

RESEARCH

Open Access

# Sub-carrier shaping for BOC modulated GNSS signals

Pratibha B Anantharamu\*, Daniele Borio and Gérard Lachapelle

## Abstract

One of the main challenges in Binary Offset Carrier (BOC) tracking is the presence of multiple peaks in the signal autocorrelation function. Thus, several tracking algorithms, including Bump-Jump, Double Estimator, Autocorrelation Side-Peak Cancellation Technique and pre-filtering have been developed to fully exploit the advantages brought by BOC signals and mitigate the problem of secondary peak lock. In this paper, the advantages of pre-filtering techniques are explored. Pre-filtering techniques based on the concepts of Zero-Forcing and Minimum Mean Square Error equalization are proposed. The BOC sub-carrier is modeled as a filter that introduces secondary peaks in the autocorrelation function. This filtering effect can be equalized leading to unambiguous tracking and allowing autocorrelation shaping. Monte Carlo simulations and real data analysis are used to characterize the proposed algorithms.

**Keywords:** binary offset carrier, BOC, equalization, global navigation satellite system, GNSS, MMSE, sub-carrier, zero-forcing

## 1 Introduction

Recent developments in the Galileo program have introduced a variety of new modulation schemes including the Binary Offset Carrier (BOC) [1] that has several advantages over traditional Binary Phase Shift Keying (BPSK) signals. BOC signals have increased resilience against multipath and provide improved tracking performance. However, they are characterized by autocorrelation functions (ACF) with multiple peaks that may lead to false code lock. This has led to the design of various BOC tracking algorithms such as Bump-Jump (BJ) [2], Autocorrelation Side-Peak Cancellation Technique (ASPeCT) [3] and its extensions [4], Double Estimator (DE) [5], Side Band Processing (SBP) [6] and pre-filtering [7].

In BJ, the BOC autocorrelation function is continuously monitored using additional correlators. A control logic detects and corrects false peak locks exploiting these additional correlators. In ASPeCT and its extensions, i.e., Side-lobes Cancellation Methods (SCM) [4], the BOC signal is correlated with its local replica and a modified local code. Thus, two correlation functions are computed: the first one is the ambiguous BOC autocorrelation, whereas the

second only contains secondary peaks. An unambiguous cost function is determined as a linear combination of the two correlations. The DE technique maps the BOC ambiguous correlation over an unambiguous bidimensional function [5]. The sub-carrier and the Pseudo-Random Number (PRN) code, the two components of a BOC signal, are tracked independently and an additional tracking loop for the sub-carrier is required. In SBP, the spectrum of BOC signals is split into side band components through modulation and filtering. Each side band component leads to unambiguous correlation functions. Non-coherent processing can be used for combining the results of the different processing branches [6]. The techniques mentioned above are characterized by different performance and different computational requirements. In this paper, pre-filtering techniques are considered for their generality and applicability to different contexts, such as unambiguous tracking and multipath mitigation. Pre-filtering techniques [7] are based on the fact that the spectrum of a signal can be modified by filtering. BOC signals are filtered in order to reproduce BPSK-like spectra and autocorrelations.

In this paper, a new class of pre-filtering techniques is derived from a convolutional representation of the transmitted signal. More specifically, the useful BOC-modulated signal is represented as the convolution of a

\* Correspondence: pratibha.ba@accord-soft.com  
Department of Geomatics Engineering, University of Calgary, 2500 University Dr NW, Calgary, AB T2N 1N4, Canada

Pseudo-Random Sequence (PRS) and a sub-carrier. The sub-carrier is interpreted as the equivalent impulse response of a selective communication channel that needs to be equalized. From this principle, filters analogous to the Zero-Forcing (ZF) and Minimum Mean Square Error (MMSE) equalizers [8] are derived. The proposed pre-filtering techniques shape the BOC ACF for unambiguous tracking and are herein called ZF Shaping (ZFS) and MMSE Shaping (MMSES). These techniques can be considered an extension of algorithms proposed in the communication context such as the mismatch filter (MMF) [9] and the 'CLEAN' algorithm [10]. The MMF operates on the temporal input data to obtain a desired sequence, whereas the 'CLEAN' algorithm works in the frequency domain to obtain a desired spectrum. In these techniques, a different signal structure was considered and the spectrum of the received signal was shaped for Inter Symbol Interference (ISI) cancellation. The problem of secondary autocorrelation peaks was not considered. In [7], several pre-filtering techniques were proposed. The filter design was however based on the combination of PRS and sub-carrier. This was causing severe noise amplification making the algorithms impractical for moderate to low signal-to-noise ratio conditions. In this paper, the noise amplification problem is mitigated using an innovative filter design based on the sub-carrier alone. The feasibility of the proposed algorithms is shown using live Global Navigation Satellite System (GNSS) data.

The filters for sub-carrier shaping are initially designed in the frequency domain. This approach requires a high processing load, and thus, a more computationally efficient time domain implementation is subsequently derived. A modified tracking loop architecture is also proposed to independently track code and carrier phase. Sub-carrier equalization performed for autocorrelation shaping is only required for unambiguous code tracking. Thus, the modified tracking architecture operates Phase Lock Loop (PLL) and Delay Lock Loop (DLL) independently. The filtered signal is exploited for generating the correlator outputs used for driving the DLL, whereas the unfiltered samples are exploited by the PLL. This further mitigates the noise amplification problem, since the PLL is unaffected by the filtering performed by the sub-carrier shaping algorithms.

Sub-carrier shaping algorithms are thoroughly analyzed and figures of merit such as tracking jitter, tracking threshold, Mean Time to Lose Lock (MTLL), tracking error convergence analysis and multipath error envelope (MEE) are introduced and adopted for performance evaluation. Although several unambiguous BOC tracking algorithms are present in the literature, only BJ and DE have been used as comparison terms. The BJ has been chosen because it has been one of the first algorithms

proposed for BOC tracking. In addition to this, its low computational requirements make it attractive for low complexity receivers. The DE technique has been selected for its close approximation to a matched filter and its improved performance in the absence of multipath. A comprehensive characterization of unambiguous BOC tracking algorithms is out of the scope of this paper. Additional material on the performance of BOC tracking techniques can be found in [4] and [11]. A comparison between standard pre-filtering techniques and ZFS is provided in [12] showing the superiority of the latter algorithm.

Real data from the second Galileo experimental satellite, GIOVE-B, have been used for extensively testing the proposed algorithms. Different Carrier-power-to-Noise-density ratios ( $C/N_0$ ) have been obtained using a variable gain attenuator. Signals from the GIOVE-B satellite have been progressively degraded simulating weak signal conditions.

From the tests and analysis, it is observed that MMSES provides a tracking sensitivity close to that provided by DE technique. When using real data, ZFS provides satisfactory results only for moderate to high  $C/N_0$ . This is due to the inherent noise amplification that can only be partially compensated for. On the other hand, MMSES is able to track weaker signals for a given bandwidth, leading to a performance close to that of the DE. Sub-carrier shaping provides satisfactory tracking performance maintaining the flexibility of pre-filtering techniques with the possibility of autocorrelation shaping. The slightly increased noise variance of the delay estimates is compensated by the flexibility of the algorithm that results in enhanced multipath mitigation capabilities. This work is an extension of the conference paper [12] that only considered the ZFS. The innovative contributions of the paper are the design of the MMSES algorithm and the novel implementation of pre-filtering techniques in time domain. In addition to this, separate carrier and code tracking is introduced to further mitigate the noise amplification problem. A thorough characterization of pre-filtering techniques is also provided.

The remainder of this paper is organized as follows: Section 2 introduces two different signal representations that are used as basis for the derivation of sub-carrier shaping algorithms. The basic principles of pre-filtering, BJ and DE are also briefly reviewed. Section 3 details sub-carrier shaping techniques, their time domain implementation and the modified tracking structure suggested for reducing the noise amplification problem. Section 4 provides a brief theoretical and computational analysis of the proposed pre-filtering techniques. Experimental setup, simulation and live data results are detailed in Section 5. Finally, some conclusions are drawn in Section 6.

## 2 Signal and system model

The complex baseband sequence at the input of a GNSS tracking loop can be modeled as the sum of a useful signal and a noise term,

$$\begin{aligned} y(t) &= x(t) + \eta(t) \\ &= A d(t - \tau_0) c(t - \tau_0) \exp\{j\theta_0(t)\} + \eta(t) \end{aligned} \quad (1)$$

where

- $A$  is the received signal amplitude;
- $d(\cdot)$  is the navigation message;
- $c(\cdot)$  is the ranging sequence used for spreading the transmitted data;  $c(\cdot)$  is usually made of several components and two different representations are discussed in the following;
- $\tau_0$  models the delay introduced by the communication channel whereas  $\theta_0(t)$  is used to model the phase variations due to the relative dynamics between receiver and satellite;
- $\eta(t)$  is a Gaussian random process whose spectral characteristics depend on the filtering and downconversion strategies applied at the front-end level.

In (1), the presence of a single useful signal is assumed. Although several signals from different satellites enter the antenna, a GNSS receiver is able to independently process each received signal, thus justifying model (1).

The ranging code,  $c(t)$ , is made of several components including a primary spreading sequence, a secondary or overlay code and a sub-carrier. In the following, the combination of primary sequence and overlay code will be denoted by  $p(t)$  and referred to as PRS. The ranging code can be expressed as:

$$c(t) = \sum_{i=-\infty}^{+\infty} p(iT_c) s_b(t - iT_c) \quad (2)$$

where  $s_b(\cdot)$  is the sub-carrier of duration  $T_c$ . Equation (2) can be interpreted in different ways leading to different signal representations.

### 2.1 Convolutional representation

Equation (2) can be represented as the convolution of the PRS with the sub-carrier signal:

$$\begin{aligned} c(t) &= \sum_{i=-\infty}^{+\infty} p(iT_c) s_b(t - iT_c) \\ &= \sum_{i=-\infty}^{+\infty} p(iT_c) \delta(t - iT_c) * s_b(t) = \tilde{p}(t) * s_b(t) \end{aligned} \quad (3)$$

where  $\tilde{p}(t)$  indicates a sequence of Dirac deltas,  $\delta(t)$ , modulated by the PRS. From (3), it is noted that  $s_b(t)$  acts as a filter that shapes the spectrum and

autocorrelation function of the useful signal. In Figure 1, the convolutional representation of the ranging code,  $c(t)$ , is better illustrated. The final ranging code is obtained by filtering the PRS modulated Dirac comb with the sub-carrier. In Figure 1, the case of a BOCs (1,1) is considered where

$$s_b(t) = \begin{cases} 1 & 0 \leq t \leq T_c/2 \\ -1 & T_c/2 < t \leq T_c \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

### 2.2 Multiplicative representation

An alternative representation of the ranging code,  $c(t)$ , is given by

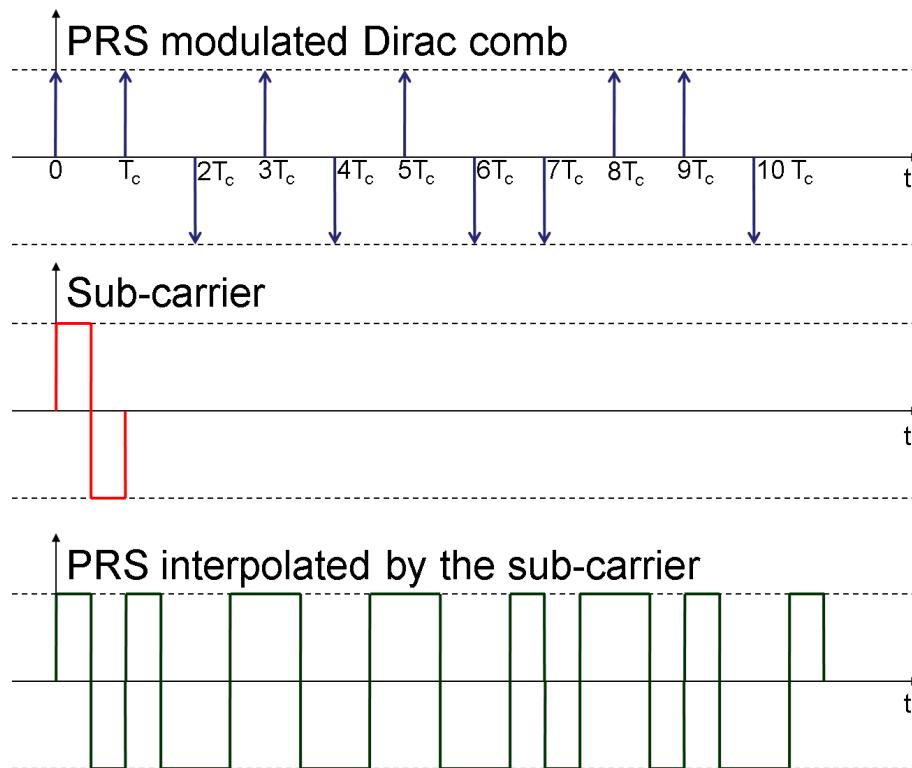
$$\begin{aligned} c(t) &= \sum_{i=-\infty}^{+\infty} p(iT_c) s_b(t - iT_c) \\ &= \sum_{i=-\infty}^{+\infty} p(iT_c) s_{\text{BPSK}}(t - iT_c) \cdot \sum_{k=-\infty}^{+\infty} s_b(t - kT_c) \\ &= c_{\text{BPSK}}(t) \tilde{s}_b(t) \end{aligned} \quad (5)$$

where  $s_{\text{BPSK}}(t)$  is the BPSK sub-carrier and is equal to a rectangular window of duration  $T_c$ ,  $\tilde{s}_b(t)$  is the signal obtained by periodically repeating the subcarrier  $s_b(t)$  and  $c_{\text{BPSK}}(t) = \sum_{i=-\infty}^{+\infty} p(iT_c) s_{\text{BPSK}}(t - iT_c)$ . Representation (5) is based on the bipolar nature of the components of the ranging code,  $c(t)$ , and is better illustrated in Figure 2 where a BPSK modulated PRS is multiplied by the periodic repetition of the sub-carrier. It is noted that the final signal obtained in Figure 2 is equal to the one in Figure 1. The multiplicative representation is reported here for a better understanding of the DE that is used as a comparison term for the proposed pre-filtering techniques.

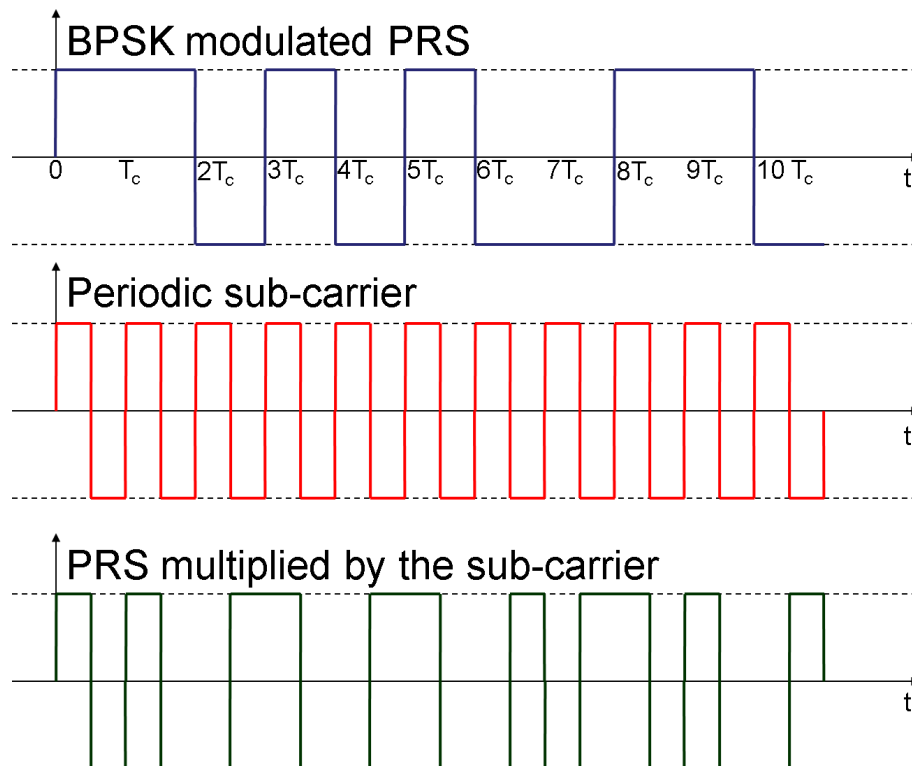
### 2.3 The correlation process

The main operation performed by a GNSS receiver consists in correlating the input signal,  $y(t)$ , with a locally generated replica. Correlation allows the reduction of the input noise and the extraction of the signal parameters. The local signal replica is obtained by generating a complex carrier that is used for recovering the effect of the signal phase,  $\theta(t)$ , and a local ranging code  $c_l(t) = c(t)$ . The  $k$ th correlator output,  $Q_k$ , for a given code delay,  $\tau$ , and carrier phase,  $\theta(t)$ , can be expressed as

$$\begin{aligned} Q_k(\tau, \theta) &= \int_{(k-1)T_i}^{kT_i} y(t) c_l(t - \tau) \exp\{-j\theta(t)\} dt \\ &= A \int_{(k-1)T_i}^{kT_i} d(t - \tau_0) c(t - \tau_0) \exp\{j\theta_0(t)\} c_l(t - \tau) \exp\{-j\theta(t)\} dt + \tilde{\eta} \\ &= A \int_{(k-1)T_i}^{kT_i} d(t - \tau_0) c(t - \tau_0) \exp\{j\Delta\theta(t)\} c_l(t - \tau) dt + \tilde{\eta} \end{aligned} \quad (6)$$



**Figure 1** Convolutional representation of the ranging code,  $c(t)$ . The useful signal is obtained by filtering the PRS modulated Dirac comb with a sub-carrier filter.



**Figure 2** Multiplicative representation of the ranging code,  $c(t)$ . The useful signal is obtained by multiplying the BPSK modulated PRS by the periodic repetition of the sub-carrier.

where  $\Delta\theta(t) = \theta_0(t) - \theta(t)$ .  $T_i$  is the coherent integration time and  $\tilde{\eta}$  is a noise term obtained by processing the input noise,  $\eta(t)$ . In this paper, it is assumed that the receiver is able to perfectly recover the signal phase, and so  $\Delta\theta(t) = 0$ . Assuming the navigation message,  $d(t)$ , constant during the integration period, Eq. (6) simplifies to

$$\begin{aligned} Q_k(\tau) &= A \int_{t=(k-1)T_i}^{kT_i} c(t - \tau_0) c_l(t - \tau) dt + \tilde{\eta} \\ &= AR(\tau_0 - \tau) + \tilde{\eta} \\ &= AR(\Delta\tau) + \tilde{\eta} \end{aligned} \quad (7)$$

where  $R(\Delta\tau)$  is the correlation function between the incoming and locally generated signal. The shape of  $R(\Delta\tau)$  is essentially determined by the signal sub-carrier. For a BPSK signal,  $R(\Delta\tau)$  is characterized by a single peaked triangular function. But when a BOC is used,  $R(\Delta\tau)$  is characterized by several secondary peaks that can lead to false code locks.

Several techniques have been developed on the basis of the multiplicative and convolutional representations described above. Figure 3 shows the basic principles of different BOC tracking techniques designed on the basis of the mentioned representations. In the DE technique, the transmitted signal is assumed to be generated using the multiplicative representation detailed in Section 2.2. The received signal after passing through the transmission channel is correlated with a periodic version of the sub-carrier. This is achieved by generating a local sub-carrier,  $\tilde{s}_b(t)$  and estimating the sub-carrier delay

introduced by the communication channel. When the delay of the locally generated sub-carrier matches the sub-carrier delay of the incoming signal, the sub-carrier effect is completely removed from the ranging code and a BPSK-like signal is obtained.

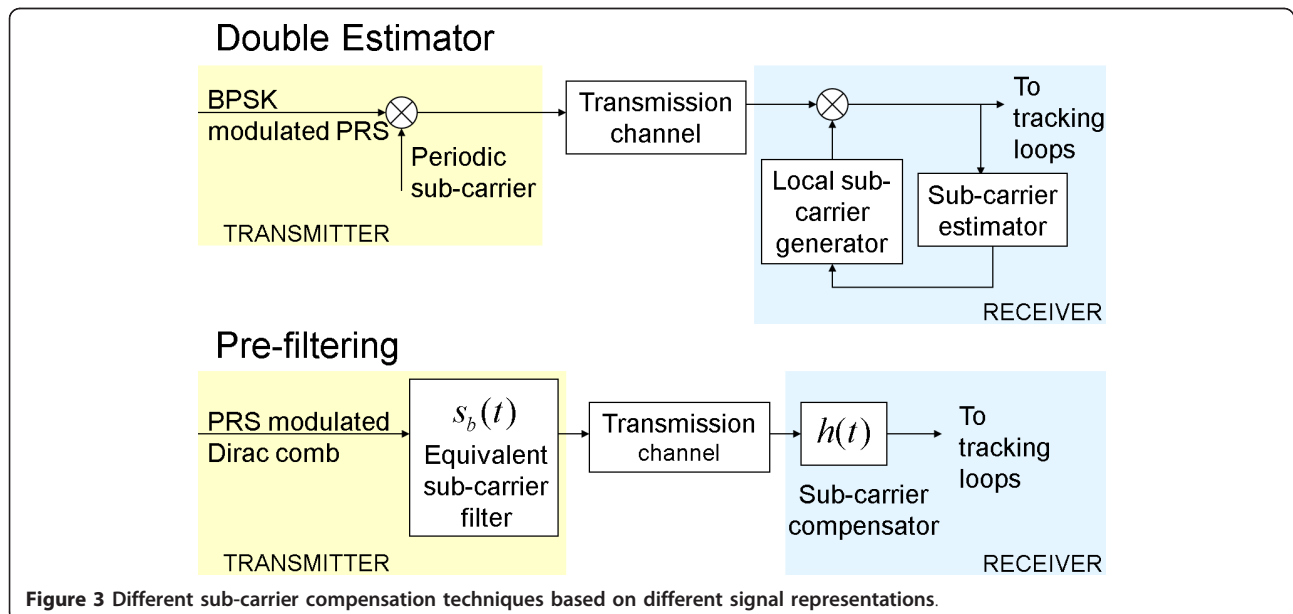
In the pre-filtering case, the transmitted signal is assumed to be generated using the convolutional representation described in Section 2.1. The sub-carrier effect is alleviated using a filter denoted sub-carrier compensator,  $h(t)$ . These techniques exploit the fact that the sub-carrier effect can be removed by filtering the ranging code

$$c(t) * h(t) = \tilde{p}(t) * s_b(t) * h(t) = \tilde{p}(t) * s_h(t) \quad (8)$$

with the objective to make the filtered sub-carrier,  $s_h(t) = s_b(t) * h(t)$ , have a correlation function without side-peaks. The third BOC tracking technique considered is the BJ [2] based on post-correlation techniques. These techniques do not directly operate on the signal but on the correlation function and they require additional correlators that are used for monitoring the code lock condition.

### 3 Sub-carrier shaping

In communications, the effect of a frequency selective transmission channel is usually compensated by the adoption of equalization techniques. In the considered research, the effect of sub-carrier is interpreted as a selective communication channel that distorts the useful signal. Thus, a similar equalization approach can be adopted for mitigating the impact of the sub-carrier. The convolutional representation of BOC signals is used here as basis to derive sub-carrier equalizers to shape the BOC ACF.



**Figure 3** Different sub-carrier compensation techniques based on different signal representations.



### 3.1 MMSES

The main goal of MMSES is to produce an output signal with unambiguous ACF. A BPSK-like spectrum is thus the desired signal spectrum and the transfer function of the MMSES,  $H(f) = \mathcal{F}\{h(t)\}$ , needs to be designed accordingly. Here,  $\mathcal{F}$  denotes the Fourier transform operation. The solution leading to  $H(f)$  is given by the MMSE approach that minimizes the following cost function [8]:

$$\varepsilon_{\text{MMSES}} = \int_{-B}^B \left[ |G_D(f) - G_x(f)H(f)|^2 + \frac{\lambda N_0}{C} G_x(f) |H(f)|^2 \right] df \quad (9)$$

where

- $G_D(f)$  is the desired signal spectrum. Its inverse Fourier transform is the desired correlation function;
- $G_x(f)$  is the Fourier transform of the correlation between incoming and local signals.  $G_x(f)$  and  $G_D(f)$  have been normalized in order to have unit integral;
- $G_L(f)$  is the spectrum of the local code;
- $N_0$  is the power spectral density (PSD) of  $\eta(t)$ , the input noise is assumed to be white within the receiver bandwidth;
- $\lambda$  is a constant factor used to weight the noise impact;
- $B$  is the receiver front-end bandwidth;

It is noted that  $\varepsilon_{\text{MMSES}}$  incorporates two terms. The first is the mismatch between desired and actual correlation functions, whereas the second is the noise variance after correlation and filtering. This second term is multiplied by the inverse of the  $C/N_0$  in order to account for the relative impact of signal and noise components. The division by  $C$  in the second term of (9) is due to the

normalization adopted for  $G_x(f)$  and  $G_D(f)$ . The factor  $\lambda$  allows one to weight the relative contribution of the two terms. Under the assumption that the local code is matched to the incoming signal,  $G_x(f) = G_L(f)$ , (9) reduces to

$$\varepsilon_{\text{MMSES}} = \int_{-B}^B \left[ |G_D(f) - G_x(f)H(f)|^2 + \frac{\lambda N_0}{C} G_x(f) |H(f)|^2 \right] df \quad (10)$$

and the error in (10) is minimized by

$$H(f) = \frac{G_D(f)}{G_x(f) + \frac{\lambda N_0}{C}}. \quad (11)$$

ZFS is a special case of MMSES in which the noise effect is ignored. Setting  $\lambda = 0$  in (11) results in the ZFS algorithm:

$$H(f) = \frac{G_D(f)}{G_x(f)}. \quad (12)$$

In (12),  $G_x(f)$  can contain zeros that would make  $H(f)$  diverge to infinity. This is avoided by clipping the amplitude of  $H(f)$  to certain limits, thus removing the singularities in  $G_x(f)$ .

MMSES was performed on BOC(1,1) to obtain an unambiguous ACF. Figure 4 shows the ACF obtained after applying MMSES on Intermediate Frequency (IF) simulated data. The input  $C/N_0$  was set to 40 dB-Hz and the ACF was averaged over 1 s of data. From Figure 4, it can be observed that the multi-peaked BOC ACF (indicated as 'Standard') was successfully modified by MMSES to produce a BPSK-like ACF without secondary peaks. Similar results were obtained for BOCc(10,5) and BOCc(15,2.5), as shown in Figure 5. The results in Figure 5 shows the flexibility of MMSES to provide unambiguous

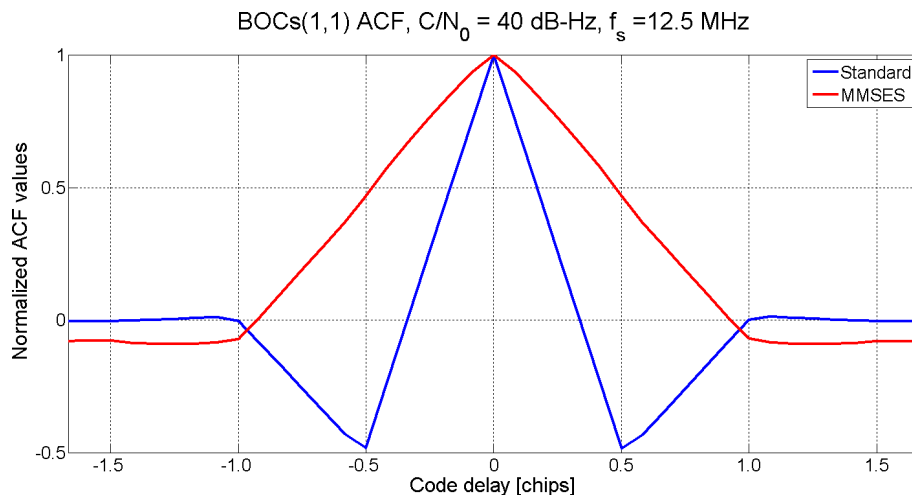
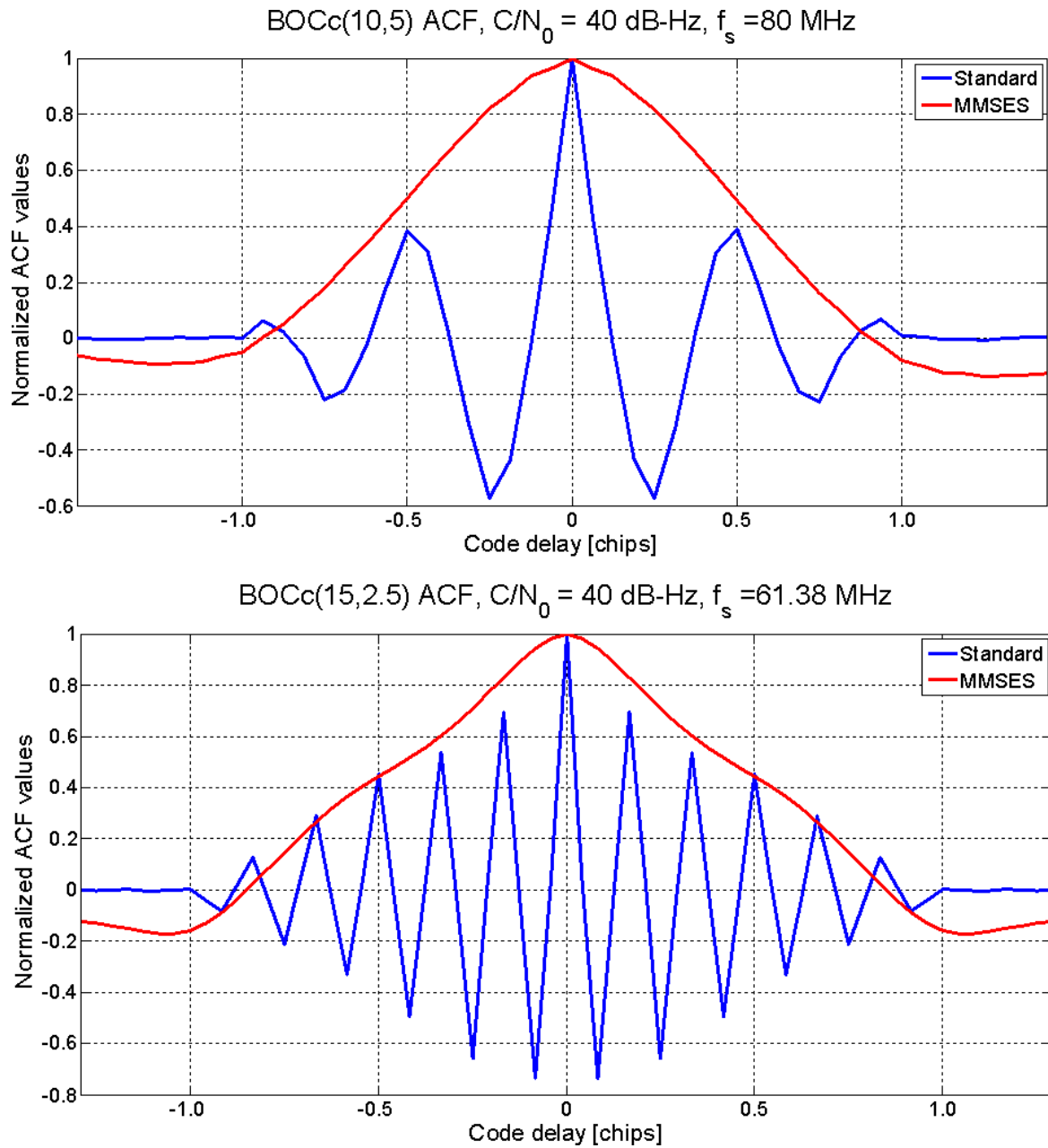


Figure 4 BOCs(1, 1) ACF with standard BOC sub-carrier and after MMSES.



**Figure 5** Autocorrelation functions of BOCc(10, 5) and BOCc(15, 2.5) before and after applying MMSES.

ACF for higher sub-carrier rate ratios of the BOC family. The sub-carrier rate ratio for BOCc(10,5) is 2, while that of BOCc(15,2.5) is 6. Although the theory provided above has been developed in the continuous time domain, the algorithms have been practically implemented using digital versions of the incoming and local signals. For this reason, the correlation functions in Figures 4 and 5 are sampled with a sampling frequency  $f_s$ .

In the proposed approach, it is assumed that the spectrum of the different signal components is essentially

determined by the Fourier transform of the local and desired sub-carriers. More specifically, the following assumptions are made

$$G_D(f) = |S_D(f)|^2, G_x(f) = G_L(f) = |S_b(f)|^2 \quad (13)$$

where  $S_D(f)$  is the Fourier transform of the desired sub-carrier,  $s_D(t)$ , and  $S_b(f)$  is the Fourier transform of the local sub-carrier,  $s_b(t)$ . Condition (13) implies that the spectrum of the PRS modulated Dirac comb can be

effectively approximated as a Dirac delta. This approach is similar to the methodology described in [12] and allows the design of shaping filters independent from the PRS. This approach has been proven to be more effective than other pre-filtering techniques in mitigating the noise amplification problem [12]. The main advantage of the proposed ZFS and MMSES is the ability to reshape the autocorrelation function. This can be used for multipath mitigation. This clearly appears in Figures 6 and 7 where live BOCs(1, 1) signals from the GIOVE-B satellite have been used. The desired autocorrelation functions for the ZFS and MMSES are obtained by changing the spectrum of the desired signal. From Figures 6 and 7, it can be noted that the base width of the autocorrelation function is reduced by decreasing the duration,  $T_d$ , of the desired sub-carrier,  $s_D(t)$ . From Figure 7, the advantage of MMSES over ZFS clearly appears: the secondary lobes of the MMSES ACF are clearly attenuated with respect to the ZFS case. This is due to the ability of MMSES to mitigate the noise amplification problem. This shows the advantage of using the ZFS and MMSES over the DE technique. In the DE technique, the autocorrelation function is fixed whereas in pre-filtering, the autocorrelation function can be selected according to different applications.

In the following,  $\lambda$  will be set to 1 and  $N_0$  is adapted according to the input  $C/N_0$  and scaling applied to the signal power density,  $G_x(f)$ . Comparison of ZFS and existing pre-filtering techniques [7] have been performed

in [12] and the analysis proved that ZFS is able to successfully compensate for secondary autocorrelation peaks, whereas standard approaches are unable to mitigate secondary peak locks for moderate to low  $C/N_0$  values. Since standard pre-filtering techniques [7] are outperformed by ZFS, they would not be further considered in the remainder of this paper. The interested reader is referred to the findings presented in [12].

### 3.2 Time domain implementation

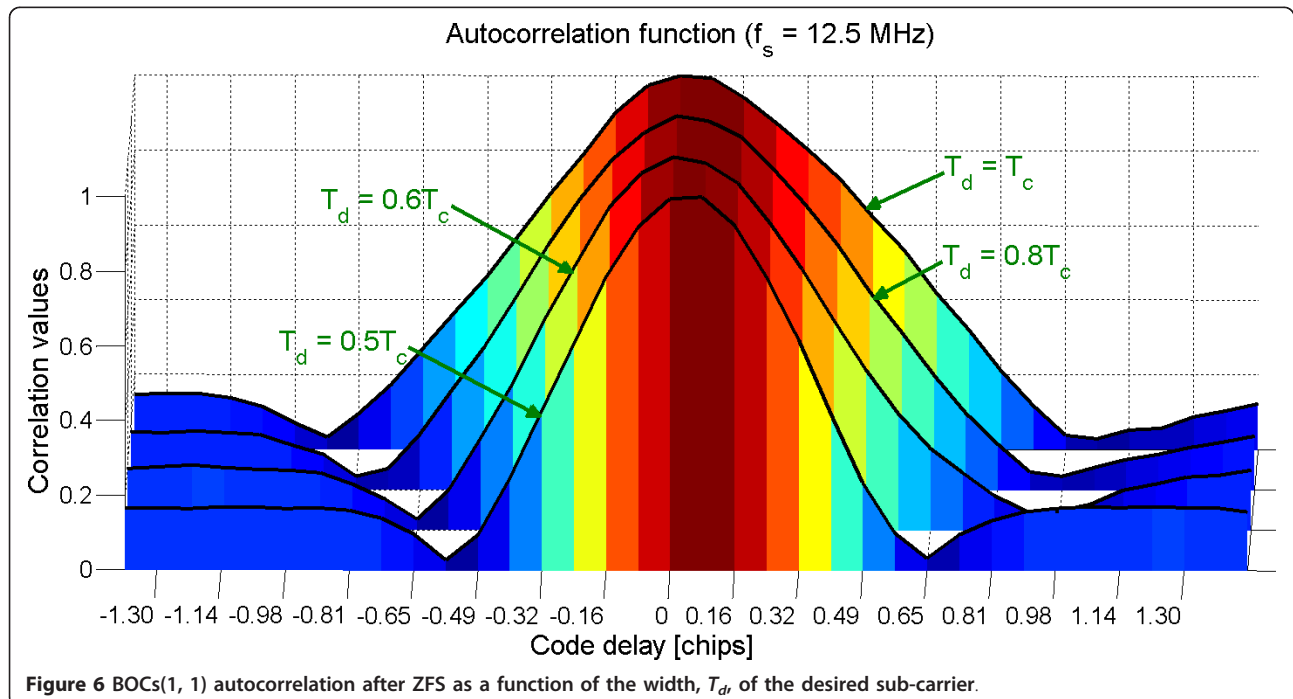
The development of both ZFS and MMSES has been performed at first in the frequency domain as discussed in Section 3.1. The processing load required to track signals in the frequency domain is significant since it involves Fourier transform operations (Fast Fourier Transforms, FFTs, in the discrete time domain). Hence, a more efficient time domain implementation, requiring the evaluation of only three correlators, has been developed. The final correlator output after frequency domain processing can be expressed as

$$Q(\tau) = \mathcal{F}^{-1} \{ \mathcal{F} \{ y(t) \} \cdot H(f) \cdot \mathcal{F} \{ c_l(t) \}^* \} |_{t=\tau} \quad (14)$$

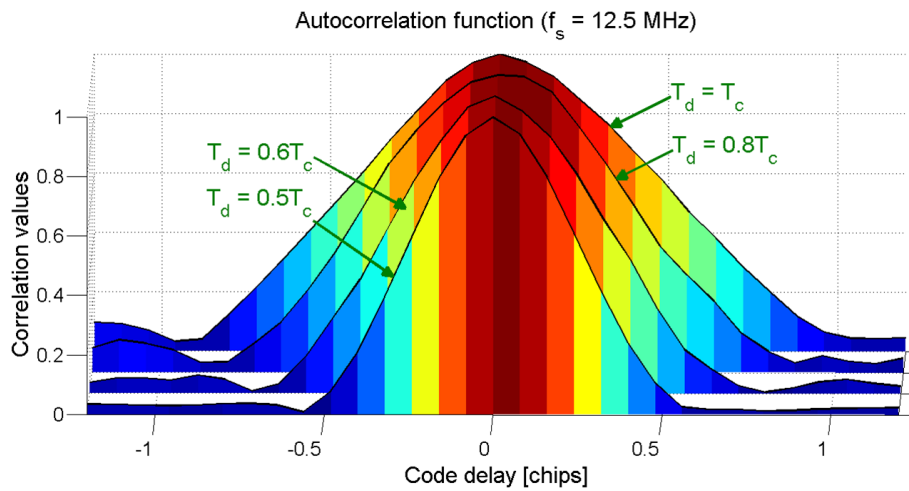
where  $\mathcal{F}^{-1} \{ \cdot \}$  is the Inverse Fourier transform.

Rearranging the terms in (14), the filtering operation can be performed solely on the local signal as

$$Q(\tau) = \mathcal{F}^{-1} \{ \mathcal{F} \{ y(t) \} \cdot \mathcal{F} \{ \tilde{c}_l(t) \}^* \} |_{t=\tau} \quad (15)$$







**Figure 7** BOCs(1, 1) autocorrelation after MMSES as a function of the width,  $T_d$ , of the desired sub-carrier.

where

$$\tilde{c}_l(t) = \mathcal{F}^{-1} \{ H^*(f) \cdot \mathcal{F}\{c_l(t)\} \} \quad (16)$$

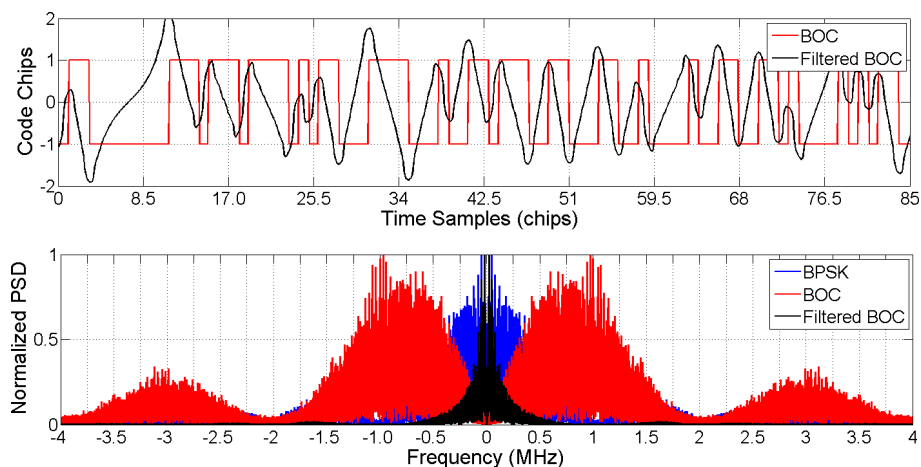
is an equivalent code accounting for the filtering performed by  $H(f)$ .

In this way, pre-filtering can be implemented as the time domain correlation with a modified local code. It is noted that the receiver has to allow multi-level correlation. More specifically,  $\tilde{c}_l(t)$  is no longer a binary sequence. The modified local code along with its PSD before and after pre-filtering is shown in Figure 8. The PSD plot shows that the dual-lobed BOC spectrum is replaced by a single-lobe narrow spectrum after filtering. The main advantage of using (16) to perform time domain filtering is the reduced computational complexity. The Fourier transform and the operations in the frequency domain are replaced by three correlators, Early,

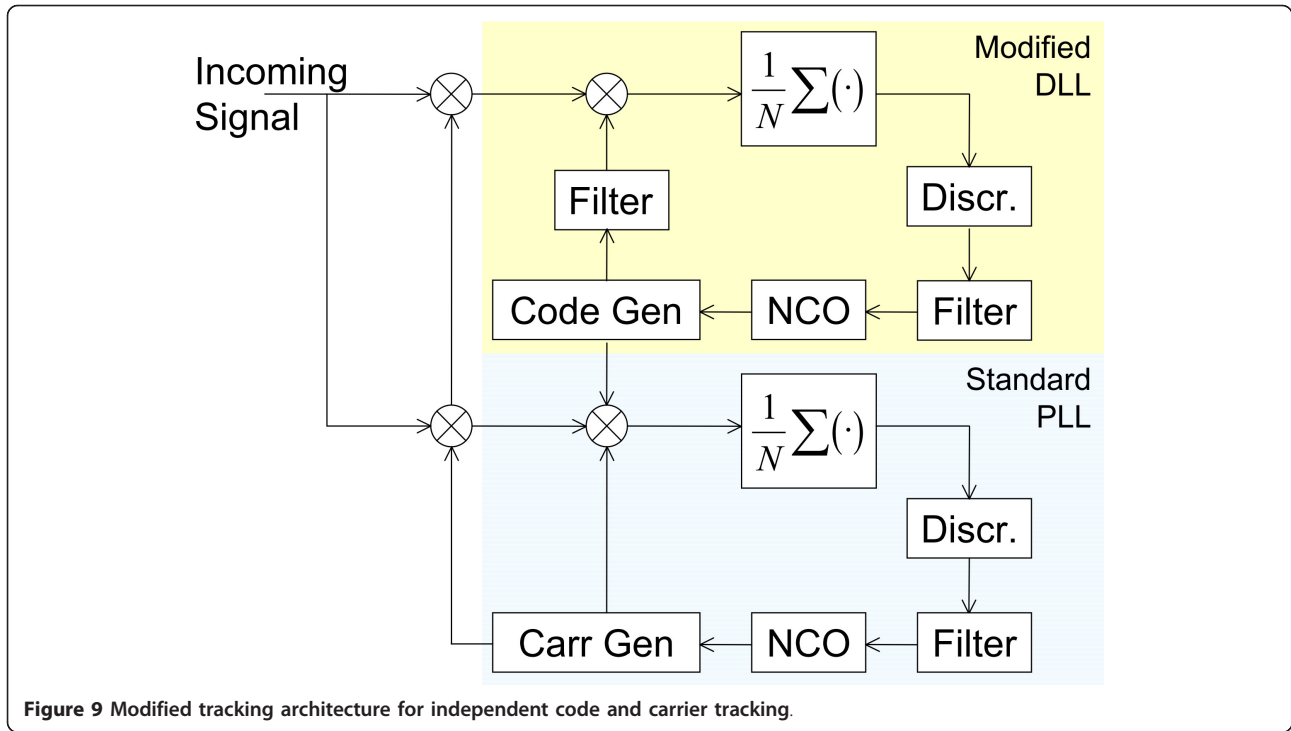
Prompt and Late codes, directly computed in the time domain.

### 3.3 Delay and phase independent tracking

The PLL is always the weakest link in a GNSS receiver [13] and filtering further amplifies the input noise degrading the PLL performance and resulting in a poor tracking sensitivity. For weak signal environments, it would be beneficial if the PLL and filtering process were independent. For this reason, a new architecture, using independent correlators for PLL and DLL has been developed. The proposed architecture is shown in Figure 9. Here, the DLL is driven by the filtered correlators ensuring unambiguous code tracking. On the other hand, the PLL is driven by an additional unfiltered correlator. In this way, the PLL is unaffected by the noise amplification caused by pre-filtering. Attenuated live signals from



**Figure 8** Filtering effects on BOC signal and spectrum.



GIOVE-B satellites were used to verify the effect of the modified tracking structure. PLL driven by the unfiltered correlator provided a 5 dB better performance compared with the one driven by filtered correlator.

#### 4 Algorithm characterization

##### 4.1 Theoretical analysis

The filter used to shape the signal autocorrelation modifies the signal and noise properties. More specifically, a loss in the signal-to-noise ratio (SNR) at the correlator output is introduced. This effect is the already mentioned noise amplification problem, and its impact can be determined using an approach similar to the one adopted by [14-16].  $H(f)$  generates a colored noise and the post-correlation SNR becomes [14,15]:

$$\text{SNR} = \frac{C}{N_0} T_i \gamma \quad (17)$$

where  $\gamma$  is the filtering loss equal to

$$\gamma = \frac{\left| \int_{-B}^B G_x(f) H(f) df \right|^2}{\int_{-B}^B G_x(f) |H(f)|^2 df}. \quad (18)$$

It is noted that the numerator and denominator in (18) are the signal and noise terms of the cost function (9). The MMSE tries to find a compromise between making  $G_x(f) H(f)$  as close as possible to the desired

spectrum,  $G_D(f)$ , and reducing the noise term at the denominator of (18).

If the amplitude of the Prompt correlator output is assumed to be normalized to unity, the inverse of (18) determines the variance of the post-correlation noise components:

$$\sigma_n^2 = \frac{1}{C/N_0 T_i \gamma}. \quad (19)$$

The signal component after correlation is proportional to the filtered correlation function

$$R(\Delta\tau) = \mathcal{F}^{-1} \{G_x(f) H(f)\} \big|_{t=\Delta\tau} \quad (20)$$

whereas the noise components of different correlator outputs are characterized by a correlation coefficient equal to

$$R_n(\Delta\tau) = \mathcal{F}^{-1} \{G_x(f) |H(f)|^2\} \big|_{t=\Delta\tau}. \quad (21)$$

In (20),  $\Delta\tau$  denotes the additional delay used for computing a specific correlator.  $\Delta\tau = 0$  for the Prompt correlator and  $\Delta\tau = \pm d_s/2$  for Early and Late correlators.  $d_s$  is the Early-Late correlator spacing. In (21),  $\Delta\tau$  is used to denote the delay difference between two correlators. Early and Late are separated by a delay equal to  $d_s$ , whereas the Prompt correlator is characterized by a delay difference equal to  $d_s/2$  with respect to the other correlators.

The results listed above can be used for computing the tracking jitter. The tracking jitter is one of the most used metrics for determining the quality of estimates produced by tracking loops. More specifically, the tracking jitter quantifies the residual amount of noise present in the final loop estimate, in this case the code delay [17]. The tracking jitter is directly proportional to the standard deviation of the tracking error defined as the difference between true and estimated tracking parameters. A large tracking jitter indicates poor quality measurements and a large uncertainty in the estimated parameters.

The tracking jitter can be computed as [17]:

$$\sigma_j = \frac{1}{G_d} \sqrt{2B_{eq}T_i\sigma_d^2} \quad (22)$$

where  $B_{eq}$  is the loop equivalent bandwidth and  $\sigma_d^2$  is the variance of the discriminator output. In a tracking loop, the correlator outputs are combined in a nonlinear way by a discriminator that produces a control signal. The filtered version of this control signal is used to correct the loop estimates and maintain lock conditions [13].  $G_d$  is the discriminator gain defined as

$$G_d = \left. \frac{\partial E[D(\Delta\tau)]}{\partial \Delta\tau} \right|_{\Delta\tau=0} \quad (23)$$

where  $D(\Delta\tau)$  defines the discriminator input-output function.

In a coherent discriminator,

$$D(\Delta\tau) = \text{Re}\{E - L\} \quad (24)$$

where  $E$  and  $L$  denote the complex Early and Late correlators. Using (20), it is possible to show that for a coherent discriminator

$$G_d = -2\dot{R}(d_s/2) \quad (25)$$

where the symbol  $\dot{x}$  denotes the first derivative. In addition to this,

$$\text{Var}\{\text{Re}\{E - L\}\} = \frac{1}{2}[\text{Var}\{E\} + \text{Var}\{L\} - 2\text{Cov}\{E, L\}] = \sigma_n^2[1 - R_n(d_s)]. \quad (26)$$

From these results, it is finally possible to determine the tracking jitter for a coherent Early minus Late discriminator in the presence of pre-filtering:

$$\sigma_j = \sqrt{\frac{B_{eq}T_i\sigma_n^2(1 - R_n(d_s))}{2\dot{R}(d_s/2)^2}} = \sqrt{\frac{B_{eq}(1 - R_n(d_s))}{C/N_0\gamma 2\dot{R}(d_s/2)^2}}. \quad (27)$$

The tracking jitter for the quasi-coherent dot-product and the non-coherent early minus late power discriminators [13] can be determined using a similar approach. The theoretical formulas for the tracking jitter for the different discriminators are reported in Table 1.

## 4.2 Computational analysis

The computational complexity of the considered algorithms is detailed in the following. Table 2 summarizes the computational complexity of pre-filtering, BJ and DE. The computation of the correlator outputs is the most demanding task of a GNSS receiver. Thus, the computational complexity is determined as a function of the number of required correlations. The final execution speed of each algorithm depends on the hardware specifications of the platform where the techniques are implemented. For example, modern general purpose processors and DSPs are able to perform real multiplications in a single clock cycle making pre-filtering an attractive solution in terms of computational complexity. The different algorithms have been implemented in MATLAB and tested using live GIOVE-B data. An indication of the effective computational time required by each technique is provided in Table 3 where the average times required to process a second of data by the different techniques is reported. It is noted that the code implementing the different algorithms was not designed

**Table 1 Theoretical tracking jitter for different discriminator types**

Discriminator ( $D$ )	Tracking jitter ( $\sigma_j$ )
Coherent $\text{Re}\{E - L\}$	$\sqrt{\frac{B_{eq}T_i\sigma_n^2(1 - R_n(d_s))}{2\dot{R}(d_s/2)^2}}$
Quasi-coherent dot-product $\text{Re}\{(E - L)P^*\}$	$\sqrt{\frac{B_{eq}T_i\sigma_n^2(1 - R_n(d_s))}{2\dot{R}(d_s/2)^2}}(1 + \sigma_n^2)$
Non-coherent early minus late power $ E ^2 -  L ^2$	$\sqrt{\frac{B_{eq}T_i\sigma_n^2(1 - R_n(d_s))}{2\dot{R}(d_s/2)^2}} \left[ 1 + \sigma_n^2 \frac{(1 + R_n(d_s))}{2R_n^2(d_s/2)} \right]$

**Table 2 Computational complexity of pre-filtering, BJ and DE**

Algorithm	Number of complex correlators	Notes
BJ	5–bipolar/binary	The local code is a bipolar sequence and code multiplication can be effectively implemented using sign changes. Additional logic/circuitry is required for the generation of the local subcarrier replica. The number of multiplications is double since local code and sub-carrier are wiped-off separately.
DE	5–bipolar/binary	
MMSES (time domain implementation and independent phase tracking)	3–real 1–bipolar/binary (for independent phase tracking)	The filtered local code is stored in memory and multi-bit multiplications are required for the code wipe-off.

for real-time operations; however, the results in Table 3 provide an indication of the relative complexity of the three techniques. The values in Table 3 have been obtained using MATLAB directives for measuring the execution time of a single loop update including the computation of the different correlator outputs. A 5 min long data set was used to average the processing times reported in Table 3. The characteristics of the input signal are summarized in Table 4. From Table 3, it emerges that the time domain implementation of the MMSES is less computationally demanding than the DE. In addition to this, the MMSES allows one to implement multipath mitigation capabilities without increasing the computation load. This is achieved by changing the filter used for code shaping.

## 5 Simulation and real data analysis

In this section, ZFS and MMSES are analyzed and compared against the DE [5] and BJ [2] techniques for BOCs(1,1) modulated signals in terms of tracking jitter, tracking threshold, MTLL, code error convergence and MEE for different Early-minus-Late chip spacing and discriminator types. The analysis is based on the semi-analytic technique described in [18].

In a semi-analytic approach, the analytical knowledge of the system is used to reduce the computational load that a full Monte Carlo approach would require [19]. In a GNSS code tracking loop, correlation is the most computationally demanding task. At the same time, it consists of simple linear operations and the correlator outputs can be easily determined in an analytical way from the  $C/N_0$  and the delay error. Thus, it is possible to simulate all the operations from the correlator

outputs to the code delay update performed by the NCO. Analytical results are used to determine the correlator outputs closing the analysis/simulation loop [18]. This approach has been widely used in GNSS, as indicated in [18] and in its references. The signal parameters used for the semi-analytic analysis are provided in Table 5.

## 5.1 Simulation results

### 5.1.1 Tracking jitter

In this section, the tracking jitter for different BOC tracking techniques have been provided. Different chip spacings,  $d_s = 0.2, 0.3$  and  $0.4$  chips, have been considered along with non-coherent, quasi-coherent and coherent discriminators [13]. The non-coherent discriminator is analyzed in detail, whereas only sample results are shown for the other two cases.

The tracking jitter of MMSES with a non-coherent discriminator is shown in Figure 10 as a function of the input  $C/N_0$  and for different chip spacing. The semi-analytic models used for the generation of these curves is described in [18]. It is noted, that for low  $C/N_0$ s, the three curves diverge. This is due to the fact that the loop is losing lock and the loop discriminator is working in its nonlinear region. As already pointed out, pre-filtering techniques enhance the noise present on the correlator outputs and this fact is reflected on the tracking jitter. In [12], it was observed that ZFS performs poorly for a medium to low  $C/N_0$  and the tracking jitter is always higher than the one obtained for the DE tracking technique. The code tracking jitter due to MMSES is lower as compared to ZFS. This is an indication of the ability of MMSES to mitigate the noise impact.

**Table 3 Average processing time per second of data for different unambiguous boc tracking techniques**

Algorithm	Average processing time (per second of data)
BJ	6.8 s
DE	7.6 s
MMSES (independent phase tracking)	Frequency domain: 10.8 s Time domain: 7.3 s

**Table 4 Parameters of the real data used for the computational analysis**

Parameter	Value
Sampling frequency	12.5 MHz
Intermediate frequency	3.42 MHz
Data duration Sampling	5 min 8 bit real samples

MMSES performs poorly for low  $C/N_0$ , but the tracking jitter is always lower than ZFS.

In Figures 11 and 12, ZFS and MMSES are compared with DE and BJ technique where quasi-coherent and coherent discriminators are used. It is noted that the MMSES is able to maintain lock for almost the same  $C/N_0$  level as the DE. In this respect, the MMSES clearly outperforms the BJ. The ability of the MMSES of shaping the BOC ACF is paid by a slight tracking jitter degradation. This loss of performance becomes however negligible for  $C/N_0$  values greater than 30 dB-Hz.

#### 5.1.2 Tracking threshold

The tracking threshold is the minimum  $C/N_0$  value at which a tracking loop is able to maintain a stable lock [13]. The tracking thresholds of the three considered BOC tracking techniques are compared in Figure 13 for different types of loop discriminators. As expected, improvements on all the three techniques are observed when moving from a non-coherent to a coherent discriminator. MMSES efficiently mitigates the noise amplification problem, leading to a tracking threshold comparable to that achieved by the DE. It is noted that the tracking threshold for the BJ seems to be unaffected by the type of discriminator. This can be an indication that, in the BJ case, loss of lock is determined by the control logic for detecting secondary peak lock. The same decision logic has been implemented for the three discriminators, and this could be the cause of a tracking threshold insensitive to the type of discriminator.

#### 5.1.3 Mean time to lose lock

The MTLL for the different tracking techniques have been evaluated using the methodologies suggested by [18,20,21]. For the DE and pre-filtering techniques, it was possible to adopt the Markov Chain (MC) based approach described in [20] whereas the MTLL for the

BJ was determined using the semi-analytic model described in [18]. The time to lose lock was measured and averaged over several simulation runs. Figure 14 shows the MTLL for the four tracking techniques as a function of different  $C/N_0$  values. The MTLL on ZFS performs relatively poorly compared with the other techniques as expected from the tracking jitter results. It can be observed that the MTLL of MMSES is better than the MTLL of ZFS with performance closer to the DE and BJ techniques.

#### 5.1.4 Convergence analysis

Tracking error convergence analysis provides the steady-state behavior of the different tracking techniques, given an initial delay error. Figure 15 provides the code tracking error for the three techniques considered over a duration of 40 s for a non-coherent discriminator. The simulated signal was characterized by a  $C/N_0$  equal to 25 dB-Hz. Code tracking error for a DLL has been obtained using the semi-analytic technique described in [18] and the curves in Figure 15a shows the average of the tracking errors for different simulations runs. The expression for the averaged tracking error for a given initial delay error is given by

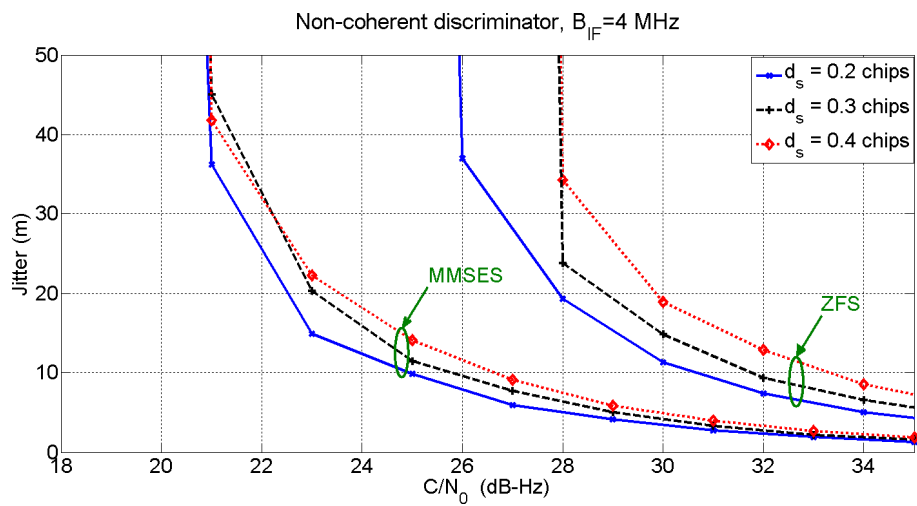
$$\tilde{\tau}_e[k] = \frac{1}{M} \sum_{i=1}^M \tau_e^i[k] | \tau_e^i[0] = \tau_{acq} \quad (28)$$

where  $\tau_{acq}$  is the code delay error from acquisition and  $M$  is the number of simulation runs used for averaging the tracking error,  $\tau_e^i[k]$ . Here  $i$  denotes the simulation run index and  $k$  denotes the time index already used for indexing the correlator outputs in (7).

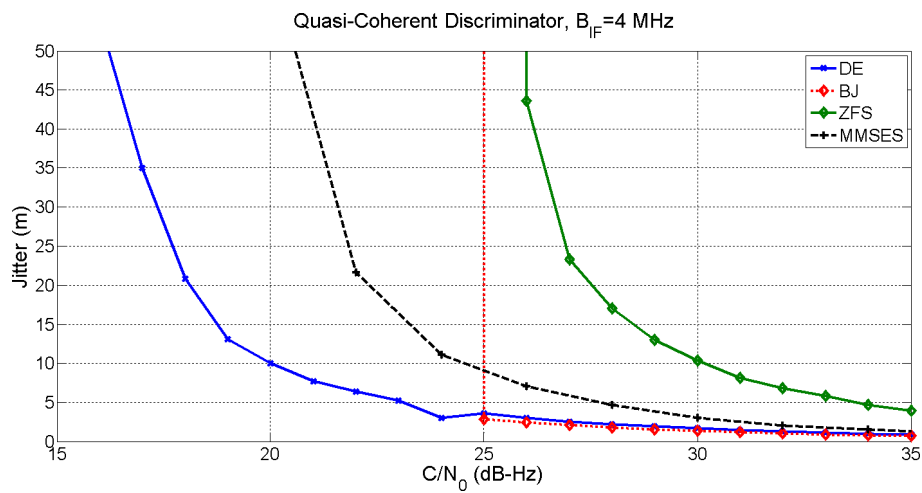
In Figure 15a, an initial acquisition error of -0.5 chips is considered to evaluate the tracking error convergence. This delay error corresponds to a secondary peak of the BOC autocorrelation function. When the DLL is initialized on a secondary peak, both MMSES and DE converge to a zero delay error, whereas BJ is characterized by a steady-state error of about -0.15 chips. This phenomenon is better investigated in Figure 15b and 15c where different error trajectories for the initial 4 s are shown for MMSES and BJ, respectively. These trajectories show the evolution of the delay error as a function of time and for different simulation runs. In the MMSES case, all the trajectories tend to reach a zero steady state error whereas the BJ code error is characterized by two different behaviors. In some cases, the BJ decision logic correctly detects the false peak lock and the code delay error is corrected accordingly. In other cases, however, tracking is too noisy and the algorithm is unable to recover the false peak lock as seen in Figure 15c. The curves in Figure 15a summarize the average behaviors of the three considered algorithms determining the average tracking error defined in (28). Only

**Table 5 Simulation parameters considered for semi-analytic analysis of BOC tracking techniques**

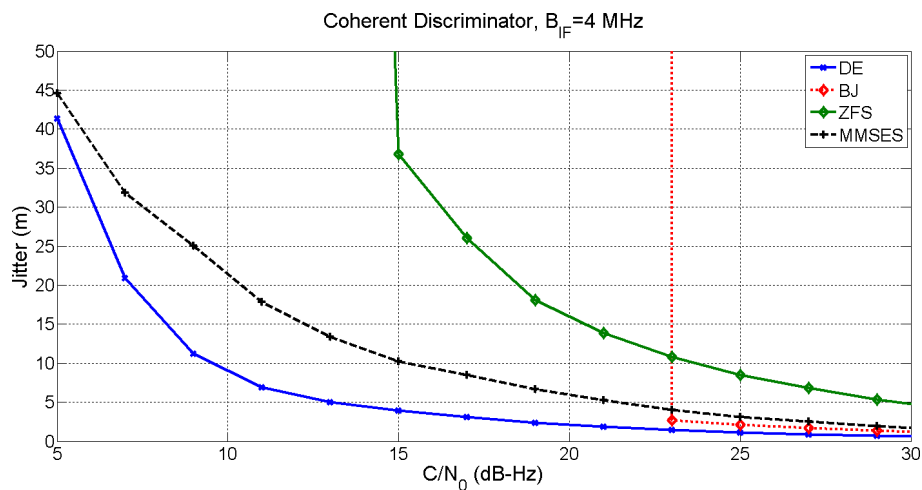
Parameter	Value
Coherent integration time	4 ms
Front-end bandwidth	4 MHz
Code bandwidth	0.5 Hz
Code filter order	1
Simulation runs	10,000
Signal type	BOCs(1,1)



**Figure 10** Tracking jitter of MMSES as a function of the  $C/N_0$  and for different  $d_s$ . Coherent integration time equal to 4 ms.

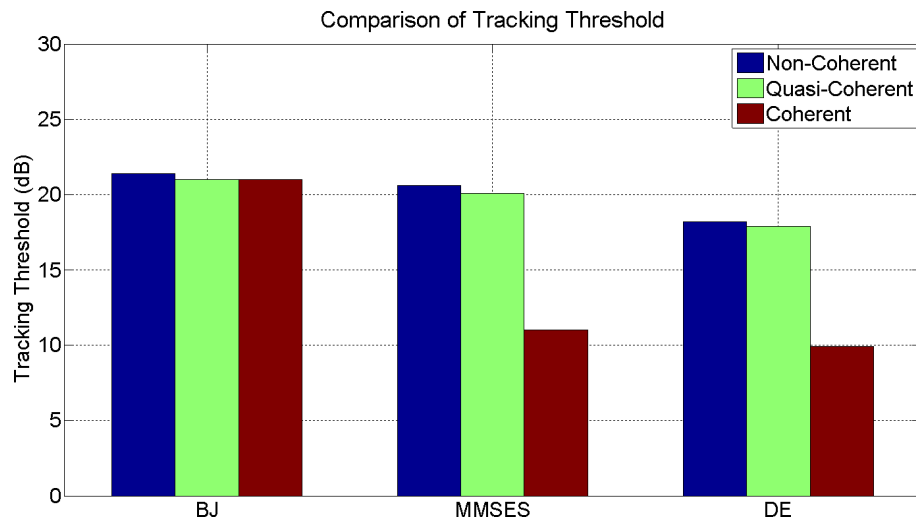


**Figure 11** Comparison of different BOC tracking algorithms in terms of tracking jitter. Quasi-coherent discriminator,  $d_s = 0.2$  chips, 4 ms coherent integration.



**Figure 12** Comparison of different BOC tracking algorithms in terms of tracking jitter. Coherent discriminator,  $d_s = 0.2$  chips, 4 ms coherent integration.





**Figure 13** Comparison of the tracking thresholds of different BOC tracking techniques.  $d_s = 0.2$  chips, 4 ms coherent integration.

MMSES and DE are able to provide a completely unambiguous BOC tracking. While all the three techniques behave similarly for high  $C/N_0$  ratios, BJ technique has higher probability to lose lock and track secondary peaks for low  $C/N_0$  s.

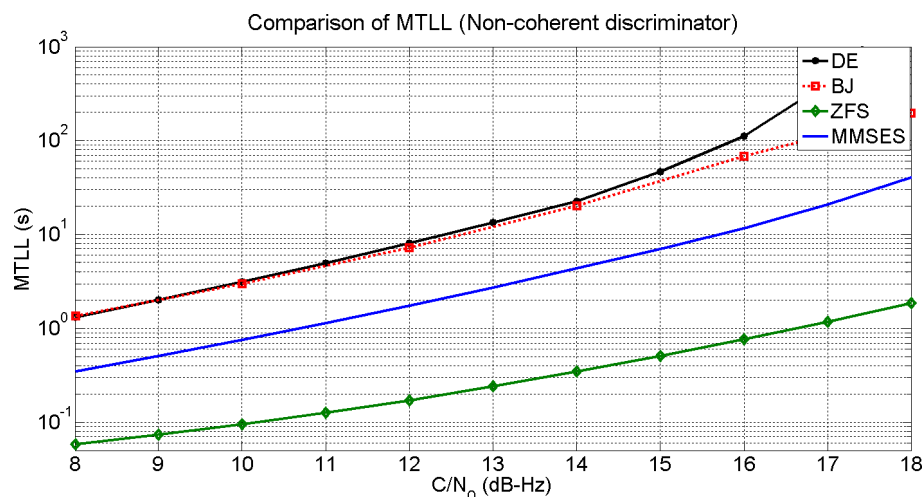
#### 5.1.5 Multipath error envelope

One of the advantages of using MMSES and ZFS is the flexibility to generate signals with varying ACF base-width as depicted in Figure 6. The multipath error envelope for the standard BPSK, DE and MMSES tracking techniques are shown in Figure 16. The case of multipath-to-direct power ratio,  $\alpha = 0.5$  is considered here with a 0.5 chip Early-minus-Late spacing. The results shown in Figure 16 have been obtained assuming an infinite front-end bandwidth. From Figure 16, it can be

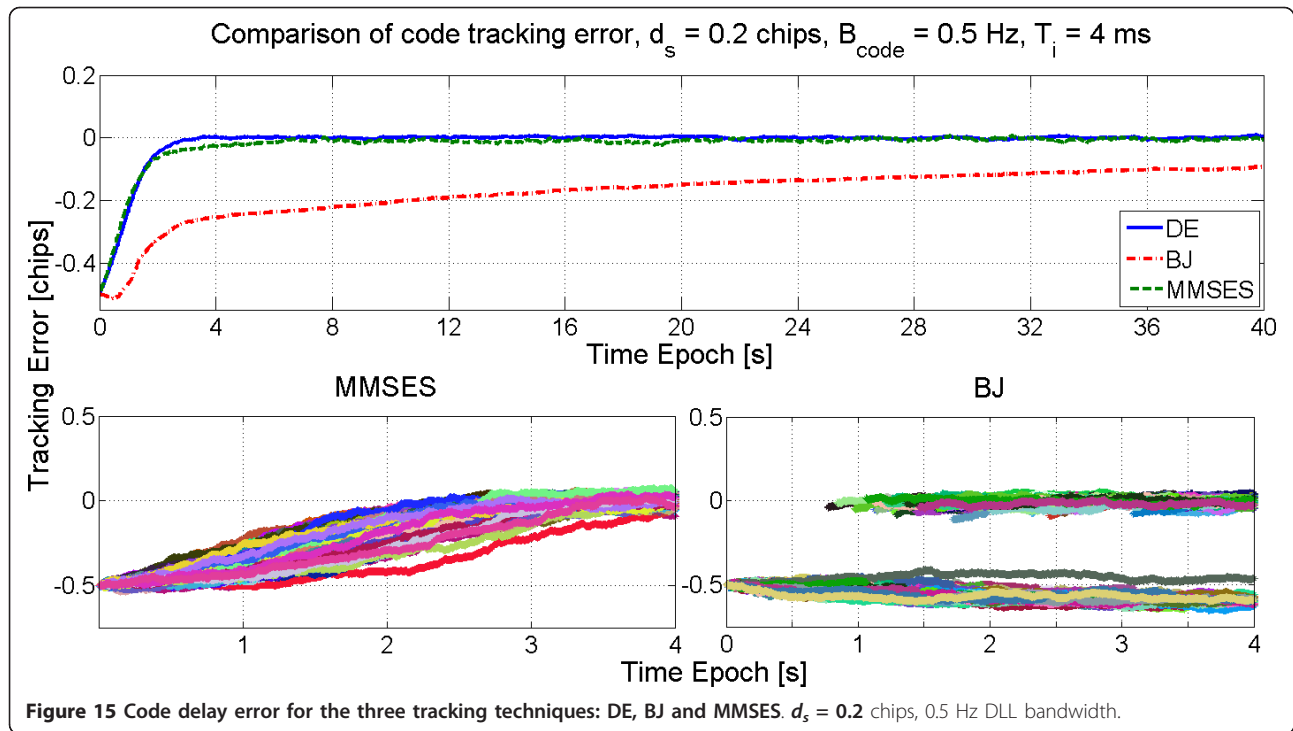
observed that in the MMSES case, when the desired sub-carrier width,  $T_d$ , is equal to the chip duration,  $T_c$ , the resulting multipath error envelope is similar to that of a standard BPSK tracking technique. Considering the flexibility of MMSES, when  $T_d = 0.5T_c$ , the error envelope is similar to the DE tracking technique. Further reducing the desired sub-carrier width,  $T_d = 0.25T_c$ , leads to improved performance that cannot be achieved by the DE. Also, the effect of secondary peaks observed in the DE envelope (the presence of a second lobe in the curve) is not present in the MMSES technique.

#### 5.2 Real data analysis

In order to further test the tracking techniques described above, live data from the experimental



**Figure 14** Comparison of the MTLL for the different tracking techniques.  $d_s = 0.2$  chips, 0.5 Hz DLL bandwidth, 4 ms coherent integration.

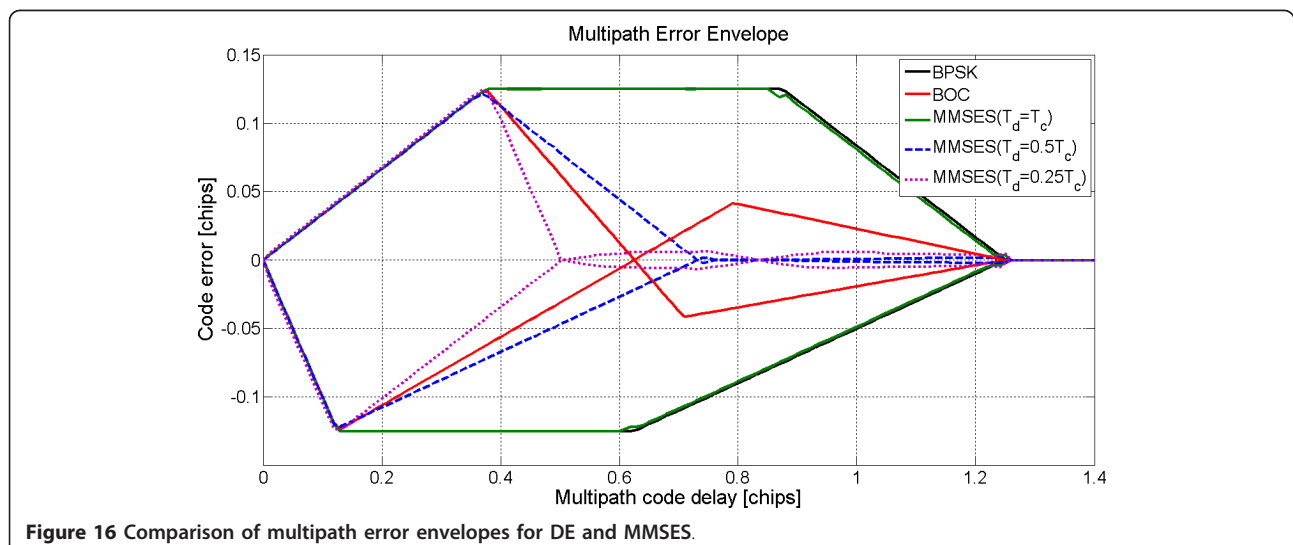


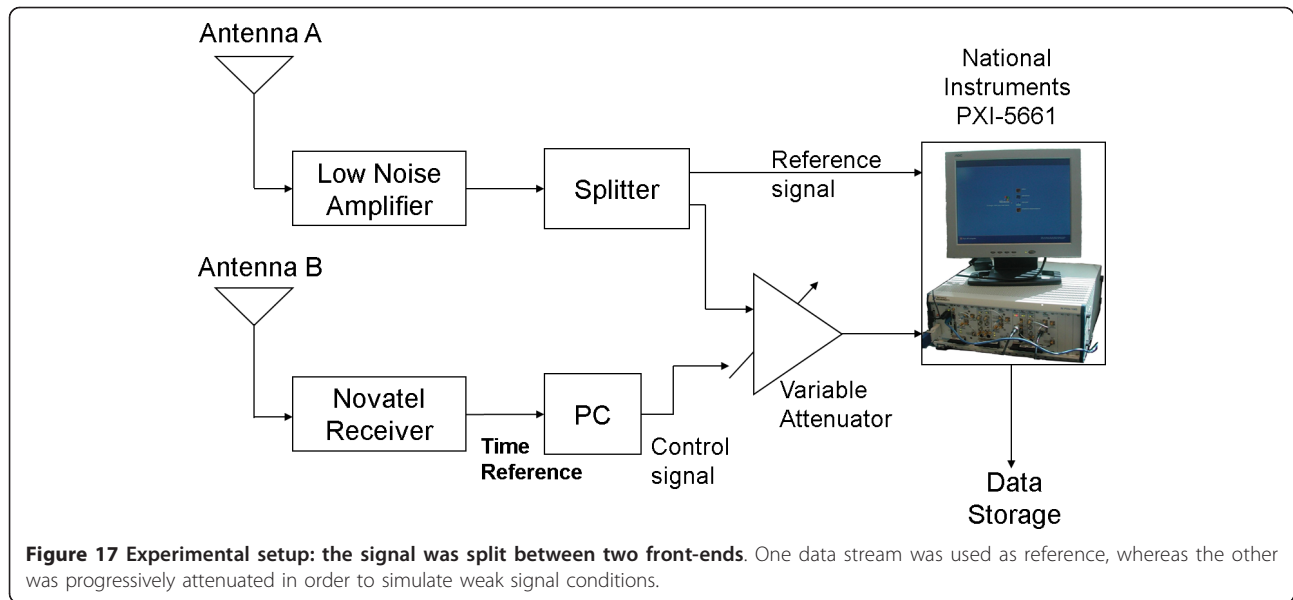
GIOVE-B satellite have been used. The signal parameters for the data collection are provided in Table 4. Data were progressively attenuated in order to simulate weak signal conditions. The setup adopted for the experiment is shown in Figure 17.

The GIOVE-B signal was split between two different front-ends. One of the signal streams was used as a reference, whereas the second was progressively attenuated. The signal was maintained at its nominal strength for 30 s, the attenuation was then progressively increased by 1 dB every 10 s. Data were collected using

a National Instruments (NI) vector analyzer equipped with three PXI-5661 front-ends [22]. The results obtained using the progressively attenuated signals are summarized in Figure 18 where the estimated  $C/N_0$  is shown for the different techniques.

As explained by [23], the  $C/N_0$  estimator is often used as a delay lock indicator. More specifically, the  $C/N_0$  is estimated from the average post-correlation power, i.e. the  $C/N_0$  is directly proportional to the correlation value that is in turn an indicator of the delay error. If a large delay error is committed then the correlation value and



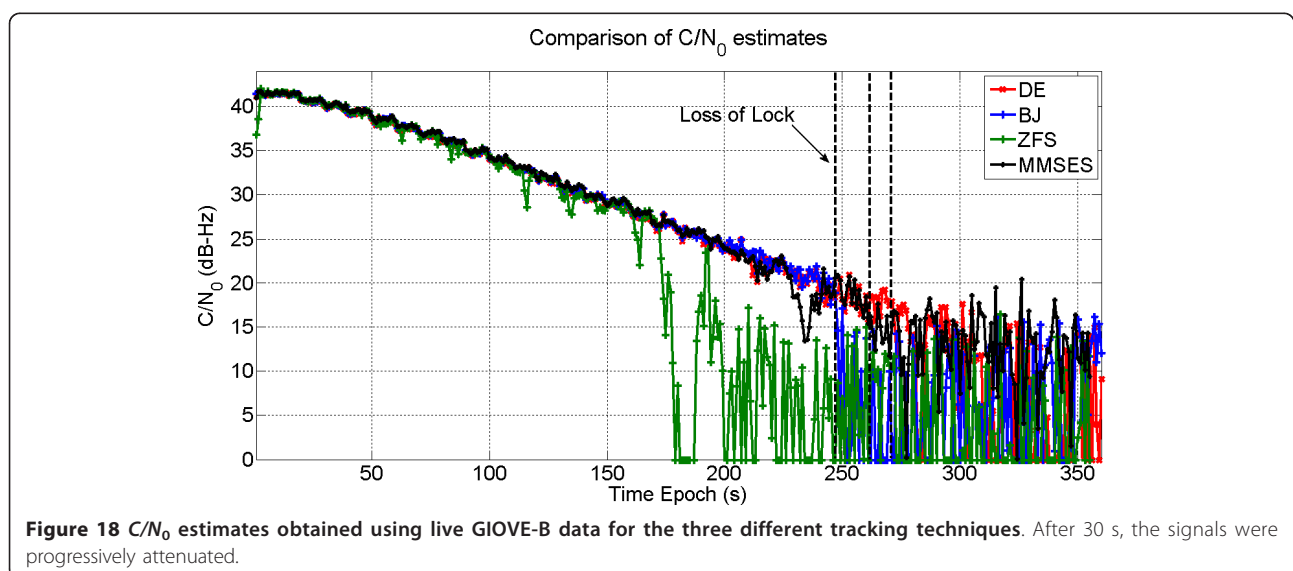


the  $C/N_0$  are significantly reduced. Loss of lock on the delay is thus reflected in randomly varying  $C/N_0$  estimates. In Figure 18, loss of lock is declared on the basis of the true signal parameters. More specifically, the experiment has been conducted using two front-ends collecting synchronized signals. From the first unattenuated signal, reference parameters, i.e., Doppler frequency and code delay, were determined. When the parameters estimated from the second front-end started differing from the reference ones, loss of lock was declared. MMSES loses lock for a  $C/N_0$  of approximately 2 dB-Hz lower compared with BJ. The  $C/N_0$  of the MMSES was determined using the unfiltered Prompt correlator used for carrier tracking. These findings are in

agreement with the simulation results obtained in Section 5.1. It shall be noted that MMSES achieves performance similar to the DE. The ZFS performs poorly with respect to the other techniques.

## 6 Conclusions

In this paper, a new class of pre-filtering techniques for shaping the autocorrelation function of GNSS signals has been proposed. The developed techniques substantially mitigate the noise amplification problem affecting previous pre-filtering algorithms extending their applicability to moderate to low  $C/N_0$  values. The proposed algorithms are based on a convolutional representation of GNSS signals that allows one to apply the concepts



of ZF and MMSE equalization to the signal sub-carrier. The proposed algorithms retain all the flexibility of standard pre-filtering techniques and can be used for unambiguous BOC tracking and autocorrelation shaping for multipath mitigation. From the performed analysis, simulations and real data testing, it emerges that this flexibility can be achieved with a negligible performance reduction with respect to the Double Estimator whose applicability is limited to unambiguous BOC tracking.

#### Competing interests

The authors declare that they have no competing interests.

Received: 31 October 2010 Accepted: 12 December 2011

Published: 12 December 2011

#### References

1. JW Betz, Binary offset carrier modulations for radionavigation. *J Inst Navig.* **48**(4), 227–246 (2001)
2. P Fine, W Wilson, Tracking algorithm for GPS offset carrier signal, in *Proceedings of the ION/NTM (National Technical Meeting)*, San Diego, CA, pp. 671–676 (Jan. 1999)
3. O Julien, C Macabiau, ME Cannon, G Lachapelle, ASPECT: unambiguous sine-boc(n,n) acquisition/tracking technique for navigation applications. *IEEE Trans Aerosp Electron Syst.* **43**(1), 1509–1627 (2007)
4. A Burian, ES Lohan, MK Renfors, Efficient delay tracking methods with sidelobes cancellation for BOC-modulated signals. *EURASIP J Wirel Commun Netw*, Article ID 72626, 20 (July 2007).
5. MS Hodgart, PD Blunt, M Unwin, Double estimator—a new receiver principle for tracking BOC signals. *Inside GNSS*, Spring 26–36 (2008)
6. ES Lohan, A Burian, M Renfors, Low-complexity unambiguous acquisition methods for BOC-modulated CDMA signals. *Int J Satell Commun.* **26**(6), 20 (July 2008)
7. C Yang, M Miller, T Nguyen, D Akos, Generalized frequency-domain correlator for software GPS receiver: preliminary test results and analysis. in *Proceedings of the ION/GNSS, Fort Worth, TX* 2346–2630 (Sept. 2006)
8. SUH Qureshi, Adaptive equalization. *Proc IEEE.* **73**(9), 1349–1387 (1985)
9. RM Nuthalapati, Design of mismatched filters for long binary codes. in *Proceedings of IEEE Radar Conference* 1–6 (May 2008)
10. J Hoggom, Aperture synthesis with a non-regular distribution of interferometer baselines. *Astron Astrophys Suppl.* **15**, 417–426 (1974)
11. O Julien, C Macabiau, E Bertrand, Analysis of Galileo E1 OS unbiased BOC/CBOC tracking techniques for mass market applications. in *2010 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*. Netherlands (2010)
12. PB Anantharamu, D Borio, G Lachapelle, Pre-filtering, side-peak rejection and mapping: several solutions for unambiguous BOC tracking, in *Proceedings of ION/GNSS*, Savannah, Georgia, (Sept. 2009)
13. Kaplan ED, Hegarty CJ (eds.), *Understanding GPS: Principles and Applications*, 2nd edn. (Artech House Publishers, Norwood, MA, USA, 2005)
14. D Borio, A statistical theory for GNSS signal acquisition. PhD Thesis, Politecnico di Torino [http://plan.geomatics.ucalgary.ca/papers/phdthesis\\_danieleborio\\_02apr08.pdf](http://plan.geomatics.ucalgary.ca/papers/phdthesis_danieleborio_02apr08.pdf) (Apr. 2008)
15. JW Betz, KR Kolodziejki, Generalized theory of code tracking with an early-late discriminator Part I: lower bound and coherent processing. *IEEE Trans Aerosp Electron Syst.* **45**(4), 1538–1550 (2009)
16. JW Betz, KR Kolodziejki, Generalized theory of code tracking with an early-late discriminator Part II: noncoherent processing and numerical results. *IEEE Trans Aerosp Electron Syst.* **45**(4), 1551–1564 (2009)
17. AJV Dierendonck, P Fenton, T Ford, Theory and performance of narrow correlator spacing in a GPS receiver. *J Inst Navig.* **39**(3), 265–283 (1992)
18. D Borio, PB Anantharamu, G Lachapelle, Semi-analytic simulations: an extension to unambiguous BOC tracking, in *Proceedings of the ITM (International Technical Meeting)*, San Diego, pp. CA1023–1036 (Jan. 2010)
19. WH Tranter, KS Shanmugan, TS Rappaport, KL Kosbar, *Principles of Communication Systems Simulation with Wireless Applications*, Prentice Hall, Communications Engineering and Emerging Technologies Series, (Jan. 2004)
20. AR Golshan, Loss of lock analysis of a first-order digital code tracking loop and comparison of results to analog loop theory for BOC and NRZ signals. in *Proceedings of the ION/NTM (National Technical Meeting)* 299–305 (Jan. 2005)
21. AR Golshan, SN Lu, TH Dang, Analysis of mean-time to lose lock in a first-order digital DLL for NRZ and BOC signals in the presence of a single-tone interferer, in *Proceedings of the ION/AM (Annual Meeting)*, (Cambridge, MA, 2005)
22. 2.7 GHz RF Vector Signal Analyzer with Digital Downconversion, National Instruments <http://sine.ni.com/nips/cds/print/p/lang/en/nid/203038>
23. AJV Dierendonck, GPS Receivers [Chapter 8] in *Global Positioning System: Theory and Applications*, vol. 1, (American Institute of Aeronautics and Astronautics, Washington, DC, 1996)

doi:10.1186/1687-6180-2011-133

**Cite this article as:** Anantharamu et al.: Sub-carrier shaping for BOC modulated GNSS signals. *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:133.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)