Galileo Sensor Station Ground Reference Receiver Performance Characteristics

Neil Gerein, *NovAtel Inc.*
Allan Manz, *NovAtel Inc.*
Michael Clayton, *NovAtel Inc.*
Michael Olynik, *NovAtel Inc.*

BIOGRAPHY

Neil Gerein is a GPS Software Engineer with NovAtel Inc. has been involved with Galileo based receiver studies for the past two years. He has a B.Sc. in Electrical Engineering from the University of Saskatchewan, and is currently a part time graduate student at the same institution.

Allan Manz is currently doing GPS receiver research and development for NovAtel Inc. Prior to joining NovAtel Inc., he was employed by the National Research Council of Canada to investigate autonomous collision avoidance on mobile platforms.

Michael Clayton is a computer systems engineer with over twenty years of progressive responsibility developing and managing system solutions to meet user requirements. Michael graduated from the Royal Military College in 1978 with a Bachelor of Engineering (Electrical) and from Carleton University with a Masters of Engineering (Electrical) in 1984. Michael is a Registered Professional Engineer in Alberta (APEGGA). He was a Communications and Electronics Officer in the Canadian Forces from 1978 through 1990. From 1990 to 19991 Michael was the System Security Engineer on the Canadian Automated Air Traffic System. From 1991 through 1998, Michael was the Director of Software Engineering for a software services company. In 1998 Michael joined NovAtel as Senior Project Manager – Aviation Group

Michael Olynik has a B.Sc. and M.Sc. in Geomatics Engineering from the University of Calgary. He is currently doing development and testing on the Galileo Reference Receiver MATLAB Simulator.

ABSTRACT

The Galileo Ground Segment design includes a global network of Galileo Sensor Stations (GSS) to be used for orbit determination, time synchronization, and integrity determination. NovAtel, under contract to ESA, is developing the requirements for the high quality Ground Reference Receivers to be used in the GSS. During the design process NovAtel is leveraging their experience as the world's leading supplier of Ground Reference Receivers to satellite augmentation systems in the USA, Europe, Japan and China.

The first step in this design process is the development of receiver requirements together with the confidence that these requirements can be met. The Binary Offset Carrier (BOC), multiplexed codes, multiple carrier frequencies, potential use of digital pulse blanking, and new high rate spreading codes make the design of a Galileo Reference Receiver challenging. To meet this challenge NovAtel is developing an A/D sample level software simulation of a Ground Reference Receiver to verify performance characteristics during the requirements definition phase.

The critical performance characteristics of the Galileo Reference Receiver will be reviewed. The anticipated tracking, multipath mitigation and interference rejection performance of the Galileo Reference Receiver will be discussed. An overview of the A/D sample level software simulator will be presented. Test results from the simulator will be presented showing the predicted code and carrier tracking performance of the receiver.

INTRODUCTION

Approximately thirty Galileo Sensor Stations (GSS) will be distributed worldwide to provide measurements to the Galileo Control Centres (GCC). Each GSS will contain two to three reference receivers. The primary function of the receivers in a GSS is to consistently provide demodulated signal symbols and high precision pseudorange and carrier phase measurements. The ability to provide this information in less than ideal environments is also a requirement. In order to meet design assurance levels, many of the ancillary functions usually performed by a satellite based positioning receiver are eliminated, as they are not required for this application. The GSS receivers are optimized for fixed positions, continuous operation, and high quality reference oscillator inputs. Additionally, the network comprised of multiple receivers provides redundant information. This redundant information can be used to detect errors and improve performance with greater reliability and accuracy than is possible for a stand-alone receiver. A receiver in a network can therefore be more aggressive in collecting data and thus provide more information because of the additional safeguards provided though the network. Currently the development of the Galileo Reference Receiver (GRR) is in the requirements definition phase. NovAtel, under contract to ESA, is developing a high fidelity software simulator to be used to verify performance requirements during this phase. NovAtel is also developing a high-level architecture design for the GRR.

The high level conceptual design for the GRR is based on the NovAtel Common Reference Receiver (CRR) currently in development for WAAS (see Figure 1). A single RF/IF analog radio for every Galileo frequency is implemented. In addition, the design optionally supports GPS frequencies. The tuned RF/IF radio supports the digitization of only the signals in the frequency band containing the desired transmitted signal. The digitized signals are then correlated by a number of parallel mechanisms. Each mechanism is optimized to track one transmitted signal. The resultant correlation accumulations are used in code and carrier tracking control loops. The correlation accumulations are also used to extract the transmitted symbols. The state of the various tracking loops is periodically sampled at precise moments with respect to the time of the receiver. This information forms the basis of the pseudorange and carrier phase measurements that are output by the receiver. These measurements are accompanied by asynchronously gathered channel state information, such as channel tracking state, measured signal Doppler, estimated signal C/N_0, estimated carrier phase and pseudorange control loops errors, etc. The baseline system will track signals comprising the Open Service and Safety-of-Life Service. In total the receiver will have the ability to track 15 L1B (data) signals, 15 L1C (pilot) signals, 15 E5a-I (data) signals, 15 E5a-Q (pilot) signals, 15 E5b-I (data) signals, 15 E5b-Q (pilot), and 15 AltBOC signals simultaneously. The receiver will also have the capability to support additional cards to track the L1A (PRS) and E6 signals.

The decoding of the demodulated data and the use of this information in combination with the pseudorange measurements to compute a receiver Galileo time provides the means of computing unambiguous pseudorange measurements, generating a 1 pulse per second (PPS) signal and facilitates the use of the receiver data at a network level.

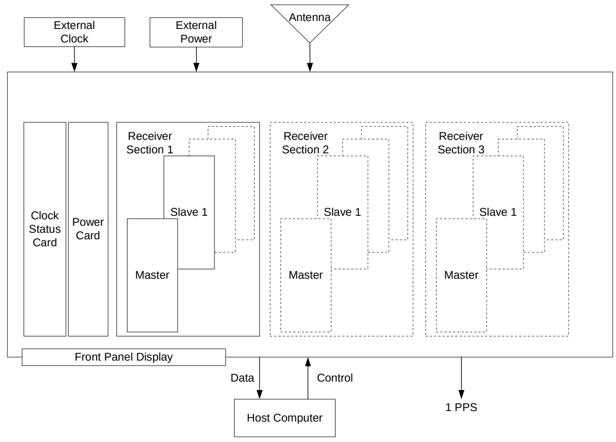

Figure 1 - GRR Functional Architecture

The proposed receiver architecture provides for a flexible arrangement of hardware to accommodate future enhancements. This flexibility requires little or no changes in high level functional components, but provides the means of improving receiver performance such as: pseudorange and carrier phase measurement accuracy; multipath mitigation; signal distortion detection; and receiver throughput. For example, additional receiver cards can be added within the chassis to provide the extra correlators needed for the implementation of Signal Quality Monitoring (SQM) or Multipath Estimating Delay Lock Loop (MEDLL^{TM}). Multiple cards can share the same digitized data across the backplane, thus eliminating RF biases.

CRITICAL PERFORMANCES

A ground reference receiver has a very specific purpose – to act as a raw measurement engine to upstream processing. The purpose of the high fidelity software simulation is to verify that a ground reference receiver can meet the proposed requirements. This section reviews some of the GRR performance requirements that can be verified through simulation prior to receiver development.

CODE AND CARRIER TRACKING ERROR

Common measures of a GNSS receiver's performance are the 1 sigma code and carrier tracking noise and biases. The noise is dependent on tracking loop bandwidth, predetection integration time, discriminator spacing, front-end bandwidth, and other factors. The biases are dependent on signal corruptions such as multipath. Various numerical approximations exist for estimating both the code and carrier tracking noise and biases, and these estimations can be used to specify the receiver requirements. High fidelity receiver simulation can be used to confirm code and carrier tracking requirements.

MULTIPATH

Isolating a signal distortion due to errors at the satellite from multipath is exceedingly difficult at a local level. The effects of multipath at local sites can be mitigated at the network level if sufficient information is made available to the central processing facility. Nevertheless, in a ground reference receiver the local multipath effects can be mitigated through the use of techniques such as Narrow Correlator™ processing, and Multipath Estimating Delay Lock Loop (MEDLL™).

The proposed Galileo BOC(2,2) signals on L1 will be transmitted with the excess bandwidth required for Narrow Correlator™ processing. During the development of receiver requirements the effects of front-end filtering and correlator spacing will be studied by simulation. With a software simulator new multipath models may be implemented and tested with less cost than implementing new models on a hardware simulator.

INTERFERENCE MITIGATION

The effect of Radio Frequency Interference (RFI) is to reduce the C/N_0 level of the received signals. If the C/N_0 level drops below the tracking threshold a loss of lock will occur. Care can be taken with the design of the receiver tracking loops to reduce the effect of RFI.
The following generalizations can be made with regard to tracking loops:

- The pre-detection integration period should be as short as possible under high dynamic stress. However, because a ground reference receiver is stationary the pre-detection integration period can be increased to improve the tracking threshold for weak signals and during periods of RFI.
- A narrow bandwidth loop filter will filter out more noise (hence improve the RFI capability). A wide bandwidth loop filter settles faster but is only desirable under high dynamic stress.
- The loop order is sensitive to the same order of dynamics (i.e first order is sensitive to velocity stress, second order is sensitive to acceleration stress, third order is sensitive to jerk stress).

One of the key features of the proposed Galileo signal structure is the use of pilot signals (i.e. no data modulation). The GNSS-1 receiver designer has traditionally been limited to using Costas Loop PLL discriminators that are insensitive to 180-degree phase reversals due to data modulation. Since the pilot signals have no data, and therefore no 180-degree phase reversals, a true four-quadrant arctangent PLL discriminator can be used. This means the pre-detection integration period can be extended beyond the data period, improving the receiver's performance in the presence of RFI. The tracking error threshold of the true PLL (full 360-degrees) is double that of the Costas PLL. If the receiver is stationary and has a high quality clock, as is the case for a ground reference station receiver, then narrowing the PLL bandwidth is a viable solution for interference mitigation[1].

The software simulation described in this paper simulates the proposed spreading codes to be generated by the GNSS-2 satellites. This allows the designer to study the effects an interfering signal has on a specific spreading code spectrum.

The pulsed interference from Distance Measuring Equipment (DME)/Tactical Air Navigation (TACAN) in the E5a/L5 and E5b frequency bands is of concern. The use of digital pulse blanking has been shown to mitigate the effects of the pulsed interference from DME/TACAN sources[2]. Because digital pulse blanking operates on a sample-by-sample basis it is suitable to use high fidelity software simulation to study its performance in a ground reference receiver environment.

Definition of the expected interference environments for the GSS is ongoing. The preliminary proposed values for in-band interference (assuming nominal received power levels of –152 dBW for L1, -152 dBW for E6, –155 dBW for E5a, and –155 dBW for E5b) are given in Table 1, and the proposed out-of-band interference is given in Table 2. The proposed pulsed interference conditions are given in Table 3.

Table 1 - Proposed In-Band Interference Assumptions

Nominal in-band interference	-141.3 dBW in any 1 MHz
Extreme in-band interference	-131.3 dBW in any 1 MHz

Table 2 - Proposed Out-of-Band Interference Assumptions

Frequency (MHz)	Total Interference/Minimum Desired Signal Power Ratio (I/S)
f<1127.95	100 dB
1127.95<f<1164.45	100-2*(f-1127.95) dB
1188.45<f<1192.07	27 dB
1216.07<f<1237.41	27+2*(f-1216.07) dB
1237.41<f<1258.75	69.7-2*(f-1237.41) dB
1298.75<f<1335.25	27+2*(f-1298.75) dB
1335.25<f<1522.552	100 dB
1522.552<f<1559.052	100-2*(f-1522.522) dB
1591.788<f<1628.29	27+2*(f-1591.788) dB
f>1628.29	100 dB

Table 3 - Pulsed Interference Parameters

Parameter	Value
Interference power (dBm)	+20
Duty cycle (%)	10
Pulse width (ms)	0.125
Bandwidth (kHz)	100

SOFTWARE SIMULATOR OVERVIEW

To aid in the development of the software simulator, NovAtel purchased the commercial MATLAB GPS Signal Simulation Toolbox from NAVSYS Corporation. The NAVSYS Toolbox is a collection of source code files that can be used to study the effects of the GPS C/A code satellites on a conventional GPS receiver. NovAtel is using the core building blocks of the NAVSYS Toolbox, along with building blocks modified for the Galileo signal structure, to develop simulations of the GRR. In this section we provide an overview of the software simulation.

The simulation consists of two main steps: 1) signal generation, and 2) tracking the received signal. Figure 2 is the high-level flow diagram of the signal generation step[3]. The user's initial position and time are used to determine the pseudorange to each satellite in view. To decrease the amount of time needed for simulation the user may select a subset of the visible satellites. The user specified spreading codes are generated and modulated with the navigation message and a carrier signal. Interfering signals can be added if desired. The composite signals are then passed through a receiver front-end software module, where the signal is filtered to a finite bandwidth and sampled. The output of the receiver front-end block is a vector of samples that a receiver Application Specific Integrated Circuit (ASIC) would "see" at the output of an analogue-to-digital (A/D) converter. This vector of digital samples is saved in a Digital Signal Format (DSF) file for later input into the receiver simulation.

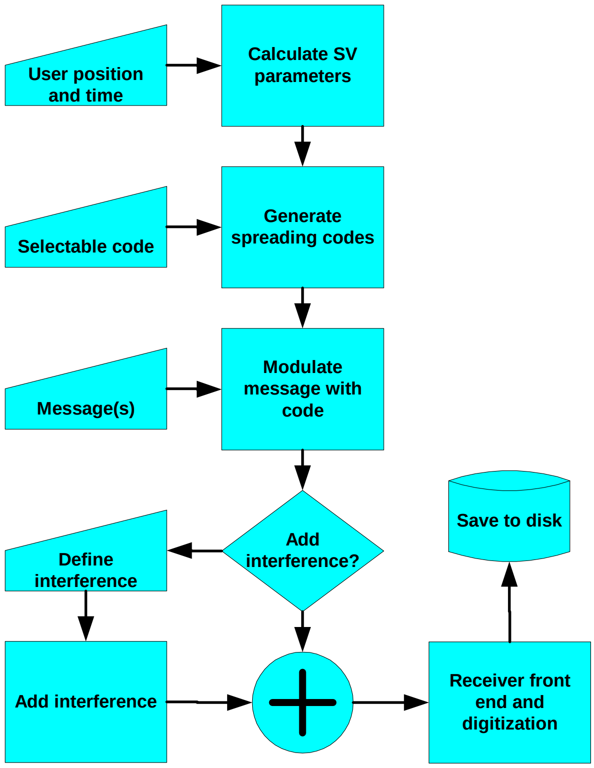

Figure 2 - Simulation Signal Generation Flow Diagram

The simulation signal generation shown in Figure 2 has a number of attractive features. The MATLAB programming language allows for new spreading codes and signal characteristics to be added with relative ease. Creating additional pseudoranges during the "Calculate SV parameters" step allows the simulation of multipath signals. Interference signals can also be defined. Simulating the filtering and sampling effects of the receiver front-end creates an accurate representation of the signal for baseband processing. Saving the digital samples to disk allows the user to compare different baseband processing configurations with the exact same set of A/D samples.

The receiver tracking flow diagram is shown in Figure 3. The receiver simulation consists of three major steps: 1) reading data from an existing DSF file, 2) processing the data through the receiver tracking loops, 3) update the tracking states. The latter two steps will now be described in detail.

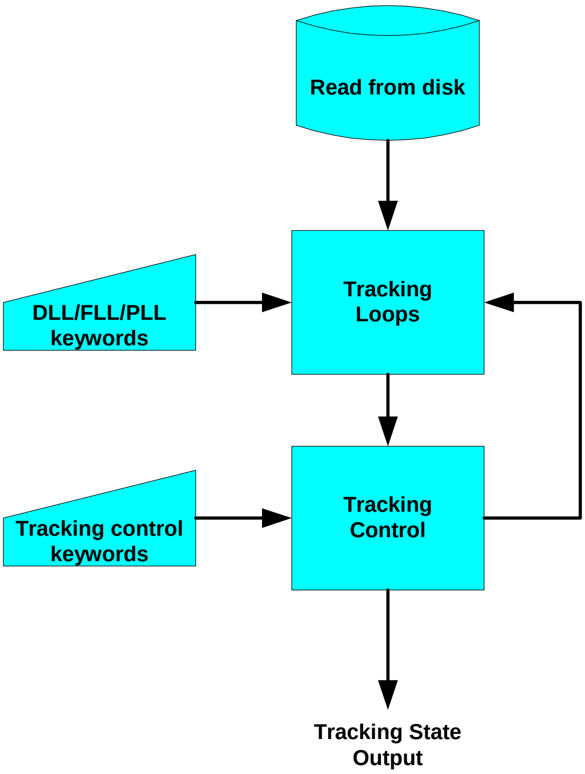

Figure 3 - Simulation Tracking Flow Diagram

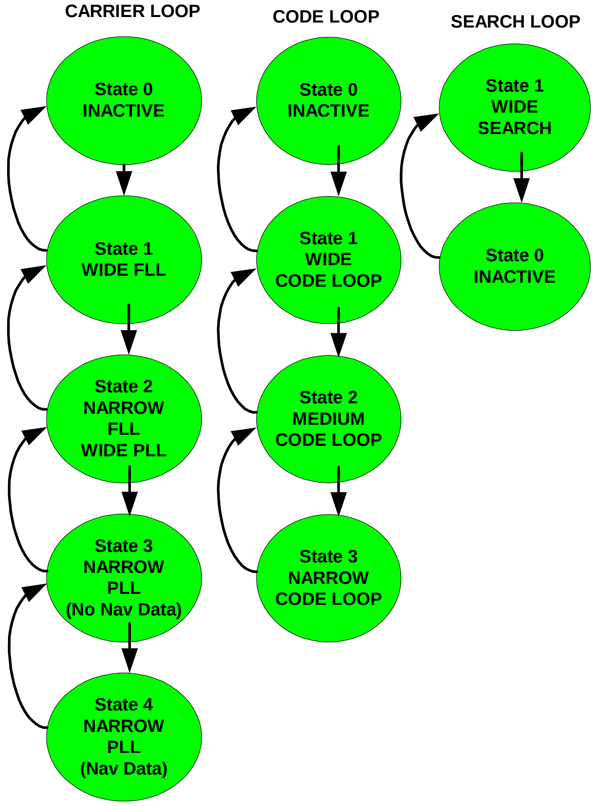

Figure 4 - Tracking States

The tracking loops consist of traditional delay lock loops (DLL), frequency lock loops (FLL), and phase lock loops (PLL). The MATLAB high level programming language offers considerable flexibility for development. For example, an almost unlimited number of correlators may be implemented, at relatively arbitrary locations along the correlation function.

The tracking control block shown in Figure 3 is used to transition between tracking states based on user-defined thresholds. The tracking state is defined as a 3 digit number, with the 100's place representing the carrier tracking state, the 10's place representing the code tracking state, and the 1's place representing the search state. A diagram illustrating the various tracking state transitions is shown in Figure 4. A typical test run starts with the receiver simulation in a wide search (state 001). After the user-defined acquisition declare threshold is reached the receiver starts the DLL and advances the code state. The carrier tracking loop is also started. The transitions between the different carrier tracking states is controlled through calculation of a locksum. The locksum returns a value between 0 and 1 that indicates the level of frequency and phase error in the tracking loop. As shown in Figure 4, the carrier loop first implements a wide FLL, then transitions to a narrow FLL/wide PLL, and finally to a narrow PLL. If the received signal contains navigation data then the carrier loop will attempt bit sync to transition to the final narrow PLL state.

User-defined keywords are used to control the signal generation and receiver simulation. Signal generation keywords include sample rate, front-end bandwidth, intermediate mixing frequency and noise figure. Receiver simulation keywords include bandwidths for code and carrier tracking loops, thresholds for advancing and reversing the code and carrier tracking states, and filter time constants used in the calculation of the carrier tracking loop locksum. The output from the receiver simulation is a tracking state output vector, with a frequency defined by the log rate.

SIMULATION EXAMPLES

The high fidelity simulation allows the user to modify a number of receiver parameters and retest, without the expense of modifying analogue and digital hardware.

The first examples simulate the effect of the front-end bandwidth on the receiver code tracking noise. Algebraic approximations are provided by Betz[4] to estimate the expected noise as a function of front-end bandwidth and early-minus-late discriminator spacing. Betz identifies three cases: 1) Spacing limited, where the noise depends primarily on the early-late spacing and not the front-end bandwidth, 2) Bandwidth limited, where the noise depends primarily on the front-end bandwidth and not the early-late spacing, and 3) A transition region between the other two cases. These cases are shown below in equations 1 to 3 respectively, where D is the normalized early-late spacing, b is the normalized front end bandwidth, T is the pre-detection integration time, T_c is the chip period, B_L is the DLL bandwidth, and C/N_0 is the carrier to noise ratio.

$$\left(\frac{\sigma_\tau^2}{T_c^2}\right)_{NELP} \cong \frac{B_L(1-0.5B_LT)}{2\frac{C}{N_0}}D\left[1+\frac{2}{T\frac{C}{N_0}(2-D)}\right]$$

$$\pi \le Db$$

(1)

$$\left(\frac{\sigma_\tau^2}{T_c^2}\right)_{NELP} \cong \frac{B_L(1-0.5B_LT)}{2\frac{C}{N_0}}\left(\frac{1}{b}\right)\left[1+\frac{1}{T\frac{C}{N_0}}\right]$$

$$Db \le 1$$

(2)

$$\left(\frac{\sigma_\tau^2}{T_c^2}\right)_{NELP} \cong \frac{B_L(1-0.5B_LT)}{2\frac{C}{N_0}}\left[\frac{1}{b}+\frac{b}{\pi-1}\left(D-\frac{1}{b}\right)^2\right]\left[1+\frac{2}{T\frac{C}{N_0}(2-D)}\right]$$

$$1 < Db < \pi$$

(3)

Figure 5 shows the spacing limited case described by equation 1. For this example the normalized bandwidth is set to 10 and the early-late discriminator spacing is set to 1 chip.

Figure 6 shows the bandwidth limited case described by equation 2. For this example the normalized bandwidth is set to 2 and the early-late discriminator spacing is set to 0.2 chips. For reference the spacing limited case described by equation 1 is shown as the solid line.

Figure 7 shows the transition region case described by equation 3. For this example the normalized bandwidth is set to 2 and the early-late discriminator spacing is set to 1 chip. For reference the spacing limited case described by equation 1 is shown as the solid line.

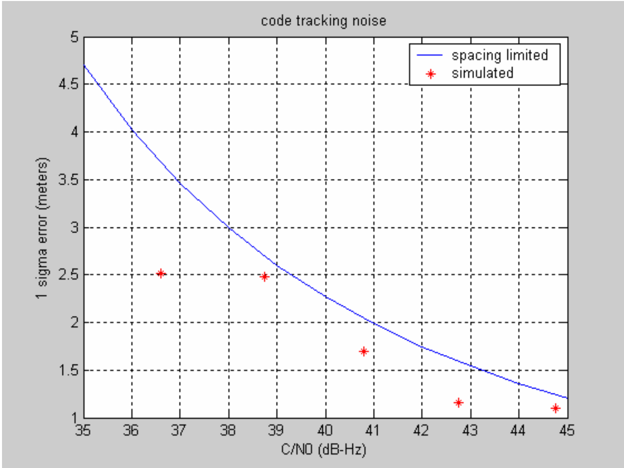

Figure 5 - Spacing Limited Tracking

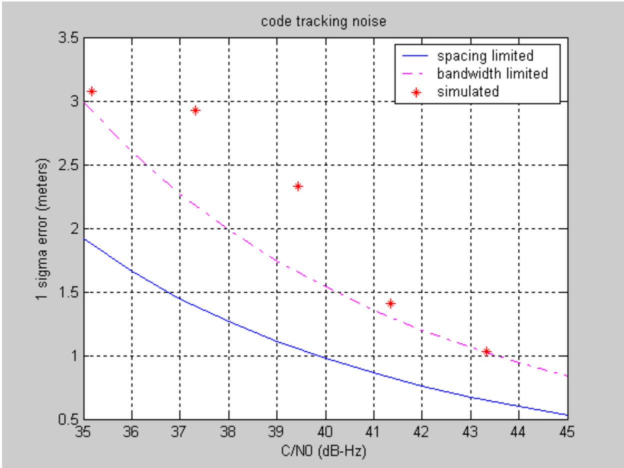

Figure 6 - Bandwidth Limited Tracking

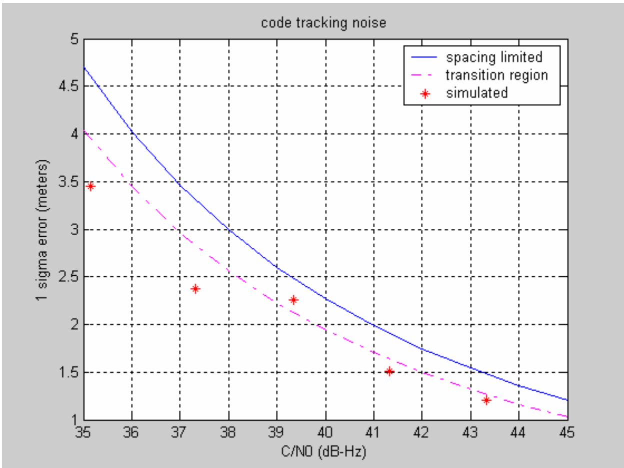

Figure 7 - Transition Region Tracking

The testing of new codes using hardware correlators involves implementation on a field programmable gate array (FPGA) or a custom made ASIC. With software simulation the receiver designer can gain insight and design experience with new GNSS-2 signals without the expense of hardware design, implementation, and debugging. This is shown in Figure 8 and Figure 9 below. For these figures the solid line is the expected code tracking noise calculated using equation 1, and the solid dots are the 1 sigma code tracking noise output from the receiver simulation. The spreading code has a chipping rate of 2.046 MHz, and a code length of 8184 chips (code period of 4 milliseconds) for the results presented in Figure 8. The same spreading code was modulated with a 2.046 MHz square wave to generate the BOC(2,2) code and used to create the results in Figure 9. In both figures the receiver front-end bandwidth was set to 20 MHz, the early-minus-late correlator spacing was set to 0.4 chips, the DLL bandwidth was set to 1 Hz, and the pre-detection integration time was set to 1 millisecond. The BOC(2,2) signal in Figure 9 has improved tracking performance when compared with the BPSK(2) signal in Figure 8, due to the "sharper" autocorrelation function of BOC(2,2).

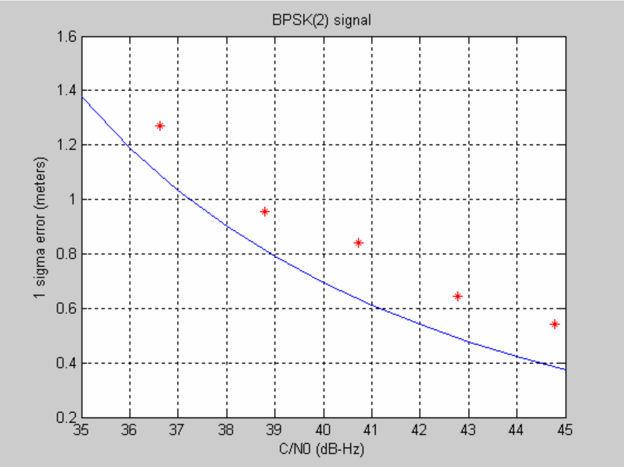

Figure 8 - Code Tracking Error (1 sigma) for BPSK 2.046 MHz chipping rate

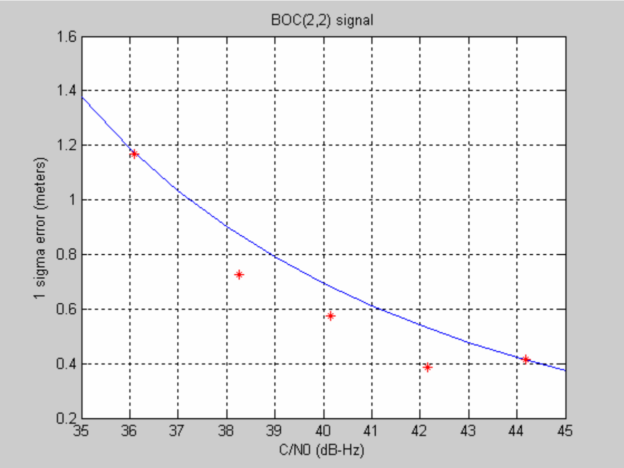

Figure 9 - Code Tracking Error (1 sigma) for BOC(2,2)

The simulation allows generation of the Alternate BOC (AltBOC) signal. AltBOC is used to generate E5AI, E5AQ, E5BI, and E5BQ as a single coherent constant envelope wideband signal[5]. Tracking of the data free pilot portion of the wideband AltBOC signal is accomplished by generating the E5AQ and E5BQ local reference signals and modulating them with the AltBOC subcarrier reference functions (see Figure 10). In the receiver simulation a the user can select the AltBOC_complex option to receive an AltBOC generated signal, but the local reference signal uses a complex BOC subcarrier reference function[6] as shown in Figure 10 (a linear offset carrier (LOC) subcarrier function is also shown for reference). The AltBOC generated signal can also be tracked separately as any of the E5AI, E5AQ, E5BI, or E5BQ signals. The user selectable simulation option AltBOC_S is the same as ALTBOC, except with the same spreading codes on E5AQ and E5BQ (E5AQ spreading code on both).

In the simulation the L1 A, B, and C signals can be generated as one signal using Coherent Adaptive Subcarrier Modulation (CASM). The three E6 signals can also be generated as a CASM signal. The CASM signals can be tracked as any one of the A, B, or C signals.

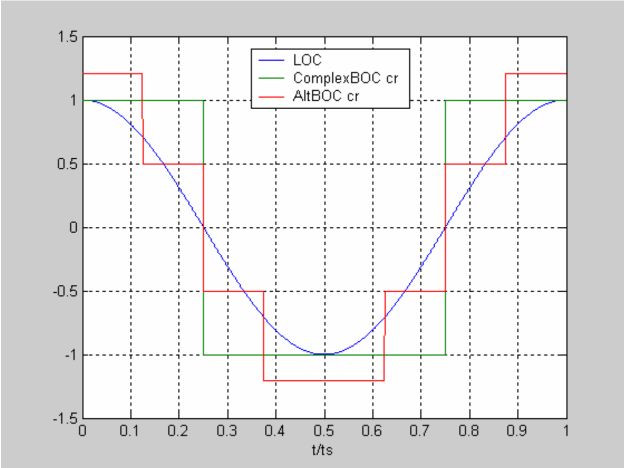

Figure 10 - AltBOC Subcarrier Tracking Functions

Results for the E5AQ and E5BQ signals are given in Figure 11 and Figure 13 showing the code and carrier phase noise at several C/N_0 values and with front-end bandwidths of 26 and 32 MHz. At lower signal strengths the results were generally better with the 32 MHz bandwidth. The level of the code and carrier phase noise is about the same for E5AQ and E5BQ, as would be expected. At a C/N_0 of 44 dB-Hz the code noise is 8-10 cm.

The code and carrier phase noise levels on the AltBOC_S signal was higher in most cases than for the AltBOC and AltBOC_complex signals, as shown in Figure 12 and Figure 14. An exception is at the lowest C/N_0. The AltBOC and AltBOC_complex code results are comparable to each other and are much better then the E5AQ and E5BQ results because of the sharp AltBOC correlation function. The phase noise is better for the AltBOC compared to AltBOC_compex at low C/N_0 values, but the values are about the same at higher signal strengths. The AltBOC and AltBOC_S as simulated in the receiver configuration used in these tests have a code noise of about 2 cm at a C/N_0 of 47 dB-Hz.

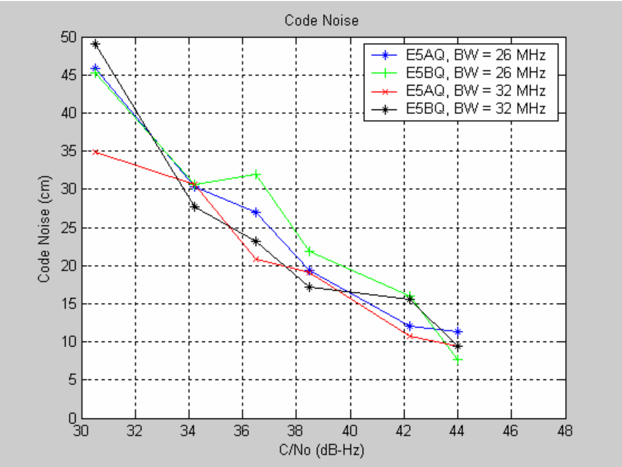

Figure 11 - Code Noise for E5 Signals

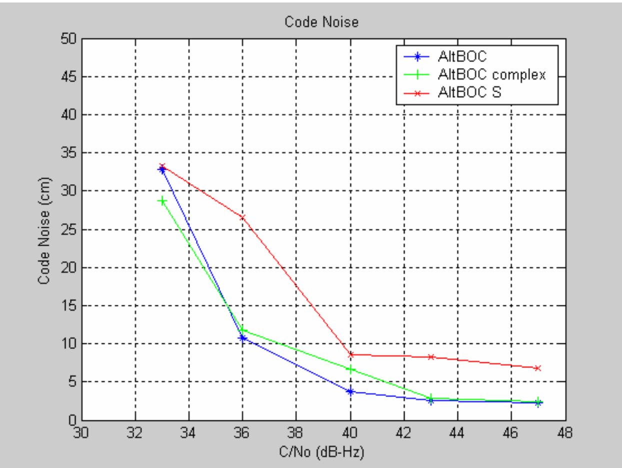

Figure 12 - Code Noise for AltBOC Signals

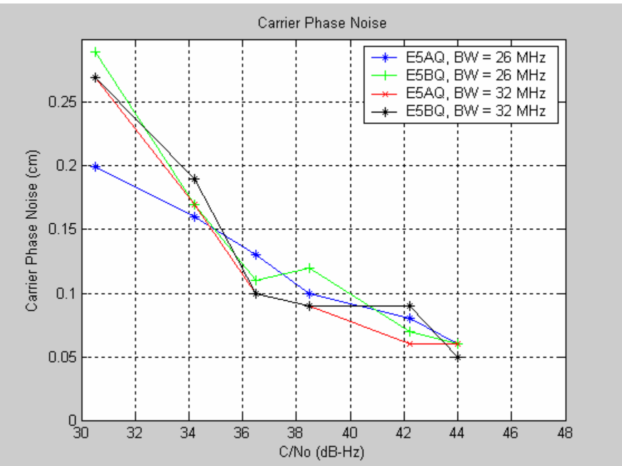

Figure 13 - Carrier Phase Noise for E5 Signals

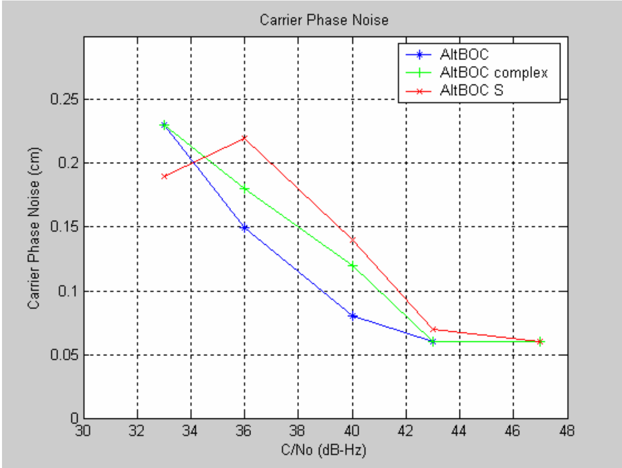

Figure 14 - Carrier Phase Noise for AltBOC Signals

MULTIPATH RESULTS

The signal and multipath parameters for the multipath tests are given in Table 4. The tests use a single ray multipath model with a multipath delay that is equal to the value at the peak of the multipath error envelope. The predetection integration time is 10 ms, and the correlator late-prompt spacing is 0.5 for the AltBOC and E5 signals and 0.1 for the E6 signals. The psuedorange measured for the multipath simulation is compared to the measurement when no multipath is present, and the average difference is the multipath error.

Table 4 - Parameters for Multipath Tests

C/N_0 (dB-Hz)	CMR (dB)
43.0	6.0
46.7	14.0
48.0	23.0
43.0	23.5
46.7	33.3
48.0	35.6

The AltBOC and AltBOC_complex multipath performance is much better than for the AltBOC_S signal, as shown in Figure 15. The AltBOC results are marginally better than the AltBOC_complex results due to generation of the AltBOC reference functions within the receiver.

Results for tracking of the E5 signals are given in Figure 16. The multipath errors are about the same for the E5AQ and E5BQ signals, as expected. Also in line with expectations, the results are much worse at the extreme carrier to multipath ratio (CMR) levels.

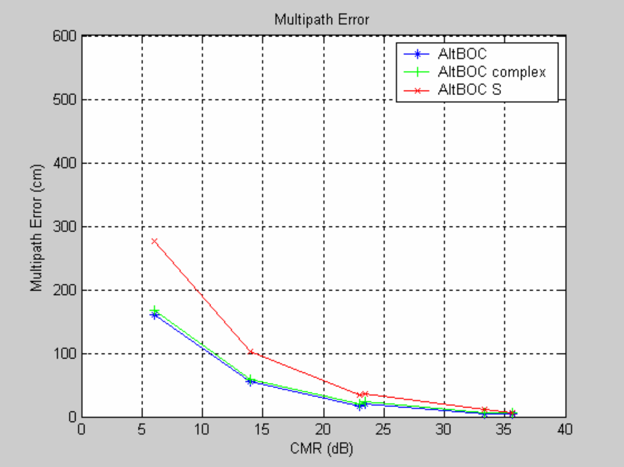

Figure 15 - AltBOC Peak Multipath Error

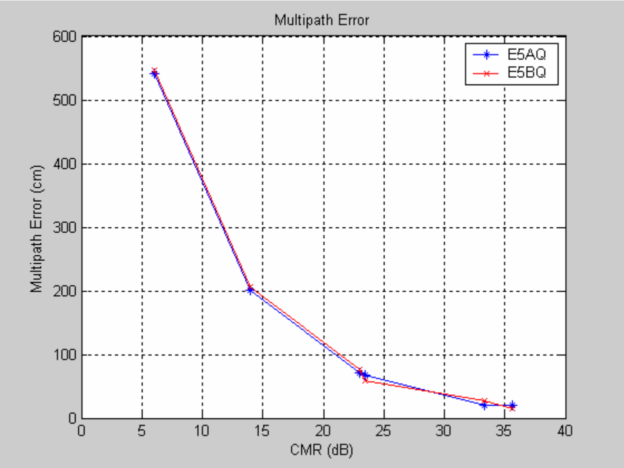

Figure 16 - E5 Peak Multipath Error

The multipath error envelopes for the L1C BOC(2,2) received signal with a CMR of 6.0 dB, a correlator spacing of 0.1 chips, and a front end bandwidth of 26 MHz is given in Figure 17. The corresponding graph for the E6A BOC(10,5) signal is shown in Figure 18.

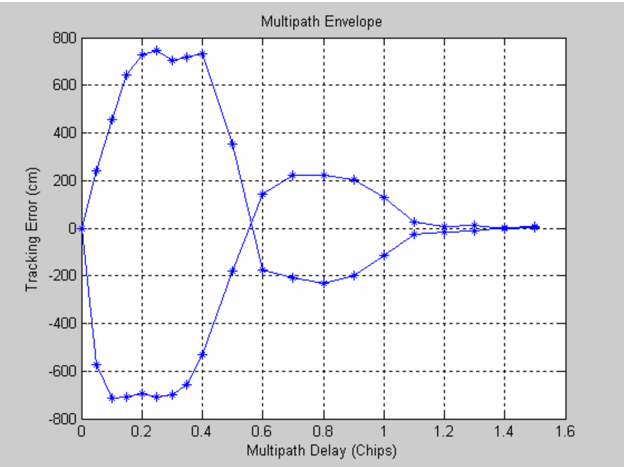

Figure 17 - BOC(2,2) Simulated Multipath Error Envelope

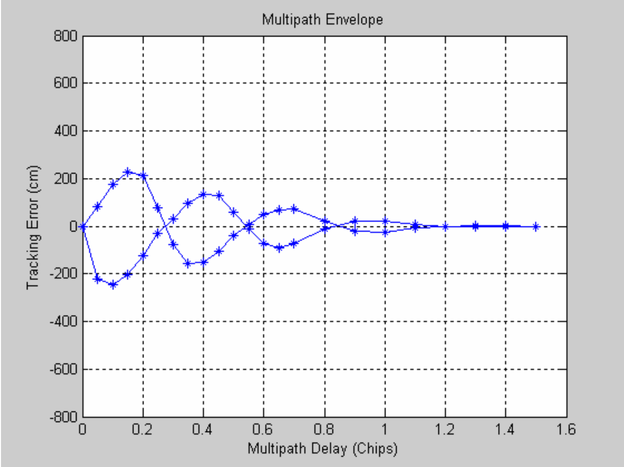

Figure 18 - BOC(10,5) Simulated Multipath Error Envelope

PULSED INTERFERENCE RESULTS

Pulsed interference tests on the E5AQ signal were done using the parameters from Table 3. Pulse blanking is necessary to track the signal, as the pulsed interference is at –10 dBW. Without pulse blanking the signal could not be tracked. The results in Figure 19 show that the C/N_0 decreased by about 4.3 dB with pulsed interference and pulse blanking, and the code and carrier phase noise both increased. The pulse blanking blanks out a significant part of the signal, which causes the lower C/N_0 and higher noise values. Plots of the code noise and carrier phase noise against the generated C/N_0 are given in Figure 20 and Figure 21.

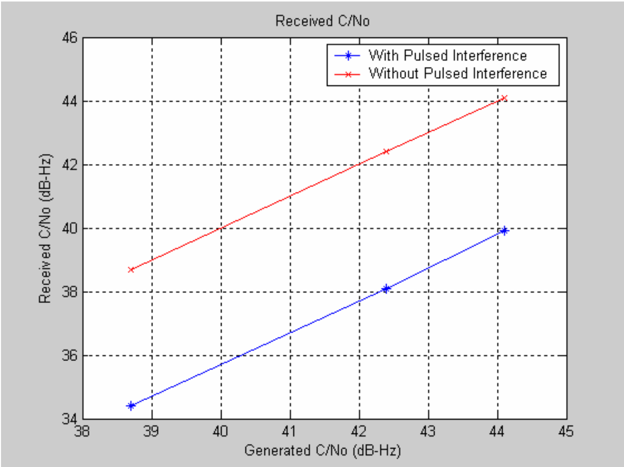

Figure 19 - Received C/N_0 with and without Pulsed Interference

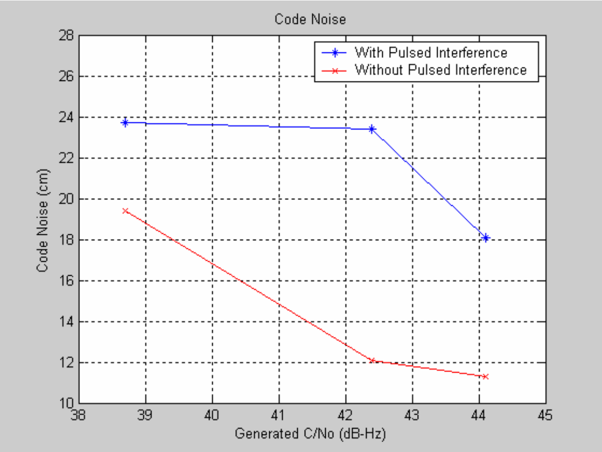

Figure 20 - Code Noise with and without Pulsed Interference

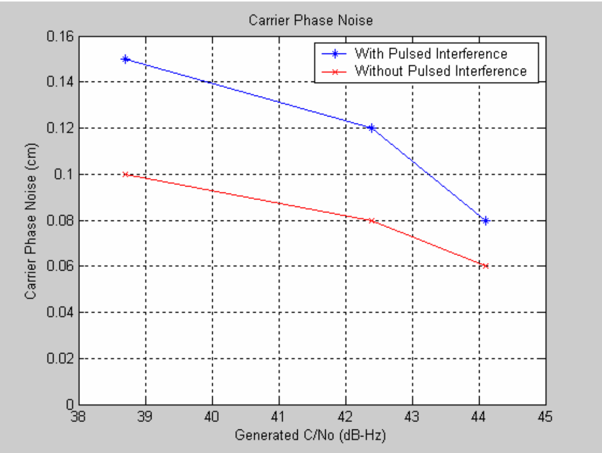

Figure 21 - Carrier Phase Noise with and without Pulsed Interference

CONCLUDING REMARKS AND FUTURE WORK

The powerful signal generation capabilities of the simulator can accommodate changes in the Galileo signal structure, new classes of interference, and updated multipath models. The simulation of the receiver design allows for testing of tracking loop changes between simulation runs while using the same set of A/D samples for each test.

Results for code and carrier accuracy tests have been presented. In addition the results of multipath and pulsed interference tests have been given.

Work is continuing on the critical performance requirements testing. Testing of various acquisition methods is currently underway.

ACKNOWLEGEMENTS

The authors with to thank ESA-ESTEC for financial and program support under ESTEC Contract 16670/02/NL/DS. The authors also thank Alcatel Space Industries for technical review support.

REFERENCES

[1] A. Manz, K. Shallberg, P. Schloss, "Improving WAAS Receiver Radio Frequency Interference Rejection", *Proceedings of the ION GPS-2000*, Salt Lake City, Utah, September 2000

[2] J. Grabowski, C. Hegarty, "Characterization of L5 Receiver Performance Using Digital Pulse Blanking", *Proceedings of the ION GPS-2002*, Portland, OR, September 2002

[3] A. Brown, N. Gerein, K. Taylor, "Modeling and Simulation of GPS Using Software Signal Generation and Digital Signal Reconstruction," *Proceedings of ION Technical Meeting*, Anaheim, CA, January 2000.

[4] J.W. Betz, K.R. Kolodziejski, "Extended Theory of Early-Late Code Tracking for a Bandlimited GPS Receiver", *Navigation: Journal of the Institute of Navigation*, Vol. 47, No. 3, Fall 2000

[5] "End-to-End Performance Verification Test Plan", GAL2-ASMD-RP-42156-003, Issue 1, Rev 2, August 12, 2002.

[6] M. Soellner, "Comparison of AWGN Code Tracking Accuracy for Alternative-BOC, Complex-LOC and Complex-BOC Modulation Options in Galileo E5 Band", *Proceedings of GNSS2003*, Graz, Austria, April 2003.