part of the software in pure C to see if we could achieve better performance. There was almost no difference in speed between that code and the code automatically generated with Real-Time Workshop.”

To learn more about DEIMOS Space and GRANADA, visit
www.deimos-space.com/granada

www.mathworks.com

APPLICATION AREAS

- Aerospace and defense
- Communications
- Signal processing
- Modeling and simulation
- Algorithm development and deployment

PRODUCTS USED

- Simulink®
- MATLAB®
- Signal Processing Blockset
- Real-Time Workshop®