sensors

MDPI

Article

Evaluation of Adaptive Loop-Bandwidth Tracking Techniques in GNSS Receivers †

Iñigo Cortés [1,*], Johannes Rossouw van der Merwe [1], Jari Nurmi [2], Alexander Rügamer [1] and Wolfgang Felber [1]

[1] Satellite Based Positioning Systems Department, Fraunhofer IIS, Nordostpark 84, 90411 Nuremberg, Germany; Johannes.roussouw.vandermerwe@iis.fraunhofer.de (J.R.v.d.M.); alexander.ruegamer@iis.fraunhofer.de (A.R.); wolfgang.felber@iis.fraunhofer.de (W.F.)
[2] Electrical Engineering, Tampere University, 33014 Tampere, Finland; jari.nurmi@tuni.fi
* Correspondence: inigo.cortes@iis.fraunhofer.de; Tel.: +49-911-58061-6426

† This paper is an extended version of our paper published in Cortés, I.; Iñiguez de Gordoa, J.A.; van der Merwe, J.R.; Rügamer, A.; Felber, W. Performance and complexity comparison of adaptive loop-bandwidth tracking techniques. In Proceedings of the 2020 International Conference on Localization and GNSS (ICL-GNSS), Tampere, Finland, 2–4 June 2020; pp. 1–7.

Abstract: Global navigation satellite system (GNSS) receivers use tracking loops to lock onto GNSS signals. Fixed loop settings limit the tracking performance against noise, receiver dynamics, and the current scenario. Adaptive tracking loops adjust these settings to achieve optimal performance for a given scenario. This paper evaluates the performance and complexity of state-of-the-art adaptive scalar tracking techniques used in modern digital GNSS receivers. Ideally, a tracking channel should be adjusted to both noisy and dynamic environments for optimal performance, defined by tracking precision and loop robustness. The difference between the average tracking jitter of the discriminator's output and the square-root Cramér-Rao bound (CRB) indicates the loops' tracking capability. The ability to maintain lock characterizes the robustness in highly dynamic scenarios. From a system perspective, the average lock indicator is chosen as a metric to measure the performance in terms of precision, whereas the average number of visible satellites being tracked indicates the system's robustness against dynamics. The average of these metrics' product at different noise levels leads to a reliable system performance metric. Adaptive tracking techniques, such as the fast adaptive bandwidth (FAB), the fuzzy logic (FL), and the loop-bandwidth control algorithm (LBCA), facilitate a trade-off for optimal performance. These adaptive tracking techniques are implemented in an open software interface GNSS hardware receiver. All three methods steer a third-order adaptive phase locked loop (PLL) and are tested in simulated scenarios emulating static and high-dynamic vehicular conditions. The measured tracking performance, system performance, and time complexity of each algorithm present a detailed analysis of the adaptive techniques. The results show that the LBCA with a piece-wise linear approximation is above the other adaptive loop-bandwidth tracking techniques while preserving the best performance and lowest time complexity. This technique achieves superior static and dynamic system performance being 1.5 times more complex than the traditional tracking loop.

Keywords: global navigation satellite system (GNSS); adaptive scalar tracking loop (A-STL); fast adaptive bandwidth (FAB); fuzzy logic (FL); loop-bandwidth control algorithm (LBCA); piece-wise linear approximation of non-linearities (PLAN)

check for updates

Citation: Cortés, I.; van der Merwe, J.R.; Nurmi, J.; Rügamer, A.; Felber, W. Evaluation of Adaptive Loop-Bandwidth Tracking Techniques in GNSS Receivers. *Sensors* **2021**, *21*, 502. https://doi.org/10.3390/s21020502

Received: 14 December 2020
Accepted: 9 January 2021
Published: 12 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A global navigation satellite system (GNSS) receiver needs to maintain lock on the satellite signals in order to decode their navigation message and determine the pseudo-range [1]. Locking onto and following the signal's Doppler frequency, carrier phase, and code phase is referred to as tracking [2]. A GNSS receiver must utilize the best possible

means to track the signals since it leads to a better pseudo-range estimation, and in turn, to a more precise and reliable position, velocity, and time (PVT) solution. However, noise, receiver dynamics, and multi-path effects make this a challenging task.

Before tracking can be initiated, the signal must first be acquired: a coarse synchronization of the carrier and code is determined for each received GNSS signal [1]. The tracking uses this coarse estimation to initialize the scalar tracking loops (STLs). These loops improve the estimates by continuously monitoring the signal. They synchronize to the carrier frequency and Doppler offset with a frequency locked loop (FLL), the carrier phase with a phase locked loop (PLL), and the code phase with a delay locked loop (DLL). An STL tracks each of the received GNSS signals independently from the others. An alternative approach is to track and adapt the channels jointly with vector tracking (VT) [3], but currently there are several hardware implementation issues that currently limit practical use [4]. Therefore, VT is not considered for the remainder of the paper. The conventional digital tracking loop consists of a correlator, a discriminator, a loop filter, and a numerically controlled oscillator (NCO) [5]. The integration time τ_{int}, the correlator spacing, the discriminator type, the order, the noise bandwidth of the loop filter, and the oscillator determine the tracking performance at a given carrier-to-noise density ratio (C/N_0).

Depending on the scenario, different loop settings are ideal. Dynamics stress the tracking loop's accuracy, whereas noise limits precision [6,7]. The former is characterized by changes in position, velocity, acceleration, jerk, or even higher orders of dynamics. The latter arises due to the thermal noise, the oscillator phase noise, quantization noise, signal interference, and other noise sources [8]. Multi-path, fading effects, and scintillation also shape the noise and introduce biases [9]. The loop bandwidth of conventional STLs presents a trade-off between the ability to stay locked in the presence of dynamics and the amount of filtered noise to achieve better precision [6]. A large loop bandwidth is ideal for dynamic scenarios, as it presents tracking lock robustness, but results in low noise rejection and limits precision. A small loop bandwidth is suitable for noisy scenarios, as it rejects most noise, but dynamics can easily disrupt the loop and break the lock. Therefore, loop bandwidth tuning directly impacts tracking loop performance. However, it is impossible to choose a single, static value ideal for all scenarios and signals. Further, the geometry, path-loss, and environment between the receiver and each satellite in the constellation directly impact each signal's dynamics and noise. Hence, even in a single scenario, each signal has an ideal loop bandwidth, which requires individualized tuning.

Adaptive loop-bandwidth tracking techniques address this problem by adapting the loop bandwidth of the STL depending on the scenario conditions [7]. These techniques determine a suitable loop bandwidth based on the tracking channel's behavior. One method is the fast adaptive bandwidth (FAB) technique [10–13]. It uses a loop bandwidth dependent cost function, which serves to estimate the loop bandwidth in the next iteration. A second method is the fuzzy logic (FL) technique [14,15]: This technique updates the loop bandwidth based on a fuzzy decision of adequate inputs. The inputs can be any tracking channel information such as the discriminator's output and its statistics. A third technique is the loop-bandwidth control algorithm (LBCA) [16]. This algorithm updates the loop bandwidth performing a weighted difference of estimated noise and estimated signal dynamics. Alternative tracking methods with Kalman filtering (KF) are promising and have gained popularity in the literature [17–23]; however, this approach still has some hardware implementation limitations and will not be considered for the remainder of the paper.

This paper expands a conference paper [7] by including the theory of adaptive tracking techniques and improving the scope of the results. It implements various adaptive loop-bandwidth tracking techniques on a GNSS receiver and verifies them. Furthermore, it presents a method to evaluate the performance and complexity of these techniques. The FAB, the FL, and the LBCA are implemented in the GOOSE© receiver [24,25]. The GOOSE receiver is a GNSS receiver with an open software interface. This receiver is tested in simulated scenarios with different dynamics and noise cases. This paper aims to achieve

a static and dynamic system performance comparison between different adaptive tracking techniques. The system performance is also compared with the complexity of each implemented algorithm. It leads to a complete evaluation of the implemented adaptive tracking techniques.

The rest of the paper is organized as follows. Section 2 presents a detailed STL background, followed by a precise description of the implemented variable loop-bandwidth tracking techniques in Section 3. Section 4 analyzes the metric used to evaluate the tracking and system performance and evaluates the complexity of adaptive loop-bandwidth tracking techniques. The experimental setup and implementation in an open software interface GNSS hardware receiver are described in Section 5. Section 6 presents the results of the adaptive loops. These results are synthesized and discussed in Section 7. Finally, Section 8 concludes and indicates future work.

2. Scalar Tracking Loops

This section presents an introduction to standard digital STLs. Two models explain the operation of the STL. First, the non-linear model in the discrete-time domain describes the main components of an STL. Second, a linear model in the z-domain addresses the STL transfer function and the STL discrete state space model (SSM). These two representations include sufficient background to describe the behavior of STLs fully.

2.1. Non-Linear Model

GNSS receivers commonly deploy FLLs, PLLs, and DLLs [1]. All three of these loops track a different signal parameter (i.e., Doppler frequency, carrier phase, or code phase) but have similar architectures. Figure 1 shows the non-linear STL model in the discrete-time domain [2]. The STL consists of a detector, a loop filter, an NCO, and a replica generator.

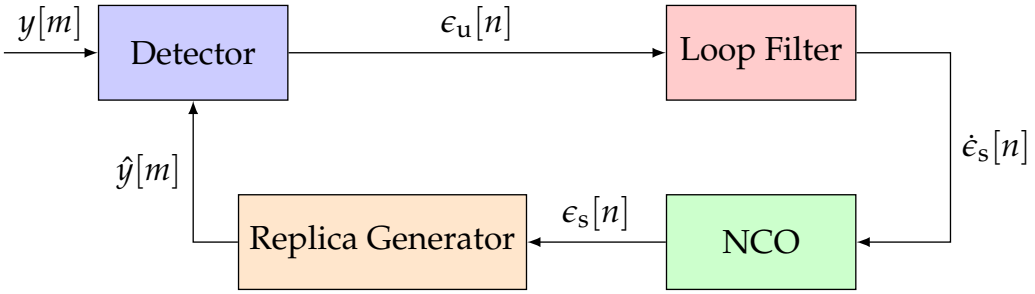


Figure 1. Architecture of a conventional scalar tracking loop (STL). ©IEEE. Adapted, with permission, from [7].

The first STL component is the detector. The detector calculates the error $\epsilon_u[n]$ of the representative tracking parameter. The error $\epsilon_u[n]$ is determined by comparing the incoming signal $y[m]$ and an estimated replica of the signal $\hat{y}[m]$. Figure 2 presents the general structure of a detector. A GNSS receiver correlates an input signal with a locally generated replica to retrieve the GNSS signal. Therefore, a multiplier and an integration and dump (IAD) stage correlate a vector on input data samples, to extract the correlation in-phase and quadrature-phase (IQ) samples. The correlator decimates the signal to a sample-rate related to the integration time τ_{int}, resulting in multi-rate processing. The noise (i.e., the variance of the error $\epsilon_u[n]$) is inversely proportional to the integration time τ_{int} of the IAD. Hence, the larger the integration time τ_{int}, the higher the processing gain—the less noisy the errors $\epsilon_u[n]$. However, with larger integration time, the loop response becomes slower, and consequently, it is less sensitive against signal dynamics. The last component of the detector is the discriminator function. It determines the error $\epsilon_u[n]$ for a specific signal parameter (i.e., Doppler frequency, carrier phase, or code phase) based on the correlation IQ samples. Each tracked signal parameter requires a different discriminator.

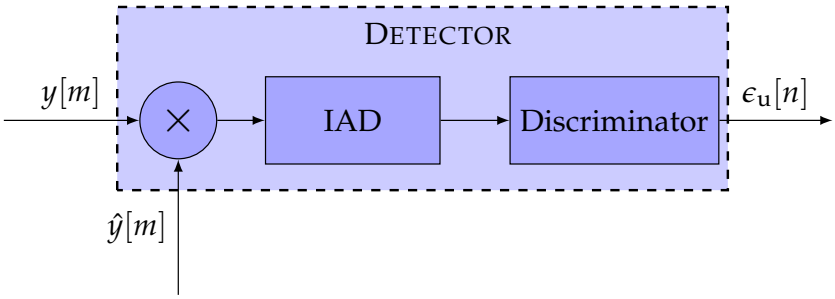


Figure 2. Architecture of the detector. ©IEEE. Adapted, with permission, from [7].

The second component of the STL is the loop filter. It shapes the noisy errors $\epsilon_n[n]$ to a smoothed error rate $\dot{\epsilon}[n]$. Hence, the filter input is an un-smoothed signal, and the filter output a smoothed one. The loop bandwidth affects the response time and noise suppression capabilities of the loop filter. Small loop bandwidths are ideal for suppressing unwanted noise, whereas large loop bandwidths respond quicker to signal dynamics. The loop filter is typically implemented as an infinite impulse response (IIR) filter [1,2,5]. The order of the IIR loop filter regulates the robustness of the tracking against high-order dynamics. A higher-order loop filter can track higher-order components of the error signal but adds complexity, and the possibility for instability in the system. The resulting smoothed error rate $\dot{\epsilon}_s[n]$ drives an NCO. The NCO integrates the output of the loop filter and sends the smoothed error $\epsilon_s[n]$ to the replica generator. The replica generator reconstructs a new estimate $\hat{y}[m]$ for the for the signal. The tracking loop minimizes the error $\epsilon_u[n]$ such that the generated replica signal $\hat{y}[m]$ converges to the input signal $y[m]$.

2.2. Linear Model

The linear model of an STL simplifies Figure 1 by removing the non-linear mapping of the discriminator and the multi-rate conversion of the correlator [2]. A discrete approximation of continuous-time systems is performed. This section considers the backward Euler transform (BET) [26,27] as it is used in current STL implementations. Figure 3 presents the linear model in the z-domain. The input signal $\epsilon[n]$ represents the carrier phase offset, the frequency Doppler, or the code phase offset of the input signal $y[m]$. The same functional blocks to Figure 1 are visible, but linearized.

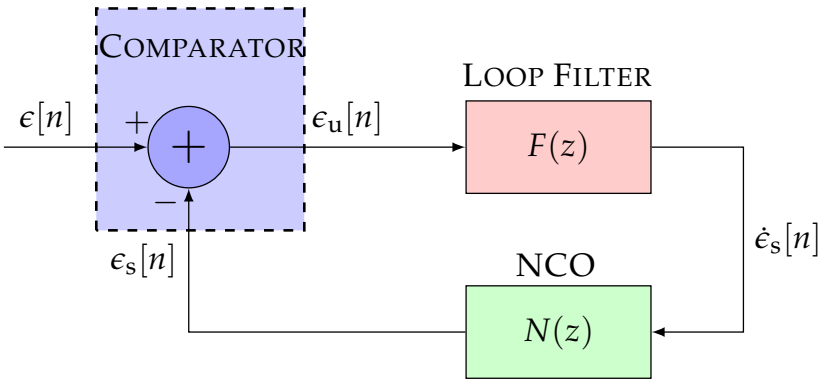


Figure 3. Linear model of the STL. ©IEEE. Adapted, with permission, from [7].

The STL transfer function is often analyzed without considering the SSM representation analysis. This hinders the comparison with other tracking methods that are commonly represented by its SSM representation (e.g., KF-based STLs). Therefore, it is necessary to consider the transfer function and SSM representation since they are valuable concepts that

serve to understand the STL behavior fully. From Figure 3, the controllable canonical form of the discrete SSM can be calculated [28]:

$$\frac{\mathbf{x}_{p\times 1}[n]-\mathbf{x}_{p\times 1}[n-1]}{\tau_{\text{int}}}=\underbrace{\begin{bmatrix}0&1&0&\cdots&0\\0&0&1&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\\0&0&0&\cdots&0\end{bmatrix}}_{\mathbf{A}_{\text{cont}_{p\times p}}}\mathbf{x}_{p\times 1}[n]+\underbrace{\begin{bmatrix}\alpha_{p-1}\\\alpha_{p-2}\\\vdots\\\alpha_0\end{bmatrix}}_{\mathbf{B}_{\text{cont}_{p\times 1}}}\epsilon_{\text{u}}[n] \tag{1}$$

$$\epsilon_{\text{s}}[n]=\underbrace{\begin{bmatrix}1&0&0&\cdots&0\end{bmatrix}}_{\mathbf{C}_{\text{cont}_{1\times p}}}\mathbf{x}_{p\times 1}[n]=x_1[n] \tag{2}$$

where $\mathbf{x}$ is the discrete state vector, p is the order of the STL, $\mathbf{A}_{\text{cont}}$ is the state matrix, $\mathbf{B}_{\text{cont}}$ is the input matrix, and $\mathbf{C}_{\text{cont}}$ is the output matrix of the SSM in the continuous-time domain. $\mathbf{B}_{\text{cont}}$ contains the coefficients α_k of the STL. Equation (1) can be further developed:

$$\mathbf{x}_{px1}[n]=\underbrace{(\mathbf{I}_{p\times p}-\mathbf{A}_{\text{cont}_{p\times p}}\tau_{\text{int}})^{-1}}_{\mathbf{A}_{\text{discr}_{p\times p}}}\mathbf{x}_{p\times 1}[n-1]+\underbrace{(\mathbf{I}_{p\times p}-\mathbf{A}_{\text{cont}_{p\times p}}\tau_{\text{int}})^{-1}\mathbf{B}_{\text{cont}_{p\times 1}}\tau_{\text{int}}}_{\mathbf{B}_{\text{discr}_{p\times 1}}}\epsilon_{\text{u}}[n] \tag{3}$$

where $\mathbf{I}_{p\times p}$ is the identity matrix, $\mathbf{A}_{\text{discr}}$ the discrete state matrix and $\mathbf{B}_{\text{discr}}$ the discrete control matrix.

Extending the previous equation, the following expression is achieved:

$$\mathbf{x}_{p\times 1}[n]=\begin{bmatrix}1&\tau_{\text{int}}&\tau_{\text{int}}^2&\cdots&\tau_{\text{int}}^{p-1}\\0&1&\tau_{\text{int}}&\cdots&\tau_{\text{int}}^{p-2}\\\vdots&\vdots&\ddots&\ddots&\vdots\\0&0&0&\cdots&\tau_{\text{int}}\\0&0&0&\cdots&1\end{bmatrix}\mathbf{x}_{p\times 1}[n-1]+\begin{bmatrix}\sum_{k=0}^{p-1}\alpha_k\tau_{\text{int}}^{p-k}\\\sum_{k=0}^{p-1}\alpha_k\tau_{\text{int}}^{p-k-1}\\\vdots\\\alpha_0\tau_{\text{int}}\end{bmatrix}\epsilon_{\text{u}}[n] \tag{4}$$

Performing the z-transform $\mathcal{Z}(\cdot)$ of Equations (2) and (4), the open loop transfer function $H_{\text{o}}(z)$ of the discrete SSM can be obtained:

$$\begin{aligned}H_{\text{o}}(z)=\frac{\mathcal{Z}(\epsilon_{\text{s}})}{\mathcal{Z}(\epsilon_{\text{u}})}&=\mathbf{C}_{\text{cont}_{1\times p}}\left(I_{p\times p}-\mathbf{A}_{\text{discr}_{p\times p}}z^{-1}\right)^{-1}\mathbf{B}_{\text{discr}_{p\times 1}}=\\&=\sum_{k=0}^{p-1}\frac{\alpha_k\tau_{\text{int}}^{p-k}}{(1-z^{-1})^{p-k}}=\underbrace{\sum_{k=0}^{p-1}\frac{\alpha_k\tau_{\text{int}}^{p-k-1}}{(1-z^{-1})^{p-k-1}}}_{F(z)}\underbrace{\frac{\tau_{\text{int}}}{1-z^{-1}}}_{N(z)}\end{aligned} \tag{5}$$

where $F(z)$ is the transfer function of the loop filter and $N(z)$ the transfer function of the NCO.

From the open loop transfer function, the closed loop transfer function $H_{\text{c}}(z)$ can be obtained:

$$H_{\text{c}}(z)=\frac{\mathcal{Z}(\epsilon_{\text{s}})}{\mathcal{Z}(\epsilon)}=\frac{H_{\text{o}}(z)}{1+H_{\text{o}}(z)}=\frac{F(z)\cdot N(z)}{1+F(z)\cdot N(z)} \tag{6}$$

The zeros and poles of the closed loop transfer function depend on the coefficients of the tracking loop α_k and the integration time τ_{int}. Hence, the one-sided normalized bandwidth B_N of the closed-loop transfer function is dependant on these coefficients:

$$2B_N(\alpha_0, \alpha_1, \cdots, \alpha_{p-1}) = 2B(\alpha_0, \alpha_1, \cdots, \alpha_{p-1})\tau_{\text{int}} = \frac{1}{2\pi \text{j}} \oint_{|z|=1} H_{\text{c}}(z) H_{\text{c}}(z^{-1}) z^{-1} dz \quad (7)$$

where B is the one-sided equivalent noise bandwidth.

The adaptive tracking techniques are implemented in a third-order STL. Therefore, the SSM and transfer function of a third-order STL are analyzed in Figure 4 displaying its linear model.

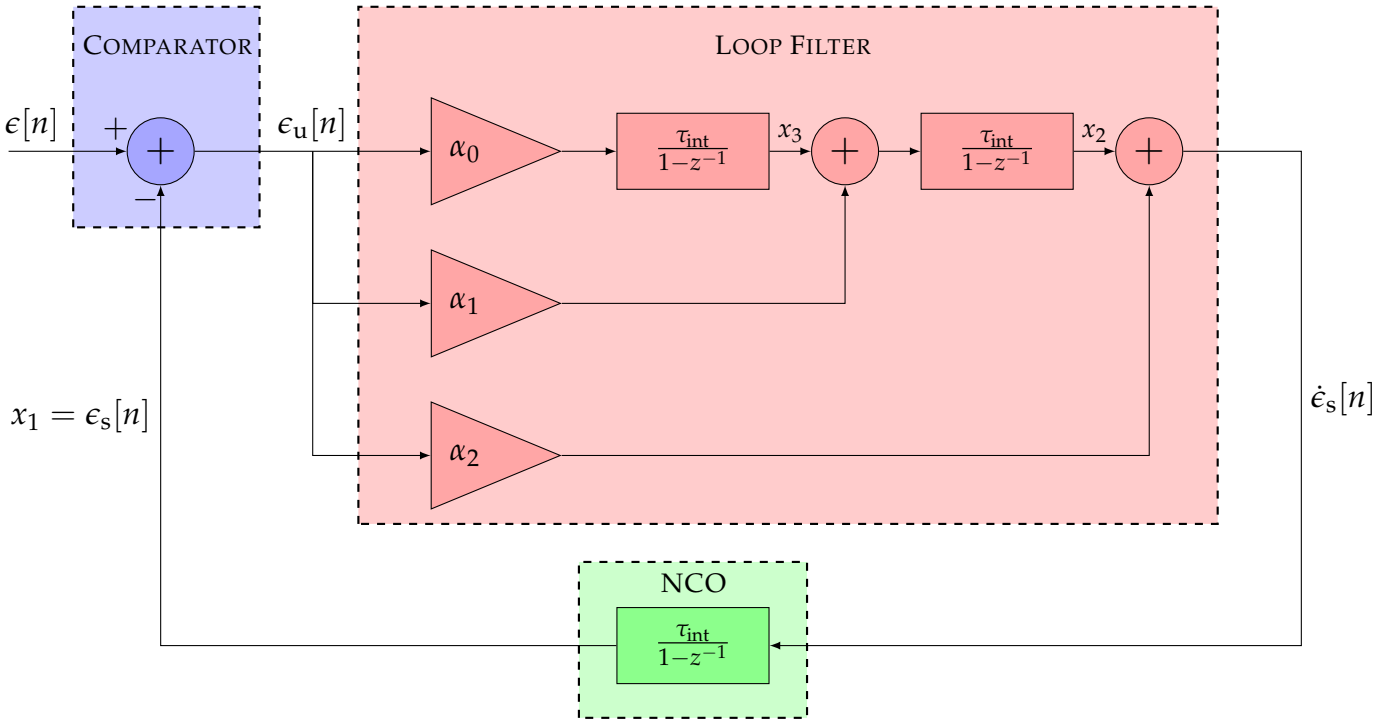


Figure 4. Linear model of a third-order STL.

The equivalent discrete SSM is:

$$\begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \end{bmatrix} = \begin{bmatrix} 1 & \tau_{\text{int}} & \tau_{\text{int}}^2 \\ 0 & 1 & \tau_{\text{int}} \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1[n-1] \\ x_2[n-1] \\ x_3[n-1] \end{bmatrix} + \begin{bmatrix} \alpha_2\tau_{\text{int}} + \alpha_1\tau_{\text{int}}^2 + \alpha_0\tau_{\text{int}}^3 \\ \alpha_1\tau_{\text{int}} + \alpha_0\tau_{\text{int}}^2 \\ \alpha_0\tau_{\text{int}} \end{bmatrix} \epsilon_{\text{u}}[n] \quad (8)$$

$$\epsilon_s[n] = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \end{bmatrix} = x_1[n] \quad (9)$$

The open loop transfer function is characterized as:

$$H_{o_3}(z) = \frac{\alpha_2\tau_{\text{int}}(1-z^{-1})^2 + \alpha_1\tau_{\text{int}}^2(1-z^{-1}) + \alpha_0\tau_{\text{int}}^3}{(1-z^{-1})^3} \quad (10)$$

and the closed loop transfer function as:

$$H_{c_3}(z) = \frac{\alpha_2\tau_{\text{int}}(1-z^{-1})^2 + \alpha_1\tau_{\text{int}}^2(1-z^{-1}) + \alpha_0\tau_{\text{int}}^3}{(1-z^{-1})^3 + \alpha_2\tau_{\text{int}}(1-z^{-1})^2 + \alpha_1\tau_{\text{int}}^2(1-z^{-1}) + \alpha_0\tau_{\text{int}}^3} \tag{11}$$

If the integration time tends to zero ($\tau_{\text{int}} \rightarrow 0$), the analog one sided equivalent loop bandwidth is equal to the digital one [29–31]. The relation between the analog loop bandwidth and the third-order tracking loop coefficients is well-known [1,32]:

$$B = \frac{\alpha_2^2\alpha_1 - \alpha_2\alpha_0 + \alpha_1^2}{4(\alpha_2\alpha_1 - \alpha_0)} \tag{12}$$

A relation between these coefficients can simplify the previous expression [1]:

$$\mathbf{B}_{\text{cont}_{3\times1}} = \begin{bmatrix}\alpha_2\\ \alpha_1\\ \alpha_0\end{bmatrix} = \begin{bmatrix}2.4\cdot\omega\\ 1.1\cdot\omega^2\\ \omega^3\end{bmatrix} \tag{13}$$

where ω is the so-called natural frequency. Further developing Equation (12), the relation between the loop bandwidth B and ω is achieved:

$$B = \frac{\omega}{0.7845} \tag{14}$$

A change of the loop bandwidth B leads to an update of the natural frequency w that, in turn, reconfigures the loop filter coefficients (i.e., the control matrix $\mathbf{B}_{\text{cont}_{3\times1}}$). The following section shows three methods that adapt the loop bandwidth of an STL sub-optimally.

3. Variable Loop-Bandwidth Tracking Techniques

Variable loop-bandwidth tracking techniques set a connection between the loop bandwidth and time-varying scenario conditions [7]. In dynamic scenarios, a fast loop response with a large loop bandwidth is preferred to follow the dynamics, whereas, in stationary scenarios, a noise-rejecting low loop bandwidth is appropriate. Variable loop-bandwidth tracking techniques adapt the loop bandwidth depending on the noise and signal dynamics of the tracking channel. They facilitate optimal operation for the tracking loops.

Figure 5 shows the general structure of this technique [33]. The loop bandwidth $B[n]$ and a set of measurements $R[n]$ are the inputs to the adaptive loop-bandwidth estimator. These techniques estimate a loop bandwidth $\hat{B}[n]$. Since the set of measurements is noisy, a Schmitt trigger module reduces the noise instabilities of the loop bandwidth estimate.

The Schmitt trigger only changes the next loop bandwidth $B[n+1]$ by ΔB if the absolute difference between the estimated loop bandwidth and the actual one exceed ΔB:

$$B[n+1] = \begin{cases} 0 & \text{if } n = 0 \\ \hat{B}[n] + \Delta B & \text{if } \hat{B}[n] - B[n] \geq \Delta B \quad \wedge \quad n > 0 \\ \hat{B}[n] - \Delta B & \text{if } B[n] - \hat{B}[n] \leq \Delta B \quad \wedge \quad n > 0 \\ B[n] & \text{otherwise} \end{cases} \tag{15}$$

A big ΔB value (e.g., 10 Hz) could unstable the tracking due to big loop bandwidth changes, but a very low value (e.g., 0.01 Hz) could also cause instabilities due to excessive changes. 0.5 Hz has been found an appropriate value for ΔB. Finally, the updated loop bandwidth $B[n+1]$ changes the coefficients of the STL's transfer function (see Equations (13) and (14)).

This section presents three techniques from this category: the FAB, the FL, and the LBCA.

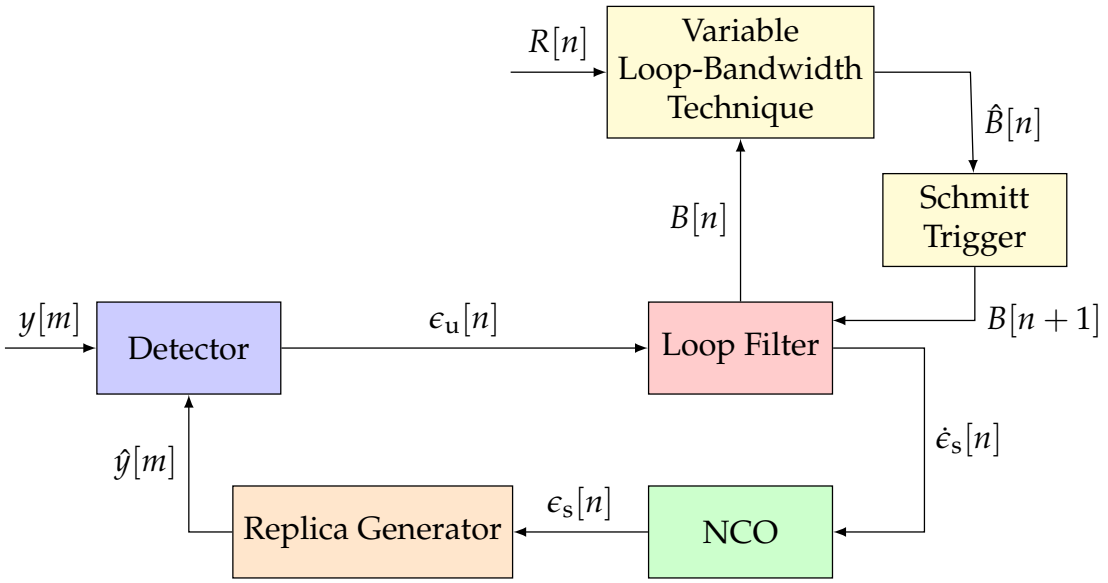


Figure 5. General architecture of the implemented variable loop-bandwidth tracking techniques.

3.1. Fast Adaptive Bandwidth (FAB)

The FAB tracking technique is a model-based approach and estimates the input signal parameters (thermal noise, phase noise, or steady state error (SSE)) of the STL [33]. A model is used to define a loop-bandwidth dependent cost function $c[n, B[n]]$. Setting the first derivative of the cost function to zero with respect to the loop bandwidth $B[n]$, leads to the minimum loop bandwidth B_{min} to which the STL can handle the estimated dynamics while filtering as much noise as possible.

The three-sigma rule-of-thumb of the tracking loop error is a commonly selected cost function [6]:

$$c[n, B[n]] = \sigma_{\epsilon}^{s}[n, B[n]] = \sigma_{\text{thermal}}^{s}[n, B[n]] + \frac{\epsilon_{\text{sse}}[n, B[n]]}{3} \tag{16}$$

where σ_{ϵ}^{s} is the jitter of the smoothed error, $\sigma_{\text{thermal}}^{s}$ is the thermal noise, and ϵ_{sse} is the dynamic stress error. Since σ_{ϵ}^{s} is loop-bandwidth dependent, the first order derivative of σ_{ϵ}^{s} regarding the loop bandwidth can be performed. Setting the derivative to zero, the minimum loop bandwidth B_{min} is estimated [6]:

$$\frac{\partial \sigma_{\epsilon}^{s}[n, B[n]]}{\partial B[n]} = 0 \rightarrow B[n] = B_{\text{min}} \tag{17}$$

The estimated minimum loop bandwidth B_{min} depends on the linear C/N_0, the integration time τ_{int}, the order p, and the steady state line-of-sight (LOS) dynamics $\frac{\partial^p R}{\partial t^p}$.

As an example, if the FAB is implemented in a third-order PLL, the following cost function is achieved equalling to zero the first derivative of the smoothed carrier phase error jitter σ_{θ}^{s}:

$$B_{\text{min}} = \sqrt[7]{\frac{\left(2\eta^3 \frac{\partial^3 R}{\partial t^3}\right)^2}{\frac{1}{C/N_o}\left(1 + \frac{1}{2\tau_{\text{int}} C/N_o}\right)\left(\frac{360}{2\pi}\right)^2}} \tag{18}$$

In this case, B_{min} is dependent on the linear C/N_0 in Hz, the integration time τ_{int} in seconds and the LOS jerk dynamics $\frac{\partial^3 R}{\partial t^3}$ in deg/s^3. For a third-order PLL, the constant η equals to 0.7845. The C/N_0 and steady state LOS dynamics must be estimated, whereas the other parameters are fixed.

Several considerations must be taken into account when implementing the FAB. First, the noisy output of the discriminator makes it difficult to measure the SSE correctly (even more for higher order loops). One solution is to accumulate the noisy measurements

for a large interval [10] or to implement a dynamic error filter [11]. Second, an abrupt change of the estimated optimal loop bandwidth $B_{\min}$ may create tracking instabilities, and needs to be minimized. An empirical scaling factor for the dynamic stress estimator and a constrained loop bandwidth addresses this issue [11]. Another solution is to smooth the loop bandwidth update using the Newton-Raphson method and an IIR filter [12].

Figure 6 shows the structure of the implemented three-sigma rule-of-thumb based FAB: a C/N_0 estimator, a dynamic stress estimator, the FAB, a gradient decent, an IIR filter and a threshold limiter. Each component is explained in the subsequent paragraphs.

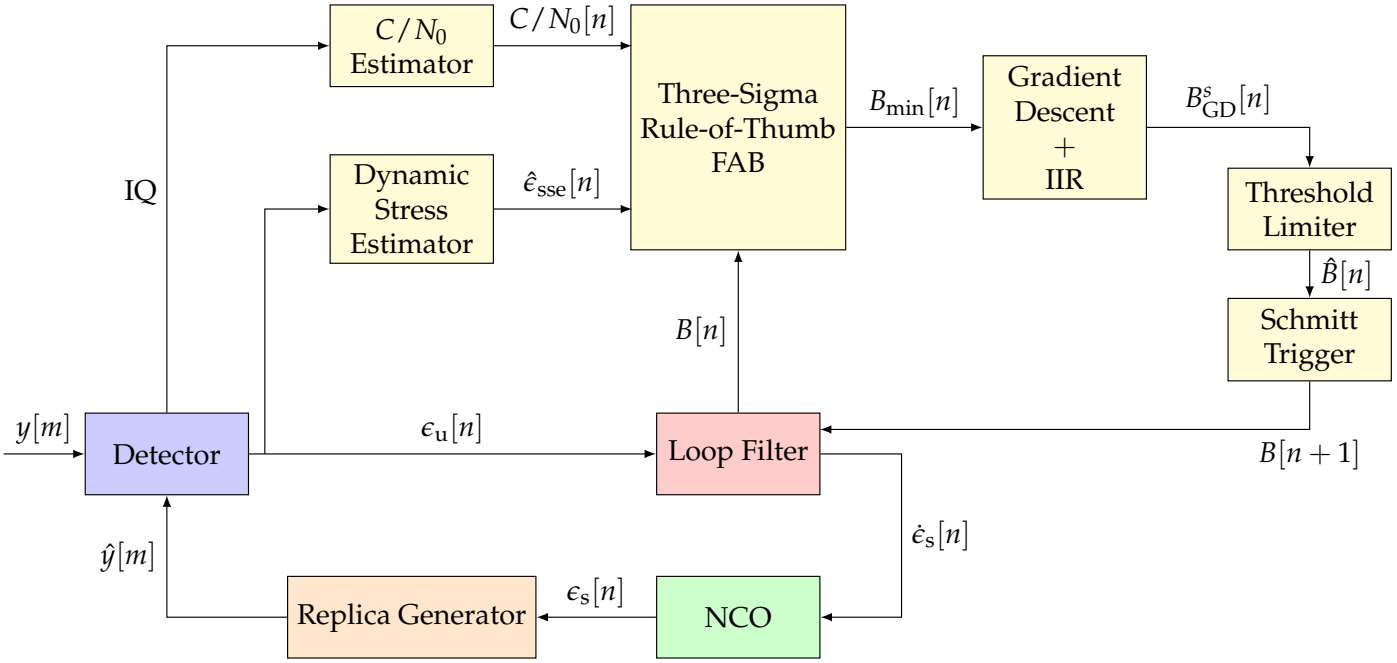


Figure 6. Architecture of the three-sigma rule-of-thumb based Fast Adaptive Bandwidth (FAB). ©IEEE. Adapted, with permission, from [7].

The C/N_0 estimator updates each tracking epoch. The dynamic stress estimator first filters the discriminator's output. The filter is a first order IIR filter with a decay time of Δt. The value of Δt represents the algorithm's sensitivity to dynamics. Second, it performs the p–th derivative of the filtered discriminator's output μ_ϵ^u in order to achieve an estimated SSE LOS dynamic measurement $\hat{\epsilon}_{\text{sse}}$:

$$\hat{\epsilon}_{\text{sse}} = \frac{\sum_{i=0}^{p} (-1)^i \binom{p}{i} \mu_\epsilon^u[n-i]}{\Delta t^p} \tag{19}$$

where $\binom{p}{i}$ is the binomial coefficient. Since the un-smoothed error is normalized, $\hat{\epsilon}_{\text{sse}}$ is in units of (cycles/s^p).

Continuing with the example of the third-order PLL, the dynamic stress estimator estimates the LOS jerk dynamics. This module filters the phase discriminator's output μ_θ^u and measures the angular jerk dynamics $\hat{\theta}_{\text{sse}}$:

$$\hat{\theta}_{\text{sse}}[n] = \frac{\mu_\theta^u[n] - 3\mu_\theta^u[n-1] + 3\mu_\theta^u[n-2] - \mu_\theta^u[n-3]}{\Delta t^3} \tag{20}$$

Next, the LOS jerk dynamics $\frac{\partial^3 R}{\partial t^3}$ in deg/s^3 is calculated as the following:

$$\frac{\partial^3 R}{\partial t^3}[n] = 360^\circ \times \hat{\epsilon}_{\text{sse}}[n] \tag{21}$$

Abrupt loop bandwidth changes may result in tracking instabilities. Therefore, the gradient descent method and a first order IIR filter smooth out the update. The gradient descent method updates the actual loop bandwidth $B[n]$ gradually:

$$B_{\text{GD}}[n] = \begin{cases} B[n] + \tau_{\text{int}} \cdot \frac{B_{\text{min}}[n] - B[n]}{|\Delta B_{\text{min}}[n]|} & \text{if } |\Delta B_{\text{min}}[n]| > 0.01 \\ B_{\text{min}}[n] & \text{otherwise} \end{cases} \tag{22}$$

where $\Delta B_{\text{min}}[n]$ is,

$$\Delta B_{\text{min}}[n] = B_{\text{min}}[n-1] - B_{\text{min}}[n] \tag{23}$$

The updated loop bandwidth $B_{\text{GD}}[n]$ depends on the actual loop bandwidth $B[n]$, the estimated minimum loop bandwidth $B_{\text{min}}[n]$ and the previous estimated loop bandwidth $B_{\text{min}}[n-1]$. To avoid instabilities, $B_{\text{GD}}[n]$ equals $B_{\text{min}}[n]$ when $|\Delta B_{\text{min}}[n]|$ is below 0.01 Hz.

Next, an IIR filter smooths $B_{\text{GD}}[n]$, achieving $B^s_{\text{GD}}[n]$. Finally, the filtered loop bandwidth is passed through a threshold limiter that constraints $B^s_{\text{GD}}[n]$ between a maximum B_{Max} and a minimum B_{Max} allowed loop bandwidth:

$$\hat{B}[n] = \begin{cases} B_{\text{Max}} & \text{if } B^s_{\text{GD}}[n]\tau_{\text{int}} > B_{\text{Max}}\tau_{\text{int}} \\ B^s_{\text{GD}}[n] & \text{if } B_{\text{Min}}\tau_{\text{int}} \leq B^s_{\text{GD}}[n]\tau_{\text{int}} \leq B_{\text{Max}}\tau_{\text{int}} \\ B_{\text{Min}} & \text{if } B^s_{\text{GD}}[n]\tau_{\text{int}} < B_{\text{Min}}\tau_{\text{int}} \end{cases} \tag{24}$$

Two main observations of the FAB can be made prior to implementation. First, the speed of the SSE estimation $\hat{\epsilon}_{\text{sse}}$ determines the speed of the algorithm to react against signal dynamics. The standard cost function (see Equation (16)) used does not consider other sources of signal dynamics such as clock drift and low order transient dynamics. Hence, it is possible that B_{min} decreases erroneously due to the fact that $\hat{\epsilon}_{\text{sse}}$ is negligible and other dynamic sources are not included. Second, the complexity of the algorithm can be significant due to the B_{min} estimation at high-order STLs.

3.2. Fuzzy Logic (FL)

Compared to the FAB technique, FL based tracking techniques significantly simplify the control algorithm. Figure 7 shows the implemented FL algorithm structure. The FL consists of four stages: pre-processing, fuzzy control rules, defuzzification, and post-processing [15]. The pre-processing stage converts the discriminator's output into two fuzzy sets: The standard deviation σ^u_ϵ and the absolute mean $|\mu^u_\epsilon|$. The former fuzzy input represents the tracking channel's noise, whereas the latter indicates the error dynamics. These fuzzy sets are finally normalized:

$$\tilde{N} = \frac{\sigma^u_\epsilon}{\sigma^u_\epsilon + |\mu^u_\epsilon|} \tag{25}$$

$$\tilde{D} = \frac{|\mu^u_\epsilon|}{\sigma^u_\epsilon + |\mu^u_\epsilon|} \tag{26}$$

where $\tilde{N}$ is the normalized estimated noise and $\tilde{D}$ the normalized estimated dynamics.

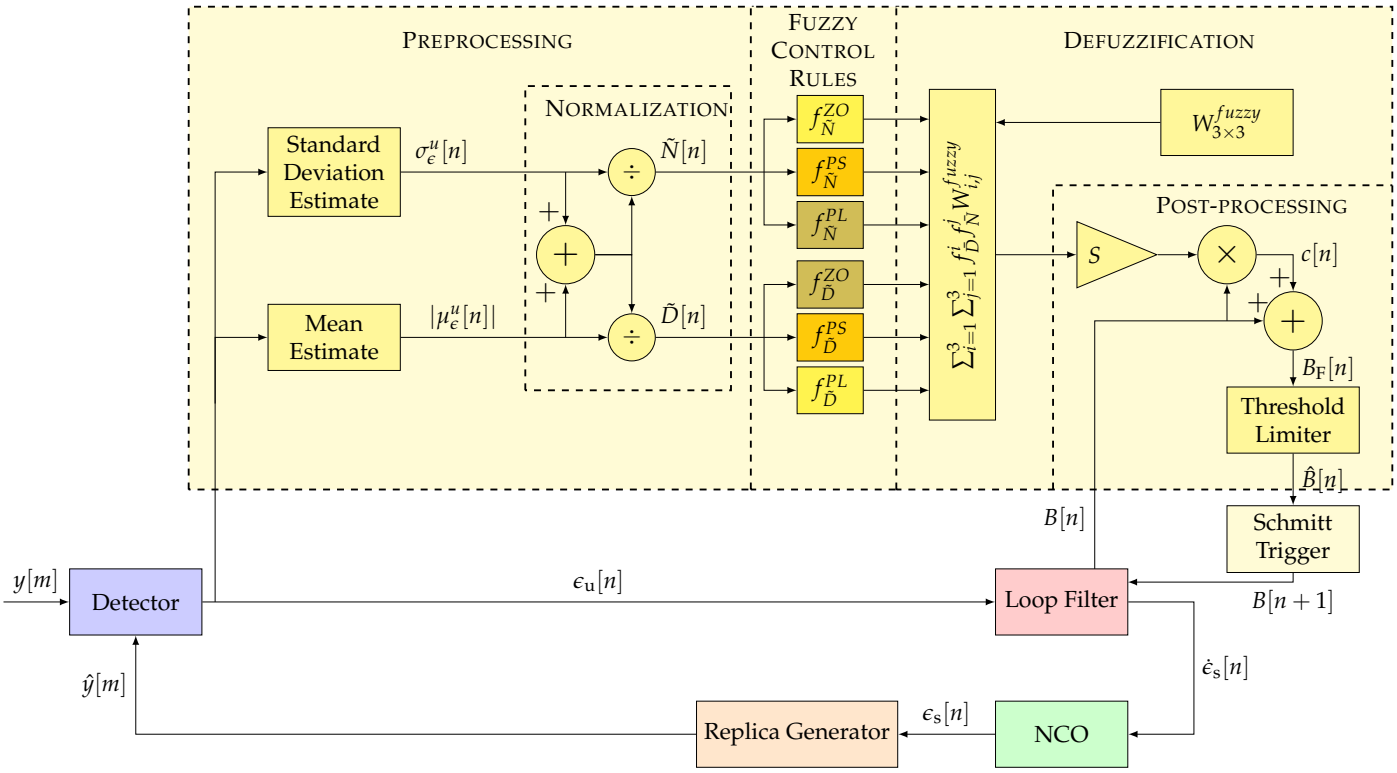


Figure 7. Architecture of implemented Fuzzy Logic (FL) technique. ©IEEE. Adapted, with permission, from [7].

Next, the fuzzy control rules are applied. The zero f^{ZO}, the positive-small f^{PS} and the positive-large f^{PL} linear membership functions weight each normalized estimation [15]. These fuzzy functions are characterized as follows:

$$f^{\mathrm{ZO}}[n,\psi[n]] = \begin{cases} \frac{T^{\psi}_{\mathrm{Fuzzy}} - \psi[n]}{T^{\psi}_{\mathrm{Fuzzy}}} & \text{if } 0 \le \psi[n] \le T^{\psi}_{\mathrm{Fuzzy}} \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

$$f^{\mathrm{PS}}[n,\psi[n]] = \begin{cases} \frac{\psi[n]}{T^{\psi}_{\mathrm{Fuzzy}}} & \text{if } 0 \le \psi[n] \le T^{\psi}_{\mathrm{Fuzzy}} \\ \frac{1-\psi[n]}{1-T^{\psi}_{\mathrm{Fuzzy}}} & \text{if } T^{\psi}_{\mathrm{Fuzzy}} \le \psi[n] \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

$$f^{\mathrm{PL}}[n,\psi[n]] = \begin{cases} \frac{\psi[n] - T^{\psi}_{\mathrm{Fuzzy}}}{1-T^{\psi}_{\mathrm{Fuzzy}}} & \text{if } T^{\psi}_{\mathrm{Fuzzy}} \le \psi[n] \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

where $\psi[n]$ is the input estimation ($\tilde{N}$ or $\tilde{D}$) and $T^{\psi}_{\mathrm{Fuzzy}}$ is the function threshold that defines the regions of these three fuzzy functions. The relation between the thresholds of $\tilde{D}$ and $\tilde{N}$ is:

$$T^{\tilde{N}}_{\mathrm{Fuzzy}} = 1 - T^{\tilde{D}}_{\mathrm{Fuzzy}} \tag{30}$$

Figure 8 shows the fuzzy functions for $\tilde{D}$ and $\tilde{N}$. These functions have a triangular shape. The base of these triangles is so wide that it allows each fuzzy input to be a member

of two sets at least. An interesting observation is the symmetry between the weighted estimates since $\tilde{D} = 1 - \tilde{N}$:

$$f^{\mathrm{ZO}}[n, \tilde{D}[n]] = f^{\mathrm{PL}}[n, \tilde{N}[n]] \tag{31}$$

$$f^{\mathrm{PS}}[n, \tilde{D}[n]] = f^{\mathrm{PS}}[n, \tilde{N}[n]] \tag{32}$$

$$f^{\mathrm{PL}}[n, \tilde{D}[n]] = f^{\mathrm{ZO}}[n, \tilde{N}[n]] \tag{33}$$

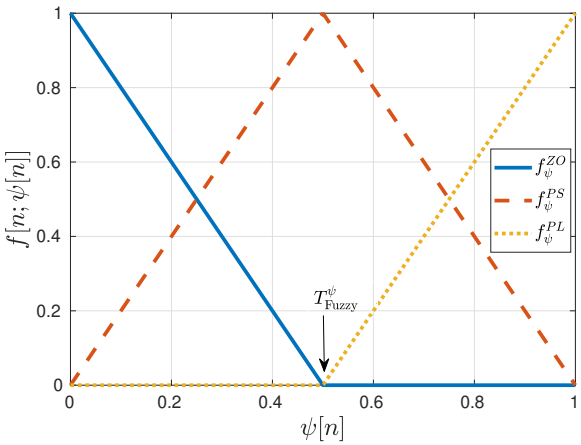


Figure 8. Fuzzy functions for normalized estimates $\tilde{D}$ and $\tilde{N}$. ©IEEE. Reprinted, with permission, from [7].

The deffuzication stage combines the degrees of membership of each fuzzy rule. The best-known deffuzification method of the FL technique is the center of gravity (COG) [14]. The method used in this paper is similar to the fuzzy mean method (FMM) [34]. In contrast to the previous method, integration is done instead of an average. The fuzzy-weighted estimates are combined with a fuzzy weighting matrix $W^{\mathrm{fuzzy}}_{3\times3}$:

$$P[n] = \sum_{i=1}^{3}\sum_{j=1}^{3} f^{i}[n, \tilde{N}[n]] f^{j}[n, \tilde{D}[n]] W^{\mathrm{fuzzy}}_{i,j} \tag{34}$$

where $\{f^1, f^2, f^3\}$ are $\{f^{Z0}, f^{\mathrm{PS}}, f^{\mathrm{PL}}\}$ respectively.

The fuzzy weighting matrix $W^{\mathrm{fuzzy}}_{3\times3}$ is a hollow matrix in which the upper triangle matrix has positive elements, and the lower triangle matrix negative values. Table 1 shows the structure of $W^{\mathrm{fuzzy}}_{3\times3}$ [7].

Table 1. Structure of the fuzzy Matrix $W^{\mathrm{fuzzy}}_{3\times3}$.

		$\tilde{D}$		
		ZO	**PS**	**PL**
$\tilde{N}$	ZO	0	>0	>0
	PS	<0	0	>0
	PL	<0	<0	0

The postprocessing unit scales the resultant value $P[n]$ by S and multiplies it by the current loop bandwidth $B[n]$, achieving the final control signal $c[n]$:

$$c[n, P[n], B[n]] = P[n] \cdot S \cdot B[n] \tag{35}$$

Next, the loop bandwidth is updated:

$$B_{\mathrm{F}}[n] = B[n] + c[n, P[n], B[n]] \tag{36}$$

Finally, as the FAB technique, a threshold limiter constraints the loop bandwidth (see Equation (24)).

Two examples are addressed. If the normalized dynamics $\tilde{D}$ tends to zero, the normalized noise $\tilde{N}$ goes to one. Therefore, $f^{\mathrm{ZO}}[n, \tilde{D}[n]]$ and $f^{\mathrm{PL}}[n, \tilde{N}[n]]$ tend to one and the other fuzzy functions to zero. Consequently, $P[n]$ is:

$$P[n] = W_{3,1}^{\mathrm{fuzzy}} < 0 \tag{37}$$

The updated loop bandwidth $B[n+1]$ decreases each iteration since the control signal is always negative. Due to the loop bandwidth decrease, the control signal $c[n]$ increases. After some iterations, $B[n]$ and $c[n]$ tends to zero. This proves the lower bound stability of the algorithm.

In the opposite case, if $\tilde{D}$ tends to one, the normalized noise $\tilde{N}$ goes to zero. In such a case, $P[n] = W_{1,3}^{\mathrm{fuzzy}} > 0$ and the loop bandwidth increases together with the control signal. The bigger the loop bandwidth, the bigger the update (see Equation (35)). This can lead to instabilities due to abrupt changes of the loop bandwidth. Since there is no upper bound, the same threshold limiter as in the FAB technique is implemented.

3.3. Loop-Bandwidth Control Algorithm

Similar to the FL method, the LBCA [16] uses the discriminator's statistics to adapt the loop bandwidth. However, it uses a normalized bandwidth B_N dependent sigmoid-based weighting function to combine these values. Figure 9 shows the structure of the LBCA.

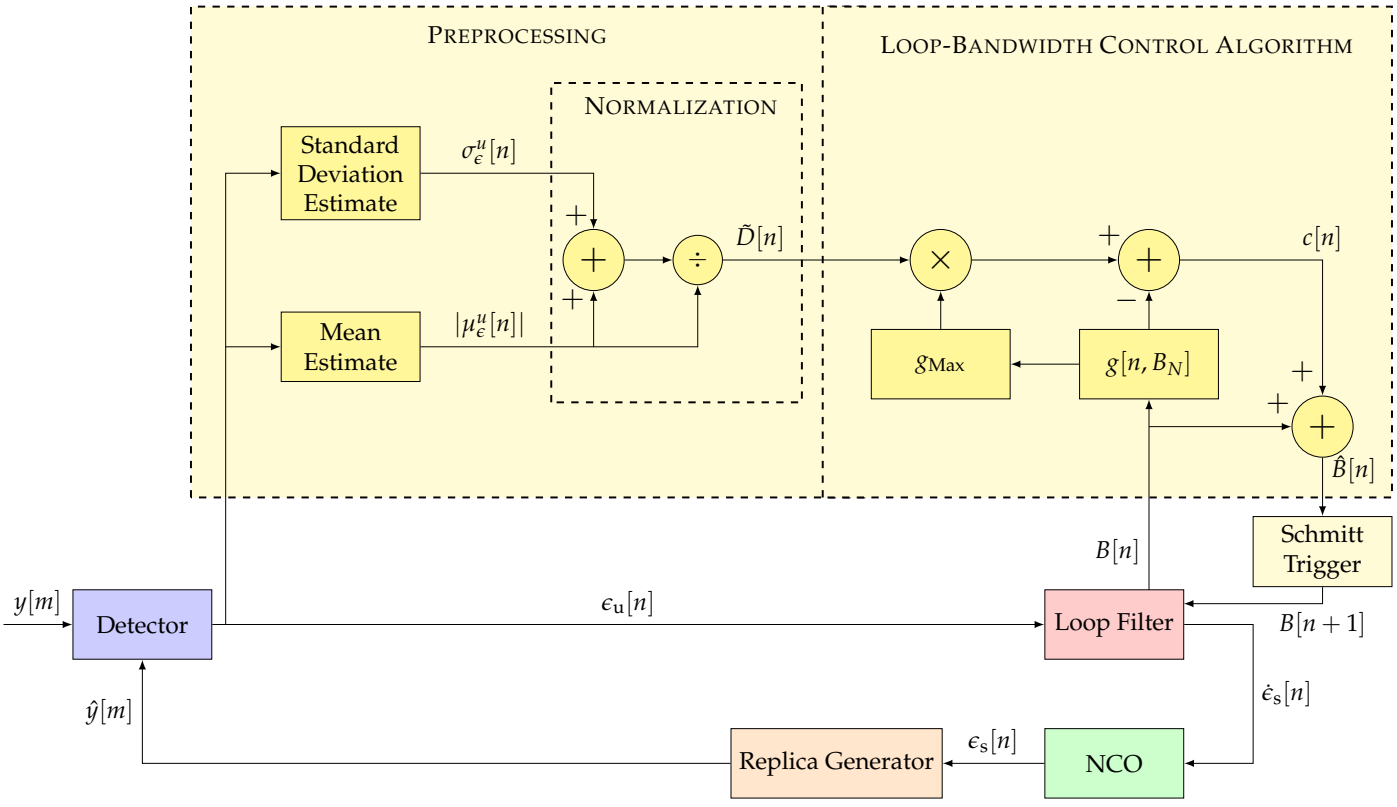


Figure 9. Architecture of loop-bandwidth control algorithm (LBCA). ©IEEE. Adapted, with permission, from [7].

The algorithm's inputs are the absolute mean $|\mu_\epsilon^u|$ and the standard deviation σ_ϵ^u estimates of the discriminator's output. The absolute mean $|\mu_\epsilon^u|$ is interpreted as the dynamics and the standard deviation σ_ϵ^u as the noise of the tracking channel. The signal dynamic estimate is normalized, similar to the FL approach.

At the core, the LBCA is a weighing function $g[n, B_N]$. The weighting function $g[n, B_N]$ and the normalized dynamic estimate $\tilde{D}$, that is scaled by the maximum value of the weighting function g_{Max}, determine the control signal $c[n, B_N]$:

$$c[n, B_N] = g_{\text{Max}} \cdot \tilde{D}[n] - g[n, B_N] \tag{38}$$

The control signal $c[n, B_N]$ and the current loop bandwidth $B[n]$ determine the estimated loop bandwidth $\hat{B}[n]$:

$$\hat{B}[n] = B[n] + c[n, B_N] \tag{39}$$

The weighting function $g[n, B_N]$ directly determines the adaption performance of the LBCA. It is a linear combination of m normalized positive sigmoid functions:

$$g[n, B_N] = \sum_{k=1}^{m} w_k \text{Sig}_k(S_k(B_N[n] - P_k)) = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}^T \times \begin{bmatrix} \text{Sig}_1(S_1(B_N[n] - P_1)) \\ \vdots \\ \text{Sig}_m(S_m(B_N[n] - P_m)) \end{bmatrix} \tag{40}$$

where P_k is the shift parameter, S_k is the horizontal scaling and w_k the vertical scaling. The sigmoid function $\text{Sig}(x)$ is defined as [35]:

$$\text{Sig}(x) = \frac{1}{1 + e^{-x}} \tag{41}$$

The maximum value of the weighting function g_{Max} is the sum of the vertical scaling values:

$$g_{\text{Max}} = \sum_{k=1}^{m} w_k \tag{42}$$

g_{Max} indicates the maximum update the algorithm can perform at each iteration. It implicitly constrains the control value:

$$|c[n, B_N]| \leq g_{\text{Max}} \tag{43}$$

If the noise and signal dynamics estimates are reliable, a larger g_{Max} value is appropriate to facilitate faster reaction.

The shape of the weighting function $g[n, B_N]$ is crucial to achieving stable functionality and optimal performance. Two examples follow to illustrate the weighting function $g[n, B_N]$ in extreme cases. In both cases, the maximum of the weighting function is unitary ($g_{\text{Max}} = 1$). In the first example, a noisy static scenario causes the estimated normalized dynamics $\tilde{D}$ to tend to zero. In this case, the loop bandwidth should decrease in order to filter as much noise as possible. Therefore, the weighting function must be more significant than $\tilde{D}$ to have a negative control value. However, if the loop bandwidth approaches zero, the weighting function should also tend to zero to avoid a negative loop bandwidth and destabilize the STL. In the PLL, the loop bandwidth should be greater than a certain value to not lose lock due to clock-error frequency drifts [36]. If a temperature-compensated crystal oscillator (TCXO) is used, a good minimum normalized PLL bandwidth is around 0.06. Values less than the selected minimum normalized loop bandwidth are located in the low normalized dynamic region. In this region, the weighting function approaches zero, and the normalized bandwidth stops decreasing.

In the second example, the opposite scenario is considered. High signal dynamics at high C/N_0, in which the normalized dynamics tend to one. Since the STL must react as

fast as possible to signal dynamics, the loop bandwidth should increase. Consequently, the weighting function is lower than $\tilde{D}$ in order to have a positive control value $c[n]$. A condition is that the analog to digital mapping does not hold anymore if the normalized bandwidth is bigger than 0.4 [29–31]. Therefore, values bigger than 0.4 are located in the high normalized dynamic region.In this region the weighting function is close to one, meaning that the loop bandwidth can no longer increase.

The mentioned extreme scenarios are helpful to achieve a first glance of the weighting function's shape. However, intermediate scenarios in which the signal dynamics and noise are equivalent must also be noted. These cases occur in the transient region.The value of the weighting function in this region is the normalized dynamic threshold T_{LBCA}. T_{LBCA} determines the sensitivity to normalized dynamics. The higher the threshold, the less sensitive to signal dynamics.

Figure 10 shows the shape of the weighting function being a linear combination of two sigmoid functions.

$$g[n, B_N] = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}^T \times \begin{bmatrix} \mathrm{Sig}(S_1(B_N - P_1)) \\ \mathrm{Sig}(S_2(B_N - P_2)) \end{bmatrix} \tag{44}$$

The biases P_1, P_2 determine the borders of the regions, the horizontal scaling S_1, S_2 indicate the slope of the transition between regions, and the vertical scaling w_1, w_2 define the sensitivity to normalized dynamics.

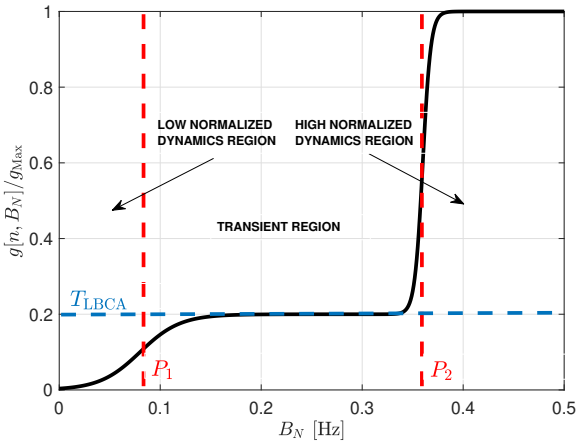
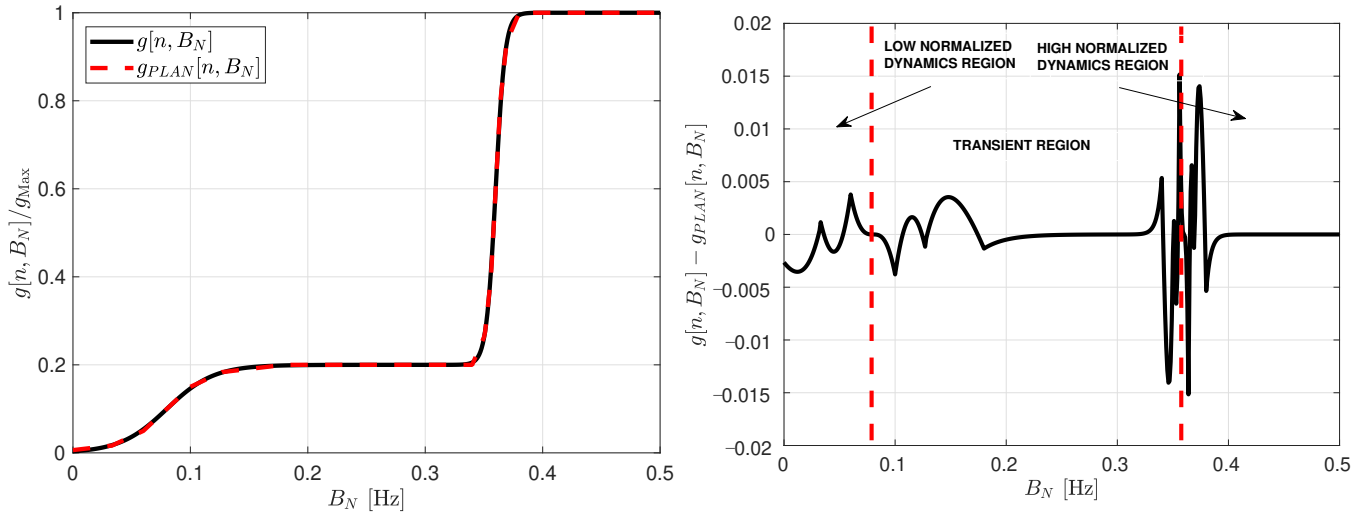


Figure 10. Weighting function of LBCA.

The weighted function's complexity is significant due to the sigmoid function in which a division of an exponential is required. An approach to reduce the complexity is to perform an approximation using the piecewise linear approximation of nonlinearities (PLAN) technique [37]:

$$\mathrm{Sig}_{PLAN}(S_k(B_N - P_k)) = \begin{cases} 1 & \text{if } S_k(B_N - P_k) \geq 5 \\ 0.03125 \cdot S_k(B_N - P_k) + 0.84375 & \text{if } 2.375 \leq S_k(B_N - P_k) < 5 \\ 0.125 \cdot S_k(B_N - P_k) + 0.625 & \text{if } 1 \leq S_k(B_N - P_k) < 2.375 \\ 0.25 \cdot S_k(B_N - P_k) + 0.5 & \text{if } 0 \leq S_k(B_N - P_k) < 1 \\ 1 - \mathrm{Sig}_{PLAN}(S_k(B_N - P_k)) & \text{if } S_k(B_N - P_k) < 0 \end{cases} \tag{45}$$

Figure 11 shows the comparison between the weighting function and its piece-wise linear approximation. The approximation errors are located in the limits between regions. However, it is a good approximation since there is maximum error of 0.15 %. In the next section, both weighted functions are evaluated.



(**a**) Weighting functions (**b**) Approximation error

Figure 11. Comparison between sigmoid weighting function and its piecewise linear approximation.

4. Evaluation Method of Adaptive Loop-Bandwidth Tracking Techniques

This section analyses the most adequate quality factors that represent the performance and the complexity of adaptive tracking techniques.

4.1. Tracking Performance

Two parameters that define an adaptive tracking technique's tracking performance are considered: the Cramér-Rao bound (CRB) and the one-sigma rule threshold of the unsmoothed error $\sigma_{\epsilon_u}^{th}$. The CRB indicates the minimum error variance of an unbiased estimator [38], whereas the one-sigma rule threshold is a conservative threshold that ensures a stable tracking lock if the tracking error is less than this threshold [1]. The standard deviation of the discriminator's error σ_ϵ^u is a good performance metric [8]. The tracking performance is defined as the difference between the average σ_ϵ^u and the lower bound standard deviation σ_{LB}^u:

$$P_{\text{Tracking}} = \overline{\sigma}_\epsilon^u - \sigma_{\text{LB}}^u \tag{46}$$

The average of σ_ϵ^u, $\overline{\sigma}_\epsilon^u$, is defined as:

$$\overline{\sigma}_\epsilon^u = \sum_{n=0}^{K_{\text{sim}}} \frac{\sigma_\epsilon^u[n]}{K_{\text{sim}}} \tag{47}$$

where K_{sim} is the discrete simulation time in samples. K_{sim} is represented as the product between the simulation time T_{sim} in seconds and the logging data rate f_s in Hz:

$$K_{\text{sim}} = T_{\text{sim}} f_s \tag{48}$$

Since the adaptive tracking techniques are implemented in a third-order Costas PLL, the PLL tracking performance is analyzed. Considering the PLL as a time of arrival (ToA) unbiased estimator [39], the resulting square-root CRB of the un-smoothed carrier phase error θ_u in meters is:

$$\sigma_{\text{LB}}^u = \left(\frac{\lambda}{2\pi}\right)\sqrt{\frac{1}{2\tau_{\text{int}} C/N_0}\left(1 + \frac{1}{2\tau_{\text{int}} C/N_0}\right)} \tag{49}$$

where λ is the wavelength of the GNSS signal and the term $1 + \frac{1}{2\tau_{\text{int}} C/N_0}$ is the squaring loss. Figure 12 shows the lower bound of the un-smoothed carrier phase error with an integration time τ_{int} of 20 ms at different C/N_0 levels.

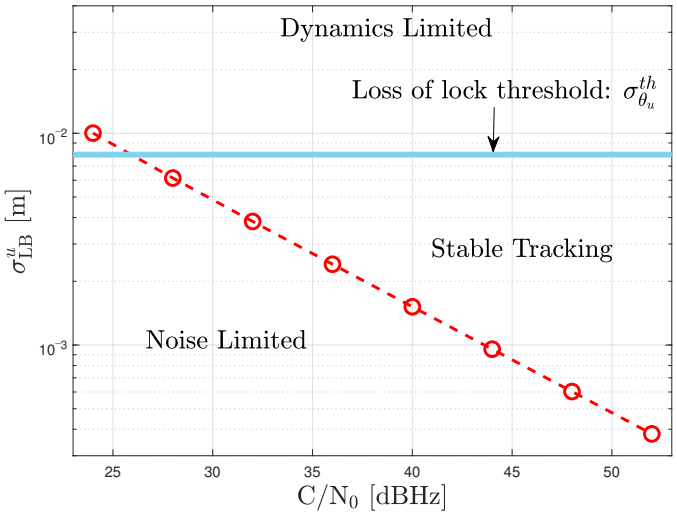


Figure 12. Square root CRB of the un-smoothed carrier phase error at different C/N_0 levels, $\tau_{\text{int}} = 20$ ms.

From the three-sigma rule-of-thumb, the carrier phase error jitter σ^s_θ must be less than a conservative threshold to ensure stable tracking and no cycle-slips [1]. This upper threshold is also applied to the un-smoothed phase error jitter σ^u_θ. Since a two-quadrant arctangent discriminator is used, the one-sigma rule threshold in meters is:

$$\sigma^{th}_{\theta^u} = \frac{1}{3} \times \frac{\Omega}{4} \times \frac{\lambda}{2\pi} = \frac{\pi}{12} \times \frac{\lambda}{2\pi} = \frac{\lambda}{24} \tag{50}$$

where Ω is the phase pull-in range in radians. The 1/12 factor in Eqution (50) is included because the one-sigma rule threshold is one-third of the three-sigma rule threshold and one-fourth of Ω is selected to have a conservative threshold.

4.2. System Performance

The system performance considers the performance of all the tracking channels together. The average lock indicator for each tracking channel is an appropriate metric to measure the system performance in terms of precision [40,41].

Since this study evaluates the performance of the PLL, the phase-lock indicator (PLI) is considered:

$$\text{PLI} = \frac{I_p^2 - Q_p^2}{I_p^2 + Q_p^2} \tag{51}$$

where I_p and Q_p are the in-phase and quadrature prompt correlation values.

The selected quality factor is the average of the PLI with respect the discrete simulation time K_{sim} and the accumulated number of tracked satellite vehicles (SVs) $N^{\text{Acc}}_{\text{sat}}$:

$$\overline{\text{PLI}} = \sum_{l=0}^{N^{\text{Acc}}_{\text{sat}}} \sum_{n=0}^{K_{\text{sim}}} \frac{\text{PLI}^l[n]}{K_{\text{sim}} N^{\text{Acc}}_{\text{sat}}} \tag{52}$$

If a SV is tracked at least once during the simulation, the PLI of this SV is considered to calculate $\overline{\text{PLI}}$.

In order to measure the system performance in terms of robustness against dynamics, a good quality factor is the average number of visible satellites being tracked $\overline{\mathrm{N}}_{\mathrm{sat}}$:

$$\overline{\mathrm{N}}_{\mathrm{sat}} = \sum_{n=0}^{K_{\mathrm{sim}}} \frac{N_{\mathrm{sat}}[n]}{K_{\mathrm{sim}} N_{\mathrm{sat}}^{\mathrm{Total}}} \tag{53}$$

where $N_{\mathrm{sat}}^{\mathrm{Total}}$ is the total number of visible satellites in the scenario.

A final metric is achieved combining (52) and (53):

$$\mathrm{P}_{\mathrm{System}} = \overline{\mathrm{PLI}} \times \overline{\mathrm{N}}_{\mathrm{sat}} \tag{54}$$

The average system performance $\overline{\mathrm{P}}_{\mathrm{System}}$ with respect the C/N_0 levels determines the overall performance of the adaptive tracking technique.

$$\overline{\mathrm{P}}_{\mathrm{System}} = \sum_{k=1}^{N_{\mathrm{CN0}}} \frac{\mathrm{P}_{\mathrm{System}}^{k}}{N_{\mathrm{CN0}}} \tag{55}$$

where N_{CN0} is the number of C/N_0 levels. The system performance metric $\overline{\mathrm{P}}_{\mathrm{System}}$ which accounts for both noise and dynamics for tracking, is a novel contribution of this paper.

4.3. Complexity

A theoretical method to quantify an adaptive tracking technique's complexity is to measure the number of required mathematical operations. This method provides a "best-case" comparison, and neglects any implementation limitations. Table 2 shows the theoretical number of additions, multiplications, divisions and different operations required for each case. The total number of operations for each adaptive tracking technique is labeled in colors from the highest, in red, to the lowest number of operations, in green.

Table 2. Complexity of the adaptive loop-bandwidth tracking techniques based on the number of operations.

Tracking Technique	Sub-Module	Number of Operations:			
		Additions	**Multiplications**	**Divisions**	**Other**
FAB	Dynamic Stress Estimator	4	6	1	-
	Three-Sigma FAB	1	11	2	$\sqrt[7]{\cdot}$
	Gradient Descent	3	1	1	-
	Total	8	18	4	1
FL	Dynamic/Noise Estimator	4	4	2	-
	FL Algorithm	13	25	0	-
	Total	17	29	2	0
LBCA	Dynamic/Noise Estimator	2	1	1	-
	LBCA algorithm	4	5	1	exp
	Total	6	6	2	1
LBCA + PLAN	Dynamic/Noise Estimator	2	1	1	-
	LBCA algorithm	4	6	0	-
	Total	6	7	1	0

Since the algorithms are implemented in software, the time complexity is a good practical complexity indicator. The time complexity measures the processing time the algorithm takes in software on the processing platform. Although this approach depends on how exemplary the algorithm's software implementation is, the results show a correlation between the number of operations and the time complexity. The time complexity is

measured in an Intel Skylake micro-architecture with a clock speed of 3700 MHz. The code is implemented in C++, and a for-loop is used to iterate the operation of a loop filter 3×10^8 times. The *chrono* library is used to measure the processing time of the algorithm [42]. Also, the performance profiler tool for Linux *operf* serves to have a statistical report of the libraries used during the algorithm's execution [43]. In order to avoid multi-threading, the *taskset* command is used to bind the application process to one single core [44]. Table 3 shows the total time complexity T_C in seconds, the average time complexity at each iteration T_{Iter} in nanoseconds, and the added time complexity T_{Added} compared to the standard loop filter T_C^{Standard} (i.e., with no adaption algorithms). The T_{Added} values are labeled in colors from the most complex algorithm, in red, to the less complex one, in green. The iteration time complexity is defined as:

$$T_{\text{Iter}} = \frac{T_C}{3 \times 10^8} \times 10^9 \quad \text{(nanoseconds)} \tag{56}$$

and the added time complexity is expressed as:

$$T_{\text{Added}} = \frac{T_C}{T_C^{\text{Standard}}} \text{(times)} \tag{57}$$

Table 3. Time complexity of the adaptive loop-bandwidth tracking techniques, 3×10^8 iterations.

Tracking Technique	Total Time Complexity T_C (s)	Iteration Time Complexity T_{Iter} (ns)	Added Time Complexity T_{Added} (Times)
Standard	18.1	60.3	1
FAB	133.7	445.8	7.4×
FL	55.0	183.5	3.0×
LBCA	31.9	106.4	1.8×
LBCA + PLAN	27.8	92.8	1.5×

The FAB technique presents high complexity due to the seventh root (see Equation (18)). This is verified from the profiler report, which shows high use of the *pow* function from the *libm* library [45]. The derivation of the Newton-Raphson method [46] can approximate this operation in order to achieve a lower complexity. However, this approximation is not performed in the actual implementation. The FL technique presents a lower time complexity than the FAB technique but higher than the LBCA. The profiler report of the LBCA shows the use of the *exp* function from the *libm* library [45]. The piece-wise linear approximation of the weighting function in the LBCA technique removes that function's utilization, reducing the algorithm's complexity.

5. Experimental Setup

This section describes the experimental setup and the expected results.

5.1. Receiver and Algorithm Implementation

The GOOSE© platform, developed by Fraunhofer IIS and marketed through TeleOrbit GmbH, is a GNSS receiver with an open software interface [24,25]. A picture of the receiver is shown in Figure 13. The receiver is based on a Xilinx Kintex7 field-programmable gate array (FPGA), depicted in green, connected to an external processor, in purple, using a peripheral component interconnect express (PCIe) interface. The FPGA receives digital samples from a customized tri-band radio-frequency front-end (RFFE), in red. One acquisition module and sixty tracking channels are implemented, which can be controlled by an external processor.

Processing intense portions of the tracking stage are implemented in hardware ("Tracking Correlators") and the remainder in software ("Tracking Loops"). The correlators and the NCOs are implemented on an FPGA, whereas the software part includes the discriminators, the loop filters, and the adaptive tracking algorithms. Figure 14 shows a block diagram of the entire receiver. The analog hardware is shown in red, the digital firmware in green, and the digital software in blue. It also contains the RFFE, an analog-to-digital converter (ADC), signal conditioning, acquisition, the receiver manager, symbol decoding, and the PVT engine.

Figure 13. Photo of the GOOSE receiver @Fraunhofer IIS/Paul Pulkert.

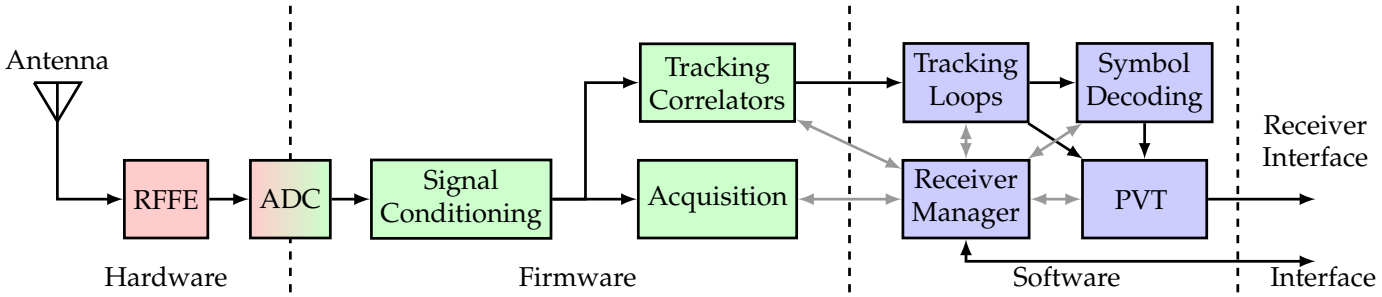


Figure 14. GOOSE architecture diagram.

Each tracking channel contains a second-order FLL, a third-order Costas PLL and a second-order DLL with PLL-assisted-DLL (PAD) enabled. The FLL is first enabled in order to track and refine the acquired Doppler frequency. Once a stable frequency lock is achieved, the transition to the FLL-assisted-PLL (FAP) is done. In this stage the PLL lock indicator is stabilizing with the assistance of the FLL. After obtaining a good and stable PLL lock, the FLL gets disabled in order to improve the tracking accuracy. In this final stage, the bit synchronization is performed to decode the navigation message and get a PVT solution. If the PLL lock deteriorates, the receiver goes back to the previous stage FAP by default in order to recover the carrier lock. However, for these tests this condition is disabled in order to evaluate correctly the adaptive carrier phase tracking capabilities.

The purpose is to evaluate the performance and the complexity of each variable loop-bandwidth tracking technique implemented in this receiver's PLL against simulated scenarios with different dynamics and noise levels. In this study, only Global Positioning System (GPS) L1 C/A signals are considered. Moreover, the algorithms are evaluated at an integration time τ_{int} of 20 ms. Table 4 shows the initial configuration of the DLL, FLL, and PLL that is used for all the tests.

Table 4. Initial configuration of the DLL, FLL and PLL for each tracking channel.

	DLL	FLL	PLL
Order	2	2	3
B [Hz]	0.1	15	8
Discriminator	CELP	Diff Atan	Atan

The FAB, the FL, the LBCA, and the LBCA with PLAN algorithms are implemented on the GOOSE receiver in software. These adaptive tracking techniques are configured to adapt the loop bandwidth of a third-order Costas PLL.

5.1.1. FAB Configuration

Since the integration time τ_{int} is 20 ms, a loop bandwidth bigger than 18 Hz could destabilize the tracking because of the wrong mapping from analog to digital domain [29–31]. Hence, the selected upper bound B_{Max} of the threshold limiter (see Equation (24)) is set to 18 Hz. It has been observed that a PLL loop bandwidth lower than 4 Hz cannot remain in lock, probably because of permanent signal dynamics due to nonlinearities. Consequently, in case the estimation tends erroneously to zero, a lower bound 4 Hz is set in the threshold limiter.

$$\hat{B}[n] = \begin{cases} 18 & \text{if } B^s_{\text{GD}}[n] > 18 \\ B^s_{\text{GD}}[n] & \text{if } 4 \leq B^s_{\text{GD}}[n] \leq 18 \\ 4 & \text{if } B^s_{\text{GD}}[n] < 4 \end{cases} \tag{58}$$

In the following section, the FAB algorithm's performance is evaluated in terms of the filtered time Δt of the steady state error dynamics estimator (see Equation (20)). This parameter is selected since it directly affects the algorithm's sensitivity to dynamics.

5.1.2. FL Configuration

Table 5 shows the empirically selected values of the fuzzy matrix $W^{\text{fuzzy}}_{3\times3}$ [7].

Table 5. Selected values of the fuzzy Matrix $W^{\text{fuzzy}}_{3\times3}$.

		$\tilde{D}$		
		ZO	**PS**	**PL**
$\tilde{N}$	ZO	0	0.5	0.75
	PS	−0.25	0	0.5
	PL	−0.5	−0.25	0

The same threshold limiter as in the FAB technique is used (see Equation (58)). Another parameter to consider is the fuzzy threshold T^{ψ}_{Fuzzy}. Based on the three-sigma rule-of-thumb, the optimal normalized dynamics can be calculated [16]. D^{opt} is the optimal normalized dynamics in order to achieve best tracking performance. For a third-order PLL the optimal normalized dynamics D^{opt} is 1/7. In the following section, the FL technique's performance is evaluated, varying the normalized dynamics fuzzy threshold $T^{\tilde{D}}_{\text{fuzzy}}$ around D^{opt}. Moreover, the effects of the scale factor S is shown.

5.1.3. LBCA Configuration

The selected weighting function is a linear combination of two sigmoid functions and has the following expression:

$$g[n, B_N] = S \begin{bmatrix} T_{\text{LBCA}} \\ 1 - T_{\text{LBCA}} \end{bmatrix}^T \times \begin{bmatrix} \text{Sig}(50(B_N - 0.06)) \\ \text{Sig}(250(B_N - 0.36)) \end{bmatrix} \tag{59}$$

The biases constraints the normalized bandwidth B_N. The algorithm cannot go below a B_N lower than 0.06. In theory, the lower limit should be set at zero if there is no signal dynamics. However, it is considered that there is always some dynamics due to the clock drift or other non-linearities. Therefore, 0.06 is chosen as a lower limit. The upper limit is chosen due to the mapping limitation from the analog domain to the digital domain. The empirically selected horizontal scaling values have an adequate slope in the borders between regions. The vertical scaling depends on the normalized dynamics threshold T_{LBCA}, and a scale factor S. The former parameter determines the sensitivity to dynamics, whereas the latter indicates the maximum loop bandwidth update of the algorithm. S is equivalent to g_{Max}:

$$g_{\mathrm{Max}} = S \cdot (T_{\mathrm{LBCA}} + 1 - T_{\mathrm{LBCA}}) = S \tag{60}$$

As in the FL configuration, T_{LBCA} must be around the optimal normalized dynamics D^{opt}. Different normalized dynamics threshold T_{LBCA} values and scale factors g_{Max} are presented in the following section.

5.2. Evaluation Setup

Figure 15 presents the test set-up. It is the same as with previous studies [16], where a Spirent GSS9000 radio-frequency constellation simulator (RFCS) generates controlled scenarios. The simulation duration is 20 min and the simulation repeats for different C/N_0 levels. In this case, the simulation repeats 8 times, from 24 dBHz to 52 dBHz in 4 dB steps. Since the sensitivity of the acquisition is lower than the sensitivity of the tracking, the simulation always starts at the highest C/N_0 level, 52 dBHz. The C/N_0 level is reduced by 4 dB in 30 s intervals, until reaching the desired level. For example, 3.5 min are necessary to reach a C/N_0 level of 24 dBHz. Therefore, in order to assure that the measured tracking and system performance is reliable, the last 10 min of the simulation are considered, $T_{\mathrm{sim}} = 600$ s. Only GPS L1 C/A signals are generated by the RFCS, as only these signals are evaluated in this study. However, for future evaluations, other GNSS signals will also be considered.

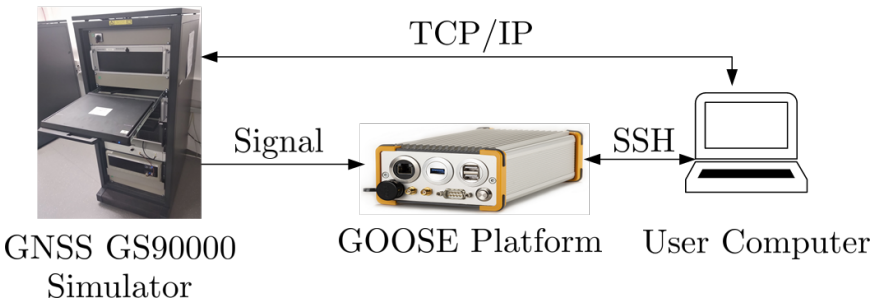


Figure 15. Setup for the simulation. ©IEEE. Adapted, with permission, from [7].

The generated scenarios are either static or have receiver dynamics. The static scenarios represent stationary use-cases such as GNSS reference stations, whereas the dynamic scenarios represent harsh vehicular conditions. Figure 16 shows the sky-plot of both scenarios. There are 10 visible satellites during the simulation. However, SV G1 disappears behind the horizon after two minutes of simulation and SV G30 rises above the horizon near the end of the simulation. Therefore, the maximum number of visible satellites $N_{\mathrm{sat}}^{\mathrm{Total}}$ is limited to eight.

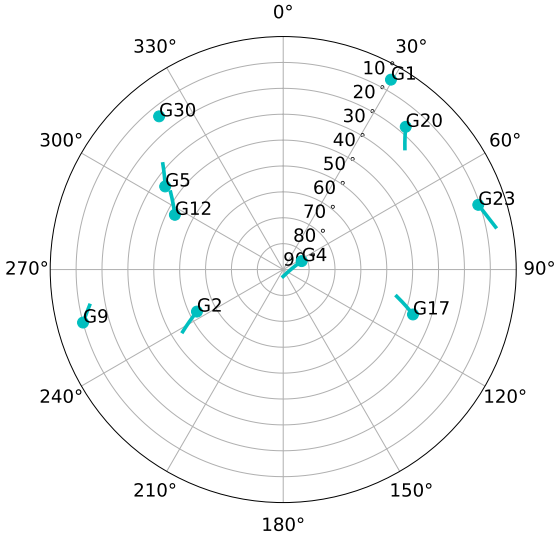
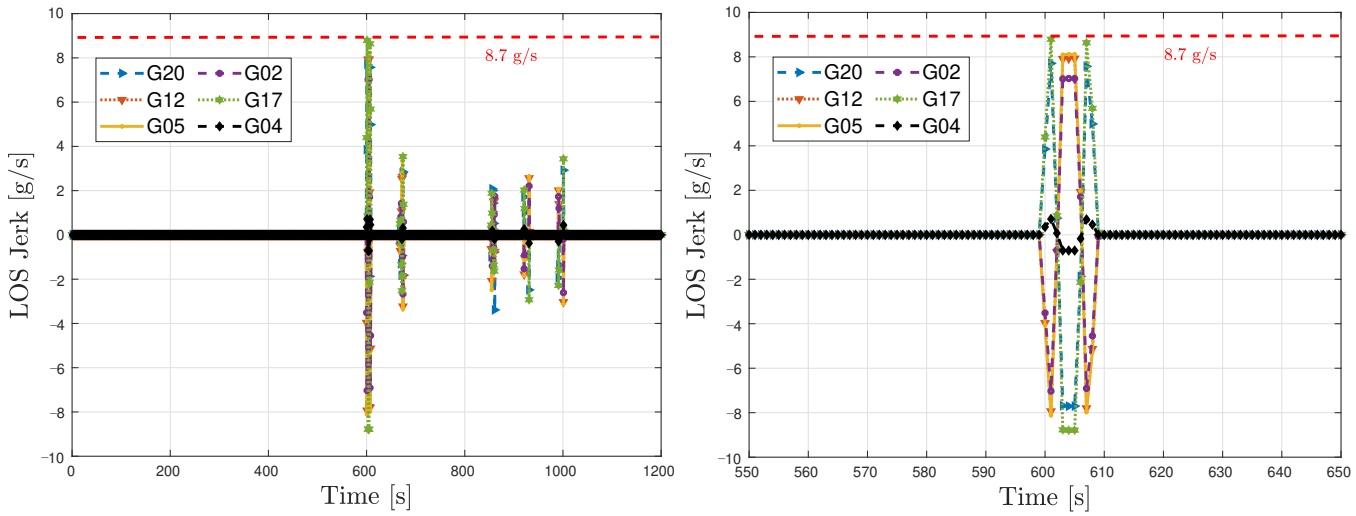


Figure 16. Sky-plot of simulated scenario.

Figure 17 shows the LOS jerk dynamics of six out of the eight GPS SVs in the simulated dynamic scenario. The first 10 min of the simulated dynamic scenario is static. In this way, the adaptive tracking converges to a steady estimation of the loop bandwidth equal to the one in the simulated static scenario. After 10 min of simulation, the high dynamic vehicular scenario begins. The simulated trajectory presents the highest LOS dynamics at low elevation satellites, whereas high elevation ones have reduced dynamics. For example, SV G4 is located in the zenith, and a maximum LOS jerk of 1 g/s is observed, whereas SV G17 presents the highest LOS jerks dynamics with 8.7 g/s. Figure 17b focuses on the time with the highest dynamics. In this period, the tracking of most of the satellites will be lost.



(**a**) Entire simulation (**b**) Zoom to highest dynamics time period

Figure 17. Jerk line-of-sight dynamics of simulated dynamic scenario.

Before evaluating the adaptive tracking techniques, the tracking and system performance of the traditional PLL are measured. This helps to observe the performance improvement of adaptive loop-bandwidth tracking techniques compared to standard tracking. A single tracking channel is tested to evaluate the tracking performance, and the tracking outputs, including the correlator IQ values, discriminator outputs, current loop bandwidth, and the estimated C/N_0, are logged. For the system performance, all the eight visible GPS SVs are considered and the open GNSS receiver protocol (OGRP) is used for logging the receiver measurements [47,48]. The average system performance $\overline{P}_{System}$ metric

for the standard PLL and the adaptive algorithms are calculated (see Equation (55)). Lastly, basic Pareto optimization [49–51] is applied to identify the best techniques based on the static $\overline{\mathrm{P}}_{\mathrm{System}}$ and dynamic $\overline{\mathrm{P}}_{\mathrm{System}}$ trade-off.

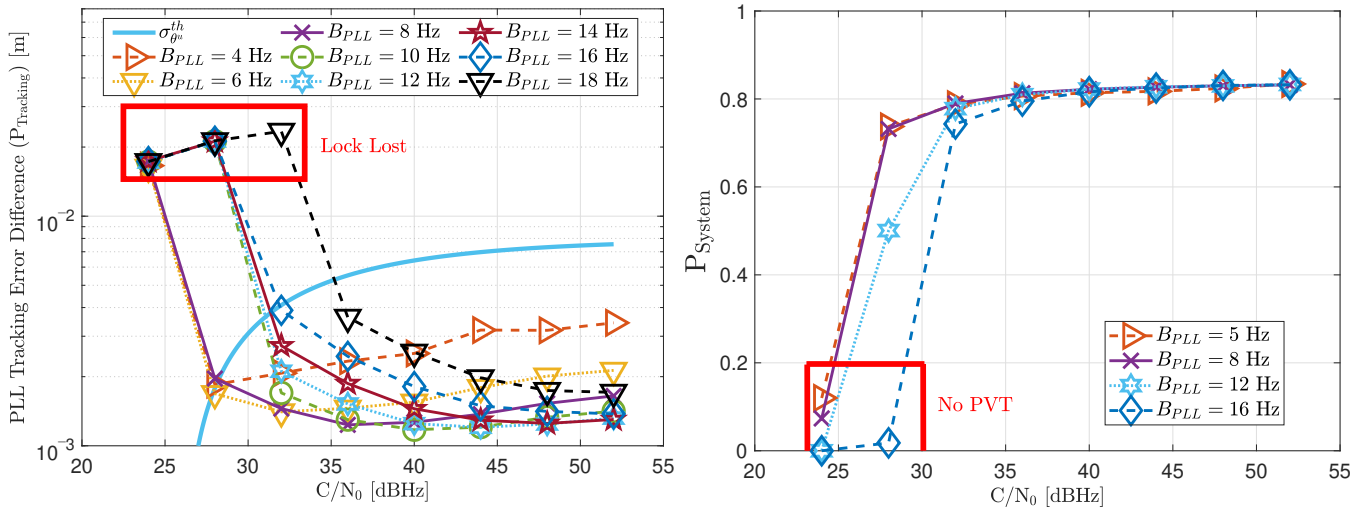
The data rate f_s of the logged measurements equals $1/\tau_{\mathrm{int}}$. Since the integration time τ_{int} is 20 ms, the data rate is 50 Hz. Therefore, considering Equation (48) and $T_{\mathrm{sim}} = 600$ s, the discrete simulation time K_{sim} equals 30,000 samples.

6. Results

The results are separated into four sections. First, the static performance is evaluated to show tracking capabilities in noise only environments. Second, dynamic performance is measured to show the capabilities for vehicular applications. Third, the synthesis of all the gathered results is performed. The last section presents a final comparison between system performance and time complexity. The dataset used to plot the presented results are available on the cloud [52].

6.1. Static Scenario

Figure 18 shows the GNSS-Receiver with Open Software Interface (GOOSE) receiver's tracking and system performance in a static scenario at different PLL loop bandwidths and C/N_0 levels. Figure Figure 18a presents the tracking performance of SV G4. The one-sigma rule threshold $\sigma_{\theta^u}^{th}$ in meters is also included. Values greater than $\sigma_{\theta^u}^{th}$ indicate that the lock of the tracking is likely lost and experiencing severe cycle slips. Therefore, only values below this threshold can be regarded as being on a stable track. The tracking performance at different loop bandwidths differs in order of millimeters. The results have the same tendency as previous results obtained by another GOOSE receiver [16]. At a higher C/N_0 level, a bigger loop bandwidth reduces the tracking error difference. This is due to the fast response of the filter to correct the carrier phase error. On the contrary, if the C/N_0 decreases, a lower loop bandwidth is adequate to reduce the error's noise.



(**a**) Tracking performance of SV G4

(**b**) System performance

Figure 18. Static performance of standard PLL at different C/N_0 levels and loop bandwidths.

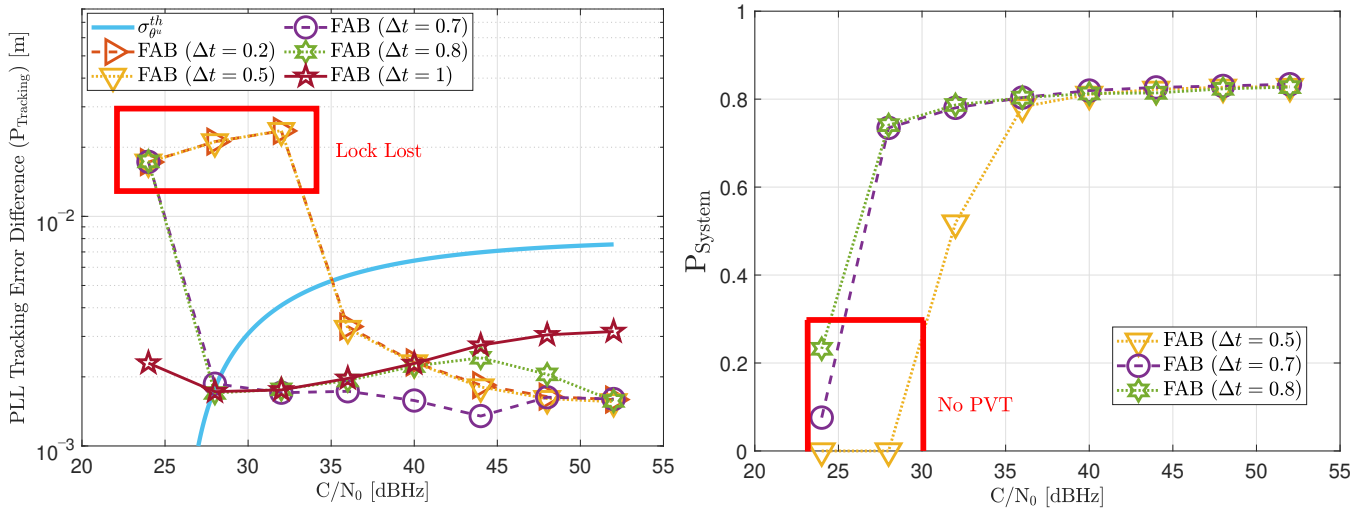
The system performance $\mathrm{P}_{\mathrm{System}}$ is the product between the average PLI, $\overline{\mathrm{PLI}}$, and the average number of tracked visible satellites $\overline{\mathrm{N}}_{\mathrm{sat}}$ (see Equation (54)). Figure 18b shows the system performance of the GOOSE receiver at different C/N_0 levels. At high C/N_0 values, all bandwidths achieve good performance. However, at low C/N_0, broader loop bandwidths result in poor performance. This shows how effective a narrow loop bandwidth can suppress noise effects.

Figure 19 includes the tracking performance of SV G4 using the FAB PLL (see Figure 19a) and the system performance considering all the visible SVs (see Figure 19b).

The FAB technique changes its sensitivity to dynamics depending on the IIR filter decay time Δt to estimate the SSE (see Equation (20)). Hence, different decay times are evaluated to see how this parameter affects the static performance. If the time of filtered data is less than 0.5 s, the algorithm is too sensitive to dynamics, and the tracking performance deteriorates at low C/N_0. From a system perspective, this configuration is the only one that does not achieve PVT at 28 dBHz. The system performance is equivalent to the other configurations at higher C/N_0 levels. If the decay time Δt is above 0.8 s, the tracking performance deteriorates at higher C/N_0 level. An intermediate Δt such as 0.7 s seems to have the best tracking performance. Therefore, an adequate IIR filter with a decay time Δt is determined to be between 0.6 s and 0.8 s.

Figure 20 shows the average loop bandwidth of the adaptive FAB PLL $\overline{B}_{\text{APLL}}$. The average loop bandwidth is calculated as:

$$\overline{B}_{\text{APLL}} = \sum_{k=0}^{K_{\text{sim}}} \frac{B_{\text{APLL}_k}}{K_{\text{sim}}} \tag{61}$$



(**a**) Tracking performance of SV G4

(**b**) System performance

Figure 19. Static performance of FAB PLL at different C/N_0 levels.

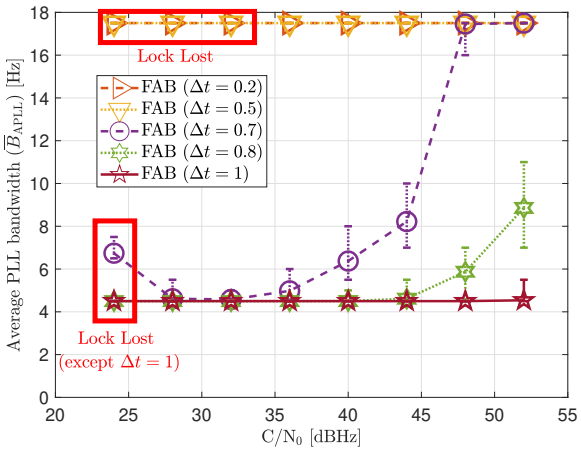
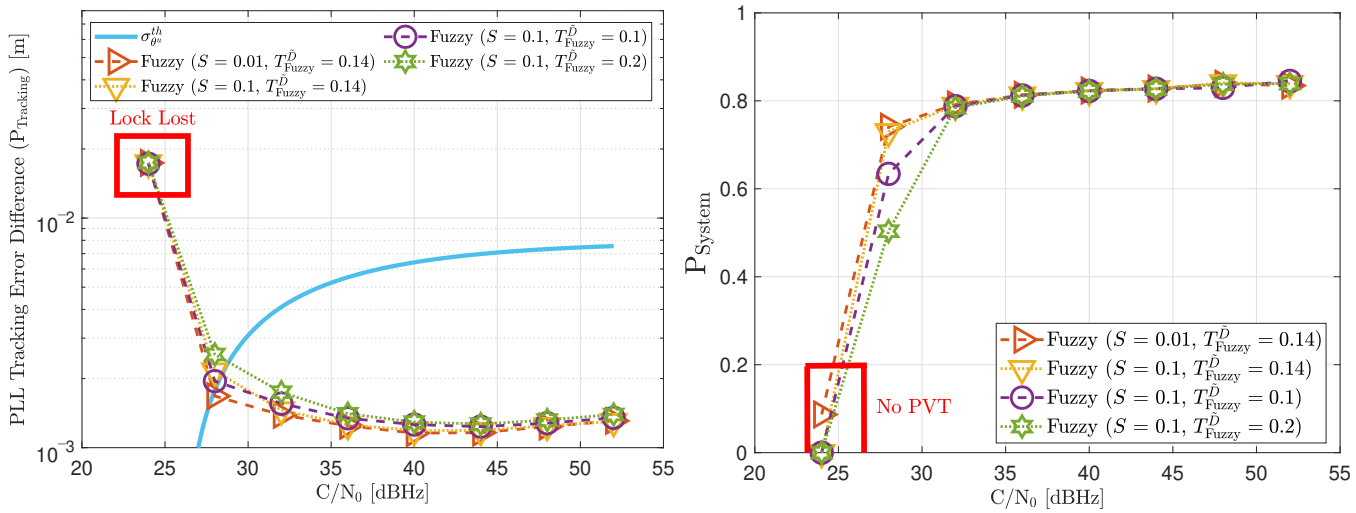


Figure 20. Average FAB PLL loop bandwidth of SV G4 trackingin static scenario.

The added error bars indicate the maximum and minimum estimated loop bandwidth during the entire simulation. If Δt is less than 0.5 s, the average loop bandwidth is around 17.5 Hz for all the C/N_0 levels. The estimated loop bandwidth goes beyond 18 Hz, but

the threshold limiter constraints it to 17.5 Hz in order to avoid destabilizing the tracking. On the contrary, if the decay time Δt is above 0.8 s, the algorithm does not detect any steady-state dynamics. Therefore, the estimated loop bandwidth tends to zero. However, the threshold limiter constraints the loop bandwidth to 4.5 Hz.

Figure 21 shows the static performance of the FL PLL. The normalized dynamic function threshold $T^{\tilde{D}}_{Fuzzy}$ defines the sensitivity to dynamics, and the scale factor S indicates the speed of the update rate. Different function thresholds and scale factors are selected to evaluate the FL. At high C/N_0 levels, each configuration's tracking error performance is similar (see Figure 21a). Conversely, at lower C/N_0, the tracking performance degrades if the FL is too sensitive ($T^{\tilde{D}}_{Fuzzy} = 0.1$) or too insensitive ($T^{\tilde{D}}_{Fuzzy} = 0.2$) to signal dynamics. An intermediate threshold such as $T^{\tilde{D}}_{Fuzzy} = 0.14$ results in better tracking performance. The same behaviour is observed in the system performance (see Figure 21b). At 28 dBHz, $T^{\tilde{D}}_{Fuzzy} = 0.1$ and $T^{\tilde{D}}_{Fuzzy} = 0.2$ presents worse performance than $T^{\tilde{D}}_{Fuzzy} = 0.14$.



(**a**) Tracking performance of SV G4

(**b**) System performance

Figure 21. Static performance of FL PLL at different C/N_0 levels.

Figure 22 reveals that a decrease in $T^{\tilde{D}}_{Fuzzy}$ leads to a bigger average loop bandwidth $\overline{B}_{\text{APLL}}$. If the scale factor S increases, the maximum and minimum value of the estimated loop bandwidth also rise. This indicates that the loop bandwidth oscillates over a broader range, and it can be a problem at low C/N_0 levels. Another interesting observation is the algorithm's behavior once the lock is lost. For instance, at 24 dBHz, all the presented configurations lose the tracking lock. On the one hand, if the configuration is too sensitive to signal dynamics (e.g., $T^{\tilde{D}}_{Fuzzy} = 0.1$), the loss of lock is interpreted as signal dynamics, and the loop bandwidth increases at each iteration. Since there is a threshold limiter, the loop bandwidth is constrained to the upper limit, 17.5 Hz. On the other hand, if the configuration is less sensitive to dynamics, the loop bandwidth correctly decreases to zero when there is a lock loss. However, the loop bandwidth is limited to the lower bound, 4.5 Hz.

Figure 23 shows the static performance of the LBCA STL. The LBCA technique depends on the maximum value of the weighting function g_{Max} and the normalized dynamic threshold T_{LBCA} (see Figure 23a). As the FL technique, g_{Max} indicates the amount of oscillation of the loop bandwidth, and T_{LBCA} sets the sensitivity to signal dynamics. Figure 23a shows that a low g_{Max} (e.g., 0.01) can still track the signal at 24 dBHz. However, at a system level, the lock of at least 4 SV cannot be achieved and there is no PVT solution (see Figure 23b). An LBCA configuration with a higher g_{Max} makes the loop bandwidth update unstable at that C/N_0 level, and even the lock of SV G4 is lost. It is also observed that a higher normalized dynamic threshold T_{LBCA} results in a worse tracking error difference.

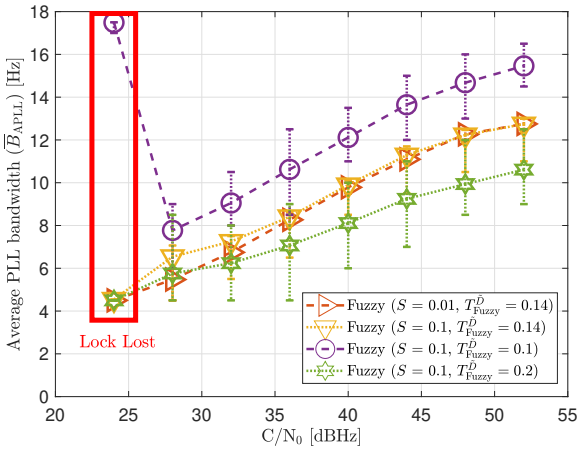
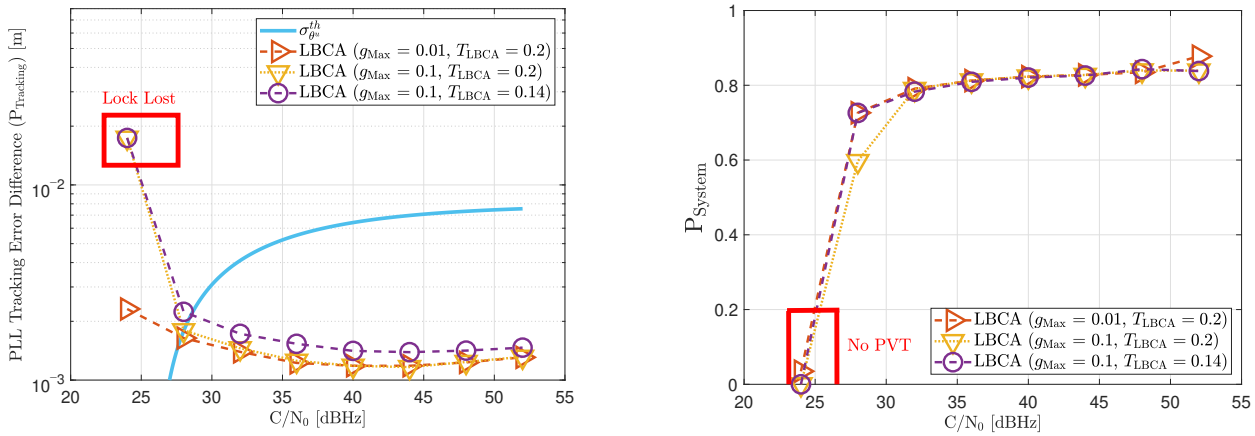


Figure 22. Average FL PLL loop bandwidth of SV G4 tracking in static scenario.



(**a**) Tracking performance of SV G4

(**b**) System performance

Figure 23. Static performance of loop-bandwidth control algorithm (LBCA) PLL at different C/N_0 levels.

As expected, a lower g_{Max} leads to a smaller oscillation range of the estimated loop bandwidth (see Figure 24). Also, a decrease in T_{LBCA} increases the average loop bandwidth estimate.

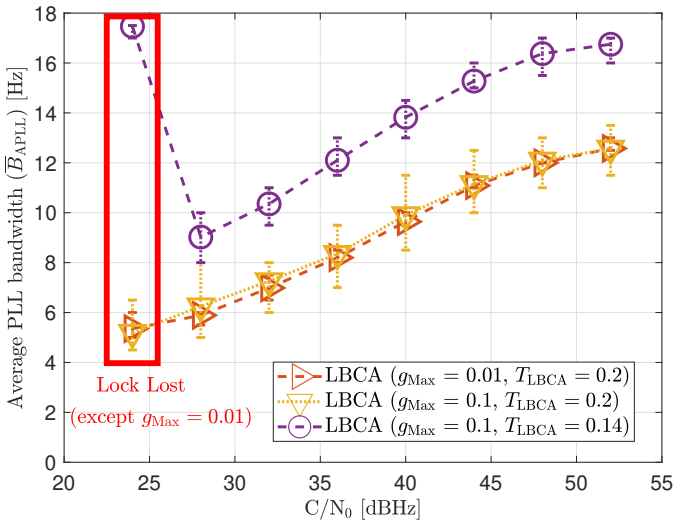
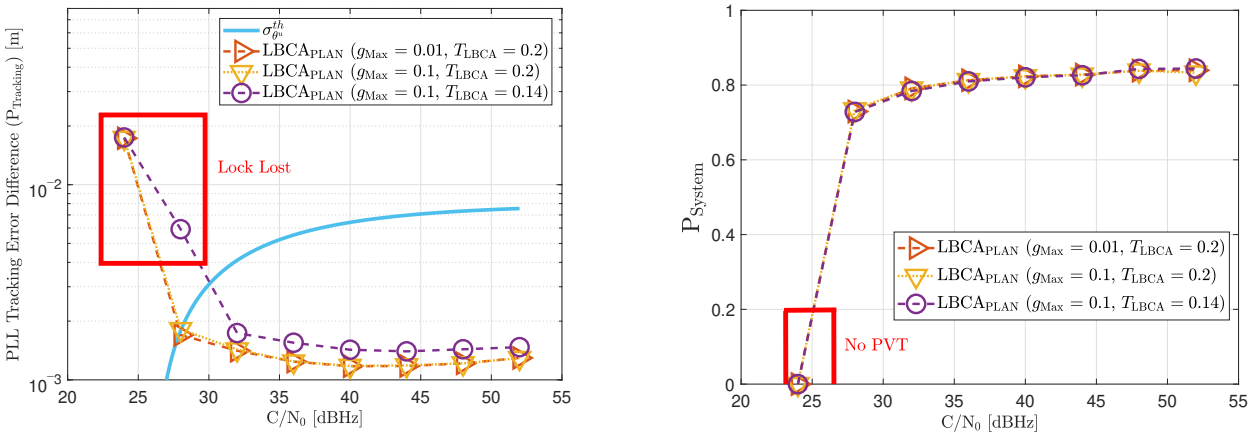


Figure 24. Average LBCA PLL loop bandwidth of SV G4 tracking in static scenario.

Figure 25 shows the tracking performance of the LBCA with PLAN interpolation STL. Due to approximation errors, the lock of SV G4 is not achieved at 24 dBHz even with a low g_{Max}. These errors make the algorithm respond slower to achieve adequate loop bandwidth. Otherwise, the static tracking performance of this algorithm seems to be very similar to the LBCA technique.



(**a**) Tracking performance of SV G4

(**b**) System performance

Figure 25. Static performance of LBCA PLL with PLAN interpolation at different C/N_0 levels.

Figure 26 reveals that the average loop bandwidth estimate of this technique is similar to the one without the piece-wise approximation.

6.2. Dynamic Scenario

First, the performance of the standard tracking should be analyzed with different PLL loop bandwidths. Figure 27a shows that a loop bandwidth below 10 Hz loses the tracking lock of the SV G17 due to the large LOS dynamics at any C/N_0 level. If the loop bandwidth is bigger than 10 Hz, the tracking can handle the LOS dynamics of SV G17. At lower C/N_0 levels (e.g., 32 dBHz), a loop bandwidth of 14 Hz has the best performance, whereas, at a higher C/N_0, a loop bandwidth between 16 Hz and 18 Hz is preferable. Figure 27b shows the system performance of the dynamic scenario. Loop bandwidths lower than 10 Hz achieve a PVT solution from 40 dBHz, mainly because it can track other SV with lower LOS dynamics. However, the system performance is still worse than higher bandwidths. Only a PLL loop bandwidth of 16 Hz can achieve a PVT down to 32 dBHz. At 28 dBHz, a loop bandwidth of 8 Hz has the best system performance, even without a PVT solution.

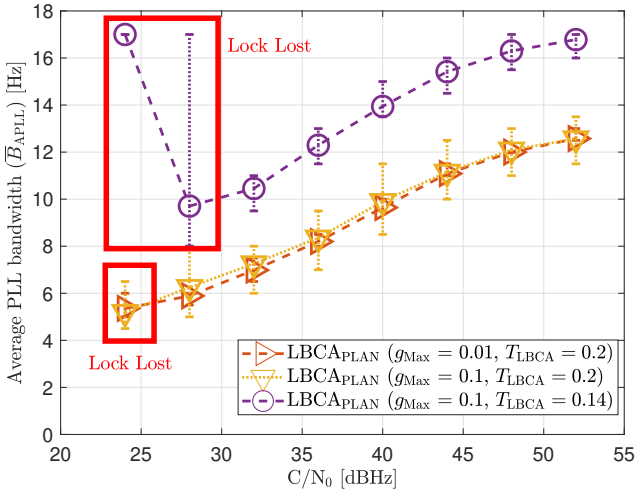
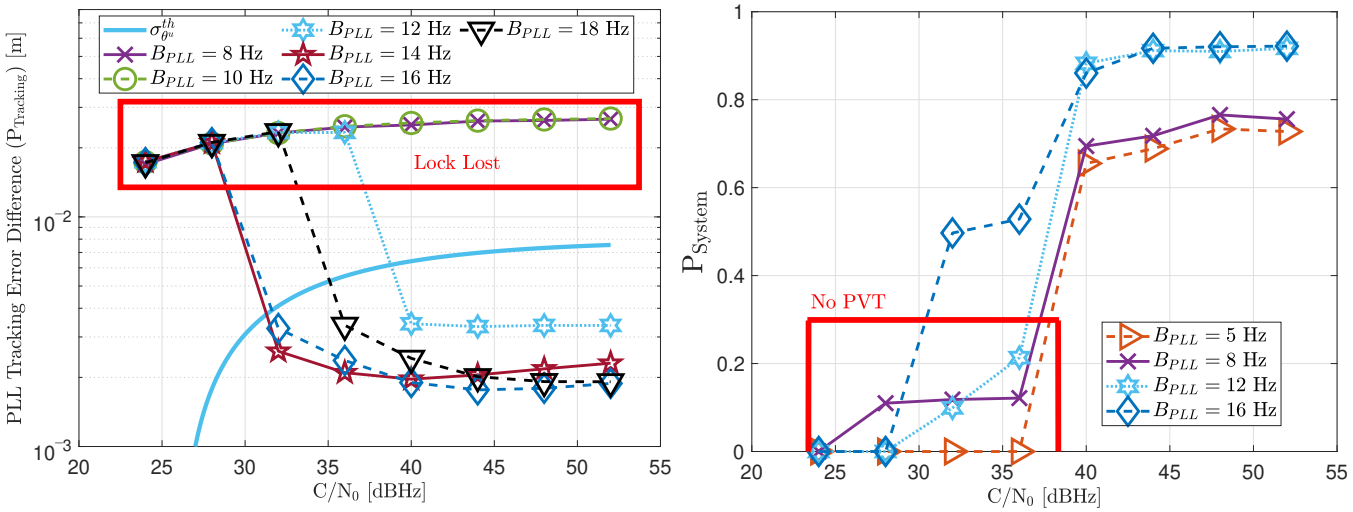


Figure 26. Average LBCA + PLAN PLL loop bandwidth of SV G4 tracking in static scenario.



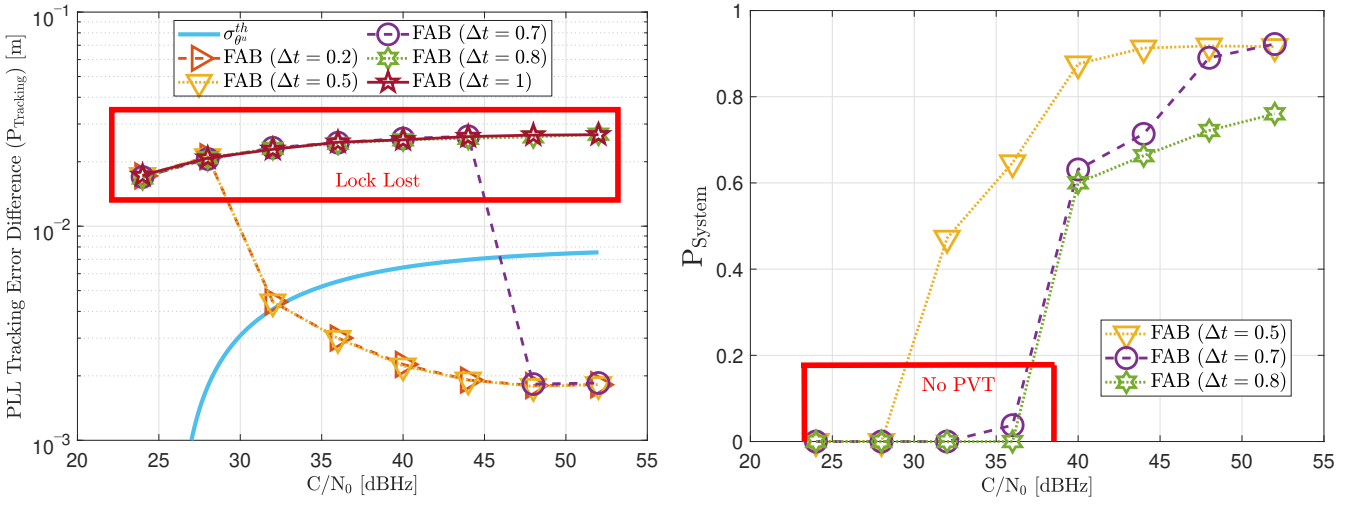
(**a**) Tracking performance of SV G17

(**b**) System performance

Figure 27. Dynamic performance of standard PLL at different C/N_0 levels and loop bandwidths.

The following results present a detailed description of the adaptive tracking techniques' capability to maintain the tracking lock against the dynamic scenario. Ideally, these techniques increase the loop bandwidth once signal dynamics are detected.

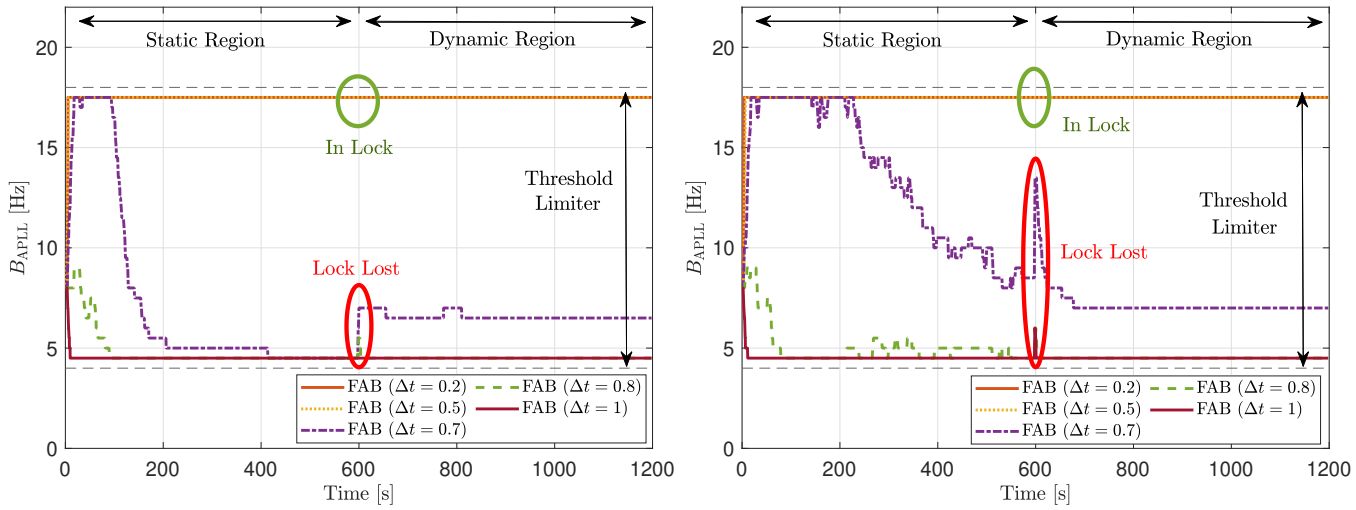
Figure 28 includes the tracking the tracking performance of SV 17 (Figure 28a) and the system performance (Figure 28b) of the FAB technique at different C/N_0 levels. A lower filtered time Δt of the SSE estimator makes the algorithm react faster to dynamics. For instance, a Δt lower than 0.5 s presents better tracking performance than the other configurations. It also presents the best system performance. However, this configuration makes the algorithm too sensitive to dynamics, always having the maximum loop bandwidth possible (17.5 Hz) independent of the C/N_0 level (see Figure 29). If Δt is greater than 0.7, the algorithm cannot track SV 17 at any C/N_0 level. With this configuration, the loop bandwidth goes to the lower threshold 4.5 Hz at any C/N_0 level. With Δt equal to 0.7 s, the estimated loop bandwidth achieves a value between the thresholds. Once dynamics are present, the FAB tries to increase the loop bandwidth to follow the signal dynamics. However, the algorithm does not succeed in maintaining the lock. A better solution should be a Δt between 0.5 s and 0.7 s.



(**a**) Tracking performance of SV G17

(**b**) System performance

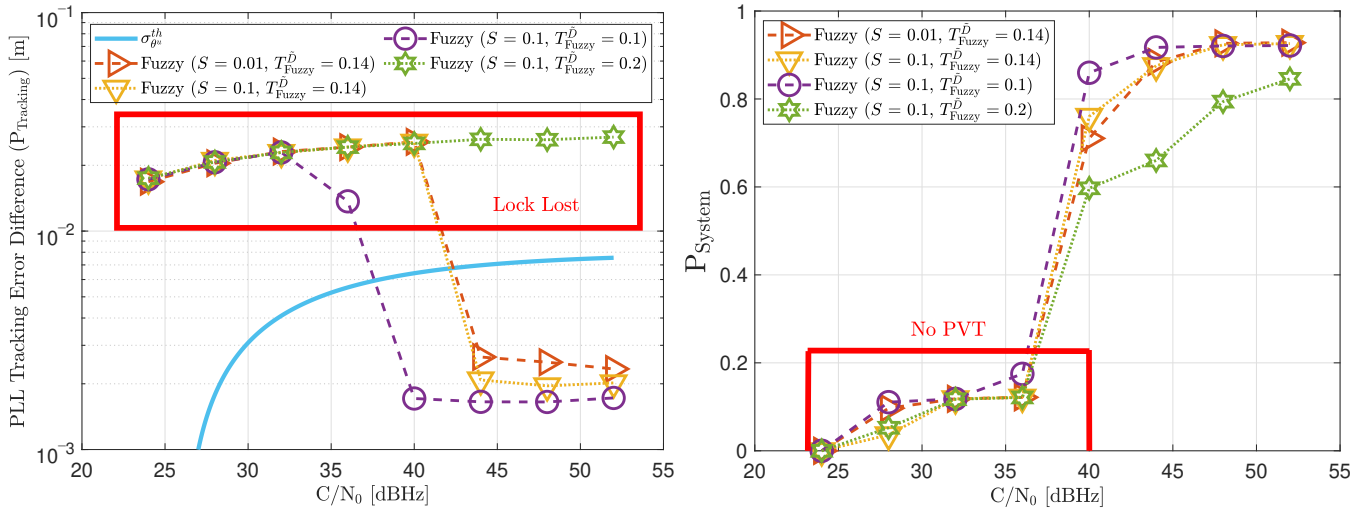
Figure 28. Dynamic performance of FAB at different C/N_0 levels.



(**a**) C/N_0 at 32 dBHz (**b**) C/N_0 at 44 dBHz

Figure 29. PLL loop bandwidth update with FAB of SV G17 tracking in dynamic scenario.

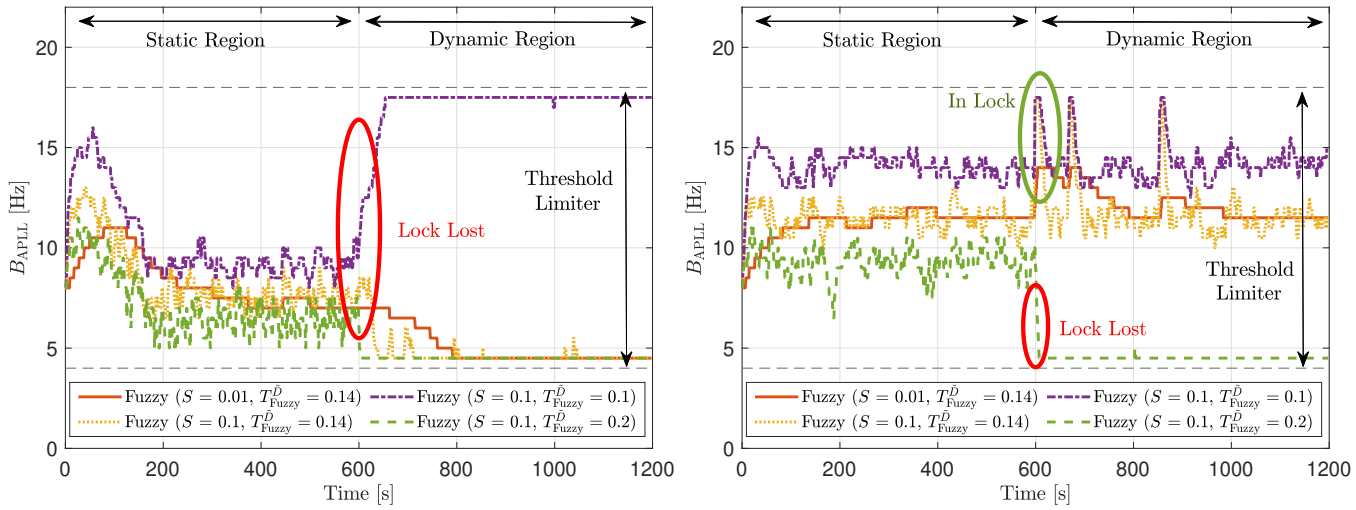
Figure 30 shows the dynamic performance of the FL technique. A lower $T^{\tilde{D}}_{Fuzzy}$ presents better tracking and system performance. Figure 30a shows that a $T^{\tilde{D}}_{Fuzzy}$ equals to 0.1 achieves the lock of SV 17 down to 40 dBHz, whereas $T^{\tilde{D}}_{Fuzzy} = 0.2$ do not achieve lock at any C/N_0 level. Regarding the system performance, any configuration of the FL technique does not achieve a PVT solution up to 36 dBHz (see Figure 30b).



(**a**) Tracking performance of SV G17 (**b**) System performance

Figure 30. Dynamic performance of FL at different C/N_0 levels.

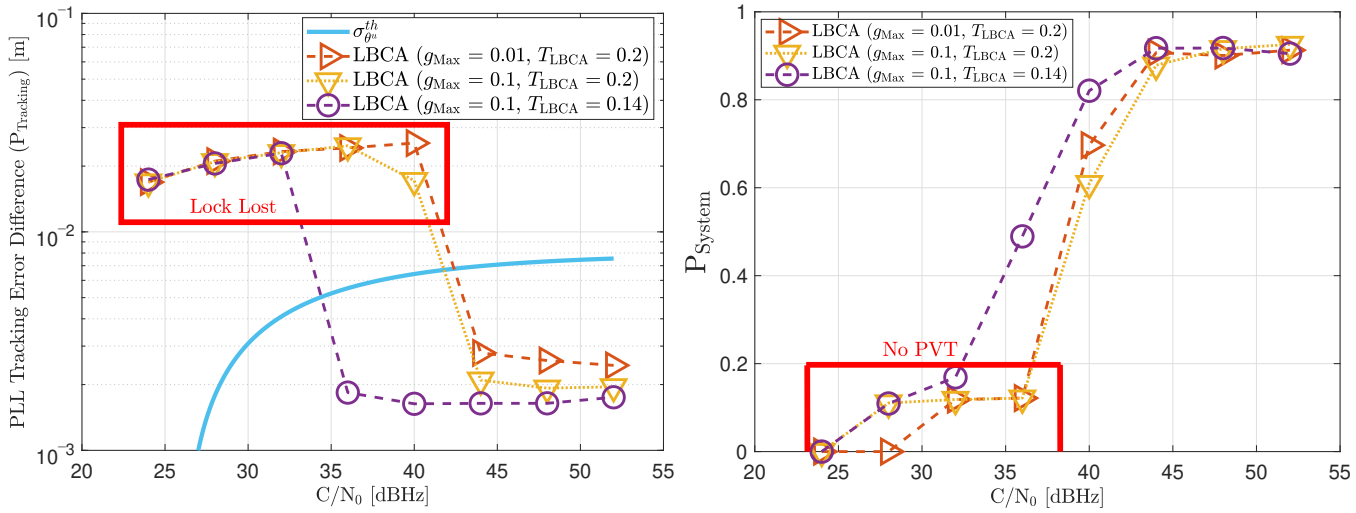
Figure 31 shows how the PLL loop bandwidth adjusts during the dynamic scenario using the presented FL configurations. At 32 dBHz, all the configurations of the FL fail with tracking the signal. When the fuzzy technique with the lowest $T^{\tilde{D}}_{Fuzzy}$ loses lock, unlike the other configurations, the loop bandwidth goes to the maximum loop bandwidth threshold. This means that the loss of lock is considered as dynamics. This result is incorrect since the loop bandwidth should tend to zero to filter all the erroneous phase errors. At 44 dBHz, all the FL's presented configurations can track the signal dynamics except with low sensitivity to dynamics, $T^{\tilde{D}}_{Fuzzy} = 0.2$. Once this last configuration loses lock, the loop bandwidth goes to the lower loop bandwidth threshold, 4.5 Hz.



(**a**) C/N_0 at 32 dBHz (**b**) C/N_0 at 44 dBHz

Figure 31. PLL loop bandwidth update with FL of SV G17 tracking in dynamic scenario.

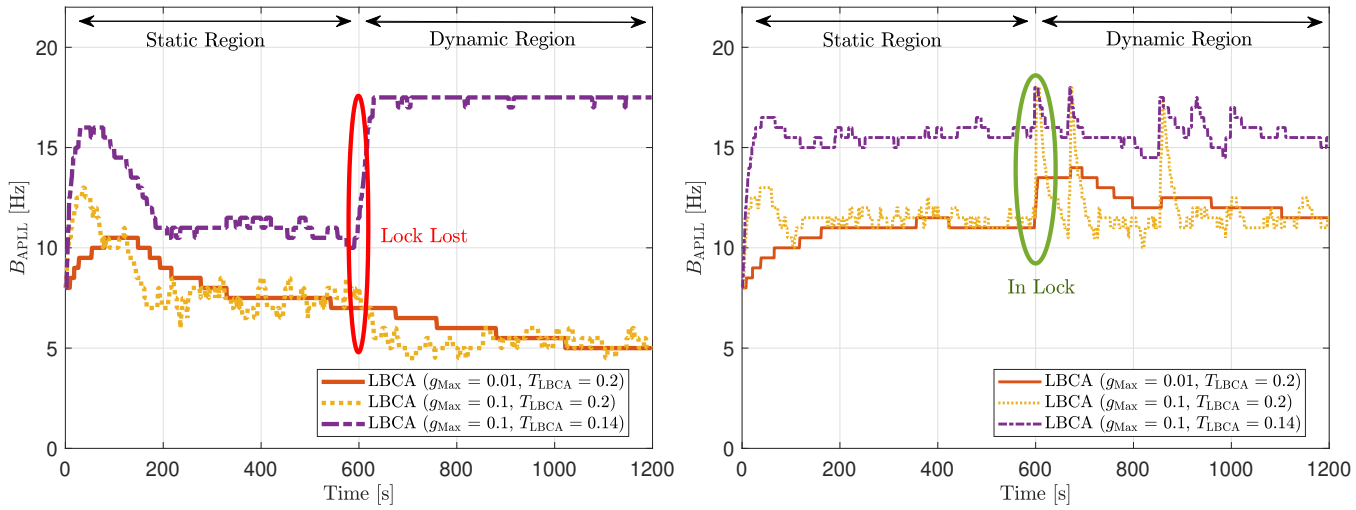
Figures 32 and 33 show that the LBCA technique presents a better dynamic performance with a lower dynamic threshold ($T_{LBCA} = 0.14$). A bigger scale factor g_{Max} improves the tracking performance of SV 17 (see Figure 32a) because of the bigger loop bandwidth update. However, a bigger update can lead also to instabilities and worse performance in the presence of lower LOS dynamics. Figure 32b shows that, at $T_{LBCA} = 0.2$, a lower scale factor can achieve better system performance at some C/N_0 levels (e.g., 40 dBHz).



(**a**) Tracking performance of SV G17 (**b**) System performance

Figure 32. Dynamic performance of LBCA at different C/N_0 levels.

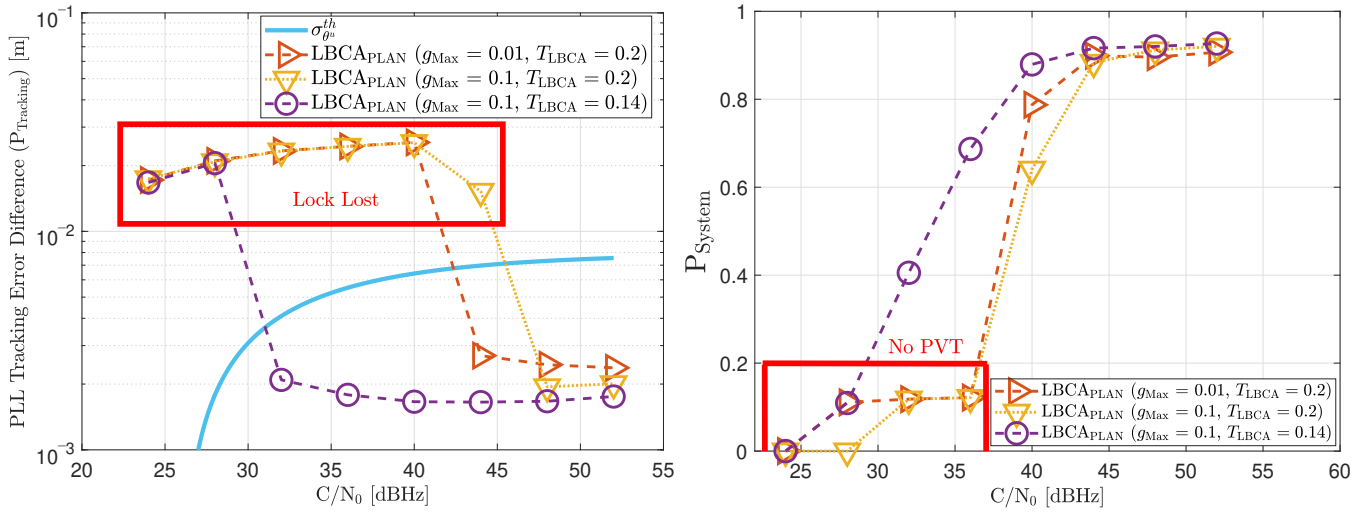
At 32 dBHz, the LBCA technique cannot achieve lock at any configurations (see Figure 33a). The same as in the FL technique happens when the algorithm is too sensitive to dynamics (e.g., $T_{LBCA} = 0.14$) that it goes erroneously to the highest loop bandwidth when the lock is lost. At 44 dBHz, all the configurations can track SV 17 during the dynamic scenario. Figure 33b presents the effects of the dynamic threshold and the scale factor. A bigger dynamic threshold leads to bigger bias of the estimated loop bandwidth. This is also observed in Figure 24. Moreover, a bigger scale factor results in a noisier loop bandwidth estimate, but with a better capability to react faster.



(**a**) C/N_0 at 32 dBHz (**b**) C/N_0 at 44 dBHz

Figure 33. PLL loop bandwidth update with LBCA of SV G17 tracking in dynamic scenario.

The LBCA with PLAN interpolation presents a similar dynamic performance as the LBCA at high C/N_0 levels (see Figure 34). At lower C/N_0, the dynamic performance using a scale factor of $g_{\text{Max}} = 0.1$ and a normalized dynamics threshold $T_{LBCA} = 0.14$ stands out above the rest of the configurations. This configuration maintains the lock even at 32 dBHz. In addition, the configuration with $g_{\text{Max}} = 0.1$ and $T_{LBCA} = 0.2$ has poor tracking performance at 44 dBHz. However, the system performance is similar to the other configurations at that C/N_0 level.



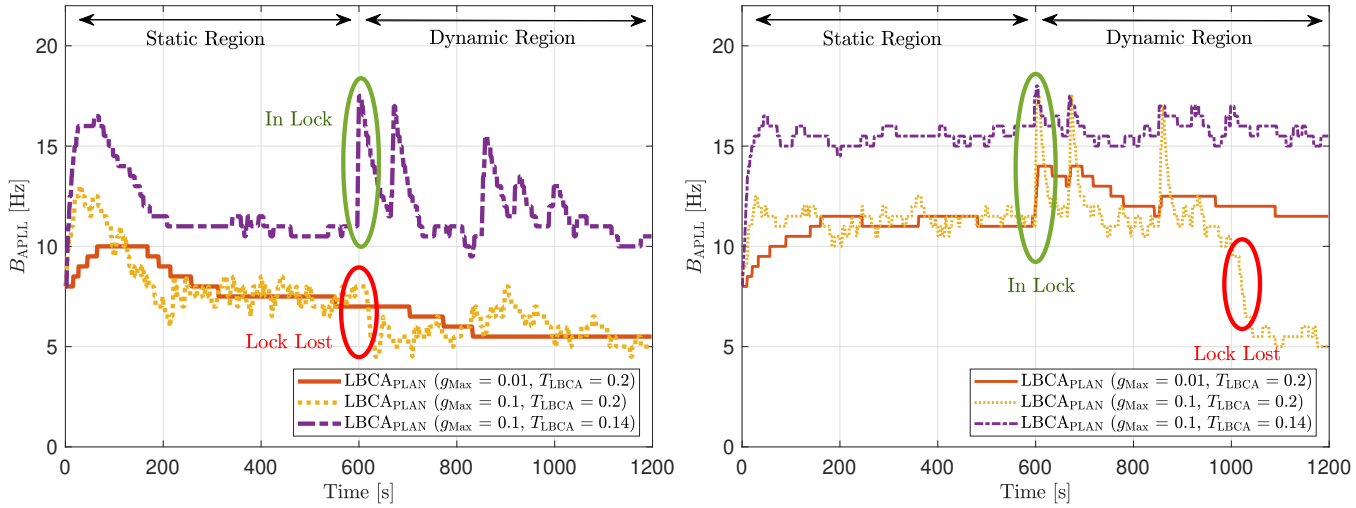
(**a**) Tracking performance of SV G17 (**b**) System performance

Figure 34. Dynamic performance of LBCA with PLAN interpolation at different C/N_0 levels.

The LBCA + PLAN technique with $S = 0.1$ and $T_{LBCA} = 0.14$ is the unique adaptive tracking technique that can still track the signal at 32 dBHz in the presence of high LOS jerk dynamics. Once dynamics are present, the estimated loop bandwidth goes from 11 Hz to 17.5 Hz to follow the phase error dynamics (see Figure 35a). After that, the loop bandwidth goes back to 11 Hz and repeats when new dynamics are detected.

The LBCA + PLAN technique with $S = 0.1$ and $T_{LBCA} = 0.2$ loses lock at time $t = 1000$ s (see Figure 35b). This explains the poor tracking performance of this configuration at that C/N_0 level (see Figure 34a). The high scale-factor causes unstable loop bandwidth estimates. It can happen that, once dynamics are present, the loop band-

width estimate has such a low value that it is not able to respond to signal dynamics and loses lock.



(**a**) C/N_0 at 32 dBHz (**b**) C/N_0 at 44 dBHz

Figure 35. PLL loop bandwidth update with LBCA with PLAN interpolation of SV G17 tracking in dynamic scenario.

6.3. Total System Performance

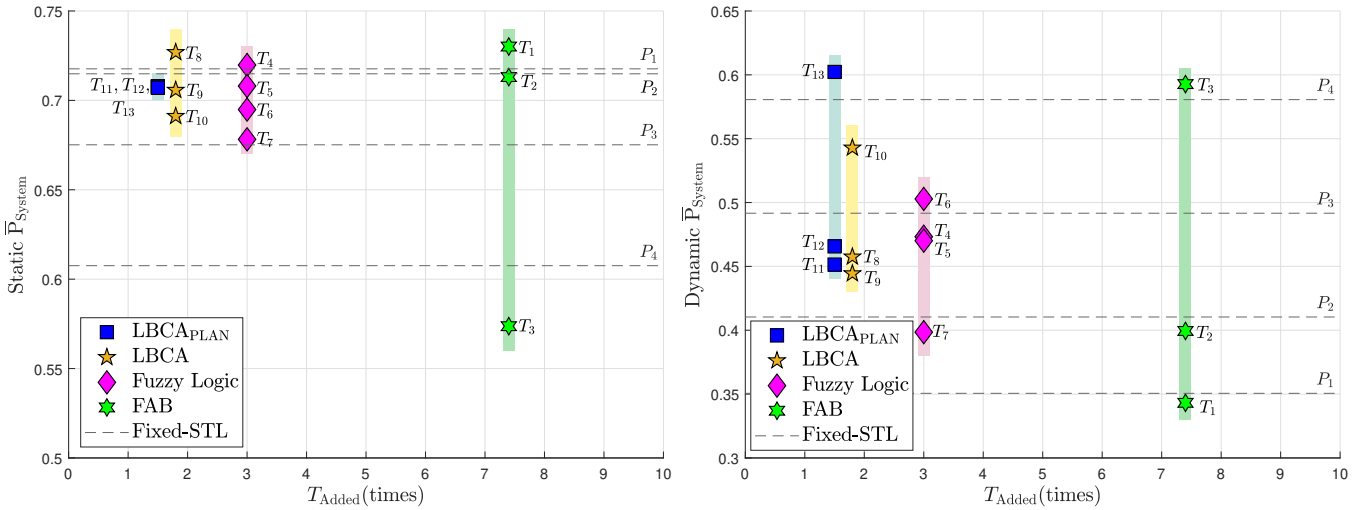
This section summarizes the previous sections' obtained results, taking a single quality factor that determines the adaptive tracking techniques' system performance in a static and a dynamic scenario. The average system performance $\overline{\mathrm{P}}_{\mathrm{System}}$ (see Equation (55)) for both the static and dynamic scenario is calculated and displayed in Table 6. The labels-column in this table is used for further analysis in a subsequent section. Furthermore, for each tracking technique, the best system performance is marked in green, whereas the worst one is in red.

Table 6. System performance of adaptive tracking techniques.

Tracking Technique	Configuration	Label	$\overline{\mathrm{P}}_{\mathrm{System}}$ Static	$\overline{\mathrm{P}}_{\mathrm{System}}$ Dynamic
Standard PLL	$B_{\mathrm{PLL}} = 5$Hz	P_1	0.718	0.351
	$B_{\mathrm{PLL}} = 8$Hz	P_2	0.715	0.410
	$B_{\mathrm{PLL}} = 12$Hz	P_3	0.675	0.492
	$B_{\mathrm{PLL}} = 16$Hz	P_4	0.608	0.581
FAB	$\Delta t = 0.8$ s	T_1	0.730	0.343
	$\Delta t = 0.7$ s	T_2	0.713	0.399
	$\Delta t = 0.5$ s	T_3	0.574	0.593
FL	$S = 0.01$, $T^{\tilde{D}}_{\mathrm{Fuzzy}} = 0.14$	T_4	0.720	0.473
	$S = 0.1$, $T^{\tilde{D}}_{\mathrm{Fuzzy}} = 0.14$	T_5	0.708	0.470
	$S = 0.1$, $T^{\tilde{D}}_{\mathrm{Fuzzy}} = 0.1$	T_6	0.695	0.503
	$S = 0.1$, $T^{\tilde{D}}_{\mathrm{Fuzzy}} = 0.2$	T_7	0.678	0.398
LBCA	$S = 0.01$, $T_{\mathrm{LBCA}} = 0.2$	T_8	0.727	0.458
	$S = 0.1$, $T_{\mathrm{LBCA}} = 0.2$	T_9	0.691	0.444
	$S = 0.1$, $T_{\mathrm{LBCA}} = 0.14$	T_{10}	0.706	0.543
LBCA + PLAN	$S = 0.01$, $T_{\mathrm{LBCA}} = 0.2$	T_{11}	0.708	0.466
	$S = 0.1$, $T_{\mathrm{LBCA}} = 0.2$	T_{12}	0.708	0.451
	$S = 0.1$, $T_{\mathrm{LBCA}} = 0.14$	T_{13}	0.707	0.602

6.4. Performance vs. Complexity Comparison

Figure 36 contains the performance and complexity comparison of the implemented adaptive tracking techniques for a third-order PLL in a static (Figure 36a) and dynamic scenario (Figure 36b). These graphs combine the data of Tables 2 and 6.



(**a**) Static scenario (**b**) Dynamic scenario

Figure 36. System performance vs. added time complexity comparison.

In the static scenario, the standard tracking loop has the best average system performance with a PLL loop bandwidth of 5 Hz (P_1), whereas the worst performance is performed with 16 Hz (P_4). However, the opposite occurs in the dynamic scenario. A greater loop bandwidth improves the average system performance. This shows the big trade-off that exists in standard tracking loops between the loop bandwidth and the type of scenario.

The adaptive tracking algorithms should minimize this trade-off, having a good performance in both scenarios. The FAB technique with a high decay time Δt (e.g., 0.8 s, T_1) results in best performance in static scenario, but poor in dynamic scenario. A lower decay time (e.g., 0.5 s, T_3) have the opposite behavior—poor in static and good in dynamic. There is no minimization of the mentioned trade-off. The FAB technique presents poor results, while having high complexity.

The FL technique presents good system performance in a static scenario with $T^{\tilde{D}}_{\text{Fuzzy}} = 0.14$ (T_4, T_5). A lower (T_6) or a higher (T_7) fuzzy threshold decreases the performance. In a dynamic scenario, a lower fuzzy (T_6) threshold achieves better performance. T_6, T_5 and T_4 show an improvement over the standard PLL. However, the complexity is still significant.

The LBCA and LBCA with PLAN interpolation achieve good system performance, while having a low complexity. In the LBCA technique, a lower scale factor g_{Max} presents better performance in the static and the dynamic scenario with the same normalized dynamics threshold ($T_{\text{LBCA}} = 0.2$, T_8 and T_9). Furthermore, a lower T_{LBCA} deteriorates the static system performance, but improves the dynamic one. In the LBCA with PLAN interpolation, the presented configurations (T_{11}, T_{12}, T_{13}) have the same good static system performance. The dynamic performance of T_{13} stands out above all the adaptive tracking techniques. To conclude, this technique presents the best performance, maintaining the lowest complexity.

7. Discussion

The previous section collects all the obtained results to evaluate the FAB, FL, LBCA, and LBCA + PLAN techniques. The adaptive tracking techniques are evaluated in a static and a dynamic scenario at different C/N_0 levels. For each scenario, one specific tracking

channel (SV G4 in static, SV G17 in dynamic) is analyzed to understand how the presented algorithms work using different configurations. In addition, the metric for the system performance (see Equation (55)) is measured for each adaptive tracking technique and compared to its complexity (see Figure 36).

Figure 37 compares the measured static system performance with the dynamic system performance of the adaptive tracking techniques with different configurations. This Figure is based on Table 6 and synthesizes all the previous sections' results into a single graph. The performance of the standard PLL at different loop bandwidths is also included. The dashed black line is an interpolation of the standard tracking results. Above this line, the values present a better system performance than the standard PLL, whereas results located below this line indicate a worse system performance. The STL clearly demonstrates the loop-bandwidth trade-off between static and dynamic scenarios The best-performing adaptive tracking technique should have the highest average system performance in static and dynamic scenarios.

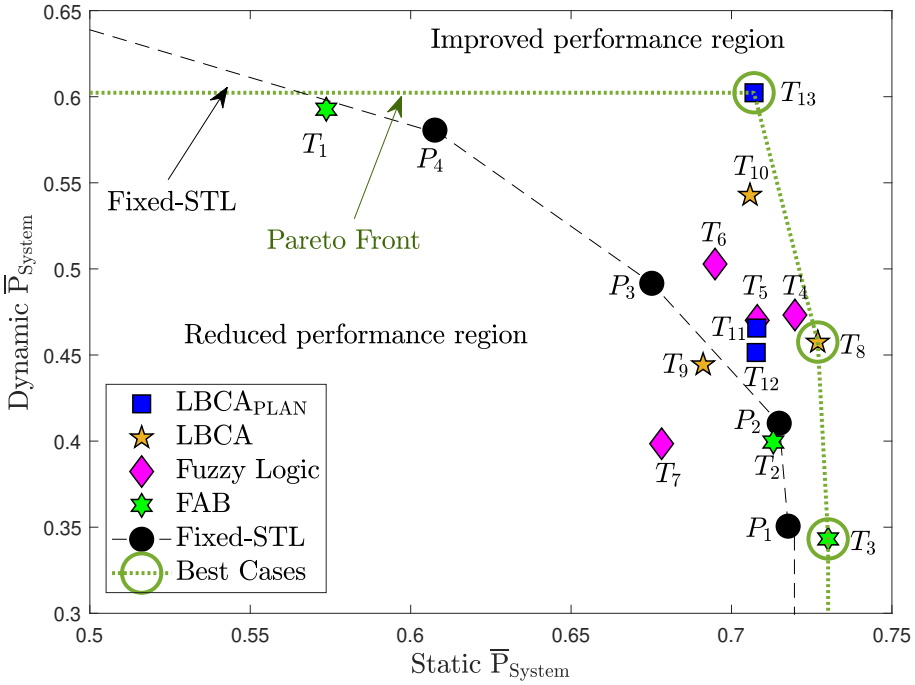


Figure 37. System performance comparison between dynamic and static scenario.

The FAB technique (T_1, T_2, T_3) has a similar performance as the standard PLL while having the highest complexity. Therefore, although not all the possible configurations have been tested, it seems that this technique does not bring any improvement. A better choice of the decay time Δt can improve the performance of this algorithm. The FL technique presents better performance than the standard tracking except for one configuration, T_7. A fuzzy threshold $T^{\tilde{D}}_{\text{Fuzzy}}$ of 0.2 makes the algorithm too sensitive to noise, worsening the dynamic performance (see Figure 30b). Also, the high scale factor ($S = 0.1$) results in a more unstable tracking, leading to a worse static performance at low C/N_0 (see Figure 21b). The other configurations (T_4, T_5, T_6) are above the standard PLL's performance. A lower fuzzy threshold $T^{\tilde{D}}_{\text{Fuzzy}}$ than configuration T_7 gains dynamic and static performance (T_5, T_4). If $T^{\tilde{D}}_{\text{Fuzzy}}$ is even lower (T_6), the dynamic performance improves with the trade-off of losing some static performance. The LBCA technique also achieves a good performance except for T_9. A large scale factor ($g_{\text{Max}} = 0.1$) with a high normalized threshold dynamics T_{LBCA}, as in the FL technique, can make the tracking unstable. A lower scale factor (T_8) significantly improves the static performance and slightly the dynamic performance, whereas a low threshold dynamics (T_{10}) significantly improves the dynamic performance and slightly

the static one. Finally, the LBCA with PLAN technique presents better performance than standard PLL in all the presented configurations (T_{11}, T_{12}, T_{13}). These three configurations have the same static performance, and the dynamic performance is improved significantly when the normalized threshold dynamics is equal to the optimal normalized dynamics ($T_{\mathrm{LBCA}} = 1/7$).

The green dotted line in the top right of the graph is the Pareto front [49–51]. It isolates a set of optimal values to facilitate the trade-off between static and dynamic performance. All other values are sub-optimal in the Pareto sense. There are three values on the Pareto front: T_3, T_8 and T_{13}. However, T_3 (FAB) exhibits faulty behavior as it always adapts to the minimum loop bandwidth of 4.5 Hz, and cannot be used for reliable tracking. Therefore, it is excluded. The remaining correct performance-optimal adaptive tracking techniques are LBCA with PLAN using $S = 0.1$ and $T_{\mathrm{LBCA}} = 0.14$ (T_{13}), and LBCA with $S = 0.01$ and $T_{\mathrm{LBCA}} = 0.2$ (T_8). The optimization demonstrates the superiority of LBCA based adaptive algorithms with the correct tuning.

8. Conclusions

This paper presents a method to compare the performance and complexity of different state-of-the-art adaptive loop-bandwidth tracking techniques. First, the theoretical background of an STL is presented. Second, the FAB, the FL, and the LBCA adaptive tracking techniques are described. Third, the metrics used to evaluate the adaptive tracking techniques are analyzed. Furthermore, each adaptive tracking technique's complexity is measured in terms of the required number of operations as well as the time complexity. Fourth, the algorithms are implemented on the GOOSE receiver platform with simulated static and dynamic scenarios. The dynamic scenarios represent harsh vehicular movement.

The results show each algorithm's tracking and system performance in a static and a dynamic simulated scenario. The FAB presents insignificant improvements compared to STLs, has poor system performance and is the most complex adaptive algorithm. The FL is conceptually simple and shows—when tuned correctly– significant improvement to STLs. It is an algorithm with much potential. The LBCA and LBCA + PLAN techniques have superior system performance with the least amount of complexity. These algorithms–depending on the configurations–are also determined optimal in the static to dynamic trade-off, illustrating a clear advantage.

A broader statistical evaluation to test the adaptive tracking techniques with a wider variety of configurations, C/N_0 levels, and scenarios is proposed for future work. It can additionally include multi-path and non-line-of-sight (NLOS) effects. This would improve the generalization and applicability of the adaptive tracking algorithms, and make these less dependent on the chosen scenarios. However, this paper presents an initial investigation and includes a framework to evaluate the adaptive techniques. The adaptive techniques in this paper are only evaluated for the PLL. However, a tracking channel also has a DLL and FLL. Therefore, future work to incorporate adaptive techniques to these loops are also proposed. The results for the LBCA and FL have shown that these algorithms' performances are sensitive to their configurations. Therefore, further research to optimize the configurations are suggested. Furthermore, a vector-based LBCA approach which considers multiple GNSS SVs simultaneously is also suggested. Lastly, an extension of the adaptive loop-bandwidth algorithm can be done by also adapting the integration time.

This paper encloses an initial study of robust tracking techniques with the primary objective of developing a method that achieves the best performance while maintaining low complexity. The LBCA with the PLAN interpolation is currently the best candidate among the other techniques.

Author Contributions: Conceptualization, I.C.; methodology, I.C. and J.R.v.d.M.; software, I.C.; validation, I.C.; formal analysis, I.C.; investigation, I.C. and J.R.v.d.M.; resources, I.C. and A.R.; data curation, I.C.; writing—original draft preparation, I.C. and J.R.v.d.M.; writing—review and editing, I.C., J.R.v.d.M., J.N. and A.R.; visualization, I.C.; supervision, J.N.; project administration, A.R.;

funding acquisition, W.F. and A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partly carried out within the GUARDIAN project funded by the German Federal Ministry of Transport and Digital Infrastructure (FKZ 50NP1707), which is gratefully acknowledged.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://owncloud.fraunhofer.de/index.php/s/LGoWPVtV5xbQ9mB.

Acknowledgments: This work has been partly carried out within the GUARDIAN project funded by the German Federal Ministry of Transport and Digital Infrastructure (FKZ 50NP1707), which is gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADC	analog-to-digital converter
BET	backward Euler transform
C/N_0	carrier-to-noise density ratio
COG	center of gravity
CRB	Cramér-Rao bound
DLL	delay locked loop
FAB	fast adaptive bandwidth
FAP	FLL-assisted-PLL
FL	fuzzy logic
FLL	frequency locked loop
FMM	fuzzy mean method
FPGA	field-programmable gate array
GNSS	global navigation satellite system
GOOSE	GNSS-Receiver with Open Software Interface
GPS	Global Positioning System
IAD	integration and dump
IIR	infinite impulse response
IQ	in-phase and quadrature-phase
KF	Kalman filtering
LBCA	loop-bandwidth control algorithm
LOS	line-of-sight
NCO	numerically controlled oscillator
NLOS	non-line-of-sight
OGRP	open GNSS receiver protocol
PAD	PLL-assisted-DLL
PCIe	peripheral component interconnect express
PLAN	piecewise linear approximation of nonlinearities
PLI	phase-lock indicator
PLL	phase locked loop
PVT	position, velocity, and time
RFCS	radio-frequency constellation simulator
RFFE	radio-frequency front-end
SSE	steady state error
SSM	state space model
STL	scalar tracking loop
SV	satellite vehicle

TCXO	temperature-compensated crystal oscillator
ToA	time of arrival
VT	vector tracking

References

1. Kaplan, E.D.; Hegarty, C.J. *Understanding GPS: Principles and Applications*, 2nd ed.; Artech House Mobile Communications Series; Artech House: Norwood, MA, USA, 2006.
2. Won, J.; Pany, T. Signal Processing. In *Springer Handbook of Global Navigation Satellite Systems*; Springer International Publishing: Cham, Switzerland, 2017; pp. 401–442. [CrossRef]
3. Liu, J.; Cui, X.; Lu, M.; Feng, Z. Vector tracking loops in GNSS receivers for dynamic weak signals. *J. Syst. Eng. Electron.* **2013**, *24*, 349–364. [CrossRef]
4. Dietmayer, K.; Saad, M.; Strobel, C.; Garzia, F.; Overbeck, M.; Felber, W. Real time implementation of Vector Delay Lock Loop on a GNSS receiver hardware with an open software interface. In Proceedings of the 2019 European Navigation Conference (ENC), Warsaw, Poland, 9–12 April 2019; pp. 1–7. [CrossRef]
5. Gardner, F.M. *Phaselock Techniques*, 3rd ed.; Wiley: New York, NY, USA, 2005.
6. Jwo, D.J. Optimisation and sensitivity analysis of GPS receiver tracking loops in dynamic environments. *IEE Proc. Radar Sonar Navig.* **2001**, *148*, 241–250. [CrossRef]
7. Cortés, I.; Iñiguez de Gordoa, J.A.; van der Merwe, J.R.; Rügamer, A.; Felber, W. Performance and complexity comparison of adaptive loop-bandwidth tracking techniques. In Proceedings of the 2020 International Conference on Localization and GNSS (ICL-GNSS), Tampere, Finland, 2–4 June 2020; pp. 1–7. [CrossRef]
8. Stevanovic, S.; Pervan, B. A GPS phase-locked loop performance metric based on the phase discriminator output. *Sensors* **2018**, *18*, 296. [CrossRef] [PubMed]
9. Vilà-Valls, J.; Linty, N.; Closas, P.; Dovis, F.; Curran, J.T. Survey on signal processing for GNSS under ionospheric scintillation: Detection, monitoring, and mitigation. *Navigation* **2020**, *67*, 511–536. [CrossRef]
10. Skone, S.; Lachapelle, G.; Yao, D.; Yu, W.; Watson, R. Investigating the impact of ionospheric scintillation using a GPS software receiver. In Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005), Long Beach, CA, USA, 13–16 September 2005; pp. 1126–1137.
11. Muthuraman, K. Tracking Techniques for GNSS Data/Pilot Signals. Ph.D. Thesis, University of Calgary, Calgary, AB, Canada, 2010.
12. Legrand, F.; Macabiau, C.; Issler, J.; Lestarquit, L.; Mehlen, C. Improvement of pseudorange measurements accuracy by using fast adaptive bandwidth lock loops. In Proceedings of the 13th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 2000), Salt Lake City, UT, USA, 19–23 September 2000; pp. 2346–2356.
13. Legrand, F.; Macabiau, C. Results of the implementation of the fast adaptive bandwidth lock loops on a real GPS receiver in a high dynamics context. In Proceedings of the International Conference on Global Navigation Satellite System (GNSS 2001), Seville, Spain, 8–11 May 2001.
14. Chen, Y.; Zheng, H.; Wang, Y. Adaptive bandwidth PLL design based on fuzzy logic control. In Proceedings of the 4th IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, Beijing, China, 1–3 November 2011; pp. 645–648. [CrossRef]
15. Hangos, K.M.; Lakner, R.; Gerzson, M. Applied Optimization 60. In *Intelligent Control Systems: An Introduction with Examples*; Springer: New York, NY, USA, 2004; Volume 1.
16. Cortes, I.; Van der Merwe, J.R.; Rügamer, A.; Felber, W. Adaptive loop-bandwidth control algorithm for scalar tracking loops. In Proceedings of the 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, OR, USA, 20–23 April 2020; pp. 1178–1188. [CrossRef]
17. Sun, X.; Qin, H.; Niu, J. Comparison and analysis of GNSS signal tracking performance based on kalman filter and traditional loop. *WSEAS Trans. Signal Process.* **2013**, *9*, 99–108.
18. Rao, K.D.; Narayana, J.L. An approach for a faster GPS tracking extended Kalman filter. *Navigation* **1995**, *42*, 619–632. [CrossRef]
19. Shen, J.; Cui, X.; Zhu, Y.; Lu, M. A frequency-compensation algorithm in Kalman filter-based tracking loop for high-dynamics GNSS receivers. In Proceedings of the 2016 International Technical Meeting of the Institute of Navigation (ION ITM), Monterey, CA, USA, 25–28 January 2016; pp. 429–436. [CrossRef]
20. Lin, W.; Chen, X.; Wang, Y.; He, D. The design of asynchronous Kalman filter-based tracking loop in digital domain. In Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017), Portland, OR, USA, 25–29 September 2017; pp. 3853–3864. [CrossRef]
21. Susi, M.; Borio, D. Kalman filtering with noncoherent integrations for Galileo E6-B tracking. *Navigation* **2020**, *67*, 601–618. [CrossRef]
22. Won, J.W.; Eissfiller, B. A tuning method based on signal-to-noise power ratio for adaptive PLL and its relationship with equivalent noise bandwidth. *IEEE Commun. Lett.* **2013**, *17*, 393–396. [CrossRef]
23. Won, J.H. A novel adaptive digital phase-lock-loop for modern digital GNSS receivers. *IEEE Commun. Lett.* **2014**, *18*, 46–49. [CrossRef]

24. Overbeck, M.; Garzia, F.; Popugaev, A.; Kurz, O.; Forster, F.; Felber, W.; Ayaz, A.S.; Ko, S.; Eissfeller, B. GOOSE-GNSS Receiver with an Open Software Interface. In Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015), Tampa, FL, USA, 14–18 September 2015.
25. Seybold, J. *GOOSE: Open GNSS Receiver Platform*; Technical Report; TeleOrbit GmbH: Nuremberg, Germany, 2020. Available online: https://teleorbit.eu/en/satnav/ (accessed on 10 January 2021).
26. Aguirre, S.; Hurd, W.; Kumar, R.; Statman, J. *A Comparison of Methods for DPLL Loop Filter Design*; Telecommunications and Data Acquisition Progress Report 42–79; Jet Propulsion Laboratory: Pasadena, CA, USA, 1986.
27. Jury, E.I. *Theory and Application of the Z-Transform Methods*; Wiley: New York, NY, USA, 1964.
28. Taylor, C.J.; Young, P.C.; Chotai, A. *True Digital Control: Statistical Modelling and Non-Minimal State Space Design*; Wiley: New York, NY, USA, 2013; Volume 1.
29. Humphreys, T.E.; Psiaki, M.L.; Kintner, P.M.; Ledvina, B.M. GPS carrier tracking loop performance in the presence of ionospheric scintillations. In Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005), Long Beach, CA, USA, 13–16 September 2005; pp. 156–167.
30. Aguirre, S.; Hurd, W. *Design and Performance of Sampled Data Loops for Subcarrier and Carrier Tracking*; Telecommunications and Data Acquisition Progress Report 42–79; Jet Propulsion Laboratory: Pasadena, CA, USA, 1984.
31. Jaffe, R.; Rechtin, E. Design and performance of phase-lock circuits capable of near-optimum performance over a wide range of input signal and noise levels. *IRE Trans. Inf. Theory* **1955**, *1*, 66–76. [CrossRef]
32. Stephens, S.; Thomas, J. Controlled-root formulation for digital phase-locked loops. *IEEE Trans. Aerosp. Electron. Syst.* **1995**, *31*, 78–95. [CrossRef]
33. López-Salcedo, J.A.; Peral-Rosado, J.A.D.; Seco-Granados, G. Survey on Robust Carrier Tracking Techniques. *IEEE Commun. Surv. Tutor.* **2014**, *12*, 670–688. [CrossRef]
34. Leekwijck, W.V.; Kerre, E.E. Defuzzification: criteria and classification. *Fuzzy Sets Syst.* **1999**, *108*, 159–178. [CrossRef]
35. Domingos, P. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*; Basic Books, Inc.: New York, NY, USA, 2018.
36. Alban, S.; Akos, D.M.; Rock, S.M.; Gebre-Egziabher, D. Performance analysis and architectures for INS-aided GPS tracking loops. In Proceedings of the 2003 National Technical Meeting of The Institute of Navigation (ION NTM), Anaheim, CA, USA, 22–24 January 2003; pp. 611–622.
37. Amin, H.; Curtis, K.; Hayes-Gill, B. Piecewise linear approximation applied to nonlinear function of a neural network. *IEE Proc. Circuits Devices Syst.* **1997**, *144*, 313–317. [CrossRef]
38. Kay, S.M. *Fundamentals of Statistical Signal Processing: Estimation Theory*; Prentice-Hall Signal Processing Series; Prentice Hall: Upper Saddle River, NJ, USA, 1993; Volume 1.
39. Betz, J.W.; Kolodziejski, K.R. Generalized Theory of Code Tracking with an Early-Late Discriminator Part I: Lower Bound and Coherent Processing. *IEEE Trans. Aerosp. Electron. Syst.* **2009**, *45*, 1538–1556. [CrossRef]
40. Cong, L.; Li, X.; Jin, T.; Yue, S.; Xue, R. An adaptive INS-aided PLL tracking method for GNSS receivers in harsh environments. *Sensors* **2016**, *16*, 146. [CrossRef] [PubMed]
41. Bhaskar, S. Exploiting quasi-periodicity in receiver dynamics to enhance GNSS carrier phase tracking. In Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014), Tampa, FL, USA, 8–12 September 2014; pp. 2563–2578. [CrossRef]
42. Josuttis, N.M. *The C++ Standard Library: A Tutorial and Reference*, 2nd ed.; Addison-Wesley Professional: Boston, MA, USA, 2012.
43. OProfile Manual. Available online: https://oprofile.sourceforge.io/doc/index.html (accessed on 14 December 2020).
44. Taskset(1)—Linux Manual Page. Available online: https://man7.org/linux/man-pages/man1/taskset.1.html (accessed on 14 December 2020).
45. Libm. Available online: https://sourceware.org/newlib/libm.html (accessed on 14 December 2020).
46. Atkinson, K. *An Introduction to Numerical Analysis*; Wiley: New York, NY, USA, 1989.
47. Garzia, F.; Strobel, C.; Overbeck, M.; Kumari, N.; Joshi, S.; Förster, F.; Felber, W. A multi-frequency multi-constellation GNSS development platform with an open interface. In Proceedings of the 2016 European Navigation Conference (ENC), Helsinki, Finland, 30 May–2 June 2016; pp. 1–7. [CrossRef]
48. Ogrp—The Open gnss Receiver Protocol. 2016. Available online: https://github.com/Fraunhofer-IIS/ogrp (accessed on 14 December 2020).
49. Schmitendorf, W.; Leitmann, G. A simple derivation of necessary conditions for Pareto optimality. *IEEE Trans. Autom. Control* **1974**, *19*, 601–602. [CrossRef]
50. Gandibleux, X. *Metaheuristics for Multiobjective Optimisation*; Springer: Berlin, Germany, 2004.
51. van der Merwe, J.R.; Maasdorp, F.D.V.; du Plessis, W.P.; Linde, L.P. Practical direction-of-arrival system evaluation. In Proceedings of the 2015 IEEE Radar Conference, Johannesburg, South Africa, 27–30 October 2015; pp. 487–492. [CrossRef]
52. Robust Tracking Techniques Dataset Using GOOSE Receiver. Available online: https://owncloud.fraunhofer.de/index.php/s/LGoWPVtV5xbQ9mB (accessed on 10 January 2021).