Tracking of High Bandwidth GPS/Galileo Signals with a Low Sample Rate Software Receiver

Thomas Pany, Bernd Eissfeller
Institute of Geodesy and Navigation, University FAF Munich,
85577, Neubiberg, Germany
Thomas.Pany@unibw-muenchen.de

Jón Winkel
IfEN GmbH, Munich, Germany

1 Introduction

The field of satellite navigation will see two important developments within the next years: On the one hand, new navigation signals will be broadcast from the modernized GPS [4, 7] and the European counterpart Galileo [8]. On the other hand, GNSS receiver development will change by the introduction of so-called software correlator (SWC) receivers [13, 10, 12]. Tracking of the new navigation signals has to be carefully assessed in the next years and this paper tries to estimate to which extend this is possible with a SWC receiver.

To give an example for possible high performance navigation signals, we list the Galileo binary offset carrier (BOC) signals in the E5a+E5b, in the E6 and in the E2-L1-E1 frequency band. Those signals have a large dual-sided bandwidth, which might be as large as 51 MHz for the BOC(15,10) signal in E5a+E5b even if only the two BOC main lobes of the power spectral density are considered. The bandwidth of BOC signals or of the well-known GPS binary phase shift keying signals (BPSK) further increases if side lobes are taken into account.

A pure global navigation satellite system (GNSS) SWC receiver performs the GNSS signal correlation process by a conventional personal computer (PC) instead of using dedicated chip sets, like field programmable gate arrays (FPGAs) or application-specific integrated circuits (ASICs).

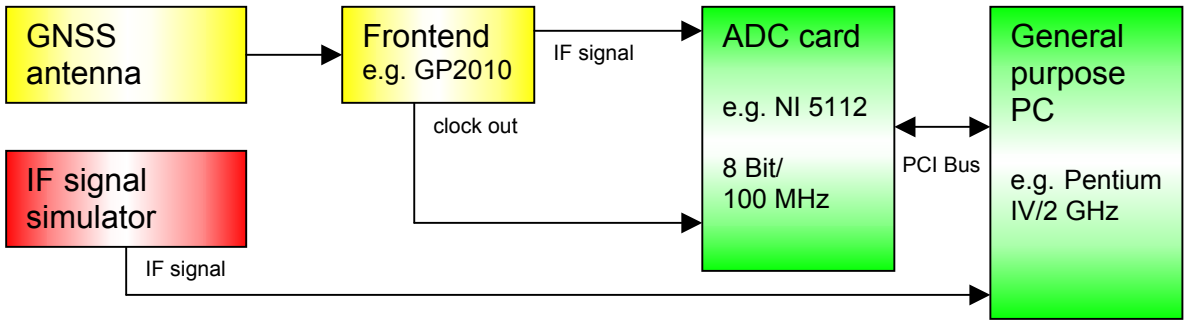

Figure 1
Block diagram of a PC based SWC receiver.

As shown in Figure 1, it consists of an antenna with an integrated low noise amplifier connected to a frontend for downconversion to an intermediate frequency (IF)[1]. An analog to digital conversion (ADC) card performs bandpass sampling of the received GNSS signals. The digital samples are processed by the PC to estimate the conventional tracking results, like pseudoranges or positions.

The major drawback of this concept is that it demands high processing power, because the tracking process requires correlating the incoming signal with reference signals at the ADC sample rate. Normally, one would expect that the sample rate must be at least twice as large as the dual-sided navigation signal bandwidth to fulfill the Nyquist criterion[2]. This would pose a problem for a SWC receiver, because tests indicate that the maximum sample rate might be on the order of 6-10 MHz with modern PCs if 8 satellites are tracked in parallel.

However, in this paper we show that the Nyquist criterion does not need not to be fulfilled to track navigation signals. This is true, because it is not necessary to retrieve the whole information (basically the pseudo-random noise (PRN) code sequence) of the navigation signals. Instead, it is sufficient to reconstruct the signal autocorrelation function. In that case, the Nyquist criterion does not apply. Consequently, Galileo and GPS high bandwidth signals can be tracked with a low sample rate at the cost of a higher tracking error due to thermal noise.

The paper starts by briefly describing a selection of three GPS/Galileo navigation signals and their generation at IF level in a software signal generator. These simulated signals are tracked in a Matlab/Simulink simulation of a GNSS receiver tracking channel. The tracking channel makes use of a new code correlator, called Cramer-Rao code correlator. Carrier phase is tracked with an *atan* PLL.

The tracking channel is capable of processing simulated signals and real (C/A code) signals from an ADC card.

The tracking channel samples the IF signals with a rate equal to two times the dual-sided bandwidth divided by a factor *u,* called undersampling factor. For $u>1$ the tracking channel works in undersampling mode. The processing demands on the CPU are reduced by a factor of *u* and the signal-to-noise ratio increases by the same factor *u*.

We calculate the code and phase tracking error due to thermal noise as a function of *u*, in two different ways. The first approach is mainly analytical and it is verified by a Monte-Carlo (MC) simulation. It is shown that the performance degradation of high bandwidth signals due to undersampling is relatively low. In fact, the thermal noise error of a BPSK(10) signal (e.g. GPS L5) is lower than for a BPSK(1) signal (e.g. GPS C/A code) even if both are tracked with the same sample rate. Conclusions and implications for SWC receiver design are given at the end.

---

1 In principle, the frontend can be skipped and the ADC card can be directly connected to the low noise amplifier. Downconversion is performed during analog to digital conversion.

2 For this statement, we assume that sampling is performed in one channel at IF level. The IF frequency is assumed to be larger than the one-sided bandwidth of the signal. The signal is split into in-phase (I) and quadra-phase (Q) channels after ADC conversion.

2 New Navigation Signals

The modernized GPS and Galileo will broadcast navigation signals using BPSK and BOC modulation schemes as outlined in [4] and [8]. Based on this signal scenario we select three representative navigation signals, to test the concept of the under sampling receiver. The signals are listed in Table 1.

Table 1 Navigation signal scenario

	Dual-sided bandwidth	Predetection integration time	Code length	Carrier frequency
BPSK(1)	4 MHz	20 ms	1023	1.57542 GHz
BPSK(10)	20 MHz	20 ms	10230	1.57542 GHz
BOC(2,2)	8 MHz	20 ms	10230	1.57542 GHz

The BPSK(1) signal is representative for the well known GPS C/A code signal on L1 and also for the L2 civil signal. The signal investigated here is bandlimited to 4 MHz. This includes the main lobe of the power spectral density and two side lobes, one each to the upper and lower side of the main lobe.

The BPSK(10) signal is bandlimited to 20 MHz (i.e. only the main lobe of the power spectral density is received) and represents the GPS P(Y) code, the future GPS L5 signal and the Galileo Open Service E5a and E5b signal.

The BOC(2,2) signal might be transmitted from Galileo satellites as a part of the Open Service on L1. Bandlimiting it to 8 MHz implies that the two main lobes of the signal are processed.

Bandlimiting of the signal is achieved by setting frequency components outside the passband region to zero. We assume that for our purposes, i.e. for simulating thermal noise error, this is reasonable approximation for real signal filters.

Figure 2 shows the signals as a function of time at baseband. All signals are received with a signal-to-noise ratio of C/N_0=45 dBHz and are assumed to be modulated onto the L1 carrier with a carrier frequency of 1.57542 GHz. The Nyquist rate is given by two times the dual-sided bandwidth of the signal.

It should be noted, that in this paper only simulated signals are considered. The simulated signals plus thermal noise are generated at IF level. The IF frequency f_{IF} is chosen such that frequency aliases of 2 f_{IF} are far from 0 MHz. Ignoring the data bit, the signal samples at IF level can be written as

$$s_\mu = Sc(t_\mu f_c - \tau_{sat})\cos(2\pi f_{IF} t_\mu - \varphi_{sat}) + n_\mu . \qquad (1)$$

μ Sample index, sample rate = f_s
s_μ Received navigation signal samples at IF level
t_μ Time of sample in [s], $t_\mu = \mu / f_s$
f_c Code rate [chip/s]

τ_{sat}	Code delay of navigation signal [chip]
f_{IF}	IF frequency [Hz]
φ_{sat}	Phase delay of navigation signal [rad]
S	Amplitude of signal
n_μ	Samples of bandlimited white thermal noise, $< n_\mu n_\nu > = N \delta_{\mu\nu}$ [3]
N	Thermal noise power
c	Bandlimited navigation signal at baseband as function of code phase

A simple calculation shows that the signal power is given by $S^2/2$. The total thermal noise power is given by N which equals the dual-sided signal bandwidth B in [Hz] times the noise density n_0 in [Hz^{-1}]. Thus the signal-to-noise ratio can be expressed as

$$C / N_0 = \frac{S^2}{2n_0} = \frac{S^2 B}{2N} . \qquad (2)$$

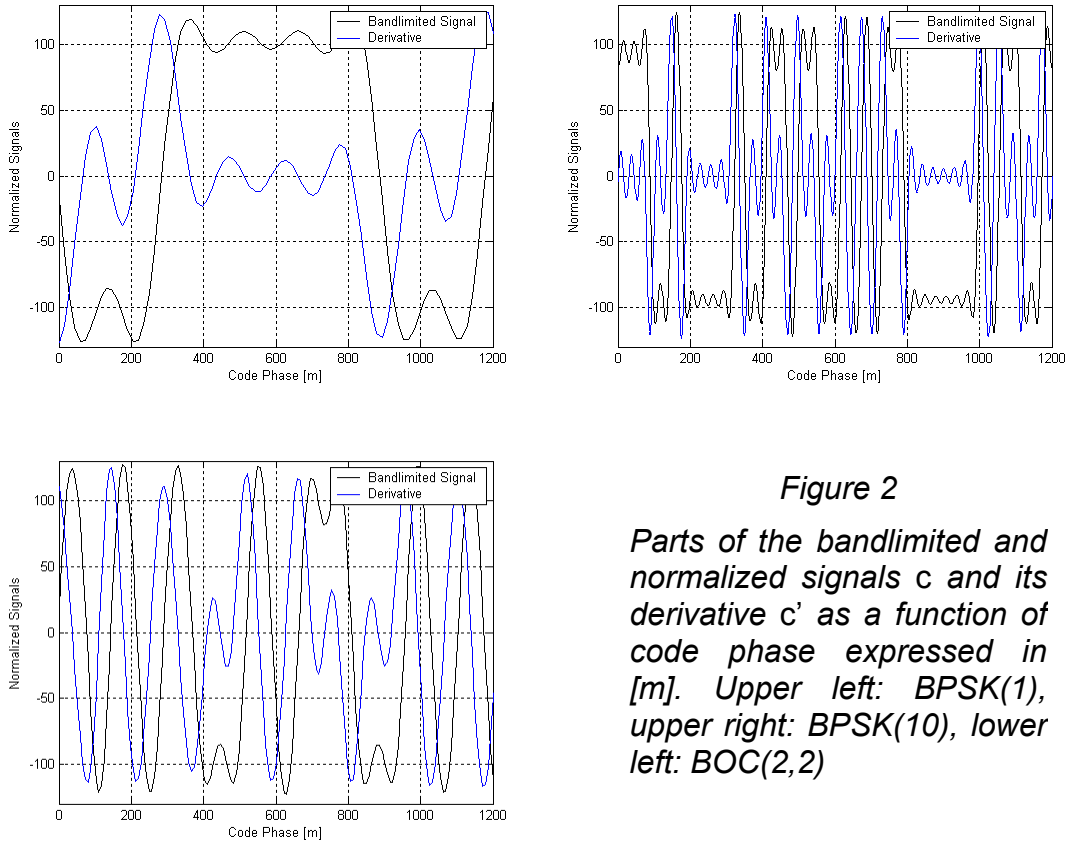

Figure 2

Parts of the bandlimited and normalized signals c *and its derivative* c' *as a function of code phase expressed in [m]. Upper left: BPSK(1), upper right: BPSK(10), lower left: BOC(2,2)*

3 The Undersampling SWC Receiver Concept

Tracking of navigation signals, as described for example in [3], consists of correlation of the received signal with internally generated reference signals to follow the position of the maximum of the correlation function. Signal acquisition must have taken place beforehand and is not discussed here.

[3] This relation is strictly true, only if the sample rate equals an integer multiple of the Nyquist rate. If not, noise samples at different times are correlated. Possible correlations are ignored here.

Various code and phase discriminators have been discussed in the literature. For the investigation here, we present a new code correlation technique, which is a generalization of the so called dot-product discriminator. We developed this technique, because it has an optimal thermal noise behavior, i.e. the code tracking error due to thermal noise is nearly equal to the Cramer-Rao lower bound [11] and it is easily implemented in a SWC receiver, where large memory is available to allow a precomputation of the reference signals.

The overall structure of the tracking channel is shown in Figure 3.

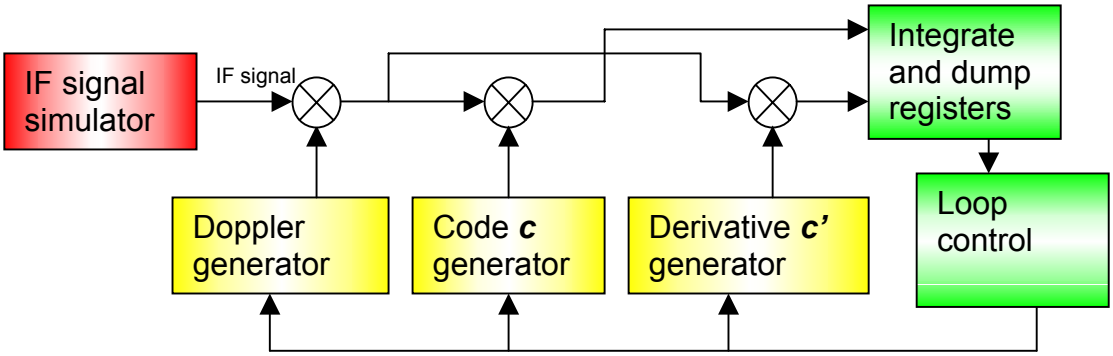

Figure 3

Block diagram of a tracking channel with a Cramer-Rao code correlator

The IF signal, coming from the IF signal simulator (or from the ADC card), is sampled with a sample rate of f_s. As mentioned above, the sample rate may lie below the Nyquist rate of 2*B*.

To shift the received signal from the IF + Doppler frequency to baseband, it is multiplied by an internally generated complex carrier frequency whose samples can be written as

$$p_\mu = 2P^D \exp\{2\pi i(f_{IF} + \Delta f)t_\mu - i\varphi_{rec}\}. \qquad (3)$$

p_μ	Internally generated Doppler frequency samples
φ_{rec}	Phase delay of internal signal [rad]
P^D	Amplitude of internally generated Doppler frequency
Δf	Doppler frequency [Hz]

After Doppler and IF removal, the signal is at baseband and separated into I and Q channel. Using the complex notation we do not explicitly indicate I and Q channel. Image frequencies at about 2 f_{IF} occur. These image frequencies are aliased to within the range [0, f_s/2] if the receiver works in undersampling mode, but if all frequencies are chosen properly they are filtered out in the integrate and dump registers and can be ignored.

For code tracking, the samples are multiplied with the first derivative *c'* of the navigation signal at baseband. Examples of navigation signals at baseband and their first derivatives are shown in Figure 2.

In hardware correlator (HWC) receivers, as documented in [3], this step is usually performed by multiplying the received signal with an early and a late replica of the code

and by forming the difference between the early and late correlator. The reason, for using early and a late code sequences, is that such sequences can be easily generated in hardware, for example by shifting the punctual code sequence. In a SWC receiver the flexibility of the general purpose processor can be exploited and the first derivative of the navigation signal can be precomputed and stored in memory. The intermediate step of an early and late sequence is not necessary in a SWC receiver. By using the first derivative, optimal thermal noise performance is assured, as will be shown later. Therefore, this code correlation technique will be called Cramer-Rao correlator.

The samples of the first derivative of Cramer-Rao generator are

$$c_\mu^D = C^D c'(t_\mu f_{c,rec} - \tau_{rec}) . \qquad (4)$$

c_μ^D Samples of first derivative of internally generated navigation signal at baseband
τ_{rec} Code delay of internal signal [chip]
c' First derivative of bandlimited navigation signal at baseband as function of code phase
C^D Amplitude of first derivative
$f_{c,rec}$ Code rate of internally generated navigation signal [chip/s]

Finally, the product of the incoming signal, the internally generated Doppler + IF frequency and the Cramer-Rao generator are summed up in the integrate and dump registers as

$$\tilde{D} = \sum_{\mu=1}^{n_f} s_\mu p_\mu c_\mu^D \Big/ \left(S P^D C^D n_f R''(0) \right) . \qquad (5)$$

n_f Number of samples within predetection integration time $n_f = T f_s$
f_s Sample rate [Hz]
T Predetection integration time [s]
$R(\Delta\tau)$ Autocorrelation function of c as a function of code phase $\Delta\tau$. $\Delta\tau$ in [chip]
R'' Second derivative of autocorrelation function

The integration period is given by the so-called predetection integration time and must be less than the duration of one data bit for data bearing channels, like the GPS C/A code. For data less channels, as they are foreseen for Galileo and the modernized GPS, T can –in principle– be arbitrarily large, minimizing the squaring loss.

The output of the (coherent) Cramer-Rao correlator $\tilde{D}$ equals in a first approximation the code phase tracking error $\tau_{sat} - \tau_{rec}$ multiplied by the complex exponential of the phase tracking error. The output could be used to form a coherent code tracking loop. Since a coherent code tracking loop loses lock, if the PLL loses lock, we do not consider this approach. By using the punctual code correlator, a non-coherent DLL will be formed later.

For phase tracking and for non-coherent code tracking, the samples are correlated with the punctual code sequence. The samples of the punctual code generator take the form

$$c_\mu^P = C^P c(t_\mu f_{c,rec} - \tau_{rec}) , \qquad (6)$$

c_μ^P Samples of internally generated navigation signal at baseband
C^P Amplitude of internally generated navigation signal

and the output of the integrate and dump registers can be written for the punctual code correlator as

$$P = \sum_{\mu=1}^{n_f} s_\mu p_\mu c_\mu^P \Big/ \left(SP^D C^P n_f \right). \qquad (7)$$

The expectation value of *P* with respect to thermal noise is given by

$$\langle P \rangle \simeq \exp\left\{ i\left(\varphi_{sat} - \varphi_{rec} \right) \right\} \qquad (8)$$

and by taking the real and imaginary part the phase tracking error can be retrieved. In this work we use an *atan* discriminator [3], since it provides optimal performance and can be easily calculated in a SWC receiver. The phase tracking error is used to steer the PLL tracking loop.

To form a non-coherent code loop, the coherent Cramer-Rao correlator output $\tilde{D}$ is multiplied with the punctual correlator,

$$D = \tilde{D}\bar{P}, \quad \langle D \rangle \simeq \tau_{sat} - \tau_{rec}\,. \qquad (9)$$

$\bar{P}$ Complex conjugate of *P*

The expectation value of *D* with respect to thermal noise is –in first approximation– independent of the phase tracking error and equals the code tracking error. This procedure is similar to the dot-product discriminator, where the early-late difference is multiplied by the punctual code correlator.

In Figure 4 we show the output of the non-coherent Cramer-Rao correlator *D* and of the absolute value of the punctual code correlator *P* as a function code tracking error for different values of the undersampling factor *u*. This image was obtained from a simulation with the BOC(2,2) signal of Table 1. The code and phase delay of the simulated IF signal were held fixed and the code delay of the internally generated signals was swept from –2 to +2 chips with a rate of 0.1 chip/s. The tracking loops were open. The frequency of the internally generated IF Doppler signal equals the frequency of the reference signal and was also fixed.

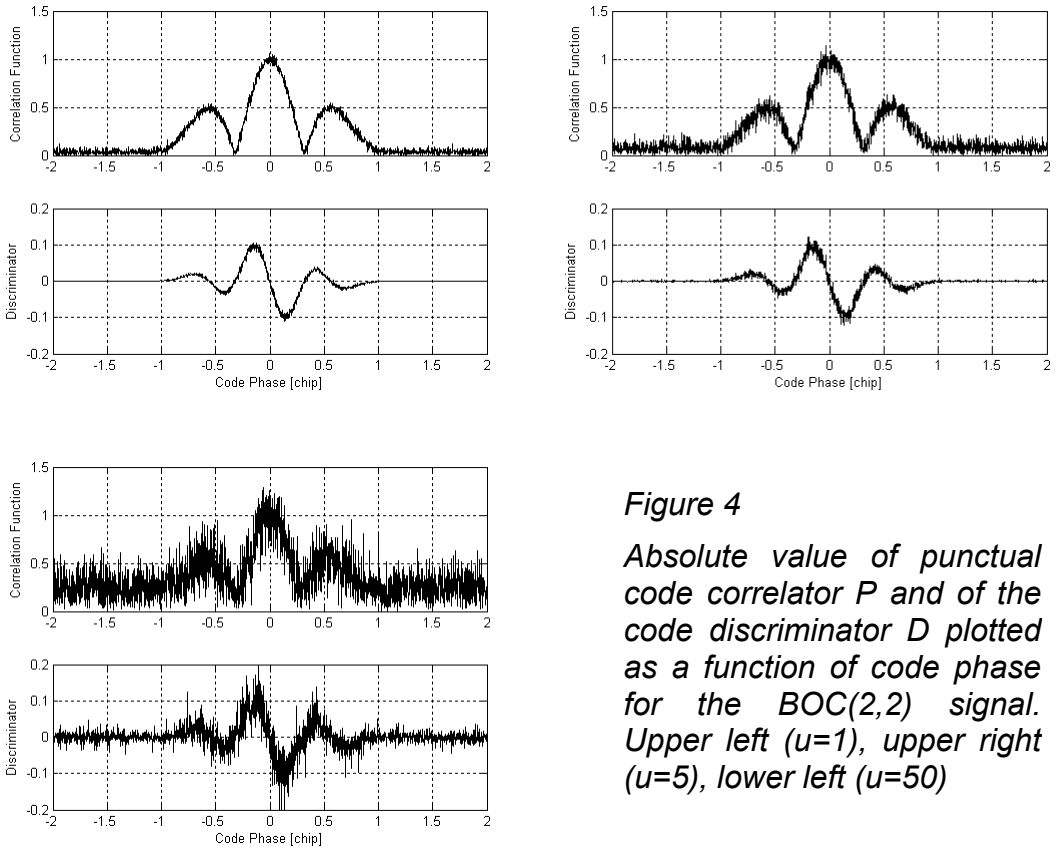

Figure 4

Absolute value of punctual code correlator P and of the code discriminator D plotted as a function of code phase for the BOC(2,2) signal. Upper left (u=1), upper right (u=5), lower left (u=50)

From Figure 4 one clearly sees the increasing thermal noise in the correlator outputs as the undersampling factor increases. In the case of no undersampling (u=1) and the chosen value for C/N_0=45 dBHz, the thermal noise influence on the correlator output is very small. The code correlation function and its first derivative appear as lines. As a consequence, the code tracking accuracy is high. For large undersampling of u=50, the correlator outputs are very noisy, but there is still sufficient signal in the outputs to steer the tracking loops and maintain lock.

To close the tracking loops, the code and phase tracking errors are fed into the code and phase loop filters and finally the code and phase generators are steered to follow the incoming signal as shown in Figure 3.

Here it should be noted that for BOC tracking it must be ensured that the main peak of the correlation function is tracked. This issue is not discussed here, as it is unimportant for the thermal noise behavior.

3.1 Thermal Noise Analysis

The variance of the code and phase tracking error with respect to thermal noise is calculated similar to the results presented in [9]. In contrast to other works, like [5] or [1], the finite sample rate is taken explicitly into account.

The analysis assumes, that image frequencies at $2f_{IF}$ are eliminated in the integrate and dump registers, that the discriminator outputs are linear in code and phase tracking

errors and that the difference between satellite and receiver Doppler frequency is negligible (for both, code and phase Doppler). For the DLL analysis we assume that the PLL tracking error vanishes and for the PLL analysis, that the DLL tracking error vanishes. Finally, for the PLL analysis, the phase tracking error has to be recovered from the complex punctual code correlator output. We assume, that the noise in the punctual code correlator output is much smaller than the value of the punctual code correlator to perform a Taylor series expansion in the thermal noise.

Making all these assumption, we arrive at the following result for the steady-state code tracking error variance in [chip^2]

$$\left\langle \left(\tau_{sat} - \tau_{rec}\right)^2 \right\rangle \simeq -\frac{B_{DLL}}{R''(0)}\left(\frac{u}{C/N_0} + \frac{u^2}{T\left(C/N_0\right)^2}\right). \qquad (10)$$

u Undersampling factor $u=f_s/(2B)$
B Dual-sided signal and noise bandwidth (before sampling)
B_{DLL} DLL bandwidth [Hz]

It is very similar to the well-known formulas for the non-coherent dot product code discriminator [2]. Note that the autocorrelation function is normalized in the sense that *R(0)=1.* Undersampling enters in the formula by the factor *u.* The only effect of this factor is that it increases the signal-to-noise ratio. Optimal performance is assured if u=1, i.e. when the tracking channel processes the signal with a sample rate equal two times the dual-sided signal bandwidth B, i.e. with the Nyquist rate.

For BPSK signals, the inverse of the second derivate of the autocorrelation function takes the role of the early minus late discriminator spacing d,

$$\frac{d}{2} \simeq -\frac{1}{R''(0)}. \qquad (11)$$

In that way, an effective correlator spacing can be defined and the Cramer-Rao correlator can be compared to other techniques. This effective correlator spacing is defined only for thermal noise analysis. To which extend it can be useful to characterize multipath errors, will be investigated in the future. Table 2 shows the effective correlator spacing for the two investigated BPSK signals. Since no formulas are available for BOC signals, an analogues relation can not be established for them.

Table 2 Effective correlator spacing

	Dual-sided bandwidth	Effective correlator spacing d
BPSK(1)	4 MHz	0.25
BPSK(10)	20 MHz	0.45

The second term in the parentheses of (10) is the squaring loss, as we use a non-coherent DLL. If the squaring loss is omitted from (10), the code thermal noise performance equals the Cramer-Rao lower bound [11]. As will be shown in Figure 5, the squaring loss is here rather small, for the scenarios investigated in this work.

For the steady-state phase tracking error variance in [rad^2] due to thermal noise, we get

$$\left\langle \left(\varphi_{sat} - \varphi_{rec}\right)^2 \right\rangle \simeq \frac{uB_{PLL}}{C/N_0}\left(1 + \frac{u}{2TC/N_0}\right). \tag{12}$$

B_{PLL} PLL bandwidth [Hz]

Again, this formula is very similar to the well-known formula for a non-coherent PLL. The only difference is the undersampling factor *u,* which increases the thermal noise.

4 Simulation

We implemented an IF signal generator and a tracking channel in Matlab/Simulink. The tracking channel is part of a complete GNSS SWC receiver performing all other tasks like acquisition, data decoding or navigation processing. Time critical parts of this simulation are realized in C++. This includes signal generation and code correlation.

The IF signal is generated with a resolution of 8 bit using (1). It should be noted that the amplitude of the navigation signal is much smaller than the standard deviation of the thermal noise, especially for large signal bandwidths. In fact, the signal amplitude might even be below one bit, and carefully designed algorithms are necessary to combine signal and noise properly.

A resolution of 8 bit was chosen, because our ADC card, a NI 5112 from National Instruments, has the same resolution and because the CPU (Pentium IV) can work most effectively with this number of bits. Lowering the number of bits would decrease the CPU performance drastically. Again, this is an important design difference to a HWC receiver.

Reference signals in the tracking channel are also generated with a resolution of 8 bit. Precomputed values for the navigation signal, its first derivative and precomputed sine/cosine tables are used to speed up performance.

It should be noted, that the C++ code has not been optimized yet, but first tests indicate, that a Pentium IV with 2 GHz can run about 4 tracking channels with a sample rate of 8 MHz, or 40 channels with a sample rate of 800 kHz. This indicates the importance of the undersampling concept, where a nearly arbitrary number of channels can be tracked with a conventional PC at the cost of an increased signal-to-noise ratio.

4.1 DLL Thermal Noise Performance

To determine the code tracking error due to thermal noise and to verify our analytical calculation we perform a MC simulation. We run the simulation for 10 times, each time for 100 s. All three signals of Table 1 are analyzed. From the data of each run, the code tracking accuracy (standard deviation) is determined. Analyzing the 10 runs, we obtain the mean value of the code accuracy and the standard deviation of the code accuracy.

The procedure is repeated for undersampling factors of u=1, 5, 10, 20 and 50. Results are shown in Figure 5. In this figure, the result based on (10) is shown too (blue line) as well as the result for a coherent Cramer-Rao correlator (green line).

Generally a good consistency between analytical results and the MC simulation is obtained. Deviation can be seen for high undersampling factors. The good agreement justifies the assumptions leading to (10).

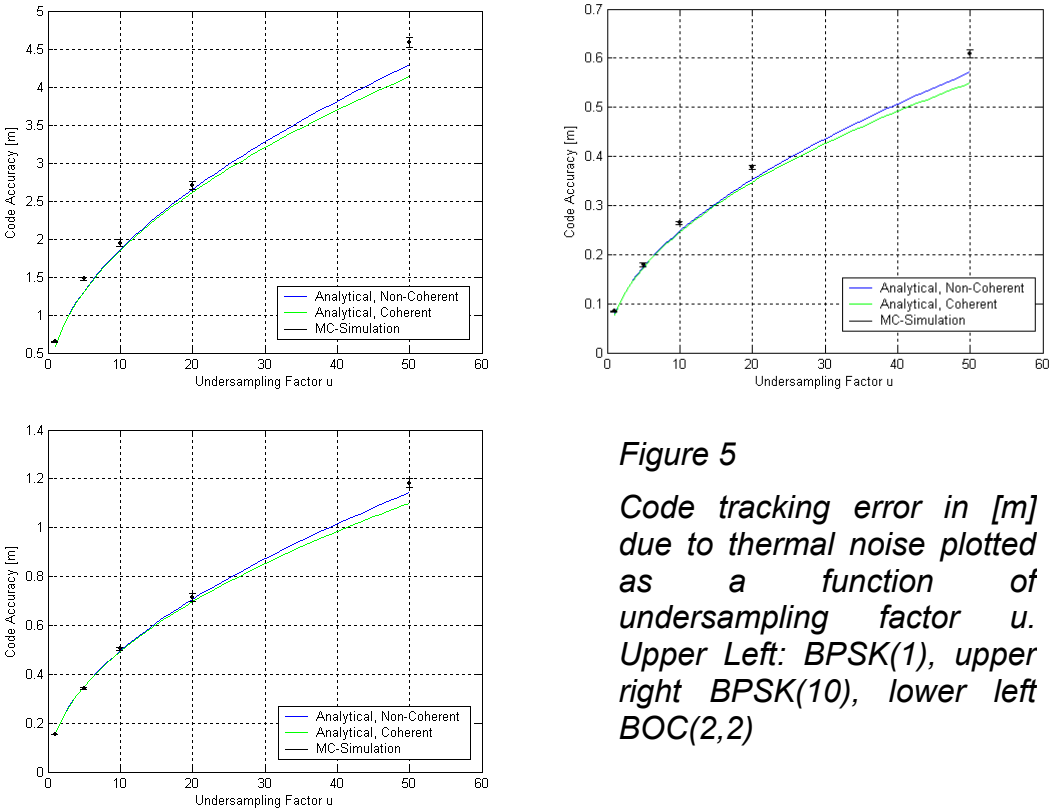

Figure 5

Code tracking error in [m] due to thermal noise plotted as a function of undersampling factor u. Upper Left: BPSK(1), upper right BPSK(10), lower left BOC(2,2)

In general the code accuracy increases with the square root of *u*, as the squaring loss is low for a predetection integration time of Table 1.

When comparing the three figures, it should be noted that the Nyquist rate (sample rate corresponding to *u*=1 or two times the dual-sided signal bandwidth) is different for all three cases. Table 3 lists all Nyquist rates and the undersampling factors corresponding to a common sample rate of 4 MHz.

Table 3 *Code and phase tracking accuracy*

	Nyquist rate	*u* for f_s=4 MHz	σ_{CODE} (f_s=4 MHz)	σ_{PHASE} (f_s=4 MHz)
BPSK(1)	8 MHz	2	83 cm	0.76 mm
BPSK(10)	40 MHz	10	25 cm	1.7 mm
BOC(2,2)	16 MHz	4	31 cm	1.1 mm

It is somehow surprising, that the code tracking accuracy of Table 3 is better for the large bandwidth BPSK(10) signal compared to the small bandwidth BPSK(1) signal, even if both are processed with the same sample rate. The shorter chip period (29 m) of

the BPSK(10) signal compared to the long chip period of 293 m for the BPSK(1) signal more than overcompensates the increase in the undersampling factor.

For a given sample rate and a given modulation scheme, one might question what is the optimal dual-sided signal bandwidth to achieve minimum thermal noise? To answer this question, we note, that the undersampling factor *u* is directly proportional to *B*. The second factor in (10) which depends on *B* is the second derivative of the autocorrelation function. For BPSK signals, its *B* dependency can be approximated as

$$R''(0) \propto \int_{-B/2}^{+B/2} f^2 S(f) df \propto B \quad \text{with} \quad S(f) \propto \frac{\sin\left(f\pi/f_c\right)^2}{f^2\pi^2}. \tag{13}$$

S(f) Power spectral density of a BPSK signal
$\propto$ Proportionality operator

Inserting these two relations into (10) we conclude, that the code thermal noise performance does **not** depend –to a first approximation– on the signal bandwidth *B* (assuming *u*>1, a fixed sample rate f_s, a fixed code rate and neglecting the squaring loss). In other words, when undersampling is employed, the code thermal noise error is determined mainly by the sample rate f_s and not by the signal bandwidth *B*.

Signal bandwidth is determined by satellite and receiver bandpass filters. We conclude, that for an undersampling receiver, the receiver bandpass filter needs not to have a small bandwidth; it can be chosen to optimize other criteria, for example to have a good multipath mitigation performance.

Finally we would like to note, that for *u*=1, *B* (and consequently the sample rate) should be chosen as large as possible to minimize thermal noise errors.

To which extend these statements apply for BOC signals, will be tested in the future.

4.2 PLL Thermal Noise Performance

The PLL tracking accuracy is computed similarly to the code accuracy, both analytically and by a MC simulation. Results are shown in Figure 6.

The phase tracking process, does not depend –to a good approximation– on the code modulation scheme, since the punctual code correlator removes the code structure from the navigation signal. Thus, phase tracking accuracy looks similar for all three cases and depends on the undersampling factor, the PLL bandwidth and the signal-to-noise ratio.

In Table 3 the phase accuracies for all three cases are shown for the same sample rate. Again a common C/N_0 of 45 dBHz is used. The best results are obtained for the BPSK(1) signal where the undersampling factor is smallest.

There is also a significant difference between the analytical result of (12) and the MC simulation. For the BPSK signals the MC values are approximately 1.30 times larger

than the analytical result and for the BOC signal the factor is about 1.25. The origin for this discrepancy has not been identified yet, but we think, that it originates from the Taylor series approximation of the PLL discriminator output.

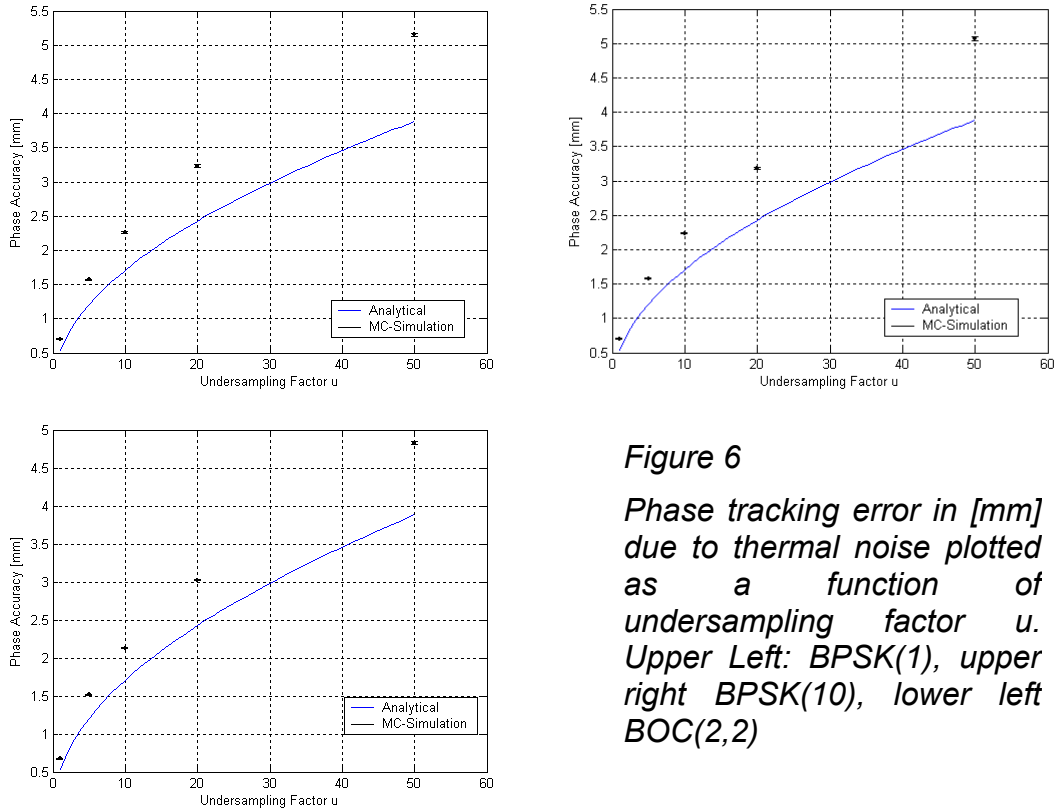

Figure 6

Phase tracking error in [mm] due to thermal noise plotted as a function of undersampling factor u. Upper Left: BPSK(1), upper right BPSK(10), lower left BOC(2,2)

5 Conclusion

With this work, we try to show, that a pure SWC GNSS receiver, i.e. a conventional PC without using FPGAs or ASICs to process digitized signals, will be an option for developing future navigation receivers. The concept of existing SWC C/A code receivers [12, 10] can be extended to high bandwidth signals, as they are foreseen for the modernized GPS and Galileo.

Due to limitations in the sample rate, the Nyquist criterion can not be fulfilled, which will increase the signal-to-noise ratio. Regarding the tracking process, an increased signal-to-noise ratio mainly affects the thermal noise error. Since this error source usually has a little contribution to the overall error budget, compared for example to multipath errors, there is some margin, which can be exploited to work with a lower sample rate. This will consequently reduce the processing demands on the processor.

An optimal code correlator, the Cramer-Rao correlator, has been designed in this work, and it has been shown, that its thermal noise performance equals the Cramer-Rao lower bound. Furthermore, code thermal noise variance does not depend on the signal bandwidth (due to bandpass filtering) but mainly on the sample rate. The Cramer-Rao correlator can be well implemented in software using look-up tables.

In future work, we will determine other performance parameters of the undersampling receiver, like multipath performance, acquisition time or data bit error rate.

6 References

[1] Betz J. and K. Kolodziejski (2000): Extended Theory of Early-Late Code Tracking for a Bandlimited GPS Receiver, Navigation, Vol. 47, No. 3, pp. 211-226.

[2] Van Dierendonck, A. and P. Fenton and T. Ford (1992): Theory and Performance of Narrow Correlator Spacing in a GPS receiver, Navigation Vol. 39, No. 3, pp. 265-283.

[3] Van Dierendonck A. (1996): GPS Receivers, In: *W. Parkinson and J. Spilker: GPS Positioning System Theory and Applications Vol., Progress in Astronautics and Aeronautics Vol. 163. American Institute of Aeoronautics and Astronautics.,* pp. 57-120.

[4] Van Dierendonck A. and C. Hegarty (2000): The New L5 Civil GPS Signal, GPS World, Vol. 11, No. 9, pp. 64-71.

[5] Eissfeller B. (1997): Ein dynamisches Fehlermodell für GPS Autokorrelations-empfänger. Schriftenreihe der Universität der Bundeswehr München, Heft 55.

[6] Eissfeller B., C. Tiberius, T. Pany, R. Biberger, T. Schueler and G. Heinrichs (2001): Real-Time Kinematic in the Light of GPS Modernization and Galileo, Proc. ION GPS 2001, Salt Lake City, September, pp. 650-662.

[7] Fontana R., W. Cheung, P. Novak and A. Thomas (2001): The New L2 Civil Signal: Proc. ION GPS 2001, Salt Lake City, September, pp. 617-631.

[8] Hein G., J. Godet, J. Issler, J. Martin, P. Erhard, R. Lucas-Rodriguez and T. Pratt (2002): Status of Galileo Frequency and Signal Design, Proc. of ION GPS 2002, Portland, September 24-27, 2002.

[9] Pany, T., M. Irsigler, B. Eissfeller and J. Winkel (2002): Code and Carrier Phase Tracking Performance of a Future Galileo RTK Receiver, Proc. of ENC-GNSS 2002, Copenhagen, May 2002.

[10] Schamus J., J. Tsui, D. Lin: Real-Time Software GPS Receiver, Proc. of ION GPS 2002, Portland, September 24-27, 2002.

[11] Spilker J. (1996): GPS Signal Structure and Theoretical Performance, In: *W. Parkinson and J. Spilker: GPS Positioning System Theory and Applications Vol., Progress in Astronautics and Aeronautics Vol. 163. American Institute of Aeoronautics and Astronautics.,* pp. 57-120.

[12] Thor, J., P. Normark and C. Ståhlberg: A High-Performance Real-Time GNSS Software Receiver and its Role in Evaluating Various Commercial Front End ASICs, Proc. of ION GPS 2002, Portland, September 24-27, 2002.

[13] Tsui J. (2000): Fundamentals of Global Positioning System Receivers - A Software Approach, Wiley Series in Microwave and Optical Engineering, New York.