sensors

MDPI

Article

Evaluation of Low-Complexity Adaptive Full Direct-State Kalman Filter for Robust GNSS Tracking †

Iñigo Cortés [1,2,*], Johannes Rossouw van der Merwe [3], Elena Simona Lohan [2], Jari Nurmi [2] and Wolfgang Felber [1]

[1] Satellite Based Positioning Systems Department, Fraunhofer IIS, Nordostpark 84, 90411 Nuremberg, Germany
[2] Electrical Engineering, Tampere University, 33014 Tampere, Finland
[3] Focal Point Positioning, Cambridge CB4 3NP, UK
* Correspondence: inigo.cortes@iis.fraunhofer.de; Tel.: +49-911-58061-6426
† This paper is an extended version of our paper published in Proceedings of the 2022 International Conference on Localization and GNSS (ICL-GNSS), Tampere, Finland, 7–9 June 2022; pp. 1–7.

Abstract: This paper evaluates the implementation of a low-complexity adaptive full direct-state Kalman filter (DSKF) for robust tracking of global navigation satellite system (GNSS) signals. The full DSKF includes frequency locked loop (FLL), delay locked loop (DLL), and phase locked loop (PLL) tracking schemes. The DSKF implementation in real-time applications requires a high computational cost. Additionally, the DSKF performance decays in time-varying scenarios where the statistical distribution of the measurements changes due to noise, signal dynamics, multi-path, and non-line-of-sight effects. This study derives the full lookup table (LUT)-DSKF: a simplified full DSKF considering the steady-state convergence of the Kalman gain. Moreover, an extended version of the loop-bandwidth control algorithm (LBCA) is presented to adapt the response time of the full LUT-DSKF. This adaptive tracking technique aims to increase the synchronization robustness in time-varying scenarios. The proposed tracking architecture is implemented in an GNSS hardware receiver with an open software interface. Different configurations of the adaptive full LUT-DSKF are evaluated in simulated scenarios with different dynamics and noise cases for each implementation. The results confirm that the LBCA used in the FLL-assisted-PLL (FAP) is essential to maintain a position, velocity, and time (PVT) fix in high dynamics.

Keywords: global navigation satellite system (GNSS); full direct-state Kalman filter (DSKF); lookup table direct-state Kalman filter (LUT-DSKF); loop-bandwidth control algorithm (LBCA); adaptive tracking techniques

Citation: Cortés, I.; van der Merwe, J.R.; Lohan, E.S.; Nurmi, J.; Felber, W. Evaluation of Low-Complexity Adaptive Full Direct-State Kalman Filter for Robust GNSS Tracking. *Sensors* **2023**, *23*, 3658. https://doi.org/10.3390/s23073658

Academic Editor: Robert Odolinski

Received: 2 March 2023
Revised: 18 March 2023
Accepted: 29 March 2023
Published: 31 March 2023

1. Introduction

Global navigation satellite system (GNSS) receivers require reliable synchronization with incoming GNSS signals to achieve a continuous position, velocity, and time (PVT) solution [1]. The synchronization process consists of two stages: acquisition and tracking. Acquisition coarsely estimates the code phase and the carrier Doppler of received GNSS signals. The tracking stage refines these synchronization parameters and includes the fine estimation of the carrier phase. A successful synchronization permits the decoding of the navigation message and the estimation of the pseudo-range and pseudo-range rate, which finally leads to the PVT calculation [2,3].

The carrier phase ϕ, carrier Doppler f, and code phase τ are the main parameters the GNSS receiver synchronizes with. Standard tracking techniques use scalar tracking loops (STLs) in the tracking stage. This tracking scheme synchronizes with a single synchronization parameter of a GNSS signal at a time [1,3]. Thus, a tracking channel includes three STLs: phase locked loop (PLL), frequency locked loop (FLL), and delay locked loop (DLL). The STL contains a correlator, a discriminator, a loop filter, and a numerically controlled

oscillator (NCO) [4,5]. The configuration parameters of the STL include the discriminator type, the loop bandwidth B, the integration time τ_{int}, the order p, and the correlator spacing Δ_s. These parameters determine the performance and robustness against noise and signal dynamics. The well-known trade-off between noise filtering capabilities and signal dynamics resistance is the main challenge of fix-configured STLs. In particular, this problem is aggravated in time-varying scenarios. These scenarios are characterized by different realizations of signal dynamics, noise, and fading effects that lead to challenges regarding synchronization capability [1]. For instance, a high-order STL with wide loop bandwidth and short integration time is adequate to track rapidly changing parameters. In contrast, a low-order STL with narrow loop bandwidth and long integration time is preferable to track noisy parameters. Therefore, a fixed configuration of the STL is a sub-optimal solution for time-varying scenarios.

Carrier-phase continuity in mobile devices is fundamental to achieving decimeter-level positioning through real-time kinematic (RTK) [6] or precise point positioning (PPP) [7,8]. However, smartphones use ultra-low-cost GNSS chipsets and low-gain antennas leading to poor GNSS observations [9], challenging carrier phase continuity and, in turn, decimeter-level positioning. Moreover, GNSS observations are highly affected by multipath, particularly in dense urban scenarios [10]. Additionally, vehicular scenarios usually experience short outages, where GNSS signals can be shortly blocked by residential buildings, overpasses, or short tunnels, interrupting the GNSS observations. Therefore, searching for a robust tracking technique that maintains the carrier phase continuity under these scenarios is highly necessary.

Size, weight, and power (SWAP) are key metrics for GNSS mass-market chip manufacturing. In particular, power consumption is a relevant topic in mobile devices, and several power-saving techniques have been proposed [11,12]. When the GNSS receiver loses the synchronization of the GNSS signals, the re-acquisition is performed, returning to the acquisition stage. Since acquisition is a power-consuming process, a robust tracking architecture can avoid re-acquisition by not losing the lock of the GNSS signal, decreasing the power consumption significantly.

GNSS receivers like the GOOSE©platform [13] partially implement the tracking stage in hardware (correlators and NCO) and software (discriminators and loop filters). These receivers try to close the loop of all the tracking channels before a new correlation is performed. A low time complexity of the software implementation is essential to close the loop on time, avoiding synchronization failures. Furthermore, the lower the time complexity, the more tracking channels the GNSS receiver can manage. Hence, a low-complexity robust tracking architecture is critical to achieving a low time complexity and, in turn, more tracking channels.

The Kalman filter (KF) is an optimal infinite impulse response (IIR) estimator under the assumption of linear Gaussian error statistics [14–16]. Knowledge of the process noise covariance $\mathbf{Q}$ and the measurement noise covariance $\mathbf{R}$ allows the KF to adapt its coefficients optimally, achieving the minimum mean square error (MMSE) [17]. There are several KF implementation methods in STLs [18] grouped into error-state Kalman-filter (ESKF) and direct-state Kalman filter (DSKF) [19]. The former replaces the loop filter of the STL with a KF [20–23], whereas the latter considers the whole STL as part of the KF [24–28]. The implementation of the DSKF is straightforward due to the relation between the STL's coefficients and the DSKF's Kalman gains [24].

The MMSE is only achieved if prior knowledge of $\mathbf{Q}$ and $\mathbf{R}$ is available or if these are accurately estimated [17]. If this is not the case, the KF converges to a suboptimal solution [29]. Hence, for time-varying scenarios in which $\mathbf{Q}$ and $\mathbf{R}$ continuously change, the DSKF and STL share the same challenge in synchronization capability.

There has been significant research towards robust tracking solutions to solve this problem [30]. However, there are still ample opportunities to find the best technique in terms of performance and complexity [25,31]. Adaptive tracking methods can improve the tracking performance in time-varying scenarios. Different methods to estimate the noise

covariances of the KF have been summarized in a review study [32]. One solution can be to implement a moving average filter to estimate **Q** and **R** and, consequently, adapt the response time of the KF optimally [33]. Moreover, it is possible to implement a carrier-to-noise density ratio (C/N_0)-based DSKF, in which **R** depends on the variance of the STL discriminator output [27]. **Q** can also be adapted according to the dynamic stress error [28]. Recent research implements the loop-bandwidth control algorithm (LBCA)-based DSKF for the PLL [24]. The LBCA performs a loop bandwidth-dependent weighted difference between estimated noise and estimated dynamics of the discriminator output [34]. This algorithm updates the loop bandwidth and, in turn, **Q**, based on the steady-state relationship.

Despite the tracking performance advantage of the KF, its implementation in real-time applications requires a high computational cost compared to the STL. Therefore, efficient low-complexity methods have been studied [23,25]. The complexity of the ESKF can be reduced by taking advantage of the Kalman gain convergence in the steady state [23]. The same can be done for the DSKF, leading to the so-called lookup table (LUT)-DSKF [25]. The implementation of an LBCA-based LUT-DSKF in a PLL tracking scheme has been presented recently [25]. The ratio between the steady-state process variance and the measurement variance provides a one-to-one relationship between the steady-state Kalman gains and the loop bandwidth. Hence, the LBCA can adapt the loop bandwidth to, in turn, adapt the steady-state Kalman gains.

Aiding the FLL in the PLL can significantly improve the robustness against high signal dynamics [35]. Recent research implements an LBCA-based FLL-assisted-PLL (FAP) architecture [36]. This adaptive tracking architecture consists of two independent LBCAs to adapt the bandwidths of a second-order FLL and a third-order PLL. Despite the promising results, extensive tuning was required to find the optimal weighting functions for each LBCA. Furthermore, the second LBCA doubles the complexity.

Figure 1 shows the relation of relevant research on LBCA-based techniques. First, the implementation of the LBCA in tracking schemes with only one measurement has been studied. From the STL [31,34,37–39] to more advance tracking schemes such as the DSKF [24] and the LUT-DSKF [25]. Second, the research has been recently expanded by implementing the LBCA in tracking schemes with two measurements. Recent research implements an adaptive LUT-DSKF in a FAP tracking architecture [26]. The derivation of the discrete algebraic Riccati equation (DARE) of this tracking architecture presents an inter-dependency between FLL and PLL coefficients. Only one LBCA can adapt the LUT-DSKF's response time based on the found inter-dependency. This architecture has been also evaluated under simulated controlled fading scenarios [40]. Furthermore, recent studies show the tracking performance of the LBCA under simulated moon exploration missions [41].

This research expands a conference paper [26]. First, the code phase estimation is included in the DSKF leading to the full DSKF. Second, the DARE derivation of the full DSKF obtains the full LUT-DSKF: a low-complexity tracking structure that uses the steady-state Kalman gains. The derivation shows the same steady-state coefficients that update the frequency Doppler and the carrier phase for the full LUT-DSKF and the LUT-DSKF in the FAP tracking scheme. This study also presents the full LUT-DSKF steady-state coefficients that update the code phase and remarks on the impact of the PLL-assisted-DLL (PAD) on the coefficients. Third, the LBCA is expanded to adapt the full LUT-DSKF. The same LBCA as in the conference paper is used to adapt the response time of the FAP in the full LUT-DSKF. Additionally, this research presents a second LBCA to update the DLL's response time. Fourth, instead of evaluating the tracking performance of a particular satellite vehicle (SV), as presented in the conference paper [26], the carrier and code system performance metrics are selected. These metrics consider all the visible tracked SVs and indicate an overall performance of the tracking architectures under test.

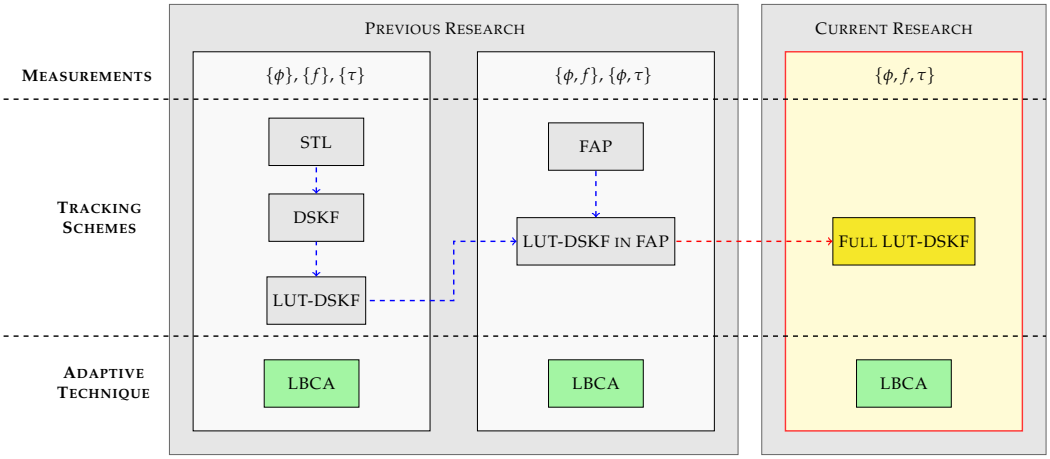


Figure 1. Research survey and comparison to other publications.

This paper shows the adaptive LUT-DSKF, a compact representation of a robust single-frequency adaptive tracking architecture considering all the primary synchronization parameters. This architecture is implemented in the tracking stage of a GOOSE© receiver [13]. The system performance of different adaptive full LUT-DSKF configurations are evaluated under simulated scenarios with different dynamics and noise levels.

The rest of the paper is organized as follows. Section 2 describes the full DSKF. The analysis of this tracking scheme in the analog and discrete domain is performed, presenting the system and measurement model, the state space model (SSM) representation, and the steady-state convergence. Section 3 shows the architecture of the integration of the LBCA in the full LUT-DSKF. Section 4 presents the experimental setup and Section 5 the achieved results. Finally, Section 6 concludes and indicates future work.

2. Full Direct-State Kalman Filter in Tracking Stage

This section describes the full DSKF tracking scheme of a GNSS receiver. First, the full DSKF is analyzed in the analog domain. The system and measurement models, the SSM representation, and the derivation of the continuous domain algebraic Riccati equation (CARE) is shown. Second, the full DSKF in the discrete domain is presented. As in the analog domain, the system and measurement models, the SSM, and the DARE are derived. Finally, the linear model of the steady-state full DSKF, the LUT-DSKF is shown.

2.1. Analog Domain

Assuming a Brownian motion model for the angular acceleration state and the code phase [42], the system model is represented as:

$$\underbrace{\begin{bmatrix} \dot{\tau}(t) \\ \dot{\phi}(t) \\ \dot{f}(t) \\ \dot{a}(t) \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} 0 & 0 & v & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \tau(t) \\ \phi(t) \\ f(t) \\ a(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} w_\tau(t) \\ 0 \\ 0 \\ w_a(t) \end{bmatrix}}_{\mathbf{w}(t)} \tag{1}$$

where t is the time index, $\mathbf{x}$ is the state vector composed of the code phase τ, the carrier phase ϕ, the carrier Doppler f, and the angular acceleration a. The the rate of the state vector $\dot{\mathbf{x}}(t)$ consists of the respective deviates (i.e., rates) $\{\dot{\tau}, \dot{\phi}, \dot{f}, \dot{a}\}$. $\mathbf{A}$ is the state transition matrix, and $\mathbf{w}$ is the process noise vector. $\mathbf{w}$ consists of the zero-mean Gaussian distributed perturbations that suffer the code phase in chips/s and angular acceleration in cycles/s^3, denoted as w_τ and w_a. The parameter v is a scaling factor that determines the aiding of

the carrier Doppler state f into the code phase rate $\dot{\tau}$. This parameter changes if PAD is enabled or disabled. It is defined as:

$$v = \begin{cases} 0 & \text{if PAD disabled} \\ \frac{f_c}{f_r} & \text{if PAD enabled} \end{cases} \tag{2}$$

where f_c is the chipping rate in chips/s and f_r is the carrier frequency of the GNSS signal in Hz.

The process variance of the analog system model $\mathbf{Q}$ is represented as:

$$\mathbf{Q} = \mathrm{E}[\mathbf{w}\mathbf{w}^{\mathrm{T}}] = \begin{bmatrix} q_\tau & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_a \end{bmatrix} \tag{3}$$

where $\mathrm{E}[\cdot]$ is the average operation, and q_τ and q_a are the variances of the random processes w_τ in chips^2/s^2 and w_a in cycles^2/s^6.

The full DSKF has three measurements from the main synchronization parameters: the code phase z_τ, the carrier phase z_ϕ, and the carrier frequency z_f. The relation between measurements and states is:

$$\underbrace{\begin{bmatrix} z_\tau(t) \\ z_\phi(t) \\ z_f(t) \end{bmatrix}}_{\mathbf{z}(t)} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x}(t) + \underbrace{\begin{bmatrix} v_\tau(t) \\ v_\phi(t) \\ v_f(t) \end{bmatrix}}_{\mathbf{v}(t)} \tag{4}$$

where $\mathbf{z}$ is the measurement vector, $\mathbf{H}$ is the measurement matrix, and $\mathbf{v}$ is the measurement noise vector. The random variables (RVs) $\{v_\tau, v_\phi, v_f\}$ represent the zero-mean Gaussian distributed noise of $\{z_\tau, z_\phi, z_f\}$.

The measurement noise covariance matrix $\mathbf{R}$ is:

$$\mathbf{R} = \mathrm{E}\left[\mathbf{z}\mathbf{z}^{\mathrm{T}}\right] = \begin{bmatrix} R_\tau & 0 & 0 \\ 0 & R_\phi & 0 \\ 0 & 0 & R_f \end{bmatrix} \tag{5}$$

where $\{R_\tau, R_\phi, R_f\}$ are the variances of $\{v_\tau, v_\phi, v_f\}$.

The continuous SSM is derived from Equations (1) and (4):

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \underbrace{\begin{bmatrix} \nu_3 & \alpha_3 & \beta_3 \\ \nu_2 & \alpha_2 & \beta_2 \\ \nu_1 & \alpha_1 & \beta_1 \\ \nu_0 & \alpha_0 & \beta_0 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} \delta\tau(t) \\ \delta\phi(t) \\ \delta f(t) \end{bmatrix}}_{\delta\mathbf{z}(t)} \tag{6}$$

$$\underbrace{\begin{bmatrix} \hat{z}_\tau(t) \\ \hat{z}_\phi(t) \\ \hat{z}_f(t) \end{bmatrix}}_{\hat{\mathbf{z}}(t)} = \mathbf{H}\,\mathbf{x}(t) \tag{7}$$

where $\mathbf{K}$ is the coefficient matrix, also known as the Kalman gain matrix. $\hat{\mathbf{z}}$ is the predicted measurement that includes the estimated code phase $\hat{z}_\tau$, the estimated carrier phase $\hat{z}_\phi$ and

the estimated carrier Doppler $\hat{z}_f$. The innovation vector $\delta\mathbf{z}$ is represented as the difference between the measurement $\mathbf{z}$ and the estimated measurement $\hat{\mathbf{z}}$:

$$\delta\mathbf{z}(t) = \mathbf{z}(t) - \hat{\mathbf{z}}(t) \tag{8}$$

The presented SSM in Equations (6) and (7) is equivalent to a Kalman-Bucy filter [43]. In the steady-state region, the error covariance matrix $\mathbf{P}$ converges to a steady-state value, denoted as $\mathbf{P}_{\text{ss}}$ given a constant $\mathbf{Q}$ and $\mathbf{R}$. If the process and measurement variance matrices are known, the trace of $\mathbf{P}_{\text{ss}}$ represents the MMSE. $\mathbf{P}_{\text{ss}}$ is calculated solving the CARE [44,45]:

$$\mathbf{0} = \mathbf{A}\mathbf{P}_{\text{ss}} + \mathbf{P}_{\text{ss}}\mathbf{A}^{\text{T}} + \mathbf{Q} - \mathbf{P}_{\text{ss}}\mathbf{H}^{\text{T}}\mathbf{R}^{-1}\mathbf{H}^{\text{T}}\mathbf{P}_{\text{ss}} \tag{9}$$

The following assumption facilitates the CARE solution [23]:

$$\mathbf{R}_{i,j} \gg \left(\mathbf{H}\,\mathbf{P}_{\text{ss}}\,\mathbf{H}^{\text{T}}\right)_{i,j} \quad \forall i,j = 1,2 \tag{10}$$

$$R_\tau \gg R_f v^2 \gg R_\phi v^2 \tag{11}$$

The approximated positive-definite solution of the CARE gives the steady-state value of the error covariance matrix $\mathbf{P}_{\text{ss}}$.

$$\mathbf{P}_{\text{ss}} = \begin{bmatrix} q_\tau^{1/2} R_\tau^{1/2} & v2q_a^{1/6} R_\phi^{5/6} & v2q_a^{1/3} R_\phi^{2/3} & vq_a^{1/2} R_\phi^{1/2} \\ \text{sym.} & 2q_a^{1/6} R_\phi^{5/6} & 2q_a^{1/3} R_\phi^{2/3} & q_a^{1/2} R_\phi^{1/2} \\ \text{sym.} & \text{sym.} & 3q_a^{1/2} R_\phi^{1/2} & 2q_a^{2/3} R_\phi^{1/3} \\ \text{sym.} & \text{sym.} & \text{sym.} & 2q_a^{5/6} R_\phi^{1/6} \end{bmatrix} \tag{12}$$

where sym. is the abbreviation of symmetrical.

The steady-state Kalman gain $\mathbf{K}_{\text{ss}}$ is derived based on the calculated $\mathbf{P}_{\text{ss}}$ in Equation (12):

$$\mathbf{K}_{\text{ss}} = \mathbf{P}_{\text{ss}}\mathbf{H}^{\text{T}}\mathbf{R}^{-1} = \begin{bmatrix} \nu_{\text{ss}_3} & \alpha_{\text{ss}_3} & \beta_{\text{ss}_3} \\ \nu_{\text{ss}_2} & \alpha_{\text{ss}_2} & \beta_{\text{ss}_2} \\ \nu_{\text{ss}_1} & \alpha_{\text{ss}_1} & \beta_{\text{ss}_1} \\ \nu_{\text{ss}_0} & \alpha_{\text{ss}_0} & \beta_{\text{ss}_0} \end{bmatrix} \tag{13}$$

$$= \begin{bmatrix} \kappa & 2v\gamma & 2v\gamma^2 \frac{R_\phi}{R_f} \\ 2v\gamma \frac{R_\phi}{R_\tau} & 2\gamma & 2\gamma^2 \frac{R_\phi}{R_f} \\ 2v\gamma^2 \frac{R_\phi}{R_\tau} & 2\gamma^2 & 3\gamma^3 \frac{R_\phi}{R_f} \\ v\gamma^3 \frac{R_\phi}{R_\tau} & \gamma^3 & 2\gamma^4 \frac{R_\phi}{R_f} \end{bmatrix} \approx \begin{bmatrix} \kappa & 2v\gamma & 2v\gamma^2 \frac{R_\phi}{R_f} \\ 0 & 2\gamma & 2\gamma^2 \frac{R_\phi}{R_f} \\ 0 & 2\gamma^2 & 3\gamma^3 \frac{R_\phi}{R_f} \\ 0 & \gamma^3 & 2\gamma^4 \frac{R_\phi}{R_f} \end{bmatrix} \tag{14}$$

where γ is the ratio $(q_a/R_\phi)^{1/6}$ and κ is the ratio $(q_\tau/R_\tau)^{1/2}$ in Hertz. The variables γ and κ simplify the natural formulation to improve readability. Considering the assumption in Equation (11), the steady-state coefficients of the DLL $\{\nu_{\text{ss}_0}, \nu_{\text{ss}_1}, \nu_{\text{ss}_2}\}$ are approximated to zero. The steady-state coefficients of the PLL $\{\alpha_{\text{ss}_0}, \alpha_{\text{ss}_1}, \alpha_{\text{ss}_2}, \alpha_{\text{ss}_3}\}$ and the FLL $\{\beta_{\text{ss}_0}, \beta_{\text{ss}_1}, \beta_{\text{ss}_2}, \beta_{\text{ss}_3}\}$ depend on γ, whereas the remainder coefficient of the DLL ν_{ss_3} is dependent on κ. These two parameters $\{\gamma, \kappa\}$ determine the time of response of the full LUT-DSKF. Furthermore, the dependency of $\{\alpha_{\text{ss}_3}, \beta_{\text{ss}_3}\}$ with v indicates that, if PAD is enabled, the carrier phase error $\delta\phi$ and the code phase error $\delta\tau$ will have an influence on the code phase rate estimation $\dot{\tau}$ (see Equation (6)).

2.2. Digital Domain

Based on the backward Euler transform (BET) [46,47], the discrete system model of the full DSKF is represented as:

$$\underbrace{\begin{bmatrix}\tau[n]\\ \phi[n]\\ f[n]\\ a[n]\end{bmatrix}}_{\mathbf{x}[n]} = \underbrace{\begin{bmatrix}1 & 0 & \upsilon\tau_{\text{int}} & \upsilon\tau_{\text{int}}^2\\ 0 & 1 & \tau_{\text{int}} & \tau_{\text{int}}^2\\ 0 & 0 & 1 & \tau_{\text{int}}\\ 0 & 0 & 0 & 1\end{bmatrix}}_{\mathbf{A}_\text{d}} \underbrace{\begin{bmatrix}\tau[n-1]\\ \phi[n-1]\\ f[n-1]\\ a[n-1]\end{bmatrix}}_{\mathbf{x}[n-1]} + \underbrace{\begin{bmatrix}\tau_{\text{int}} & 0 & \upsilon\tau_{\text{int}}^2 & \upsilon\tau_{\text{int}}^3\\ 0 & \tau_{\text{int}} & \tau_{\text{int}}^2 & \tau_{\text{int}}^3\\ 0 & 0 & \tau_{\text{int}} & \tau_{\text{int}}^2\\ 0 & 0 & 0 & \tau_{\text{int}}\end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix}w_\tau[n]\\ 0\\ 0\\ w_a[n]\end{bmatrix}}_{\mathbf{w}[n]} \tag{15}$$

where n is the sample index, $\mathbf{A}_\text{d}$ is the discrete state matrix, τ_{int} is the integration time, and the term $\mathbf{Gw}$ determines the discrete process noise that drives the signal dynamics.

The discrete process covariance matrix $\mathbf{Q}_\text{d}$ is defined as:

$$\mathbf{Q}_\text{d} = \mathbf{G}\,\mathrm{E}[\mathbf{w}\mathbf{w}^\mathrm{T}]\,\mathbf{G}^\mathrm{T} = q_\tau \begin{bmatrix}\tau_{\text{int}}^2 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} + q_a \begin{bmatrix}\upsilon^2\tau_{\text{int}}^6 & \upsilon\tau_{\text{int}}^6 & \upsilon\tau_{\text{int}}^5 & \upsilon\tau_{\text{int}}^4\\ \upsilon\tau_{\text{int}}^6 & \tau_{\text{int}}^6 & \tau_{\text{int}}^5 & \tau_{\text{int}}^4\\ \upsilon\tau_{\text{int}}^5 & \tau_{\text{int}}^5 & \tau_{\text{int}}^4 & \tau_{\text{int}}^3\\ \upsilon\tau_{\text{int}}^4 & \tau_{\text{int}}^4 & \tau_{\text{int}}^3 & \tau_{\text{int}}^2\end{bmatrix} \tag{16}$$

The discrete measurement model is as follows:

$$\underbrace{\begin{bmatrix}z_\tau[n]\\ z_\phi[n]\\ z_f[n]\end{bmatrix}}_{\mathbf{z}[n]} = \underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}}_{\mathbf{H}} \mathbf{A}_\text{d}\mathbf{x}[n-1] + \underbrace{\begin{bmatrix}v_\tau\\ v_\phi\\ v_f\end{bmatrix}}_{\mathbf{v}[n]} \tag{17}$$

The innovation vector $\delta\mathbf{z}$ is represented as in Equation (8):

$$\delta\mathbf{z}[n] = \mathbf{z}[n] - \hat{\mathbf{z}}[n] \tag{18}$$

The measurement covariance matrix $\mathbf{R}$ is the same as in Equation (5).

The system and measurement models in Equations (15) and (17) present four states, $p = 4$, and three measurements, $m = 3$. The open-loop discrete SSM is represented as:

$$\mathbf{x}[n] = \mathbf{A}_\text{d}\mathbf{x}[n-1] + \underbrace{\begin{bmatrix}\nu_3 & \alpha_3 & \beta_3\\ \nu_2 & \alpha_2 & \beta_2\\ \nu_1 & \alpha_1 & \beta_1\\ \nu_0 & \alpha_0 & \beta_0\end{bmatrix}}_{\mathbf{K}_\text{d}} \tau_{\text{int}} \underbrace{\begin{bmatrix}\delta\tau[n]\\ \delta\phi[n]\\ \delta f[n]\end{bmatrix}}_{\delta\mathbf{z}[n]} \tag{19}$$

$$\underbrace{\begin{bmatrix}\hat{z}_\tau[n]\\ \hat{z}_\phi[n]\\ \hat{z}_f[n]\end{bmatrix}}_{\hat{\mathbf{z}}[n]} = \mathbf{H}\,\mathbf{A}_\text{d}\,\mathbf{x}[n-1] \tag{20}$$

To calculate the steady-state Kalman gains in discrete domain $\mathbf{K}_{\text{ss}_\text{d}}$, the solution of the DARE can be derived [44,45]:

$$\mathbf{P}_{\text{ss}} = \mathbf{A}_\text{d}\,\mathbf{P}_{\text{ss}}\,\mathbf{A}_\text{d}^\mathrm{T} + \mathbf{Q}_\text{d} - \mathbf{A}_\text{d}\,\mathbf{P}_{\text{ss}}\,\mathbf{H}^\mathrm{T}(\mathbf{H}\,\mathbf{P}_{\text{ss}}\,\mathbf{H}^\mathrm{T} + \mathbf{R})^{-1}\mathbf{H}\,\mathbf{P}_{\text{ss}}\,\mathbf{A}_\text{d}^\mathrm{T} \tag{21}$$

Different methods such as the Schur decomposition can be used to solve the DARE [48]. This research takes a simplified approach using the analog Kalman gain coefficients $\mathbf{K}_{\text{ss}}$ derived from the CARE (see Equation (14)) in the discrete Kalman gain $\mathbf{K}_{\text{ss}_\text{d}}$:

$$\mathbf{K}_{\text{ss}_\text{d}} \approx \tau_{\text{int}} \mathbf{K}_{\text{ss}} = \tau_{\text{int}} \begin{bmatrix} \kappa & \upsilon 2\gamma & \upsilon 2\gamma^2 \frac{R_\phi}{R_f} \\ 0 & 2\gamma & 2\gamma^2 \frac{R_\phi}{R_f} \\ 0 & 2\gamma^2 & 3\gamma^3 \frac{R_\phi}{R_f} \\ 0 & \gamma^3 & 2\gamma^4 \frac{R_\phi}{R_f} \end{bmatrix} \tag{22}$$

Figure 2 shows the linear model of the discrete full LUT-DSKF tracking architecture, which consists of three main components: the comparator, the loop filters (i.e., for DLL, FLL, and PLL), and the NCO. The comparator is the innovation stage of the DSKF (see Equation (18)), and the rest of the modules perform the state prediction and update of the DSKF (see Equations (19) and (20)). The steady-state Kalman gains are the loop filter coefficients (see Equation (22)).

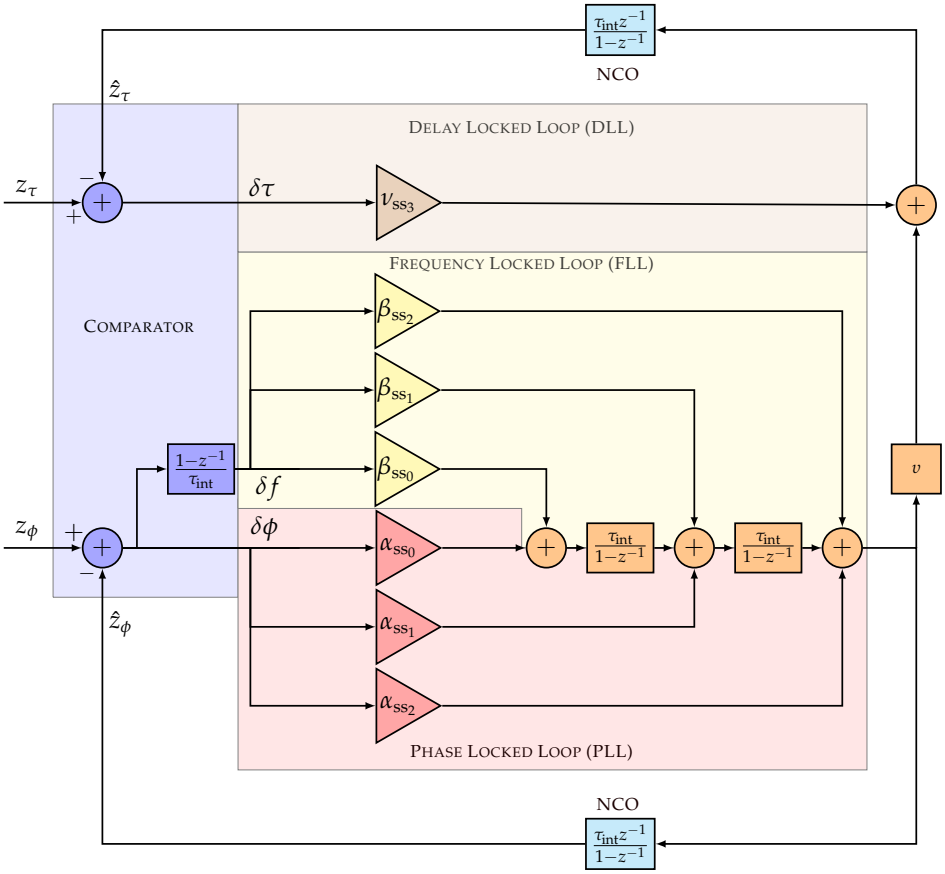


Figure 2. Linear model of full LUT-DSKF tracking architecture.

The open-loop transfer functions of the full LUT-DSKF are derived to show the match between the described system and the presented architecture. The code and carrier phase open-loop transfer functions results by combining the $\mathcal{Z}$-transform of Equations (19) and (20):

$$\hat{\mathbf{z}}(z) = \mathbf{H}\,\mathbf{A}_\text{d} \left(\mathbf{I} - \mathbf{A}_\text{d}\, z^{-1}\right)^{-1} \mathbf{K}_{\text{ss}_\text{d}}\, z^{-1}\, \delta\mathbf{z}(z) \tag{23}$$

Equation (23) is derived as follows:

$$\hat{z}_\tau = \underbrace{\frac{\tau_{\text{int}} z^{-1}}{1 - z^{-1}}}_{\text{NCO}(z)} \Bigg(\underbrace{\nu_{\text{ss}_3}}_{\text{F}_{\text{DLL}}(z)} \delta\tau + v \underbrace{\sum_{l=0}^{2} \alpha_{\text{ss}_{2-l}} \frac{\tau_{\text{int}}^l}{(1 - z^{-1})^l}}_{\text{F}_{\text{PLL}}(z)} \delta\phi + v \underbrace{\sum_{l=0}^{2} \beta_{\text{ss}_{2-l}} \frac{\tau_{\text{int}}^l}{(1 - z^{-1})^l}}_{\text{F}_{\text{FLL}}(z)} \delta f \Bigg) \tag{24}$$

$$\hat{z}_\phi = \text{NCO}(z) \Big(\text{F}_{\text{PLL}}(z) \delta\phi + \text{F}_{\text{FLL}}(z) \delta f \Big) \tag{25}$$

In Equation (24), if PAD is disabled (i.e., $v = 0$), the FAP is uncoupled from the DLL. In this case, only the code phase difference $\delta\tau$ drives the DLL loop filter and the code NCO to output the estimated code phase measurement $\hat{z}_\tau$. On the contrary, if PAD is enabled (i.e., $v = f_c / f_r$), the carrier phase and carrier Doppler difference $\{\delta\phi, \delta f\}$, smoothed by the PLL and FLL loop filter $\{\text{F}_{\text{PLL}}, \text{F}_{\text{PLL}}\}$, aids information to the code phase estimation.

Equation (25) shows the open-loop transfer function for the carrier phase estimation. The smoothed carrier Doppler, derived from the unsmoothed carrier phase and carrier Doppler error $\{\delta\phi, \delta f\}$, drives to the carrier NCO to obtain the estimated carrier phase $\hat{z}_\phi$. For simplicity, the linear model considers in the comparator the following relation between carrier Doppler and carrier phase:

$$\delta f = \frac{1 - z^{-1}}{\tau_{\text{int}}} \delta\phi \tag{26}$$

Two main findings can be addressed from the steady-state coefficients of the full LUT-DSKF. First, there is one single response time parameter γ for the FAP, significantly reducing the complexity of implementing an adaptive tracking technique. Second, the DLL has an additional time of response parameter κ. This parameter must be adapted independently, which implies an increase in complexity. In the following section, an extension of the LBCA adapts the time of response parameters of the full LUT-DSKF, $\{\gamma, \kappa\}$.

3. Adaptive Full Direct-State Kalman Filter

This section describes the architecture of the LBCA-based full LUT-DSKF. The LBCA updates the response time based on a weighted difference between estimated dynamics and noise statistics from the innovation vector [34]. In previous studies, the LBCA has been implemented in the standard STL [31,36], the DSKF [24], and the LUT-DSKF in a PLL [25] and a FAP [26] tracking scheme. Furthermore, this algorithm has been implemented in the interference mitigation stage to adapt the FLL of an adaptive notch filter (NF) [38,39,49].

The LBCA can update any parameter related to the system's time of response. Equation (22) shows that γ and κ are related to the coefficients of $\mathbf{K}_{\text{ss}_\text{d}}$. Since there are two times of the response parameters, two LBCAs are required.

Figure 3 shows the architecture of the LBCA to adapt γ and κ. First, the normalized dynamics of the carrier phase error $\bar{D}_{\delta\phi}$ and the code phase error $\bar{D}_{\delta\tau}$ are calculated:

$$\bar{D}_{\delta\phi}[n] = \frac{|\mu_{\delta\phi}[n]|}{|\mu_{\delta\phi}[n]| + \sigma_{\delta\phi}[n]} \tag{27}$$

$$\bar{D}_{\delta\tau}[n] = \frac{|\mu_{\delta\tau}[n]|}{|\mu_{\delta\tau}[n]| + \sigma_{\delta\tau}[n]} \tag{28}$$

where $\{|\mu_{\delta\phi}|, |\mu_{\delta\tau}|\}$ is the absolute mean and $\{\sigma_{\delta\phi}, \sigma_{\delta\tau}\}$ the standard deviation of the carrier phase and code phase discriminator output, respectively. Second, the difference between the normalized dynamics and weighting function is performed:

$$c_{\mathrm{FAP}}[n] = g_{\mathrm{FAP}_{\mathrm{Max}}} D_{\delta\phi}[n] - g_{\mathrm{FAP}}[n,\, \gamma\tau_{\mathrm{int}}] \tag{29}$$

$$c_{\mathrm{DLL}}[n] = g_{\mathrm{DLL}_{\mathrm{Max}}} D_{\delta\tau}[n] - g_{\mathrm{DLL}}[n,\, \kappa\tau_{\mathrm{int}}] \tag{30}$$

where $\{c_{\mathrm{FAP}}, c_{\mathrm{DLL}}\}$ are the control values use to update the $\{\gamma, \kappa\}$ response times. $\{g_{\mathrm{FAP}}, g_{\mathrm{DLL}}\}$ are the weighting functions that depend on the product between the integration time τ_{int} and the response time parameter $\{\gamma, \kappa\}$. The maximum values of $\{g_{\mathrm{FAP}}, g_{\mathrm{DLL}}\}$ are defined as $\{g_{\mathrm{FAP}_{\mathrm{Max}}}, g_{\mathrm{DLL}_{\mathrm{Max}}}\}$. The control logic module accumulates the control values until there is an update of the response time parameter:

$$c^{\mathrm{acc}}_{\mathrm{FAP}}[n] = \begin{cases} c_{\mathrm{FAP}}[n] + c_{\mathrm{FAP}}[n-1] & \text{if } \gamma[n] = \gamma[n+1] \\ c_{\mathrm{FAP}}[n] & \text{otherwise} \end{cases} \tag{31}$$

$$c^{\mathrm{acc}}_{\mathrm{DLL}}[n] = \begin{cases} c_{\mathrm{DLL}}[n] + c_{\mathrm{DLL}}[n-1] & \text{if } \kappa[n] = \kappa[n+1] \\ c_{\mathrm{DLL}}[n] & \text{otherwise} \end{cases} \tag{32}$$

where $\{c^{\mathrm{acc}}_{\mathrm{FAP}}, c^{\mathrm{acc}}_{\mathrm{DLL}}\}$ are the accumulated control values. Finally, the accumulated control values update the current parameters $\{\gamma[n], \kappa[n]\}$:

$$\hat{\gamma}[n] = \gamma[n] + c^{\mathrm{acc}}_{\mathrm{FAP}}[n] \tag{33}$$

$$\hat{\kappa}[n] = \kappa[n] + c^{\mathrm{acc}}_{\mathrm{DLL}}[n] \tag{34}$$

where $\{\hat{\gamma}, \hat{\kappa}\}$ are the estimated response time parameters. A Schmitt trigger is used to avoid possible noise instabilities from $\{\hat{\gamma}, \hat{\kappa}\}$:

$$\gamma[n+1] = \begin{cases} \frac{6}{5} B_{\mathrm{PLL}_0} & \text{if } n = 0 \\ \hat{\gamma}[n] + \Delta_{\mathrm{FAP}} & \text{if } \hat{\gamma}[n] - \gamma[n] \geq \Delta_{\mathrm{FAP}} \\ \hat{\gamma}[n] - \Delta_{\mathrm{FAP}} & \text{if } \gamma[n] - \hat{\gamma}[n] \leq \Delta_{\mathrm{FAP}} \\ \gamma[n] & \text{otherwise} \end{cases} \tag{35}$$

$$\kappa[n+1] = \begin{cases} 4 B_{\mathrm{DLL}_0} & \text{if } n = 0 \\ \hat{\kappa}[n] + \Delta_{\mathrm{DLL}} & \text{if } \hat{\kappa}[n] - \kappa[n] \geq \Delta_{\mathrm{DLL}} \\ \hat{\kappa}[n] - \Delta_{\mathrm{DLL}} & \text{if } \kappa[n] - \hat{\kappa}[n] \leq \Delta_{\mathrm{DLL}} \\ \kappa[n] & \text{otherwise} \end{cases} \tag{36}$$

where $\{\Delta_{\mathrm{FAP}}, \Delta_{\mathrm{DLL}}\}$ are the update steps set to $\{0.5, 0.01\}$ Hz, and $\{B_{\mathrm{PLL}_0}, B_{\mathrm{DLL}_0}\}$ are the initial loop bandwidths of the PLL and the DLL set to $\{8, 1\}$ Hz.

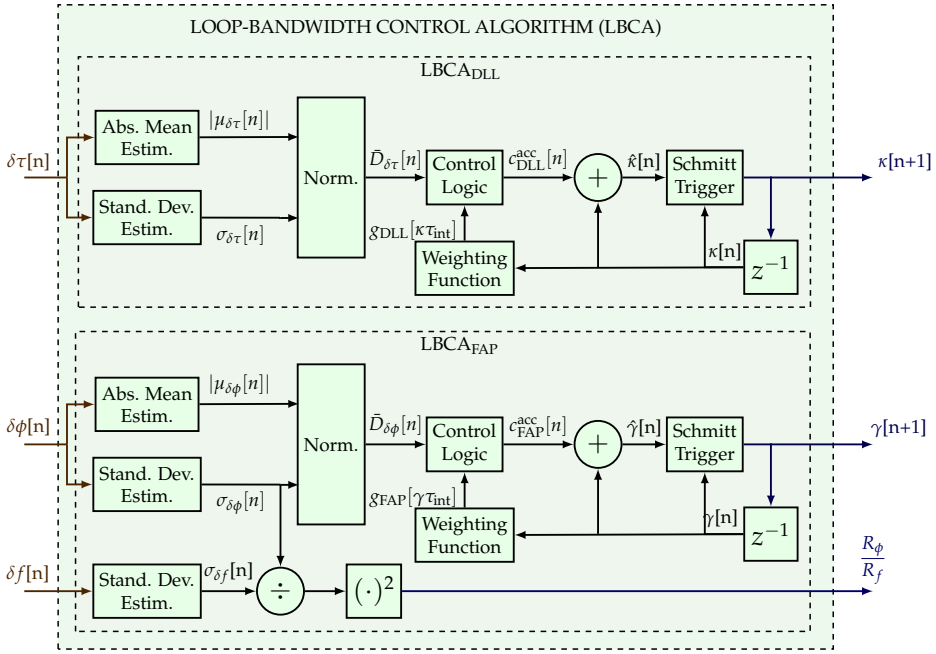


Figure 3. LBCA architecture used in the full LUT-DSKF.

Moreover, the standard deviation estimation of the frequency discriminator output is required to calculate the ratio between R_ϕ and R_f. Due to this operation, the LBCA used in the FAP requires an extra division and multiplication compared to the LBCA implemented in the PLL [25].

Figure 4 presents the selected weighting function for the FAP, g_{FAP} and the DLL, g_{DLL}:

$$\begin{aligned} g_{\text{FAP}}[\gamma\tau_{\text{int}}] &= g_{\text{FAP}_{\text{Max}}} \begin{bmatrix} T_{\text{FAP}} \\ 1 - T_{\text{FAP}} \end{bmatrix}^{\text{T}} \begin{bmatrix} \text{Sig}(50(\gamma\tau_{\text{int}} - 0.06)) \\ \text{Sig}(250(\gamma\tau_{\text{int}} - 0.36)) \end{bmatrix} \\ &= 0.1 \begin{bmatrix} 0.14 \\ 0.86 \end{bmatrix}^{\text{T}} \begin{bmatrix} \text{Sig}(50(\gamma\tau_{\text{int}} - 0.06)) \\ \text{Sig}(250(\gamma\tau_{\text{int}} - 0.36)) \end{bmatrix} \end{aligned} \tag{37}$$

$$\begin{aligned} g_{\text{DLL}}[\kappa\tau_{\text{int}}] &= g_{\text{DLL}_{\text{Max}}} \begin{bmatrix} T_{\text{DLL}} \\ 1 - T_{\text{DLL}} \end{bmatrix}^{\text{T}} \begin{bmatrix} \text{Sig}(200(\kappa\tau_{\text{int}} - 0.002)) \\ \text{Sig}(250(\kappa\tau_{\text{int}} - 0.1)) \end{bmatrix} \\ &= 0.001 \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}^{\text{T}} \begin{bmatrix} \text{Sig}(200(\kappa\tau_{\text{int}} - 0.002)) \\ \text{Sig}(250(\kappa\tau_{\text{int}} - 0.1)) \end{bmatrix} \end{aligned} \tag{38}$$

where $\text{Sig}(\cdot)$ is the Sigmoid function [50], and $\{T_{\text{FAP}}, T_{\text{DLL}}\}$ are the dynamic thresholds of $\{g_{\text{PLL}}, g_{\text{DLL}}\}$. The lower the dynamic threshold, the more sensitive the LBCA is to dynamics. On the contrary, a high dynamic threshold implies a higher sensitivity to noise. To reduce the Sigmoid function complexity, the piecewise linear approximation of nonlinearities (PLAN) technique is used [31,51] to piece-wise interpolate it. The weighting function depends on the normalized bandwidth B_{N}, which represents the product between the loop bandwidth and the integration time $\{\gamma\tau_{\text{int}}, \kappa\tau_{\text{int}}\}$.

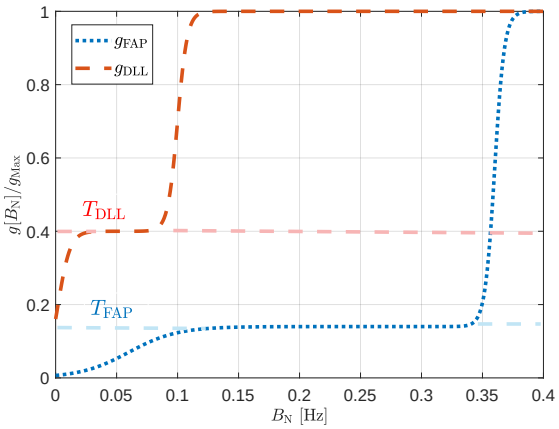


Figure 4. Selected normalized weighting functions for $\mathrm{LBCA}_{\mathrm{FAP}}$ and $\mathrm{LBCA}_{\mathrm{DLL}}$.

Figure 5 presents the architecture of the LBCA-based full LUT-DSKF. Compared to Figure 2, only the $\mathrm{LBCA}_{\mathrm{DLL}}$ and the $\mathrm{LBCA}_{\mathrm{FAP}}$ are added to adapt the steady state coefficients of the full LUT-DSKF (see Equation (22)).

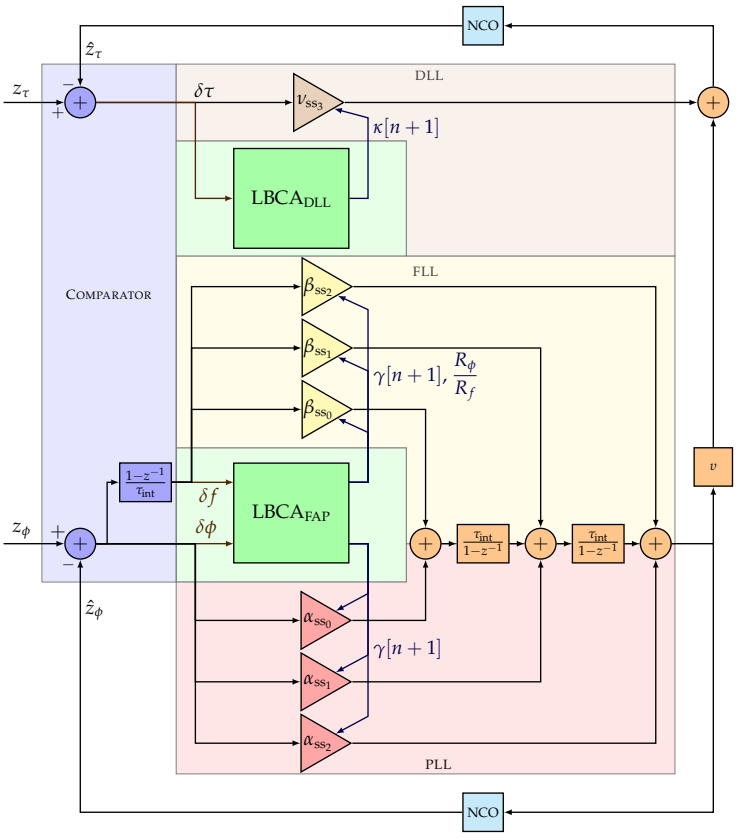


Figure 5. Adaptive full LUT-DSKF using LBCA.

4. Experimental Setup

This section describes the GNSS receiver under test, the metric used to determine the system performance, and the simulated scenarios.

4.1. GNSS Receiver

The GOOSE© platform, developed by Fraunhofer IIS and marketed through TeleOrbit GmbH, is a GNSS receiver with an open software interface [13,52,53]. This receiver contains a customized tri-band radio-frequency front-end (RFFE), a Xilinx Kintex7 field-

programmable gate array (FPGA), and a peripheral component interconnect express (PCIe) interface to connect to an external processor. Figure 6 shows the GOOSE single board computer (SBC) receiver. The RFFE amplifies, filters, downconverts, discretizes the GNSS signals, and sends the digital samples to the FPGA. The analog-to-digital converter (ADC) discretizes each frequency band at a sample rate of 81 MHz and a resolution of 8 bits for the in-phase and quadrature-phase (IQ) components. The FPGA includes one acquisition module and sixty single point correlator (SPC) tracking channels, which the processor can control. The Kintex7 FPGA of the GOOSE SBC is connected to a dual-core ARM processor with an Ubuntu 16.04 operating system and 1GB RAM. The processor performs the acquisition of the incoming digital samples using the acquisition module of the FPGA. The tracking starts once the acquisition achieves a rough estimate of the frequency Doppler f and code phase τ. The tracking stage of this GNSS receiver is partially implemented in the FPGA (e.g., correlators and NCO) and software (e.g., discriminators and loop filters). This stage consists of three steps. First, the FLL and the DLL refine the acquired f and τ estimates. Second, the PLL starts and synchronizes with the carrier phase. Finally, the FLL stops and the PLL can work unaided when the latter successfully achieves a good lock with the carrier phase. The receiver synchronizes with the navigation data at this stage, and the integration time increases to the symbol period. In the case of Global Positioning System (GPS) L1 C/A, the integration time is increased to 20 ms. Table 1 presents the configuration of the tracking scheme during the fine carrier phase synchronization. Once the navigation data is decoded, the PVT solution is computed based on the pseudo range measurements.

Figure 6. Photo of the GOOSE SBC receiver @Fraunhofer IIS/Paul Pulkert.

Table 1. Configuration of tracking stage in GOOSE receiver.

Configuration Parameter	FLL	PLL	DLL
Discriminator type	Atan2(·)	Atan(·)	Dot product
Initial bandwidth [Hz]	0	8	1
Chip spacing, Δ_s [chips]		0.5	
Integration time, τ_{int} [ms]		20	
GNSS signal		GPS L1 C/A	

The LBCA-based full LUT-DSKF is implemented in the tracking stage of the GOOSE receiver in software. In order to evaluate correctly the performance of this tracking architecture, the reacquisition is disabled.

4.2. System Performance Metric

Two metrics are selected to evaluate the system performance of the proposed adaptive tracking architecture. The first metric evaluates the system performance in terms of the carrier phase. The carrier system performance P_ϕ is the same as in previous studies [25]:

$$P_\phi = \overline{PLI} \times \overline{N}_{sat} \tag{39}$$

where $\overline{PLI}$ denotes the phase-lock indicator (PLI) average between the satellites that remain on track, and $\overline{N}_{sat}$ indicates the normalized minimum number of tracked visible satellites during the entire simulation.

The expression of $\overline{\text{PLI}}$ is:

$$\overline{\text{PLI}} = \frac{1}{K_{\text{sim}}\, N_{\text{sat}}^{\min}} \sum_{k=k_0}^{k_0+K_{\text{sim}}} \sum_{l=1}^{N_{\text{sat}}^{\min}} \text{PLI}^l[k] \tag{40}$$

where K_{sim} is the number of measurement epochs under evaluation, and k_0 is the starting time in samples. $N_{\text{sat}}^{\min}$ is the minimum number of visible satellites that remain on track during the simulation time under evaluation:

$$N_{\text{sat}}^{\min} = \min(N_{\text{sat}}[k]) \qquad \forall k = k_0, \cdots, k_0 + K_{\text{sim}} \tag{41}$$

The $\text{PLI}^l[k]$ of the lth SV being tracked is calculated based on the prompt IQ components $\{I_{\text{p}}^l, Q_{\text{p}}^l\}$ [25]:

$$\text{PLI}^l[k] = \frac{(I_{\text{p}}^l[k])^2 - (Q_{\text{p}}^l[k])^2}{(I_{\text{p}}^l[k])^2 + (Q_{\text{p}}^l[k])^2} \tag{42}$$

The second term of Equation (39), $\overline{\text{N}}_{\text{sat}}$, is defined as:

$$\overline{\text{N}}_{\text{sat}} = \frac{N_{\text{sat}}^{\min}}{N_{\text{sat}}^{\text{total}}} \tag{43}$$

where $N_{\text{sat}}^{\text{total}}$ is the total number of visible SVs during the simulation.

The second metric evaluates the system performance in terms of the code phase. The GOOSE SBC has been configured only to compute the PVT based on the pseudo ranges derived from the code phase estimates. The horizontal root mean square error (HRMSE) gives a good indicator of the code system performance. To avoid infinite values of the HRMSE in case there is no fix of the PVT solution during the simulation time under evaluation, the inverse of the HRMSE is considered. Then, the code system performance P_τ is defined as:

$$\text{P}_\tau = \frac{1\text{m}}{\text{HRMSE}} = \text{E}\left[\sqrt{\left(r^{\text{N}} - r_{\text{G}}^{\text{N}}\right)^2 + \left(r^{\text{E}} - r_{\text{G}}^{\text{E}}\right)^2}\right]^{-1} \tag{44}$$

where $\text{E}[\cdot]$ is defined as the mean operation, r^{N} and r^{E} are the calculated north and east user position from the GOOSE platform, and r_{G}^{N} and r_{G}^{E} are the north and east ground truth user position.

P_ϕ and P_τ are both unitless. A high value of P_ϕ and P_τ indicates a good system performance. The opposite case means an increased probability of losing the PVT fix. A final metric is achieved combining (39) and (44):

$$\text{P}_{\text{system}} = \text{P}_\phi \cdot \text{P}_\tau \tag{45}$$

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

$$\overline{\text{P}}_{\text{system}} = \sum_{k=1}^{N_{\text{CN0}}} \frac{\text{P}_{\text{system}}^k}{N_{\text{CN0}}} \tag{46}$$

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

4.3. Evaluation Setup

Figure 7 shows a block diagram of the evaluation setup. It is the same as in previous studies [25,31,34,37]. The setup consists of three main parts: the Spirent GSS9000 radio-frequency constellation simulator (RFCS), the GOOSE SBC, and the user computer. The

user computer manages the simulator and the GOOSE SBC through transmission control protocol (TCP) to perform the test automation. First, the user computer configures RFCS. It selects the desired scenario and sets the C/N_0 to 50 dBHz to all the GNSS signals. A high C/N_0 level is selected to ensure that all the visible SVs achieve tracking at the beginning of the scenario. Second, once the RFCS is ready and starts the simulation ($T_{sim} = 0$ s), the user sends the application to the GOOSE SBC and commands the GOOSE to run it. Third, the user reduces the C/N_0 level by 4 dB until reaching the desired C/N_0 level. Finally, after 20 min of simulation ($T_{sim} = 1200$ s), the user stops the application that is running in GOOSE, collects all the logging data, and stops the simulation at the RFCS. The process repeats until evaluating all the desired C/N_0 levels for all the applications, and for all the selected scenarios.

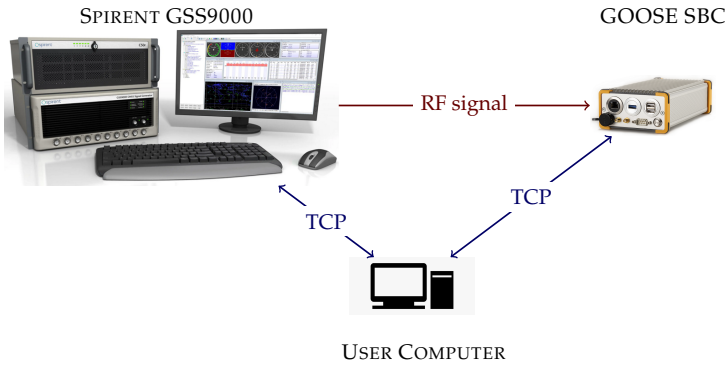


Figure 7. Evaluation setup consisting of a Spirent RFCS, a GOOSE SBC receiver, and a control computer.

The selected C/N_0 levels are {26, 30, 34, 38, 42, 46, 50} dBHz. The first 10 min of the simulations,$T_{sim} = \{0 - 600\}$ s, are used to reach the desired C/N_0 level and, in case of the adaptive tracking, to reach also to stability in the response time. The last 10 min of the simulation, $T_{sim} = \{600 - 1200\}$ s, are under evaluation. Considering that the sampling period of the logged measurements is equal to the integration time τ_{int}, 20 ms, the starting evaluation time k_0 and the simulation time under evaluation K_{sim} are:

$$k_0 = \frac{600\,\text{s}}{\tau_{int} \times 10^{-3}} = 30,000\,\text{samples} \tag{47}$$

$$K_{sim} = \frac{1200\,\text{s}}{\tau_{int} \times 10^{-3}} - k_0 = 30,000\,\text{samples} \tag{48}$$

A static scenario and a dynamic scenario are selected to evaluate these applications. In both scenarios, the radiation pattern of the antenna is simulated as isotropic, and it is direct line-of-sight (LOS) with the SVs. In future work, different antenna patterns and more challenging environments will be simulated. Figure 8 shows the skyplot of both scenarios. There are 10 visible satellites during the simulation. However, SV G1 disappears behind the horizon after two minutes of simulation, SV G30 rises above the horizon near the end of the simulation, and SV G23 is discarded due to its low elevation. Therefore, the maximum number of visible satellites N_{sat}^{total} is limited to seven.

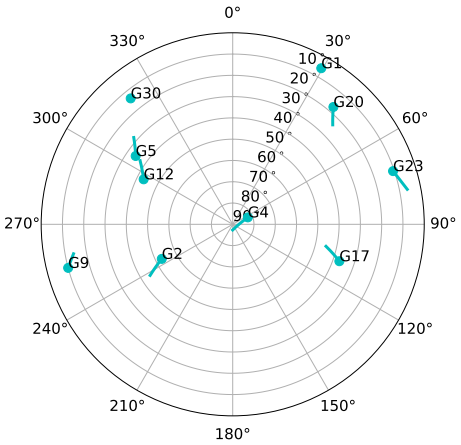
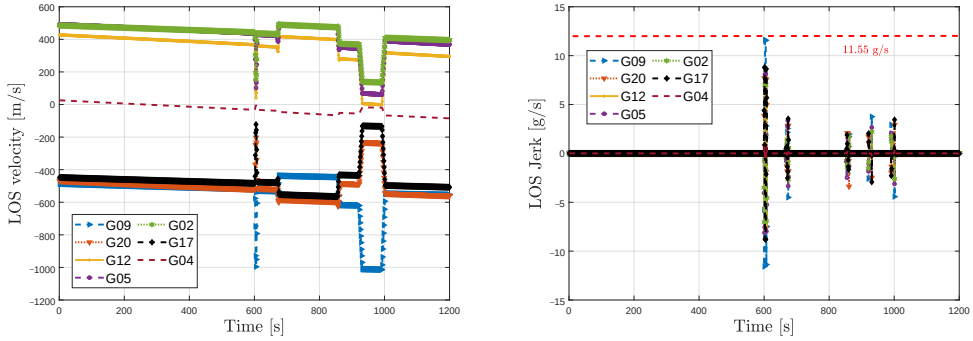


Figure 8. Sky-plot of the simulated scenarios.

Figure 9 presents the LOS dynamics of the simulated dynamic scenario. During the first 10 min of the simulation, the vehicle remains static. During the second half of the simulation, the vehicle moves at high speed generating some jerk dynamics that can challenge the tracking stage. In this scenario, the maximum LOS signal jerk dynamics is 11.55 g/s in SV G9.



(**a**) (**b**)

Figure 9. LOS dynamics in simulated dynamic scenario. (**a**) LOS velocity dynamics. (**b**) LOS jerk dynamics.

From Figure 5, different configurations of this adaptive tracking architecture can be considered. For instance, the LBCA_{FAP} and the LBCA_{PAD} can be bypassed. Furthermore, the FAP can be disabled by setting the FLL coefficients to zero once the PLL achieves lock. Furthermore, the PAD can be enabled or disabled based on Equation (2). Therefore, in this research 10 applications with different configurations are selected to be evaluated. Table 2 shows the different tracking schemes derived from the LBCA-based LUT-DSKF. These applications are evaluated under different dynamics and noise levels. Since there are seven C/N_0 levels and two scenarios, the total amount of time required to evaluate a single application is 280 min.

Table 2. Tracking applications under evaluation. The tracking configurations used for each application are marked with x.

Tracking Scheme	Tracking Configuration:			
	$LBCA_{FAP}$	**$LBCA_{DLL}$**	**FAP**	**PAD**
LBCA FAP + DLL	x		x	
LBCA FAP + PAD	x		x	x
LBCA PLL + DLL	x			x
LBCA PLL + PAD	x			x
LBCA FAP + LBCA DLL	x	x	x	
LBCA FAP + LBCA PAD	x	x	x	x
LBCA PLL + LBCA DLL	x	x		x
LBCA PLL + LBCA PAD	x	x		x
Standard PLL + PAD				x
Standard PLL + DLL				

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 3 presents the theoretical complexity based on the added number of additions, multiplications, and divisions for each tracking configuration. The $LBCA_{FAP}$ includes an additional multiplication and division compared to the $LBCA_{DLL}$ due to the ratio R_ϕ / R_f used to adapt the FLL coefficients (see Equation (22). The aiding the FLL into the PLL (i.e., FAP) adds three additions and three multiplications (see Figure 2). The aiding of the estimated carrier Doppler to the DLL (i.e., PAD) adds only one addition. The division presented in Equation (2) can be precomputed during initialization.

Table 3. Complexity of tracking configurations based on the added number of operations.

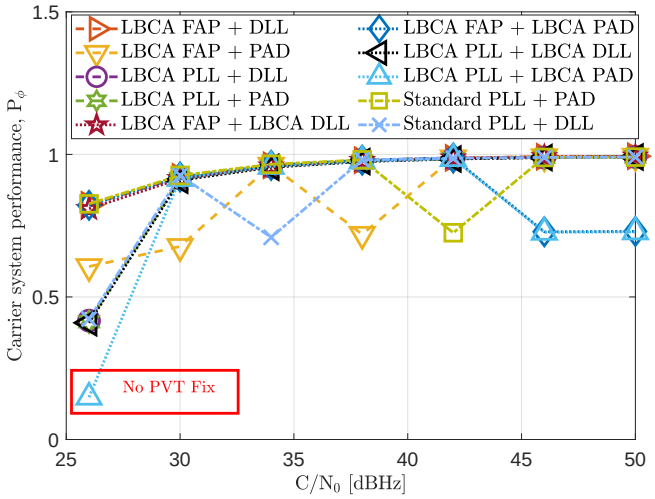
Tracking Configuration	Added Number of Operations:		
	Additions	**Multiplications**	**Divisions**
$LBCA_{FAP}$ [31]	6	8	2
$LBCA_{DLL}$ [31]	6	7	1
FAP	3	3	0
PAD	1	0	0

5. Results

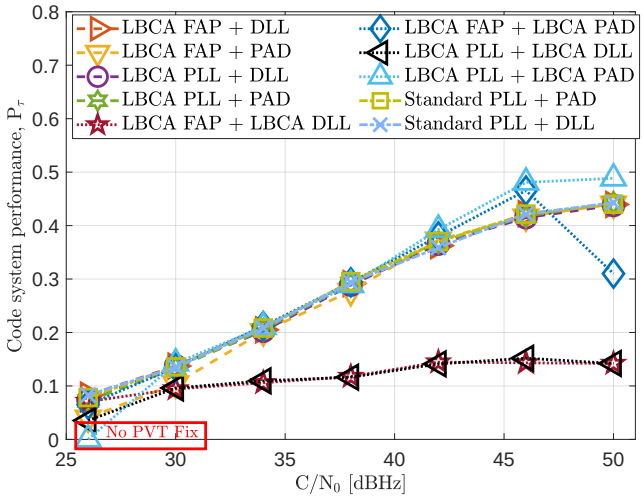
The results are separated into three sections. First, the system performance of the static scenario is evaluated. Second, the system performance of the dynamic scenario is presented. Finally, the average system performance of each adaptive tracking configuration is summarized in a table. The dataset used to plot the presented results is available online to download [54].

5.1. Static Scenario

Figure 10 shows the carrier system performance P_ϕ and the code system performance P_τ in a static scenario under different C/N_0 levels. The selected tracking configurations under evaluation are listed in Table 2. All the tracking configurations present similar carrier and code system performance. The only tracking configuration that loses the PVT solution is the LBCA-based PLL with an unaided LBCA-based DLL. The LBCA's weighting function used for the carrier phase synchronization (see Equation (37)) is configured to be sensitive to dynamics. Therefore, low C/N_0 levels can challenge carrier synchronization with this configuration. However, this is a sporadic error since the other tracking configurations using the same LBCA can maintain a PVT fix.



(**a**)



(**b**)

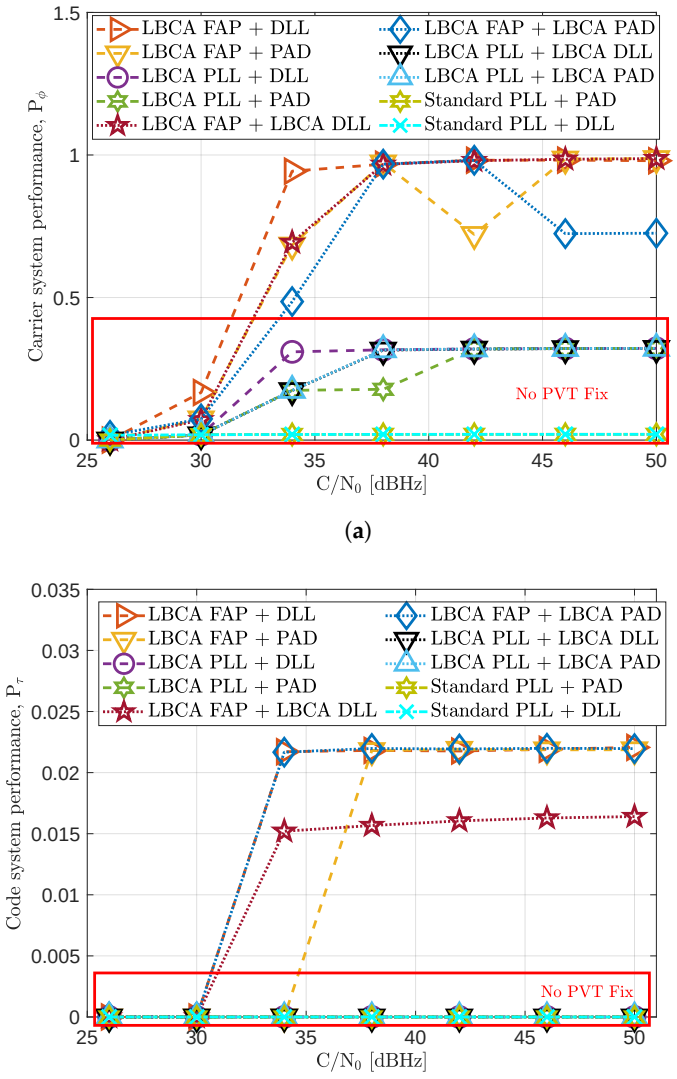
Figure 10. System performance evaluation in static scenario. (**a**) Carrier system performance. (**b**) Code system performance.

An interesting result can be observed in Figure 10b. When no PAD is enabled, the LBCA-based DLL performs poorly compared to the other tracking techniques at any C/N_0 level. The LBCA used for the DLL is highly noise-sensitive, leading to a constant decrease in the DLL bandwidth until reaching a minimum bandwidth of 0.25 Hz. Since there is no carrier aiding to mitigate the dynamics, the DLL suffers a code phase bias error. Although being a static scenario, these dynamics can be generated by the receiver's clock. For more extended simulations, this tracking configuration probably loses the PVT fix.

5.2. Dynamic Scenario

Figure 11 depicts the dynamic system performance of the selected tracking configurations. The standard tracking techniques have no PVT fix at any C/N_0 level. The LBCA-based PLL techniques do not maintain either the PVT fix, but they manage to keep the tracking of at least two to three SVs from 34 dBHz to 50 dBHz. The high P_ϕ score compared to the standard tracking techniques can be observed in Figure 11a. Only the LBCA-based FAP architectures keep a continuous PVT solution from 34 dBHz to 50 dBHz. From the LBCA-based FAP techniques, the one with the standard DLL presents the best carrier system performance. At 30 dBHz, although the PVT fix is lost, this tracking technique

maintains a continuous track of three SVs during the entire simulation. The LBCA-based FAP combined with the LBCA-based DLL presents a continuous PVT from 34 dBHz to 50 dBHz, but Figure 11b shows a degraded P_τ compared to the rest of LBCA-based FAP architectures. The reason behind this is the low bandwidth of the DLL set by theLBCA and the unaided carrier dynamics.



(**a**)

(**b**)

Figure 11. System performance evaluation in dynamic scenario. (**a**) Carrier system performance. (**b**) Code system performance.

5.3. Total System Performance

Table 4 summarizes the final static and dynamic system performance of each tracking configuration under evaluation. The best static and dynamic system performance is marked green, whereas the worse performance is marked red. Moreover, the added time complexity of each adaptive tracking technique is included. The same procedure to calculate the added time complexity as in previous research is carried out [25,31]. The complexity is marked as red, orange, and green, depending on the level of complexity. The colors vary from the most complex one, marked in red, to the simplest one, marked in green.

Table 4. System performance of adaptive tracking techniques.

Tracking Technique	Static $\overline{P}_{System}$	Dynamic $\overline{P}_{System}$	Added Time Complexity
LBCA FAP + DLL	0.0386 *	0.0022	1.94 ‡
LBCA FAP + PAD	0.0349	0.0016	1.94
LBCA PLL + DLL	0.0375	0	1.90
LBCA PLL + PAD	0.0377	0	1.90
LBCA FAP + LBCA DLL	0.0160	0.0015	2.84
LBCA FAP + LBCA PAD	0.0329	0.0017	2.84
LBCA PLL + LBCA DLL	0.0153 †	0	2.81
LBCA PLL + LBCA PAD	0.0348	0	2.81
Standard PLL + PAD	0.0368	0	1.00
Standard PLL + DLL	0.0369	0	1.00

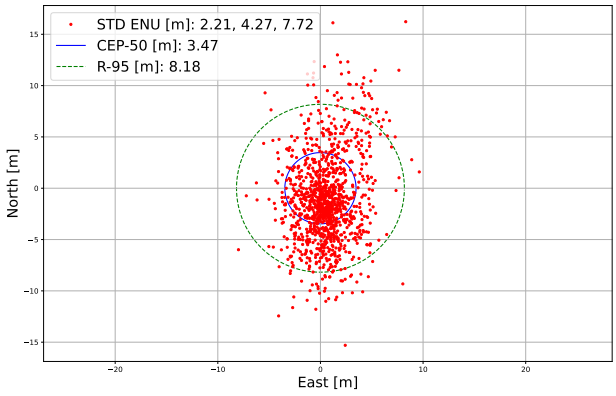
Added time complexity is the factor that the algorithm takes to process compared to a standard tracking architecture. * Values in green indicate best performance or least added complexity. † Values in red indicate worst performance or most added complexity. ‡ Values in orange indicate medium added complexity.

The main outcome of these results is the significant improvement of the system performance in the high dynamic scenario using the LBCA-based FAP. It presents an excellent system performance in dynamic scenarios, keeping a great tracking sensitivity in static scenarios. This tracking configuration is the only one that maintains a continuous PVT solution during the entire high-dynamic scenario. The LBCA-based PLL and the standard techniques are not robust enough to maintain a continuous PVT solution. It is possible to improve the LBCA-based PLL to be more sensitive to dynamics by decreasing the dynamic threshold T_{FAP} of the LBCA weighting function (see Equation (37)). However, that change can degrade the tracking sensitivity in stationary scenarios. It highlights the trade-off for tuning for sensitivity to dynamic scenarios.

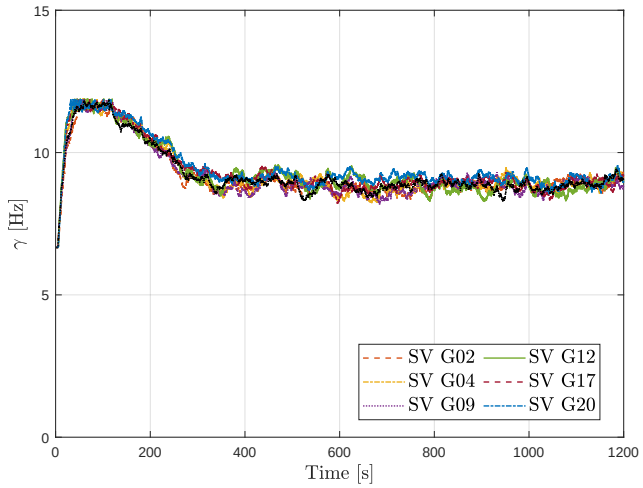
The system performance using the LBCA-based DLL could be improved. The high sensitivity to noise drives the selected time of response parameter κ to a low value. Some further tuning of its weighting function g_{DLL} is required. When carrier aiding is enabled, the LBCA-based FAP aids the dynamics into the DLL, achieving good scores in the system performance for both scenarios. However, among the LBCA-based FAP tracking schemes, the LBCA-based FAP with unaided DLL achieves the best performance. In the static scenario, aiding the carrier Doppler into the code phase estimation can be counterproductive at low C/N_0 levels, since the carrier Doppler estimate becomes noisy. On the contrary, at high C/N_0 levels, carrier aiding is a solution to decrease the DLL bandwidth and improve the code system performance. In the dynamic scenario, the carrier aiding was expected to outperform the other configurations. However, a slight decrease in performance is observed. Further testing involves fine-tuning the weighting function for the LBCA-based DLL.

A separate LBCA for the DLL increases the complexity. Further investigations will be conducted on reducing the adaptive full LUT-DSKF using one single LBCA. In addition, a negligible increase in complexity is observed while enabling PAD. The use of FAP also shows a minor increase.

The configuration with the best system performance is the LBCA-based FAP with unaided DLL, being 1.9 times more complex than the standard tracking, the second least complex from the presented techniques. Figures 12b and 13 present the position estimation and the variation of the FAP's response time parameter γ in the simulated static and dynamic scenario at 34 dBHz. Further analysis of the other configurations is available on the cloud [54]. Figure 12a shows the estimated position in the static scenario using this adaptive tracking configuration at 34 dBHz. Furthermore, the response time parameter γ is depicted in Figure 12b. Initially, the C/N_0 level is 50 dBHz, and each LBCA increases γ to 13 Hz. Every 30 s, the C/N_0 decreases until reaching 34 dBHz. The LBCA reduces the response time parameter accordingly. At 34 dBHz, the LBCA maintains a γ value of 8 Hz.



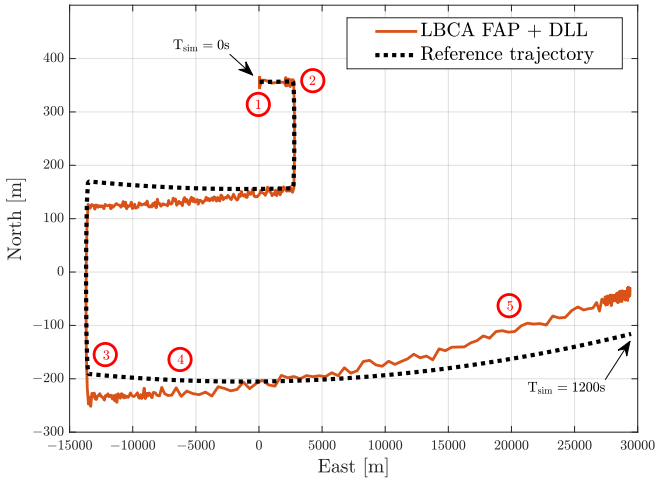
(**a**)



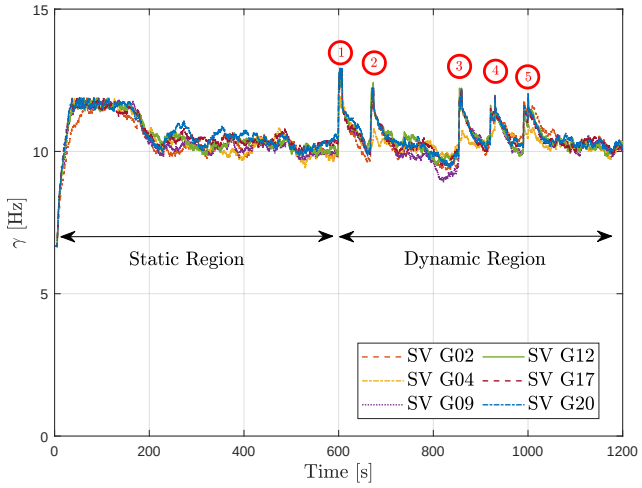
(**b**)

Figure 12. Results of LBCA-based FAP with unaided DLL at 34 dBHz in the static scenario. (**a**) Position point-cloud. (**b**) Loop-bandwidth variation of FAP architecture.

In the dynamic scenario, the comparison between the estimated trajectory based on this adaptive tracking configuration and the reference trajectory is shown in Figure 13a. The circles depicted in red indicate the high dynamic events with significant LOS jerk dynamics. Figure 13b presents the γ adaptation done by the LBCA. During the static region, each tracking channel's LBCA lstabilizes the bandwidth to 8 Hz. Once dynamic events are present, marked in red circles, the LBCA increases the bandwidth to maintain the carrier phase lock. The bandwidth peaks are closely related to the LOS jerk dynamics of the simulated scenario presented in Figure 9b. After the dynamic events, the LBCA returns to a low bandwidth to maintain a good carrier phase synchronization at 34 dBHz.



(**a**)



(**b**)

Figure 13. Results of LBCA-based FAP with unaided DLL at 34 dBHz in a dynamic scenario. (**a**) Comparison between reference trajectory and estimated trajectory from GOOSE. (**b**) Loop-bandwidth variation of FAP architecture.

These results confirm the robustness of the presented adaptive full LUT-DSKF to maintain carrier-phase continuity under different noise and signal dynamic levels while keeping low time complexity.

6. Conclusions

This paper presents a complete single-frequency adaptive scalar tracking architecture: the LBCA-based full LUT-DSKF. First, the full DSKF is analyzed by explaining the system and measurement model, the state space model, and the transfer function. Second, to reduce the complexity of the full DSKF, the convergence of the Kalman gains is calculated by solving the CARE, deriving the so-called full LUT-DSKF. Previous research shows that the LUT-DSKF reduces the time complexity by more than half compared to the DSKF [25]. This simplification shows a relationship between Kalman gains based on the ratio parameter γ and κ (see Equation (22)). Third, the response time of the full LUT-DSKF is adapted through γ and κ using two LBCAs. Fourth, the carrier and code system performance of different configurations of the LBCA-based full LUT-DSKF are compared in a static and a dynamic scenario under different C/N_0 levels. A metric to evaluate the code system

performance is presented based on the HRMSE. The product of the code and carrier system performance leads to a final metric in which the tracking scheme can be nicely evaluated. The results show the importance of the LBCA-based FAP in high dynamic scenarios. This tracking configuration is the only one that maintains the PVT solution, requiring a slightly increased complexity compared to the LBCA-based PLL.

The inter-dependency between FLL and PLL coefficients in a FAP architecture benefits the implementation of a low-complexity adaptive technique using a single LBCA. Another important observation is the fact that it is not necessary to set the third coefficient of the FLL, β_2, to zero, as it is usually done. While deriving the DARE, one can observe that β_2 equals zero is not the optimal configuration. No relationship between the DLL and the FAP coefficients has been found. Therefore, another LBCA is required to adapt the DLL coefficients.

Future work follows-up recent research [36] testing the presented adaptive tracking architecture in simulated rocket scenarios. Next studies consist of analyzing the effect of the sounding rocket's attitude and its antenna's radiation pattern in the adaptive full LUT-DSKF. Moreover, an extension of the presented tracking architecture is proposed: an LBCA-based multi-frequency adaptive tracking architecture. Multi-frequency tracking architectures imply a selective frequency diversity that can benefit tracking sensitivity [55]. The addition of the LBCA in this tracking architecture can optimize the tracking performance by weighting the filter coefficients depending on the estimated dynamics and noise of each band. Furthermore, as multipath affects each frequency band differently, multi-frequency adaptive tracking techniques can suppress the bands affected by multipath while allowing only the non-affected ones. The use of the LBCA in this multi-frequency tracking scheme cannot only improve the tracking performance in dynamic and noisy scenarios, but it can also increase the robustness against interferences and multipath effects.

Like the loop bandwidth, the integration time also affects the tracking response time. A method to adapt the integration time based on the LBCA's loop-bandwidth update has been presented [41]. Future research will introduce the normalized-bandwidth control algorithm (NBCA): an extension of the LBCA that adapts the loop bandwidth and the integration time simultaneously. An improvement in tracking sensitivity is expected, particularly in pilot signals.

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.; data curation, I.C.; writing—original draft preparation, I.C.; writing—review and editing, I.C., J.R.v.d.M., E.S.L. and J.N.; visualization, I.C.; supervision, J.N. and E.S.L.; project administration, W.F.; funding acquisition, W.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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 (accessed on 18 March 2023).

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
CARE	continuous domain algebraic Riccati equation
C/N_0	carrier-to-noise density ratio
DARE	discrete algebraic Riccati equation
DLL	delay locked loop
DSKF	direct-state Kalman filter
ESKF	error-state Kalman-filter
FAP	FLL-assisted-PLL
FLL	frequency locked loop
FPGA	field-programmable gate array
GNSS	global navigation satellite system
GPS	Global Positioning System
HRMSE	horizontal root mean square error
IIR	infinite impulse response
IQ	in-phase and quadrature-phase
KF	Kalman filter
LBCA	loop-bandwidth control algorithm
LOS	line-of-sight
LUT	lookup table
MMSE	minimum mean square error
NBCA	normalized-bandwidth control algorithm
NCO	numerically controlled oscillator
NF	notch filter
PAD	PLL-assisted-DLL
PCIe	peripheral component interconnect express
PLAN	piecewise linear approximation of nonlinearities
PLI	phase-lock indicator
PLL	phase locked loop
PPP	precise point positioning
PVT	position, velocity, and time
RFCS	radio-frequency constellation simulator
RFFE	radio-frequency front-end
RTK	real-time kinematic
RV	random variable
SBC	single board computer
SPC	single point correlator
SSM	state space model
STL	scalar tracking loop
SV	satellite vehicle
SWAP	size, weight, and power
TCP	transmission control protocol

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. Van Dierendonck, A.J. GPS Receivers. In *Global Positioning System: Theory and Applications*; American Institute of Aeronautics and Astronautics, AJ Systems: Los Altos, CA, USA, 1996; Volume 1. [CrossRef]
3. Won, J.; Pany, T. Signal Processing. In *Springer Handbook of Global Navigation Satellite Systems*; Springer International Publishing: Cham, Switzerland, 2017; pp. 401–442. [CrossRef]
4. 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]
5. Gardner, F.M. *Phaselock Techniques*, 3rd ed.; Wiley: New York, NY, USA, 2005.
6. Dabove, P.; Di Pietra, V. Single-baseline RTK positioning using dual-frequency GNSS receivers inside smartphones. *Sensors* **2019**, *19*, 4302. [CrossRef]
7. Aggrey, J.; Bisnath, S.; Naciri, N.; Shinghal, G.; Yang, S. Multi-GNSS precise point positioning with next-generation smartphone measurements. *J. Spat. Sci.* **2020**, *65*, 79–98. [CrossRef]
8. Wu, Q.; Sun, M.; Zhou, C.; Zhang, P. Precise point positioning using dual-frequency GNSS observations on smartphone. *Sensors* **2019**, *19*, 2189. [CrossRef]
9. Zangenehnejad, F.; Gao, Y. GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives. *Satell. Navig.* **2021**, *2*, 24. [CrossRef]
10. Xie, P.; Petovello, M.G. Measuring GNSS multipath distributions in urban canyon environments. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 366–377. [CrossRef]
11. Linty, N.; Lo Presti, L.; Dovis, F.; Crosta, P. Performance analysis of duty-cycle power saving techniques in GNSS mass-market receivers. In Proceedings of the 2014 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 5–8 May, 2014; pp. 1096–1104. [CrossRef]
12. Morales Ferre, R.; Seco-Granados, G.; Lohan, E.S. Energy-efficiency considerations for GNSS signal acquisition. *Inside GNSS* **2021**, *16*, 32–39. Available online: https://insidegnss.com/energy-efficiency-considerations-for-gnss-signal-acquisition/ (accessed on 18 March 2023).
13. 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.
14. Driessen, P.F. DPLL bit synchronizer with rapid acquisition using adaptive Kalman filtering techniques. *IEEE Trans. Commun.* **1994**, *42*, 2673–2675. [CrossRef]
15. Gelb, A. The Analytic Sciences Corporation. In *Applied Optimal Estimation*; The MIT Press: Cambridge, MA, USA, 1974.
16. Thacker, N.A.; Lacey, A.J. *Tutorial: The Likelihood Interpretation of the Kalman Filter*; Tina Memo; University of Manchester: Manchester, UK, 2006.
17. Vilá-Valls, J.; Closas, P.; Navarro, M.; Fernández-Prades, C. Are PLLs dead? A tutorial on Kalman filter-based techniques for digital carrier synchronization. *IEEE Aerosp. Electron. Syst.* **2017**, *32*, 28–45. [CrossRef]
18. Won, J.H.; Dötterböck, D.; Eissfeller, B. Performance comparison of different forms of Kalman filter approaches for a vector-based GNSS signal tracking loop. *Navigation* **2010**, *57*, 185–199. [CrossRef]
19. Won, J.H.; Pany, T.; Eissfeller, B. Characteristics of Kalman filters for GNSS signal tracking loop. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 3671–3681. [CrossRef]
20. O'Driscoll, C.; Lachapelle, G. Comparison of traditional and Kalman filter based tracking architectures. In Proceedings of the 2009 European Navigation Conference (ENC), Warsaw, Poland, 3–6 May 2009.
21. O'Driscoll, C.; Petovello, M.; Lachapelle, G. Choosing the coherent integration time for Kalman filter based carrier phase tracking of GNSS signals. *GPS Solut.* **2011**, *15*, 345–356. [CrossRef]
22. Tang, X.; Falco, G.; Falletti, E.; Lo Presti, L. Theoretical analysis and tuning criteria of the Kalman filter-based tracking loop. *GPS Solut.* **2014**, *19*, 489–503. [CrossRef]
23. Tang, X.; Falco, G.; Falletti, E.; Lo Presti, L. Complexity reduction of the Kalman filter-based tracking loops in GNSS receivers. *GPS Solut.* **2017**, *21*, 685–699. [CrossRef]
24. Cortés, I.; Marín, P.; van der Merwe, J.R.; Simona Lohan, E.; Nurmi, J.; Felber, W. Adaptive techniques in scalar tracking loops with direct-state kalman-filter. In Proceedings of the 2021 International Conference on Localization and GNSS (ICL-GNSS), Tampere, Finland, 1–3 June 2021; pp. 1–7. [CrossRef]
25. Cortés, I.; van der Merwe, J.R.; Lohan, E.S.; Nurmi, J.; Felber, W. Performance evaluation of adaptive tracking techniques with direct-State Kalman filter. *Sensors* **2022**, *22*, 420. [CrossRef]
26. Cortés, I.; Conde, N.; van der Merwe, J.R.; Lohan, E.S.; Nurmi, J.; Felber, W. Low-complexity adaptive direct-state Kalman filter for robust GNSS carrier tracking. In Proceedings of the 2022 International Conference on Localization and GNSS (ICL-GNSS), Tampere, Finland, 7–9 June 2022; pp. 1–7. [CrossRef]
27. 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]

28. Won, J.H. A novel adaptive digital phase-lock-loop for modern digital GNSS receivers. *IEEE Commun. Lett.* **2014**, *18*, 46–49. [CrossRef]
29. Vilà-Valls, J.; Closas, P.; Fernández-Prades, C. On the identifiability of noise statistics and adaptive KF design for robust GNSS carrier tracking. In Proceedings of the 2015 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2015; pp. 1–10. [CrossRef]
30. 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]
31. 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. [CrossRef]
32. Duník, J.; Straka, O.; Kost, O.; Havlík, J. Noise covariance matrices in state-space models: A survey and comparison of estimation methods—Part I. *Int. J. Adapt. Control Signal Process.* **2017**, *31*, 1505–1543. [CrossRef]
33. Bolla, P. Advanced Tracking Loop Architectures for Multi-Frequency GNSS Receiver. Ph.D. Thesis, Tampere University of Technology, Tampere, Finland; Samara University, Samara, Russia, 2018. Available online: http://urn.fi/URN:ISBN:978-952-15-4309-8 (accessed on 18 March 2023).
34. 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 Proceedings of IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, OR, USA, 20–23 April 2020; pp. 1178–1188. [CrossRef]
35. Hurd, W.; Statman, J.; Vilnrotter, V. High dynamic GPS receiver using maximum likelihood estimation and frequency tracking. *IEEE Trans. Aerosp. Electron. Syst.* **1987**, *AES-23*, 425–437. [CrossRef]
36. Cortés, I.; Urquijo, S.; Overbeck, M.; Felber, W.; Agrotis, L.; Mayer, V.; Schönemann, E.; Enderle, W. Robust tracking strategy for modern GNSS receivers in sounding rockets. In Proceedings of the ESA Workshop on Satellite Navigation User Equipment Technologies (NAVITEC), Noordwijk, The Netherlands, 5–7 April 2022; pp. 1–7. [CrossRef]
37. 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]
38. van der Merwe, J.R.; Cortés, I.; Garzia, F.; Lohan, E.S.; Nurmi, J.; Felber, W. Resilient interference mitigation with adaptive frequency locked loop based adaptive notch filtering. In Proceedings of the 2021 Navigation, Edinburgh, UK, 15–18 November 2021; pp. 1–18.
39. van der Merwe, J.R.; Cortés, I.; Garzia, F.; Rügamer, A.; Felber, W. Exotic FMCW waveform mitigation with an advanced multi-parameter adaptive notch filter (MPANF). In Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, CO, USA, 19–23 September 2022; pp. 3783–3819. [CrossRef]
40. Conde, N.; Cortés, I.; van der Merwe, J.R.; Rügamer, A.; Felber, W. Analysis of multipath effect in the tracking stage using loop bandwidth control algorithm. In Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, CO, USA, 19–23 September 2022; pp. 1236–1256. [CrossRef]
41. Song, Y.J.; Won, J.H. Table-Based adaptive digital phase-locked loop for GNSS receivers operating in moon exploration missions. *Sensors* **2022**, *22*, 10001. [CrossRef]
42. Lasota, A.; Michael, C.M. *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*, 2nd ed.; Applied Mathematical Sciences; Springer: New York, NY, USA, 1994. [CrossRef]
43. Bucy, R.S.; Joseph, P.D. *Filtering for Stochastic Processes with Applications to Guidance*; Interscience Publishers: New York, NY, USA, 1968.
44. Einicke, G. *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*; InTechOpen: London, UK, 2012. [CrossRef]
45. Brown, R.G.; Hwang, P.Y.C. *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises and Solutions*, 3rd ed.; Wiley: New York, NY, 1997.
46. Aguirre, S.; Hurd, W.; Kumar, R.; Statman, J. A comparison of methods for DPLL loop filter design. In *Telecommunications and Data Acquisition Progress Report 42-79*; Jet Propulsion Laboratory: Pasadena, CA, USA, 1986.
47. Jury, E.I. *Theory and Application of the Z-Transform Method*; Wiley: New York, NY, USA, 1964.
48. Laub, A. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. Contr.* **1979**, *24*, 913–921. [CrossRef]
49. van der Merwe, J.R.; Cortés, I.; Garzia, F.; Rügamer, A.; Felber, W. Multi-parameter adaptive notch filter (MPANF) for enhanced interference mitigation. *J. Navig.* 2023, *70* 2. [CrossRef]
50. 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.
51. 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]
52. Seybold, J. *GOOSE: Open GNSS Receiver Platform*; Technical Report; TeleOrbit GmbH: Nuremberg, Germany, 2020. Available online: https://teleorbit.eu/en/satnav/ (accessed on 18 March 2023).
53. Welcome to the Open Gnss RecEiver (OGRE) Wiki! Available online: https://github.com/Fraunhofer-IIS/ogre/wiki/ (accessed on 18 March 2023).

54. Robust Tracking Techniques Dataset using GOOSE Receiver. Available online: https://owncloud.fraunhofer.de/index.php/s/LGoWPVtV5xbQ9mB (accessed on 18 March 2023).
55. Yang, R.; Xu, D.; Morton, Y.T. Generalized multifrequency GPS carrier tracking architecture: Design and performance analysis. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 2548–2563. [CrossRef]