International Journal of Navigation and Observation

Advances in Signal Tracking for GNSS Receivers: Theory and Implementation

Guest Editors: Carles Fernández-Prades, Heidi Kuusniemi, Pau Closas, and Marteen Uijt de Haag

Advances in Signal Tracking for GNSS Receivers: Theory and Implementation

International Journal of Navigation and Observation

Advances in Signal Tracking for GNSS Receivers: Theory and Implementation

Guest Editors: Carles Fernández-Prades, Heidi Kuusniemi, Pau Closas, and Marteen Uijt de Haag

Editorial Board

Contents

Advances in Signal Tracking for GNSS Receivers: Theory and Implementation, Carles Fernández-Prades, Heidi Kuusniemi, Pau Closas, and Marteen Uijt de Haag
Volume 2012, Article ID 821516, 2 pages

Nonlinear Bayesian Tracking Loops for Multipath Mitigation, Pau Closas, Carles Fernández-Prades, José Diez, and David de Castro
Volume 2012, Article ID 359128, 15 pages

Two-Step Galileo E1 CBOC Tracking Algorithm: When Reliability and Robustness Are Keys!, Aleksandar Jovanovic, Cécile Mongrédien, Youssef Tawk, Cyril Botteron, and Pierre-André Farine
Volume 2012, Article ID 135401, 14 pages

Optimized Carrier Tracking Loop Design for Real-Time High-Dynamics GNSS Receivers, Pedro A. Roncagliolo, Javier G. García, and Carlos H. Muravchik
Volume 2012, Article ID 651039, 18 pages

A Novel Quasi-Open Loop Architecture for GNSS Carrier Recovery Systems, Muhammad Tahir, Letizia Lo Presti, and Maurizio Fantino
Volume 2012, Article ID 324858, 12 pages

A New Multipath Mitigation Method for GNSS Receivers Based on an Antenna Array, Sébastien Rougerie, Guillaume Carrié, François Vincent, Lionel Ries, and Michel Monnerat
Volume 2012, Article ID 804732, 11 pages

Hindawi Publishing Corporation
International Journal of Navigation and Observation
Volume 2012, Article ID 821516, 2 pages
doi:10.1155/2012/821516

Editorial

Advances in Signal Tracking for GNSS Receivers: Theory and Implementation

Carles Fernández-Prades,[1] Heidi Kuusniemi,[2] Pau Closas,[1] and Marteen Uijt de Haag[3]

[1] *Communications Subsystems Area, Centre Tecnològic de Telecomunicacions de Catalunya, 08860 Castelldefels, Spain*
[2] *Department of Navigation and Positioning, Finnish Geodetic Institute, 02431 Masala, Finland*
[3] *School of Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701, USA*

Correspondence should be addressed to Carles Fernández-Prades, carles.fernandez@cttc.cat

Received 18 July 2012; Accepted 18 July 2012

Synchronization tracking loops are key components of the GNSS receiver architecture. Their function is to estimate the evolution of the satellite signals' code and carrier phases at the receiver's antenna, yielding the observations required for computing the users' position, velocity, and time. Hence, the tracking loop performance is intimately related to the receiver behavior in terms of precision, sensitivity, reliability, and robustness to interference and multipath.

While the traditional delay lock loop (DLL) and phase lock loop (PLL) architectures have been present in commercial GPS receivers since their inception, various new developments have introduced new challenges in the design of tracking loops for modern GNSS receivers, including the introduction of new GNSS and their associated signal structures, the application of GNSS technology in safety-of-life-related systems, and its use in the distributed synchronization of infrastructures such as the power grid (and thus the associated need for robustness against unintentional and intentional jamming).

The papers in this special issue constitute a representative set of the various possible approaches that the synchronization problem admits.

A. Jovanovic et al. present a tracking algorithm designed for Galileo E1 CBOC signals, analyzing its performance in terms of tracking accuracy, sensitivity, and robustness. The paper provides a full theoretical analysis of the proposed two-step tracking algorithm for Galileo E1 CBOC signals and confirms the results through simulations as well as using real Galileo satellite data.

The particularities of receivers operating in high-dynamics environments have been tackled by P. A. Roncagliolo et al. Their paper proposes a new loop structure named unambiguous frequency-aided phase-locked loop (UFA-PLL) that outperforms classical coupled-loop schemes while allowing simpler design and implementation. Their loop design includes the selection of the correlation time and loop bandwidth that minimize the pull-out probability, without relying on typical rules of thumb. Hence, high-quality phase measurements usually exploited in offline and quasistatic applications become practical for real-time and high-dynamics receivers. Experiments with fixed-point implementations of the proposed loops and actual radio signals are also shown.

M. Tahir et al. propose and demonstrate the use of a quasiopen loop architecture to estimate the time-varying carrier frequency of GNSS signals, showing that this new scheme provides an additional degree of freedom to the design. These results are especially convenient for the design of loop filters operating in electromagnetically harsh environments.

Antenna diversity is known to be a possible way to address the problem of estimating the propagation time delay of the line-of-sight signal in a GNSS receiver under severe multipath conditions. S. Rougerie et al. present a new, complexity-reduced implementation of the space-alternating generalized expectation maximization (SAGE) algorithm. This paper focuses on the trade-off between complexity and performance thanks to the Cramér-Rao bound derivation and shows how the proposed algorithm can be integrated

with a classical GNSS tracking loop, constituting a very promising approach for multipath mitigation.

Finally, P. Closas et al. undertake the synchronization problem from a statistical standpoint, resorting to Bayesian estimation techniques and proposing algorithms that mitigate the bias introduced by multipath. The analysis includes trade-off among realistic propagation channel models and the use of a realistic simulation framework. The authors propose a filtering technique that implements Rao-Blackwellization of linear states and a particle filter for the nonlinear partition and compare it to traditional DLL/PLL-based schemes.

Acknowledgments

We would like to thank many people whose efforts made this special issue possible. We thank the authors for their response to our initial call for papers, and we would like to acknowledge the significant contributions of the reviewers who provided feedback on multiple versions of the papers.

Carles Fernández-Prades
Heidi Kuusniemi
Pau Closas
Marteen Uijt de Haag

Hindawi Publishing Corporation
International Journal of Navigation and Observation
Volume 2012, Article ID 359128, 15 pages
doi:10.1155/2012/359128

Research Article

Nonlinear Bayesian Tracking Loops for Multipath Mitigation

Pau Closas,[1] Carles Fernández-Prades,[1] José Diez,[2] and David de Castro[2]

[1] *Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Parc Mediterrani de la Tecnologia, Avenida Carl Friedrich Gauss 7, Barcelona, 08860 Castelldefels, Spain*
[2] *DEIMOS Space S.L.U, Ronda de Poniente 19, Tres Cantos 28760, Madrid, Spain*

Correspondence should be addressed to Pau Closas, pclosas@cttc.es

Received 28 February 2012; Revised 22 June 2012; Accepted 3 July 2012

Academic Editor: Maarten Uijt de Haag

This paper studies Bayesian filtering techniques applied to the design of advanced delay tracking loops in GNSS receivers with multipath mitigation capabilities. The analysis includes tradeoff among realistic propagation channel models and the use of a realistic simulation framework. After establishing the mathematical framework for the design and analysis of tracking loops in the context of GNSS receivers, we propose a filtering technique that implements Rao-Blackwellization of linear states and a particle filter for the nonlinear partition and compare it to traditional delay lock loop/phase lock loop-based schemes.

1. Introduction

Global Navigation Satellite Systems (GNSS) are the general concept used to identify those systems that allow user positioning based on a constellation of satellites. Specific GNSS are the well-known American GPS, the Russian GLONASS, or the forthcoming European Galileo. All those systems rely on the same principle: the user computes its position by means of measured distances between the receiver and the set of in-view satellites. These distances are calculated estimating the propagation time that synchronously transmitted signals take from each satellite to the receiver. Therefore, GNSS receivers are only interested in estimating the delays of signals which are received directly from the satellites, referred to as line-of-sight signal (LOSS), since they are the ones that carry information of direct propagation time. Hence, reflections distort the received signal in a way that may cause a bias in delay and carrier-phase estimations. Multipath is probably the dominant source of error in high-precision applications, especially in urban scenarios, since it can introduce a bias up to a hundred of meters when employing a 1-chip wide (standard) delay lock loop (DLL) to track the delay of the LOSS, which is a common synchronization method used in spread-spectrum receivers. This error might be unacceptable in many applications.

Sophisticated synchronization techniques estimate not only LOSS parameters but those of multipath echoes. This results in enhanced, virtually bias-free pseudorange measurements. In this paper, we investigate multipath estimating tracking loops in realistic scenarios, where this effect is known to be severe. The analysis is driven in two directions. Firstly, a review of statistical characterization of the channel model in such situations is performed and a commercial signal simulator. Secondly, a novel multipath estimating tracking loop is discussed, providing details on the implementation, as well as comparisons to state-of-the-art techniques when different channel characteristics are considered. This tracking loop resorts to the Bayesian nonlinear filtering framework, sequentially estimating the unknown states of the system (i.e., parameters of the LOSS and echoes) and providing robust pseudorange estimates, subsequently used in the positioning solution. The so-called multipath estimating particle filter (MEPF) considers Rao-Blackwellization of signal amplitudes and the use of a suitable nonlinear filter for the rest of nonlinear states, for example, time-delays and their rate. More precisely, Rao-Blackwellization involves marginalization of linear states and the use of a standard Kalman filter to track signal amplitudes with the goal of reducing the estimation variance, since (i) the dimensionality of the problem that nonlinear filters solve is reduced and (ii) linear states are optimally tackled. For the nonlinear part of the state space we consider sequential Monte-Carlo methods (specifically, the standard particle filtering) as one of the most promising alternatives in advanced

GNSS receiver designs. Realistic computer simulation results are presented using the GRANADA FCM signal simulator and the performance of the MEPF is evaluated.

The remainder of the paper is organized as follows. Section 2.1 provides a brief overview of the fundamentals of GNSS, their signal structure, available channel models, and receivers' architecture and describes a realistic simulation platform. Section 3 sketches the basics of particle filters, and Section 4 is devoted to their application to GNSS signal synchronization in the presence of multipath. Section 5 presents computer simulations, and finally Section 6 concludes the paper. For the sake of completeness, the paper shows in the Appendix the equivalence between precorrelation and postcorrelation processing of GNSS signals. Notice that in this paper, the MEPF method operates after correlation is performed in order to operate at a lower data rate.

2. Fundamentals of Global Navigation Satellite Systems

GNSS space vehicles broadcast a low-rate navigation message that modulates continuous repetitions of pseudorandom spreading codes, that in turn are modulating a carrier signal allocated in the L band. The navigation message, after proper demodulation, contains among other information the so-called ephemeris, a set of parameters that allow the computation of the satellite position at any time. These positions, along with the corresponding distance estimations, allow the receiver to compute its own position and time, as we will see hereafter. Basically, a GNSS receiver performs trilateration, a method for determining the intersections of three or more sphere surfaces given the centers and radii of the spheres. In this case, the centers of the spheres are the satellites, whose position can be computed from the navigation message, and the radii of the spheres are the distances between the satellites and the receiver, estimated from the time of flight.

The distance between the receiver and a given satellite can be computed by

$$\rho_i = c\left(t_i^{\mathrm{Rx}} - t_i^{\mathrm{Tx}}\right), \tag{1}$$

where $c = 299792458\,\mathrm{m/s}$ is the speed of light, t_i^{Rx} is the receiving time in the receiver's clock, and t_i^{Tx} the time of transmission for a given satellite i. Receiver clocks are inexpensive and not perfectly in sync with the satellite clock, and thus this time deviation is another variable to be estimated. The clocks on all of the satellites belonging to the same system s, where $s = \{\mathrm{GPS}, \mathrm{Galileo}, \mathrm{GLONASS}, \ldots\}$, are in sync with each other, so the receiver's clock will be out of sync with all satellites belonging to the same constellation by the same amount $\Delta t^{(s)}$. In GNSS, the term *pseudorange* is used to identify a range affected by a bias, directly related to the bias between the receiver and satellite clocks. There are other factors of error: since propagation at speed c is only possible in the vacuum, atmospheric status affects the propagation speed of electromagnetic waves modifying the propagation time and thus the distance estimation. For instance, the ionosphere, that is the part of the atmosphere above 60 km until 2000 km of the Earth surface, is a plasmatic medium that causes a slowdown in the group velocity and a speed up of the phase velocity, having an impact in code and phase delays and, thus, impeding precise navigation when its effects are not mitigated. Actually, errors can be on the order of tens of meters in geomagnetic storm episodes [1].

For each in-view satellite i of system s, we can write

$$\rho_i = \sqrt{\left(x_i^{\mathrm{Tx}} - x\right)^2 + \left(y_i^{\mathrm{Tx}} - y\right)^2 + \left(z_i^{\mathrm{Tx}} - z\right)^2} + c\Delta t^{(s)} + \sigma_e, \tag{2}$$

where $(x_i^{\mathrm{Tx}}, y_i^{\mathrm{Tx}}, z_i^{\mathrm{Tx}})$ is the satellite's position (known from the navigation message), (x, y, z) the receiver's position, and σ_e gathers other sources of error. Since the receiver needs to estimate its own 3D position (three spatial unknowns) and its clock deviation with respect to the satellites' time basis, at least $3 + N_s$ satellites must be seen by the receiver at the same time, where N_s is the number of different navigation systems available (in-view) at a given time. Each received satellite signal, once synchronized and demodulated at the receiver, defines one equation such as the one defined in (2), forming a set of nonlinear equations that can be solved algebraically by means of the Bancroft algorithm [2] or numerically, resorting to multidimensional Newton-Raphson and weighted least square methods [3]. When *a priori* information is added we resort to Bayesian estimation, a problem that can be solved recursively by a Kalman filter or any of its variants. The problem can be further expanded by adding other unknowns (for instance, parameters of ionospheric and tropospheric models), sources of information from other systems, mapping information, and even motion models of the receiver. In the design of multi-constellation GNSS receivers, the vector of unknowns can also include the receiver clock offset with respect to each system in order to take advantage of a higher number of in-view satellites and using them jointly in the navigation solution, therefore increasing accuracy.

2.1. Signal Model. A general signal model for most navigation systems consists of a direct-sequence spread-spectrum (DS-SS) signal [4], synchronously transmitted by all the satellites in the constellation. This type of signals enables code division multiple access (CDMA) transmissions, that is, satellite signals are distinguished by orthogonal (or quasi-orthogonal) codes. At a glance, these signals consists of two main components: a ranging code (the PRN spreading sequence) and a low rate data link (broadcasting necessary information for positioning such as satellites orbital parameters and corrections). The complex baseband model of the signal transmitted by a GNSS space vehicle reads as

$$s_T(t) = \sqrt{P_T}\left(\gamma \sum_{u=-\infty}^{\infty} d_I(u)\, p_I(t - uT_{b_I}) + j\sqrt{1-\gamma^2} \sum_{l=-\infty}^{\infty} d_Q(l)\, p_Q\left(t - lT_{b_Q}\right)\right), \tag{3}$$

where

$$p_I(t) = \sum_{n=0}^{N_{c_I}-1} q_I(t - nT_{\mathrm{PRN}_I}),$$
$$q_I(t) = \sum_{k=0}^{L_{c_I}-1} c_I(k) g_{T,I}(t - kT_{c_I}), \tag{4}$$

being P_T the transmitting power, γ a parameter controlling the power balance, $d_I(m) \in \{-1, 1\}$ the data symbols, T_{b_I} the bit period, N_{c_I} the number of repetitions of a full codeword that spans a bit period, $T_{\mathrm{PRN}_I} = T_{b_I}/N_{c_I}$ the codeword period, $c_I(k) \in \{-1, 1\}$ a chip of a spreading codeword of length L_{c_I} chips, $g_{T,I}(t)$ the transmitting chip pulse shape, which is considered energy normalized for notation clarity, and $T_{c_I} = T_{b_I}/N_{c_I}L_{c_I}$ is the chip period. Figure 1 aims at clarifying the relation between those bits/chips parameters. Subindex I refers to the in-phase component, and all parameters are equivalently defined for the quadrature component, referred to with the subindex Q. This signal model describes all GNSS's signals-in-space, for instance GPS L1, GPS L5, Galileo E1, and Galileo E5. Refer to [5] for the details.

2.2. Propagation Channel Model. A key aspect in the definition of the propagation channel model between satellites' antenna and the user's receiver antenna is whether it can be considered narrowband or wideband, which depends on the bandwidth of the propagation channel in which a given signal is transmitted, being assessed with respect to the channel *coherence bandwidth*. The coherence bandwidth is defined as the frequency band within which all frequency components are equally affected by fading due to multipath. In narrowband systems, all the components of the signal are equally influenced by multipath, while in wideband systems the various frequency components of the signal are differently affected by fading. Narrowband systems, therefore, are affected by nonselective fading, whereas wideband systems are affected by selective fading. The coherence bandwidth depends on the environment and is given by

$$B_c = \frac{1}{2\pi T}, \tag{5}$$

where T is the *delay spread*, which is the time span between the arrival of the first and the last multipath signals that can be sensed by the receiver. In a fading environment, a propagated signal arrives at the receiver through multiple paths. For a typical GNSS multipath propagation channel in which $T < 0.5\,\mu$s (the limit can be greater in nonurban areas, but in general it is not lower), we obtain that the system is wideband if transmitted signals are wider than 320 kHz, which is the case for GNSS waveforms (in the order of MHz). Hence, we conclude that we need to define propagation channel models considering wideband systems. Another important definition within this context concerns *coherence time*. The coherence time, T_{coh}, is defined as the time interval

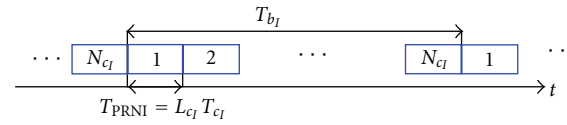


Figure 1: Relation among the parameters defining bits and spreading sequences in a generic navigation signal (in-phase component).

during which the characteristics of the propagation channel remain approximately constant, and it is given as

$$T_{\mathrm{coh}} = \frac{1}{2f_m}, \tag{6}$$

where f_m is the maximum Doppler shift. The Doppler shift is given as v/λ, where v is the radial speed of the mobile terminal with respect to the satellite and λ is the signal wavelength. A channel is considered WSSUS (wide-sense stationary with uncorrelated scatterers) during the coherence time.

In the following, we describe four of the most relevant satellite channel models found in the literature.

2.2.1. Jahn's Channel Characterization. Jahn et al. provided a wideband channel model for land mobile satellite services [6]. The model was derived from a channel measurements campaign performed in the L band at 1820 MHz. An aircraft transmitted a spread spectrum signal of 30 MHz, being received by a mobile receiver (handheld or car terminal). From those measurements, authors characterized the channel assuming WSSUS and modeling it as a filter structure with delay taps. Then, they provided statistical models for LOS (Rician probability density function for the amplitude of the direct path), shadowing (ray amplitude following a Raileigh distribution with a lognormal distributed mean power), near echoes (the number of the near echoes follows a Poisson distribution, with delays being exponentially distributed and amplitudes following a Rayleigh distribution), and far echoes (same distributions than near echoes but with other parameters). Table 1 summarizes the main features of Jahn's statistical channel model.

2.2.2. Loo's Channel Characterization. The Loo's land mobile satellite channel model [7] is a statistical model that assumes that the LOS component under foliage attenuation (shadowing) is lognormally distributed and that the multipath effect is Rayleigh distributed. This model provides complete statistical descriptions for different shadowing and multipath conditions based on an extensive measurement campaign for different frequency bands. For the L band, the "Inmarsat's Marecs A" satellite was used as transmitter, while a mobile laboratory was considered for signal reception, resulting in a fixed 19° elevation. Many more investigations on L-band measurements are also referred to in [8], obtaining results for other elevation angles. Table 1 summarizes the main features of Loo's statistical channel model.

2.2.3. Pérez-Fontán's Channel Characterization. The model presented by Fontán et al. in [9] addressed the statistical

Table 1: Comparison between channel models.

	Jahn	Pérez-Fontán	Loo	Steingass/Lehner
Frequency	1820 MHz (L-band)	L, S, Ka band	UHF, L, S, Ka band	GPS L1, but others can be selected
Bandwidth	30 MHz (wideband)	Narrowband, wideband	Depends on meas. campaign	Wideband
User	(i) Handheld at 4 km/h. (ii) Vehicle at 40 km/h	(i) Handheld. (ii) Vehicle	Depends on meas. campaign	(i) Vehicle (ii) Pedestrian
Dynamics	Constant velocity	Not provided for L band, wideband	Not provided	Variable velocity
Types	(i) Direct path (ii) Near echoes (iii) Far echoes	(i) Direct signal (ii) Multipath echoes. (iii) Diffuse components	(i) Direct signal (ii) Multipath echoes	(i) Direct Path, up to 3 diffracted LOS (ii) Reflected echoes
Number	Scenario dependant	50 rays	Not provided	Up to 50
Attenuation	10 to 30 dB	9 to 16 dB	<30 dB	Rice fading
Doppler	Not straightforward	Included in generated time series	Not straightforward	Deterministic included in the generated time series
Shadowing	(i) LOS (ii) Shadowing (iii) NLOS	(i) LOS (ii) Moderate shadowing (iii) Deep shadowing conditions	(i) LOS (ii) Light shadowing (iii) Heavy shadowing	(i) LOS (ii) House front diffraction (iii) Trees (iv) Lamp posts
Elevation dependant	15°, 25°, 35°, 45°, 55°	15°, 30°, 45°, 60°, 70° 80°	19° (other based on different references)	[5°, 90°], based on 9 levels of measurement data: 5°, 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80°
Azimuth dependant	None.	Uniform azimuth distribution.	None.	Distribution extracted from measurements and depends on elevation
Environments	(i) Open (ii) Rural (iii) Suburban (iv) Urban (v) Highway	(i) Open (ii) Suburban (iii) Urban (iv) Tree shadowed	(i) Suburban (ii) Tree shadowed (iii) Other based on different references	Urban and suburban for car and pedestrian
Pros.	(i) Different scenarios available (ii) Complete statistical model (iii) Good shadowing and NLOS approach (iv) Available parameters for GNSS L-wideband	(i) Time series are obtained prior to statistical model generation (ii) Complete statistical model for both LOS and multipath rays	(i) Complete statistical model (ii) To be used as baseline for other statistical models	(i) Downloadable model available (ii) Urban, suburban and rural (iii) Vehicle and pedestrian (iv) Realistic model (v) Includes different elevations (vi) Includes houses, trees and lamp posts (vii) Low computational cost. (viii) Takes into account generated scenario (ix) Generates MP profile (x) Developed for GNSS

Table 1: Continued.

	Jahn	Pérez-Fontán	Loo	Steingass/Lehner
Cons.	(i) Measurements-based model parameters	(i) Measurements-based model parameters	(i) Measurements-based model parameters	(i) The license does not allow the modification of the code. Although own propagation data can be used to modify the model
	(ii) Not possible to obtain parameters for user-defined scenarios	(ii) Not possible to obtain parameters for user-defined scenarios	(ii) Not straightforward multipath parameters	(ii) Scenario needs to be reexecuted for each considered satellite
	(iii) Constant velocity with fixed values	(iii) Not enough measured data to extract significant model parameters	(iii) Not enough available data to extract parameters for different scenarios	
	(iv) No elevations over 55°	(iv) Limited data for applicable GNSS conditions: L band, wideband	(iv) No detailed user dynamics description	
	(v) Rays Doppler not fully defined		(v) Limited data for applicable GNSS conditions: L band, wideband	

modeling of shadowing and multipath effects in land mobile satellite applications for a wide range of environments with different clutter densities (from open to dense urban areas) and elevation angles (from 5° to 90°) at L, S, or Ka Bands, using a comprehensive experimental database to extract the model parameters for the different bands, environments, and elevations. One of its main contributions consists of producing time series of any channel parameter whose study is required, instead of just cumulative distribution functions. These ones may be computed later from the generated series. The model uses a first-order Markov chain to describe the slow variations of the direct signal, basically due to shadowing/blockage effects. The overall signal variations due to shadowing and multipath effects within each individual Markov state are assumed to follow a Loo distribution with different parameters for each shadowing condition (Markov state). Up to this point the model is of the narrow-band type since it does not account for time dispersion effects. These effects are introduced by using an exponential distribution to represent the excess delays of the different echoes. Table 1 summarizes the main features of Pérez-Fontán's channel model.

2.2.4. Steingass/Lehner's Channel Characterization. The Steingass/Lehner land mobile channel model presented in [10] was developed using data recorded in a high-resolution measurement campaign carried out in Munich in 2002. Different types of environments (urban, suburban, and rural) were measured for car and pedestrian applications. It has been approved as standard by the ITU [11]. For the measurements, a 100 MHz signal near the GPS L1 band was used. This signal provided a time resolution of about 10 ns. The received signal was processed using a super-resolution algorithm to extract the single reflections. With this information, the probability density distribution of the parameters of the reflected rays, such as Doppler shift, power of echoes, duration of a reflector, and number of echoes, were extracted. In urban environments, three major obstacles influence the propagation of the LOS signal: house fronts, trees, and lamp posts. The model is comprised of a deterministic part with a generated scenery, which computes geometrically the LOS signal shadowing and knife-edge diffraction for house fronts, lamp posts, and trees. The other observables like the number of coexisting echoes, life span of reflectors, and the mean power of the echoes are generated stochastically, using the probability density distribution extracted from the measurements. The output of the model is a complex time-variant channel impulse response recalculated each time step. Table 1 summarizes the main features of Steingass/Lehner's channel model.

2.3. A Realistic Signal/Channel Simulator. When transmitted, satellite's signals travel through a propagation channel which modifies its amplitude, phase, and delay. Indeed, many replicas of the same transmitted signal can reach the receiver's antenna due to multipath propagation. In general, these replicas are caused by reflections of the direct signal in surrounding obstacles (e.g., buildings, trees, and ground etc.). As shown above, such propagation channel is generically modeled by a linear time-varying impulse response with M_i propagation paths:

$$h_i(t;\xi) = \sum_{m=0}^{M_i-1} \alpha_{i,m}(t) e^{j\phi_{i,m}(t)} \delta(\xi - \tau_{i,m}(t)), \tag{7}$$

where $\alpha_{i,m}$, $\phi_{i,m}(t)$ and $\tau_{i,m}(t)$ are the amplitude, phase, and delay of the mth propagation path for the ith satellite, ξ is the multipath delay axis and the index $m = 0$ stands for the line-of-sight signal. These channel parameters can be seen as realizations of random processes with underlying probability density functions $f_{\alpha_p}(\alpha)$, $f_{\phi_p}(\phi)$, and $f_{\tau_p}(\tau)$, respectively, whose shape and parameters are approximated by the models outlined above.

Therefore, considering M_s visible satellites, the signal $r(t)$ received at the receiver's antenna is the superposition of the transmitted signals, as propagated through the corresponding channel, and corrupted by additive noise, $w(t)$. This reads as

$$\begin{aligned} r(t) &= \sum_{i=0}^{M_s-1} s_{T,i}(t) * h_i(t;\xi) + w(\mathrm{t}) \\ &= \sum_{i=0}^{M_s-1} \int_{-\infty}^{+\infty} s_{T,i}(t-\xi) h_i(t;\xi) d\xi + w(t) \\ &= \sum_{i=0}^{M_s-1} \sum_{m=0}^{M_i-1} \alpha_{i,m}(t) e^{j\phi_{i,m}(t)} s_{T,i}(t - \tau_{i,m}(t)) + w(t), \end{aligned} \tag{8}$$

where $s_{T,i}(t)$ is the transmitted signal $s_T(t)$ corresponding to the i-th satellite.

As shown in [12], the term $\phi_{i,m}(t)$ can be approximated by its first-order Taylor expansion as $\phi_{i,m}(t) \approx 2\pi f_{d_{i,m}}(t)t + \phi_{i,m,0}$. Hence, the general baseband equivalent model that will be used along this paper is

$$\begin{aligned} r(t) = \sum_{i=0}^{M_s-1} \sum_{m=0}^{M_i-1} &\alpha_{i,m}(t) e^{j2\pi f_{d_{i,m}}(t)t} \\ &\cdot e^{j\phi_{i,m,0}} s_{T,i}(t - \tau_{i,m}(t)) + w(t). \end{aligned} \tag{9}$$

The first element in the receiver RF chain is a right hand circularly polarized (RHCP) antenna, usually with nearly hemispherical gain coverage, with the mission to receive the radionavigation signals of all the satellites in view. The RF signals collected by the antenna are immediately amplified by a low noise amplifier (LNA), a key element which is the most contributing block to the noise figure of the receiver. The LNA also acts as a filter, minimizing out-of-band RF interferences and setting the sharpness of the received code. After the LNA, the amplified and filtered RF signals are then downconverted to an intermediate frequency (IF) using signal mixing frequencies from local oscillators (LOs). These LOs are derived from a receiver reference oscillator, often an oven-stabilized clock with typical accuracies of 10^{-8}. There is a need for one LO per down-conversion stage. Two or three down-conversion stages are commonly devoted to reject mirror frequencies or large out of band jamming signals, in particular the 900 MHz used by the GSM mobile

communication system. However, depending on the subsequent analog-to-digital converter (ADC) characteristics, a one-stage downconversion or even a direct L-band sampling is also possible [13]. The lower sideband generated by the mixer process is selected, while the upper sideband is filtered by a postmixer bandpass filter. It is important to point out that signal Doppler's and PRN codes are preserved after the mixing stage, only the carrier frequency is lowered.

In the sequel, we focus on the contribution of a single satellite and thus omit the dependence with i of the signal model. Considering a generic data sequence d, chip code c, chip-shaping pulse $g_T(t)$, chip period T_c, N_c full codes in a whole bit, and data period T_b, the baseband equivalent received signal for a channel model as in (7) but particularized to $M_i = 1$ (i.e., only one line of sight signal) can be put in the form

$$\begin{aligned}\widetilde{r}_0(t) &= a_0(t)\sum_{u=-\infty}^{\infty} d(u)\sum_{n=0}^{N_c-1}\sum_{k=0}^{L_c-1} c(k)\\ &\quad\cdot \widetilde{g}_R(t-\tau_0(t)-kT_c-nT_{\mathrm{PRN}}-uT_b)+\widetilde{w}(t)\\ &= a_0(t)\sum_{u=-\infty}^{\infty} d(u)\widetilde{p}(t-\tau_0(t)-uT_b)+\widetilde{w}(t)\\ &= |a_0(t)|e^{j(2\pi f_d(t)t+\phi_0)}\sum_{u=-\infty}^{\infty} d(u)\widetilde{p}(t-\tau_0(t)-uT_b)\\ &\quad+\widetilde{w}(t),\end{aligned} \tag{10}$$

where $\widetilde{g}_R(t)$ is the pulse received at the antenna and then filtered by a precorrelation filter (usually the LNA), $\widetilde{p}(t)$ is the filtered version of $p(t) = p_I(t) + jp_Q(t)$, and the term $\widetilde{w}(t)$ stands for the filtered thermal noise and other unmodeled terms. The objective of a synchronization method is to estimate the time delay $\tau_0(t)$, Doppler shift $f_d(t)$ and the carrier phase information ϕ_0 embedded into the phase of the complex amplitude $a_0(t) = |a_0(t)|e^{j(2\pi f_d(t)t+\phi_0)}$.

The analog-to-digital conversion and the automatic gain control (AGC) processes take place at IF or baseband, where all the signals from GNSS satellites in view are buried in thermal noise. Once the received signal is digitized, it is ready to feed each of the N digital receiver channels. Every receiver channel is intended to acquire and track the signal of a single GNSS satellite; typical receivers are equipped with $N = 12$ channels. The multiplication of the IF digitized signal by a local replica of its carrier frequency allows to produce the in-phase (I) and quadrature-phase (Q) components of the digitized signal.

Assuming $\widetilde{w}(t)$ as additive white Gaussian noise (AWGN), at least in the band of interest, it is well known that the optimum receiver is the code matched filter, expressed as

$$\begin{aligned}h_{\mathrm{MF}}\left(t;\widehat{\tau}_0,\widehat{f}_{d_0},\widehat{\phi}_0\right) &= \sum_{k=0}^{L_c-1} c^*(k)g_R^*(-t-kT_c+\widehat{\tau}_0+L_cT_c)\\ &\quad\cdot e^{-j\widehat{\phi}_0}e^{-j2\pi\widehat{f}_{d_0}(t)t}\\ &= q_R^*(-t+\widehat{\tau}_0+L_cT_c)e^{-j\widehat{\phi}_0}e^{-j2\pi\widehat{f}_{d_0}(t)t},\end{aligned} \tag{11}$$

where $\widehat{\tau}_0$, $\widehat{f}_{d_0}$, $\widehat{\phi}_0$ are local estimates of the time delay, Doppler shift, and carrier phase of the received signal, and $(dot)^*$ stands for the complex conjugate operator. Theoretically $g_R(t) = g_T(t)$, but actual implementations make use of approximated versions: while $g_T(t)$ is a rectangular pulse filtered at the satellite, $g_R(t)$ is digitally generated at the receiver and therefore not filtered. In addition, $g_T(t)$ is usually filtered again by a precorrelation filter before the matched filter, as expressed in (10) with $\widetilde{g}_R(t)$. The code matched filter output can be written in the form

$$y\left(t;\check{\tau}_0,\check{f}_{d_0},\check{\phi}_0\right) = \widetilde{r}_0\left(t;\tau_0,f_{d_0},\phi_0\right) * h_{\mathrm{MF}}\left(t;\check{\tau}_0,\check{f}_{d_0},\check{\phi}_0\right). \tag{12}$$

Notice that, in the matched filter, we have substituted the estimates $\widehat{\tau}_0$, $\widehat{f}_{d_0}$, and $\widehat{\phi}_0$ for trial values obtained from previous (in time) estimates of these parameters which we have defined as $\check{\tau}_0$, $\check{f}_{d_0}$, and $\check{\phi}_0$, respectively. This is the usual procedure in GNSS receivers, since the estimates are not really available, but to be estimated after correlation.

In DS-SS terminology, the matched filter is often referred to as *correlator*, while the processing it performs is called *despreading*. Since the correlators perform accumulation of the sampled signal during a period T_{int} and then release an output, we can write the discrete version of the signal as

$$y_n = \sum_{s=0}^{\lfloor T_{\mathrm{int}}/T_s-1\rfloor} y\left(nT_{\mathrm{int}}-sT_s;\check{\tau}_0,\check{f}_{d_0},\check{\phi}_0\right), \tag{13}$$

where T_s is the sampling period, T_{int} is the integration time (usually, $T_{\mathrm{int}} = T_cL_c$) and $\lfloor\cdot\rfloor$ stands for the nearest integer towards zero.

Equation (13) can be expressed more conveniently by solving the convolution in (12), which yields [14]

$$\begin{aligned}y_{n,I} &= \frac{|a_0|}{2}K\frac{\sin\left(\pi\overline{\Delta f}T_{\mathrm{int}}\right)}{\pi\overline{\Delta f}T_{\mathrm{int}}}d\left([n]_{T_b/T_{\mathrm{int}}}\right)R_{\widetilde{p}q}\left(\overline{\Delta\tau_0}\right)\\ &\quad\cdot\cos\left(\pi\overline{\Delta f}T_{\mathrm{int}}+\Delta\phi\right)+v_I(n),\end{aligned} \tag{14}$$

where we defined $\overline{\Delta f} = f_{d_0}-\check{f}_{d_0}$, $\overline{\Delta\phi} = \phi_0-\check{\phi}_0$ and $\overline{\Delta\tau_0} = \tau_0-\check{\tau}_0$ (i.e., the estimation errors), $\lfloor\cdot\rfloor$ stands for the nearest integer toward zero, and $[n]_{T_b/T_{\mathrm{int}}}$ means the integer part of nT_{int}/T_b, being T_b the navigation bit period, and

$$R_{\widetilde{p}q}(\xi) = \frac{1}{T_{\mathrm{PRN}}}\int_{\xi}^{T_{\mathrm{PRN}}+\xi}\widetilde{p}(t)q^*(t-\xi)dt \tag{15}$$

is the correlation function. An equivalent derivation for the Q arm leads to

$$\begin{aligned}y_{n,Q} &= \frac{|a_0|}{2}K\frac{\sin\left(\pi\overline{\Delta f}T_{\mathrm{int}}\right)}{\pi\overline{\Delta f}T_{\mathrm{int}}}d\left([n]_{T_b/T_{\mathrm{int}}}\right)R_{\widetilde{p}q}\left(\overline{\Delta\tau_0}\right)\\ &\quad\cdot\sin\left(\pi\overline{\Delta f}T_{\mathrm{int}}+\overline{\Delta\phi}\right)+v_Q(n).\end{aligned} \tag{16}$$

Terms $\overline{\Delta f}$, $\overline{\Delta\phi}$, and $\overline{\Delta\tau_0}$ should be regarded as the average local phase error over the integration interval, that is, $\overline{\Delta\phi} = \Delta\phi + 2\pi\Delta f(T_{\mathrm{int}}/2)$, assuming a frequency rate error $\Delta\dot{f}$

(i.e., a phase acceleration error) equal to zero. In case of inclusion of such effect in the model, the average phase error can be expanded as

$$\overline{\Delta\phi} = \Delta\phi + 2\pi\Delta f \frac{T_{\text{int}}}{2} + 2\pi \frac{T^2}{6}\Delta\dot{f}. \tag{17}$$

In this expression, the terms $\Delta\phi$, Δf, and $\Delta\dot{f}$ are referred to the error values at the beginning of the integration interval.

In the following, we will consider $K = (T_{\text{int}}/T_s)$ as the integer number of samples collected in an accumulation. This number will not be integer in receiver configurations having a sample rate incommensurable with the chip rate, and thus some integration blocks will have $K + 1$ samples instead of K. This effect can be considered negligible for the analysis presented in this paper.

In the case of $M_i > 1$ (i.e., in the presence of multipath), (12) becomes a sum of all the replicas convoluted with a filter matched to the line of sight signal, whose estimated parameters are possibly biased by the presence of multipath. Since the convolution is a linear operator, the correlator output will be a linear combination of the contributions made by each signal path.

Note that an arbitrary number of correlators (very early, early, prompt, late, very late, etc.) can be used in the filter update, just adding or subtracting the correlator offset to the argument of $R_{\tilde{p}q}$ (i.e., $R_{\tilde{p}q}(\Delta\tau_{0,n} + \delta)$, $R_{\tilde{p}q}(\Delta\tau_{0,n} - \delta)$, etc.). The correlators' output can be stacked in a vector $\mathbf{y}_n$, which will be the measurements used in next section.

In the context of this work, we used the GRANADA (Galileo Receiver ANAlysis and Design Application) simulation platform to simulate realistic channel and receiver scenarios. The GRANADA Factored Correlator Model (FCM) blockset (see Figure 2) is a MATLAB/Simulink (MATLAB and Simulink are registered trademarks of The MathWorks, Inc.) library that provides a swift, flexible, and realistic way of simulating different signal processing architectures, either of standalone GNSS receivers or multisystem solutions. The FCM was included in a Simulink blockset, which, since 2007, has been commercially available as part of the GRANADA product family, whose remaining products were developed by DEIMOS Space in the frame of the Galileo Receiver Development activities (GARDA), funded by the Galileo Joint Undertaking (now European GNSS Agency, GSA) under the 6th Framework Program of the European Union.

The FCM separates the effects of carrier and code Doppler and misalignment on a GNSS receiver's correlator outputs into several multiplicative factors and allows the inclusion (or not) of each factor independently. Since it is an analytical model, the computation rate can be as low as the tracking loop rate, dramatically increasing simulation speed: the FCM provides directly the correlators' output, precluding the need of simulating the lower-level signal processing stages, significantly reducing the computational load and hence decreasing processing and memory requirements, while still accounting for various effects (as filtering, carrier phase and frequency errors, code delay error, code Doppler, noise, and multipath), thus keeping a high level of realism [15]. Since, statistically speaking, it is equivalent to work with samples before or after the correlation process (proof in the Appendix), we take advantage of working at the correlator output since it considerably reduces the computational load.

Once configured (type of signal, propagation channel, user dynamics, sampling frequency before correlation, number of correlators and their spacing, integration period, environment, etc., see Figure 3), FCM provides the measurements $\mathbf{y}_n$ used in the simulations presented in Section 5.

3. Particle Filtering

Bayesian filtering involves the recursive estimation of states $\mathbf{x}_n \in \mathbb{R}^{n_x}$ given measurements $\mathbf{y}_n \in \mathbb{R}^{n_y}$ at time n based on all available measurements, $\mathbf{y}_{1:n} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. To that aim, we are interested in the filtering distribution $p(\mathbf{x}_n|\mathbf{y}_{1:n})$, which can be recursively expressed as

$$p(\mathbf{x}_n \mid \mathbf{y}_{1:n}) = \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_n\right) p(\mathbf{x}_n \mid \mathbf{x}_{n-1})}{p(\mathbf{y}_n \mid \mathbf{y}_{1:n-1})} p(\mathbf{x}_{n-1} \mid \mathbf{y}_{1:n-1}), \tag{18}$$

with $p(\mathbf{y}_n|\mathbf{x}_n)$ and $p(\mathbf{x}_n|\mathbf{x}_{n-1})$ referred to as the likelihood and the prior distributions, respectively. Unfortunately, (18) can only be obtained in closed-form in some special cases. For instance, when the model is linear and Gaussian, the Kalman Filter (KF) [16] provides the optimal solution. In more general setups—nonlinear and/or non-Gaussian—we should resort to more sophisticated methods [17]. In this paper we consider particle filters (PFs) [18, 19].

PFs approximate the filtering distribution by a set of N weighted random samples, forming the so-called set of particles $\{\mathbf{x}_n^{(i)}, w_n^{(i)}\}_{i=1}^N$. These random samples are drawn from the importance density distribution, $\pi(\cdot)$,

$$\mathbf{x}_n^{(\text{i})} \sim \pi\left(\mathbf{x}_n \mid \mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{1:n}\right), \tag{19}$$

and weighted according to the general formulation

$$w_n^{(i)} \propto w_{n-1}^{(i)} \frac{p\left(\mathbf{y}_n \mid \mathbf{x}_{0:n}^{(i)}, \mathbf{y}_{1:n-1}\right) p\left(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{n-1}^{(i)}\right)}{\pi\left(\mathbf{x}_n^{(i)} \mid \mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{1:n}\right)}. \tag{20}$$

Algorithm 1 outlines the operation of the Standard PF (SPF) when a new measurement $\mathbf{y}_n$ becomes available. After particle generation, weighting, and normalization, a minimum mean square error (MMSE) estimate can be obtained by a weighted sum of particles. A typical problem of PFs is the degeneracy of particles, where all but one weight tend to zero. This situation causes the particle to collapse to a single state point. To avoid the degeneracy problem, we apply resampling, consisting in eliminating particles with low importance weights and replicating those in high-probability regions [20, 21]. In this work, we consider a multinomial sampling scheme for the resampling step.

3.1. Rao-Blackwellized Particle Filter. In this paper, we analyze a way to alleviate the dimensionality problem based on the marginalization of linear states. The basic idea is that a KF can optimally deal with these states, while reducing the dimension of the state space that the nonlinear filter

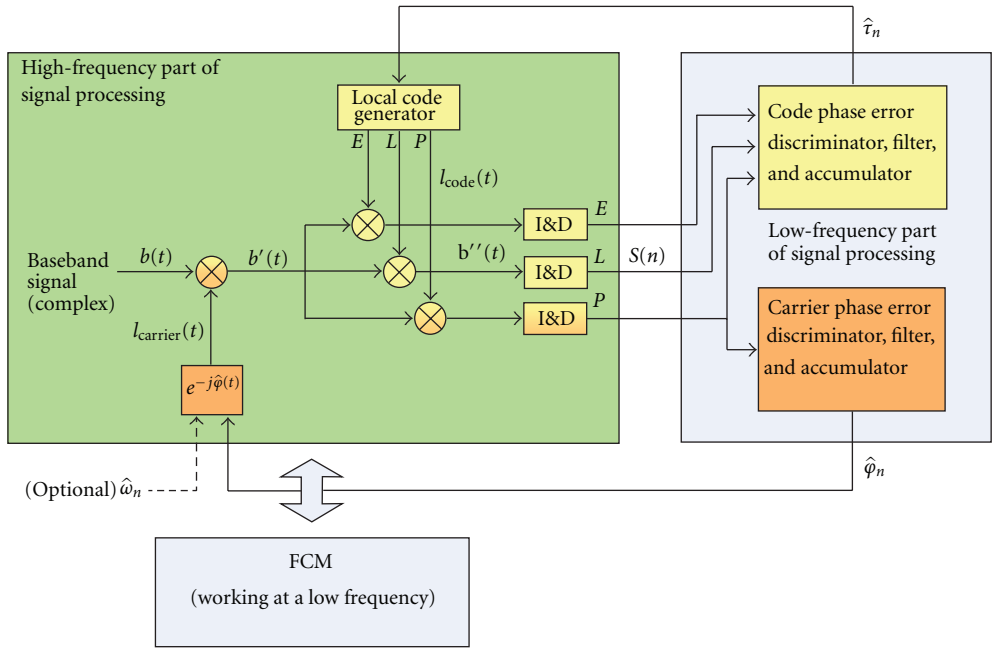


Figure 2: Schematic of the tracking stage of a GNSS receiver and substitution of the high frequency stage of the receiver (correlation and carrier wipe-off) with the FCM blockset [22].

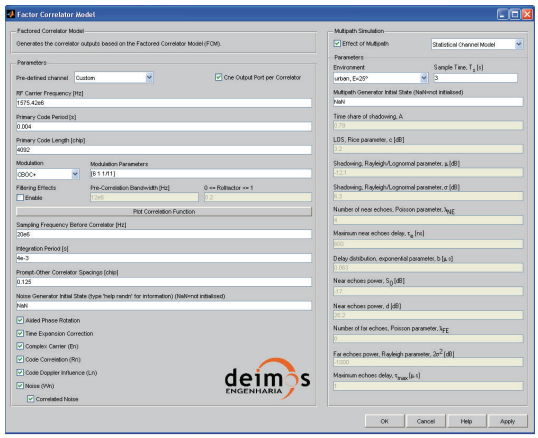


Figure 3: Configuration screen of the FCM blockset.

Require: $\{\mathbf{x}_{n-1}^{(i)}, w_{n-1}^{(i)}\}_{i=1}^{N}$ and $\mathbf{y}_n$
Ensure: $\{\mathbf{x}_n^{(i)}, w_n^{(i)}\}_{i=1}^{N}$ and $\hat{\mathbf{x}}_n$
1: **for** $i = 1$ to N **do**
2: Generate $\mathbf{x}_n^{(i)} \sim \pi(\mathbf{x}_n|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{1:n})$
3: Calculate $\tilde{w}_n^{(i)} = w_{n-1}^{(i)} \dfrac{p(\mathbf{y}_n|\mathbf{x}_{0:n}^{(i)}, \mathbf{y}_{1:n-1}) p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{\pi(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)}, \mathbf{y}_{1:n})}$
4: **end for**
5: **for** $i = 1$ to N **do**
6: Normalize weights: $w_n^{(i)} = \dfrac{\tilde{w}_n^{(i)}}{\sum_{j=1}^{N} \tilde{w}_n^{(j)}}$
7: **end for**
8: MMSE state estimation: $\hat{\mathbf{x}}_n = \sum_{i=1}^{N} w_n^{(i)} \mathbf{x}_n^{(i)}$
9: $\{\mathbf{x}_n^{(i)}, 1/N\}_{i=1}^{N} = \text{Resample}(\{\mathbf{x}_n^{(i)}, w_n^{(i)}\}_{i=1}^{N})$

Algorithm 1: Standard particle filtering (SPF).

has to explore. The procedure was proposed in [23, 24] for the case of dealing with the nonlinear states with a PF. The algorithm was termed Marginalized particle filter (MPF), although the same concept is also referred to as Rao-Blackwellized PF (RBPF) in other works [25, 26]. The latter nomenclature is because marginalization resorts to a general result due to [27, 28] referred to as the Rao-Blackwell theorem, which shows that the performance of an estimator can be improved by using information about conditional probabilities. The Rao-Blackwell theorem states let $\hat{\theta} = g(\mathbf{x})$ represent any unbiased estimator for θ and $T(\mathbf{x})$ be a sufficient statistic for θ under $p(\mathbf{x}, \theta)$. Then the conditional expectation $\hat{\theta}^{\text{RB}} = \mathbb{E}\{g(\mathbf{x})|T(\mathbf{x})\}$ is independent of θ, and it is the uniformly minimum variance unbiased estimator (cf. [29, 30] for the details) The result of a corollary points out that the use of a Rao-Blackwellized estimator effectively reduces the variance of the estimation error. Therefore, when possible, it is desirable to apply marginalization procedures. Corollary: let $\hat{\theta}$ be an unbiased estimator and let $\hat{\theta}^{\text{RB}}$ be the Rao-Blackwell estimator, then

$$\mathbb{E}\left\{\left(\theta - \hat{\theta}\right)^2\right\} \geq \mathbb{E}\left\{\left(\theta - \hat{\theta}^{\text{RB}}\right)^2\right\}. \tag{21}$$

Final remarks on Rao-Blackwellization are worth mentioning.

(i) Rao-Blackwellization is a procedure suitable when linear substructures are present in the dynamical model.

(ii) It is a variance reduction technique, in the sense that the estimation variance of a filter considering this marginalization procedure is less than a filter estimating the complete state space.

(iii) Filtering linear states with a Kalman filter has twofold benefits: (1) linear states are optimally filtered and (2) the system coped by the nonlinear filter has reduced dimensionality (with large benefits in terms of computational resources).

4. Joint Filtering of LOSS and Multipath Parameters

The technique herein investigated attempts to estimate the synchronization parameters of both the LOSS and $M-1$ multipath components. We refer to the algorithm as the multipath estimating particle filtering, or MEPF for short. Here the term Bayesian means that the algorithm is using some sort of a *priori* information regarding these parameters (such as interdependencies and time evolution models). This approach was first introduced in [31] and further refined in [32], although other papers might be found following the same scheme [33] with more complex time-evolving models. The application of Bayesian filtering techniques becomes straightforward when one describes the problem at hand in terms of a measurement equation and a process equation (i.e., how unknowns evolve randomly over time).

4.1. Observations. A receiver implementing such Bayesian tracking loops typically processes each satellite independently, and most of the work in the literature discusses architectures using IF signal. Here we are interested in operating at the output of the bank of correlators.

Observations for the i-th satellite are gathered into a random vector $\mathbf{y}_n$, where we omitted the subindex i for the sake of clarity. The ℓth element in $\mathbf{y}_n$ corresponds to the sample of the ℓ-th correlator, and it is expressed as

$$y_{n,\ell}(\boldsymbol{\alpha}_n, \phi_n, \boldsymbol{\tau}_n) = \sum_{m=0}^{M-1} \alpha_{m,n} e^{j\Delta\phi_{m,n}} R_{n,\ell}(\Delta\tau_{\ell,m,n}) + \nu_{n,\ell}, \quad (22)$$

accounting that $\Delta\tau_{\ell,m,n} = \tau_{m,n} - \widehat{\tau}_{0,n-1} + \delta_\ell$ corresponds to the point where the ℓ-th early/late sample is evaluated. As usual, $m = 0$ denotes LOSS. Here we consider a noncoherent tracking architecture that operates with the squared outputs. This scheme avoids the estimation of carrier phases, and thus it reduces the state-space dimension. In our implementation, a conventional PLL/FLL network is used in parallel to the MEPF. Therefore, the observations are the parallel outputs of the correlation bank, which we denote as

$$\mathbf{y}_n = \left(|y_{n,1}(\boldsymbol{\alpha}_n, \boldsymbol{\tau}_n)|^2, \ldots, |y_{n,L}(\boldsymbol{\alpha}_n, \boldsymbol{\tau}_n)|^2 \right)^T, \quad (23)$$

where L is the total number of correlators used at the receiver. We made apparent the dependence of measures on unknown states: real amplitude ($\boldsymbol{\alpha}_n$) and time delay ($\boldsymbol{\tau}_n$) of each replica m of the signal.

4.2. Process Dynamics. The state space is composed of the unknown parameters of the model, namely, delay, delay rate, and real amplitude of the LOSS and its multipath replica:

$$\mathbf{x}_n = \left(\underbrace{\tau_{0,n}, \ldots, \tau_{M-1,n}}_{\boldsymbol{\tau}_n}, \underbrace{\dot{\tau}_{0,n}, \ldots, \dot{\tau}_{M-1,n}}_{\dot{\boldsymbol{\tau}}_n}, \underbrace{\alpha_{0,n}, \ldots, \alpha_{M-1,n}}_{\boldsymbol{\alpha}_n} \right)^T, \quad (24)$$

where $\dot{\tau}_{m,n}$ is the delay rate of the m-th component, related to the Doppler shift. We have introduced this delay rate to better capture the dynamics of the time-evolving delay of the signals.

One could adopt many alternatives to specify the time-evolving processes for each state, ranging from the simplistic (although effective in some situations) autoregressive model to more sophisticated models. Here, we adopt a channel state model based on that presented in [34], adapted to the noncoherent scheme. This model was motivated by channel modeling work for multipath prone environments such as the urban satellite navigation channel [35].

The dynamics of time delay and delay rate for the LOSS (i.e., $m = 0$) are described by

$$\begin{pmatrix} \tau_{0,n} \\ \dot{\tau}_{0,n} \end{pmatrix} = \begin{pmatrix} 1 & T_{\text{int}} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tau_{0,n-1} \\ \dot{\tau}_{0,n-1} \end{pmatrix} + \mathbf{u}_{0,n}^{\tau}, \quad (25)$$

where T_{int} is the integration period and the process noise is an uncorrelated zero-mean Gaussian random variable with diagonal entries $\sigma_{0,\tau}^2$ and $\sigma_{0,\dot{\tau}}^2$.

The evolution of $\tau_{m,n}$ and $\dot{\tau}_{m,n}$ for the echoes is modeled with a truncated Gaussian distribution as in [31], which allows us to introduce the fact that due to physical reasons

$$\tau_{m,n} > \tau_{0,n} \quad \forall m \in \{1, \ldots, M-1\}, \quad (26)$$

in outdoor propagation channels [6, 11, 36]. Taking (26) into account, we force this situation using the evolution

$$\begin{aligned} \tau_{m,n} &= \tau_{0,n} + \left| \tau_{m,n-1} + u_{m,n}^{\tau} \right|, \\ \dot{\tau}_{m,n} &= \dot{\tau}_{0,n} + u_{m,n}^{\dot{\tau}}, \end{aligned} \quad (27)$$

with $u_{m,n}^{\tau}$ and $u_{m,n}^{\dot{\tau}}$ being zero-mean Gaussian random variables with variances $\sigma_{m,\tau}^2$ and $\sigma_{m,\dot{\tau}}^2$, respectively. For the evolution of each $\alpha_{m,n}$ we consider independent autoregressive models with variance $\sigma_{m,\alpha}^2$. The overall covariance matrix of the process is denoted as $\boldsymbol{\Sigma}_x$ and is constructed with $\sigma_{0,\tau}^2$, $\sigma_{m,\tau}^2$, $\sigma_{0,\dot{\tau}}^2$, $\sigma_{m,\dot{\tau}}^2$, $\sigma_{0,\alpha}^2$, and $\sigma_{m,\alpha}^2$ in its diagonal.

4.3. Algorithm Implementation. From the previous modeling, we realize that the state space can be partitioned into linear and nonlinear subspaces. Clearly, these can be identified as

$$\begin{aligned} \mathbf{x}_n^{\text{l}} &= \boldsymbol{\alpha}_n, \\ \mathbf{x}_n^{\text{nl}} &= \left(\boldsymbol{\tau}_n^T, \dot{\boldsymbol{\tau}}_n^T \right)^T. \end{aligned} \quad (28)$$

By the chain rule of probability, linear states can be analytically marginalized out from $p(\mathbf{x}_n \mid \mathbf{y}_{1:n})$:

$$p\left(\mathbf{x}_n^{\mathrm{l}}, \mathbf{x}_{0:n}^{\mathrm{nl}} \mid \mathbf{y}_{1:n}\right) = p\left(\mathbf{x}_n^{\mathrm{l}} \mid \mathbf{x}_{0:n}^{\mathrm{nl}}, \mathbf{y}_{1:n}\right) p\left(\mathbf{x}_{0:n}^{\mathrm{nl}} \mid \mathbf{y}_{1:n}\right) \quad (29)$$

and, taking into consideration that $\mathbf{x}_n^{\mathrm{l}}$ generates a linear Gaussian state-space, $p(\mathbf{x}_n^{\mathrm{l}} \mid \mathbf{x}_{0:n}^{\mathrm{nl}}, \mathbf{y}_{1:n})$ can be updated analytically via a KF conditional on $\mathbf{x}_{0:n}^{\mathrm{nl}}$ and only the nonlinear part of $\mathbf{x}_n$ needs to be estimated with a nonlinear filter. In the proposed scheme, an SPF is run to characterize $p(\mathbf{x}_{0:n}^{\mathrm{nl}} \mid \mathbf{y}_{1:n})$ and a KF is executed to obtain $p(\mathbf{x}_n^{\mathrm{l}} \mid \mathbf{x}_{0:n}^{\mathrm{nl}}, \mathbf{y}_{1:n})$.

Notice that both linear and nonlinear states are interdependent, thus the algorithm has to be aware of this coupling. The details might be consulted in [23] for the general algorithm and in [12] for the specific GNSS setup considered here. At a glance, each particle in the PF has an associated KF that tracks amplitudes. Then, before particle generation, KF prediction is run and the results are used in the particle filter. Similarly, once particles are weighted this information is used in the update step of the KF.

5. Results in Realistic Scenarios

We used the GRANADA FCM blockset of Simulink to simulate the GPS L1 C/A signal, the propagation channel, and the inaccuracies of the receiver front end. An initial set of controlled scenarios is simulated to analyze the method. Then, from the set of reviewed channel models, we have selected Jahn's to show simulation results in a realistic environment. The GPS signal is spread spectrum with a code length of 1023 chips and a chip rate of 1.023 Mchips/s (notice that a chip of the signal corresponds to approximately 300 meters in length and the duration of an entire codeword is one millisecond). The carrier frequency of the transmitted signal was 1575.42 MHz and the receivers precorrelation bandwidth was 2 MHz. Estimates of time delay were performed at a rate of 50 Hz, which corresponds to an integration time of 20 milliseconds, assuming bit synchronization. The carrier-to-noise density ratio ($\mathrm{C/N_0}$) of the simulated satellite was 38 dB-Hz. The dynamics of the scenario were due to the relative motion of the satellite-receiver, which is completely simulated by the GRANADA FCM blockset, and the receiver performed a pedestrian-like trajectory at 1 m/s. Simulation time was 50 seconds.

We compared the performance of the MEPF with the results of a narrow 0.125-chip spacing DLL (state-of-the-art in GNSS receivers) with an equivalent noise bandwidth of 1 Hz. This architecture uses 3 correlators. Also, the benchmark receiver implements a coherent phase lock loop (PLL) carrier phase discriminator using a second-order filter and an error accumulator with equivalent noise bandwidth 10 Hz. The initial time-delay ambiguity at which the filter was initialized was drawn from $\mathcal{N}(\tau_{0,0}, T_{\mathrm{chip}}/2)$, with T_{chip} the chip period.

It has been reported in [37] that the number of correlators (L) used in the PF plays an important role. For instance, in AWGN on the order of $L = 11$ correlators are required to obtain stable results. Also, the algorithm improves its performance with the number of particles although this improvement saturates at 300 particles.

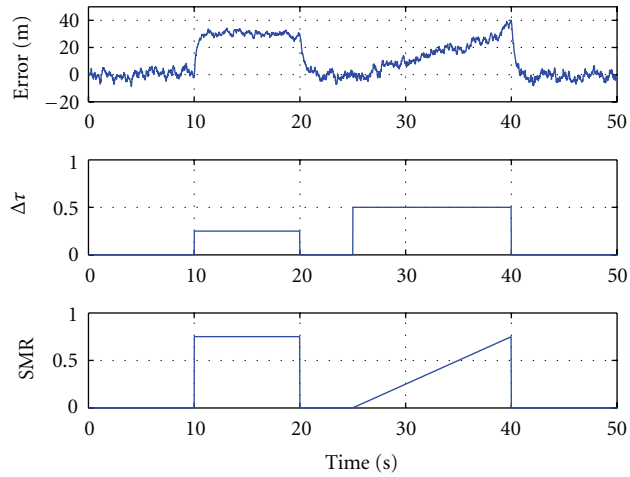


Figure 4: Evolution of pseudorange error in DLL/PLL scheme under severe multipath propagation.

Figures 4–7 show the behavior of the classical DLL/PLL scheme and the proposed MEPF, respectively, in a multipath scenario. In this experiments, we used $L = 21$ correlators for the MEPF in order to span correlators along regions of interest in terms of multipath estimation and mitigation. The results are organized as follows. Top figure represents the obtained pseudorange error. Central figure is the relative delay between the LOSS and the multipath replica, in the first representative interval ($t \in [10, 20]$) it has been set to 0.25 chips and in the second interval ($t \in [25, 40]$) to 0.5 chips. Bottom figure plots the signal-to-multipath ratio (SMR) in linear scale of the simulated scenario. During the first interval, the SMR was abruptly kept constant to 0.75 and during the second interval it grew linearly from 0 to 0.75. Since the MEPF is very sensitive to the tuning of process covariance matrix—as many Bayesian filtering solutions,—we have investigated three different setups with $N = 1000$ particles. Namely, (i) in Figure 5 we used standard deviations $\sigma_{0,\tau} = .03/c$, $\sigma_{0,\dot{\tau}} = 0.03/c$, $\sigma_{m,\tau} = 100/c$, $\sigma_{m,\dot{\tau}} = 0.03/c$, $\sigma_{0,\alpha} = 0.0001$, and $\sigma_{m,\alpha} = 0.01$; (ii) in Figure 6 we used $\sigma_{0,\tau} = 30/c$, $\sigma_{0,\dot{\tau}} = 0.3/c$, $\sigma_{m,\tau} = 30/c$, $\sigma_{m,\dot{\tau}} = 0.3/c$,$\sigma_{0,\alpha} = 0.0001$, and $\sigma_{m,\alpha} = 0.0001$; and finally (iii) in Figure 7 we used $\sigma_{0,\tau} = 3/c$, $\sigma_{0,\dot{\tau}} = 0.3/c$, $\sigma_{m,\tau} = 30/c$, $\sigma_{m,\dot{\tau}} = 0.3/c$,$\sigma_{0,\alpha} = 0.0001$, and $\sigma_{m,\alpha} = 0.01$. At the light of the results, the latter configuration provided a good performance as it allowed for sufficient delay excursions to explore the state space and fast variations in multipath amplitude were coped. A summary of results in terms of bias, variance, and RMSE over the entire simulation can be consulted in Table 2. We can observe that, compared to DLL schemes, a remarkable performance improvement can be obtained after properly adjusting the covariances.

Finally, we tested the algorithm in a more realistic scenario. We selected the Jahn's channel model with the same receiver parameters as before. Particularly, the considered channel was that of a satellite at an elevation angle of 55° in an urban scenario with an average $\mathrm{C/N_0}$ of 38 dB-Hz.

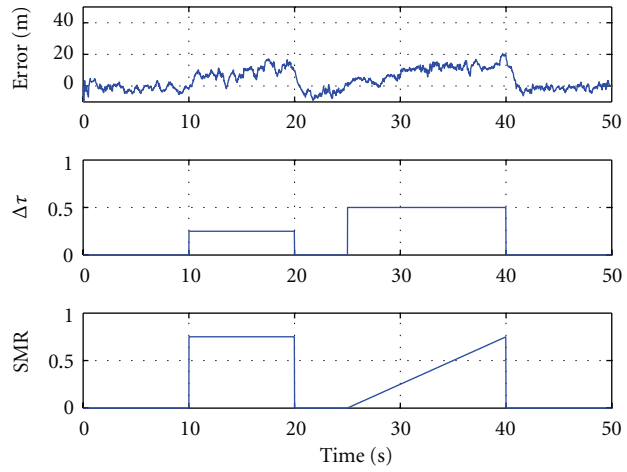


Figure 5: Evolution of pseudorange error in MEPF scheme under severe multipath propagation, setup number 1.

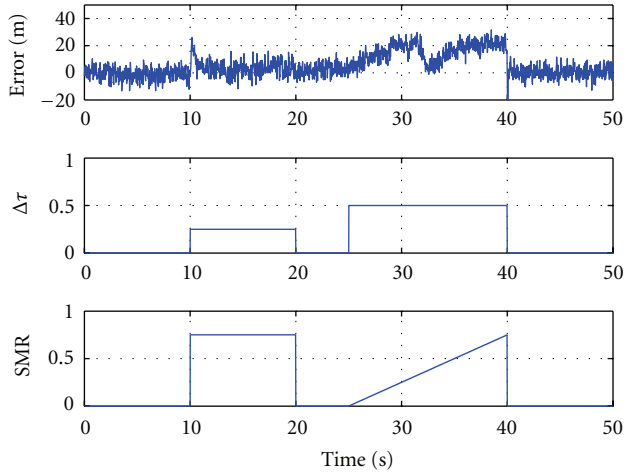


Figure 6: Evolution of pseudorange error in MEPF scheme under severe multipath propagation, setup number 2.

Table 2: Comparison of pseudorange error metrics in meters.

	DLL/PLL	MEPF (no. 1)	MEPF (no. 2)	MEPF (no. 3)
Bias (m)	10.66	3.97	5.48	1.24
std dev (m)	13.23	6.42	8.75	4.05
RMSE (m)	17.00	7.55	10.33	4.24

The results can be consulted in Figure 8, where it can be observed that MEPF requires an initial convergence time (depending on the covariance matrix set) larger than DLL schemes. Conversely, it appears more robust to channel impairments. Numerically, the RMSE in the overall simulation is of 8.48 m and 4.82 m for DLL and MEPF, respectively. For the MEPF we used $M = 2$ paths, $N = 1000$ particles, and $\sigma_{0,\tau} = .03/c$, $\sigma_{0,\dot{\tau}} = 0.03/c$, $\sigma_{m,\tau} = 3/c$, $\sigma_{m,\dot{\tau}} = 0.03/c$, $\sigma_{0,\alpha} = 0.0001$, and $\sigma_{m,\alpha} = 0.001$.

6. Conclusions

In this paper we have analyzed an advanced tracking loop for time-delay and carrier-phase estimation in a GNSS receiver based on sequential Monte-Carlo methods. The algorithm builds upon previous work by the authors on Rao-Blackwellized particle filtering while introducing more realistic process dynamics and the usage of postcorrelation observations, that reduce the computational burden at the receiver. The paper presents the general signal model, GNSS concept, and trade-offs the most common propagation channel models. A realistic scenario simulator based on the FCM blockset of Simulink was used Section 5. Results point out the need for properly setting not only the number of particles but the number of correlation outputs used as observations. Also, degradation of conventional DLL/PLL schemes in multipath-rich scenarios became clear. Nevertheless, the correct selection of a process covariance matrix was seen to affect significantly the performance of the MEPF and future work should be devoted in self-adjustment of such matrix.

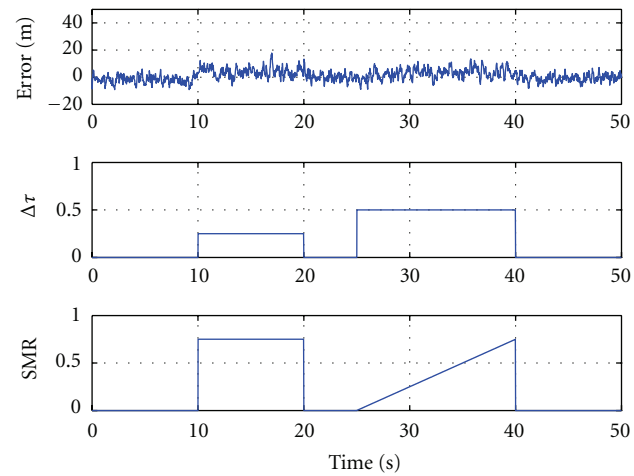


Figure 7: Evolution of pseudorange error in MEPF scheme under severe multipath propagation, setup number 3.

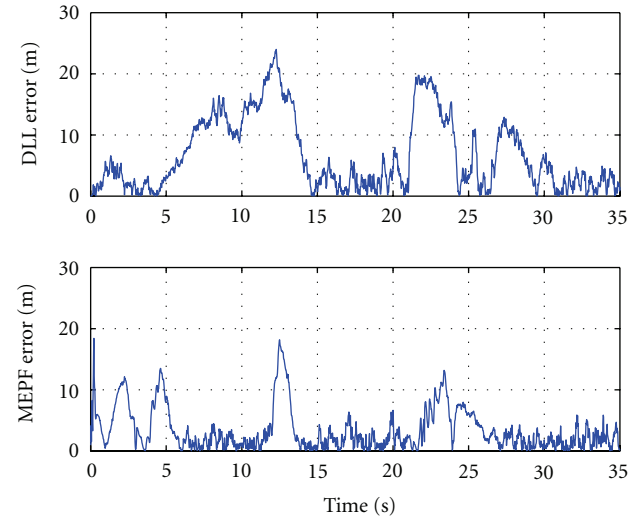


Figure 8: Pseudorange error in DLL and MEPF schemes in an urban scenario at an elevation of 55°, as modeled by Jahn's model.

Appendix

Equivalence of Pre/Postcorrelation Receiver Architectures

In this appendix we establish a basic result showing the equivalence between processing pre- and postcorrelation signals. That is to say, from a statistical point of view, an estimator of a given parameter (e.g., time delay) computed using a bunch of snapshots taken at the IF signal level ($r_0(t;\boldsymbol{v})$) is the same as that which is derived using the output of the correlators ($y(t;\boldsymbol{v})$). It is a well-known result in statistical signal processing that both signals are sufficient statistics, and thus one is able to derive an estimator of $\boldsymbol{v} = (\tau_0, f_{d_0}, \phi_0)^T$ using either. However, we will see that this equivalence becomes evident when one examines the likelihood distribution (the density where the information from measurements is gathered) for each approach.

If we analyze first the case of using the IF signal we should be aware of the following.

(i) This approach does not force an implementation based on early, prompt, and late samples; as observations are directly the baseband signal at the sampling frequency.

(ii) It is necessary to use a sufficiently large set of IF data to be able to infer any parameter from it. That is, one has to integrate over a certain integration time, T_{int}, since the signal-to-noise ratio of GNSS signals is typically well below the noise level.

The term $\mathbf{r}_n \in \mathbb{C}^K$ stands for the vector of snapshots of the IF signal, as gathered for the nth integration interval, defined as

$$\mathbf{r}_n = \begin{pmatrix} r_0(nT_{\text{int}}) \\ \vdots \\ r_0(nT_{\text{int}} + (K-1)T_s) \end{pmatrix}, \tag{A.1}$$

using the same notation conventions used along the document. Then, the likelihood can be decomposed as the independent contribution of each snapshot

$$\log(p(\mathbf{r}_n \mid \boldsymbol{v}_n)) = \prod_{k=1}^{K} p\left(\underbrace{r_0(nT_{\text{int}} + (k-1)T_s)}_{r_{0,n}(k)} \mid \boldsymbol{v}_n \right), \tag{A.2}$$

and assuming Gaussianity for the noise term, we could identify that

$$\log(p(\mathbf{r}_n \mid \boldsymbol{v}_n)) \propto -\sum_{k=1}^{K} \left\| r_{0,n}(k) - h_n^{(1)}(k;\boldsymbol{v}_n) \right\|^2, \tag{A.3}$$

where $h_n^{(1)}$ stands for the precorrelation signal model, which was defined earlier as

$$h_n^{(1)}(t;\boldsymbol{v}_n) = |a_0(t)| e^{j(2\pi f_{d_0} t + \phi_0)} d_n \widetilde{p}(t - \tau_0). \tag{A.4}$$

Further manipulation of the loglikelihood yields to

$$\begin{aligned} \log(p(\mathbf{r}_n \mid \boldsymbol{v}_n)) &\propto \sum_{k=1}^{K} |\, r_{0,n}(k) \widetilde{p}^*(t - \boldsymbol{v}_n) \,|^2 - |\, r_{0,n}(k) \,|^2 \\ &\propto |\, y_n(\boldsymbol{v} - \boldsymbol{v}_n) \,|^2, \end{aligned} \tag{A.5}$$

with the latter step being clear if one accounts for the definition of y_n as the output of a correlator. Recall that $\boldsymbol{v}$ is the true unknown parameter of the signal. An ML estimator of $\boldsymbol{v}$ could be obtained after maximizing the latter equation.

Drawbacks of this approach are twofold.

(i) It might be computationally expensive as large data sets need to be processed to increase the signal-to-noise ratio, and thus K might be large depending on T_s.

(ii) There is a requirement for performing signal processing operations at a high rate, since it operates at the sampling frequency.

If we turn our attention to the conventional approach in which one uses samples at the output of a bank of correlators, we should see the following.

(i) This approach forces an implementation based on early, prompt, and late samples; this means that samples are taken assuming a previous estimation (prompt) of the parameters, denoted as $\check{\boldsymbol{v}} = (\check{\tau}_0, \check{f}_{d_0}, \check{\phi}_0)^T$.

(ii) Few samples are sufficient to infer estimates of $\boldsymbol{v}$. After correlation an integration over a certain interval is already done, T_{int}, and therefore the signal-to-noise ratio is relatively high.

In this case, measurements can be expressed as $y_n(\boldsymbol{v} - \check{\boldsymbol{v}})$ at the output of the n-th integration interval. In this measurement we explicitly expressed that samples are taken with respect to the error between true and prompt parameters, $\Delta\boldsymbol{v} = \boldsymbol{v} - \check{\boldsymbol{v}}$. Notice that we considered that only the prompt is used for the sake of clarity. It is easy to obtain a similar result, as the one shown here, when one accounts for several early and late samples.

Then, the log-likelihood under the Gaussian assumption is

$$\begin{aligned} \log(p(y_n \mid \boldsymbol{v}_n)) &\propto \left\| y_n(\Delta\boldsymbol{v}) - h_n^{(2)}(\boldsymbol{v}_n - \check{\boldsymbol{v}}) \right\|^2 \\ &\propto \Re\left\{ y_n(\Delta\boldsymbol{v}) \left(h_n^{(2)}(\boldsymbol{v}_n - \check{\boldsymbol{v}}) \right)^* \right\}, \end{aligned} \tag{A.6}$$

with $h_n^{(2)}$ being the postcorrelation signal model

$$\begin{aligned} h_n^{(2)}(\Delta\boldsymbol{v}) = \frac{|a_0|}{2} K \frac{\sin(\pi\Delta f T_{\text{int}})}{\pi\Delta f T_{\text{int}}} d\left([n]_{T_b/T_{\text{int}}} \right) \\ \cdot R_{\widetilde{p}q}(\Delta\tau_0) \cos(\pi\Delta f T_{\text{int}} + \Delta\phi), \end{aligned} \tag{A.7}$$

and $\boldsymbol{v}_n$ the unknown parameter we want to estimate at n.

If we set $\boldsymbol{v}_n = \check{\boldsymbol{v}}$, we can identify that

$$\begin{aligned}\log(p(y_n \mid \boldsymbol{v}_n)) &\propto \Re\left\{y_n(\boldsymbol{v}-\boldsymbol{v}_n)\left(h_n^{(2)}(0)\right)^*\right\} \\ &\propto \left|y_n(\boldsymbol{v}-\boldsymbol{v}_n)\right|^2.\end{aligned} \tag{A.8}$$

From the latter mathematical derivations, we can conclude an important result:

$$\boxed{\log(p(\mathbf{r}_n \mid \boldsymbol{v}_n)) \propto \log(p(y_n \mid \boldsymbol{v}_n))} \tag{A.9}$$

for a given integration interval T_{int} considering KT_s snapshots. As said, similar results apply for larger integration and more early/late samples.

As a consequence, we can state the following: the ML estimator of $\boldsymbol{v}$ computed from the data sets $\mathbf{r}_n$ and y_n is equivalent.

To sum up, from a statistical point of view, both approaches are equivalent and the choice should be made considering implementation aspects. For instance, it is clear that using precorrelation measurements $\mathbf{r}_n$ involves larger computational burden than using post-correlation samples. Another important conclusions is that since in the precorrelation approach we also need to integrate in order to increase the signal-to-noise ratio, effects happening faster than T_{int} will not be captured by the estimation algorithm. The same happens in the post-correlation case. Therefore, the limitation of which phenomena could be tracked is inherent to the GNSS signal, instead of the way it is processed (i.e., pre-or postcorrelated samples).

Acknowledgment

P. Closas and C. Fernández-Prades were supported by the European Commission under COST Action IC0803 (RFCSET).

References

[1] M. Hernández-Pajares, J. M. J. Zornoza, J. S. Subirana, R. Farnworth, and S. Soley, "EGNOS test bed ionospheric corrections under the October and November 2003 storms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 10, pp. 2283–2293, 2005.

[2] S. Bancroft, "An algebraic solution of the GPS equations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 21, no. 1, pp. 56–59, 1985.

[3] G. Strang and K. Borre, *Linear Algebra, Geodesy, and GPS*, Wellesley Cambridge Press, 1997.

[4] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS -Global Navigation Satellite Systems: GPS, GLONASS, Galileo & More*, Springer-Verlag, Wien, Austria, 2008.

[5] C. Fernández-Prades, L. L. Presti, and E. Falletti, "Satellite radiolocalization from GPS to GNSS and beyond: novel technologies and applications for civil mass market," *Proceedings of the IEEE*, vol. 99, no. 11, pp. 1882–1904, 2011.

[6] A. Jahn, H. Bischl, and G. Heiss, "Channel characterization for spread spectrum satellite communications," in *Proceedings of the 4th International Symposium on Spread Spectrum Techniques & Applications (ISSSTA '96)*, pp. 1221–1226, September 1996.

[7] C. Loo and J. S. Butterworth, "Land mobile satellite channel measurements and modeling," *Proceedings of the IEEE*, vol. 86, no. 7, pp. 1442–1462, 1998.

[8] M. A. V. Castro, F. P. Fontan, A. A. Villamarín, S. Buonomo, P. Baptista, and B. Arbesser, "L-band Land Mobile Satellite (LMS) amplitude and multipath phase modeling in urban areas," *IEEE Communications Letters*, vol. 3, no. 1, pp. 12–14, 1999.

[9] F. P. Fontán, M. Vázquez-Castro, C. E. Cabado, J. P. García, and E. Kubista, "Statistical modeling of the LMS channel," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 6, pp. 1549–1567, 2001.

[10] A. Steingass and A. Lehner, "A channel model for land mobile satellite navigation," in *Proceedings of the the European Navigation Conference*, pp. 2132–2138, German Institute of Navigation (DGON), July 2005.

[11] Recommendation ITU-R P.681-7, "Propagation data required for the design of Earth-space land mobile telecommunication systems," 2009, http://www.itu.int/rec/R-REC-P.681-7-200910-I/en/.

[12] P. Closas, *Bayesian signal processing techniques for GNSS receivers: from multipath mitigation to positioning [Ph.D. dissertation]*, Universitat Politècnica de Catalunya (UPC), Department of Signal Theory and Communications, Barcelona, Spain, 2009.

[13] D. M. Akos, M. Stockmaster, J. B. Y. Tsui, and J. Caschera, "Direct bandpass sampling of multiple distinct RF signals," *IEEE Transactions on Communications*, vol. 47, no. 7, pp. 983–988, 1999.

[14] B. Parkinson and J. Spilker, Eds., *Global Positioning System: Theory and Applications*, vol. 1 of *Progress in Astronautics and Aeronautics*, American Institute of Aeronautics, Washington, DC, USA, 1996.

[15] J. S. Silva, P. F. Silva, A. Fernández, J. Diez, and J. F. M. Lorga, "Factored correlator model: a solution for fast, flexible, and realistic GNSS receiver simulations," in *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS '07)*, pp. 2676–2686, Fort Worth, TX, USA, September 2007.

[16] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[17] Z. Chen, "Bayesian filtering: from Kalman filters to particle filters, and beyond," Tech. Rep., Adaptive Systems Laboratory, McMaster University, Ontario, Canada, 2003.

[18] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[19] P. M. Djurić, J. H. Kotecha, J. Zhang et al., "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[20] M. Bolić, P. M. Djuric, and S. Hong, "Resampling algorithms for particle filters: a computational complexity perspective," *Eurasip Journal on Applied Signal Processing*, vol. 2004, no. 15, pp. 2267–2277, 2004.

[21] R. Douc, O. Cappé, and E. Moulines, "Comparison of resampling schemes for particle filtering," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA '05)*, pp. 64–69, Zagreb, Croatia, September 2005.

[22] GRANADA Galileo Receiver ANalysis And Design Application. The Reference Galileo Simulation Toolkit for GNSS Receiver Research And Development. Factored Correlator

Model Blockset v2.0 User Manual, Deimos Engenharia, S.A., 2009.

[23] T. Schön, F. Gustafsson, and P. J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, 2005.

[24] R. Karlsson, *Particle filtering for positioning and tracking applications [Ph.D. dissertation]*, Linköping University, Linköping, Sweden, 2005.

[25] R. Chen and J. S. Liu, "Mixture Kalman filters," *Journal of the Royal Statistical Society B*, vol. 62, no. 3, pp. 493–508, 2000.

[26] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[27] C. Rao, "Information and the accuracy attainable in the estimation of statistical parameters," *Bulletin of Calcutta Mathematical Society*, vol. 37, pp. 81–91, 1945.

[28] D. Blackwell, "Conditional expectation and unbiased sequential estimation," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 105–110, 1947.

[29] E. Lehmann, *Theory of Point Estimation. Probability and Mathematical Statistics*, John Wiley & Sons, 1983.

[30] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New Delhi, India, 4th edition, 2001.

[31] P. Closas, C. Fernández-Prades, and J. A. Fernández-Rubio, "Bayesian DLL for multipath mitigation in navigation systems using particle filters," in *Proceedings of the IEEE (ICASSP '06)*, Toulouse, France, May 2006.

[32] P. Closas, C. Fernández-Prades, and J. A. Fernández-Rubio, "A Bayesian approach to multipath mitigation in GNSS receivers," *IEEE Journal on Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 695–706, 2009.

[33] M. Lentmaier, B. Krach, and P. Robertson, "Bayesian time delay estimation of GNSS signals in dynamic multipath environments," *International Journal of Navigation and Observation*, vol. 2008, Article ID 372651, 11 pages, 2008.

[34] B. Krach, P. Robertson, and R. Weigel, "An efficient two-fold marginalized Bayesian filter for multipath estimation in satellite navigation receivers," *Eurasip Journal on Advances in Signal Processing*, vol. 2010, Article ID 287215, 2010.

[35] A. Steingass and A. Lehner, "Measuring the navigation multipath channel—a statistical analysis," in *Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS '04)*, pp. 1157–1164, Long Beach, Calif, USA, September 2004.

[36] M. Irsigler, J. A. Ávila-Rodríguez, and G. W. Hein, "Criteria for GNSS multipath performance assessment," in *Proceedings of the International Technical Meeting of the Institute of Navigation(ION GPS/GNSS '05)*, Long Beach, Calif, USA, September 2005.

[37] P. Closas, C. Fernández-Prades, J. Diez, and D. de Castro, "Multipath estimating tracking loops in advanced GNSS receivers with particle filtering," in *Proceedings of the IEEE Aerospace Conference*, Big Sky, Mont, USA, March 2012.

Hindawi Publishing Corporation
International Journal of Navigation and Observation
Volume 2012, Article ID 135401, 14 pages
doi:10.1155/2012/135401

Research Article

Two-Step Galileo E1 CBOC Tracking Algorithm: When Reliability and Robustness Are Keys!

Aleksandar Jovanovic,[1] Cécile Mongrédien,[2] Youssef Tawk,[1] Cyril Botteron,[1] and Pierre-André Farine[1]

[1] *Electronics and Signal Processing Laboratory (ESPLAB), École Polytechnique Fédérale de Lausanne (EPFL), Rue A.-L. Breguet 2, 2000 Neuchâtel, Switzerland*
[2] *Fraunhofer Institute for Integrated Circuits IIS, Nordostpark 93, 90411 Nuernberg, Germany*

Correspondence should be addressed to Aleksandar Jovanovic, aleksandar.jovanovic@epfl.ch

Received 30 December 2011; Revised 4 April 2012; Accepted 22 April 2012

Academic Editor: Heidi Kuusniemi

The majority of 3G mobile phones have an integrated GPS chip enabling them to calculate a navigation solution. But to deliver continuous and accurate location information, the satellite tracking process has to be stable and reliable. This is still challenging, for example, in heavy multipath and non-line of sight (NLOS) environments. New families of Galileo and GPS navigation signals, such as Alternate Binary Offset Carrier (AltBOC), Composite Binary Offset Carrier (CBOC), and Time-Multiplex Binary Offset Carrier (TMBOC), will bring potential improvements in the pseudorange calculation, including more signal power, better multipath mitigation capabilities, and overall more robust navigation. However, GNSS signal tracking strategies have to be more advanced in order to profit from the enhanced properties of the new signals.In this paper, a tracking algorithm designed for Galileo E1 CBOC signal that consists of two steps, coarse and fine, with different tracking parameters in each step, is presented and analyzed with respect to tracking accuracy, sensitivity and robustness. The aim of this paper is therefore to provide a full theoretical analysis of the proposed two-step tracking algorithm for Galileo E1 CBOC signals, as well as to confirm the results through simulations as well as using real Galileo satellite data.

1. Introduction

New GPS and Galileo signals use new modulations, such as AltBOC, CBOC, and TMBOC that have the potential to improve navigation through advanced signal properties, such as more signal power, better multipath mitigation capabilities, and overall improved signal cross-correlation properties. Certainly, a major innovation brought by the new modulation schemes consists of the presence of two distinct components, namely, the data and pilot channels that carry two different pieces of information. The data channel contains the navigation message, whereas the pilot channel is dataless, allowing long coherent signal integration that, in turn, allows more precise determination of the ranging information. For carrier tracking, the presence of a pilot channel enables the combined use of pure PLL (Phase Lock Loop) discriminators and longer coherent integration time. Code tracking can be organized as data/pilot collaborative tracking [1, 2], where two channels (data and pilot) are used in the estimation of the code error, decreasing the thermal noise error and improving overall tracking.

Several tracking algorithms proposed for Galileo E1 CBOC signals were derived from tracking schemes developed for BPSK (Binary Shift Keying) and BOC(1,1) signals. BOC(1,1) tracking faces the problem of biased tracking as already explained in [3]. Until now, the main algorithms that were proposed to specifically address the problem of biased tracking for BOC(1,1) and can be applied to CBOC tracking too are Single Side Lobe, bump and jump algorithm [4], ASPeCT (Autocorrelation Side-Peak Cancellation Technique) [3], and Double Estimator [5, 6]. The Single Side Lobe technique provides a robust solution to resolving BOC ambiguity, but it is suitable only for low-precision receivers. Note that the ASPeCT algorithm modifies the

shape of the autocorrelation function and eliminates side peaks that can be points of false locking. However, it requires a different and more complex correlator architecture. Double Estimator assumes an additional loop, the Subcarrier Lock Loop (SLL), that tracks subcarrier delay. More correlators are needed, as well as more complex loop implementation. In addition, two tracking techniques have been proposed exclusively for Galileo E1 CBOC tracking: TM61 and Dual Correlator. The TM61 technique generates a one-bit local replica, either BOC(1,1) or BOC(6,1), simplifying the architecture, but also degrading tracking by at least 3 dB [7]. The Dual Correlator technique is based on the investigation of two parallel correlations: one between the incoming MBOC and a BOC(1,1) replica, and one between the incoming MBOC and a BOC(6,1) replica. Each correlation is weighted, and two outputs are linearly added, such that by changing the values of the weights the tracking can be easily modified [8]. These two E1 CBOC tracking techniques assume separate correlations, which degrades the tracking and brings additional complexity. In contrast, the two-step tracking algorithm considered in this paper is based on full four-level bit local replica generation and consists of two steps, coarse and fine, with different tracking parameters. The main requirements that were taken into consideration when defining this algorithm were (1) a relatively low implementation complexity suitable for a mass-market solution, (2) the shape of the E1 CBOC autocorrelation function, and (3) the correlator's structure since it conditions the tracking algorithm's properties. Regarding the second point, since the CBOC autocorrelation function has secondary peaks that can be potential false lock points, one of the main objectives was to design a tracking algorithm that can avoid such false locks or, at least, minimize their occurrence. Therefore, five complex correlators are used with several correlator spacing options (very early (VE) and a very late (VL) correlators have been added to the three conventional Early (E), Prompt (P), and Late (L) correlators). Regarding the third point, we wanted to design a tracking architecture that can minimize the tracking error caused by thermal noise and multipath. Moreover, DP (Dot-Product) and HRC (High Resolution Correlator) discriminators are used equally in the fine tracking step, depending on the tracking conditions.

This paper extends our previous results by fully analyzing the theoretical performance of the previously proposed two-step tracking schemes with respect to tracking accuracy, sensitivity, and robustness, as well as testing the algorithm using real satellite data. The paper is organized as follows. In Section 2, the Galileo E1 CBOC signal is described. Section 3 provides an overview of the two-step tracking algorithm with a thorough analysis of the discriminator curve outputs and discussion of its stability and linearity regions. Section 4 provides theoretical analysis of the tracking loops' performance, especially derivation and calculation of code tracking error and tracking thresholds for different discriminators. A multipath mitigation analysis is provided in Section 5, followed by simulation-based and realistic results provided in Section 6. Finally, conclusions and outlook are provided in Section 7.

2. Galileo E1 CBOC Signal Properties

New generations of Global Navigation Satellite Systems (Galileo, Glonass, Compass, etc.) are rapidly evolving, and old GPS system served localization purposes to users for more than twenty years in positioning that is currently undergoing a modernization process, and new European Galileo system will be available within a few years. MBOC modulation was chosen to be implemented on the Galileo E1 and GPS L1C signals [9]. The power density function of the MBOC modulation is a sum of the 10/11 normalized BOC(1,1) and the 1/11 normalized BOC(6,1) signal spectrum. MBOC is defined in the frequency domain, and two different implementations have been specified for Galileo and GPS: CBOC and TMBOC, respectively. Although having different time domain implementations, the power spectral densities in both configurations of the MBOC(6,1,1/11) are the same.

The GPS L1C signal has a pure BOC(1,1) data channel carrying 25% of the total signal power, while the pilot signal uses a TMBOC modulation with 75% of the total signal power. The Galileo E1 signal on the other hand shares its power equally between data and pilot channels, with both channels using a CBOC modulation based on a four-level subcarrier formed by the weighed sum of BOC(1,1) and BOC(6,1). To acquire 100% of the signal energy, a bandwidth of 16 MHz is needed. The only difference between data and pilot channels besides having different PRN (Pseudo-Random Noise) codes is in the sign of the weighted sum. The data channel is "in phase" (CBOC(6,1,"+")), and the pilot channel is in "antiphase" (CBOC(6,1,"−")). "Antiphase" configuration exhibits better tracking performance: autocorrelation peak is narrower. The analytical expression for the E1 CBOC signal can be represented as

$$\begin{aligned} s_{\mathrm{Gal}}(t) = \sqrt{\frac{1}{2}}\Big[& d_{\mathrm{Gal}}(t)c_d(t)\mathrm{CBOC}(6,1,p,\text{“+”})(t) \\ & + c_p(t)\mathrm{CBOC}(6,1,p,\text{“−”})(t)\Big], \\ \mathrm{CBOC}(6,1,p,\text{“+”})(t) = & \left(\sqrt{(1-p)}s_{\mathrm{BOC}(1,1)}(t) + \sqrt{p}s_{\mathrm{BOC}(6,1)}\right), \\ \mathrm{CBOC}(6,1,p,\text{“−”})(t) = & \left(\sqrt{(1-p)}s_{\mathrm{BOC}(1,1)}(t) - \sqrt{p}s_{\mathrm{BOC}(6,1)}\right), \end{aligned} \tag{1}$$

and $p = 1/11$ for OS (Open Service), $c_d(t), c_p(t)$ are the data and pilot spreading codes, and $\mathrm{CBOC}(6,1,p,$ "+") and $\mathrm{CBOC}(6,1,p,$"−") are the pilot and data subcarriers, respectively.

2.1. Galileo E1 CBOC Autocorrelation Function. The study of the autocorrelation function is important when trying to evaluate the tracking performance of the signal. The E1 CBOC autocorrelation function for the data and pilot channels, as well as for the combination of the two channels (for the combined tracking case), is shown in Figure 1 assuming an 18 MHz bandwidth. The CBOC autocorrelation function exhibits a very narrow main peak and two side-correlation peaks located at approximately 0.5 chips around

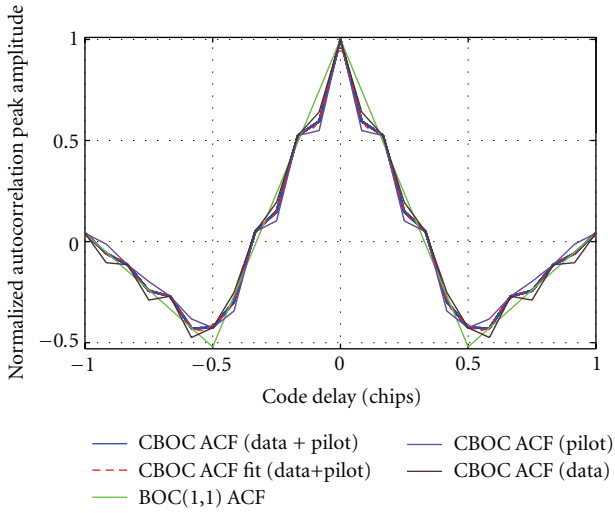


Figure 1: CBOC autocorrelation function for pilot, data, and averaged data/pilot channels and its fit by a sum of sinusoidal functions for bandwidth of 18 MHz.

the main peak. Dangerously, these secondary peaks can be potential false locking points, as explained earlier.

The CBOC autocorrelation function has a narrower main peak when compared to BOC(1,1), but it is not fully linear. The expression for the autocorrelation function can be approximated more simply, as it is done with the expression for the BOC(1,1) autocorrelation function inside the one chip width of the main autocorrelation peak [3]:

$$R_{\mathrm{BOC}(1,1)}(\tau) = 1 - \alpha|\tau|, \tag{2}$$

where α corresponds to the absolute value of the slope of the spreading sequence autocorrelation function's main peak. The value of α is equal to 3 for the BOC(1,1) signal, and it is larger than 3 for the CBOC data/pilot autocorrelation function for the narrow E-L spacing around the main correlation peak. Due to the existence of ripples in CBOC autocorrelation function, the value for the α is not constant and it depends on correlator spacing.

In the following, the first derivative of the autocorrelation function (α) is used as a parameter for the evaluation of the code tracking error and tracking threshold. α varies with the E-L correlator spacing, but for fixed E-L distances it can be considered constant. Analytical expressions for autocorrelation function should be able to provide both the properties of the CBOC modulation and the effects of front-end filtering on the autocorrelation peak. In order to compute α, the analytical expression for the E1 CBOC autocorrelation function is determined by fitting the autocorrelation curve with a sum of sinusoidal functions. By doing so, it is possible to compute α of the filtered autocorrelation function for different E-L correlator spacing. The resulting analytical expression for the CBOC autocorrelation function obtained by fitting the autocorrelation curve and α can be written as [10]:

$$\widetilde{R}_F(t) = \sum_{i=1}^{8} a_i \cdot \sin(b_i \cdot t + c_i),$$
$$\alpha = \left(\frac{d\widetilde{R}_F(x)}{dx}\right)\Bigg|_{x=-d/2}, \tag{3}$$

where t is the distance from the main correlation peak (half of E-L spacing), and a_i, b_i, and c_i, where $i = 1, \ldots, 8$, are fitted constants that are different for the data and pilot channels. Since we assume later both data and pilot channels are used for tracking, and thus the averaged CBOC(+) and CBOC(−) autocorrelation is used for fitting. In this case, the infinite bandwidth is used so that 99.99% of the CBOC signal is received. Therefore, three sets of a_i, b_i, and c_i parameters were used. As can be observed from Figure 1, the analytical fit closely follows the averaged CBOC autocorrelation function over the considered range. α for different E-L spacings and three different combinations is provided in Table 1. If the correlator spacing decreases below $d = 1/12$ chips, the slope of the autocorrelation peak decreases as well, because of the rounding effect of the front-end filtering on the autocorrelation peak. Additionally, a smaller E-L spacing reduces the linear tracking region which, at some point, can make the tracking unreliable. Therefore, $d = 1/12$ was chosen as a good tradeoff between accuracy and reliability for the considered bandwidth of 18 MHz.

The existence of secondary code on the pilot channel on top of the primary PRN code of duration 100 ms or 25 chips additionally complicates the tracking and requires techniques for wiping off the secondary code before further process the signal. The main advantages of secondary code are increased resistance to narrow-band interference due to the additional spectrum line, better cross-correlation properties (which mostly help during acquisition to avoid the near-far effects), and more robust data bit synchronization than with the histogram method, although that does not apply to Galileo E1-B/C, since the data bit and PRN code have the same duration.

The algorithm used for secondary code extraction is based on a combination of serial and parallel searches. It is well described in [11, 12]. The search for the primary code phase is performed serially within one primary code length, and the secondary code phase is searched in parallel over the entire length of the secondary code. The algorithm takes into consideration the residual frequency offset, and it is shown to reduce the acquisition time compared to other methods.

3. Two-Step Tracking Scheme and Discriminator Curves Analysis

Achieving the favorable properties of the new signals is possible by increasing the complexity of the tracking loops design. There are many techniques proposed to provide unambiguous tracking, as already explained in the introduction. Most of them are listed with the corresponding tracking architectures in [13, 14] as well as the code tracking

Table 1: Slope of the autocorrelation function for different early-late spacing for data, pilot channels, and averaged data/pilot channel for infinite bandwidth.

	Autocorrelation slope α		
Correlator spacing	Data/pilot averaged	Pilot channel	Data channel
1/20	3.98	4.50	3.39
1/12	5.35	5.96	4.63
1/10	5.51	6.06	4.82
1/5	1.49	0.09	1.89

error comparison. Most of them need complex tracking architecture or modify the correlation function. In this section, we present the analysis about our proposed two-step tracking scheme, simple and with less complex tracking architecture. Carrier and code delay tracking are analyzed separately, putting more weight on code delay tracking.

3.1. Galileo E1 CBOC Carrier Tracking (PLL). The right choice of the PLL discriminator is the first step towards obtaining accurate phase error estimation. The choice of the PLL discriminator is dependent upon the parameters of the E1 CBOC signal structure. The pilot channel alone can be used for the phase error estimation, since there is no data bit on it, and, theoretically, coherent integration can be used for as long as needed. As a consequence, a discriminator that is insensitive to phase jumps can be used. Using pure PLL tracking on the pilot channel as well as longer coherent integration improves carrier tracking sensitivity. It also enhances the carrier tracking loop resistance to receiver dynamics. Although 3 dB are lost by ignoring the data channel for the carrier phase estimation, the noise sensitivity is improved by 3 dB. Therefore, the use of an extended arctangent discriminator has been selected (atan2), providing the widest linear tracking region. The discriminator output can be analytically expressed as

$$D_{\text{atan2}} = \arctan 2\left(\frac{Q}{I}\right). \tag{4}$$

I and Q are in-phase and quadrature-phase correlator outputs for the pilot channel. The extended *arctangent* (four-quadrant) discriminator has an operational range twice as large as the traditional *arctangent* discriminator ($[-\pi, \pi]$). It has good noise resistance performance for high C/N_0 (carrier-to-noise ratio). It can also track the phase modulo 2π, without a half cycle of ambiguity. Although pilot-only tracking is noisier than data/pilot combined tracking, the chance of losing lock is smaller, and the tracking is more stable and less complex.

3.2. Galileo E1 CBOC Code Delay Tracking (DLL). Code delay tracking is more stable than carrier tracking, since it provides the user with more robust measurements and initial estimates of the receiver position through pseudoranges. The main idea behind the considered code tracking algorithm that we call the "two-step tracking technique" [1, 15] is to benefit from the narrowness of the CBOC autocorrelation peak while minimizing the risk of locking onto one of the secondary peaks of the autocorrelation function. This algorithm can also be applied to BOC(1,1) as well as CBOC(6,1,1/11) tracking and consists of two steps. The first step is a coarse tracking step that is used to ensure proper convergence towards the true lock point, since it relies on an unambiguous discriminator. This is achieved using an original unambiguous combination of all five correlators (E, L, P, VE, and VL), where VE and VL are always positioned on the secondary peaks. This ensures that the prompt correlator is indeed located on the main correlation peak.

However, once the code tracking has converged to the main peak, the VEMLP (Very Early Minus Late Power) discriminator that uses five correlators' outputs offers sub-optimal code tracking performance since it does not benefit from the narrow CBOC correlation peak. At this point, the tracking process switches to a fine tracking step, where a DP (or HRC) discriminator is used with a narrow correlator spacing. The description and parameters of the two-step tracking technique are provided in Table 2 [1]. Power in the name VEMLP comes from the fact that the discriminator function contains squared correlator outputs that relate to the power of the signal.

As shown in Table 2, the fine tracking step fully exploits the narrow autocorrelation peak by using (1) a DP (HRC) discriminator with narrow E-L spacing and (2) a reduced loop bandwidth (as enabled by the use of carrier aiding). We proposed an HRC discriminator [16] for the fine tracking step instead of a DP discriminator in order to improve multipath mitigation [1]. More specifically, a combination of traditional tracking (DP) and tracking with an HRC is used. In the case of strong multipath, an HRC correlator is used, since it shows good multipath mitigation, otherwise traditional tracking (DP) is assumed, since it has a wider stability tracking region. Details are provided in Section 5.2. Normalization of the discriminator outputs was also performed in order to provide an unbiased estimate of the small code errors. The use of a bump-jump algorithm [4] minimizes the false lock risk. It measures and compares the received power in the VE, P, and VL correlator outputs and jumps left or right if the VE or VL correlator output power is found to be consistently higher than that of the P. Note that the E-L correlator spacing should remain flexible, depending on the front-end bandwidth of the receiver. We used bandwidth of 18 MHz throughout the paper and generated the results based on this assumption. However, the parameters of the two-step tracking scheme, such as early-late spacing, width of the stability and linearity tracking regions, slope of the autocorrelation functions (as it is provided in Table 1), and tracking thresholds change depending on the available bandwidth. The scheme is designed to be flexible, for example, when decreasing the front-end bandwidth, early-late correlator spacing increases, and so forth.

In order to improve code tracking accuracy, both pilot and data channels are used for the code tracking delay estimation, making it more robust. Common data/pilot code

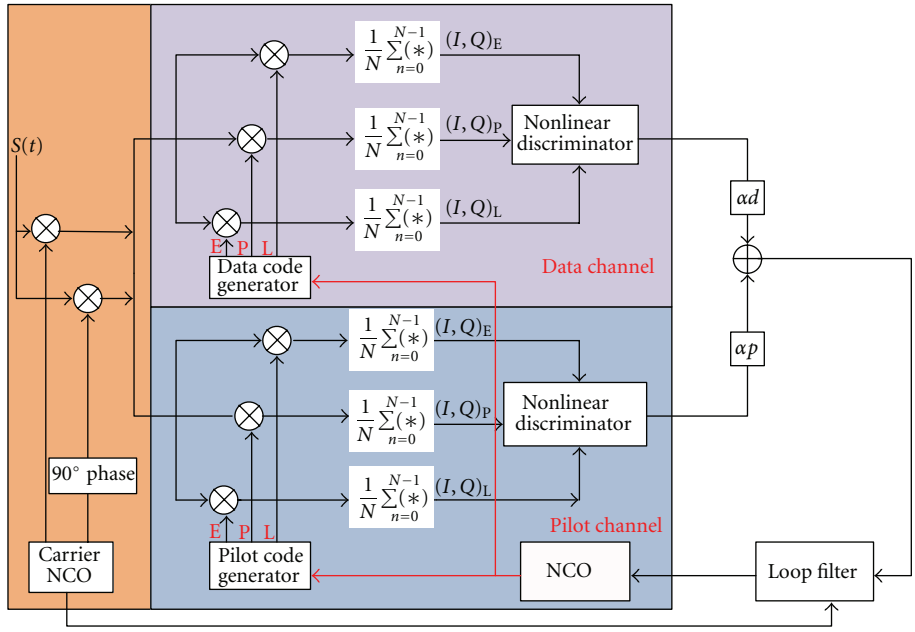


Figure 2: Noncoherent collaborative tracking architecture for the fine tracking step of two-step tracking algorithm.

Table 2: Two-step tracking scheme description for the front-end bandwidth of 18 MHz.

	Two-step tracking technique		
Parameters	Coarse tracking	Fine tracking (pilot channel)	Fine tracking (data channel)
E-L spacing	[−1/2; −1/4; 0; 1/4; 1/2]	[−1/2; −1/24; 0; 1/24; 1/2]	[−1/2; −1/24; 0; 1/24; 1/2]
Discriminator type	VEMLP	DP/HRC	DP
Loop filter BW	2 Hz	1 Hz	1 Hz
False lock detector	Embedded	Bump-jump	Bump-jump
Carrier aiding	No	Yes	Yes
Advantage	Unambiguous	Accurate multipath gain	Accurate
Disadvantage	Noise and MP increased	Ambiguous	Ambiguous

tracking is performed by linearly combining them noncoherently. The incoming signal is separately combined with data and pilot PRN codes. The outputs are then combined in a noncoherent way, as shown in Figure 2. Coherent combining outperforms noncoherent combining from an accuracy standpoint, but it also requires data bit sign recovery for integration times greater than 4 ms, which increases the loop complexity and becomes unreliable at low C/N_0. Therefore, data and pilot discriminator outputs are averaged using a noncoherent combining method and fed into a unique loop filter that updates both pilot and data NCO (Numerically Controlled Oscillator). The discriminator output can then be written as

$$D_{\text{out}} = \alpha_p D_d + \alpha_d D_p, \tag{5}$$

where α_p and α_d are weights for the pilot and data channels, respectively, applied to the discriminator. The weights have to fulfill the following requirements:

$$\begin{aligned} \alpha_p + \alpha_d &= 1, \\ \alpha_p &= \frac{\sigma_p^2}{\sigma_p^2 + \sigma_d^2}, \\ \alpha_d &= \frac{\sigma_d^2}{\sigma_p^2 + \sigma_d^2}, \end{aligned} \tag{6}$$

where σ_p and σ_d are the variances of the discriminator on the pilot and data channel, respectively. If the same modulation and code tracking loop update rate are used for the data and pilot channels, the same variance will be observed on both channels, reducing the channel combining

to a simple averaging [2, 15]. However, since the data and pilot autocorrelation functions are not rigorously identical, the pilot channel having a slightly sharper main lobe peak that could provide slightly better tracking accuracy, it is therefore possible to further refine the weighting scheme so that the tracking relies more heavily on the pilot channel. Theoretical values for the weights for the pilot and data channels are shown to be approximately 0.6 and 0.4, respectively. They were obtained empirically, by computing the standard variation of the data and pilot channels and using a setup that consists of a Spirent GSS8000 simulator [17], connected with a wideband front-end [18], and the data was postprocessed in software.

The resulting collaborative tracking architecture for the fine tracking step of the two-step tracking algorithm is shown in Figure 2, highlighting the DLL part divided in separate tracking architectures for data and pilot channels. After wiping off the carrier using multiplication of the incoming signal with a local replica that is aligned using the information from the carrier NCO, the resulting in-phase and quadrature-phase components are multiplied with five code replicas: E, P, and L as well as VE and VL (not shown in Figure 2). Ideally, if the carrier phase is aligned with the carrier phase of the incoming signal, all the energy will be in the in-phase component. Estimation of the code delay error is performed in the DLL loop, using DLL discriminators. The output is then filtered by the DLL loop filter, and updates are provided for the code delay rate for the NCO. In the next section, analysis of the three different types of discriminators used in the tracking scheme described above is provided.

3.3. DLL Discriminators. As shown in Table 2, three types of discriminators are used in the two-step tracking scheme: VEMLP and DP, for the coarse and fine tracking steps, respectively, and HRC as an option when high multipath error is present in the fine tracking step. The VEMLP discriminator is noncoherent and needs two more correlators (VE and VL) with a fixed distance between them, to mark points that can be potential sources of false locks. It has a higher computational load than DP, which is quasicoherent. After converting a signal to *IF* frequency, I and Q baseband correlation outputs for the E, L, P, VE, and VL pilot channel can be written as

$$I_X = \sqrt{\frac{\mathrm{P}}{2}}\widetilde{R}(\varepsilon_\tau + \beta)\frac{\sin\left(\pi\varepsilon_f T_i\right)}{\pi\varepsilon_f T_i}\cos\left(\varepsilon_{\phi_p}\right) + n_{IX},$$
$$Q_X = \sqrt{\frac{\mathrm{P}}{2}}\widetilde{R}(\varepsilon_\tau + \beta)\frac{\sin\left(\pi\varepsilon_f T_i\right)}{\pi\varepsilon_f T_i}\sin\left(\varepsilon_{\phi_p}\right) + n_{QX}, \tag{7}$$

where $\widetilde{R}$ is the correlation of the local spreading code with the filtered incoming spreading code, ε_f is the frequency error, ε_τ is the code group delay error, and ε_ϕ is the carrier phase delay error. The noise components n_{IX}, n_{QX} are independent Gaussian noise components. X relates the correlator type (E, P, L, VE, VL), and β represents the spacing between correlators (for P $\beta = 0$, E $\beta = \delta/2$, for L $\beta = -\delta/2$, VE $\beta = \delta$, and VL $\beta = -\delta$). δ represents the E-L correlator spacing and for the coarse tracking step ($\beta = 2\delta$). Therefore, the output for the types of the discriminators can be written analytically as

$$D_{\mathrm{VEMLP}} = \sqrt{I_{\mathrm{VE}}^2 + Q_{\mathrm{VE}}^2 + I_{\mathrm{E}}^2 + Q_{\mathrm{E}}^2} - \sqrt{I_{\mathrm{VL}}^2 + Q_{\mathrm{VL}}^2 + I_{\mathrm{L}}^2 + Q_{\mathrm{L}}^2}, \tag{8}$$

$$D_{\mathrm{DP}} = (I_{\mathrm{E}} - I_{\mathrm{L}})I_{\mathrm{P}} + (Q_{\mathrm{E}} - Q_{\mathrm{L}})Q_{\mathrm{P}}, \tag{9}$$

$$D_{\mathrm{HRC}} = (I_{\mathrm{E}} - I_{\mathrm{L}}) + \frac{(I_{\mathrm{VE}} - I_{\mathrm{VL}})}{2}. \tag{10}$$

Analytical expressions for the VEMLP, DP, and HRC discriminators functions can be written using filtered autocorrelation function ($\widetilde{R}$) with the approximation that in tracking stage $\sin(\pi\varepsilon_f T_i)/\pi\varepsilon_f T_i \simeq 1$ and $\cos(\varepsilon_{\phi_d}) \simeq 1, \sin(\varepsilon_{\phi_d}) \simeq 0$ as

$$D_{\mathrm{VEMLP}} = \sqrt{\frac{\mathrm{P}}{2}}\left(\sqrt{\widetilde{R}^2\left(\tau + \frac{\delta}{2}\right) + \widetilde{R}^2(\tau + \delta)} - \sqrt{\widetilde{R}^2\left(\tau - \frac{\delta}{2}\right) + \widetilde{R}^2(\tau - \delta)}\right), \tag{11}$$

$$D_{\mathrm{DP}} = \sqrt{\frac{\mathrm{P}}{2}}\left(\widetilde{R}\left(\tau + \frac{\delta}{2}\right) - \widetilde{R}\left(\tau - \frac{\delta}{2}\right)\right)\widetilde{R}(\tau), \tag{12}$$

$$D_{\mathrm{HRC}} = \sqrt{\frac{\mathrm{P}}{2}}\left(\widetilde{R}\left(\tau + \frac{\delta}{2}\right) - \widetilde{R}\left(\tau - \frac{\delta}{2}\right) - \frac{\widetilde{R}(\tau + \delta) - \widetilde{R}(\tau - \delta)}{2}\right). \tag{13}$$

The performance of the tracking loops in the presence of multipath and code tracking errors can be assessed by studying the discriminators curves. These curves present the discriminators' outputs as a function of code delays (inputs). Tracking parameters of interest that can be derived from the curves are the linear tracking region, defined as the region in which the discriminator responds without any bias, and the stability region (pull-in region), defined as the region where the discriminator reacts in the right direction [3].

A wider stability tracking region means more robust tracking. A wider linear tracking region corresponds to correct and unbiased discriminator response. As already shown in [3], the discriminator output does not directly estimate the input error. In order to obtain unbiased code delay error, the discriminator output has to be normalized by its gain, therefore achieving perfect normalization. Normalization removes amplitude sensitivity, improving performance under rapidly changing SNR (Signal-to-Noise Ratio) conditions and providing unbiased code delay error estimation. As normalization depends on the power of the signal, discriminator outputs should be first normalized by the estimated power of the signal. Therefore, using this

approach, the normalization factors for three discriminator types can be written as

$$N_{\mathrm{VEMLP}} = \sqrt{I_{\mathrm{VE}}^2 + Q_{\mathrm{VE}}^2 + I_{\mathrm{E}}^2 + Q_{\mathrm{E}}^2} + \sqrt{I_{\mathrm{VL}}^2 + Q_{\mathrm{VL}}^2 + I_{\mathrm{L}}^2 + Q_{\mathrm{L}}^2}, \tag{14}$$

$$N_{\mathrm{DP}} = (I_{\mathrm{E}} + I_{\mathrm{L}})I_{\mathrm{P}} + (Q_{\mathrm{E}} + Q_{\mathrm{L}})Q_{\mathrm{P}}, \tag{15}$$

$$N_{\mathrm{HRC}} = (I_{\mathrm{E}} + I_{\mathrm{L}}) + \frac{(I_{\mathrm{VE}} + I_{\mathrm{VL}})}{2}. \tag{16}$$

Expressions for the discriminator outputs can be simplified using the assumptions that the estimated code delay error is smaller than half of the correlator spacing, and the E and L correlator values correspond to the points on the autocorrelation function that are on the slope forming its main peak. Normalization provides unbiased estimation of the code delay error within $\pm\delta/2$ chips. Using the equations for the correlator outputs provided above and using the assumption that the autocorrelation function can be approximated by small linear parts with different slopes, the discriminator outputs can be written as

$$N_{\mathrm{VEMLP}} = \sqrt{2\mathrm{P}}\sqrt{2 - 2\alpha\varepsilon_\tau(\alpha\varepsilon_\tau - 2\delta) - \alpha\delta\left(\delta - 1 - \alpha - \alpha\frac{\delta}{4}\right)}, \tag{17}$$

$$N_{\mathrm{DP}} = \sqrt{\frac{\mathrm{P}}{2}}(2 - \alpha\delta)(1 - \alpha|\varepsilon_\tau|), \tag{18}$$

$$N_{\mathrm{HRC}} = \sqrt{\frac{\mathrm{P}}{2}}(2 - \alpha\delta)(1 - \alpha|\varepsilon_\tau|) + \sqrt{\frac{\mathrm{P}}{2}}(1 - \alpha\delta). \tag{19}$$

Therefore, using the normalization expressions provided above, as well as the normalization factors that relate to discriminator gain, the expressions for the normalized VEMLP, HRC, and DP discriminators can be written as

$$D_{\mathrm{VEMLP}_n} = \frac{2 - 3\alpha\delta + (5/4)\alpha^2\delta^2}{3\alpha\delta - 4}\frac{D_{\mathrm{VEMLP}}}{N_{\mathrm{VEMLP}}}, \tag{20}$$

$$D_{\mathrm{DP}_n} = \frac{(2 - \alpha\delta)}{2\alpha}\frac{D_{\mathrm{DP}}}{N_{\mathrm{DP}}}, \tag{21}$$

$$D_{\mathrm{HRC}_n} = \frac{2\alpha\delta - 3}{3\alpha}\frac{D_{\mathrm{HRC}}}{N_{\mathrm{HRC}}}. \tag{22}$$

These expressions are obtained from (18), (19), and (20) using the analytical expression for the autocorrelation function and obtaining the slope of the autocorrelation peak at $\alpha = -\delta/2$. Normalization factors were obtained analytically and tested empirically. They include normalization by the discriminator gain obtained by derivation of the discriminator output. Whereas normalization is straightforward for DP discriminator, for VEMLP and HRC discriminators, the following approximations should be used: $\alpha^2\varepsilon_\tau^2 \sim 0$ and $\alpha\varepsilon_\tau \sim 0$. These are valid for small code tracking errors, while the tracking is maintained.

These factors are constant for BOC(1,1) and BPSK signals, but for CBOC they vary depending on the E-L correlator distance. In both tracking steps, the correlators are located on the linear part of the discriminator curve, and the factors can be considered constant for fixed E-L spacing, as shown in Table 3. This table contains the discriminator normalization factors for the combined data/pilot tracking and the slope of the autocorrelation curve as provided in Table 1.

Table 3: Discriminator normalization factor for the two-step tracking scheme (E-L = 1/12, VE-VL = 1/2).

Discriminator normalization factors		
Discriminator type	Pilot channel	Data channel
DP	0.32	0.28
VEMLP	0.6	0.55
HRC	0.25	0.35

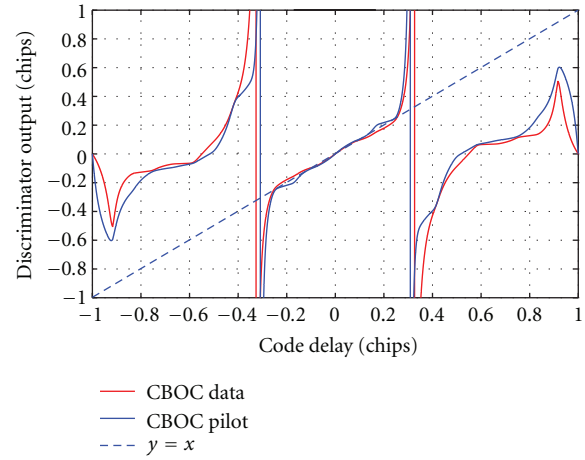


Figure 3: Normalized DP discriminator output for CBOC data and pilot channels and an E-L spacing of 1/12 chips.

The discriminator outputs for DP and VEMLP discriminators are shown in Figures 3 and 4. The most important point that can be observed from these figures is that, by applying normalization, the discriminator curve's linear tracking region increases. Also, the discriminator curves for the pilot channel alone and the VEMLP discriminator (Figure 4) contain additional ripples. These ripples are due to high-frequency components introduced by BOC(6,1) modulation. As expected, the discriminators provide an unbiased response for input code errors below roughly 0.25 chips. For input code errors greater than that, the discriminators' output tends to flatten. This, in turn, should provide a smooth convergence phase. Since VEMLP offers only limited noise and multipath mitigation capabilities, it is therefore used directly after acquisition to ensure convergence of the code tracking loop to the main peak of the autocorrelation function. Upon convergence of both the code and carrier tracking loops, the tracking software switches to the fine tracking step.

As was the case for the VEMLP discriminator, the overall behavior of the DP discriminator is invariant across the two

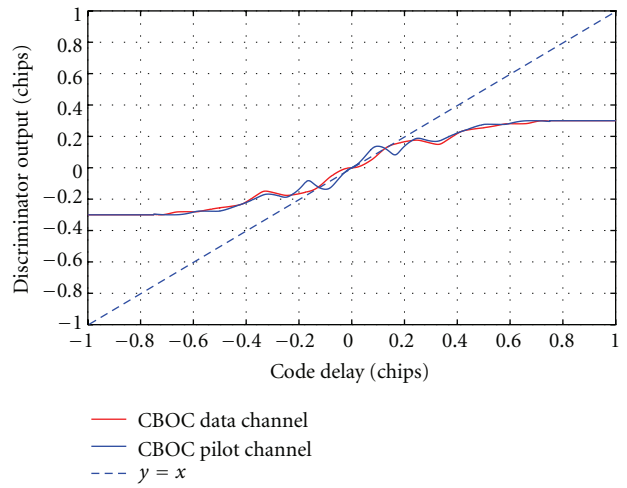


Figure 4: Normalized VEMLP discriminator output for CBOC data and pilot channels and an E-L spacing of 1/2 chips.

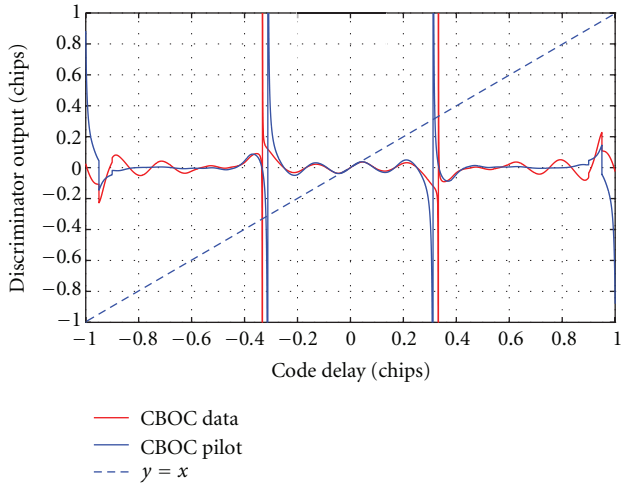


Figure 5: Normalized HRC discriminator output for CBOC data and pilot channels and an E-L spacing of 1/12 chips.

MBOC modulations apart from the additional ripples that can be observed on the two CBOC discriminators. Figure 3 also highlights the false-lock point issue inherent to all BOC and CBOC modulations (e.g., [3]). Indeed, it is obvious from this figure that the zero-crossing observed at approximately 0.5 chips would provide a stable lock point which would lead to a ranging bias of 150 m. The HRC normalized discriminator output is shown in Figure 5. It can be seen that it has a much narrower linear tracking region in comparison to DP and VEMLP.

Again, normalization brings improvement by increasing the linearity region of the discriminator, making tracking more robust. However, the stability region remains similar to the unnormalized case. Normalization thus reveals the true code error present at the output of the discriminator. To conclude this section, the main advantage of using normalization is that the discriminator will perfectly react to a wide range of code delays. Both DP and VEMLP discriminators have similar widths of the linearity and stability regions before performing normalization. After normalization, the DP discriminator shows better stability and linearity (approximately 5% and more). Also, the normalized DP discriminator has vertical asymptotes. This means that the loop will overreact for large errors, and this can cause sudden tracking jumps that might lead to a false lock or degraded tracking.

So far, tracking loop and code discriminators have been described assuming no external disturbances. Since the DLL tracking loop can be affected by different sources of disturbances, such as thermal noise, multipath and signal dynamics, our next step is to investigate the influence of these error sources on the overall behavior of the tracking loop.

4. Code Tracking Error and Tracking Threshold

Code tracking error besides multipath and dynamic stress error is the dominant source of range errors in a GNSS receiver's DLL [19, 20]. Using carrier aiding, the dynamic stress error is negligible; therefore, only the error induced by thermal noise will be considered. The code tracking threshold can be derived using an analytical expression for the thermal noise-induced CBOC code tracking error variance. For the DP discriminator used in the fine tracking step, it is given in [3, 21], whereas, for the VEMLP discriminator, it is derived below. Analytical expressions for both discriminators can be written as

$$\sigma^2_{(\mathrm{DP,VEMLP,HRC})} = \frac{2B_{\mathrm{L}}(1-0.5B_{\mathrm{L}}T_I)S_N(0)}{K_{(\mathrm{DP,VEMLP,HRC})}}. \quad (23)$$

$S_N(0)$ is the noise power spectral density, and $K_{(\mathrm{DP,VEMLP,HRC})}$ represents the discriminator loop gain given by

$$K_{(\mathrm{DP,VEMLP,HRC})} = \left.\frac{dD_{(\mathrm{DP,VEMLP,HRC})}}{d\varepsilon_\tau}\right|_{\varepsilon_\tau=0}. \quad (24)$$

Combining this equation with the expressions for the correlator outputs provided in Section 3.3 and arranging them using the following equality for the filtered version of the autocorrelation function, we derived theoretical expressions for the code tracking errors for both DP and VEMLP discriminators for infinite front-end bandwidth:

$$\sigma^2_{\mathrm{VEMLP}} = \frac{B_{\mathrm{L}}(1-0.5B_{\mathrm{L}}T_I)\left(\left(\widetilde{R}_F(0)-\widetilde{R}_F(\delta)\right)\widetilde{R}_F(\delta/2)\left(\widetilde{R}_F(\delta/2)+\widetilde{R}_F(\delta)\right)+\left(\widetilde{R}_F(0)-\widetilde{R}_F(2\delta)\right)\widetilde{R}^2_F(\delta)\right)}{2(\mathrm{P}/N_0)\left(\widetilde{R}_F(\delta)\left(d\widetilde{R}_F(x)/dx\right)\Big|_{x=\delta}+\widetilde{R}_F(\delta/2)\left(d\widetilde{R}_F(x)/dx\right)\Big|_{x=\delta/2}\right)^2}\cdot\theta,$$

$$\sigma^2_{\mathrm{DP}} = \frac{B_{\mathrm{L}}(1 - 0.5B_{\mathrm{L}}T_I)\left(\widetilde{R}_F(0) - \widetilde{R}_F(\delta)\right)}{2(C/N_0)\left\{d\widetilde{R}_F(x)/dx\right\}^2\Big|_{x=d/2}} \cdot \chi. \tag{25}$$

Expressions $\alpha_1 = (d\widetilde{R}_F(x)/dx)|_{x=\delta}$ and $\alpha_2 = (d\widetilde{R}_F(x)/dx)|_{x=\delta/2}$ represent the slopes of the autocorrelation function at E-L spacings δ and $\delta/2$. Expressions for θ and χ can be approximated as

$$\begin{aligned}\theta\left(\frac{\mathrm{P}}{N_0}, T_I\right) &= 1 + \frac{3\left(\widetilde{R}_F^2(0) - \widetilde{R}_F^2(\delta)\right) + \widetilde{R}_F^2(0) - \widetilde{R}_F^2(2\delta)}{(2CT_I/N_0)\left(\left(\widetilde{R}_F(0) - \widetilde{R}_F(\delta)\right)\widetilde{R}_F(\delta/2)\left(\widetilde{R}_F(\delta/2) + \widetilde{R}_F(\delta)\right) + \left(\widetilde{R}_F(0) - \widetilde{R}_F(2\delta)\right)\widetilde{R}_F^2(\delta)\right)},\\ \chi\left(\frac{\mathrm{P}}{N_0}, T_I\right) &= 1 + \frac{1}{(CT_I/N_0)\widetilde{R}_F(0)}.\end{aligned} \tag{26}$$

B_{L} is DLL loop bandwidth, T_I is coherent integration time, δ is the early-late spacing, C is the incoming power of the carrier, N_0 is the thermal noise, and $\widetilde{R}_F$ is the filtered correlation function of the incoming signal and $\theta, \chi = f(C/N_0, T_I)$ are factors that are a function of the integration time and carrier-to-noise ratio. The expression for the VEMLP discriminator can be simplified, using the following equality:

$$\widetilde{R}_F(0) - \widetilde{R}_F(\delta) \simeq \widetilde{R}_F\left(\frac{\delta}{2}\right) - \widetilde{R}_F\left(\frac{3\delta}{2}\right). \tag{27}$$

This simplification is valid only for wide E-L correlator spacings (E-L = 1/2) that are used in the coarse tracking step. Using the values for the slope of the autocorrelation function α from Table 1 for different correlator spacings, the code tracking error for the CBOC fine and coarse tracking steps can be computed. From our analysis, we observed that VEMLP discriminator provides very high code tracking noise error (33 m for C/N_0 of 25 dB-Hz) for the basic configuration (E-L = 1/2). High code tracking noise is the price one pays in the coarse tracking step. Decreasing the inner correlator spacing (E-L) while keeping the outer spacing (VE-VL) to one chip decreases the code tracking noise as well.

When increasing the correlator spacing, the DP discriminator code tracking error increases, but less dramatically pace than is observed for the VEMLP discriminator. The DP code tracking noise is very similar for different spacings since the E-L spacing is small (around 1.7 m for C/N_0 of 25 dB-Hz). The code tracking error for an E-L distance of 1/20 is higher than for E-L distance of 1/12. This happens due to front-end filtering, which rounds the autocorrelation peak, increasing the noise. DP outperforms VEMLP for large correlator spacings. Since the first stage of the two-step tracking scheme is short, the large code tracking noise does not greatly influence the overall tracking performance.

As was previously shown in [2], the code tracking error is approximately $\sqrt{2}$ lower for combined pilot and data channels tracking than for single channel tracking. Therefore, the tracking threshold decreases using combined tracking schemes. An analysis of the tracking threshold is provided below, starting from the code tracking noise derivation. Following the analysis provided in [19], the resulting analytical expression for the tracking threshold is obtained, and it is equal to

$$\left(\frac{C}{N_0}\right)_{\mathrm{ThDP}} = \frac{18 \cdot W_D\left(1 + \sqrt{1 + \delta^2/9W_DT_IR_{\mathrm{CBOC}}}\right)}{\delta^2}. \tag{28}$$

In the same way, the resulting tracking threshold for the VEMLP discriminator can be written as

$$\left(\frac{C}{N_0}\right)_{\mathrm{ThVEMLP}} = \frac{18 \cdot W_T\left(1 + \sqrt{1 + \delta^2/9W_TT_IY}\right)}{\delta^2}. \tag{29}$$

Parameters W_T and Y depend on the slope of the autocorrelation function, integration time, and DLL parameters and can be written as

$$\begin{aligned}W_T &= \frac{B_{\mathrm{L}}(1 - 0.5B_{\mathrm{L}}T_I)\left(\left(\widetilde{R}_F(0) - \widetilde{R}_F(\delta)\right)\widetilde{R}_F(\delta/2)\left(\widetilde{R}_F(\delta/2) + \widetilde{R}_F(\delta)\right) + \left(\widetilde{R}_F(0) - \widetilde{R}_F(2\delta)\right)\widetilde{R}_F^2(\delta)\right)}{2\left(\widetilde{R}_F(\delta)\left(d\widetilde{R}_F(x)/dx\right)\Big|_{x=\delta} + \widetilde{R}_F(\delta/2)\left(d\widetilde{R}_F(x)/dx\right)\Big|_{x=\delta/2}\right)^2},\\ Y &= \frac{2\left(\left(\widetilde{R}_F(0) - \widetilde{R}_F(\delta)\right)\widetilde{R}_F(\delta/2)\left(\widetilde{R}_F(\delta/2) + \widetilde{R}_F(\delta)\right) + \left(\widetilde{R}_F(0) - \widetilde{R}_F(2\delta)\right)\widetilde{R}_F^2(\delta)\right)}{3\left(\widetilde{R}_F^2(0) - \widetilde{R}_F^2(\delta)\right) + \widetilde{R}_F^2(0) - \widetilde{R}_F^2(2\delta)}.\end{aligned} \tag{30}$$

Following our analysis that cannot be fully presented here, the tracking threshold decreases as the integration time increases and DLL bandwidth decreases. For very long integration durations, extremely low C/N_0 values can be tracked, such as 10 dB-Hz. What was observed is that the results for tracking threshold for two types of discriminators DP and VEMLP are very similar. For VEMLP, as the inner correlator spacing decreases, the tracking threshold decreases as well whereas, for coarse tracking step, does not drop below 20 dB-Hz. For DP, the trend is the same except the fact that for E-L spacing of 1/20 the threshold is increased, due to the rounding of autocorrelation peak.

5. Multipath Mitigation Analysis

Multipath represents a phenomenon that disturbs tracking causing phase offsets in code and carrier and should thus be properly mitigated. For automotive applications in urban environments, the multipath conditions will change continuously, making multipath one of the most disturbing problems of GNSS-based navigation systems in urban environments. As already analyzed and shown in [22], carrier multipath is not critical, causing maximum absolute multipath error of only 0.015 m for E1 CBOC. Code multipath represents a much more important issue and will be discussed here.

5.1. Code Multipath Analysis. Multipath is the main factor that affects the pseudoranges, and it does so through parameters such as the number of multipath signals, geometric path delay, phase, and relative power. Pseudorange measurement is performed by the code tracking loop, which aligns the locally generated PRN code with the PRN code of the incoming signal. The impact of multipath on code tracking accuracy is often represented as an error envelope representing the maximum error resulting from a single multipath with a certain phase, delay, and amplitude. This is called multipath error envelope (MEE). MEEs are computed for each discriminator type in the two-step tracking scheme, as already shown in [1], and for both steps and analyzed the optimal solution in order to mitigate or severely minimize the multipath error.

The first step, a coarse tracking step, uses a noncoherent VEMLP discriminator with wide correlator spacings. The resulting MMEs for the CBOC and BOC(1,1) signals are high MMEs, going up to 25 m as shown in [1]. Large multipath error is the price paid for reliable tracking in the coarse tracking step. As already explained in [1], the E-L correlator spacing should be as low as possible since the multipath error rapidly increases for spacings beyond 1/20 chips. For an E-L spacing of 1/12 chips, the MEE is close to that obtained for 1/20 chips but the linear region is wider which implies more robust tracking in the fine tracking step. For wider E-L spacings between the performances of the CBOC and BOC(1,1) are almost equal. As it it shown in [1], it is clear that CBOC offers a better resistance to long delay multipath than BOC(1,1) tracking. However, for short delays (<8 m), it leads to the same multipath envelope. An additional way to mitigate multipath is to lower the loop bandwidth as much as possible (0.5 Hz). This way, the DLL will not be able to track the multipath-induced error, resulting in more accurate tracking.

5.2. Proposed Fine Tracking Step Implementation. The use of an HRC discriminator is proposed in the fine tracking step of two-step tracking algorithm. A potential issue with HRC is the limited stability tracking region, as shown in Section 3.3. This can be minimized through the implementation of carrier aiding which reduces the dynamics experienced by the discriminator. Since the HRC degrades the postcorrelation SNR and since the coherent integration time on the data channel is limited to 4 ms, it appears that implementing the HRC on the data channel would not provide a very robust solution. However, in the absence of unknown bit transition and with the help of carrier aiding, the coherent integration time can easily be extended to a couple hundred milliseconds on the pilot channel [3]. Consequently, implementation of the HRC on the pilot channel is the desired configuration.

Given the previous discussion, we proposed a CBOC tracking algorithm that combines a two-step tracking technique with the HRC on the pilot channel to lower the multipath error. Different correlator spacing configurations are used on the data and pilot channels. The data channel uses the configuration defined for the fine code tracking, that is, $[-1/2; -1/24; 0; 1/24; 1/2]$, and the pilot channel uses the HRC configuration defined as $[-1/12; -1/24; 0; 1/24; 1/12]$, where the spacings follow the pattern $[-2\delta_{\text{HRC}}; -\delta_{\text{HRC}}; 0; \delta_{\text{HRC}}; 2\delta_{\text{HRC}}]$ and δ_{HRC} is the E-L correlator spacing. The scheme then consists of using the data channel to perform false lock detection and, in case no false lock is detected, using the pilot channel to close the tracking loop, which may use HRC discriminator depending on the following two scenarios.

(1) *High Multipath Case.* Tracking using the HRC discriminator is desired. In this case, the use of the DP discriminator is not recommended as it would feed the combined discriminator with its multipath-induced error.

(2) *Reliably Critical Case.* Tracking using the DP discriminator is desired. In this case, the use of the HRC should be avoided so as to avoid integrating the potentially unreliable HRC discriminator output into the combined discriminator output.

The pilot channel should therefore implement an "either/or" discriminator combination that would alternately rely on the DP or the HRC discriminator. It was shown in [1] that the MEE does not exceed 0.01 chips for the considered E-L spacing. The discriminator output should not be greater than this if the tracking is correctly achieved. Therefore, to decide which discriminator to use, a test based on the DP correlator output is run. If the code tracking error is small (i.e., the DP discriminator output is in its linear tracking range), then the HRC is used. Otherwise, the tracking relies on the DP because of its wider stability range (0.3 chips in

one direction). To summarize, the pilot discriminator output can be written as

$$D_{\text{pilot}} = D_{\text{DP}}, \quad \text{if } D_{\text{DP}} \geqslant \delta_{\text{HRC}},$$
$$D_{\text{pilot}} = D_{\text{HRC}}, \quad \text{if } D_{\text{DP}} \leqslant \delta_{\text{HRC}}. \tag{31}$$

The choice of threshold δ_{HRC} could be made to depend on C/N_0. This approach would bring robustness and will be investigated in the future. When higher C/N_0 is available, strong multipath mitigation could be achieved, while for lower C/N_0 the tracking would remain robust.

6. Results and Discussion

After providing the overview of the theoretical performance of the two-step tracking scheme, we present in this section further the simulation results in order analyze if they are consistent with the theoretically derived results. The data collection setup consists of a Spirent GSS8000 simulator [17] to emulate the Galileo E1 OS signal, a Fraunhofer triple-band front-end [18] with 18 MHz of bandwidth to filter, downconvert, and digitize the signal of interest (i.e., E1 CBOC signal), and a postprocessing architecture that contains a software receiver implemented in MATLAB, based on a modified version of the Kai Borre GPS L1 defined software radio [20].

6.1. Simulation-Based Tracking Results. To test the proposed tracking scheme, we created different scenarios. Using the Spirent simulator's build-in multipath simulation models, we tested two types of multipath models: first type of simple multipath models, that adds a multipath ray with 0.5 chips delay and 6 dB of attenuation to the direct signal, and in the second case, a multipath with a fixed delay of 100 m is added, since this represents the point at which multipath error is the highest. Third model adds three rays that are to the direct signal with delays of 0.5, 0.3, and 0.1 chips with the same attenuation of 6 dB. Fourth group of multipath models tested was based on Land Mobile Multipath (LMM) model, statistical model that emulates direct paths and reflected paths as Rician or Rayleigh, depending on the different type of environment: Urban, Suburban, and Urban Canyon scenarios were tested. Also, two different C/N_0 configurations were investigated: low (24 dB-Hz) and high (45 dB-Hz), as well as static and mobile receiver.

The outputs from the I correlator of the pilot channel for simple multipath model that assumes one reflected path with the delay of 100 m are shown in Figure 6 for all five correlators. The most important point to observe is that all outputs stabilize after approximately 300 code periods from the beginning of tracking. After this point, the tracking is stable. The transition from coarse to fine tracking happens around 1500 code periods, when it is assured that the main autocorrelation peak is being tracked. We could observe that late correlator has higher power at the beginning. The condition for transition from coarse to fine tracking step is related to the PLI (Phase Lock Indicator). If the PLI is less than 0.9 (0.7 in case of low C/N_0), the coarse tracking step

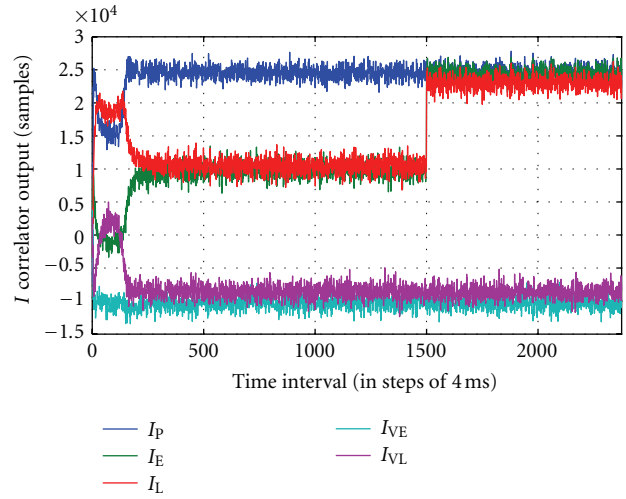


Figure 6: I correlation output for two-step tracking scheme with multipath model with fixed offset delay of 100 m.

proceeds. The transition from coarse to fine tracking step happens only when this value is exceeded. In the Figure 6, the threshold is achieved earlier, but the transition happens later in order to provide a better overview of the scheme.

The DLL outputs for the six different scenarios described above are shown in Figure 7. Note the different convergence times for some cases. Due to the deformation of the autocorrelation main peak, it takes more time for the DLL to converge when multipath is present, than when no multipath is present. Overall, approximately 500 periods are needed (or 2000 ms) to lock to the right correlation peak and then switch to the fine tracking step. The PLI indicators of the two multipath cases analyzed have multiple peaks lower than 0.5 until locked to the right correlation peak.

Once it is assured that the right correlation peak is being tracked, the fine tracking step turns on. Table 4 compares the fine and coarse tracking steps in terms of DLL output's standard deviation. It can be seen that the standard deviation drops by more than a factor of five when switching from coarse to fine tracking for all cases under investigation. It is also important to mention that the tracker succeeds to follow the right autocorrelation peak in all six cases. Peaks in the DLL output are present due to transition from coarse to fine step, as well as due to transition from HRC to DP in the fine tracking step. The PLL discriminator output is shown in Figure 8. It can be noted that variation of PLL error is similar for all cases considered, because carrier tracking is performed independently on the pilot channel only, and the frequency lock is well maintained. Only the secondary code was wiped off according to [12].

It has to be mentioned that in the fine tracking step a bump-jump algorithm was implemented, as previously proposed in [4]. Amplitude comparison was accomplished by a simple up/down counter mechanism. The absolute values of the VE, VL, and P correlators were compared and, if either VE or VL sample is the largest, the appropriate counter was increased and the other decreased. When the set

Table 4: Standard deviation of the DLL outputs for the two-step tracking scheme using six different scenarios.

	DLL standard deviation	
Scenario	Coarse tracking step	Fine tracking step
Low C/N_0 simple multipath model	0.0585	0.0124
Mobile receiver no multipath	0.0192	0.0035
One multipath ray of 0.5 chips delay	0.0169	0.0045
One multipath ray of 100 m delay	0.0180	0.0037
Land multipath model Urban Canyon	0.0154	0.0033
High C/N_0 no multipath	0.0134	0.0022

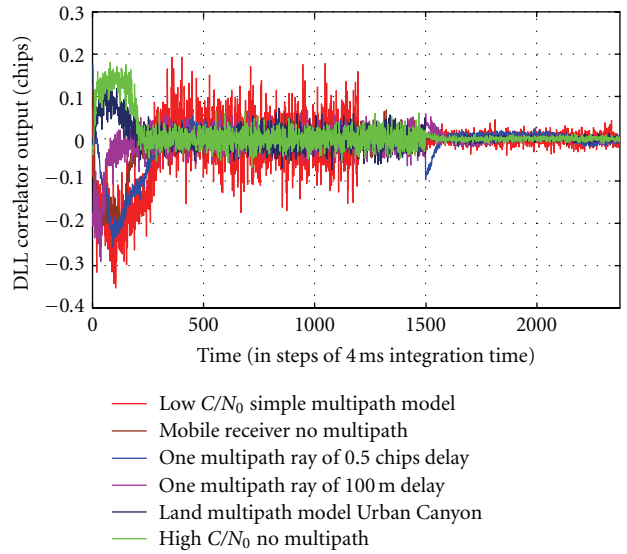


Figure 7: DLL loop output for the two-step tracking algorithm and six investigated scenarios.

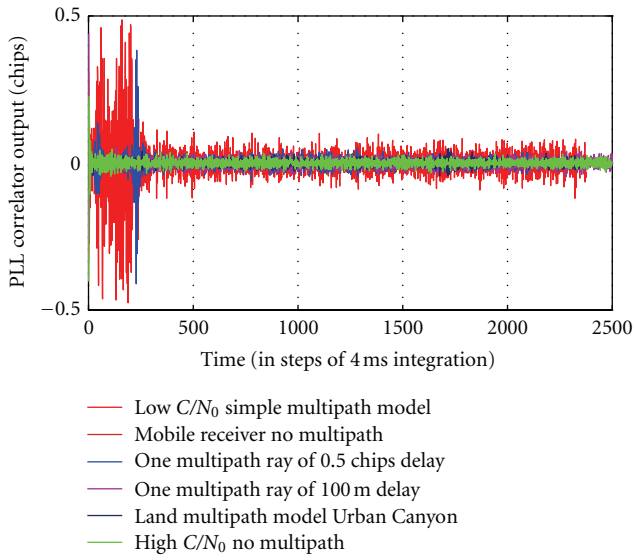


Figure 8: PLL loop output for the two-step tracking algorithm and different multipath scenarios.

threshold was exceeded, the jump to the new peak occurred. Here, we set the threshold to 50, which meant that if the VE or VL was larger than the P for 50 consecutive periods (or 200 ms), the tracking jumped to a new peak.

6.2. Galileo Real Data Tracking Results. On the 12th of December 2011, two Galileo in-orbit validation (IOV) satellites that were launched on 16th of October 2011, PFM and PM-FM2, started transmitting Galileo Open Service signals with pseudorandom code numbers SV11 and SV12 [9]. From the end of December 2011, the signals are available when the satellites are in the direct visibility domain of the receiver. Using the setup described above, we were able to receive signals from both satellites and successfully track them. We tested the two-step tracking scheme using real signals, and we show the first results here.

The setup differs from the simulation setup described above that differs only in the antenna. Instead of using the Spirent simulator output connected to the front-end receiver, we used a fixed, nondirectional rooftop wideband Antcomm antenna, connected via a two-stage amplifier to a front-end. The setup was located at our premises, in IMT Neuchatel, at latitude: 46.519617 degrees and longitude: 6.63221 degrees.

Digital *IF* (IF = 12.82 MHz) samples were stored in the memory and subsequently postprocessed, using the software receiver described earlier, on the evening of March 21st, 2012 when the satellites were in good view of the antenna. During recording, the elevation of the Galileo-PMF satellite (SV11) was 45.5 degrees and the elevation of the Galileo-FM2 satellite (SV12) 65.2 degrees.

The acquisition peak acquired in 3D space for satellite SV12 is shown in Figure 9. We can clearly observe the Galileo signal as the correlation peak rising above the noise level. After acquiring the signal, secondary code wipe-off was performed, and the values for the Doppler offset and code offset of the secondary code were passed to the tracking stage.

Correlator outputs for all five correlators are shown in Figure 10. The outputs are stable, and a clear transition from coarse to fine tracking is observed. The DLL discriminator output for both satellites is shown in Figure 11. The output is noisy in the coarse tracking step, but in the fine tracking step it is stable with a much smaller standard deviation and tracking is furthermore stable.

Once the receiver is in tracking mode, C/N_0 at the receiving antenna can be estimated. Approximately 20 code periods (around 100 ms) were used for the estimation of

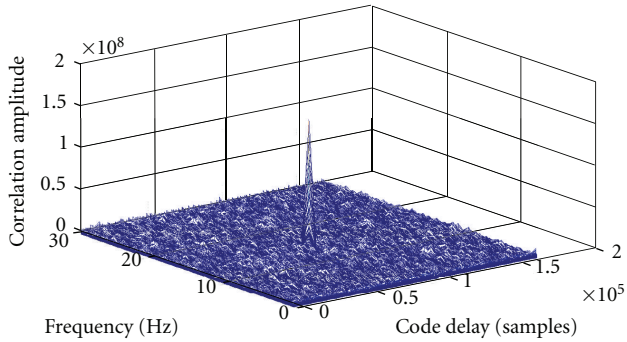


Figure 9: 3D acquisition peak acquired from the Galileo PFM satellite.

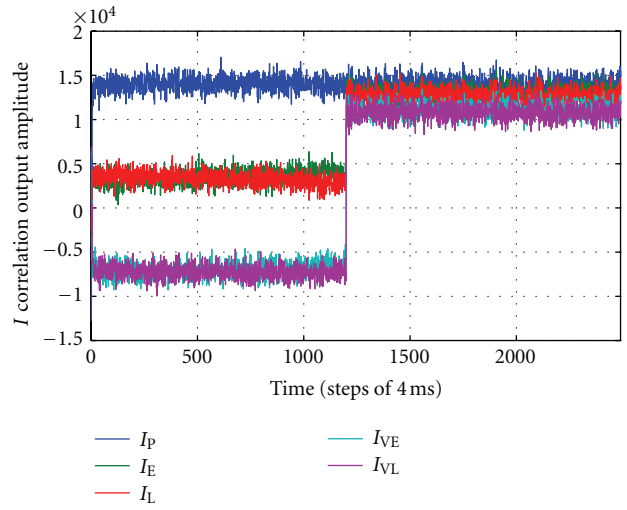


Figure 10: I correlator outputs for the two-step tracking algorithm using real data from Galileo FM2 and PFM satellites.

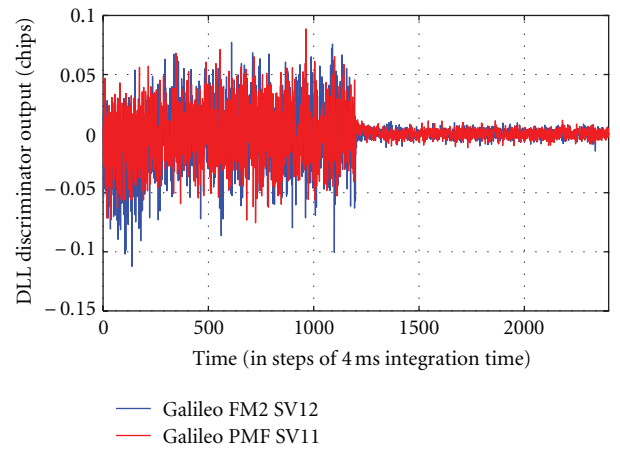


Figure 11: DLL loop output for the two-step tracking algorithm using real data from Galileo FM2 and PFM satellites.

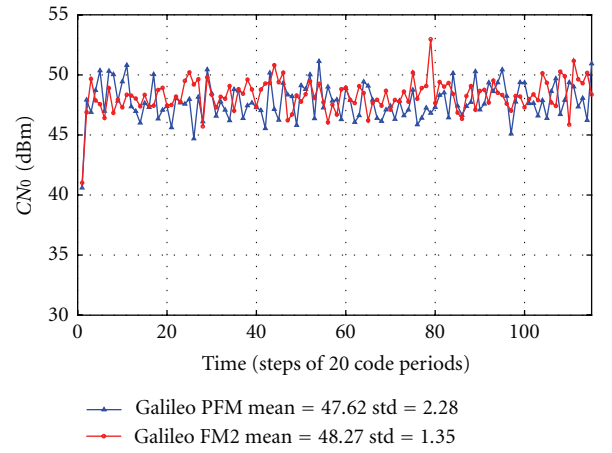


Figure 12: Estimated C/N_0 for the two-step tracking algorithm using real data from Galileo FM2 and PFM satellites.

the C/N_0. The estimated ratio is shown in Figure 12 for both satellites and is very similar C/N_0 profiles for both satellites. Note that the displayed C/N_0 relates to both the data and pilot channels. The single channel C/N_0 is around 3 dB lower. Measured C/N_0 is relatively high due to the clear view of the satellites and good position of the satellites.

7. Conclusions

This paper provided both theoretical and experimental study of the previously proposed two-step tracking algorithm for Galileo E1 CBOC tracking. The proposed algorithm has two independent tracking steps that make the algorithm flexible enough to be easily adaptable to changing signal environments. It was shown that this approach provides robust, accurate, and reliable tracking since the coarse step ensures convergence to the main autocorrelation peak while the fine step minimizes noise and multipath errors.

The analytical expression for the CBOC autocorrelation function was analyzed, as it is more complex than the BOC(1,1) autocorrelation function, and cannot be considered to be fully linear. Depending on the E-L correlator spacing, the slope of the autocorrelation function varies in the range of 1–6, giving different code tracking noise behavior. Optimal correlator spacings were discussed (VE-VL = 1/2, E-L = 1/12), in order to minimize tracking errors. Still, scheme remains flexible, and, depending on the available front-end bandwidth, tracking parameters may vary.

The analytical expression for VEMLP code tracking error is derived, and this is one of the main contributions of this paper, since it was not provided before according to our best knowledge. Using this expression, it was shown that, although the code tracking error in the first step is very high (~35 m), stable lock is provided in the first tracking step, and then it decreases exponentially (down to 1.8 m) in the fine tracking step.

We also showed experimentally using theoretical derivations that the two-step tracking algorithm has good tracking sensitivity in both tracking steps and can be used to track CBOC signals with low C/N_0 ratio without losing lock. The tracking stability was analyzed as well in form

of discriminator curves analysis. It was shown that the proposed scheme does not lose lock easily since the risk of biased tracking is circumvented. Since the two-step tracking algorithm assumes the usage of three discriminator types (VEMLP, DP, and HRC), each of them were analyzed separately and the normalization parameters optimized in order to improve the two-step tracking algorithm.

The main source of errors, including multipath and thermal noise, were evaluated, and optimal algorithm parameters (shown in Table 2) were obtained through this evaluation, but still staying flexible to different front-end bandwidths, correlator spacing, and so forth. High multipath error (up to 20 m) in the first tracking step is compensated in the fine tracking step using a combination of DP and HRC discriminators, reducing multipath error down to less than a meter. It was shown by simulations that the proposed scheme is able to well mitigate multipath in different scenarios, including both statistical and theoretical models.

Finally, the parameters of the algorithm were further optimized and tested in different configurations through simulations using a Spirent GSS8000 simulator. The two-step tracking algorithm was also implemented in a software receiver, and performance assessments were conducted using real data recorded from the recently launched Galileo PFM and FM2 satellites.

Acknowledgments

The authors are grateful for the financial support received from the Swiss National Science Foundation (http://www.snf.ch/) under Grant 200020 134766/1, as well as from the European Community's Seventh Framework Program (FP7/2007–2013) under Grant Agreement no. 228339.

References

[1] A. Jovanovic, C. Mongrédien, C. Botteron, Y. Tawk, G. Rohmer, and P. A. Farine, "Requirements and analysis for a robust E1 Galileo tracking algorithm in the scope of the GAMMA-A project," in *Proceedings of the Institute of Navigation—International Technical Meeting 2010 (ION ITM '10)*, pp. 973–985, San Diego, Calif, USA, January 2010.

[2] D. Borio, C. Mongrédien, and G. Lachapelle, "Collaborative code tracking of composite GNSS signals," *IEEE Journal on Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 613–626, 2009.

[3] O. Julien, *Design of Galileo L1F receiver tracking loops [Thesis]*, University of Calgary, 2005.

[4] P. Fine and W. Wilson, "Tracking algorithm for GPS offset carrier signals," in *Proceedings of the US Institute of Navigation NTM Conference*, San Diego, Calif, USA, January 1999.

[5] M. S. Hodgart, P. D. Blunt, and M. Unwin, "Double estimator—a new receiver principle for tracking BOC signals," Inside GNSS, pp. 26–36, 2008.

[6] M. S. Hodgart, R. M. Weiler, and M. Unwin, "A triple estimating receiver of multiplexed binary offset carrier (MBOC) modulated signals," in *Proceedings of the 21st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS '08)*, pp. 2295–2304, Savannah, Ga, USA, September 2008.

[7] O. Julien and C. Macabiau, "Two for one—tracking Galileo CBOC signal with TMBOC," Inside GNSS, 2007.

[8] O. Julien and C. Macabiau, "On potential CBOC/MBOC common receiver arcitectures," in *Proceedings of the International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS ITM '07)*, p. 1843, ENAC, 2007.

[9] Galileo Joint Undertaking, Galileo Open Service Signal in Space InterfaceControl Document, GAL OS SIS ICD, Draft 1, 2010.

[10] A. Jovanovic, C. Mongrédien, C. Botteron, Y. Tawk, G. Rohmer, and P. A. Farine, "Implementation and robustness analysis of the two-step CBOC tracking algorithm in the scope of the GAMMA-A project," in *Proceedings of the Proceedings of the ENC GNSS*, Braunscheig, Germany, October 2010.

[11] A. Jovanovic, C. Mongrédien, C. Botteron, Y. Tawk, G. Rohmer, and P. A. Farine, "Implementation and optimization of a Galileo E1 two-step tracking algorithm using data/pilot combining and extended integration time," in *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS '11)*, Portland, Ore, USA, September 2011.

[12] Y. Tawk, A. Jovanovic, J. Leclere, C. Botteron, and P. A. Farine, "A new FFT-based algorithm for secondary code acquisition for Galileo signals," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '11)*, San Francisco, Calif, USA, September 2011.

[13] P. B. Anantharamu, D. Borio, and G. Lachapelle, "Pre-filtering, side-peak rejection and mapping: several solutions for unambiguous BOC tracking," in *Proceedings of the 22nd International Technical Meeting of the Satellite Division of the Institute of Navigation 2009 (ION GNSS '09)*, pp. 3182–3195, Savannah, Ga, USA, September 2009.

[14] J. C. Juang and T. L. Kao, "Generalized discriminator and its applications in GNSS signal tracking," in *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation 2010 (ION GNSS '10)*, pp. 3251–3257, Portland, Ore, USA, September 2010.

[15] C. Mongrédien, M. Overbeck, and G. Rohmer, "Development and integration of a robust signal tracking module for the triple-frequency dual-constellation GAMMA-A receiver," in *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation 2010 (ION GNSS '10)*, pp. 2808–2819, Portland, Ore, USA, September 2010.

[16] M. S. Braasch and G. A. Mcgraw, "GNSS multipath mitigation using gated and high resolution correlator concepts," in *Proceedings of the National Technical Meeting of the Satellite Division of the Institute of Navigation (ION NTM '99)*, pp. 333–342, San Diego, Calif, USA, January 1999.

[17] http://www.spirent.com/Solutions-Directory/GSS8000.aspx.

[18] http://www.iis.fraunhofer.de/.

[19] E. D. Kaplan and C. J. Hegarty, *Understanding GPS—Principles and Applications*, Artech House, 1996.

[20] N. Bertelsen and K. Borre, *A Software Defined GPS and Galileo Receiver*, Birkhauser, Boston, Mass, USA, 2007.

[21] J. W. Betz and K. R. Kolodziejski, "Generalized theory of code tracking with an early-late discriminator part II: noncoherent processing and numerical results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 4, article 1551, 2009.

[22] A. Jovanovic, Y. Tawk, C. Botteron, and P. A. Farine, "Multipath mitigation techniques for CBOC, TMBOC and AltBOC signals using advanced correlators architectures," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS '10)*, pp. 1127–1136, Palm Springs, Calif, USA, May 2010.

Hindawi Publishing Corporation
International Journal of Navigation and Observation
Volume 2012, Article ID 651039, 18 pages
doi:10.1155/2012/651039

Research Article

Optimized Carrier Tracking Loop Design for Real-Time High-Dynamics GNSS Receivers

Pedro A. Roncagliolo, Javier G. García, and Carlos H. Muravchik

LEICI, Facultad Ingeniería, UNLP, B1900TAG La Plata, Argentina

Correspondence should be addressed to Pedro A. Roncagliolo, agustinr@ing.unlp.edu.ar

Received 27 December 2011; Accepted 21 March 2012

Academic Editor: Carles Fernández-Prades

Carrier phase estimation in real-time Global Navigation Satellite System (GNSS) receivers is usually performed by tracking loops due to their very low computational complexity. We show that a careful design of these loops allows them to operate properly in high-dynamics environments, that is, accelerations up to 40 g or more. Their phase and frequency discriminators and loop filter are derived considering the digital nature of the loop inputs. Based on these ideas, we propose a new loop structure named Unambiguous Frequency-Aided Phase-Locked Loop (UFA-PLL). In terms of tracking capacity and noise resistance UFA-PLL has the same advantages of frequently used coupled-loop schemes, but it is simpler to design and to implement. Moreover, it can keep phase lock in situations where other loops cannot. The loop design is completed selecting the correlation time and loop bandwidth that minimize the pull-out probability, without relying on typical rules of thumb. Optimal and efficient ways to smooth the phase estimates are also presented. Hence, high-quality phase measurements—usually exploited in offline and quasistatic applications—become practical for real-time and high-dynamics receivers. Experiments with fixed-point implementations of the proposed loops and actual radio signals are also shown.

1. Introduction

A fundamental task of every Global Navigation Satellite System receiver is to synchronize with the visible satellite signals. Since Direct Sequence Spread Spectrum (DS-SS) signals are utilized, code and carrier synchronization is required, but a correlation stage is necessary to despread the signals before the synchronization algorithms can be applied. In real-time receivers the required economy of operations usually precludes the use of complex estimation schemes and tracking loops are preferred. Due to the correlation process these loops are necessarily discrete. The typical trade-off in tracking loop design is bandwidth versus dynamic performance: output noise increases with a larger loop bandwidth, while dynamic tracking error decreases with it [1]. Thus, the loop design becomes particularly challenging when the receivers are subject to high dynamics. To overcome this limitation other receiver structures have been proposed in [1], claiming tracking capability up to 150 g of acceleration, in contrast with the 5 g regularly assigned to tracking loops. However, the required computational burden is large since several simultaneous correlations and Fast Fourier Transform (FFT) computations are needed. In this paper we show a careful design of the digital loops that can expand their tracking ability to acceleration steps up to 40 g or even more, keeping a low computational load and reasonable tracking threshold values at the same time.

The loop structure known as FLL-assisted PLL [2] is very often adopted for GNSS receivers. It consists of a Phase-Locked Loop (PLL) and a Frequency-Locked Loop (FLL) in a coupled mode, with the advantage of reducing locking times and avoiding false locks. This solution is also a legacy of analog loops since the FLL or Automatic Frequency Control (AFC) has been used to reduce frequency errors as a previous stage to phase lock for analog PLL [3]. The advantages of adding the FLL to track spread spectrum signals in dynamic environments were already studied in [4]. For high-dynamics GNSS receivers, the focus is on carrier loops because the carrier shares the same dynamics as the code. Then, the estimation of the carrier frequency can be used to aid the estimation of the code frequency, and a first-order code loop is enough [5]. Usually, implementations of FLL-assisted

PLL are not based on optimal digital loop solutions, with each loop designed separately, leaving the analysis of their interactions and possible modifications to the simulation stage [2, 5, 6]. Moreover, schemes adopted to discriminate phase or frequency errors are often justified because of their similarity with well-known analog solutions rather than with an optimality versus implementation complexity criterion. We will show that digital implementations of optimal discriminators are not necessarily more complex and allow designing the FLL-assisted PLL in a coupled way.

Nevertheless, the FLL-assisted PLL leads to a more complex design and a computationally more expensive implementation than a single PLL. Moreover, when coupled-loops lose phase lock for a moment, they present cycle slips introducing a phase ambiguity. We will show how to use the same frequency information as that of an FLL to build a nonambiguous phase detector, the Unambiguous Frequency-Aided (UFA) phase discriminator. A PLL with this new phase discriminator, that is, a UFA-PLL, keeps the desirable properties of an FLL without demanding an extra loop and avoiding cycle slips. Other nonambiguous phase discriminators are known for analog PLLs, that is, with analog loop filter, such as the sequential discriminators built with flip-flops presented in [7, 8] or the nonsequential discriminator of [9]. While their goals are quite similar to ours, they increase the PLL implementation complexity, demanding some digital circuitry and a digital-to-analog converter to get the analog phase error. On the contrary, the UFA phase discriminator is easily implemented and naturally suited for a software-based PLL, leading to a less complex implementation than a FLL-assisted PLL. Section 2 introduces the UFA-PLL structure for GNSS tracking loops.

The optimum loop filter structure for analog PLLs was introduced in [10], solving the mentioned bandwidth trade-off by minimizing a quadratic functional. A widespread technique for designing digital loops is discretizing an analog loop with a sample rate $1/T$ at least ten times faster than loop bandwidth B_N [5, 11, 12]. As $B_N T$ increases above the rule-of-thumb value of 0.1, the resulting loop deviates from optimal and may become unstable [5], especially when accounting for the delays of a digital implementation. This limit imposed to the loop bandwidth is not fundamental and an attempt to avoid it has been presented in [13]. They introduced a digital loop design based on pole placement that allows somewhat larger $B_N T$ values. However, the pole location is assigned with standard second-order analog-system rules. Our approach is to consider a completely digital loop model and pose the bandwidth trade-off directly in the digital domain, building upon the early and often overlooked work of [14] for hybrid loops. We include two delays in the loop to consider the effect of the correlation stage, similar to the inclusion of an accumulator before the loop error discriminator for signals without spreading codes [15]. Our method [16] allows the design of stable loops with $B_N T > 0.1$, a particularly useful feature for high-dynamics receivers. Specifically, we will focus on dynamics modeled as acceleration steps, that is, unbounded jerk, as in the case of launching vehicles when the engine turns on or off. In Section 3 we first derive the optimal loop filter for arbitrary phase inputs and then for the case of acceleration steps that produce quadratic ramps of input phase or a linear ramp of the input frequency. Simulations comparing the different loop structures are also shown.

Optimization gives the structure for the loop filter, leaving the choice of T and B_N unsolved. Usually, these parameters are selected based on some rule of thumb [2, 5], and the ultimate loop performance, as measured by the pull-out probability and/or tracking threshold, is obtained later by simulation. An optimal choice of these fundamental parameters demands an analysis of the nonlinear aspects of the tracking loop with noise. This is quite difficult, although some results are known for analog loops with relatively simple loop filters, by solving a Fokker-Planck equation [17]. They can be extended to digital loops when an analog approximation is valid [18]. Our approach is to get a reasonable approximation for the pull-out probability and its relationship to the loop parameters. This new approach introduced in [19] allows us considering dynamics modeled as acceleration steps and digital loop filters with zero stationary error response to these inputs. Previous analyses are based on stationary loop responses or sinusoidal acceleration profiles [2, 5]. For these cases, we derive approximate expressions for the probability of starting a nonlinear behavior of the mentioned loops. These expressions quantify the role of B_N and T and let us choose them in order to obtain lower tracking thresholds for different dynamic scenarios, as presented in Section 4.

Our optimized digital carrier tracking loops also allow smoothing of the phase estimates incorporating more measurements, at the expense of some delay. In general, an output delay of a few samples should not be a limitation since the navigation task in a GNSS receiver is usually slow compared with the loop sample rate. This update of the phase estimates can significantly reduce the noise variance and the transient responses in high-dynamics environments. This strategy is suitable for real-time receivers because it can be efficiently calculated. Hence, some of the precise positioning techniques would be applicable in real-time and for high-dynamics receivers. Consider, for instance, smoothing of code delay measurements with carrier phase estimates in stand-alone receivers [12, 20], or differential positioning applications [12, 21], or even attitude estimation with GNSS signals [22]. In all these cases, an improvement in the phase estimation has a direct impact on the positioning performance. The expressions for optimal smoothing filters are derived in Section 5, and their efficient implementation is also discussed there. In addition, we present experimental results obtained with actual RF signals and a fixed point implementation of our loops tracking acceleration steps of up to 40 g. Finally, the conclusions of this work are given in Section 6.

2. Digital Loops Models

Correlations of the received signal with the locally generated replicas for each visible satellite are the inputs to the discriminator of the carrier tracking loops in a GNSS receiver. The complex correlation for a given satellite with

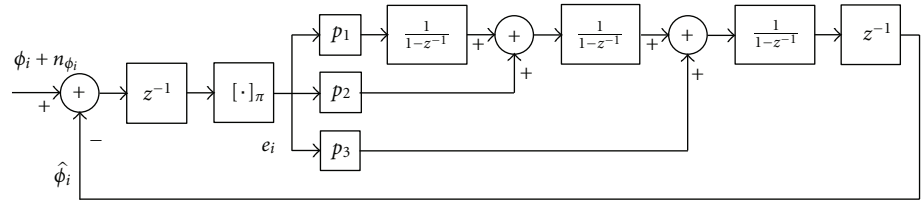


Figure 1: Block diagram of classical PLL structure.

carrier power to noise power spectral density C/N_0 and for the ith correlation interval of duration T can be written as [12]

$$C_i = D_i\sqrt{\frac{TC}{N_0}}\,\mathrm{sin}\,c(\Delta f_i)R(\Delta\tau_i)e^{j(\pi\Delta f_i+\Delta\theta_i)} + n_i, \qquad (1)$$

where $\Delta\tau_i = \tau_i - \hat{\tau}_i$ is the code delay estimation error, $\Delta f_i = f_i - \hat{f}_i$ the frequency estimation error, and $\Delta\theta_i = \theta_i - \hat{\theta}_i$ the phase estimation error, all assumed constant during the integration time. The sequence n_i is a complex white Gaussian noise process with unit variance, $R(\cdot)$ is the code correlation function, and $\mathrm{sin}\,c(x) = \sin(\pi x)/(\pi x)$. It is also assumed that the signal has binary data bits $D_i = \pm 1$ and that correlations are computed within a single bit period. This Binary Phase Shift Keying (BPSK) modulation is present in many GNSS signals like the GPS civil signals or in the data components of composite modernized GNSS signals [23].

In tracking conditions (i.e., after the acquisition process has been completed [12]), estimation errors are small and then the functions $\mathrm{sin}\,c(\cdot)$ and $R(\cdot)$ can be approximated by 1. In this case the expression (1) reduces to

$$C_i = I_i + jQ_i \approx D_i\sqrt{\frac{TC}{N_0}}e^{j\Delta\phi_i} + n_{I_i} + jn_{Q_i}, \qquad (2)$$

where I_i and Q_i are the so-called in-phase and in-quadrature correlations respectively, $n_{I_i} = \Re e\{n_i\}$, $n_{Q_i} = \Im m\{n_i\}$, and we have defined $\Delta\phi_i = \phi_i - \hat{\phi}_i$, with $\phi_i = \pi f_i + \theta_i$ and $\hat{\phi}_i = \pi\hat{f}_i + \hat{\theta}_i$. These sequences allow to model the carrier tracking loop as a purely digital single-input single-output (SISO) system. When the frequency is changing according to a constant acceleration error of $a\,\mathrm{m/s^2}$, we verified—by numerical integration—that expression (2) is a good approximation if $aT^2/\lambda \ll 1$, where λ is the wavelength of the signal. For L1 GPS $\lambda = 0.19\,\mathrm{m}$ and with $T = 5\,\mathrm{ms}$, this implies that $a \ll 7600 \equiv 775\,\mathrm{g}$. In this case, the terms Δf_i and $\Delta\phi_i$ have to be reinterpreted as the average frequency error and average phase error during the correlation interval, respectively.

In the following we briefly review the basic concepts of PLL and FLL-assisted PLL from our digital point of view, and later we introduce the UFA-PLL.

2.1. PLL Model. The phase estimation error is typically obtained using one of several possible discriminators [5], which give the desired phase modified by different memoryless nonlinearities. The optimal one—maximum likelihood estimator—is given by

$$e_i = \tan^{-1}\left(\frac{Q_i}{I_i}\right) = \left[\Delta\phi_i + n_{\phi_i}\right]_\pi, \qquad (3)$$

where the notation $[\cdot]_\pi$ indicates that its argument is kept within the interval $(-\pi/2, \pi/2]$ by adding or subtracting π as many times as needed. The zero-mean noise term n_{ϕ_i} has a rather complicated probability distribution [24], but in high C/N_0 it can be approximated by a Gaussian distribution with zero mean and variance $\sigma^2_{\phi_i} \approx 1/(2TC/N_0)$.

A four-quadrant $\tan^{-1}(\cdot)$ is not appropriate if there is BPSK data modulation because the discriminator becomes sensitive to the data phase changes. On the contrary, for signals without data the range of the discriminator can be doubled with a four-quadrant $\tan^{-1}(\cdot)$. We chose this discriminator because it is not amplitude dependent and the calculation of $\tan^{-1}(\cdot)$ can be easily implemented with a lookup table, since in practice I_i and Q_i are frequently quantized to a few bits.

In order to close the loop in our model, it is of crucial importance to consider the delays present in a real implementation. Failure to account for a delay may turn unstable an optimal loop design. Since ours loops are digital, a single sample delay is expected but in fact there are two. One of them is due to the time spent in I_i and Q_i calculations. The other delay appears because the estimated values used in the present correlations have to be known before the calculations begin. That is, the value $\hat{\phi}_i$ is obtained with the loop filter output of the $(i-1)$th correlation interval, which in turn is calculated with the estimation errors of $\hat{\phi}_{i-2}$. Then, with these considerations, the model of a PLL using the classical loop filter structure of type 3, that is, with three accumulators, is shown in Figure 1.

2.2. FLL-Assisted PLL Model. To add an FLL to our previous PLL, a frequency discriminator is needed. In a digital loop a frequency error estimate may be obtained as the difference of two successive phase errors, and in fact this is often correct. A problem appears when the discontinuities caused by $[\cdot]_\pi$ make that the difference to be wrong in $\pm\pi$. However, our discrete system cannot distinguish frequencies greater than half of the sample rate, that is, phase changes of π between consecutive samples, and so the measured frequency errors must be bounded. In fact, if the phase discriminator is insensitive to BPSK data, the phase changes caused by

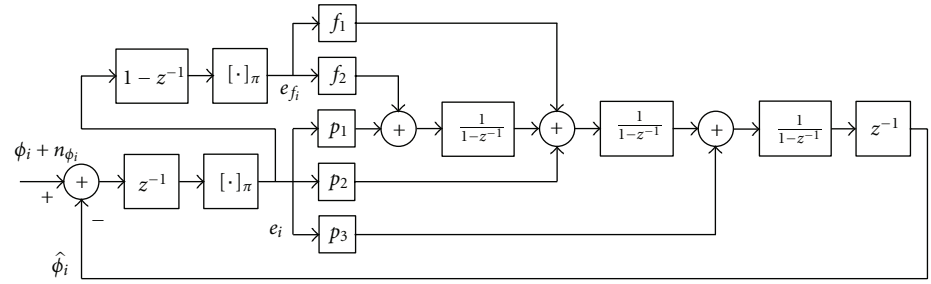


Figure 2: Block diagram of the typical FLL-assisted PLL structure.

frequency errors must lie in the interval $(-\pi/2, \pi/2]$ [25]. Thus, the difference of two consecutive outputs of the phase discriminator can be corrected just using the operation $[\cdot]_\pi$. Therefore, the frequency discriminator for the FLL can be obtained by

$$e_{f_i} = [e_i - e_{i-1}]_\pi. \tag{4}$$

Figure 2 shows a diagram of the FLL-assisted PLL presented in [2], where the second-order loop filter of the FLL shares the same cascade of accumulators used by the PLL filter.

In the locked condition $e_i = \Delta\phi_i$ and $e_{f_i} = \Delta\phi_i - \Delta\phi_{i-1}$ are small enough to justify a linear analysis of the loop. The complete loop is seen as an equivalent PLL with filter coefficients p_3, $p_2 + f_2$, and $p_1 + f_1$, instead of p_3, p_2, and p_1. Thus, the FLL is inserted into the model of the PLL at a design stage. This eliminates the constrain of using a narrow bandwidth FLL to not significantly perturb the PLL behavior, as in [2, 6]. A wide bandwidth FLL allows the loop to have two regions of operation: "phase-locked" as it was described before, and "frequency-locked" when the dynamics unlocks the PLL but the FLL keeps the frequency error within the linear range of its discriminator. In the latter region the loop is governed by the FLL (coefficients f_1 and f_2) and the phase error input acts like a zero-mean perturbation [25]. As soon as the dynamics let the loop reduce its frequency error close to zero, the phase lock can be restored.

2.3. The UFA-PLL Model. As we have seen so far, due to the cyclic nature of phase a memory-less discriminator is unable to distinguish changes of an integer number of cycles—or half cycles if there is BPSK data—, that is, its output is ambiguous. However, it is possible to obtain a frequency error estimate from these ambiguous phase error estimates correcting their difference with the nonlinear operation $[\cdot]_\pi$. This is the reason why an FLL can cope with carrier tracking in situations when a single PLL cannot. Assume that there is BPSK data and the PLL phase error is rising and crosses the value $\pi/2$. The output of the phase discriminator abruptly changes to a value close to $-\pi/2$, reversing the evolution of the PLL phase. Hence, the phase error will increase since the PLL is now moving in the wrong direction. We should instruct the phase discriminator with information of the phase derivative to keep moving in the right direction, that is, we should feed it with proper frequency information available at the FLL. Therefore, the idea of the Unambiguous Frequency-Aided (UFA) phase discriminator is to use the same frequency information used by the FLL to get a better phase discriminator. It works correcting the ambiguous values of e_i by adding or subtracting an integer number of π so that the difference of successive values of the corrected phase error, u_i, gives the right frequency error. Then, the equations that define our new phase error estimate, for $i \in \mathbb{N}$, are

$$u_i = k_i\pi + e_i, \quad k_i \text{ such that } I_\pi(u_i - u_{i-1}) = 0, \tag{5}$$

where we define $I_\pi(x) = x - [x]_\pi$, an operation similar to the function integer part but with steps at the multiples of π. A practical formula to compute k_i can be derived noting that $I_\pi(x + l\pi) = I_\pi(x) + l\pi$, $l \in \mathbb{Z}$ since

$$I_\pi(k_i\pi + e_i - u_{i-1}) = k_i\pi + I_\pi(e_i - u_{i-1}) = 0 \tag{6}$$

and then $k_i\pi = -I_\pi(e_i - u_{i-1})$. Substituting this in (5), we can recursively calculate the UFA phase error from the ambiguous e_i:

$$u_i = e_i - I_\pi(e_i - u_{i-1}) \tag{7}$$

with starting value $u_0 = e_0$. Then, the PLL structure in Figure 1 transforms into a UFA-PLL just adding a block that implements (7) immediately after the phase discriminator output, as shown in Figure 3.

It is interesting to note that the UFA scheme acts like the phase unwrapping algorithm proposed in [18] for correcting cycle slips in the phase estimates of feed-forward synchronizers. In this case, the phase correction does not affect the phase estimation process since it is done once the estimation stage is finished. On the contrary, the UFA phase discriminator modifies the behavior of our feedback estimator, the PLL, changing its nonlinear characteristics. As a result, cycle slips and the rather complex transient responses induced by them are avoided as long as the frequency error is compatible with the loop sample rate.

2.4. Equivalence between UFA-PLL and FLL. We saw that the frequency error estimate can be obtained as the difference of two successive phase errors if the result is kept in range by adding or subtracting an integer number of π. Thus, the frequency discriminator for the FLL can be obtained

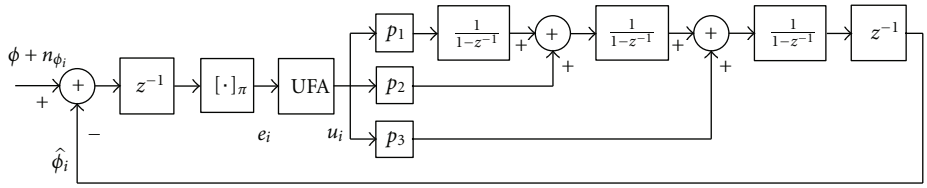


Figure 3: Block diagram of the new UFA-PLL structure.

correcting the difference of two consecutive outputs of the phase discriminator just using the operation $[\cdot]_\pi$. Figure 4(a) shows a block diagram of a digital FLL, with loop filter transfer function $F(z)$. Notice that the two delays and the accumulator that convert the frequency estimation to phase before the feedback are not included in $F(z)$.

An alternative way to obtain the same frequency error discriminator is to use the UFA algorithm previously described. Indeed, the output phase sequence u_i is built in such a way that the difference of consecutive values produces the right frequency error, as seen from (4) and (5). Therefore, the schemes of Figures 4(a) and 4(b) are equivalent. The interesting fact in Figure 4(b) is that most linear blocks are adjacent. Thus, the differentiator cancels with the accumulator without changes in the dynamic loop response, except for the mean value of the phase error, leading to the equivalent UFA-PLL model of Figure 4(c). In fact, this zero-pole cancellation shows why the FLL is insensitive to constant phase errors whereas the equivalent UFA-PLL is not. More importantly, the equivalence reveals that the nonlinear behavior of the UFA-PLL is equal to that of a FLL with the same $F(z)$, and then their tracking capacity and noise resistance are the same.

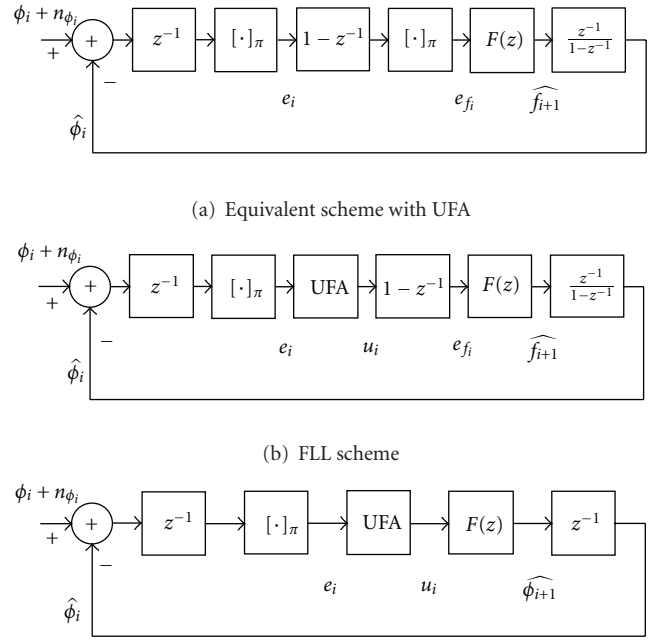


Figure 4: Equivalence between UFA-PLL and FLL.

3. Optimal Loop Filter Design

We propose to design the digital loop filter minimizing a two-term quadratic functional to handle the bandwidth trade-off mentioned in the Introduction. The input signal is assumed to have a part related to phase evolution ϕ_i plus additive, zero mean,and noise $n_{\phi i}$. The functional to be optimized is

$$J = \sigma_N^2 + \alpha^2 E_T(\phi_i), \tag{8}$$

where α^2 is a weighting factor that controls the trade-off between noise and transient response, that is, the loop bandwidth, σ_N^2 is the noise variance at the loop output, and $E_T(\phi_i)$ represents the energy of the tracking error $\Delta\phi_i$ transient response. Since the functional uses the energy of the transient response, the optimum filter must produce a zero stationary response for the given input.

Suppose $F(z)$ is the loop-filter transfer function to be found, and consider that the linear model hypothesis holds for a PLL or FLL. The closed loop transfer function including the delays is

$$T(z) = \frac{F(z)z^{-2}}{1 + F(z)z^{-2}} = Y(z)z^{-2}, \tag{9}$$

where $Y(z)$ must be a causal and stable rational transfer function. The minimization of the functional J written in terms of $Y(z)$ is shown in the Appendix A. The optimum transfer function is given by

$$Y(z) = \frac{X(z)z}{\Psi(z)}, \tag{10}$$

where $\Psi(z)$ and $X(z)$ can be obtained from the spectral factorization of (A.6) and from the partial fraction expansion of (A.8), respectively. We repeat them here for completeness,

$$\Psi(z)\Psi(z^{-1}) = 1 + \gamma^2\Phi(z)\Phi(z^{-1}), \tag{11}$$

$$G(z) = \frac{\gamma^2\Phi(z)\Phi(z^{-1})z}{\Psi(z^{-1})} = X(z) + W(z^{-1}), \tag{12}$$

where $\Phi(z)$ is the z-transform of ϕ_i. The relation between minimizing (8) to Wiener filtering [26] is sketched in Appendix B. Observe that whereas the former is a mixed criterion with a term depending on the stochastic part and a term depending on the deterministic part of the phase signal;

the latter criterion stems from a purely stochastic formulation. The connection between both approaches arises when modeling the input phase as white noise passing through a rational transfer function. The Wiener filtering approach offers other possibilities such as keeping the optimality for a wide range of admissible transfer functions via a robust approach as in [27] or considering continuous models for the phase as in [28].

Optimum loop filters for an input phase step, frequency step, and frequency ramp were derived in [16]. In the following, only the last result is presented for the sake of brevity. Analog loop filters optimized for these kind of inputs are the origin of the classical methods of filter design for type one, two, and three loops, respectively. As it will be seen, our purely discrete design for each case has one extra pole, due to the loop delays. This additional pole does not appear when discretizing analog designs, but it has a decisive influence on the stability or the range of achievable product $B_N T$.

3.1. Optimum Filter for a Frequency Ramp. The ramp is modeled as

$$\Phi(z) = \frac{\dot{\Delta}\omega T^2}{(1-z^{-1})^3}, \tag{13}$$

where $\dot{\Delta}\omega$ is the rate of frequency change. Denoting $\nu = \dot{\Delta}\omega^2 T^4 \gamma^2$, from (11) it is necessary to solve $(z-1)^6 - \nu z^3 = 0$. The six roots of this polynomial are obtained using the fact that three of them are the inverses of the other three. This allows us to express the following equations:

$$\begin{aligned} z_{1,2} + z_{1,2}^{-1} &= 2 - \frac{1 \pm j\sqrt{3}}{2}\sqrt[3]{\nu}, \\ z_3 + z_3^{-1} &= 2 + \sqrt[3]{\nu} \end{aligned} \tag{14}$$

that determine the values of z_1, z_2, complex conjugates and a real z_3. Using these values and (11), we get

$$\Psi(z) = \frac{(1-z_1 z^{-1})(1-z_2 z^{-1})(1-z_3 z^{-1})}{(1-z^{-1})^3 (z_1 z_2 z_3)^{1/2}}, \tag{15}$$

and replacing in (12)

$$G(z) = \frac{-(z_1 z_2 z_3)^{-1/2} \nu z^4}{(z-z_1^{-1})(z-z_2^{-1})(z-z_3^{-1})(z-1)^3}. \tag{16}$$

Then, the corresponding $X(z)$ has only three poles in $z = 1$, and the closed-loop transfer function of (10) is

$$Y(z) = \frac{A - Bz^{-1} + Cz^{-2}}{(1-z_1 z^{-1})(1-z_2 z^{-1})(1-z_3 z^{-1})}, \tag{17}$$

where $A = (6 - 3z_s + z_d)$, $B = (8 - 3z_s + z_p)$, and $C = 3 - z_s$, with $z_s = z_1 + z_2 + z_3$, $z_p = z_1 z_2 z_3$ and $z_d = z_1 z_2 + z_1 z_3 + z_2 z_3$. Then, the optimum loop filter with four poles, three of the input and the extra one, is

$$F(z) = \frac{A - Bz^{-1} + Cz^{-2}}{(1-z^{-1})^3 (1 + Cz^{-1})}. \tag{18}$$

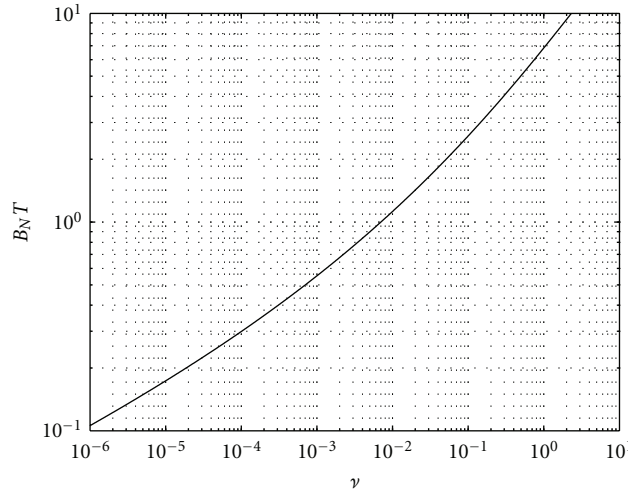


Figure 5: Type 3 loop noise equivalent bandwidth.

For the purpose of implementation, it is desirable to use a cascade of accumulators. Then, (18) can be rewritten as

$$F(z) = \frac{p_3 + p_2(1-z^{-1}) + p_1(1-z^{-1})^2}{(1-z^{-1})^3 (1 + p_1 z^{-1})}, \tag{19}$$

where $p_1 = C$, $p_2 = B - 2C$, and $p_3 = A - B + C$. The closed loop noise equivalent bandwidth is shown in Figure 5. This curve allows choosing the appropriate value of ν for a given normalized noise bandwidth. Observe also from this figure that the larger is the step or the parameter ν, the more emphasis is given to the transient energy, causing the normalized noise bandwidth to increase. The product $B_N T$ levels out to a value of approximately 54.5. This part is not included in the figure since it is of minor importance for most designs of practical interest.

3.2. Design Example: Loops for Launching Vehicles. We simulate and compare the loop models presented in Section 2 taking as an example a GPS carrier tracking loop for launching vehicles [29]. In this case the dynamic input can be modeled as an acceleration step, which becomes a quadratic ramp in terms of phase and a linear one, in terms of frequency. For these inputs the optimal loop filter for a PLL was obtained in Section 3.1. The case of an FLL-assisted PLL, taking results from [29], leads to a type 2 FLL and a type 3 PLL.

The FLL-assisted PLL in [29] was designed to operate in "phase-locked" mode with steps up to 10 g and in "frequency-locked" mode up to 20 g of acceleration. These requirements were too demanding for the commonly adopted correlation time of 10 ms, and then it was lowered to 5 ms (at the cost of almost doubling the processor load and an increase in the tracking threshold). A typical rule of thumb for keeping a reasonable distance from the pull-out values of the loop is that the peak of the error transient has a maximum value given by half the linear range of the phase discriminator, an eighth of cycle [2]. As it will be shown in the simulations this condition was obtained with a value of $\nu = 0.00025$ for the PLL. However, as we will explain

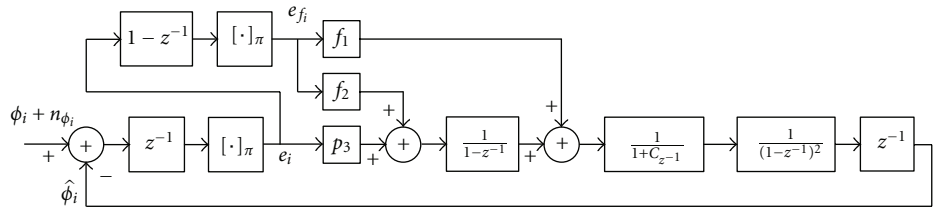


Figure 6: Block diagram of the FLL-assisted PLL in Section 3.

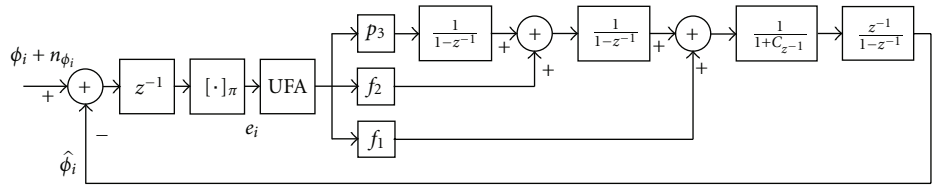


Figure 7: Block diagram of the UFA-PLL designed in Section 3.

in Section 4, this rule is not that useful. The resulting filter transfer function for the PLL was

$$F(z) = \frac{A - Bz^{-1} + Cz^{-2}}{(1 - z^{-1})^3(1 + Cz^{-1})}, \tag{20}$$

where $A = 0.6173$, $B = 1.105$, and $C = 0.5$. Then, in the structure of Figure 1 this implies $p_1 + f_1 = C = 0.5$, $p_2 + f_2 = B - 2C = 0.105$, and $p_3 = A - B + C = 0.0123$, plus a block that implements the extra pole in $z = -C$. The resulting PLL equivalent noise bandwidth is $B_N = 75.6$ Hz.

Since the FLL design does not affect the previous results, it was designed wider than strictly necessary in order to facilitate the posterior implementation. The selected transfer function is

$$F(z) = \frac{D - Ez^{-1}}{(1 - z^{-1})^2(1 + Ez^{-1})}, \tag{21}$$

where $D = 0.6$ and $E = 0.5$, resulting in that the extra pole needed for the FLL and the PLL is the same. Then, $f_1 = E = 0.5$ and $f_2 = D - E = 0.1$. This implis $p_1 = 0$ and $p_2 \approx 0$. With these simplifications the complete loop design reduces to the diagram showed in Figure 6. This FLL loop can track steps up to 40 g with transient error peaks smaller than 25 Hz, half of the linear range of the frequency discriminator, with an equivalent noise bandwidth of $B_N = 61.3$ Hz.

We will use the previous loop filter as a basis for the comparison of different loop configurations. We consider a PLL and a UFA-PLL with the same loop filter as before, that is $p_1 = C = 0.5$, $p_2 = 0.105$, $p_3 = 0.0123$. This structure is showed in Figure 7. The phase error response for a step of 10 g of acceleration, common to the three loops as expected, is depicted in Figure 8. In this case the phase error detected by the discriminators is equal to the actual phase error since its magnitude is always less than a quarter of cycle.

In Figure 9 the discriminated phase error during a 40 g step in the three loops is illustrated. It can be seen that

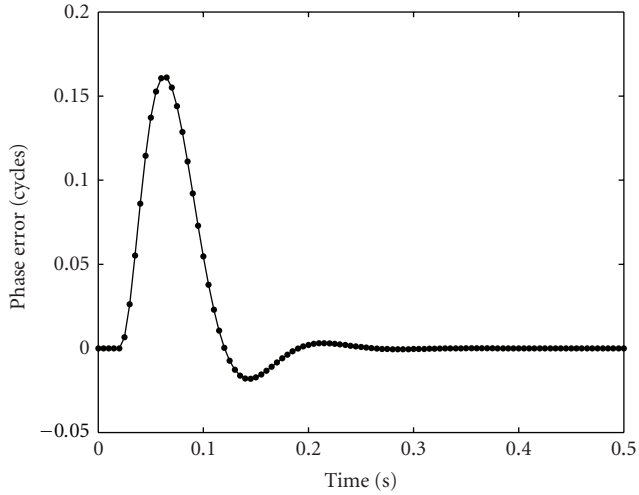


Figure 8: Phase error during a 10 g step.

the PLL cannot track this step, whereas the others can. The response of the UFA-PLL is a scaled version of the response for 10 g, showing the effect of linearization of the discriminator characteristic achieved by the UFA algorithm. In the case of the FLL-assisted PLL the loop lost phase lock for a moment, but could still track the dynamics because the FLL remained locked. This nonlinear behavior could correspond to a cycle slip. To verify this analysis the actual phase error for each configuration is shown in Figure 10. Clearly, there is one cycle slip in the tracked phase of the FLL-assisted PLL, whereas the UFA-PLL is able to track this step of acceleration without any cycle slip.

The limit of the tracking capability of the FLL-assisted PLL and the UFA-PLL is the frequency error—the phase change between samples—, since both are frequency-aided loops. If this error becomes greater than 50 Hz in magnitude, the frequency estimation will be ambiguous, in the same way

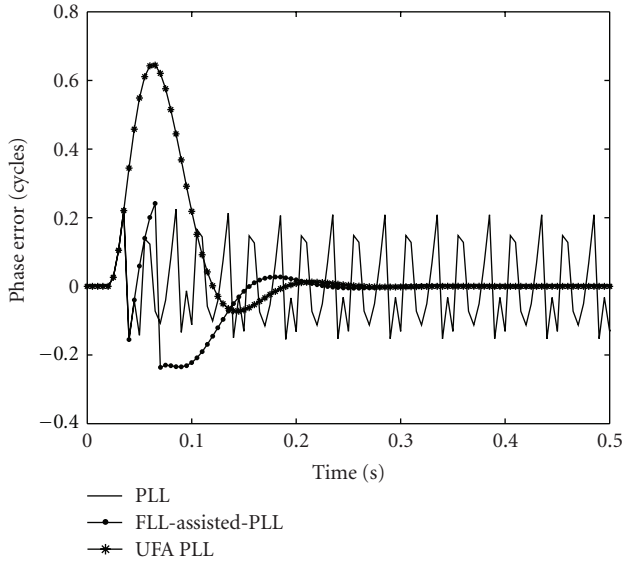


Figure 9: Discriminated phase error during a 40 g step.

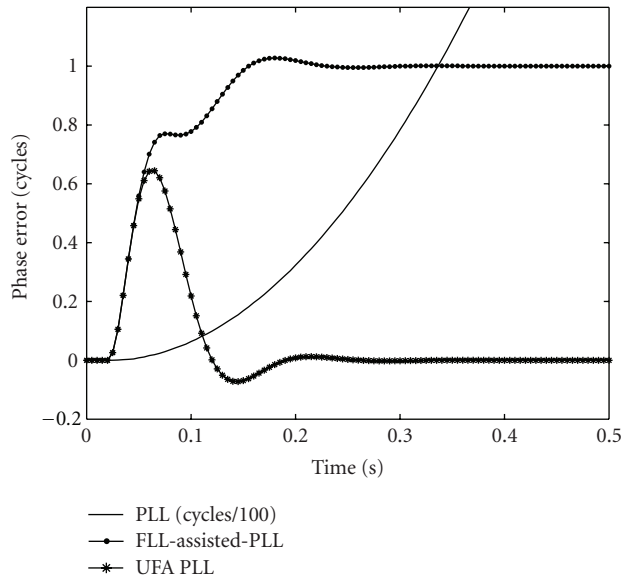


Figure 10: Actual phase error during a 40 g step.

as the phase of the PLL does, and the loops will lose their lock. This can be caused by excessively high dynamics or noise power or a combination of both. Notice that the noise power considered in this case is twice the input phase noise power due to the differencing. In fact, we can use the UFA algorithm applied to the frequency discrimination to further extend the dynamics resistance of the loops. However, it will be of little practical importance due to the noise power increase caused by a new differentiation. The frequency error of both loops during a 40 g step is shown in Figure 11. The peak error is 25 Hz—half the limit—and thus, using the same rule of thumb that we used for the phase error of the PLL, it can be argued that 40 g is the level of acceleration steps that can be tracked with a reasonable safety margin for noise effects. In Section 4 we will give a totally different approach for the consideration of the noise in the UFA-PLL.

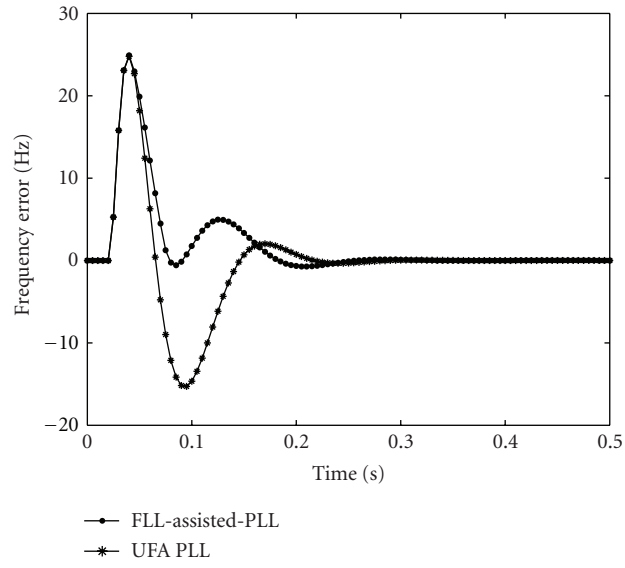


Figure 11: Frequency error during a 40 g step.

Another shared feature of the FLL-assisted PLL and the UFA-PLL is the resistance to false locks. For the sake of brevity this analysis is not included here but it can be seen in [16].

4. Pull-Out Probability Analysis

In this section we will compute an approximation to the pull-out probability for a loop in a given operating condition. Since the pull-out is necessarily a consequence of nonlinear behavior, a simple way to bound the pull-out probability is with the probability of entering nonlinear behavior, that is, the limits of the $\tan^{-1}(\cdot)$ range. We are interested in tracking acceleration steps. Then, it is clear that for a given noise level the instants when the loop is closer to these boundaries approximately correspond to the peaks of the loop transient response. Therefore, we focus on calculating the probability of entering nonlinear behavior at the instant of the transient response peak, given that the loop behavior has been linear up to that time.

4.1. PLL Analysis. The PLL enters nonlinear behavior when the phase error becomes larger than $\pi/2$. This is equivalent to a sign reversal of the in-phase component, with respect to the sign of the data bit. Then, the probability of nonlinear behavior at the transient peak $i = p$ is [19]

$$\begin{aligned} P_P &= P\left\{\left|\Delta\phi_p + n_{\phi_p}\right| > \frac{\pi}{2}\right\} \\ &= P\left\{\cos\left(\phi_p - \hat{\phi}_p\right) + n'_{I_p} < 0\right\}, \end{aligned} \tag{22}$$

where $n'_{I_p} = D_p n_{Ip}/\sqrt{TC/N_0}$ has variance $\sigma^2 = 1/(2TC/N_0)$. Assuming that the PLL has had a linear behavior up to the analyzed instant, $\hat{\phi}_p$, can be thought of as a deterministic value plus output noise $n_{\hat{\phi}_p}$. Subtracting this deterministic value from ϕ_p we find the peak value of the loop transient

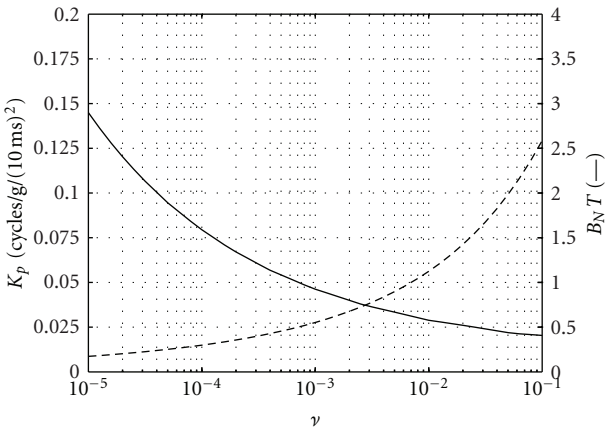


Figure 12: Noise Bandwidth and Peak Error Response for type 3 PLL.

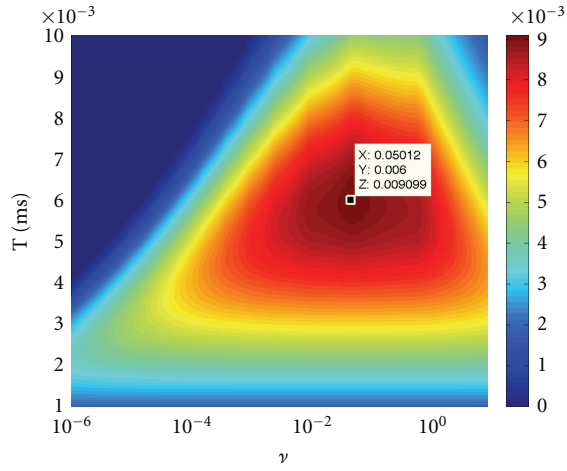


Figure 13: Function $f_{\text{PLL}}(\nu, T)$ for type 3 PLL and 5 g.

response to an acceleration step, which is shown in Figure 12 as a function of ν for an integration time of 10 ms. Note that with acceleration we refer to the Doppler rate the loop has to track, scaled in units of g = 9.8 m/s^2 instead of Hz/s to keep an easy physical interpretation. This peak value is proportional to the square of the integration time because an acceleration step is a parabolic ramp of phase. The values of the normalized closed-loop equivalent noise bandwidth are also plotted in Figure 12 for completeness.

The noise $n_{\hat{\phi}_p}$ depends on the filtered past input noise—from instants $p-2$, $p-3$, and so on—and is statistically independent of n'_{I_p}. Then, we can write $\phi_p - \hat{\phi}_p = K_p a T^2 + n_{\hat{\phi}_p}$, where a is the acceleration amplitude in g's. In (22), we obtain

$$P_P = P\left\{\cos\left(K_p a T^2 + n_{\hat{\phi}_p}\right) + n'_{I_p} < 0\right\}. \tag{23}$$

To further simplify (23), we use a "low noise" approximation $\cos(n_{\hat{\phi}_p}) \approx 1$ and $\sin(n_{\hat{\phi}_p}) \approx n_{\hat{\phi}_p}$ and obtain

$$P_P \approx P\left\{\cos\left(K_p a T^2\right) - \sin\left(K_p a T^2\right) n_{\hat{\phi}_p} + n'_{I_p} < 0\right\}. \tag{24}$$

Using central limit theorem arguments, it is reasonable to approximate the distribution of $n_{\hat{\phi}_p}$ as a zero mean Gaussian with variance approximated by $2B_N T\sigma^2_{\phi_i} \approx B_N/(C/N_0)$, using the high C/N_0 variance expression for the input noise. In this case, both random terms in (24) become a single Gaussian random variable with zero mean and variance $(1 + 2B_N T \sin^2(K_s a T^2))\sigma^2$. Therefore,

$$P_P \approx Q\left(\sqrt{\frac{2TC/N_0 \cos^2\left(K_p a T^2\right)}{1 + 2B_N T \sin^2\left(K_p a T^2\right)}}\right), \tag{25}$$

where $Q(x)$ is the cumulative Gaussian distribution from x to ∞. Since the function $Q(\sqrt{x})$ is monotonically decreasing, we can define a function

$$f_{\text{PLL}}(\nu, T, a) = \frac{2T \cos^2\left(K_p a T^2\right)}{1 + 2B_N(\nu) T \sin^2\left(K_p a T^2\right)} \tag{26}$$

such that the larger is f_{PLL}, the smaller is P_P for a given C/N_0, and then a low tracking threshold is attained. For example, $f_{\text{PLL}}(\nu, T, a)$ for a 5 g acceleration step is plotted in Figure 13. It can clearly be seen that the larger values of f_{PLL} are found in the region 5 ms $< T <$ 7 ms and $10^{-3} < \nu < 10^0$, and the maximum is approximately at $\nu = 0.05$ and $T = 6$ ms. The value of f_{PLL} in this region is about 0.009. Larger values of ν are not preferred because they lead to loops that produce larger phase estimation error and only a slightly lower P_P. However, it must be emphasized that for values of $\nu > 10^{-3}$ the loop bandwidth is $B_N T > 0.5$. These values of bandwidth show that a filter loop design based on discretization of analog solutions, only valid for $B_N T < 0.1$, is not appropriate to design loops with better tolerance to nonlinear behavior. If $\nu = 0.02$, then $B_N T = 1.5$ and, with $T = 5$ ms, leads to an optimum loop bandwidth $B_N = 300$ Hz, which is too large from the point of view of output phase error variance. This shows that the main cause of pull-out in PLLs with narrow bandwidths is the transient error response, due to the input phase and the input noise, rather than output noise. In other words, 5 g of acceleration is very demanding for a single PLL and then a large bandwidth is required to track them with small pull-out probability.

4.2. UFA-PLL Analysis. An equivalent description of the UFA algorithm presented in Section 2 is to consider it as a modified $\tan^{-1}(\cdot)$ function that produces output values in the range $(-\pi/2 + u_{i-1}, \pi/2 + u_{i-1}]$ instead of $(-\pi/2, \pi/2]$. Hence, we conclude that nonlinear behavior will occur if the actual phase error differs from the previously discriminated one by more than $\pi/2$. Assuming that the behavior before the analysis time has been linear, $u_{i-1} = \Delta\phi_{i-1} + n_{\phi_{i-1}}$ and then

$$P_U = P\left\{\left|\Delta\phi_i + n_{\phi_i} - \Delta\phi_{i-1} - n_{\phi_{i-1}}\right| > \frac{\pi}{2}\right\}. \tag{27}$$

Writing the phase noise terms as a function of the corresponding in-phase and in-quadrature components, it can be shown that this condition is equivalent to a sign reversal of

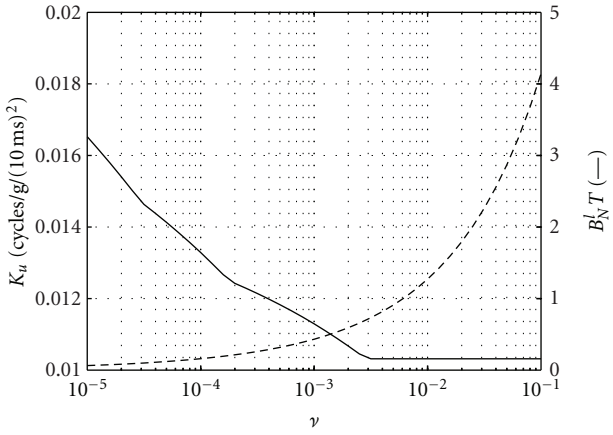


FIGURE 14: Noise bandwidth and peak error response for type 3 UFA-PLL.

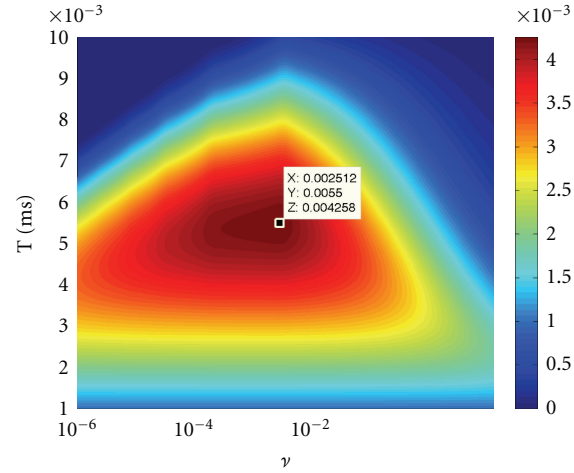


FIGURE 15: Function $f_{\text{UFA}}(\nu, T)$ for type 3 UFA-PLL and 20 g.

the noise component in phase with $\Delta\phi_{i-1}$, rather than in phase with $\hat{\phi}_{i-1}$. Therefore,

$$P_U = P\Big\{C + n_{I_i}'' + n_{I_{i-1}}'' + n_{I_i}' n_{I_{i-1}}' + n_{Q_i}' n_{Q_{i-1}}' < 0\Big\}, \tag{28}$$

where

$$n_{I_i}'' = n_{I_i}' \cos(\Delta\phi_{i-1}) - n_{Q_i}' \sin(\Delta\phi_{i-1}),$$
$$n''_{I_{i-1}} = n'_{I_{i-1}} \cos(\Delta\phi_i) - n_{Q_{i-1}}' \sin(\Delta\phi_i) \tag{29}$$

with $n_{I_i}' = D_i n_{I_i}\delta$, $n_{Q_i}' = D_i n_{Q_i}\delta$, $n_{I_{i-1}}' = D_{i-1} n_{I_{i-1}}\delta$, $n_{Q_{i-1}}' = D_{i-1} n_{Q_{i-1}}\delta$, $\delta = 1/\sqrt{TC/N_0}$ and $C = \cos(\Delta\phi_i - \Delta\phi_{i-1})$. The deterministic part of the argument of this cosine is a differenced version of the phase error transient, and the random part due to the output noise in the estimates is a differenced version of $n_{\hat{\phi}_i}$. Therefore, for the analysis of UFA-PLL two additional loop parameters are needed: the maximum difference of the error transient response, denoted by K_u, and the equivalent noise bandwidth of the linear model of the loop plus a differentiator, denoted by $B_N'T$. These quantities calculated by means of residues are plotted in Figure 14. The value K_u is constant for values of $\nu > 0.003$ because for this region the largest difference occurs between the two first samples of the transient response, which in turn are equal to the corresponding input samples due to the delays of the loop. Then, if the peak of the differenced transient occurs at $i = d$, we can write $\Delta\phi_d - \Delta\phi_{d-1} = K_u a T^2 + n_{\hat{\phi}_d} - n_{\hat{\phi}_{d-1}}$.

The quadratic terms in (28) have zero mean, and σ^4 variance and are uncorrelated between them and with the linear ones. For practical values of σ^2 their variance is much smaller than the variance of the linear terms. Even in this case, they cause the probability distribution of the sum in (28) to differ considerably from Gaussian. A more accurate calculation will require numerical computations of the actual distribution. On the contrary, our aim is to get reasonable and easy-to-handle approximation and, then, we will discard them, but being aware thatthe Gaussian assumption is only a coarse approximation. Then,

$$P_U \approx P\Big\{\cos\Big(K_u a T^2 + \Delta n_{\hat{\phi}_d}\Big) + n_{I_i}'' + n_{I_{i-1}}'' < 0\Big\} \tag{30}$$

where $\Delta n_{\hat{\phi}_d} = n_{\hat{\phi}_d} - n_{\hat{\phi}_{d-1}}$. The input noise terms are independent of each other and of the output ones because of both loop delays. Therefore, comparing (30) with (23) we replicate the reasoning for the PLL, but doubling the input noise contribution, and changing K_p by K_u and $B_N T$ by $B_N' T$. Then, using the same approximations made for (25), we get

$$P_U \approx Q\left(\sqrt{\frac{2TC/N_0 \cos^2(K_u a T^2)}{2 + 2B_N' T \sin^2(K_u a T^2)}}\right), \tag{31}$$

and then

$$f_{\text{UFA}}(\nu, T, a) = \frac{T\cos^2(K_u a T^2)}{1 + B_N'(\nu) T \sin^2(K_u a T^2)} \tag{32}$$

is the function to analyze which values of ν and T are better for the design of low tracking threshold loops. For accelerations of 20 g $f_{\text{UFA}}(\nu, T, a)$ is plotted in Figure 15 as an example. It can be seen that the larger values of f_{UFA} are found in the region of T near 5 ms and $10^{-4} < \nu < 10^{-2}$, and the maximum is approximately at $\nu = 0.0025$ and $T = 5.5$ ms. The value of f_{UFA} in this region is about 0.004. In this case, the optimum loop bandwidth is about $B_N = 70$ Hz, which is a more reasonable value than in the case of the PLL. For the UFA-PLL, the minimum pull-out probability and minimum output variance seem not to be as contradictory criteria as for the PLL. This can be understood noticing that the ability of the UFA-PLL for tracking in high dynamics depends on the smoothness of the transient error response rather than its absolute value, and then the output noise contribution becomes more relevant. Another important fact that must be emphasized is that even for 20 g accelerations it is not advisable to use correlation times T lower than 5 ms.

Notice that, as it was explained in Section 2, this probability analysis applies also for an FLL as long as the right $F(z)$ is used in the computations of K_u and $B_N' T$.

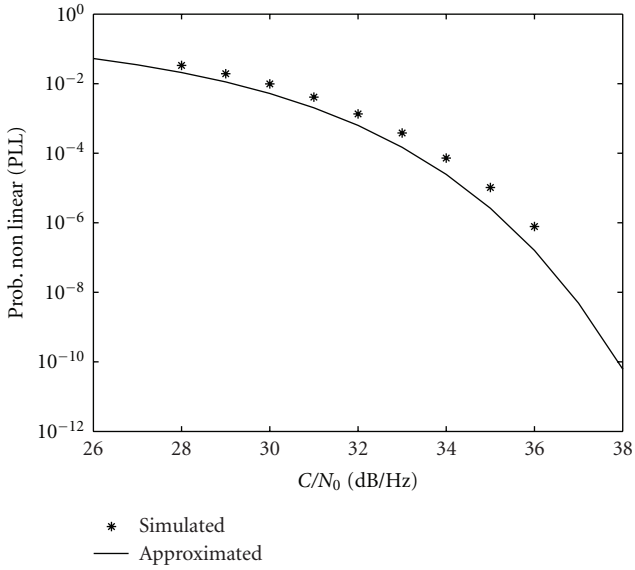


Figure 16: Probability of NL behavior for type 3 PLL and 5 g.

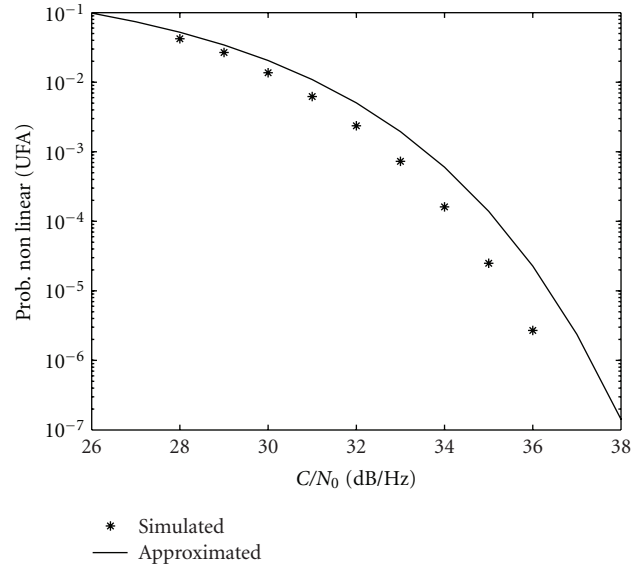


Figure 17: Probability of NL behavior for type 3 UFA-PLL and 20 g.

4.3. Simulations. In this section we assess the accuracy of the approximations made in the previous analysis. According to them the filter design of Section 3 is almost optimum when used in a UFA-PLL for tracking steps of 20 g, that is, it produces the minimum tracking threshold. The loop evolution with input noise according to a given C/N_0 was simulated until the transient peak instant. Runs entering the nonlinear region before this peak were discarded. A variable number of runs were used in order to reduce simulation time as much as possible but keeping statistical significance of the results. Specifically, 100,000 runs were enough for the lowest C/N_0 values, whereas 100 million had to be used for the highest.

A step of 5 g is considered in the simulation of the PLL, which produces a transient peak of $K_p 5\,\text{g}\,(5\,\text{ms})^2 = 1$ rad. Using (26) we found $f_{\text{PLL}} = 0.0033$. The results of the simulations compared with expression (25) are presented in Figure 16. It can be seen that the approximation is slightly optimistic and that the error is almost constant in the simulated range of C/N_0. For the UFA-PLL we adopt a step of 20 g that produces a transient peak of $K_p 20\,\text{g}\,(5\,\text{ms})^2 = 2$ rad and $K_u 20\,\text{g}\,(5\,\text{ms})^2 = 0.38$ rad of peak difference. Using (25), we get $f_{\text{UFA}} = 0.004$. The results of the simulations compared with expression (31) are shown in Figure 17. In this case the approximation is still acceptable for tracking threshold determination, but now it is pessimistic and the error grows for increasing values of C/N_0. This behavior is caused by the Gaussian approximation that neglects the quadratic noise terms in the probability expression (28).

4.4. Tracking Threshold Analysis. To illustrate how this analysis can lead to loop designs with lower tracking thresholds we consider the design of a type 3 UFA-PLL for 20 g acceleration steps. The tracking threshold is determined by a given probability of pull-out, and then we will define it by a given level of probability of starting a nonlinear behavior. Therefore, if P is the admissible pull-out probability and $C/N_0|_{\text{TH}}$ is the tracking threshold, then

$$P_U \approx Q\left(\sqrt{\frac{C}{N_0}\bigg|_{\text{TH}} f_{\text{UFA}}(\nu, T)}\right) = P, \tag{33}$$

or equivalently,

$$\frac{C}{N_0}\bigg|_{\text{TH}} \approx \frac{\left(Q^{-1}(P)\right)^2}{f_{\text{UFA}}(\nu, T)}. \tag{34}$$

Clearly, minimum tracking threshold values will be achieved when $f_{\text{UFA}}(\nu, T)$ is near its maximum, that is, using $\nu \approx 0.00025$ and $T \approx 5.5$ ms. Considering $\nu = 0.0003$ and $T = 5$ ms, for example, the loop parameters are $B_N = 80$ Hz, $B'_N = 51$ Hz $K_p a T^2 = 2$, $K_u a T^2 = 0.38$, and $f_{\text{UFA}} = 0.004$. Taking for instance a value of $P = 0.001$ and replacing in (34), we find that $C/N_0|_{\text{TH}} \approx 34$ dB/Hz.

Another design, similar to those based on analog prototypes, can use $\nu = 0.000001$ and $T = 2$ ms and then $B_N T = 0.1$. The loop parameters become $B_N = 50$ Hz, $B'_N = 13$ Hz $K_p a T^2 = 1.4$, $K_u a T^2 = 0.1$, and $f_{\text{UFA}} = 0.002$. Hence, for the same $P = 0.001$ if we replace in (34) the tracking threshold results $C/N_0|_{\text{TH}} \approx 37$ dB/Hz. Therefore, the use of the digital design method together with the proposed pull-out probability analysis can lower 3 dB the tracking threshold compared with traditional analog-based designs. An additional advantage is the use of longer values of T, requiring less computational load than analog designs.

It has to be mentioned that, even though the actual probability distribution can be different because of the Gaussian approximation, only the arguments of the $Q(\cdot)$ are used in the comparison of both designs. Therefore, the comparison is not affected by the Gaussian approximation, that is, a modified $Q(\cdot)$ function could be used for a more accurate probability calculation but the 3 dB threshold gain would remain.

5. Optimal Smoothing of the Phase Estimates

Due to the presence of two delays in the loop, the phase estimate obtained at a given instant is not computed with measurements up to this instant, but with measurements up to two previous instants. Thus, in the notation of [30] the loop phase estimate at the feedback branch is actually $\hat{\phi}[i \mid i-2] = \hat{\phi}_i$. Naturally, the use of "closer" measurements would produce a smoothing effect, that is, a better estimation. The real-time constraint does not allow taking advantage of this for the loop itself, but it is possible for other purposes as data detection and raw data generation for the navigation processes of the GNSS receivers. In this case, the optimal phase estimate can be obtained in the same way as before, but without forcing the two delays as in (9).

5.1. One Sample Smoothing. In this case the problem is equivalent to obtaining the optimal loop filter with only one delay. Then, if $F'(z)$ is the new loop-filter transfer function to be found, the corresponding closed loop transfer function is

$$T'(z) = \frac{F'(z)z^{-1}}{1 + F'(z)z^{-1}} = Y'(z)z^{-1}, \tag{35}$$

where $Y'(z)$ is the rational and stable transfer function to be found minimizing $J(Y'(z))$ in (8). Following the same optimization process done in the Appendix A, the result is

$$Y'(z) = \frac{X'(z)z}{\Psi(z)}, \tag{36}$$

where $\Psi(z)$ is the same minimum phase rational function of (11) and $X'(z)$ is the rational and stable transfer function obtained from

$$G'(z) = \frac{\gamma^2 \phi(z)\, \phi(z^{-1})}{\psi(z^{-1})} = X'(z) + W'(z^{-1}). \tag{37}$$

Noting that $G'(z) = G(z)/z$, it is simple to relate $X'(z)$ with $X(z)$ since the only change needed is to extract the possible pole in $z = 0$ of $W(z^{-1})/z$ to obtain $W'(z^{-1})$. Hence,

$$X'(z) = \frac{X(z)}{z} + \frac{(G(0) - X(0))}{z} \tag{38}$$

since $W(z^{-1}) = G(z) - X(z)$. For the case we are interested in, which is tracking of acceleration steps, according to (16) $G(0) = 0$. Then,

$$X'(z) = \frac{X(z) - X(0)}{z}, \tag{39}$$

Therefore, the new optimum close-loop transfer function of (36) is

$$Y'(z) = \frac{C + (A - 3C)z^{-1} + (3C - B)z^{-2}}{(1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1})}, \tag{40}$$

and the corresponding optimum loop filter is now only the three poles of the input,

$$F'(z) = \frac{C + (A - 3C)z^{-1} + (3C - B)z^{-2}}{(1 - z^{-1})^3}. \tag{41}$$

Even more interesting is the expression of (36) in terms of the cascade of accumulators,

$$Y'(z) = \frac{p_3 + (p_2 - p_3)(1 - z^{-1}) + (p_1 - p_2)(1 - z^{-1})^2}{(1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1})}, \tag{42}$$

If it could be possible to implement this loop filter, the feedback of the complete loop would be $\hat{\phi}[i \mid i-1]$. Of course $T'(z) = Y'(z)z^{-1}$ cannot be implemented with a real-time loop, which has two delays, but $Y'(z)z^{-2}$ can. Indeed, if the loop filter structure of the UFA-PLL is slightly modified as shown in Figure 18, it can be shown that amazingly $\hat{\phi}[i \mid i-1] = x_i^{(1)} - x_i^{(2)}$. Clearly, the delay on the feedback branch precludes the use of this value for the loop, but not for the rest of the GNSS receiver.

5.2. Two Samples Smoothing. The previous process can be applied again. Now, the problem is equivalent to obtaining the optimal loop filter without delay. Then, if $F''(z)$ is the loop-filter transfer function and the corresponding closed loop transfer function is

$$T''(z) = \frac{F''(z)z^{-1}}{1 + F''(z)} = Y''(z), \tag{43}$$

where $Y''(z)$ is the rational and stable transfer function to be found minimizing $J(Y''(z))$ in (8) and replicating (36)–(39), we obtain

$$X''(z) = \frac{X'(z) - X'(0)}{z}. \tag{44}$$

Therefore, the optimum transfer function of (43) is

$$Y''(z) = \frac{p_3 + P_2(1 - z^{-1}) + P_1(1 - z^{-1})^2}{(1 - z_1 z^{-1})(1 - z_2 z^{-1})(1 - z_3 z^{-1})}, \tag{45}$$

where $P_2 = (p_2 - 2p_3)$ and $P_1 = (p_1 - 2p_2 - p_3)$. Now $T''(z) = Y''(z)$ cannot be implemented with a real-time loop, but $Y''(z)z^{-2}$ can. Again, based on the loop structure of Figure 18 it can be shown that $\hat{\phi}[i \mid i] = \mathbf{x}_{i+1}^{(1)} - 2\mathbf{x}_{i+1}^{(2)} + \mathbf{x}_{i+1}^{(3)}$.

5.3. More Samples Smoothing. As it was previously mentioned, since $G(z)$ in (16) has four zeros at $z = 0$ for inputs modeled as accelerations steps, the previous smoothing procedure can be done two more times. In this way, if some latency is allowed, the following phase estimates can be obtained based only on the real-time tracking loop of Figure 18—the first three equations are repeated for clarity:

$$\begin{aligned}
\hat{\phi}[i \mid i-2] &= \mathbf{x}_{i-1}^{(1)}, \\
\hat{\phi}[i \mid i-1] &= \mathbf{x}_{i}^{(1)} - \mathbf{x}_{i}^{(2)}, \\
\hat{\phi}[i \mid i] &= \mathbf{x}_{i+1}^{(1)} - 2\mathbf{x}_{i+1}^{(2)} + \mathbf{x}_{i+1}^{(3)}, \\
\hat{\phi}[i \mid i+1] &= \mathbf{x}_{i+2}^{(1)} - 3\mathbf{x}_{i+2}^{(2)} + 3\mathbf{x}_{i+2}^{(3)}, \\
\hat{\phi}[i \mid i+2] &= \mathbf{x}_{i+3}^{(1)} - 4\mathbf{x}_{i+3}^{(2)} + 6\mathbf{x}_{i+3}^{(3)}.
\end{aligned} \tag{46}$$

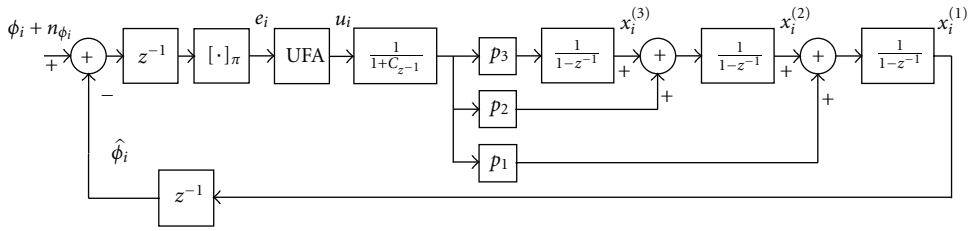


Figure 18: Block diagram of the UFA-PLL model with loop filter structure modified for phase smoothing.

These phase estimates can be interpreted as a prediction in the context of estimation theory [30]. In fact, $\hat{\phi}[i \mid i-2] = x_{i-1}^{(1)}$ can be obtained by propagating $\hat{\phi}[i-2 \mid i-2]$ with the signal dynamic model adopted for the phase estimation. If we consider $\mathbf{x}_i = [\mathbf{x}_i^{(1)}\ \mathbf{x}_i^{(2)}\ \mathbf{x}_i^{(3)}]$ as the state of the loop input phase model, the transition matrix must be

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}. \tag{47}$$

Since x_{i-1} include measurements up to the instant $(i-2)$, $\hat{\phi}[i-2 \mid i-2]$ can be found propagating backwards this state as $\{\mathbf{A}^{-2}x_{i-1}\}^{(1)}$. Notice that using this backward propagation process with the matrix $\mathbf{A}^{-1}$, all the estimators of (46) can be obtained. Actually, more smoothed estimates can be built. For example, the equation

$$\hat{\phi}[i \mid i+3] = \mathbf{x}_{i+4}^{(1)} - 5\mathbf{x}_{i+4}^{(2)} + 10\mathbf{x}_{i+4}^{(3)} \tag{48}$$

can be used, but it is not optimal. As it will be shown in the simulations it can be considered useful because of its extremely simple implementation. In some way, the quantity of zeros in (16) at $z = 0$ gives a measure of the backward propagation capacity of the states estimated by the loop filter.

5.4. Simulations. The simulated loop model is a UFA-PLL as shown in Figure 18 with the same filter coefficients of Section 3. The phase estimation error for an acceleration step of 50 g (starting at $i = 5$ and without noise) for the different estimators is plotted in Figure 19. In this situation the loop error grows up to almost one cycle and therefore the data detection during this transient will not be possible. However, applying the smoothing process described by (46) the transient error is consistently reduced each time that a new input sample is used for the estimation. The response of the suboptimal estimator of (48) is also shown. It can be seen that its transient response is slightly worse than the obtained with $\hat{\phi}[i \mid i+2]$.

The smoothing process also produces a decrease of the estimation noise variance. In Figure 20 the standard deviation of the six previous phase estimators is plotted for three different signal levels. These results were obtained simulating a linearized loop fed with Gaussian noise of variance $1/(2TC/N_0)$. As expected, an increase of 3 dB in the signal corresponds to a reduction of approximately $\sqrt{2}$ in the standard deviation. It is also possible to verify that

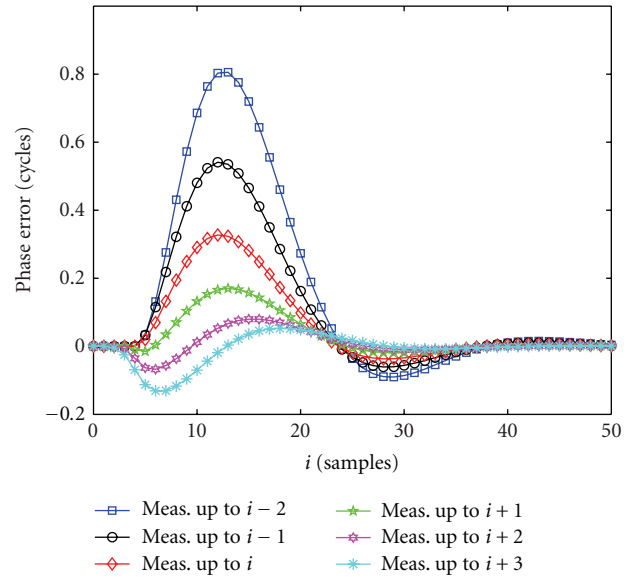


Figure 19: Phase estimation error during a step of 50 g.

the standard deviation for the loop output, that is, without smoothing, is equal to $\sigma_{\hat{\phi}_i} = \sqrt{B_N/(C/N_0)}$. This expression gives values 15.75°, 11.14°, and 7.88° when the values of C/N_0 are 30, 33, and 36 dB/Hz, respectively.

6. RF Test Experiments

The FLL-assisted PLL and the UFA-PLL designed in Section 3 were implemented in a System Developer Kit (SDK) for GPS receivers from SiRF [31]. Due to the real-time nature of this task all calculations for the loop had to be done in a fraction of 5 ms. They were programmed in fixed-point arithmetic, using some scaling and approximating coefficient values by powers of 2. Details of these implementations were given in [32]. To verify the tracking capability of the loops with real signals and without relying on expensive equipment like a GPS signal simulator, we used an RF signal generator to produce a frequency modulated carrier at 1575.42 MHz (the L1 GPS frequency). The signal was not spread with the code of a particular satellite, and thus the code generators of the GPS receiver were turned off during the test. This is not a limitation since the focus of this analysis is on the carrier loop, rather than the code loop. A triangular waveform was used as a frequency modulation to simulate steps in acceleration. The frequency deviation was selected according to the magnitude of the step (an instantaneous frequency

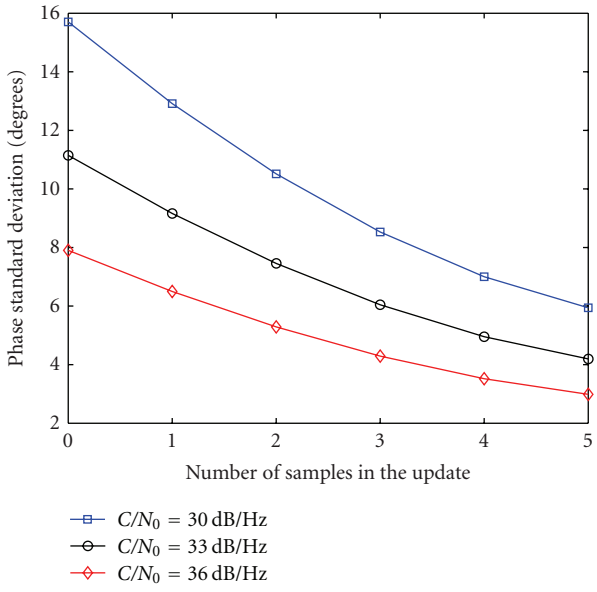


Figure 20: Phase estimation standard deviation.

deviation of Δf corresponds to a $\lambda \cdot \Delta f$ instantaneous velocity, where λ is the L1 wavelength). The selected carrier power was -113 dBm. Taking into account the noise of the 50 Ω output resistance of the generator and a noise figure of approximately 8 dB of the RF front-end gives a C/N_0 = 53 dB/Hz, which is a relatively high value for GNSS receivers. This value was selected to obtain low noise curves for a visual comparison with the simulated responses since the noise performance has been already characterized.

A test of 10 g acceleration steps is shown in Figure 21, depicting the tracked frequency detrended by a linear fit that accounts for the local clock drift. The amplitude of the triangular waveform was increased gradually up to the desired value to avoid large frequency steps. The measured phase error response of the FLL-assisted PLL at the output of the phase discriminator in one of the steps is shown in Figure 22, the response of the UFA-PLL is the same. The simulated response of the loop (as in Figure 8) is also displayed to appreciate that the implemented loop is properly characterized by our model.

The same experiment was performed for acceleration steps of 40 g. The phase error responses of the FLL-assisted PLL and the UFA-PLL are presented in Figures 23 and 24, respectively. In the case of Figure 23, the step presented is negative and therefore the phase response is upside down with respect to the simulation in Figure 9. Again, a fine agreement between the measurements with real laboratory generated signals and the simulations can be appreciated.

7. Conclusions

A new carrier tracking loop design method for real-time GNSS receivers has been presented, which is completely optimized from the perspective of the digital nature of the correlation measurements. An analysis of the phase and frequency discrimination ideas from this point of

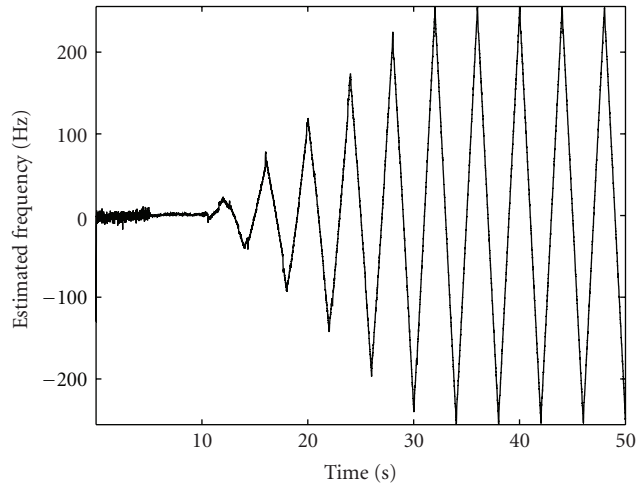


Figure 21: Tracked frequency during a 10 g test.

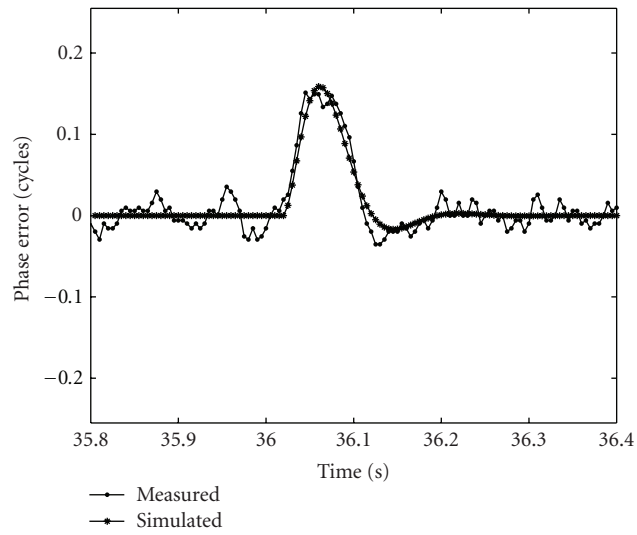


Figure 22: Measured phase error during a 10 g step.

view allowed us to choose optimum discriminators often discarded because of the complexity of their analog counterparts. Also, the known structure of FLL-assisted PLL has been considerably improved leading to a carrier loop that operates normally in phase locked condition and in frequency locked condition if the dynamics become severe enough. The effect of coupling the FLL to the PLL is considered at the design stage allowing a fine control of the effective loop bandwidth. Moreover, this approach allowed us to develop the UFA algorithm that corrects the cycle ambiguities of measured phase errors using the frequency information exploited by an FLL. With this algorithm it was possible to conceive a PLL that has the same advantages of an FLL-assisted PLL but avoids cycle slips and yet is easy to implement.

Regarding loop filters, their optimization was achieved directly in the digital domain. Our procedure solves the bandwidth trade-off considering the discrete nature of the filter and the unavoidable computational delays in the loops and makes possible to design loops extending beyond

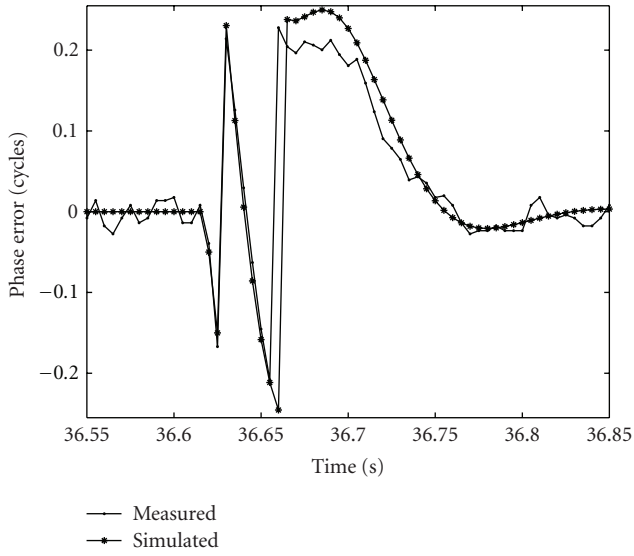


Figure 23: Phase error of FLL-assisted PLL during a 40 g step.

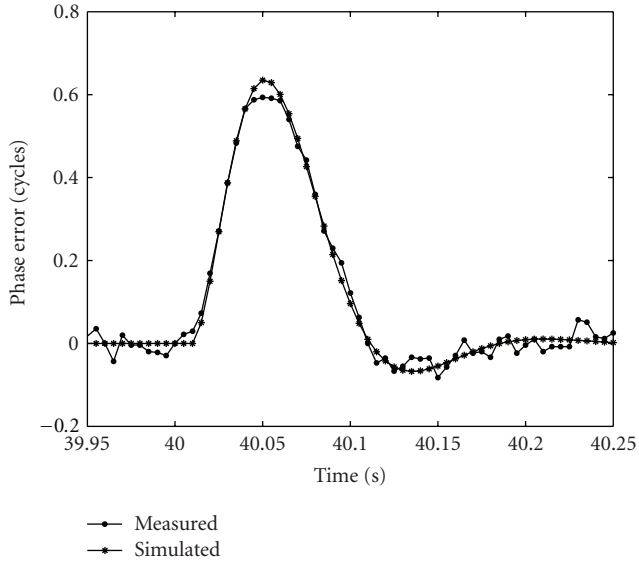


Figure 24: Phase error of UFA-PLL during a 40 g step.

the restrictive constraint $B_N T < 0.1$. This limitation is caused by the standard technique of discretizing analog loop filters, which only gives acceptable results for narrow loop designs, not suitable for high-dynamics receivers. Even more important, with our method the designed loop bandwidth is the same as the actually implemented one, since there are no approximations involved. This is of great importance considering the difficulties found in ([5] pp. 183) with respect to the stability of designed loops.

An approximate analysis of the nonlinear behavior of digital loops has been presented. The pull-out probability is approximated as an efficient tool for selecting the correlation time and the loop bandwidth so that the tracking threshold is minimized. Contrary to classical pull-out probability studies, our approach considers nonstationary scenarios as is often found in high-dynamics applications. The UFA-PLL examples presented show that 3 dB of improvement in the tracking threshold can be attained by properly selecting the integration time and loop bandwidth. It is worthwhile to emphasize that even for accelerations of 20 g, a loop sample time shorter than 5 ms is not appropriate.

It was also shown that using some state variables of the loop, smoothed phase estimates can be efficiently built with a latency of only a few samples. This can be very useful in many GNSS applications that use phase measurements, such as code measurement smoothing, differential positioning, and attitude estimation. It can also extend the use of these techniques to real-time and high-dynamics applications, where more complex phase estimation schemes are not practical and the usual tracking loop estimates do not provide enough phase accuracy. The simulations presented show that using the proposed smoothing scheme the transient responses of the loop to an acceleration step of 50 g can be almost eliminated and the estimation noise reduced by half.

Finally, our technique was used to design a loop that can optimally track steps of 20 g with a tracking threshold of $C/N_0 \approx 34$ dB/Hz. It was implemented in a GPS receiver using fixed-point arithmetic and tested with frequency modulated RF signals. Simulations and experimental results confirm that our new loop designs can track the input phase in these severe conditions, with the same implementation complexity as the usual loops.

This work has not considered adaptive schemes because they tend to increase the computational requirements and transient behavior. An adaptive version of the UFA-PLL could be formulated for situations where time-varying filters are affordable. Our assumptions do not deal with heavily unmodeled dynamics or disturbances on the tracked signals that justify a robust H_∞ approach. Nevertheless, given the Wiener filtering relation established for the present hypotheses, the loops obtained are equivalent to a steady-state version of Kalman-filter based-tracking loops [33]. Including robustness as one of the loop design considerations can also be addressed with the Wiener filtering formulation [27, 34], and a balance between robustness and adaptiveness will be pursued elsewhere.

Appendices

A. Minimizing $J(Y)$

Expression (8) for J can be minimized applying a standard procedure from variational calculus.

Assuming that the input noise is white with power spectral density $\eta/2$, the variance of the output noise can be calculated as

$$\sigma_N^2 = \frac{\eta}{2}\int_{-\pi}^{\pi} \left| Y\left(e^{j\omega}\right)\right|^2 \frac{d\omega}{2\pi} = \frac{\eta}{4\pi j}\oint Y(z)Y\left(z^{-1}\right)\frac{dz}{z}, \tag{A.1}$$

where $Y(e^{j\omega})$ is the frequency response of $Y(z)$ and the last integral extends over the unit circle of the complex plane.

Using Parseval's theorem, we find that

$$E_T(\phi_i) = \sum_{i=0}^{\infty} \Delta\phi_i^2 = \int_{-\pi}^{\pi} \left|\Delta\Phi\left(e^{j\omega}\right)\right|^2 \frac{d\omega}{2\pi} = \frac{1}{2\pi j} \oint \Phi(z)\Phi(z^{-1})[1 - Y(z)z^{-2}] \times [1 - Y(z^{-1})z^2]\frac{dz}{z}, \tag{A.2}$$

where $\Delta\Phi(z) = \Phi(z)[1 - Y(z)z^{-2}]$ is the z transform of $\Delta\phi_i$ and $\Phi(z)$ the z transform of ϕ_i. Then, replacing (A.1) and (A.2) in (8), we get the expression of the functional J with explicit dependence of $Y(z)$:

$$J(Y(z)) = \frac{\eta}{4\pi j} \oint \{Y(z)Y(z^{-1}) + \gamma^2\Phi(z)\Phi(z^{-1}) \times [1 - Y(z)z^{-2}][1 - Y(z^{-1})z^2] \quad \} \times \frac{dz}{z}, \tag{A.3}$$

where $\gamma^2 = 2\alpha^2/\eta$.

Let $Y(z)$ be the argument optimizing J and $y(z)$ any other stable causal rational transfer function:

$$\left.\frac{\partial J(Y(z) + \varepsilon y(z))}{\partial \varepsilon}\right|_{\varepsilon=0} = \frac{-\eta}{2\pi j} \oint \{\gamma^2\Phi(z)\Phi(z^{-1})z - Y(z)[1 + \gamma^2\Phi(z)\Phi(z^{-1})]z^{-1}\} \times y(z^{-1})dz, \tag{A.4}$$

where we used the following identity that holds for all $f : \mathbb{C} \to \mathbb{C}$ such that the integrals exist,

$$\oint f(z^{-1})\frac{dz}{z} = \oint f(z)\frac{dz}{z}. \tag{A.5}$$

The optimizing value $Y(z)$ has to produce a zero derivative (A.4) for every $y(z)$. To make the expression between braces zero would not assure to get a stable $Y(z)$. It is first necessary to make the following spectral decomposition:

$$\Psi(z)\Psi(z^{-1}) = 1 + \gamma^2\phi(z)\phi(z^{-1}). \tag{A.6}$$

This is always possible because the term on the right has an even number of poles and zeros. And besides, poles as well as zeros can be separated in two sets. If z_i belongs to a set, $1/z_i$ belongs to the other. Hence, if z_i is within the unit circle, there is also $1/z_i$ outside the the unit circle. Therefore, a rational, minimum phase (all zeros inside the unit circle) and stable $\Psi(z)$ can be found. Then, the required $Y(z)$ has to make zero the following integral:

$$\oint \left\{\frac{\gamma^2\Phi(z)\Phi(z^{-1})z}{\Psi(z^{-1})} - \frac{G(z)\Psi(z)}{z}\right\} g(z^{-1})\Psi(z^{-1})dz. \tag{A.7}$$

Finally, we have to split the first term between braces in (A.7) in two, separating its partial fraction expansion in a part with all poles inside the unit circle $X(z)$ and a part with all poles outside it $W(z^{-1})$, that is,

$$G(z) = \frac{\gamma^2\Phi(z)\Phi(z^{-1})z}{\Psi(z^{-1})} = X(z) + W(z^{-1}), \tag{A.8}$$

and note that

$$\oint W(z^{-1})\, y(z^{-1})\Psi(z^{-1})dz = 0 \tag{A.9}$$

because the integrand is analytic inside the unit circle.

Therefore, the optimum transfer function is given by (10), where $\Psi(z)$ and $X(z)$ can be obtained from (11) and (12), respectively. Notice the important role of asking $\Psi(z)$ to be of minimum phase: since it will be inverted in (10) its zeros become poles of $Y(z)$ that has to be causal and stable.

B. Relation to Wiener Filtering

The Wiener filtering is usually posed as minimizing the variance of the estimation error when a measured signal ω_i related to another one ϕ_i is used to produce a linear prediction $\hat{\phi}[i + m|i]$ of ϕ_i. In our case, $\omega_i = \phi_i + n_{\phi i}$. The general solution in terms of spectral densities is given for instance in [26],

$$Y(z) = \left\{\frac{S_{\phi\omega}(z)z^m}{H(z^{-1})}\right\}_+ H(z), \tag{B.1}$$

where $\{\cdot\}_+$ means the part of the partial fraction expansion with poles inside the unit circle and H is the whitening filter that produces a white sequence at its output when ω_i is the input. This solution looks simple, but it does not give yet explicit equations for the coefficients of the causal and stable rational transfer functions needed.

The so-called polynomial framework [34] permits a systematization of this issue. Assume the signal part of the phase is modeled as $\phi_i = (C/D)b_i$ where C and D are polynomials, with C being an abbreviated notation for $C(q) = c_0 + c_1q^{-1} + c_2q^{-2} + \ldots$, and the operator q^{-1} represents the delay operator such that $q^{-1}\phi_i = \phi_{i-1}$. The signal b is a white noise sequence of variance σ_b^2. The noise contaminating the measurements is also modeled linearly as $w_i = (M/N)v_i$ where M and N are polynomials and v is a white noise sequence of variance σ_v^2, independent of b.

The polynomial solution is given by

$$Y = \frac{Q_1N}{\beta}, \tag{B.2}$$

where $\beta(q) = \beta_0 + \beta_1q^{-1} + \beta_2q^{-1} + \cdots$ is a monic polynomial ($\beta_0 = 1$) obtained from the following factorization problem:

$$\sigma_\varepsilon^2\beta\beta_* = \sigma_b^2CC_*NN_* + \sigma_v^2MM_*DD_*. \tag{B.3}$$

The notation β_* means $\beta_*(q^{-1}) = \beta_0^* + \beta_1^*q + \beta_2^*q^2 + \cdots$, with β_i^* being the complex conjugate of β_i. In order to obtain Q_1, the Diophantine equation, with $r^2 = \sigma_\varepsilon^2/\sigma_b^2$,

$$q^mCC_*N_* = r^2\beta_*Q_1 + qDL_* \tag{B.4}$$

must be solved for Q_1 and L_*. Equivalently, (B.4) is the same as doing the partial fraction expansion of

$$q^{m-1}\frac{CC_*N_*}{r^2D\beta_*} = \frac{Q_1}{qD} + \frac{L_*}{r^2\beta_*}. \tag{B.5}$$

Notice that if we enforce $\Phi(z) = C(z)/D(z)$, $m = 2$, $w[n] = v[n]$, that is, $M = 1 = N$, $\sigma_v^2 = \eta/(2T)$, and $\sigma_b^2 = \gamma^2\sigma_v^2$; then (11) is the same as (B.3) and $\Psi(z) = \gamma r\beta(z)/D(z)$. Moreover, (12) is similar to (B.5), and thus $\gamma rQ_1(z)/zD(z) = X(z)$. We can see how both solutions match.

Acknowledgment

This work was funded by ANPCyT PICT 00535, UNLP 11-I-127, and CIC-PBA.

References

[1] W. J. Hurd, J. I. Statman, and V. A. Vilnrotter, "High dynamic GPS receiver using maximum likelihood estimation and frequency tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 23, no. 4, pp. 425–437, 1987.

[2] P.W. Ward, "Performance comparisons between FLL, PLL and a novel FLL-assisted-PLL carrier tracking loop under RF interference conditions," in *Proceedings of the 11th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GPS '98)*, Nashville, Tenn, USA, September 1998.

[3] F. M. Gardner, *Phaselock Techniques*, John Wiley and Sons, New York, NY, USA, 1979.

[4] C. R. Cahn, D. K. Leimer, C. L. Marsh, F. J. Huntowski, and G. D. Larue, "Software implementation of a PN spread spectrum receiver to accommodate dynamics," *IEEE Transactions on Communications*, vol. 25, no. 8, pp. 832–840, 1977.

[5] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and Applications*, Artech House, Boston, Mass, USA, 2nd edition, 2006.

[6] S. B. Son, I. K. Kim, S. H. Oh, S. H. Kim, and Y. B. Kim, "Commercial GPS receiver design for high dymanic launching vehicles," in *Proceedings of the International Symposium on GNSS/GPS*, Sydney, Australia, December 2004.

[7] P. V. Brennan, "Performance of phase-locked loop frequency synthesiser using accumulative phase detector ," *IEE Proceedings of Circuits, Devices and Systems*, vol. 143, no. 5, pp. 249–254, 1996.

[8] R. C. Den Dulk, "Digital fast acquisition method for phase-lock loops," *Electronics Letters*, vol. 24, no. 17, pp. 1079–1080, 1988.

[9] D. Weinfeld and I. Bar-David, "Phase locked loop with extended range phase detector," in *Proceedings of the 18th Convention of Electrical and Electronics Engineers in Israel*, Tel Aviv, Israel, March 1995.

[10] R. Jaffe and E. Rechtin, "Design and performance of phase-lock circuits capable of near-optimun performance over a wide range of inputs signal and noise levels," *IEEE Transaction on Information Theory*, vol. 76, no. 1, pp. 66–76, 1955.

[11] W. C. Lindsey and C. M. Chie, "Survey of digital phase-locked loops," *Proceedings of the IEEE*, vol. 69, no. 4, pp. 410–431, 1981.

[12] B. W. Parkinson and J. J. Spilker, Eds., *Global Positioning System: Theory and Applications*, American Institute of Aeronautics and Astronautics (AIAA), Washington. DC, USA, 1996.

[13] S. A. Stephens and J. B. Thomas, "Controlled-root formulation for digital phase-locked loops," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 1, pp. 78–95, 1995.

[14] S.C. Gupta, "On digital phase-locked loops," *IEEE Transactions on Communication Technology*, vol. 16, no. 2, pp. 340–344, 1968.

[15] S. Hinedi and J. I. Statman, "Digital accumulators in phase and frequency tracking loops," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 1, pp. 169–180, 1990.

[16] P. A. Roncagliolo and J. G. Garcia, "High dynamics and false lock resistant GNSS Carrier Tracking Loops," in *Proceedings of the 20th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS '07)*, Fort Worth, Tex, USA, September 2007.

[17] W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1973.

[18] H. Meyr and G. Ascheid, *Phase-, Frequency-Locked Loops, and Amplitude Control*, vol. 1 of *Synchronization in Digital Communication*, John Wiley and Sons, New York, NY, USA, 1990.

[19] P.A. Roncagliolo, J.G. Garcia, and C.H. Muravchik, "Pull-out probability considerations in high dynamics GNSS tracking loops design," in *Proceedings of the 10th International Symposium on Spread Spectrum Techniques and Applications (ISSSTA '08)*, Bologna, Italy, August 2008.

[20] F. D. Nunes, J. M. N. Leitão, and F. M. G. Sousa, "Nonlinear filtering in GNSS pseudorange dynamics estimation combining code delay and carrier phase," *IEEE Journal on Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 639–650, 2009.

[21] H. K. Lee and C. Rizos, "Position-domain hatch filter for kinematic differential GPS/GNSS," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 30–40, 2008.

[22] P. A. Roncagliolo, J. G. García, P. I. Mercader, D. R. Fuhrmann, and C. H. Muravchik, "Maximum-likelihood attitude estimation using GPS signals," *Digital Signal Processing: A Review Journal*, vol. 17, no. 6, pp. 1089–1100, 2007.

[23] S. Lo, A. Chen, P. Enge et al., "GNSS album. Image and spectral signatures of the new GNSS signals," *Inside GNSS*, pp. 46–56, 2006.

[24] V. I. Tikhonov, "The effect of noise on phase-lock oscillator operation," *Automat. i Tekmekh.*, vol. 22, no. 9, 1959.

[25] J. A. Areta, P. A. Roncagliolo, and C. H. Muravchik, "Sincronización digital de señales de espectro expandido con perturbaciones de alta dińamica," in *Proceedings of the 10th Reunión de trabajo en Procesamiento de la Información y Control*, pp. 762–767, Buenos Aires, Argentina, September 2003.

[26] T. Kailath, *Lectures on Wiener and Kalman Filterings*, Springer, New York, NY, USA, 1981.

[27] M. Sternad and A. Ahlén, "Robust filtering and feedforward control based on probabilistic descriptions of model errors," *Automatica*, vol. 29, no. 3, pp. 661–679, 1993.

[28] R. H. Milocco and C. H. Muravchik, "H2 optimal linear robust sampled-data filtering design using polynomial approach," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1816–1824, 2003.

[29] P. A. Roncagliolo, C. E. DeBlasis, and C. H. Muravchik, "GPS digital tracking loops design for high dynamic launching vehicles," in *Proceedings of the 9th International Symposium on Spread Spectrum Techniques and Applications (ISSSTA '06)*, Manaos, Brazil, August 2006.

[30] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, NJ, USA, 1993.

[31] "SiRF technology," http://www.sirf.com/.
[32] C. E. De Blasis, "Seguimiento de señales de GPS," Electronic Engineering Graduation Project, UNLP, Argentina, 2005.
[33] D. Simon, *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*, John Wiley and Sons, Hoboken, NJ, USA, 2006.
[34] A. Ahlen and M. Sternad, "Wiener filter design using polynomial equations," *IEEE Transactions on Signal Processing*, vol. 39, no. 11, pp. 2387–2399, 1991.

Hindawi Publishing Corporation
International Journal of Navigation and Observation
Volume 2012, Article ID 324858, 12 pages
doi:10.1155/2012/324858

Research Article

A Novel Quasi-Open Loop Architecture for GNSS Carrier Recovery Systems

Muhammad Tahir,[1] Letizia Lo Presti,[1] and Maurizio Fantino[2]

[1] *Dipartimento di Elettronica e Telecomunicazioni (DET), Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*
[2] *Navigation Laboratory, Istituto Superiore Mario Boella, Via P.C. Boggio 61, 10138 Torino, Italy*

Correspondence should be addressed to Muhammad Tahir, muhammad.tahir@polito.it

Received 16 December 2011; Revised 15 March 2012; Accepted 5 April 2012

Academic Editor: Carles Fernández-Prades

The problem of designing robust systems to track global navigation satellite system (GNSS) signals in harsh environments has gained high attention. The classical closed loop architectures, such as phase locked loops, have been used for many years for tracking, but in challenging applications their design procedure becomes intricate. This paper proposes and demonstrates the use of a quasi-open loop architecture to estimate the time varying carrier frequency of GNSS signals. Simulation results show that this scheme provides an additional degree of freedom to the design of the whole architecture. In particular, this additional degree of freedom eases the design of the loop filter in harsh environments.

1. Introduction

In global navigation satellite systems (GNSSs) the relative motion of both GNSS satellites and the user causes a Doppler effect, which results in a large frequency shift in the carrier and in the code of the received signal [1]. Precise estimation of this frequency shift is one of the most demanding requirements for GNSS receivers, because only an accurate tracking of the carrier frequency and Doppler shift allows the receiver to work properly, enabling reliable estimates of position velocity and timing (PVT). In any GNSS receiver, the acquisition stage provides an initial coarse estimation of the frequency shift, which is subsequently refined by the tracking systems. They are generally implemented in the form of closed loops, that is, phase lock loops (PLLs) and frequency lock loops (FLL), which track respectively the phase and the frequency of the incoming carrier, [2].

The main building block of a closed loop architecture is the loop filter. The design of a loop filter has been extensively addressed in the literature regarding the continuous-time PLLs, and many results and methods exist for different scenarios. However, modern receivers work in the discrete-time domain, and so PLLs and FLLs are digital systems, whose loop filters are often designed starting from some equivalent analog prototypes, by adopting transformation techniques from the analog to the digital domain. These tracking loops are therefore de facto digital approximations of analog loops, whose quality breaks down as the integration time increases. A valid assumption for this approximation is that the product BT between the loop noise bandwidth B, and the integration time T remains close to zero. As this product increases, the loop becomes unstable, as discussed in [3]. However, in high dynamic and weak signal applications, it is necessary to work with large BT values. In these cases the design of the loop filters based on analog-to-digital transformations does not work properly. Therefore, other techniques have been proposed in literature, which are more robust when low update rates (long integration times) or large bandwidths are required. They are the controlled root method for the design of digital filters [4], the direct design of loop filters in digital domain by minimization criterion [5], the loop architectures based on Kalman filters [6], and the fuzzy loop architectures [7].

In this paper we propose an alternative solution, based on a novel quasi-open loop architecture, which relieves the stress on the loop filter in terms of stability. In the first step of our study we considered open loop techniques, since some advantages over the closed loop counterpart

schemes can be achieved. For example, it is a known fact that PLLs are vulnerable to fading effects, typically associated with urban environments, and cycle slips [8]. Moreover, because of their closed loop structure, they need a long acquisition time before attaining the loop lock, and this may be a serious drawback when the Doppler significantly varies within this time interval. The use of open loop architectures can solve these problems. Several open loop architectures have been proposed in literature; most of them are based on the use of the Fast Fourier Transform (FFT) for the estimation of the frequency error between the incoming carrier and a local carrier replica. However, this block processing structure, typical of FFT-based methods, increases the system complexity. In our scheme we do not use FFT for frequency estimation; in our method we estimate the frequency by using conventional PLL/FLL discriminators, for example, the Kay's estimator [9], but with a different update of the frequency of the numerically controlled oscillator (NCO). In particular, instead of updating the NCO at each coherent integration time as in the closed loop architectures, the update is performed after NT integrations intervals, where N is an integer. Between two updating epochs the architecture works as an open loop. Using this approach we are actually working with a three-rate scheme, unlike the conventional two-rate closed loop systems. We have found that this additional degree of freedom can help us to ease the design of the loop filter in weak signal conditions, where it is necessary to extend the coherent integration time.

In a weak signal scenario the extension of the coherent integration time is the only possible option, since the increased processing gain allows the successful recovery of weak and extremely weak signals. However, the maximum coherent integration time in a GNSS receiver is limited by a variety of factors: the presence of navigation data modulation, the stability issues, and the demodulation losses due to a frequency mismatch. This effect is particularly detrimental when the loop update time increases (long integration time) and the incoming carrier has a high Doppler shift [3]. The first problem is usually solved by estimating the navigation data or using external assistance. The main contribution of this paper is related to the second issue, that is, the stability problem, which is solved by adopting an inherently stable architecture. We will show that the proposed architecture can work with extended integration times, which are not possible in the traditional closed loop schemes under the same conditions. In order to solve the third problem, methods of compensation of the user dynamics are necessary. For this purpose, PLLs are generally assisted by other systems, for example, an FLL or a tightly coupled inertial navigation system (INS).

The paper is organized as follows. The GNSS signal model is introduced in Section 2. Then, a general overview of the traditional GNSS carrier recovery systems is presented in Section 3, stressing the aspects which motivate the scheme proposed in this paper. Section 4 shows the novel quasi-open loop scheme for carrier recovery, proposed to overcome some limitations of the traditional schemes. Some simulation results are presented in Section 5, showing the feasibility of the proposed method in terms of tracking jitter performance, in a dynamic scenario and with weak signals. On the basis of these results some possibilities to refine the proposed architecture are also discussed. Conclusions are drawn in Section 6.

2. Signal Model

The main purpose of this paper is to propose a novel architecture for tracking the frequency of the incoming GNSS signal which combines the good properties of both open loop and closed loop architectures. These systems recover the carrier from the received signal, which can be written, after downconversion to intermediate frequency (IF), sampling and quantization, as

$$x_w[n] = \sqrt{2P}a[n]\cos[2\pi(f_0 + f_d(nT_s))nT_s + \psi_s] + w[n], \tag{1}$$

where P is the total received power of the useful signal, $a[n] = d[n]c[n]$ is the useful signal ($c[n]$ is the code and $d[n]$ the navigation message), f_0 is an intermediate frequency, $f_d(nT_s)$ is a frequency shift (which can be time variant), ψ_s is the initial phase (the phase for $n = 0$), and $w[n]$ is a noise component. This signal is obtained by sampling an analogical signal at a sampling frequency $f_s = 1/T_s$. The carrier signal is completely characterized by its *instantaneous* phase

$$\varphi_s(nT_s) = 2\pi(f_0 + f_d(nT_s))nT_s + \psi_s. \tag{2}$$

The noise term $w[n]$ is a realization of a Gaussian random process $W[n]$, with flat power spectral density $N_o/2$ over the receiver band B_r, and with power $\sigma_w^2 = N_o B_r$. $W[n]$ is not generally a white sequence with a flat power spectral density, due to front-end filtering. However, a white model is justified because the bandwidth of the front-end filter is usually close to the Nyquist sampling frequency, that is, $B_r = f_s/2$.

The purpose of the carrier recovery is to estimate $\varphi_s(nT_s)$ in order to construct a local oscillator (LO) of the type

$$x_{\mathrm{cr}}[n] = \cos\widehat{\varphi}_s(nT_s), \tag{3}$$

where $\widehat{\varphi}_s(nT_s)$ is the estimate of $\varphi_s(nT_s)$, and the subscript cr denotes that this signal is recovered from the carrier. Therefore, the LO (producing $x_{\mathrm{cr}}[n]$) can be considered the output of the carrier recovery system. We assume that both data and code can be wiped off and we focus on the operations performed by the carrier recovery system by considering the ideal signal

$$x_s[n] = x_s(nT_s) = \cos[2\pi(f_0 + f_d(nT_s))nT_s + \psi_s], \tag{4}$$

where $f_d(nT_s)$ is an unknown frequency, which varies with an unknown rule, ψ_s is also unknown, while f_0 is a known nominal frequency. In real cases, the signal $x_s(nT_s)$ is also affected by noise, and then the carrier has to be recovered from a noisy version of (4).

3. Traditional Tracking Architectures: Review and Limitations

The traditional GNSS carrier recovery systems can be classified into two categories: closed loop tracking systems

and open loop phase and frequency estimators. In this section we briefly review the traditional closed loop systems and the open loop schemes, paying attention to the aspects which motivate the quasi-open loop scheme proposed in the next sections.

3.1. Closed Loop Architectures. Digital phase lock loops (DPLLs) and FLLs lie into the category of the classical closed loop architectures, whose basic scheme is shown in Figure 1. DPLLs are able to track both the instantaneous phase and frequency of the incoming carrier, while FLLs are only able to track the carrier frequency.

In the following we briefly describe the DPLL operations (the operations performed by FLLs are almost similar except for the discriminator). The estimation of the instantaneous phase is generally performed by a DPLL, which consists of a number of subsystems: (a) the phase estimator/discriminator, (b) the loop filters $F(z)$, (c) the local oscillator (LO) also called numerically controlled oscillator (NCO). The input signal $x_w[n]$ is first multiplied by the local carrier $x_L[n]$ generated by the NCO, and integrated by the integrate and dump block. During this integration process, L input samples are processed and used to produce the prompt correlator output $y_\phi[k]$. The corresponding integration time is $T = LT_s$. At this point the sampling rate of the system changes from $1/T_s$ (time domain n) to $1/(LT_s)$ (time domain k). The instantaneous phase is estimated from $y_\phi[k]$ by the phase discriminator. This estimate is filtered through the loop filter $F(z)$, with a loop noise bandwidth B, and the filter output is used to drive the LO for the carrier generation. This estimate is then progressively updated using the information provided by the new correlator output in a closed loop manner. Below we present two key points that motivate the use of a quasi-open loop scheme (for GNSS carrier recovery systems) rather than a closed loop one.

It is important to stress that, in the traditional closed loop carrier recovery systems, the LO role is twofold: in fact it is both the output of the system and an integral part of the phase estimator. In the quasi-open loop structure proposed in this paper these two roles will be decoupled, as explained in Section 4. This is the first key point of our method, as will be clear in the next sections.

The loop filter $F(z)$ is the most critical block, whose function is also twofold. Firstly, as the received signal and, thus, the discriminator output is corrupted by thermal noise, the filter is required to provide a degree of noise rejection. Secondly, it enables the processing of higher order dynamics. Transformation methods from the Laplace domain to the Z-domain are widely used to design loop filters [2]. These methods simply provide a discrete version of the loop filters that have been previously designed for the analog loops. However, this approach neglects both the inherent delay in the digital loop and the variation in the open loop gain due to the NCO update interval. In [3] it has been shown that the transformation methods properly work only if BT is close to zero, where B is the loop noise bandwidth and T is the integration time. As BT increases, the effective loop noise bandwidth and the closed loop pole locations deviate from the desired ones and eventually the loop becomes unstable, as explained in [3]. The maximum achievable BT value depends on the type of the Laplace-Z transformation and on the characteristics of the original continuous-time filter. For most communication applications, this condition is satisfied because BT remains close to zero. Instead, for some new GNSS applications, such as for weak signal tracking and extremely high dynamic applications, larger BT values are required.

Simulation experiments have been carried out to analyze the behavior of an FLL for increasing values of BT. The incoming GNSS signal has been generated with a carrier to noise ratio $C/N_o = 40$ dB-Hz and with a ramp-type time varying frequency with a slope equal to 10 Hz/s. In order to track this frequency evolution, we have used a 2nd order FLL with a loop noise bandwidth $B = 10$ Hz, and we have changed BT (by modifying the integration time) to put progressively more stress on the loop. The frequency estimated by the FLL is shown in Figure 2 for three different values of BT, together with an estimate of C/N_o. The latter can be used as an indicator to check if the system is in lock state or not. These results show that the loop loses the lock and the FLL is no longer able to track the input frequency when $BT = 0.3$, as indicated by the estimated C/N_o. It is important to emphasize that the loop filter is not the only module in the closed loop scheme responsible for the loss of lock and the stability problems. The loop can also lose lock or can become unstable because of other factors, but here we concentrate on the effect of the loop filter design based on transformation methods on loop. To solve this problem, some techniques for designing the loop filters have been proposed in the literature, for example, the controlled root method, the direct design of loop filters in digital domain based on a minimization criterion, the loop architectures based on Kalman filters, and the fuzzy loop architectures. An alternative approach is to use the scheme proposed in Section 4, which simplifies the design of the loop filter. This is the second key point of our method.

3.2. Open Loop Architectures. In order to estimate the time delay and the carrier frequency from the incoming signal, the open loop schemes usually operate on batches of the incoming signal, as depicted in Figure 3. The open loop approach does not separate acquisition and tracking stages, as explained in [10]. An input signal batch is correlated with batches of a signal replica in order to obtain an entire 3-D image of the signal, whose dimensions are the code shift, the Doppler shift, and the signal energy. This batch-based correlation uses joint time-frequency domain techniques to allow some forms of parallel computing based on FFT. After the 3-D image has been obtained, batch estimators are applied to the 3-D function to compute the signal parameters. They operate by searching for the location of the maximum energy of the 3-D function and provide an estimate of the parameters $\Delta\psi$, $\Delta f[k]$, and the code delay corresponding to this location. Since these methods mainly rely on FFT-based correlators for the 3-D image computation, they usually exhibit a large computational complexity, which enormously increases as the batch size

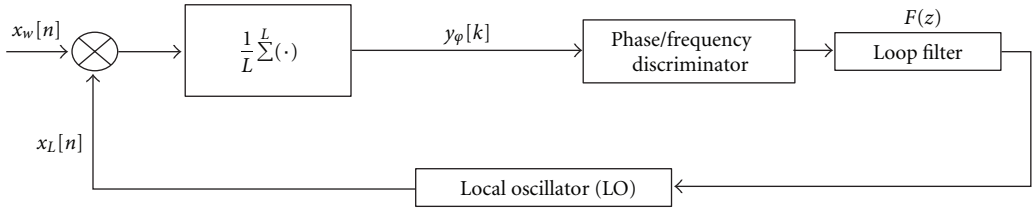
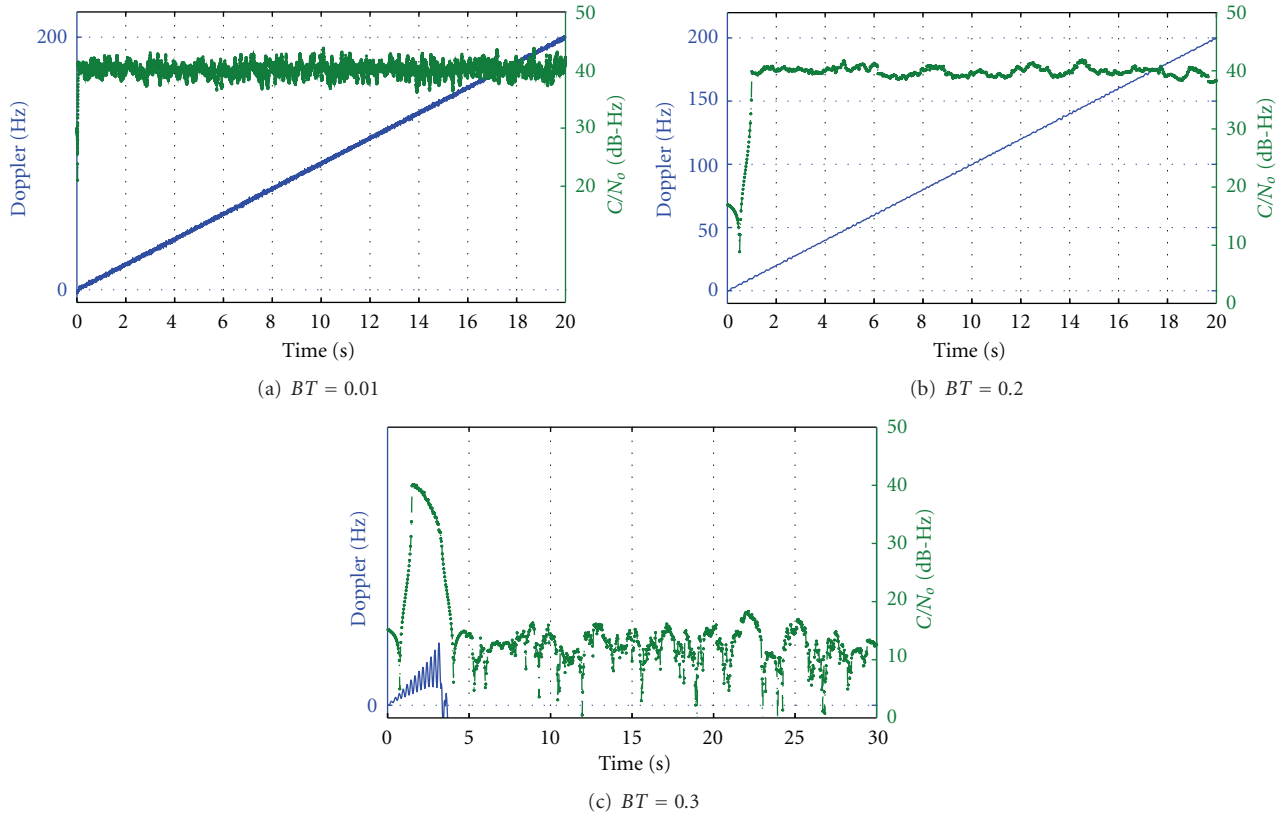


Figure 1: Basic classical closed loop carrier tracking architecture.



(a) $BT = 0.01$

(b) $BT = 0.2$

(c) $BT = 0.3$

Figure 2: 2nd order FLL tracking tesults for input ramp frequency of 10 Hz/s, $C/N_o = 40$ dB-Hz, $B = 10$ Hz and different integration Times.

increases. This is the main reason to avoid the use of these open loop processing techniques in real time receivers.

4. Proposed Quasi-Open Loop Architecture

In this section we propose a quasi-open loop scheme for carrier recovery in a GNSS receiver. We start our discussion from a basic open loop scheme, which utilizes a conventional discriminator. Then, after analyzing the behavior of the discriminator for the frequency estimation, we propose a quasi-open loop scheme for the continuous tracking of the incoming carrier frequency.

4.1. Basic Open Loop Scheme Utilizing Conventional Discriminators. The main idea of an open loop carrier recovery scheme that utilizes a conventional discriminator is depicted in Figure 4.

The main blocks of this scheme are the following.

(i) The local oscillator which generates the local signal $x_L[n]$. This complex LO is denoted by LO(E), to emphasize its role in the process of phase estimation.

(ii) The mixer which performs the multiplication between the local oscillator and the incoming signal.

(iii) The accumulator which accumulates L values of the signal $y[n]$; this block is often called *integrator* as it is equivalent to an integral in the continuous time domain. The integrator output is provided at the epochs $n = L, 2L, \ldots, kL, \ldots$.

(iv) The discriminator, which provides an estimate of the instantaneous phase/frequency of the incoming signal. It receives an input (the integrator output) at a rate which depends on the value of L.

(v) The open loop smoothing filter.

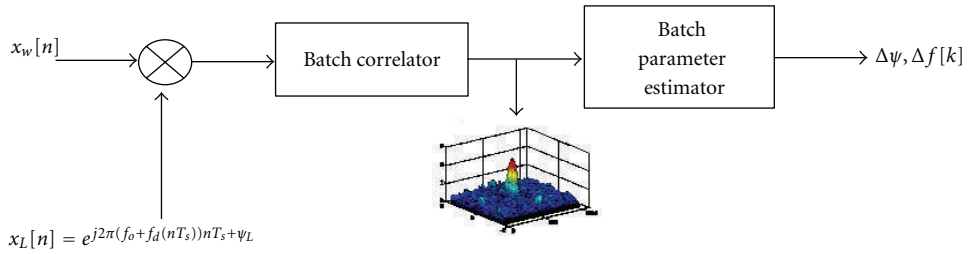


Figure 3: a generic open loop GNSS receiver.

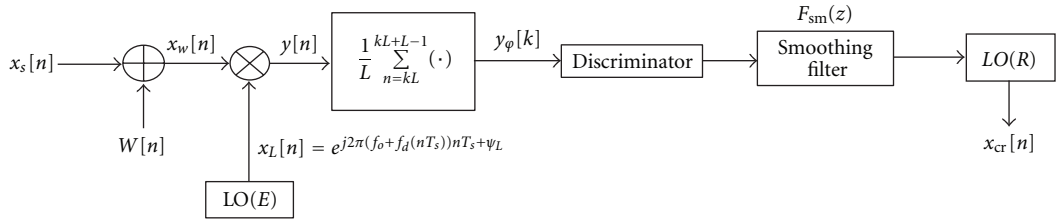


Figure 4: A generic open loop carrier recovery scheme utilizing conventional discriminators.

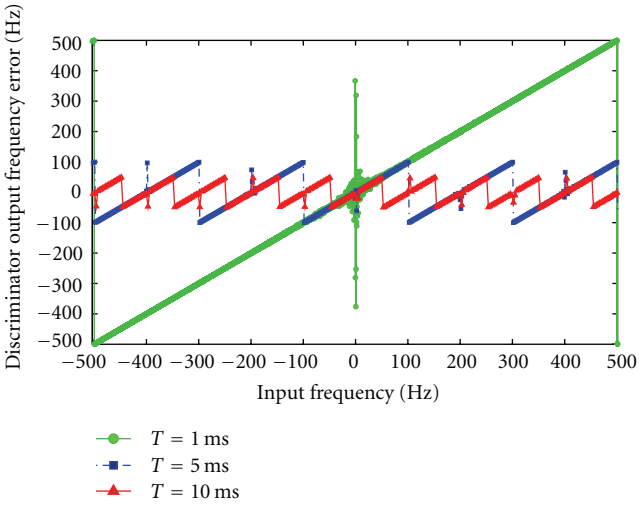


Figure 5: Frequency discriminator for different integration times.

(vi) The local oscillator LO(R), which provides a local carrier $x_{cr}[n]$, recovered from the incoming signal.

This scheme differs from its closed loop counterpart since we use each new phase/frequency estimate to update a separate local oscillator (the LO(R) block), instead of feeding back the local oscillator LO(E) that is embedded in the estimation module. This system works at two different rates, that is, in two different discrete-time domains: the mixer and the integrator works in the n domain with a rate r_n, while the discriminator works in the k domain, with a rate $r_k = r_n/L$. Once the discriminator has estimated the instantaneous phase/frequency of the incoming signal, the carrier can be recovered by building a sinusoidal signal, $x_{cr}[n]$. This is the recovered LO, denoted as LO(R), whose update rate is r_k. In a DPLL, the LO(R) is also used to update the signal $x_L[n]$, then LO(E) and LO(R) coincide.

In our scheme we decouple LO(R) and LO(E), so as to design independently their denoising filters and update rates. This idea derives from the consideration that the update rate of LO(R) depends on the input dynamic, while the update rate of LO(E) depends on the frequency range the discriminator is able to process. It is evident that the two update rates are related, but the requirements can be different. Similar considerations can be done for the denoising filters.

Below we perform an analysis of the integrator and discriminator outputs which will help us to highlight the main factors that are needed to recover a continuous carrier. The analysis is performed in the ideal case of no noise. In this case, the integrator outputs of Figure 4 can be written in complex form as

$$y_\varphi[k] = \frac{1}{2L}\sum_{n=kL}^{(k+1)L-1} e^{j(2\pi\Delta f(nT_s)nT_s+\Delta\psi)}, \tag{5}$$

where

$$\begin{aligned}\Delta f(nT_s) &= f_L(nT_s) - f_d(nT_s),\\ \Delta\psi &= \psi_L - \psi_s.\end{aligned} \tag{6}$$

In theory (5), should contain also a double frequency component, which, however, can be neglected as it is filtered out by the integrator. If $\Delta f(nT_s)$ is a constant Δf, the summation in (5) can be written as a geometrical progression, from which a closed-form expression can be found. In fact by writing $i = n - kL$, (5) becomes

$$\begin{aligned}y_\varphi[k] &= \frac{1}{2L}\sum_{i=0}^{L-1} e^{j(2\pi(i+kL)\Delta f T_s+\Delta\psi)}\\ &= e^{j(2\pi\Delta f kLT_s+\Delta\psi)} e^{j\pi\Delta f T_s(L-1)}\frac{\sin\pi\Delta f LT_s}{2L\sin\pi\Delta f T_s}\end{aligned} \tag{7}$$

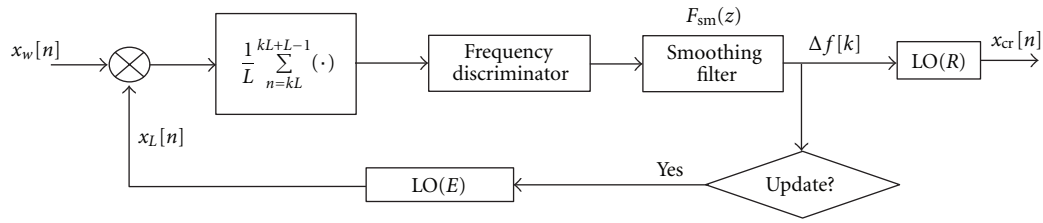


Figure 6: A quasi-open loop frequency estimator.

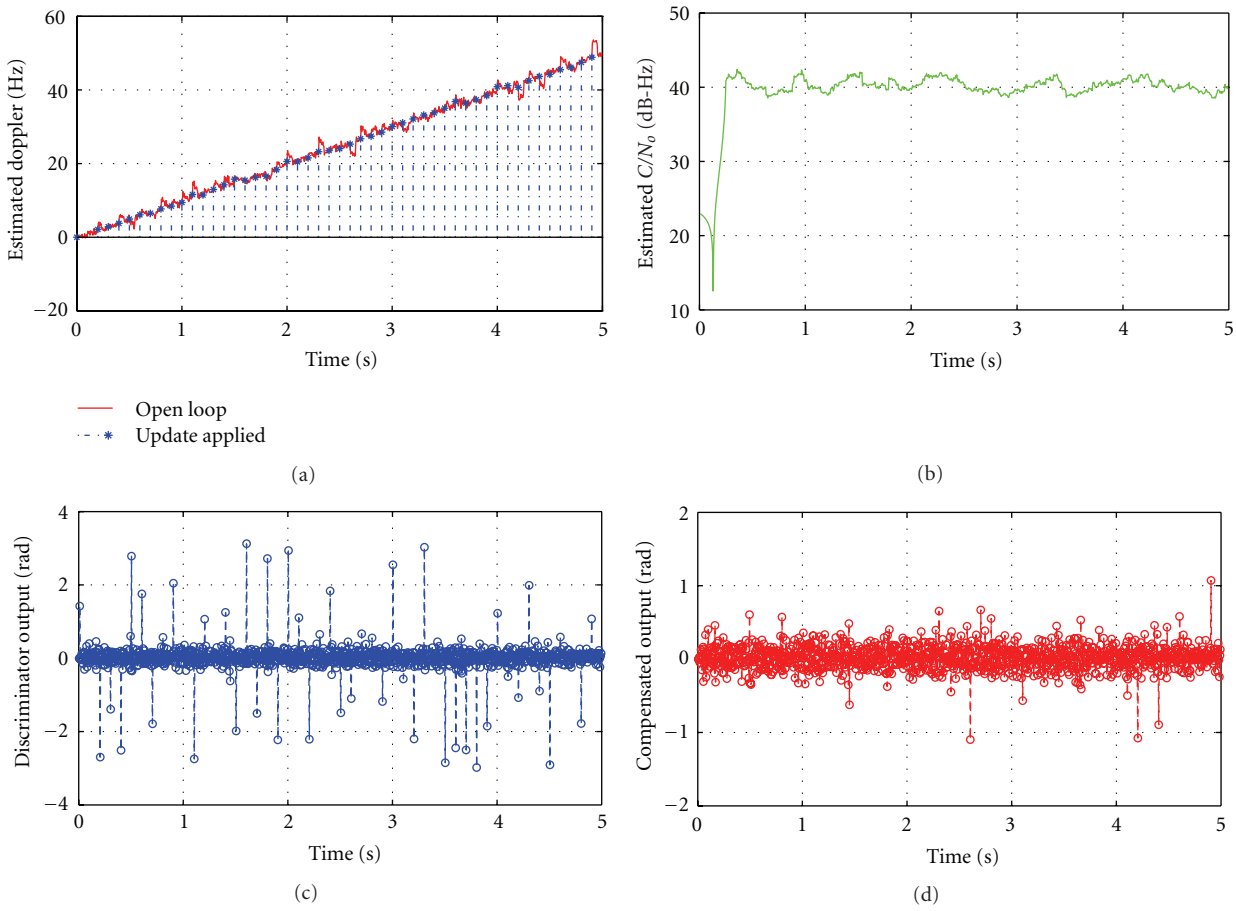


Figure 7: Proposed scheme tracking an input frequency ramp of 10 Hz/s, $C/N_o = 40$ dB-Hz, $T = 5$ ms.

whose instantaneous phase is

$$\varphi_{\text{inst}}[k] = 2\pi\Delta f\left(kL + \frac{L-1}{2}\right)T_s + \Delta\psi. \tag{8}$$

If $\Delta f(nT_s)$ is approximately constant in the integration interval LT_s, that is $\Delta f(nT_s) \cong \Delta f[k]$ for $kL \le n \le (k+1)L-1$, then it is possible to write

$$\varphi_{\text{inst}}[k] \cong 2\pi\Delta f[k]\left(kL + \frac{L-1}{2}\right)T_s + \Delta\psi. \tag{9}$$

This phase contains an integer number L_k of cycles plus a fractional part, from which

$$\varphi_{\text{inst}}[k] = L_k 2\pi + \Phi(nT_s), \tag{10}$$

where

$$\Phi(nT_s) = \text{mod}\left(\varphi_{\text{inst}}[k], 2\pi\right). \tag{11}$$

In the classical DPLL schemes for GNSS applications, this phase is estimated at the time epochs $n = kL$, for $k = 0, 1, \ldots$, as

$$\Phi(kLT_s) = \arctan\frac{Y_Q[k]}{Y_I[k]}, \tag{12}$$

where $Y_I[k] = \mathcal{R}(y_\varphi[k])$ and $Y_Q[k] = \mathcal{I}(y_\varphi[k])$. This discriminator function can also be substituted by other operations which approximate the arctan function. At each

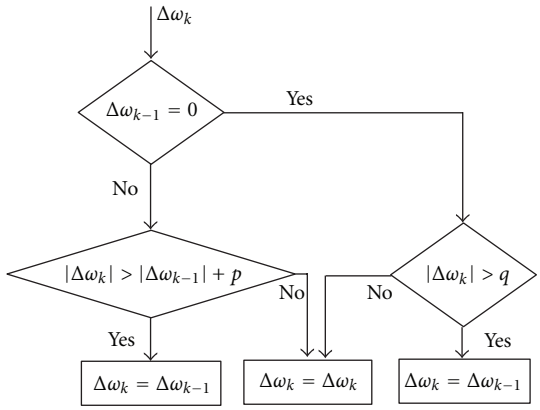


Figure 8: A simple algorithm to detect and remove discriminator output outliers.

epoch k, (12) provides a phase value available at the discrete-time instant $(k+1)L$ and representative of the instantaneous residual phase at the time instant

$$l_k T_s = T_s\left[(k+1)L - \frac{(L+1)}{2}\right], \tag{13}$$

that is, the residual phase information is provided with a delay of $(L+1)/2$. Notice that the recovered carrier $x_{cr}[n]$ has to be generated at each time instants $t_n = nT_s$, as indicated in (3). This means that the missing values of the instantaneous phase have to be evaluated in some way. This problem can be easily solved if the unknown frequency $\Delta f(nT_s)$ can be estimated from the outputs of the discriminator function (12), used for the instantaneous phase estimation. This part is described below.

In order to estimate the instantaneous frequency, a quantity $y_\varphi[k]y_\varphi^*[k-1]$ is at first evaluated, and then its phase is extracted using an arctan phase discriminator. This is the usual operation performed by a 4-quadrant arctan discriminator of the type, [2],

$$\Delta f[k] = \frac{1}{2\pi T}\arctan\left(\frac{\mathcal{I}\left(y_\varphi[k]y_\varphi^*[k-1]\right)}{\mathcal{R}\left(y_\varphi[k]y_\varphi^*[k-1]\right)}\right). \tag{14}$$

Although this discriminator is termed as maximum likelihood estimator by [2], it is important to mention that this estimator only achieves the Cramer-Rao lower bound at a sufficiently high C/N_o, as we have demonstrated in [11]. Notice that (14) provides the frequency value without evaluating the instantaneous phase, just applying the integrator output to (14). Another possibility for the frequency evaluation is to operate on the output of the phase discriminator given in (12). However, since in a frequency discriminator only the phase difference is of interest, the discriminator given in (14) is generally more convenient, as it inherently reduces the phase wrapping problem [12]. Once the estimate of $\Delta f[k]$ is available, the carrier can be continuously recovered, at each instant, by the oscillator LO(R).

Before addressing the role of the filter in this scheme we want to highlight some limitations encountered when a carrier with a time varying Doppler frequency has to be tracked. The main problem with this type of scheme is that we cannot use it alone or without a closed loop updating because of the limited linear region of the discriminator function. In fact, as the difference $\Delta f[k]$ between the frequencies of the incoming and local carriers increases with time, the system tends to operate outside the linear region, where tracking is no longer possible. Moreover, this linear region reduces as the coherent integration time T increases. Therefore, it is not possible to track the frequency evolution without updating the frequency of the local oscillator LO(E). All the conventional discriminators exhibit the same behavior, as discussed in [2].

As an example, Figure 5 shows the output of the discriminator, given by (14), for different integration times. This behavior is observed when we try to use the scheme, given in Figure 4, for tracking a time varying input Doppler at $C/N_o = 50$ dB-Hz.

4.2. Modified Scheme: Quasi Open Loop Scheme. A possible method to overcome the problem of tracking a time varying frequency is to update the frequency of the local oscillator LO(E) after N epochs, instead of updating it at each epoch, as in the closed loop systems. This means that we are proposing a scheme working with three different rates, as opposed to the typical closed loop schemes working with two different rates. In the proposed scheme, shown in Figure 6, the additional update rate of the LO(E) frequency, equal to $r_u = r_k/N$, is introduced.

The real motivation behind this three rate schemes is to make the integration time T and the NCO update interval independent of each other, obtaining in turn an additional degree of freedom. The latter allows us to ease the design of the loop, as it will be shown below. The system can be considered *quasi-open*, as it works as an open loop between two updating epochs.

Notice that in our scheme only the frequency of the NCO is updated, while the phase is kept continuous at each epoch, regardless of whether the updating is or is not applied. The value of N depends on the specific application. For example, in case of high dynamics we need to update the NCO frequency more frequently, so a smaller value of N is necessary. Furthermore, we can also decide to have either regular update intervals, that is, to update after each N epoch, or irregular intervals which depend on the incoming Doppler evolution. In this study we have only considered regular update intervals.

4.3. Choice and Design of Loop Filter. The scheme of Figure 6 allows us to approach the design of the loop filter, taking into account that now its role is only to reduce the effect of noise on the frequency estimate. This is possible thanks to the available additional degree of freedom of the quasi-open scheme. The filter can be considered as a part of the discriminator, whose only task is to smooth the discriminator output. To this purpose, any reasonable choice

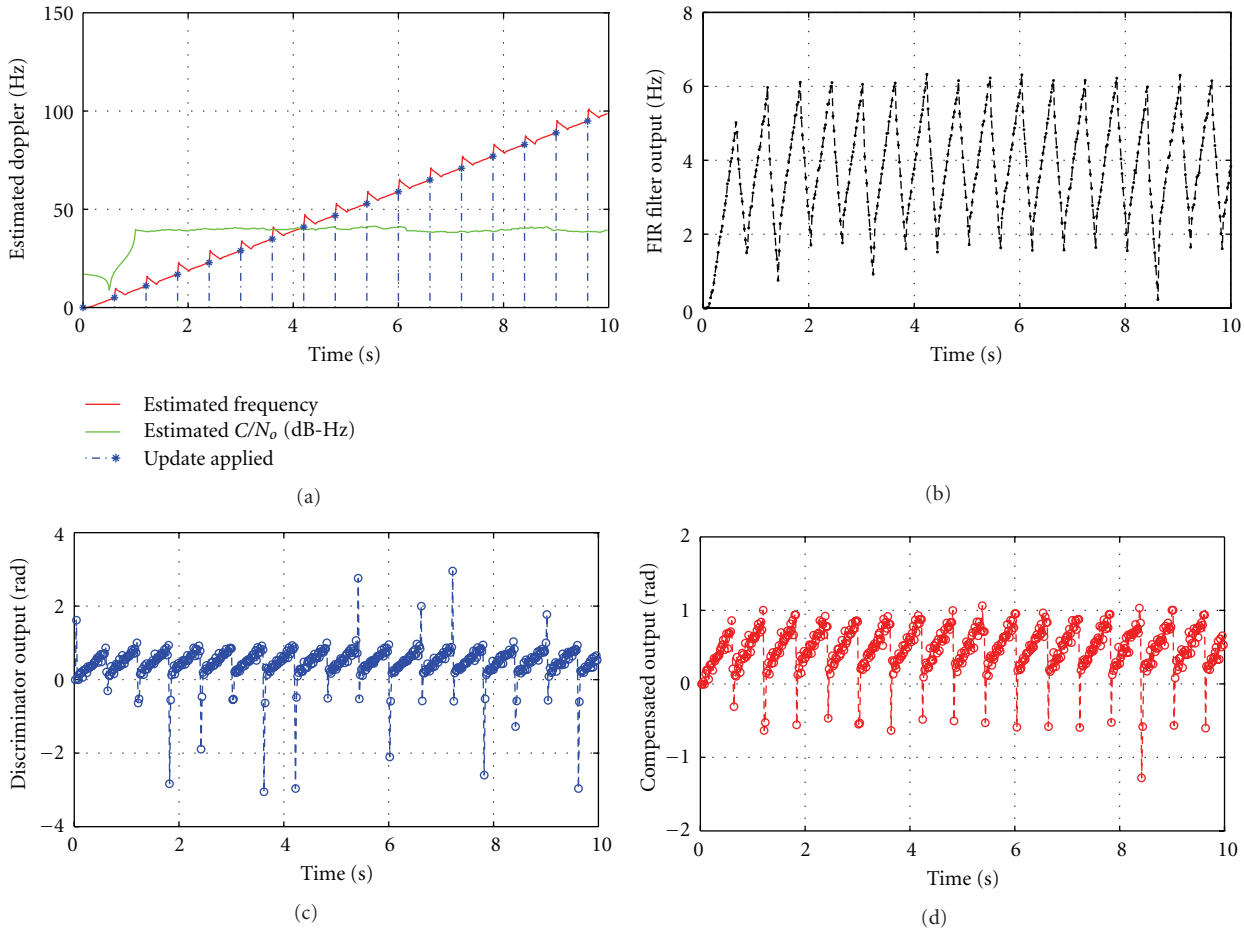


FIGURE 9: Proposed Scheme tracking an input frequency ramp of 10 Hz/s, $C/N_o = 40$ dB-Hz, $T = 20$ ms.

of the filter structure and parameters can be adopted, taking into account the specific application. Since finite impulse response (FIR) filters are simple to design, always stable, relatively insensitive to quantization and can have linear phase, we decided to explore the possibility to use them for denoising.

The transfer function $F_{\text{sm}}(z)$ of an FIR filter is

$$F_{\text{sm}}(z) = \sum_{i=0}^{L_{\text{sm}}-1} b_i z^{-i}, \tag{15}$$

where L_{sm} denotes the number of taps and b_i are the filter coefficients, which become $b_i = 1/L_{\text{sm}}$ in the simplest case of a moving average (MA) filter. An MA-FIR filter is an excellent smoothing filter, but its frequency roll-off is slow and its stopband attenuation is ghastly, making it a scarcely effective low-pass filter. This is a typical result, as a digital filter can be generally optimized for time or frequency domain performance, but not for both. Since in our application the filter task is to mitigate the noise effect, the choice of an MA-FIR structure is justified. Notice that the purpose of this paper is to show the feasibility of a quasi-open architecture, then we did not concentrate on the optimization of the structure of the smoothing filter, which could be also implemented with infinite impulse response (IIR) filters or with more optimized versions of FIR filters.

A possible consideration for choosing the type of filter could be to keep the computational complexity as low as possible. This complexity can be attributed to the number of operations (additions and multiplications) needed to compute the filter response. For digital filters, the computational complexity is more or less proportional to the number of filter coefficients. Usually, we need more than 5 taps to get good smoothing results in case of MA-FIR filters. With these values the computational complexity of a quasi-open loop scheme, which utilizes a smoothing MA-FIR filter, will be slightly higher than the one of an FLL utilizing 2nd order IIR filters. But, as mentioned above, we have many other advantages of using FIR filters as compared to IIR filters.

It is important to mention that the use of these filters changes the philosophy of the loop filter. As explained earlier, in a closed loop architecture the filter has a twofold task: one is to reduce the effect of the noise and the second one is to control the loop dynamics. In the schemes with a smoothing MA-FIR filter the loop dynamics is solely determined by the frequency discriminator. For example, it has been shown that the estimator proposed in (14) is unbiased if we have to track a constant frequency shift or a linear frequency ramp. But its

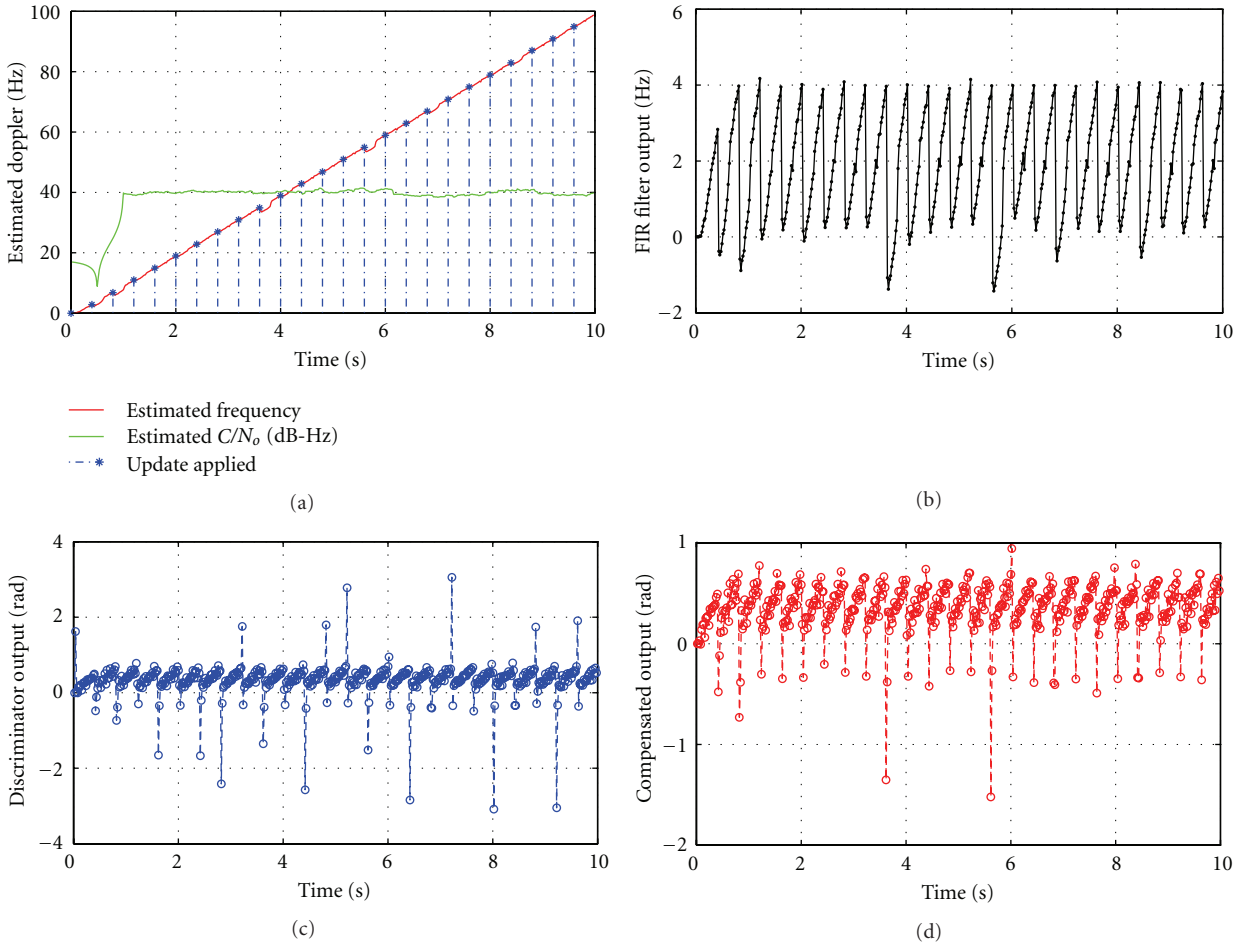


FIGURE 10: Proposed scheme tracking an input frequency ramp of 10 Hz/s, $C/N_o = 40$ dB-Hz, $T = 20$ ms.

estimate is biased when we have to track an input frequency with a quadratic component [8]. This is not a problem in our scheme as we can accommodate a higher order dynamics by taking additional measures. For example, we can either change the design of the discriminator, as discussed in [13], or we can insert a new block after the FIR filter to account for the higher dynamics. This problem has not been considered in this study, which is mainly devoted to the feasibility and performance of the quasi-open structure.

5. Results and Discussions

In this section we present some simulation results obtained by using this new type of quasi-open loop frequency estimator. We start from a basic scheme using an MA-FIR filter, we observe the results, and we introduce some modifications in the basic scheme to improve its performance. The system performance is described in terms of tracking jitter, also taking into account weak signal scenarios.

5.1. Some Implementation Aspects. In the first simulation example, a GNSS-like signal was generated with a ramp-type time-varying Doppler shift, with a slope of 10 Hz/s. The signal was processed by a quasi-open loop frequency estimator, followed by a smoothing MA-FIR filter with $L_{sm} = 10$ taps. In the first stage (after switch-on) the system is completely open, and, after a time interval equal to the transient of the FIR filter, starts updating the loop. The updating is repeated at each integration interval NT.

The results with $N = 20$ are shown in Figure 7, which shows that the proposed scheme is able to successfully track the input Doppler frequency 7(a).

However, observing the results of Figure 7, we recognize that some problems are associated with this type of scheme. First of all, at each new update there is a jump in the discriminator output 7(c). These jumps give rise to undesired outliers which depend on the type of updating. These outliers can be compensated using the scheme shown in Figure 8, which adopts a simple strategy to eliminate any kind of anomaly, based on the comparison between the current and the previous discriminator outputs, indicated in Figure 8, respectively, as $\Delta\omega_k$ and $\Delta\omega_{k-1}$, and measured in rad units. The sudden jumps are eliminated by setting two thresholds p and q, whose values depend on the rate of the input Doppler frequency. We have set both of them equal to π in all the results presented in this paper. The

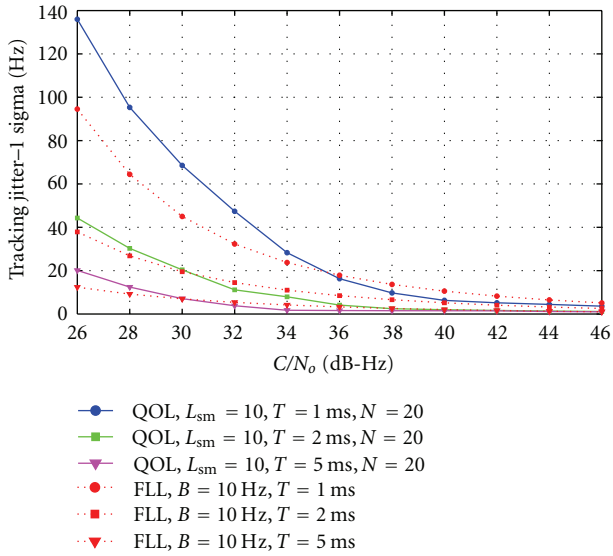


Figure 11: Tracking jitter performance.

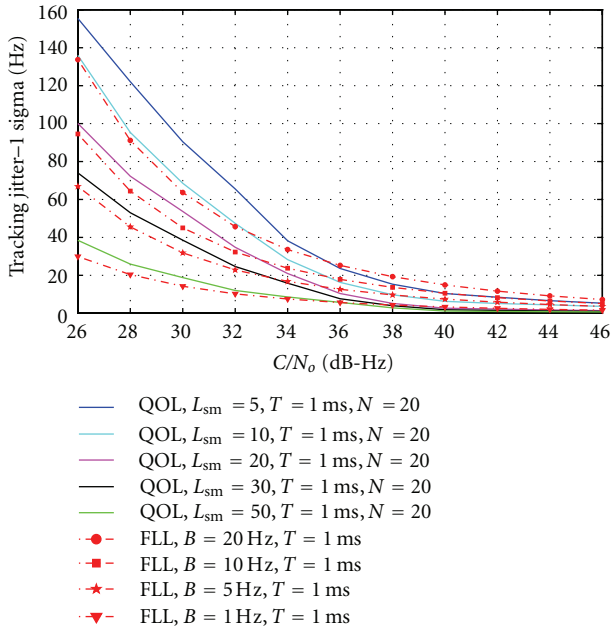


Figure 12: Tracking jitter performance with same integration time.

compensated discriminator output shown in Figure 7(d) proves the effectiveness of the method.

The second problem associated to this type of scheme is related to the inherent time-transient of any FIR filter, which generates some regular slowly decreasing jumps at each new updating interval. This is due to the fact that the filter output experiences a transient, which vanishes only when the filter memory (equal to the filter length) is completely filled with input samples. This effect is clearly visible in Figure 9(a) where the integration time is $T = 20$ ms. A possible solution to mitigate this problem is to use a moving average FIR filter

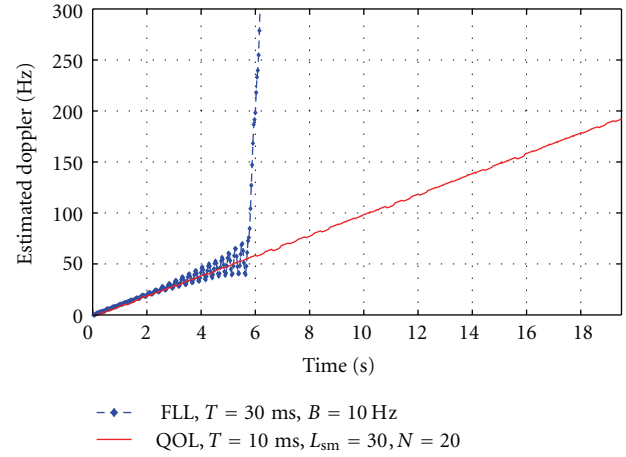


Figure 13: FLL versus QOL tracking results for incoming Doppler variation of 10 Hz/s, $C/N_o = 40$ dB-Hz.

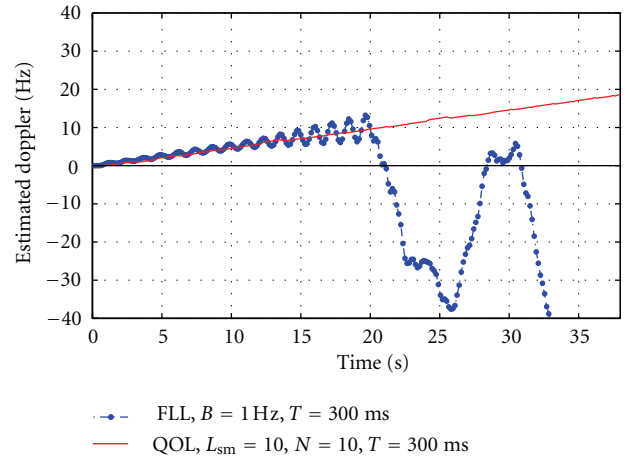


Figure 14: FLL versus QOL tracking results for incoming Doppler variation of 0.5 Hz/s, $C/N_o = 30$ dB-Hz.

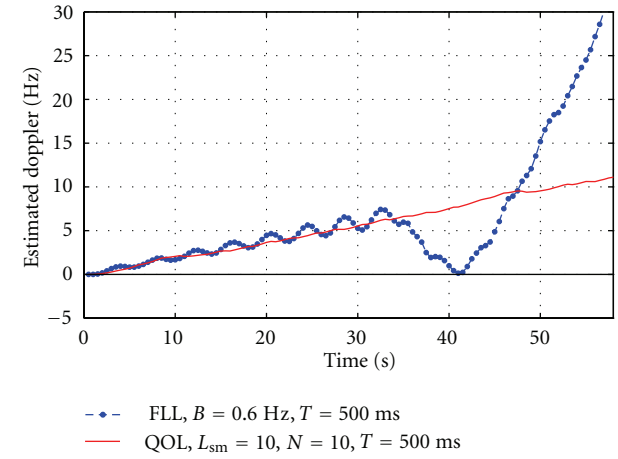


Figure 15: FLL versus QOL tracking results for incoming Doppler variation of 0.2 Hz/s, $C/N_o = 20$ dB-Hz.

with a time-variant length. The number of the taps is set to one at each new updating, and then it is gradually increased to reach the specified maximum value L_{sm}. The results are presented in Figure 10, where we can see that we have no more jumps in the output.

At this point we have an architecture where it is possible to set the update rate of the NCO independently from the integration time. We have also demonstrated the feasibility of using an MA-FIR filter with a time-varying length as loop filter. The filter design is very easy and the resulting structure is always stable.

5.2. Performance of the Quasi-Open Scheme in User Dynamics. The updating interval in the proposed scheme is strictly related to the dynamics under which the system has to work. In case of high dynamic applications the updating has to be applied more frequently, and this implies a smaller value of N, and vice versa. The dynamic range sustained by the quasi-open loop scheme does not only depend on N, but also on the linear region of the discriminator in (14), which, in turn, depends on the integration time T. We can relate the linear range of the discriminator, denoted by Δ_D (for a given T), and the rate of change of the input Doppler frequency, denoted by $\dot{f}_d$ and expressed in Hz/s as

$$\dot{f}_d TN < \frac{\Delta_D}{2} \tag{16}$$

obtaining the condition for the quasi-open loop scheme to work properly. In this equation, the term $\dot{f}_d TN$ indicates the amount of Doppler accumulated between two updating intervals of the NCO. If this term lies inside the discriminator range, then the proposed scheme successfully tracks the Doppler variations.

As an example, for $T = 10$ ms, the linear range of the discriminator is about $\Delta_D = 100$ Hz (from -50 Hz to 50 Hz in Figure 5). Thus, for $T = 10$ ms and $N = 20$, the maximum Doppler variation that can be tracked successfully by the quasi-open loop scheme is given by

$$\dot{f}_d < \frac{\Delta_D}{2NT} = 250\,\text{Hz/s}. \tag{17}$$

If the input Doppler variation exceeds this value, then the quasi-open loop is not able to track the incoming frequency. The problem can be solved by updating the NCO frequency more often. This can be achieved by decreasing the value of N.

5.3. Tracking Jitter Performance. We carried out some simulation experiments to examine the tracking jitter performance of the proposed quasi-open loop scheme and we compared it with the theoretical jitter of a classical FLL. The results shown in Figure 11 were obtained for different integration times T and different C/N_o, by keeping the number L_{sm} of the MA-FIR filter taps constant, and by updating the NCO (local oscillator LO(E)) frequency after $N = 20$ epochs. In this way we were able to analyze the effect of increasing the integration time on the jitter performance.

To compare the results with those of a traditional closed loop scheme, we used the formula of the theoretical FLL tracking jitter, due to thermal noise, given in [2], that is

$$\sigma_{\text{FLL}} = \frac{1}{2\pi T}\sqrt{\frac{4FB}{C/N_o}\left[1 + \frac{1}{TC/N_o}\right]}\,[\text{Hz}]. \tag{18}$$

These values are also shown in Figure 11 for different integration times, and for a loop noise bandwidth $B = 10$ Hz. The choice of the loop noise bandwidth suitable for comparison is not an easy task and will be discussed in the next section.

It is clear from the obtained results that the tracking jitter performance of the quasi-open loop is better than that of an FLL for all the integration times at high C/N_o, but the performance degrades at low C/N_o. The main reason for this could be the poor quality of the frequency estimator (14) at a low C/N_o.

5.4. Loop Noise Bandwidth versus Filter Taps. In this section, we discuss the equivalence between the loop noise bandwidth B of the closed loop schemes and the number L_{sm} of the filter taps in a quasi-open loop. In traditional closed loop schemes, like FLLs, the parameter B controls both the amount of noise rejected by the loop and the dynamic stress. If we increase B, the loop can sustain more dynamics, but more noise also affects the system, and vice versa.

A similar role is played by the number L_{sm} of filter taps in a quasi-open scheme. Increasing L_{sm} implies more smoothing, but imposes some constraints on the update interval N, which should be larger than L_{sm}. In fact, if $N < L_{\text{sm}}$ the filter never attains steady-state, thereby reducing the effectiveness of the smoothing, which will be incomplete. In other words, since the number of taps is variable and is set to one at each new update, the maximum number of taps will be N instead of L_{sm}. Another constraint on the value of N is the dynamic stress which can be sustained by the quasi-open loop scheme.

The equivalence between B and L_{sm} can be seen in Figure 12, which shows the tracking jitter curves for both schemes, with different values of B for the FLL and different values of L_{sm} for the quasi-open scheme. The curves were obtained by keeping the integration time constant. By observing these curves we can get an idea of the number of taps of the MA-FIR filter required to achieve the same jitter performance of an FLL with a given noise bandwidth. As shown in Figure 12, the jitter performance improves as L_{sm} increases, and B decreases.

It is very difficult to find the exact equivalence between these two parameters from these results. However, since we are usually interested in low C/N_o regions, we can somehow relate the two parameters. For example, to obtain the same performance of an FLL with $B = 10$ Hz, we need an MA-FIR filter with $L_{\text{sm}} = 20$ taps for $T = 1$ ms. On the other hand, for $B = 1$ Hz, the number of taps has to be $L_{\text{sm}} = 50$, which is quite complex in terms of computational cost. So, a reasonable choice could be to fix an approximate limit of $L_{\text{sm}} \leq 20$. In this way we will get almost the same performance of an FLL with $B = 10$ Hz.

In the following section we adopt this equivalence between B and L_{sm}, and, based on this, we show the advantages obtained with a quasi-open loop scheme in weak signal conditions.

5.5. Weak Signal Performance. In this section, we demonstrate that, by using a quasi-open scheme, we can work with extended coherent integration times which otherwise would not be possible with closed loop schemes under the same conditions, because of the resulting loop filter instability.

In the first simulation experiment, we again considered an incoming signal with a ramp-type Doppler frequency with a slope of 10 Hz/s, and with $C/N_o = 40$ dB-Hz. Although this is not a weak signal, we started with this value to better highlight the performance of both schemes, FLL and quasi-open loop, when they work under the same conditions. We set the update interval to $N = 20$, and the number of filter taps to $L_{sm} = 10$. For this choice of L_{sm}, a reasonable value of the FLL noise bandwidth could be around $B = 10$ Hz, as discussed in Section 5.4. The tracking results are shown in Figure 13 for $T = 30$ ms. Under these conditions $BT = 0.3$, which is high enough for the loop filter to become unstable. For the quasi-open loop scheme we can extend the integration time even more as long as the condition (16) remains valid.

In the next simulation experiment, we considered a weak signal with $C/N_o = 30$ dB-Hz, and with a Doppler varying at a rate of 0.5 Hz/s. Since C/N_o is very low, we set $T = 300$ ms, and the FLL bandwidth to $B = 1$ Hz, so as to have a suitable noise rejection performance for weak signals. For the quasi-open loop scheme, we chose $L_{sm} = 10$ and $N = 10$ to fulfill the condition (16). Here the value $N = 20$ used in the previous experiment cannot be kept because of condition (16). The results are shown in Figure 14. Again we observe the same situation observed in Figure 13: the FLL is no longer able to track the carrier, because of the loop filter instability, while the proposed scheme works well and successfully tracks the incoming Doppler variation.

Figure 15 shows the tracking results for an incoming signal with $C/N_o = 20$ dB-Hz, and a Doppler variation with a rate of 0.2 Hz/s. The integration time is set to $T = 500$ ms, since C/N_o is very low. For the quasi-open loop scheme, we chose $N = 10$, a value which does not violate the condition (16). The figure shows that the proposed scheme successfully tracks the frequency of the incoming signal even at this low C/N_o.

6. Conclusion

A novel quasi-open loop architecture has been proposed for tracking the frequency of received GNSS signals. The proposed architecture works with three different rates, unlike the classical closed loop schemes, PLLs and FLLs, which work with two different rates. The additional degree of freedom of the quasi-open scheme enables us to ease the design of the loop filter. Simulation results show that it is possible to design this filter with an FIR structure, by adopting some very simple rules for the design. Moreover, the system results advantageous also in terms of stability when compared to a traditional closed loop architecture. It is also important to mention that if this type of scheme is used with classical DPLL schemes in an assisted manner, then a lower bandwidth and higher coherent integration times can be utilized in DPLL to track the incoming carrier in very weak signal conditions.

References

[1] B. Parkinson and J. Spilker, Eds., *Global Positioning System: Theory and Applications*, vol. 1, American Institute of Aeronautics and Astronautics, Washington, DC, USA, 1996.

[2] E. Kaplan and C. Hegarty, Eds., *Understanding GPS Principles and Applications*, Artech House, Norwood, Mass, USA, 2006.

[3] P. L. Kazemi, *Development of new filter and tracking schemes for weak gps signal tracking*, Ph.D. dissertation, Department of Geomatics Engineering, University of Calgary, 2010.

[4] S. A. Stephens and J. B. Thomas, "Controlled-root formulation for digital phase-locked loops," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 1, pp. 78–95, 1995.

[5] P. L. Kazemi, "Optimum digital filters for GNSS tracking loops," in *Proceedings of the 21st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS '08)*, pp. 1188–1197, Center Savannah, Ga, USA, September 2008.

[6] M. L. Psiaki, "Smoother-based GPS signal tracking in a software receiver," in *Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS '01)*, pp. 2900–2913, Salt Palace Convention Center, Salt Lake City, Utah, USA, 2001.

[7] A. M. M. Kamel, *Context aware high dynamics GNSS-INS for interference mitigation*, Ph.D. dissertation, Department of Geomatics Engineering, University of Calgary, 2011.

[8] J. Riba, J. Tom, and M. Lagunas, "Instantaneous open-loop frequency estimation methods for navigation receivers," in *Proceedings of the 2nd European Symposium on Global Navigation Satellite Systems (GNSS'98)*, Toulouse, France, 1998.

[9] S. Kay, "Fast and accurate single frequency estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 1987–1990, 1989.

[10] F. van Graas, A. Soloviev, M. Uijt de Haag, and S. Gunawardena, "Closed-loop sequential signal processing and open-loop batch processing approaches for GNSS receiver design," *IEEE Journal on Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 571–586, 2009.

[11] M. Tahir, M. Fantino, and L. L. Presti, "Characterizing different open loop fine frequency estimation methods for GNSS receivers," in *Proceedings of the International Technical Meeting of The Institute of Navigation (ION ITM '12)*, Newport Beach, Calif, USA, 2012.

[12] M. L. Fowler, "Phase-based frequency estimation: a review," *Digital Signal Processing*, vol. 12, no. 4, pp. 590–615, 2002.

[13] D. Simon and H. El-Sherief, "Fuzzy logic for digital phase-locked loop filter design," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, pp. 211–218, 1995.

Hindawi Publishing Corporation
International Journal of Navigation and Observation
Volume 2012, Article ID 804732, 11 pages
doi:10.1155/2012/804732

Research Article

A New Multipath Mitigation Method for GNSS Receivers Based on an Antenna Array

Sébastien Rougerie,[1,2,3] Guillaume Carrié,[2] François Vincent,[4] Lionel Ries,[1] and Michel Monnerat[3]

[1] *CNES—The French Space Agency, Navigation/Localization Signal & Equipment Department, 18 Avenue Edouard Belin, 31401 Toulouse Cedex 9, France*
[2] *ONERA—The French Aerospace Lab ONERA, DEMR Department, 2 Avenue Edouard Belin, 31000 Toulouse, France*
[3] *Thales Alenia Space, BS Navigation & Integrated Communication, 26 Avenue Jean-François Champollion, 31037 Toulouse, France*
[4] *Institut Supérieur de L'Aéronautique et de l'Espace ISAE, DEOS Department, 10 Avenue Edouard Belin, 31055 Toulouse Cedex 4, France*

Correspondence should be addressed to Sébastien Rougerie, sebastien.rougerie@hotmail.fr

Received 4 November 2011; Revised 29 February 2012; Accepted 21 March 2012

Academic Editor: Pau Closas

The well-known Space-Alternating Generalized Expectation Maximisation (SAGE) algorithm has been recently considered for multipath mitigation in Global Navigation Satellite System (GNSS) receivers. However, the implementation of SAGE in a GNSS receiver is a challenging issue due to the numerous number or parameters to be estimated and the important size of the data to be processed. A new implementation of the SAGE algorithm is proposed in this paper in order to reach the same efficiency with a reduced complexity. This paper focuses on the trade-off between complexity and performance thanks to the Cramer Rao bound derivation. Moreover, this paper shows how the proposed algorithm can be integrated with a classical GNSS tracking loop. This solution is thus a very promising approach for multipath mitigation.

1. Introduction

In Global Navigation Satellite System (GNSS) applications, multipath (MP) errors are still one of the major error sources for conventional receivers. The additional signal replicas due to reflections on the local environment introduce a bias in the delay lock loops (DLLs), which finally leads to a positioning error [1, 2]. Several techniques have been developed for multipath mitigation. One of the most popular approaches is the Narrow Correlator Spacing [3], which reduces the chip spacing between the early and late correlators in order to mitigate the impact of multipath. However, this technique suffers from high sensitivity to noise and cannot perform with short delay multipath (<0.1 chip). Based on the Maximum Likelihood (ML) estimation, the Multipath Estimating Delay-Lock-Loop (MEDLL) [4] algorithm has also been proposed to estimate the delay and the power of all the paths by studying the shape of the cross-correlation function. This approach shows better performances than the Narrow Correlator Spacing technique, but short delay multipath mitigation is still an issue [4]. More recently, Bayesian approaches have been proposed [5–7]. Indeed, most of the time, prior information could be used in order to improve the delays estimations. However in practice, it is difficult to get correct prior information. Measurement campaigns can be used to build a first-order Markov process for a sequential estimation, but the performance will consequently be strongly dependent on the measured environment (design of the city...).

Last, the use of array antenna algorithms has been proposed for multipath mitigation [8, 9]. Array antennae enable a spatial sampling that makes it possible to distinguish different sources in the spatial domain. Therefore, mitigation techniques based on such an array are independent of the relative delays of the MP. Consequently, the rejection of any kind of MP seems possible (and especially short-time delay MP).

In conventional mobile receivers, the room available to incorporate an array antenna is reduced. Only a small number of elements can be integrated. This study will focus on a square 2×2 elements antenna. The choice of the shape of the array antenna is out of the scope of this paper, and we will only focus on the array processing.

Using appropriate weights on each channel, beamforming technics perform a spatial filtering in order to concentrate the energy beam towards the signal of interest, while trying to minimise the gain towards the interferences. However, in the MP mitigation context, these classical approaches present limited performances for the following reasons [10].

(i) Near angle signals are difficult to separate because of the low antenna resolution [10, 11].

(ii) Line-of-sight signal (LOSS) and multipaths are strongly correlated, which implies a severe degradation of adaptive beamforming algorithms.

(iii) Adaptive algorithms should operate after the correlation step in order to work with positive SNR signals. That implies that only a small number of samples are available to estimate the array covariance matrix. Bad covariance matrix estimation can lead to inversion or eigenvalues decomposition instabilities.

To overcome these problems, several solutions have been proposed. In [12, 13], for example, the authors propose to use additional Choke ring techniques, GPS microstrip array antenna, and/or angle constraint for negative elevation in order to reject ground multipaths. However, for a generic application, we cannot assume any specific MP DOA.

In order to improve the MP mitigation, another approach proposes to include the different DOA paths parameters in the estimation procedure [10, 14–18]. In other words, we estimate a set of parameters (amplitudes, times/delays, Doppler shifts, elevations, and azimuths) for all the incoming sources. The main difference with the beamforming approach is that, instead of filtering the sources in the spatial domain only, the different incoming paths are jointly identified in the space, time, and frequency domains. In order to estimate the parameters of all impinging signals, the Space Alternating Generalized Expectation Maximisation (SAGE) algorithm [16], which is a low-complexity generalization of the Maximum Likelihood (ML) algorithm, has been considered. The SAGE algorithm is usually used in communication systems [16], but the potential of SAGE in a navigation context has also been proven [17, 18]. Nevertheless, the computational cost increases due to the number of unknown parameters. Moreover, the memory size is also a challenging issue. Thus, the SAGE algorithm can hardly be directly implemented in real time for GNSS receivers. Last, the SAGE algorithm provides an estimation of the time delay and phase of the LOSS, as do the DLL and PLL (Phase Lock Loop). Thus, the use of SAGE implies to switch off the DLL/PLL, and, consequently, we lose the "smoothing effect" of the tracking loops.

In order to reach the efficiency of the SAGE algorithm with a reduced complexity, and to keep the compatibility with conventional GNSS tracking loops (DLL/PLL), we proposed a new implementation of SAGE in [10]. The main idea is to apply the SAGE algorithm after the local correlation step. Indeed, the cross-correlation between the received signal and the local code can be seen as a compression step of the baseband signal. Thus, by reducing the size of the input signal, the complexity of the algorithm will reduce in the same proportion. By applying the ML estimation on the postcorrelated signal, we estimate the relative delay and Doppler of the paths. These estimations are then used to drive the GNSS tracking loops. We named this new algorithm the SAGE/STAP multicorrelator algorithm.

Comparisons based on Monte Carlo simulations between the SAGE/STAP algorithm, the SAGE algorithm, and beamformers have been done in [10]. This paper focuses on the Cramer Rao Bound (CRB) derivation in order to present the theoretical trade-off between the complexity reduction and the estimation accuracy of the SAGE/STAP algorithm. Moreover, this paper presents an implementation of the proposed algorithm hybridized with the GNSS tracking loops

This paper is organized as follows. The signal model and the main assumptions are outlined in Section 2. A review of the SAGE algorithm, the SAGE/STAP multicorrelator algorithm, and its implementation in a GNSS receiver is given in Section 3. The simulation results in Section 4 show the trade-off between the complexity and the accuracy of the SAGE/STAP multicorrelator algorithm and present the performances of the proposed algorithm in realistic scenarios. Finally, in Section 5 we present our conclusions.

2. Signal Model

Let us assume that we receive L narrowband planar wave fronts of wavelength λ on an array of M isotropic sensors. The M-sized vectorized received signal, $\mathbf{z}(t)$, can be written as

$$\mathbf{z}(t) = \sum_{l=0}^{L-1} \mathbf{s}_l(t, \psi_l) + \mathbf{n}(t), \tag{1}$$

where $\mathbf{s}_l(t, \psi_l)$ is given by

$$\mathbf{s}_l(t, \psi_l) = \gamma_l \mathbf{a}(\theta_l, \varphi_l) \exp(2j\pi\nu_l t) c(t - \tau_l), \tag{2}$$

and $\mathbf{n}(t) \sim N(0, \sigma_n^2 \mathbf{I}_M)$ is an additional complex white Gaussian noise. Let us note that L the number of incoming paths (LOSS included) and $\psi_l = [\gamma_l, \theta_l, \varphi_l, \tau_l, \nu_l]^T$ are, respectively, the complex amplitude, elevation, azimuth, time delay, and Doppler shift of the lth path. Note that the index $l = 0$ corresponds to the LOSS. Here, c denotes the pseudorandom-noise (PRN) sequence that consists of a Gold code with a code period $T = 1$ ms, 1023 chips per period, and a rectangular chip shape. $\mathbf{a}(\theta, \varphi)$ represents the steering vector of a 2×2 square array antenna. The channel parameters are assumed constant during the observation time.

Collecting the samples of the complex baseband signal leads to a $M \times N_{\text{snap}}$, matrix, where N_{snap} denotes the number of SnapShots. Typically in GNSS, N_{snap} is larger than 2000, and, thus, a direct processing of the baseband signal is hardly implemented in real time. In order to compress the signal,

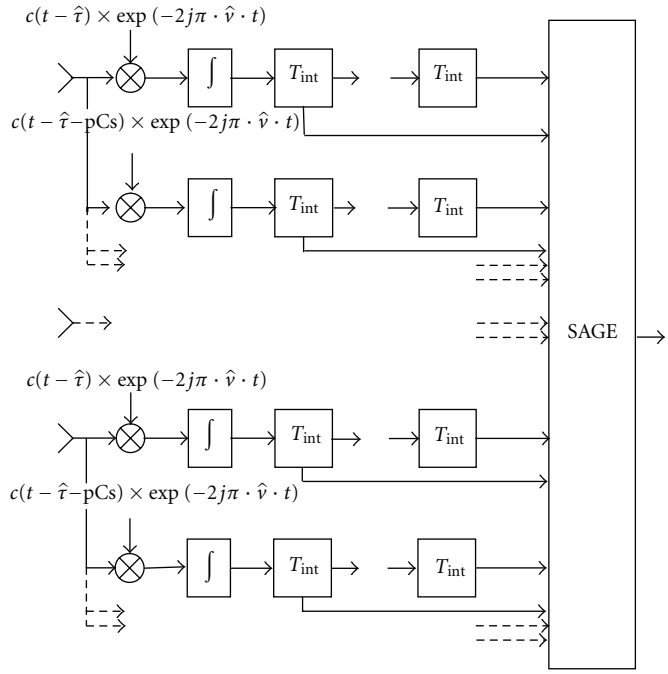


Figure 1: Principle of the STAP multicorrelator.

we propose to use the scheme presented in the Figure 1. The aim of the array antenna is to sample the wave fronts and to get access to the spatial domain. The banc of correlators is used to compress the signal and to get access to the relative delays of the sources, and finally the postcorrelation time taps enable to get access to the frequency domain. We can see that the first and last parts of this architecture are equivalent to a STAP (space time adaptive process) array applied to the postcorrelated signal. Thus, we call this architecture the STAP multicorrelator array. This array is then defined by 4 parameters: the number M of antennas, the number N of postcorrelation taps, the number P of correlators, and the time spacing Cs between each correlator. Note that the sampling time of the postcorrelated signal is equal to T_{int}. Therefore, we have the following relation between the number of SnapShots and the number of postcorrelated Taps: $N_{\text{snap}} = NT_{\text{int}}f_s$, where f_s denotes the sampling frequency of the baseband signal. The nth output of one correlator delayed by a time spacing of pCs is given by

$$\begin{aligned} &r_{C,l}(\tau_{rl}, \nu_{rl}, p, n) \\ &= \frac{1}{T_{\text{int}}} \int_{(n-1)T_{\text{int}}}^{nT_{\text{int}}} c(t-\tau_l)c(t-\hat{\tau}-pCs) \\ &\quad \times \exp[-2j\pi(\nu_l - \hat{\nu})t]dt, \end{aligned} \tag{3}$$

with $n \in [1, N]$, if P is even $p \in [-P/2, P/2]$ and if P is odd $p \in [-(P-1)/2, (P-1)/2]$. We introduce the relative delay of the DLL estimation $\hat{\tau}$ and the relative Doppler of the FLL (Frequency Lock Loop) estimation $\hat{\nu}$ with respect to the lth path parameters: $\tau_{rl} = \hat{\tau} - \tau_l, \nu_{rl} = \hat{\nu} - \nu_l$. We can then approximate the integral as

$$\begin{aligned} &r_{C,l}(\tau_{rl}, \nu_{rl}, p, n) \\ &= r(\tau_{rl} + pCs)\sin_c[\pi\nu_{rl}T_{\text{int}}] \exp\left[2j\pi\left(\nu_{rl}\frac{2n-1}{2}T_{\text{int}}\right)\right], \end{aligned} \tag{4}$$

with $r(\cdot)$ the autocorrelation function of the PRN code and $\sin_c[x] = \sin[x]/x$. The amplitude and phase components are independent of p and n; they can be inserted in the modified complex amplitude $\tilde{\gamma}_l$ of the incoming path:

$$\tilde{\gamma}_l = \gamma_l \sin_c[\pi\nu_{rl}T_{\text{int}}] \exp[-j\pi\nu_{rl}T_{\text{int}}]. \tag{5}$$

Then,

$$r_{C,l}(\tau_{rl}, \nu_{rl}, p, n) = \frac{\tilde{\gamma}_l}{\gamma_l}\tilde{r}_C(\tau_{rl}, \nu_{rl}, p, n), \tag{6}$$

with

$$\tilde{r}_C(\tau_{rl}, \nu_{rl}, p, n) = r(\tau_{rl} + pCs) \exp(2j\pi\nu_{rl}nT_{\text{int}}). \tag{7}$$

The outputs of the P correlators are rearranged in a column vector. The N cross-correlation functions obtained

are then concatenated in order to get the temporal evolution of the postcorrelation signal:

$$\widetilde{\mathbf{r}}_C(\tau_{rl},\nu_{rl}) = \begin{bmatrix} \widetilde{r}_C\left(\tau_{rl},\nu_{rl},-\dfrac{(P-1)}{2},1\right) \\ \vdots \\ \widetilde{r}_C\left(\tau_{rl},\nu_{rl},\dfrac{(P-1)}{2},1\right) \\ \widetilde{r}_C\left(\tau_{rl},\nu_{rl},-\dfrac{(P-1)}{2},2\right) \\ \vdots \\ \widetilde{r}_C\left(\tau_{rl},\nu_{rl},\dfrac{(P-1)}{2},2\right) \\ \vdots \\ \widetilde{r}_C\left(\tau_{rl},\nu_{rl},\dfrac{(P-1)}{2},N\right) \end{bmatrix}_{NP\times 1}. \quad (8)$$

Finally, the output signal of the STAP multicorrelator array can be collected in a mNP column vector:

$$\mathbf{x} = \sum_{l=0}^{L-1} \mathbf{x}_l(\Psi_l) + \mathbf{n}_{pc}, \quad (9)$$

where $\mathbf{n}_{pc}$ is the output noise and $\mathbf{x}_l(\Psi_l)$ is given by

$$\mathbf{x}_l(\Psi_l) = \widetilde{\gamma}_l \mathbf{a}(\theta_l,\varphi_l) \otimes \widetilde{\mathbf{r}}_C(\tau_{rl},\nu_{rl}), \quad (10)$$

where $\otimes$ denotes the Kronecker product, and the parameters are now $\Psi_l = [\widetilde{\gamma}_l, \theta_l, \varphi_l, \tau_{rl}, \nu_{rl}]^T$, where τ_{rl}, ν_{rl} denote the relative delay and Doppler of the path l.

3. SAGE Algorithm

3.1. Concept of SAGE. The problem associated with the signal model (2) consists in estimating the parameters $\psi_l = [\gamma_l, \theta_l, \varphi_l, \tau_l, \nu_l]^T$, $l = 0, 1, \ldots, L-1$ for all the paths. The estimation of L is not discussed in this paper. Usually, L is fixed to a value large enough to capture all the dominant impinging waves. Classical information theory methods for model size selection like Akaike's and Rissanen's [19] criteria can be used. The ML estimation is given by $\widehat{\psi} = \arg\max_{\psi} p(\mathbf{Z} \mid \psi)$, where $\mathbf{Z}$ is the sampled complex baseband signal and $p(\mathbf{Z} \mid \psi)$ is the likelihood function. The direct maximization of the likelihood function is a computationally prohibitive task since there is no analytical solution. Moreover, $p(\mathbf{Z} \mid \psi)$ is generally not a concave function of ψ, and L is usually high. To perform this optimization, we use the iterative process of the SAGE algorithm [16]. The basic concept of the SAGE algorithm is to us a hidden data space. Instead of estimating the parameters of all impinging waves in one search, the SAGE algorithm sequentially estimates the parameters of each signal. The SAGE algorithm breaks down the multidimensional optimization problem into several smaller problems. In spite of this complexity reduction, SAGE is still hardly implemented in real time due to the size of the baseband signal.

3.2. The SAGE/STAP Multicorrelator Algorithm. The main idea to take advantages of SAGE with more reasonable hardware requirements is to process the data at the output of the STAP multicorrelator array. Using the signal model (10), the likelihood function is then

$$p(\mathbf{x} \mid \Psi) = \frac{1}{\pi^{NPM} \det \mathrm{Q}_{pc}} \exp\left(-[\mathbf{x} - \mathbf{x}_{si}(\Psi)]^H \mathrm{Q}_{pc}^{-1} [\mathbf{x} - \mathbf{x}_{si}(\Psi)]\right), \quad (11)$$

where $\mathbf{x}_{si}(\Psi) = \sum_{l=0}^{L-1} \mathbf{x}_l(\Psi_l)$ contains the superimposition of the post-correlation signals. Q_{pc} denotes the covariance matrix of the postcorrelation noise. The noise is no longer white after the correlation step, and it is shown in [20] that the postcorrelation noise covariance matrix is given by

$$\mathrm{Q}_{pc} = E\left(\mathbf{n}_{pc}\mathbf{n}_{pc}^H\right) = \widetilde{\sigma}_n^2 (\mathbf{I}_{MN} \otimes \mathrm{Q}_P), \quad (12)$$

where $\widetilde{\sigma}_n^2$ denotes the noise power after the correlation step, and Q_P is define thanks to the auto-correlation function of the PRN code $r(\cdot)$:

$$\mathrm{Q}_P = \begin{bmatrix} r(0) & r(Cs) & \cdots & \cdots & r(P\cdot Cs) \\ r(Cs) & r(0) & r(Cs) & & \vdots \\ \vdots & r(Cs) & r(0) & r(Cs) & \vdots \\ \vdots & & r(Cs) & r(0) & r(Cs) \\ r(P\cdot Cs) & \cdots & \cdots & r(Cs) & r(0) \end{bmatrix}_{P\times P}. \quad (13)$$

The first step of the SAGE algorithm, so-called expectation step (E-STEP), consists in estimating the hidden data space with

$$\widehat{\mathbf{x}}_l = \mathbf{x} - \sum_{\substack{l'=0 \\ l' \neq l}}^{L-1} \mathbf{x}_{l'}(\Psi_{l'}). \quad (14)$$

The second step, so-called maximization step (M-STEP), carries out the maximization of the log-likelihood function which is associated with the estimated hidden data space. In the case of the STAP multicorrelator signal, the log-likelihood function is

$$\Lambda(\Psi_l) = \ln p(\widehat{\mathbf{x}}_l \mid \Psi_l) = -\ln \pi^{NPM} - \ln \det \mathrm{Q}_{pc} - (\mathbf{x}_l(\Psi_l) - \widehat{\mathbf{x}}_l)^H \mathrm{Q}_{pc}^{-1} (\mathbf{x}_l(\Psi_l) - \widehat{\mathbf{x}}_l). \quad (15)$$

The maximization of Λ with respect to Ψ_l can be concentrated, as the dependence to $\widetilde{\gamma}_l$ is linear:

$$\widetilde{\gamma}_l = \frac{\left[\mathbf{a}\left(\widehat{\theta}_l,\widehat{\varphi}_l\right) \otimes \check{\mathbf{r}}_C(\widehat{\tau}_{rl},\widehat{\nu}_{rl})\right]^H \widehat{\mathbf{x}}_l}{M \check{\mathbf{r}}_C^H(\widehat{\tau}_{rl},\widehat{\nu}_{rl}) \widetilde{\mathbf{r}}_C(\widehat{\tau}_{rl},\widehat{\nu}_{rl})}, \quad (16)$$

where $\check{\mathbf{r}}_C^H(\tau_{rl},\nu_{rl})$ is given by

$$\check{\mathbf{r}}_C^H(\tau_{rl},\nu_{rl}) = \widetilde{\mathbf{r}}_C^H(\tau_{rl},\nu_{rl})\left[\mathbf{I}_N \otimes \mathrm{Q}_P^{-1}\right]. \quad (17)$$

Finally, the reduced likelihood function to maximize is

$$\widetilde{\Lambda}(\Psi_l) = \frac{\left|\left[\mathbf{a}^H(\theta_l,\varphi_l) \otimes \check{\mathbf{r}}_C^H(\tau_{rl},\nu_{rl})\right]\hat{\mathbf{x}}_l\right|^2}{M\check{\mathbf{r}}_C^H(\tau_{rl},\nu_{rl})\widetilde{\mathbf{r}}_C(\tau_{rl},\nu_{rl})}. \tag{18}$$

In order to better understand the expression (18), let us assume no noise is present, and we can develop (10) as

$$\begin{aligned}\mathbf{x}_l &= \widetilde{\gamma}_l \mathbf{a}(\theta_l,\varphi_l) \otimes \widetilde{\mathbf{r}}_C(\tau_{rl},\nu_{rl}) \\ &= \widetilde{\gamma}_l \mathbf{a}(\theta_l,\varphi_l) \otimes \mathbf{e}(\nu_{rl}) \otimes \mathbf{r}(\tau_{rl}),\end{aligned} \tag{19}$$

where

$$\mathbf{e}(\nu_{rl}) = \left[\exp(j2\pi\nu_{rl}T_{\text{int}}) \cdots \exp(j2\pi\nu_{rl}NT_{\text{int}})\right]_{N\times 1}^T, \tag{20}$$

$$\mathbf{r}(\tau_{rl}) = \left[r\left(\tau_{rl} - \frac{(P-1)}{2}Cs\right) \cdots r\left(\tau_{rl} + \frac{(P-1)}{2}Cs\right)\right]_{P\times 1}^T. \tag{21}$$

Then, inserting (19), (20), and (21) in (18), we can express the reduced likelihood function as

$$\begin{aligned}&\widetilde{\Lambda}(\theta,\varphi,\tau_r,\nu_r) \\ &= |\widetilde{\gamma}_l|^2 \underbrace{\frac{|\mathbf{a}^H(\theta,\varphi)\mathbf{a}(\theta_l,\varphi_l)|^2}{M}}_{\text{space FT}} \underbrace{\frac{|\mathbf{e}^H(\nu_r)\mathbf{e}(\nu_{rl})|^2}{N}}_{\text{frequency FT}} \\ &\quad \times \frac{\left|\mathbf{r}^H(\tau_r)\mathrm{Q}_P^{-1}\mathbf{r}(\tau_{rl})\right|^2}{\mathbf{r}^H(\tau_r)\mathrm{Q}_P^{-1}\mathbf{r}(\tau_r)}.\end{aligned} \tag{22}$$

As we can see, the first two terms are similar to space and frequency Fourier transforms. Thus, the two parameters M and N will mainly influence the accuracy of the DOA and the Doppler estimations for both algorithms (SAGE and SAGE/STAP). However, the couple $\{P, Cs\}$ influences only the performances of the SAGE/STAP algorithm with respect to the paths delay estimation. This point will be detailed in more detail in Section 4.

3.3. Implementation in a GNSS Receiver. To implement the SAGE/STAP algorithm in a GNSS receiver, we propose the architecture illustrated in Figure 2.

The first front end bloc contains the RF filters, the down-conversion step, the ADC and a calibration stage [21]. A calibration stage aims to compensate the technological defects such as the RF dispersion of the filters and antenna defects.

The baseband signal $\mathbf{Z}$ is the output of this blocx. We must therefore use the STAP multicorrelator array for each tracked PRN code in order to compress the data. Afterward, we use the SAGE algorithm in the processing bloc. Finally, the SAGE estimation of the relative delay and Doppler τ_{r0}, ν_{r0} is used to drive the DLL and PLL/FLL. Thus, the key idea of our solution is to substitute the conventional DLL and PLL/FLL discriminators by the SAGE/STAP estimation.

4. Simulation Results

The aim of this section is to study the impact of the compression step (realized by the STAP multicorrelator array) on the SAGE algorithm performances. Thus, this section focuses on the comparison between the classical SAGE and the SAGE/STAP multicorrelator algorithms. The estimation performances of both algorithms were numerically evaluated with Monte Carlo simulations and theoretically through the CRB derivation. The derivation of the CRB for the SAGE/STAP algorithm is given in the annexe. For the conventional SAGE algorithm, the CRB can be found in [17]. The simulations parameters are as follows: $\sigma_n^2 = -131$ dBW, $\gamma_{0\,\text{dB}} = -160$ dBW, $\theta_0 = 61^\circ$, $\varphi_0 = 131^\circ$, $\upsilon_0 = 100$ Hz, and we use $N = 20$ blocks of 1 ms of integration. The sampling frequency of the baseband signal is $f_s = 10$ MHz. The 2×2 square array antenna contains 4 isotropic identical sensors spaced by $\lambda/2$.

In a first time, we analyze the influence of P (the number of correlators) and Cs (the correlators' delay spacing) on the performances of the SAGE/STAP algorithm. In order to cover all the cross-correlation function, we should have $PCs > 2T_c$. This condition is necessary if we want to take into account all the multipaths with a relative delay $\tau_{rl} \in [0, T_c]$. Therefore, we have the following relationship $P = \text{ceil}(2T_c/Cs)$, where the operator $\text{ceil}(x)$ denotes the nearest integer greater than or equal to x. We now focus our attention on the Cs parameters.

To study the influence of Cs, we calculate the root-mean-square error (RMSE) of the time delay based on 100 Monte Carlo simulations for both algorithms (SAGE and SAGE/STAP multicorrelator). In this first simulation, we assume no multipath, and the RMSE is plotted on Figure 3 as a function of Cs, for different signal bandwidths: $B = 2$ MHz and $B = 4$ MHz, where B denotes the two-sided RF bandwidth. When comparing the RMSE of SAGE and SAGE/STAP algorithms, we can see that both algorithms present the same performances in the case $Cs \leq 1/B$. This behavior is confirmed theoretically with the curve of the CRB as a function of Cs. We can see that the standard deviation increases when Cs becomes higher than the inverse of the signal bandwidth. We can deduce from this observation the following empirical rule which results from a tradeoff between accuracy and complexity:

$$P = \text{ceil}(2T_c/Cs), \quad Cs = \frac{1}{B} \approx \frac{1}{f_s}. \tag{23}$$

This conclusion can also be observed on the shape of the likelihood function. In Figures 4 and 5, we plot a section of the likelihood function through the delay for the SAGE and SAGE/STAP multicorrelator algorithms. In other words, we compare $|\mathbf{r}^H(\tau_r)\Sigma_P^{-1}\mathbf{r}(\tau_r)|$ and $|r(\tau_r)|^2$ for different signal bandwidths ($B = 2$ MHz on Figure 4 and $B = 4$ MHz on Figure 5) and different correlator spacings Cs. We can see that if the condition (23) is fulfilled, the likelihood functions are identical for the SAGE and SAGE/STAP multicorrelator algorithms. If not, the likelihood function of the SAGE/STAP algorithm becomes wider than the classical SAGE likelihood

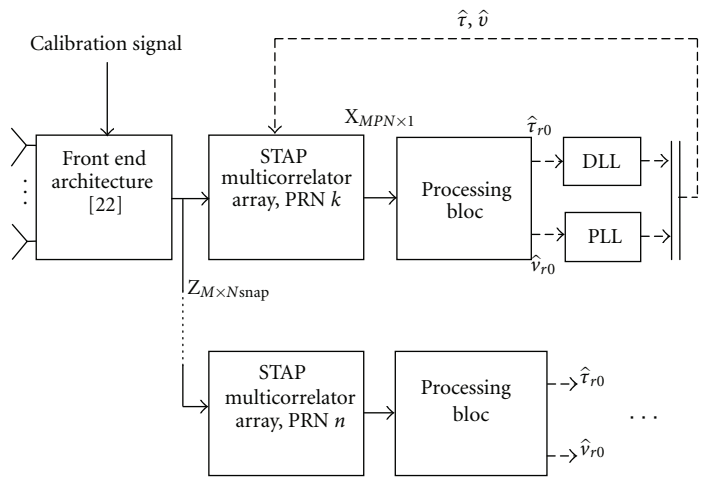


Figure 2: Proposed receiver architecture.

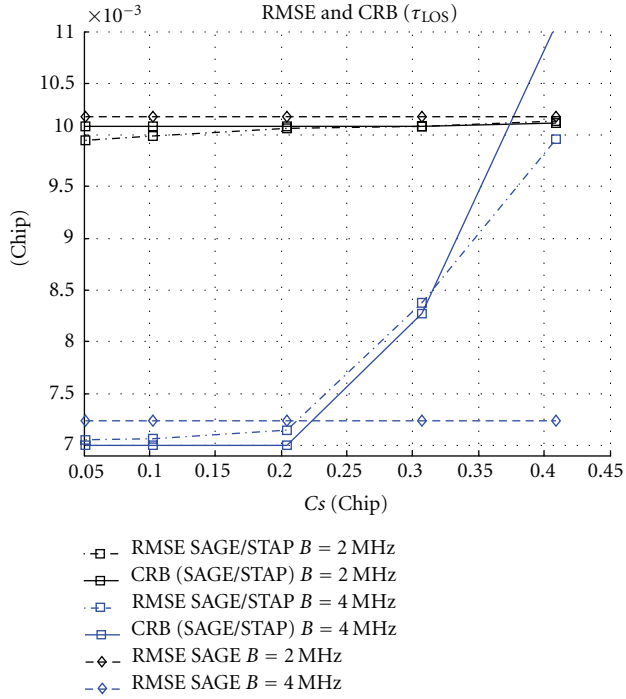


Figure 3: RMSE for the SAGE algorithm and RMSE and CRB for the SAGE/STAP algorithm. RMSE and CRB are plotted as a function of Cs, and for 2 different signal bandwidths: $B = 2$ MHz (black) and $B = 4$ MHz (blue).

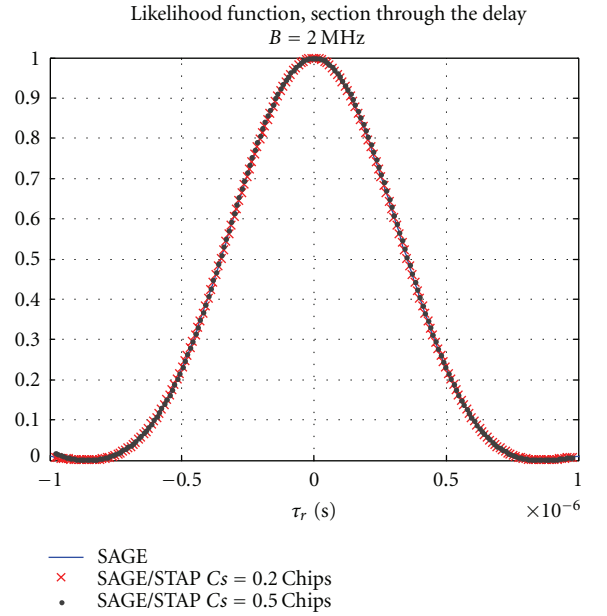


Figure 4: Likelihood function section through the delay for the SAGE and SAGE/STAP algorithms. $B = 2$ MHz and $Cs = 0.1$, 0.2 and 0.5 Chips.

function, which will consequently increase the variance of the estimation.

In a second time, we evaluate the performances of both algorithms in the case of one multipath. The reflected multipath and the LOSS are considered to be in-phase which corresponds to one of the worst possible cases. In Figure 6, we plot the RMSE of the estimated time delay for both algorithms as a function of the relative time delay and relative azimuth between the LOSS and the MP (denoted $\Delta\tau$ and $\Delta\varphi$). The multipath parameters are $\gamma_{1\,\text{dB}} = -163$ dBW, $\theta_1 = 30^\circ$, $v_1 = 105$ Hz.

We assume that the receiver is perfectly synchronized with the LOSS at the beginning of the simulation. Therefore the initial relative parameters are $v_{r0} = 0$ Hz, $v_{r1} = 5$ Hz, $\tau_{r0} = 0$ Chip. In order to choose the couple $\{P, Cs\}$, we use the results of the first simulation together with the condition (23): $f_s = B = 10$ MHz, $Cs = 1/B = 0.1$ Chip, and $P = 2T_c/Cs = 20$. Here again, we can see that both algorithms present the same performances whatever the multipath position (in space and delay). We also plot in Figure 7 the CRB

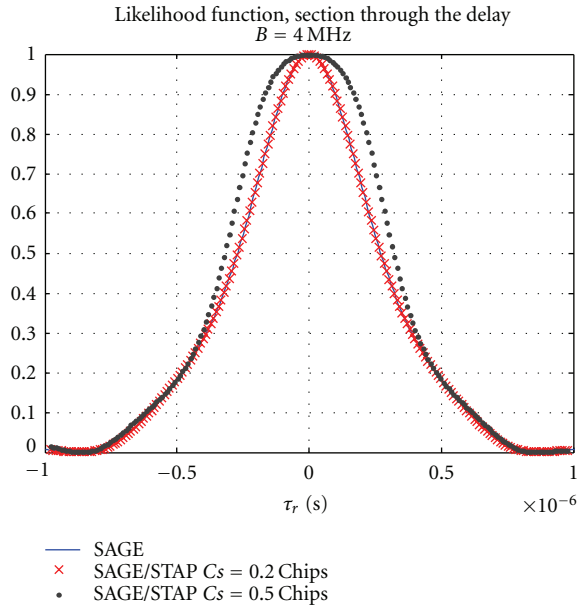


Figure 5: Likelihood function section through the delay for the SAGE and SAGE/STAP algorithms. $B = 4$ MHz and $Cs = 0.1$, 0.2 and 0.5 Chips.

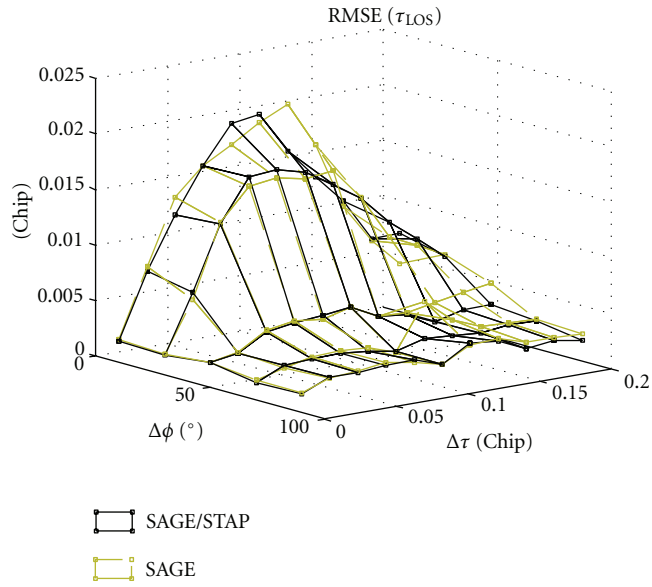


Figure 6: CRB of the LOSS delay as a function of the relative azimuth and relative delay, for the SAGE and the SAGE/STAP algorithms. $\Delta\gamma_{dB} = -3$ dB, $\Delta\theta = 31°$, $\Delta v = -5$ Hz.

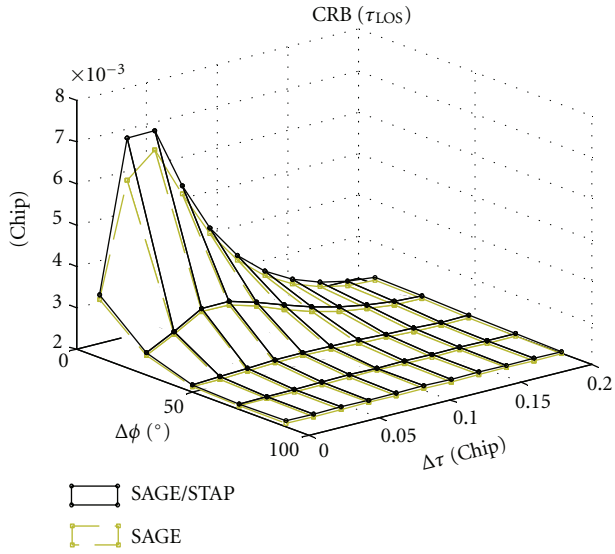


Figure 7: RMSE of the LOSS delay as a function of the relative azimuth and relative delay, for the SAGE and the SAGE/STAP algorithms. $\Delta\gamma_{dB} = -3$ dB, $\Delta\theta = 31°$, $\Delta v = -5$ Hz.

for the SAGE and SAGE/STAP algorithms. The behaviour of both CRB shows that the algorithms are theoretically equivalent. Thus, the SAGE/STAP multicorrelator can reach the efficiency of the SAGE algorithm but with a strongly reduced complexity. Indeed, the size of the baseband signal is $MfsNT_{int}$ compared to $MNP = MN2T_c f_s$ for the postcorrelated signal in the optimal configuration. The compression factor is $T_{int}/(2T_c) \approx 500$.

We compare now the RMSE and the CRB of the SAGE/STAP algorithm in Figure 8 in the same configuration. We can see that the RMSE reaches the CRB if the sources are spaced by more than 50°. This result is particularly interesting because in this situation, close and far time delay multipaths are completely mitigated by the algorithm, which is mainly due to the multiantenna contribution. In the case where the relative azimuth is weaker, the algorithms mainly use the time and frequency domains to mitigate the multipath. In this condition, the array antenna is not useful anymore, and the rejection performances are similar to single antenna mitigation techniques (e.g., MEDLL). In Figure 9, we plot the RMSE and the CRB for the LOSS elevation estimation. Here again, the RMSE reaches the CRB, excepted in the case of short-time delay ($\Delta\tau < 0.1$ Chip) and close space sources ($\Delta\varphi < 20°$). This shows the limitation of the SAGE approach for cases where the MP is close to the LOSS on the 4 dimensions considered in the estimation process.

The efficiency of the SAGE/STAP algorithm was illustrated in the previous simulations. Now, we want to address the impact of this algorithm on the performances of conventional GNSS tracking loops.

First of all, we evaluate the error envelope for a conventional DLL enhanced by a conventional beamformer (CBF) and for a DLL driven by SAGE (denoted DLL/SAGE). The loops parameters are loop order = 2, damping factor = 0.707, $B_{DLL} = 1$ Hz, $B_{PLL} = 10$ Hz. In the case of classical DLL, we use the early-minus-late discriminator (EMLD) with a chips spacing equal to 0.1 Chip and the arctangent discriminator for the PLL. In Figure 10, we plot the error envelope as a function of the relative azimuth and delay of the MP. Although the DLL is enhanced by a CBF, the SAGE/STAP algorithm provides the best multipath mitigation whatever the MP position.

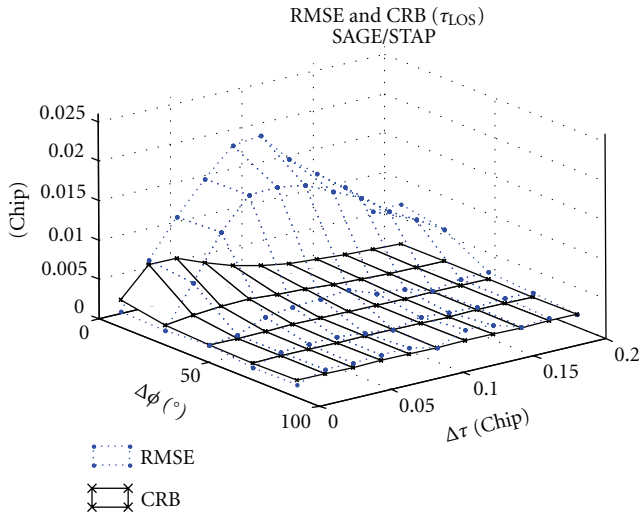


Figure 8: RMSE and CRB of the LOSS delay as functions of the relative azimuth and relative delay, for the SAGE/STAP algorithms. $\Delta\gamma_{dB} = -3$ dB, $\Delta\theta = 31^\circ$, $\Delta v = -5$ Hz.

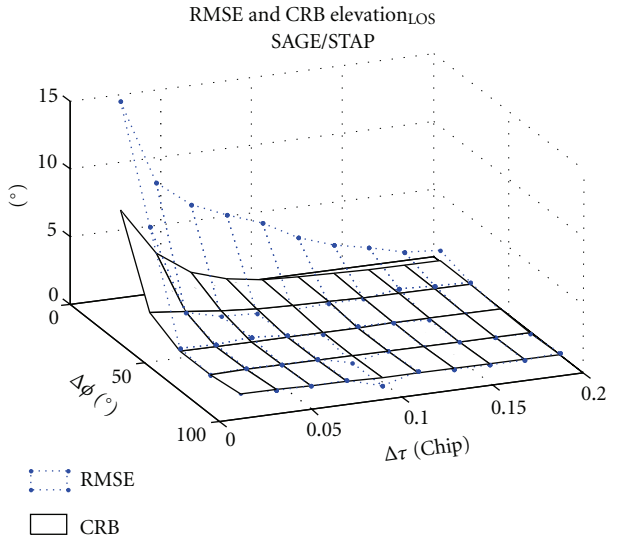


Figure 9: RMSE and CRB of the LOSS elevation as functions of the relative azimuth and relative delay, for the SAGE/STAP algorithms. $\Delta\gamma_{dB} = -3$ dB, $\Delta\theta = 31^\circ$, $\Delta v = -5$ Hz.

In the last simulation, we propose to test the SAGE/STAP algorithm with the DLR channel model available online (http://www.kn-s.dlr.de/satnav/) [22]. The DLR channel model generates the number of paths (LOSS + MP), their complex amplitudes, relative delays, and Doppler shifts. In the free version available online, the DOA are not generated. Thus, we computed the DOA of the MP thanks to the position of the scatterers, which are randomly positioned based on statistics established during measurement campaigns [22].

The LOSS parameters are $\theta_0 = 43.7^\circ$, $\varphi_0 = -88.61$, $v_0 = 395.75$ Hz, and we simulated a CBOC signal [23] on the L1 band with $B = f_s = 25$ MHz. Thus, according to (23),

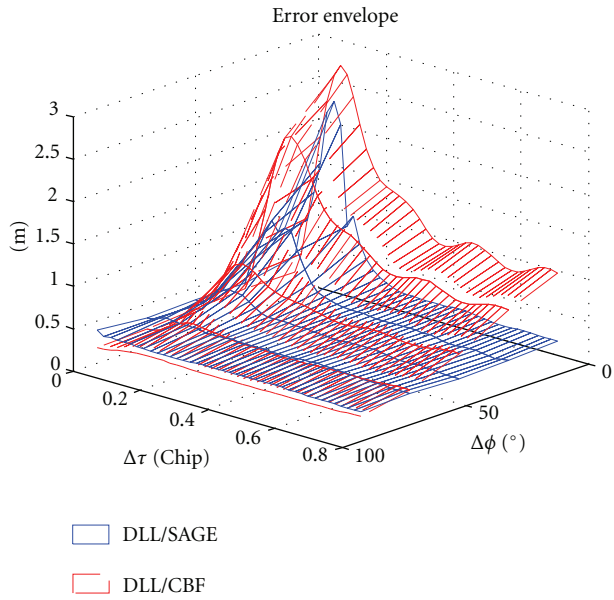


Figure 10: Error envelope for the DLL driven by SAGE and the DLL enhanced by a conventional beamforming. The envelope error is plotted as a function of the relative delay and azimuth. $\Delta\gamma_{dB} = -3$ dB, $\Delta\theta = 31^\circ$, $\Delta v = -5$ Hz.

we have $P = 49$ and $Cs = 40$ ns. The speed of the vehicle has been chosen to 0 m/s in order to tackle the problem of stationary MP. The other simulation parameters (N, DLL and PLL) are the same as in the previous simulations. In Figure 11, we plot the pseudorange error and the Doppler estimation of the conventional DLL/PLL and the DLL/PLL driven by the SAGE/STAP multicorrelators algorithm. As we can see, the SAGE approach provides a real improvement in the time delay and Doppler estimation with respect to the conventional multiantenna receiver architecture in a tracking scenario. More tracking simulations with this solution can be found in [24].

5. Conclusion

In this work we have addressed the problem of estimating the propagation time delay of the LOSS in a GNSS receiver under severe multipath conditions. To reduce the influence of the multipaths, we investigated the use of array antenna algorithms. Previous studies suggest using the SAGE algorithm with an array antenna in order to reduce the estimation error of the delay of the LOSS. However, SAGE is hardly implemented in real time due to the memory requirements and computation cost. Moreover, SAGE is hardly compatible with classical GNSS tracking loops.

In order to take the advantages of SAGE with more reasonable hardware requirements, we proposed a new implementation based on a STAP multicorrelator array. The CRB derivation and the Monte Carlo simulations were used in this paper to study the trade-off between accuracy and complexity. This trade-off appears by the influence of the parameters P (number of correlators) and Cs (correlator space) on

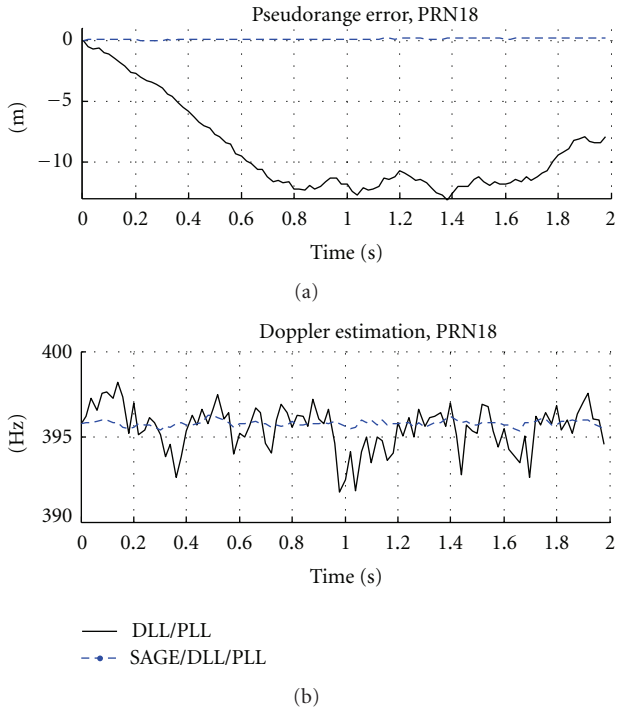


FIGURE 11: Pseudorange error and Doppler estimation for the conventional DLL/PLL and the DLL/PLL driven by the SAGE algorithm. $\theta_0 = 43.7^\circ$, $\varphi_0 = -88.61$, $\upsilon_0 = 395.75$ Hz.

the SAGE/STAP multicorrelator performances. The size of the data to be processed is then drastically reduced, and the CRB and Monte Carlo simulations attest that there is no performance loss between the classical SAGE algorithm and the SAGE/STAP multicorrelator algorithm. Moreover, the SAGE/STAP multicorrelator algorithm can constitute the discriminator of GNSS tracking loops. Thus, with this new implementation, the high-resolution performances of SAGE are now available for GNSS receiver, which is a very promising approach for the multipath mitigation problem in GNSS.

Appendix

Cramer Rao Bound

The likelihood function for the signal at the output of the STAP multicorrelator array is given in (11). Therefore, the log-likelihood function is

$$\ln p(\mathbf{x} \mid \Psi_l) = -\ln \pi^{NPM} - \ln \det \mathrm{Q}_{pc} - (\mathbf{x} - \mathbf{x}_{si}(\Psi))^H \mathrm{Q}_{pc}^{-1} (\mathbf{x} - \mathbf{x}_{si}(\Psi)), \tag{A.1}$$

where $\mathbf{x}_{si}(\Psi) = \sum_{l=1}^{L} \mathbf{x}_l(\Psi_l)$ contains the superimposition of the postcorrelated signals, Σ_{pc} denotes the covariance matrix of the postcorrelated noise, and $\Psi = [|\widetilde{\gamma}|^T \arg(\widetilde{\gamma})^T \theta^T \phi^T {\nu_r}^T {\tau_r}^T]_{5L}$ is the parameters vector with $\widetilde{\gamma} = [\widetilde{\gamma}_0, \widetilde{\gamma}_1, \ldots, \widetilde{\gamma}_{L-1}]^T_{L\times 1}$ the modified complex amplitude vector of the L impinging wave fronts, $\theta = [\theta_0, \ldots, \theta_{L-1}]^T$ the elevation vector, $\phi = [\varphi_0, \ldots, \varphi_{L-1}]^T$ the azimuth vector, $\nu_r = [\nu_{r0}, \ldots, \nu_{rL-1}]^T$ the relative Doppler vector, and $\tau_r = [\tau_{r0}, \ldots, \tau_{rL-1}]^T$ the vector of relative delay.

The CRB is found as the $[i, i]$ element of the inverse of the so-called Fisher Information Matrix (FIM) $\mathbf{I}(\Psi)$:

$$\mathrm{var}(\Psi_i) \geq \left[\mathbf{I}(\Psi)^{-1}\right]_{ii}. \tag{A.2}$$

The definition of the FIM is

$$[\mathbf{I}(\Psi)]_{i,j} = -E\left[\frac{\partial^2 \ln p(\mathbf{x} \mid \Psi)}{\partial \Psi_i \partial \Psi_j}\right]. \tag{A.3}$$

As we assume that the noise covariance is independent of the parameters, the FIM for a complex multivariate Gaussian process is [6, 11]

$$[\mathbf{I}(\Psi)]_{i,j} = 2\,\mathrm{Re}\left(\frac{\partial \mathbf{x}_{si}(\Psi)^H}{\partial \Psi_i} \mathrm{Q}_{pc}^{-1} \frac{\partial \mathbf{x}_{si}(\Psi)}{\partial \Psi_j}\right). \tag{A.4}$$

Note that the FIM can be a bloc partitioned symmetric matrix. Thus, we can write

$$\mathbf{I}(\Psi) = \begin{bmatrix} \mathbf{I}_{|\gamma|} & \mathbf{I}_{|\gamma|\arg(\gamma)} & \mathbf{I}_{|\gamma|\theta} & \mathbf{I}_{|\gamma|\phi} & \mathbf{I}_{|\gamma|\tau_r} & \mathbf{I}_{|\gamma|\nu_r} \\ & \mathbf{I}_{\arg(\gamma)} & & & & \\ & & \mathbf{I}_\theta & \ddots & & \\ \vdots & & & \mathbf{I}_\phi & & \vdots \\ & & \ddots & & \mathbf{I}_{\tau_r} & \\ \mathbf{I}_{\nu_r|\gamma|} & & & & & \mathbf{I}_{\nu_r} \end{bmatrix}_{6L}, \tag{A.5}$$

where the bloc matrices can be obtained by

$$\begin{aligned} \left[\mathbf{I}_{|\gamma|}\right]_{n,p} &= 2\,\mathrm{Re}\left(\frac{\partial \mathbf{x}_n^H}{\partial |\widetilde{\gamma}_n|} \mathrm{Q}_{pc}^{-1} \frac{\partial \mathbf{x}_p}{\partial |\widetilde{\gamma}_p|}\right), \\ \left[\mathbf{I}_{\arg(\gamma)}\right]_{n,p} &= 2\,\mathrm{Re}\left(\frac{\partial \mathbf{x}_n^H}{\partial \arg(\widetilde{\gamma}_n)} \mathrm{Q}_{pc}^{-1} \frac{\partial \mathbf{x}_p}{\partial \arg(\widetilde{\gamma}_p)}\right), \\ [\mathbf{I}_\theta]_{n,p} &= 2\,\mathrm{Re}\left(\frac{\partial \mathbf{x}_n^H}{\partial \theta_n} \mathrm{Q}_{pc}^{-1} \frac{\partial \mathbf{x}_p}{\partial \theta_p}\right), \\ \left[\mathbf{I}_\phi\right]_{n,p} &= 2\,\mathrm{Re}\left(\frac{\partial \mathbf{x}_n^H}{\partial \varphi_n} \mathrm{Q}_{pc}^{-1} \frac{\partial \mathbf{x}_p}{\partial \theta_p}\right), \\ [\mathbf{I}_\nu]_{n,p} &= 2\,\mathrm{Re}\left(\frac{\partial \mathbf{x}_n^H}{\partial \nu_{rn}} \mathrm{Q}_{pc}^{-1} \frac{\partial \mathbf{x}_p}{\partial \nu_{rp}}\right), \\ [\mathbf{I}_\tau]_{n,p} &= 2\,\mathrm{Re}\left(\frac{\partial \mathbf{x}_n^H}{\partial \tau_{rn}} \mathrm{Q}_{pc}^{-1} \frac{\partial \mathbf{x}_p}{\partial \tau_{rp}}\right), \\ \left[\mathbf{I}_{|\gamma|\theta}\right]_{n,p} &= 2\,\mathrm{Re}\left(\frac{\partial \mathbf{x}_n^H}{\partial |\widetilde{\gamma}_n|} \mathrm{Q}_{pc}^{-1} \frac{\partial \mathbf{x}_p}{\partial \theta_p}\right), \\ &\vdots \end{aligned} \tag{A.6}$$

with $1 \leq n \leq L, 1 \leq p \leq L$. Last, we need to calculate the differential of the STAP multicorrelator signal model given

in (10) with respect to $|\widetilde{\gamma}_l|, \arg(\widetilde{\gamma}_l), \theta_l, \varphi_l, \nu_{rl}, \tau_{rl}$. To that end, we use the following signal model:

$$\mathbf{x}_l = \widetilde{\gamma}_l \mathbf{a}(\theta_l, \varphi_l) \otimes \mathbf{e}(\nu_{rl}) \otimes \mathbf{r}(\tau_{rl}). \tag{A.7}$$

The differential of the Kronecker product of 2 matrixes $\mathbf{A}_{p\times q}$ and $\mathbf{B}_{r\times l}$ with respect to a parameter x is given by [6]

$$\begin{aligned}\frac{\partial}{\partial x}[\mathbf{A}(x) \otimes \mathbf{B}(x)] &= \frac{\partial \mathbf{A}(x)}{\partial x} \otimes \mathbf{B}(x) \\ &\quad + \mathbf{U}_{p\times r}\left(\frac{\partial \mathbf{B}(x)}{\partial x} \otimes \mathbf{A}(x)\right)\mathbf{U}_{l\times q},\end{aligned} \tag{A.8}$$

with $\mathbf{U}$ the permutation matrix. Note that in our model, we are working with column vectors, and each subvector is depending on different independent parameters. Therefore, the differential of the signal is simply given by

$$\begin{aligned}\frac{\partial \mathbf{x}_l}{\partial \theta_l} &= \widetilde{\gamma}_l \frac{\partial \mathbf{a}(\theta_l, \varphi_l)}{\partial \theta_l} \otimes \mathbf{e}(\nu_{rl}) \otimes \mathbf{r}(\tau_{rl}), \\ \frac{\partial \mathbf{x}_l}{\partial \varphi_l} &= \widetilde{\gamma}_l \frac{\partial \mathbf{a}(\theta_l, \varphi_l)}{\partial \varphi_l} \otimes \mathbf{e}(\nu_{rl}) \otimes \mathbf{r}(\tau_{rl}), \\ \frac{\partial \mathbf{x}_l}{\partial \nu_{rl}} &= \widetilde{\gamma}_l \mathbf{a}(\theta_l, \varphi_l) \otimes \frac{\partial \mathbf{e}(\nu_{rl})}{\partial \nu_{rl}} \otimes \mathbf{r}(\tau_{rl}), \\ \frac{\partial \mathbf{x}_l}{\partial \tau_{rl}} &= \widetilde{\gamma}_l \mathbf{a}(\theta_l, \varphi_l) \otimes \mathbf{e}(\nu_{rl}) \otimes \frac{\partial \mathbf{r}(\tau_{rl})}{\partial \tau_{rl}}, \\ \frac{\partial \mathbf{x}_l}{\partial |\widetilde{\gamma}_l|} &= \exp(j \arg(\widetilde{\gamma}_l))\mathbf{a}(\theta_l, \varphi_l) \otimes \mathbf{e}(\nu_{rl}) \otimes \mathbf{r}(\tau_{rl}) \\ &= \exp(-j\pi\nu_r T_{\text{int}})\mathbf{a}(\theta_l, \varphi_l) \otimes \mathbf{e}(\nu_{rl}) \otimes \mathbf{r}(\tau_{rl}), \\ \frac{\partial \mathbf{x}_l}{\partial \arg(\widetilde{\gamma}_l)} &= j\widetilde{\gamma}_l \mathbf{a}(\theta_l, \varphi_l) \otimes \mathbf{e}(\nu_{rl}) \otimes \mathbf{r}(\tau_{rl}).\end{aligned} \tag{A.9}$$

Acknowledgments

This work was supported by the French Space Agency (CNES), French Aerospace Lab (ONERA), and Thales Alenia Space—France.

References

[1] R. D. Nee, "Spread-spectrum code and carrier synchronization errors caused by multipath and interference," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 4, pp. 1359–1365, 1993.

[2] B. W. Parkinsin and J. J. Spilker, Eds., *Global Positioning System: Theory and Application*, vol. 1, Progress in Astronautics and Aeronautics, Washington, DC, USA , 1996.

[3] A. J. Van Dierendonck, P. Fenton, and T. Ford, "Theory and performance of narrow correlator spacing in a GPS receiver," *Navigation, Journal of the Institute of Navigation*, vol. 39, no. 3, pp. 265–283, 1992.

[4] R. D. J. van Nee, J. Siereveld, P. C. Fenton, and B. R. Townsend, "Multipath estimating delay lock loop: approaching theoretical accuracy limits," in *Proceedings of the 1994 IEEE Position Location and Navigation Symposium*, pp. 246–251, April 1994.

[5] M. Lentmaier, B. Krach, and P. Robertson, "Bayesian time delay estimation of GNSS signals in dynamic multipath environments," *International Journal of Navigation and Observation*, vol. 2008, Article ID 372651, 11 pages, 2008.

[6] P. Closas, C. Fernández-Prades, and J. A. Fernández-Rubio, "A particle filtering tracking algorithm for GNSS synchronization using Laplace's method," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*, pp. 3409–3412, Las Vegas, Nv, USA, April 2008.

[7] B. Krach, M. Lentmaier, and P. Robertson, "Joint Bayesian positioning and multipath mitigation in GNSS," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 3437–3440, Las Vegas, Nv, USA, April 2008.

[8] A. Konovaltsev, F. Antreich, and A. Hornbostel, "Performance assessment of antenna array algorithms for multipath and interference mitigation," in *Proceedings of the 2nd Workshop on GNSS Signals & Signal Processing (GNSS SIGNALS'07)*, 2007.

[9] A. K. Brown and B. Mathews, "GPS multipath mitigation using a three dimensional phased array," in *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS '05)*, pp. 659–666, long Beach, Calif, USA, September 2005.

[10] S. Rougerie, G. Carrie, L. Ries et al., "Multipath mitigation methods based on antenna array," in *Proceedings of the International Technical Meeting of Institute of Navigation (ITM '11)*, pp. 596–605, San Diego, Calif, USA, January 2011.

[11] J. H. Williams, R. J. Davis, and E. N. Rosario, "Multipath mitigation performance of planar GPS adaptive antenna for precision landing ground stations," in *Proceedings of the Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS'00)*, pp. 1309–1316, Salt Lake City, Utah, USA, 2000.

[12] B. R. Rao, J. H. Williams, E. N. Rosario, and R. J. Davis, "microstrip antenna array on a resistivity tapered ground plane for multipath mitigation," in *Proceedings of the Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS'00)*, pp. 2468–2476, Salt Lake City, Utah, USA, 2000.

[13] J. K Ray, M. E. Cannon, and P. Fenton, "Mitigation of GPS code and carrier phase multipath effects using a multi-antenna system," UCGE Reports, 1999.

[14] G. Seco-Granados, J. A. Fernández-Rubio, and C. Fernández-Prades, "ML estimator and hybrid beamformer for multipath and interference mitigation in GNSS receivers," *IEEE Transactions on Signal Processing*, vol. 53, no. 3, pp. 1194–1208, 2005.

[15] J. Selva, "An efficient Newton-type method for the computation of ML estimators in a uniform linear array," *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 2036–2045, 2005.

[16] B. H. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. I. Pedersen, "Channel parameter estimation in mobile radio environments using the SAGE algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 3, pp. 434–450, 1999.

[17] F. Antreich, J. A. Nossek, and W. Utschick, "Maximum likelihood delay estimation in a navigation receiver for aeronautical applications," *Aerospace Science and Technology*, vol. 12, no. 3, pp. 256–267, 2008.

[18] S. Rougerie, A. Konovaltsev, M. Cuntz et al., "Comparison of SAGE and classical multi-antenna algorithms for multipath mitigation in real-world environment," in *Proceedings of the 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing*

(NAVITEC'10), pp. 1–8, ESTEC, Noordwijk, Netherland, 2010.

[19] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proceedings of the 2nd International Symposium on Information theory*, B. Petrov and F. Csaki, Eds., pp. 267–281, Budapest, Hungary, 1971.

[20] G. Carrie, F. Vincent, T. Deloues, D. Pietin, A. Renard, and F. Letestu, "Optimal STAP algorithms to GNSS receivers," in *Proceedings of the European Navigation Conference*, Manchester International Convention Centre, 2006.

[21] A. Konovaltsev, N. Basta, L. A. Greda, M. Cuntz, M. V. T. Heckler, and A. Dreher, "Calibration of adaptive antennas in satellite navigation receivers," in *Proceedings of the 4th European Conference on Antennas and Propagation, (EuCAP '10)*, April 2010.

[22] A. Lehner and A. Steingass, "A novel channel model for land mobile satellite navigation," in *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation, (ION GNSS '05)*, pp. 2132–2138, Long Beach, Calif, USA, September 2005.

[23] "Galileo open service, signal in space interface control document, draft," European Space Agency / European GNSS supervisory Authority, 2008.

[24] S. Rougerie, G. Carrié, L. Ries, F. Vincent, M. Monnerat, and M. Bousquet, "A new tracking approach for GNSS multipath mitigation based on antenna array," in *Proceedings of the Ka and Broadband Communications, Navigation and Earth Observation Conference*, Palermo, Italy, 2011.