GNSS Software Simulation System for Realistic High-Multipath Environments

F. Schubert(1,2), R. Prieto-Cerdeira(2), A. Steingass(1)

(1)German Aerospace Center (DLR), Insitute of Communications and Navigation
Münchner Straße 20, 82234 Oberpfaffenhofen, Germany
Email: frank.schubert@dlr.de, alexander.steingass@dlr.de

(2)European Space Agency, ESTEC/TEC-EEP
Keplerlaan 1, 2201 AZ Noordwijk ZH, The Netherlands
Email: roberto.prieto.cerdeira@esa.int

INTRODUCTION

Compared to the various effects which degrade GNSS performance in general, nowadays multipath propagation accounts for the most dominant error in satellite navigation. Other error sources like satellite clock deviation and atmospheric effects for example can be compensated to a certain degree by the use of high-stability timing equipment (as for example the hydrogen maser in GIOVE-B represents), SBAS corrections, multi-frequency Galileo ranging and pilot signals, and the future availability of civil signals with a higher bandwidth than the currently available C/A code signal.

Especially in high-multipath environments like urban and suburban areas, the performance of GNSS receivers is severely affected by multipath propagation. Extensive measurement campaigns were undertaken in 2005 by the German Aerospace Center (DLR) for different scenarios such as the vehicular urban, sub-urban, and rural environments, and the pedestrian use case to record and model effects caused by multipath signal reception. These measurement campaigns lead to the availability of sophisticated channel models consisting of combined stochastic and deterministic parts that allow for the investigation of multipath effects on GNSS receiver performance.

These realistic channel models provide series of channel impulse responses (CIR) as outputs where the plethora of distinct echoes in urban scenarios is represented by Dirac impulses at quasi time-continuous instants. However, the application of such time-continuous CIRs to an accurate GNSS signal propagation simulation proves to be a demanding task due to the high sampling rates which are necessary to cover the channel's complexity. Before the time continuous CIRs can be applied in a simulation they have to be adjusted to fit to the time discrete sampling instants. This process is done using low-pass interpolation with a sinc function.

The presented work introduces a modular C++ framework that is able of reproducing the harsh conditions of urban environments in a very precise manner. This new efficient and flexible software tool implements the whole GNSS simulation chain consisting of “signal generation – channel model – receiver” in time-domain as a sample-true simulation. The software’s main features are given after a description of the DLR GNSS urban channel model. Additionally the interpolation process of transforming the time-continuous channel impulse responses to FIR coefficients is outlined. Eventually, a demonstration of simulation runs using the urban channel model, a BOC(1,1), and a CBOC signal is given.

GNSS MULTIPATH MODELLING IN URBAN ENVIRONMENTS

Satellite navigation receivers are most challenged in urban environments. At first, the line-of-sight signal from certain satellites is often shadowed in urban canyons. Secondly, the satellite navigation radio signal is most often reflected, diffracted, and scattered on various objects like buildings, cars, trees, and lamp poles in city areas. DLR undertook high-resolution land mobile satellite channel (LMS) sounding measurements to investigate these effects which resulted in the development of highly realistic GNSS channel models for urban and suburban areas [1][2].

The channel was sounded at a frequency of 1.51 GHz and a bandwidth of 100MHz, resulting in a time resolution of 10 ns. The transmitter was mounted on a Zeppelin which served as vibration-poor hovering platform resembling a satellite. The Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) super-resolution algorithm was used to extrapolate the measured channel sounder data and to increase the resolution in time-domain.

The model comprises of a deterministic part with a generated scenery for calculating LOS signal shadowing and knife edge diffraction for house fronts, lamp poles, and tree trunks. The other observables like the number of coexisting echoes, the life span of reflectors, and the echoes' mean power are generated stochastically. Within the artificial scenery, every echo is initialized at a random position but the excess delays and Doppler phases for each echo are calculated geometrically. The measurements for the urban channel model were conducted in the Munich city centre.
Figure 1 shows an overview of the model's structure.

The main model assumption takes only multipath effects and receiver movement into account. The channel impulse response (CIR) $h(\Delta\tau,t)$ is expressed relative to the LOS signal and consists of N discrete echoes:

$$h(\Delta\tau,t)=\sum_{i=0}^{N}a_{r,i}(t)e^{-j\varphi_{r,i}(t)}\delta\left(\Delta\tau-\Delta\tau_i(t)\right)$$

where $a_{,i}(t)=\frac{a_i(t)}{a_0(t)}$ denote the relative attenuation in terms of the direct signal path. The relative excess path delay is given by $\Delta\tau_i(t)=\tau_i(t)-\tau_0(t)$.The unobstructed LOS signal is represented by $a_{r,0}(t)=1$ and $\Delta\tau_0(t)=0$. $\varphi_{r,i}(t)$ represents phase variations dependent on receiver movement and the varying multipath environment.
An automatic recognition of start and stop times of single echoes from the ESPRIT-processed data was performed. Thus, the delay range, the life cycle, and the power distribution for each echo were determined. Additionally, the number of coexisting echoes could be calculated.

The inputs to the model define the scenery, the time-variant receiver speed and heading, and the satellite's azimuth and elevation. The model's output is a series of complex, time-variant channel impulse responses. Figure 2 shows an example for one CIR output.

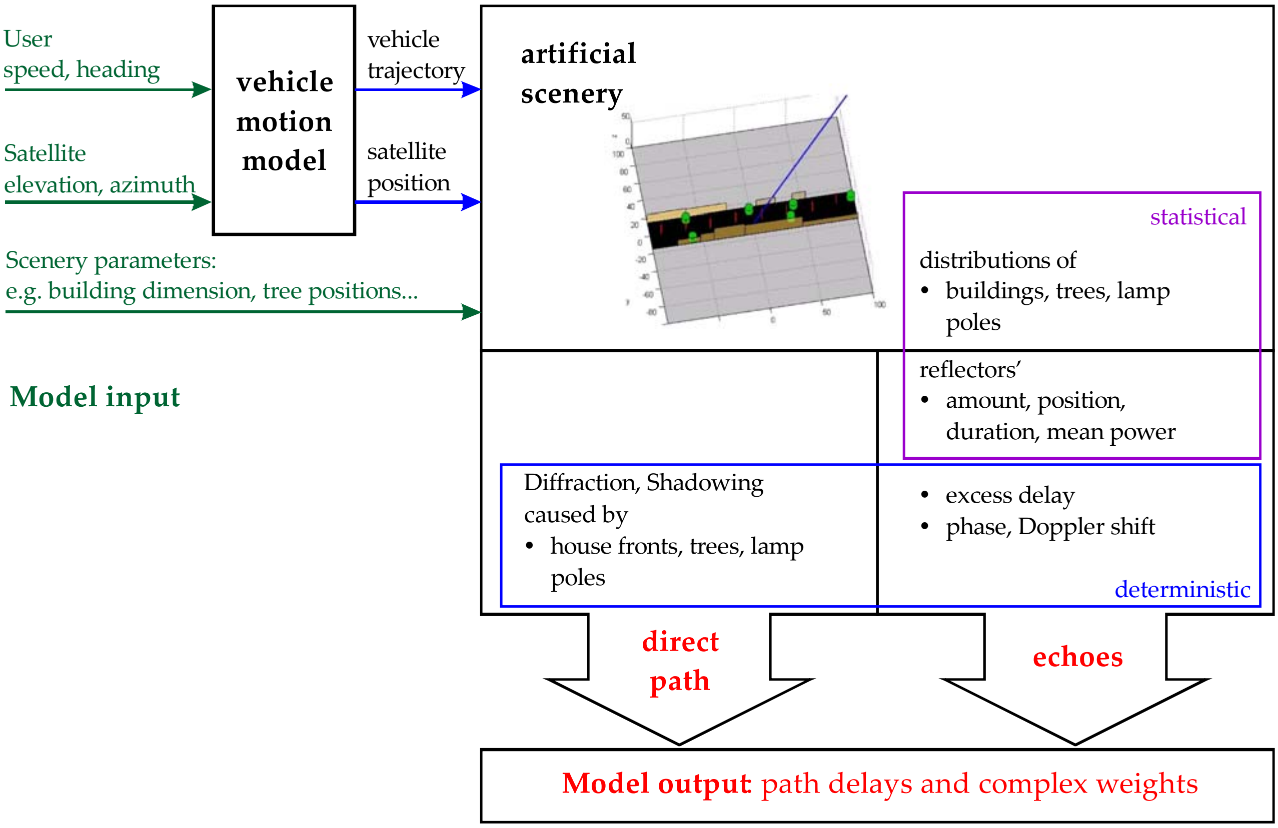

Figure 1. DLR GNSS urban channel model structure.

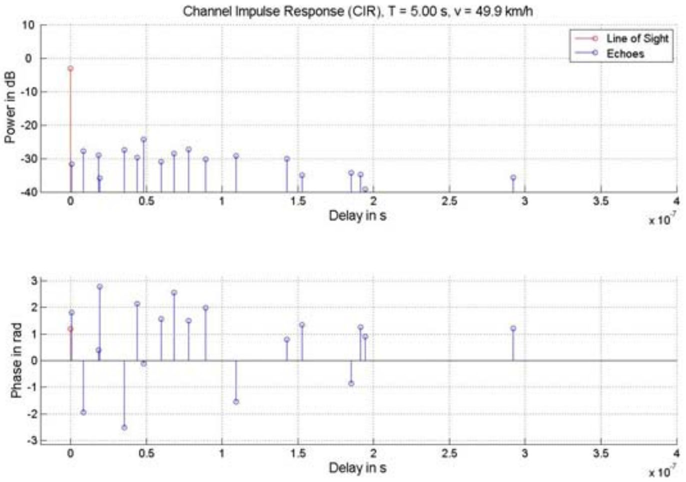

Figure 2: Output of the DLR LMS urban GNSS channel model. A single channel impulse response's magnitude and phase outputs are shown.

The model can be sampled at different rates, for the presented work, a CIR rate of R=1000 Hz was chosen. To fulfil the Nyquist Theorem regarding positive and negative Doppler, this CIR rate allows for a maximum vehicle speed of

$$v_{\max} = \frac{Rc_0}{2f_0} = \frac{1000\ \text{Hz}\ \cdot\ 3\cdot 10^8\ \frac{\text{m}}{\text{s}}}{2\cdot 1.51\cdot 10^9\ \text{Hz}} = 100\frac{\text{m}}{\text{s}}$$

with respect to stationary reflectors [3].

A Matlab version of the model can be downloaded from [4]. For the work on hand, the model output generation script has been modified to output the pseudorange in addition to the CIRs output, too. The true pseudorange is calculated using the satellite's position and the vehicle's speed at every CIR sampling instant. This reference pseudorange is transformed into a delay offset which is added to the CIR components' continuous delay. This way ensures that the vehicle dynamics are incorporated into the simulation correctly.

REALISTIC SIMULATION OF GNSS RECEPTION IN URBAN ENVIRONMENTS

Due to its realistic behaviour, the DLR urban model output comprises a high amount of drastically time-varying echoes. For the work on hand, echoes with a power of down to -100 dB with respect to LOS were taken into account. This led to a maximum of around 80 echoes in certain situations.

A new high-efficient simulation system has been developed in C++ to perform GNSS signal simulations using an object oriented approach. The primary goal of this program is to use output of highly realistic GNSS channel models and perform a convolution to simulate their impact on GNSS signals and tracking performance in time-domain. This sample-true simulator performs the following tasks:

- GNSS signal generation
- interface to channel model output (CIR time series)
- CIR to finite impulse response filter (FIR) coefficients interpolation
- FIR filtering with channel impulse responses (convolution)
- A/D conversion
- software receiver module
  - acquisition module (optional)
  - tracking module and pseudorange computation

After the tracking module, an evaluation tool is used to compare the result of the convoluted signal to the reference pseudorange.

The difficulty in simulating this chain lays in the bandwidth: For the GPS C/A code, a simulator sampling frequency of 100 MHz is being used in the presented work. The required filtering and channel convolution imposes a very high burden on the simulator software.

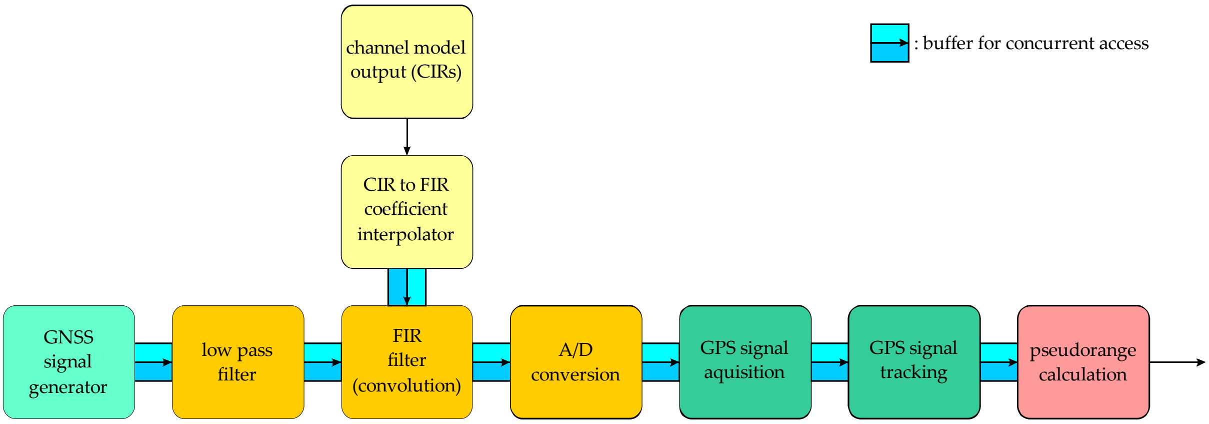

Figure 3: Structure of the newly developed sample-true GNSS simulation system.

Figure 3 shows the software's general structure. Two implementation details made this GNSS sample-true simulation in a reasonable amount of time possible: Firstly, every module runs in an own thread. Thus, all threads are connected with circular buffers for concurrent access as shown as blue bars in Figure 3. Secondly, performance-critical parts such as filtering, convolution, and correlation were optimized using Intel's Single Instruction, Multiple Data (SIMD) SSE2 instruction set. Yet, the program is still portable and compiles under Windows XP and Linux distributions.

Interpolation of continuous-valued CIR series to time-discrete FIR coefficients

The DLR GNSS urban channel model output consists of Dirac impulses at time-continuous instants. These echo pulses are not band limited. To use these CIRs in a time-domain simulation, the echoes have to be transformed into a band-limited version first. A straightforward way to do this is to use a low-pass interpolation where a sin(x)/x function is used to limit every Dirac's bandwidth to half of the sampling frequency f_s. Figure 4 shows the procedure at an example with $f_s = 100 MHz$. Three example Dirac impulses with complex weights $A_{i(i=1\ldots3)}$ are given at $\tau_1 = 0s$, $\tau_2 = 3.232 \cdot 10^{-8} s$, and $\tau_3 = 7.73 \cdot 10^{-8} s$. Each of them is represented by a respective sinc function

$$F(t) = \sum_{i=1}^{3} A_i \cdot \frac{\sin\left(2\pi \cdot \frac{f_s}{2} - \tau_i\right)}{2\pi \cdot \frac{f_s}{2} - \tau_i}$$

The sum of the resulting sinc functions F(t) is the plotted as black line in Figure 4. This resulting function can now be sampled which is shown as red crosses in the plot. These samples can directly be used as filter coefficients to an FIR filter in the simulation software.

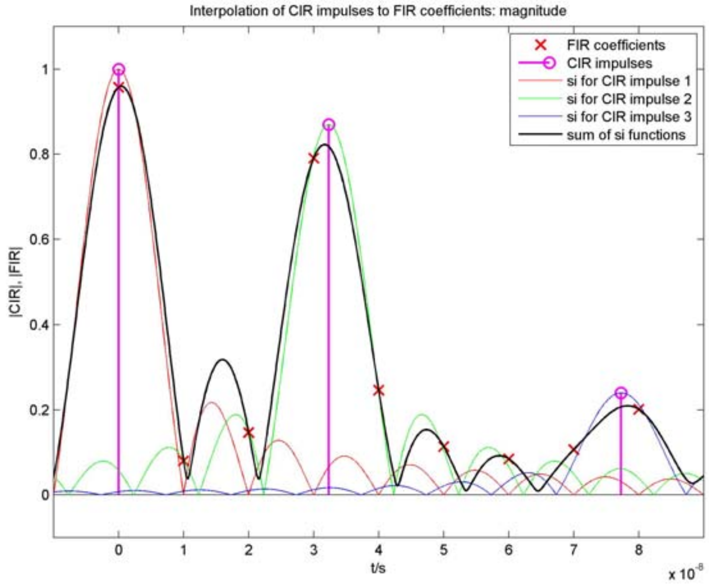

Figure 4: Three example impulses (in magenta) are interpolated using sinc interpolation and sampled afterwards. The sampling instants are shown as red crosses.

SOFTWARE SIMULATOR APPLICATION: EXAMPLE OF CBOC SIGNAL PERFORMANCE UNDER REALISTIC URBAN CONDITIONS

By now, the GPS C/A codes, the Galileo BOC(1,1) and CBOC(6,1,1/11) signals are implemented in the simulator. As an application of the simulation software, example simulations of BOC(1,1) and CBOC signals were carried out using the DLR GNSS urban channel model. The parameters of the simulations are listed in Table 1. The simulations were run noise-free to investigate the urban channel influences solely. The standard example parameters for generating the sceneries for the urban environment were set to those from the sample file which is shipped with the model. The satellite azimuth was set to 135° thus the vehicle was moving away from the satellite.

BOC(1,1) simulation

The results of a BOC(1,1) simulation run of length 120s are presented in Figure 5. The satellite elevation was set to 25° which led to frequent non-line-of-sight cases. These periods can be seen for example from 4s – 7s, 30s – 37s, and 62s – 84s. The vehicle accelerates three times up to a speed of 50km/h. Three phases of full stop can be seen from 18s – 24s, 42s – 55s, and 80s – 94s. Due to the long non-LOS phases, the overall rms error for this simulation is 11.71m.

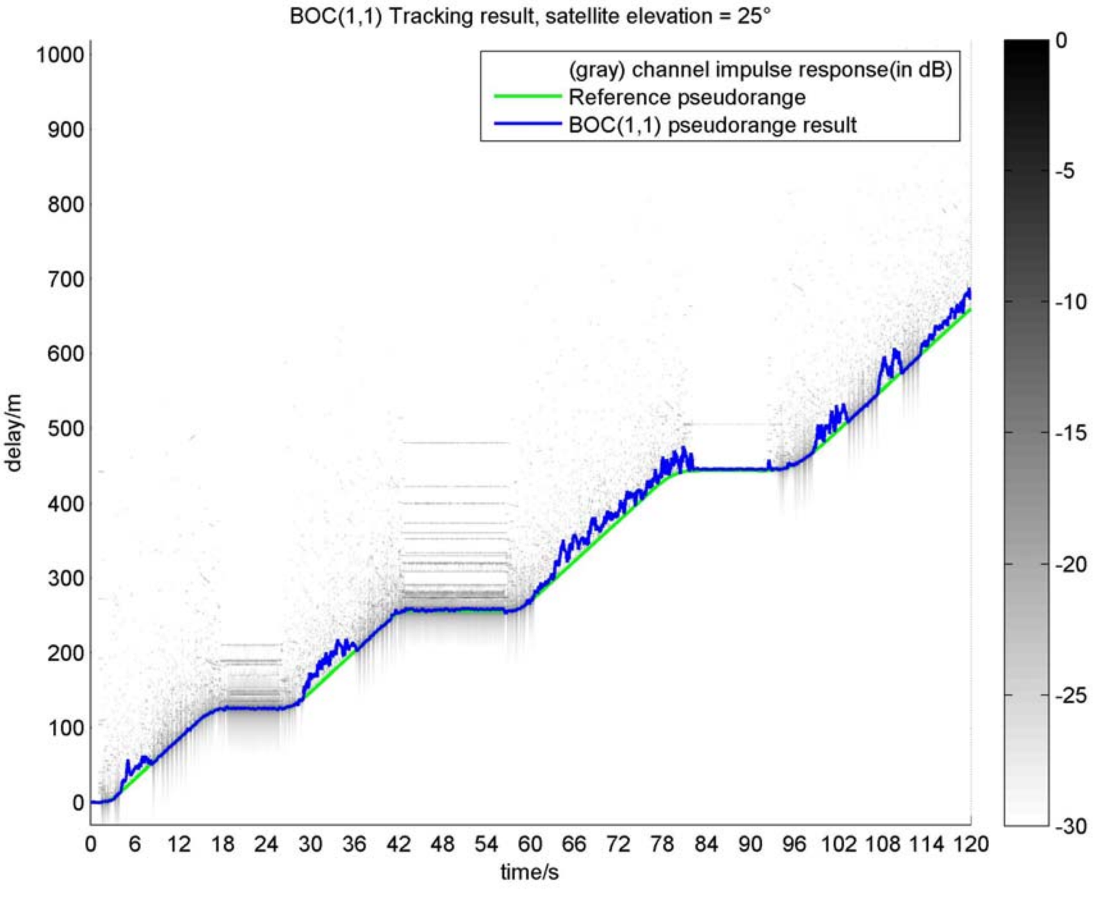

Figure 5: BOC(1,1) simulation run of 120s with the DLR GNSS urban channel model.

CBOC simulation

The CBOC signal was generated with the power spectral density [5]

$$G_{CBOC}(f) = a^2 G_{BOC(1,1)}(f) + b^2 G_{BOC(6,1)}(f)$$

and

$$a = \sqrt{\frac{10}{11}}\,,\; b = \sqrt{\frac{1}{11}}$$

The BOC(1,1) and BOC(6,1) signal components were added, yielding the time-domain waveform [6]

$$s_{CBOC}(6,1,1/11,'+')(t) = C(t)[as_{BOC(1,1)}(t) + bs_{BOC(6,1)}(t)]$$

with $C(t)$ as the code and $s_{BOC(1,1)}(t)$, $s_{BOC(6,1)}(t)$ as the BOC(1,1) and BOC(6,1) signal components, respectively.

The satellite was positioned at an azimuth of 135° and three simulation runs with the satellite elevation set to 50° and 70° were performed. The CBOC signal was generated with a two-sided bandwidth of 24MHz. The CBOC signal was sampled with 4 bits after the channel convolution to account for its four-level nature. The tracking algorithm was a multi-level CBOC correlation process.

Figure shows the first 20 seconds of a simulation with an elevation of 70° with mostly line-of-sight conditions and an rms error of 0.56m. Figure 7 shows the result of a simulation run with a satellite elevation of 70°. The direct signal path was shadowed after the fifth second which lead to an rms error of 5.11m.

Parameter	BOC(1,1) simulation	CBOC simulation
sampling frequency	100 Hz	
intermediate frequency	25 MHz	
two-sides bandwidth	8 MHz	24 MHz
ADC resolution	4 bit	
early-late spacing	0.1 chips	0.15 chips
PLL damping ratio	0.7	
PLL noise bandwidth	25 Hz	
PLL discriminator	$\varphi = \tan^{-1}\left(Q^k / I^k\right)$	
DLL damping ratio	0.7	
DLL noise bandwidth	2 Hz	
DLL discriminator	early-late	dot product
correlation time	0.004 s	

Table 1: Simulation parameters

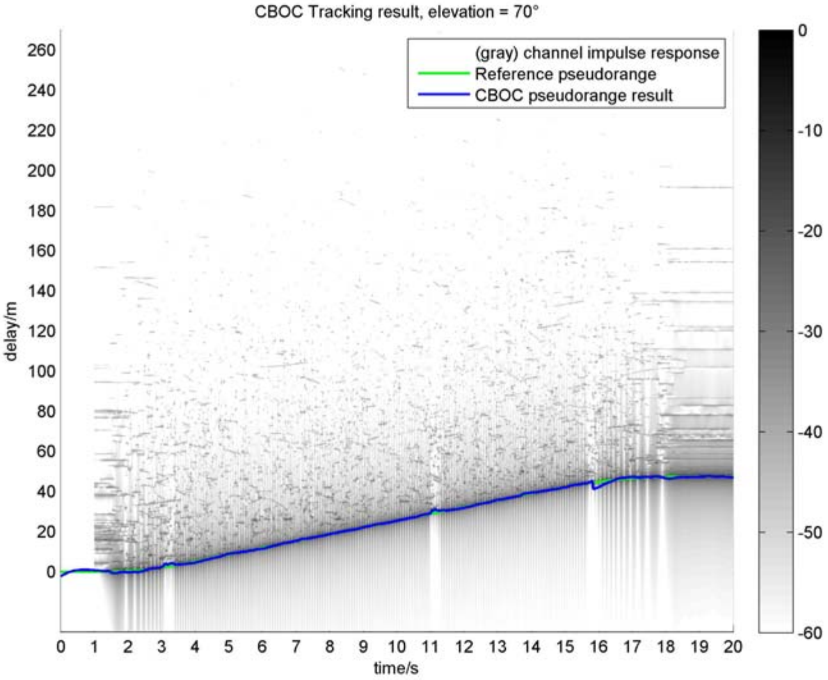

Figure 6: 20 seconds of a CBOC simulation with the satellite set to an elevation of 70° and mostly line-of-sight conditions.

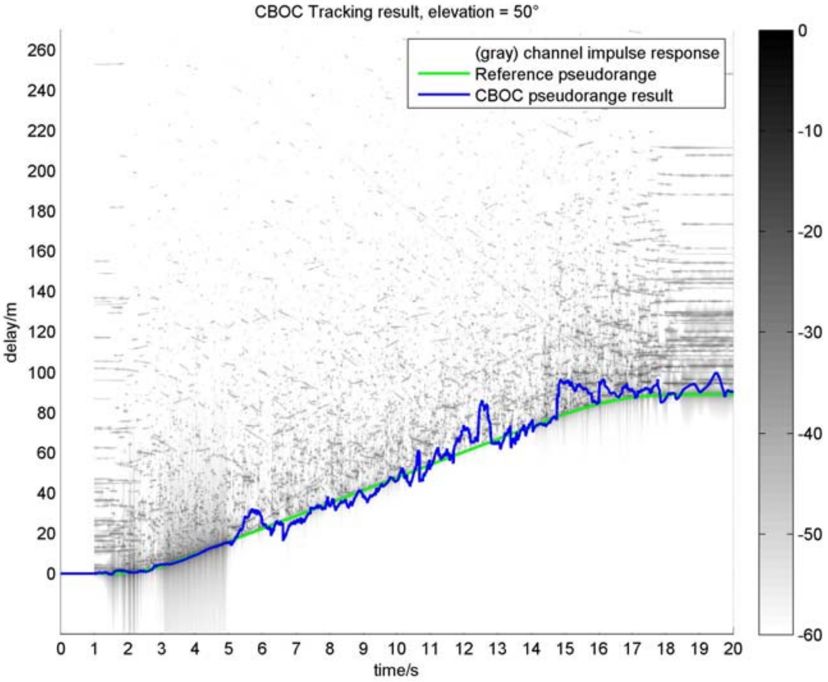

Figure 7: 20 seconds of a CBOC simulation with the satellite set to an elevation of 50° and non line-of-sight conditions beginning at the fifth second.

CONCLUSION AND FUTURE WORK

An object-oriented software framework was introduced which allows for the accurate simulation of highly time-varying GNSS channels. The software uses a low-pass interpolation procedure to transform the time-continuous Dirac-like echoes of channel impulse responses to the sampling instants of an FIR filter. This procedure allows for a precise application of the channel model output without introducing errors by mapping echoes directly to sampling points.

The software was developed using a multi-threaded approach to lower the necessary simulation time on multi-core computers. Also a set of SSE2-optimized functions is used to significantly speed up core calculations such as FIR filtering and correlation.

A demonstration of the software using BOC(1,1) and the new Galileo CBOC signal implementations together with the DLR GNSS urban channel model was given. Yet, to be able to compare BOC and CBOC tracking results using software simulations in an equal way, care has to be taken of a precise DLL gain normalization. The simulator will be extended by an AltBOC signal implementation.

The influence of parameters like user dynamics, satellite position and signal structure in challenging environments represented for example by the DLR urban channel model is to be investigated in the future using a statistical evaluation of a large set of environment parameters.

REFERENCES

[1] Lehner, Andreas; Steingass, Alexander (2005): “A Channel Model for land mobile satellite navigation,” in Proceedings of the 18th International Technical Meeting of the Institute of Navigation Satellite Division (ION GNSS 2005), Long Beach, California, USA.

[2] Steingass, Alexander; Lehner, Andreas (2007): “Navigation in Multipath Environments for Suburban Applications,” in Proceedings of the 20th International Technical Meeting of the Institute of Navigation Satellite Division (ION GNSS 2007), Fort Worth, Texas, USA.

[3] Lehner, Andreas: “Multipath Channel Modelling for Satellite Navigation Systems”, Shaker Verlag, 2007. PhD thesis.

[4] DLR GNSS urban channel model Matlab implementation download available at http://www.kn-s.dlr.de/satnav.

[5] Avila-Rodriguez, J.-A.; Wallner, S; Hein, G. W.; Issler, J.-L.; Pratt, A.R; et al: “MBOC: The New Optimized Spreading Modulation Reccommended for Galileo E1 OS and GPS L1C”.

[6] Avila-Rodriguez, Jose-Angel; Rebeyrol, Emilie; Ries, Lionel: “CBOC – An Implementation of MBOC”.