Adaptive Loop-Bandwidth Control Algorithm for Scalar Tracking Loops

Iñigo Cortés, J. Rossouw van der Merwe, Alexander Rügamer and Wolfgang Felber
Satellite Based Positioning Systems Department
Fraunhofer IIS
Nuremberg, Germany
inigo.cortes@iis.fraunhofer.de

***Abstract*— This paper presents a loop-bandwidth control algorithm for adaptive scalar tracking loops used in modern digital global navigation satellite system (GNSS) receivers. This algorithm modifies the noise bandwidth of the loop filter. The updated loop-bandwidth balances the signal dynamics and noise through a weighting function. The agility of the estimators defines the sensitivity of the algorithm against dynamics. This algorithm is applicable to the delay-, frequency- or phase-locked-loop (DLL, FLL, PLL) and to any order loop-filter, making it simpler to incorporate than other methods. The algorithm is first analyzed and evaluated in a software receiver. Second, it is implemented in an open software interface GNSS hardware receiver for testing in simulated scenarios with real-world conditions. The scenarios represent different dynamics and noise cases. The results show the algorithm's generic usability and advantage over fixed loop settings, while preserving minimum tracking jitter and stability.**

***Index Terms*—global navigation satellite system (GNSS), adaptive scalar tracking loop (A-STL), loop-bandwidth control algorithm (LBCA).**

I. Introduction

A global navigation satellite system (GNSS) receiver acquires and tracks satellite signals [1]. Acquisition and tracking are necessary in order to decode the navigation message and determine the pseudorange to a given satellite. The better the receiver can track the signal, the better the range estimation, and, in turn, the more precise and reliable the position, velocity, and time (PVT) solution of the receiver. Therefore, a GNSS receiver must deploy the best possible means to track these signals for optimum performance. However, noise, receiver dynamics, and multipath effects make this a challenging task.

Traditionally, a scalar tracking loop (STL) is employed for signal tracking. An STL synchronizes to the carrier frequency, carrier phase, and code phase of each received GNSS signal. The tracking loop consists of a correlator, a discriminator, a loop filter, and a numerically controlled oscillator (NCO) [2]. The integration time T_i, the correlator spacing, the discriminator type, the order and the noise bandwidth of the loop filter, and the used oscillator determine the tracking performance at a given carrier-to-noise density ratio (C/N_0). Depending on the scenario, different loop settings are appropriate. For example, in a static scenario where the receiver is not moving, long integration times and small loop-bandwidths may be used for high precision tracking. On the contrary, in a dynamic scenario like for vehicular applications, larger loop-bandwidths are required such that the tracking loop does not lose lock. Therefore, it is challenging to design a loop that is optimal for all scenarios. A simple method is to select different parameters based on the scenario, but this requires the user to know the scenario and to update the parameters manually.

One issue with tracking loops is the ability to stay locked onto the signal in the presence of dynamic movements of the receiver. Dynamics stress the tracking loop. Changes in position, velocity, acceleration, jerk, or even higher orders of change characterize this parameter. The specific effect of the dynamics and the capability to stay in lock depend on the tracking loop-type (e.g., delay locked loop (DLL), phase locked loop (PLL), or frequency locked loop (FLL)), the loop order, and the loop-bandwidth. Typically higher-order loops can accommodate higher-order dynamics; however, this comes at the price of complexity, higher likelihood of instability, and delayed responses. Similarly, larger loop-bandwidths can respond quicker to disturbances but also come at the risk of increased instability and loop jitter. As such, a trade-off for tracking loop design exists.

The trade-off between user dynamics and noise compromises the validity of conventional STLs. Therefore, more robust tracking techniques are required. These techniques are divided into three main categories [3].

The first category is the *constant bandwidth technique* [4]. This is related to minor modifications in the STL architecture. No adaption of the loop-bandwidth is done. The integration time T_i, the type of discriminator, and the order of the loop filter are the parameters used to improve robustness [5]. However, in harsh and diverse scenarios, the STL cannot adjust for the required resilience.

The second category is the *variable loop-bandwidth tracking technique*. It solves the previous limitation using adaptive loop-bandwidth algorithms [6]–[11]. The adaption algorithm sets a connection between the loop-bandwidth and time-varying scenario conditions. This category allows adjusting the loop-bandwidth to an optimal value for a given scenario. One method of this category is the fast adaptive bandwidth (FAB) technique. It estimates permanently the input signal parameters (thermal noise, phase noise or steady state error (SSE)) of the STL using accurate models. According to these estimations, the optimal loop-bandwidth is calculated. A loop-bandwidth

978-1-7281-0244-3/20/$31.00 ©2020 IEEE

dependent cost function can be used [6], [7]. Equaling to null the first derivative of the cost function with respect to the loop-bandwidth leads to the optimal loop-bandwidth. Then, the algorithm updates the current loop-bandwidth. Another method is the estimation of the optimum pole of the loop filter using a cost function to determine the optimal loop-bandwidth [8], [9]. Since an abrupt change of the optimal loop-bandwidth may create tracking instabilities, the update stage uses the Newton-Raphson method and a low pass filter to smooth the new estimated pole. In addition, it is possible to create a cost function based on a weighted sum of estimated statistics from the discriminator's output [10]. Another technique from this category is the fuzzy logic based tracking technique. A fuzzy logic control algorithm based on the statistics of the discriminator's output can be implemented [11]. This technique simplifies significantly the complexity of the control algorithm. However, a previous knowledge of the scenario is required in order to achieve an efficient tracking performance.

The third and last category is the *Kalman filtering (KF)-based tracking techniques*. Similar to the second category, the KF implemented in conventional STLs predicts the loop-bandwidth depending on the scenario conditions. KF-based tracking techniques present a lower equivalent noise bandwidth than traditional STLs for the same signal dynamics [12]. This leads to a better tracking performance since the KF-based technique can filter more noise without losing the lock. However, this comes at the price of more computational complexity. A KF-based technique for 3rd order digital PLL has been proposed [13]. The KF reaches a steady state equivalent noise bandwidth that represents the minimum mean square error (MSE) between the input and the replica signal generated by the NCO. This technique estimates the measurement and process noise covariance considering the thermal noise of the channel and the SSE of the filter. A variable coefficient inversely proportional to the C/N_0 varies the process noise covariance, that sets the equivalent noise bandwidth of the loop. This coefficient determines the sensitivity to dynamics. Although the proposed weighting method is of great interest, an incomplete estimation of the noise statistics may lead to inadequate predictions.

This paper presents a novel bandwidth control algorithm for STLs. The algorithm adapts the noise bandwidth of the tracking loops, depending on the dynamics and noise, for the best performance. Further, this algorithm can be applied to any loop type and any loop filter order, without the need for excessive tuning and calibration. The algorithm improves generic variable loop-bandwidth tracking techniques and stands out for its simplicity and its stability.

A background on STLs and the theoretical tracking performance is presented in Section II, followed with a detailed description of the algorithm in Section III. Section IV shows the results of an adaptive PLL in an open software interface GNSS hardware receiver. Finally, the conclusions are drawn in Section V.

II. Scalar Tracking Loop

This section presents the theory to standard STLs [14]. Fig. 1 shows the general model of an STL. The output of the detector is the error signal $\theta^u[n]$ which is the difference between the incoming signal $y[n;\theta[n]]$ and its replica $y[n;\theta^s[n]]$. The loop filter suppresses the noisy errors $\theta^u[n]$ to a smoothed error rate $\dot{\theta}[n]$. Hence, the filter input is referred to as an unsmoothed signal and the filter output as a smoothed one. The order of the loop determines the robustness of the tracking against high-order dynamics. A higher order means the possibility to track higher-order components of the error signal while adding more complexity in the system. The resulting smoothed error rate $\dot{\theta}[n]$ drives the NCO. The NCO sends the smoothed error $\theta^s[n]$ to the replica generator. The generated replica signal $y[n;\theta^s[n]]$ approaches $y[n;\theta[n]]$ such that $\theta^u[n]$ is minimized.

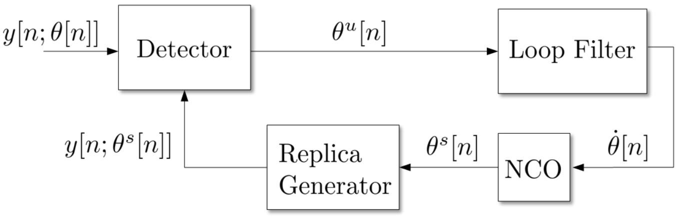


Fig. 1: Architecture of a conventional STL

Fig. 2 presents the general structure of the detector. The mentioned unsmoothed error signal $\theta^u[n]$ is the output of a discriminator block. This block extracts the error from a vector of input data. For instance, a phase discriminator measures the phase between the prompt in-phase and quadrature-phase (IQ) values of the received signal. Moreover, a code discriminator measures the code offset from the early, prompt and late IQ values. Before the discriminator block, the signal goes through an integration and dump (IAD) module in order to sample and integrate the signal. The integration time T_i reduces proportionally the noise of the input data to the discriminator. A larger integration time T_i achieves more processing gain, resulting in less noisy errors $\theta^u[n]$. However, the loop response becomes slower, and consequently, less robust against signal dynamics.

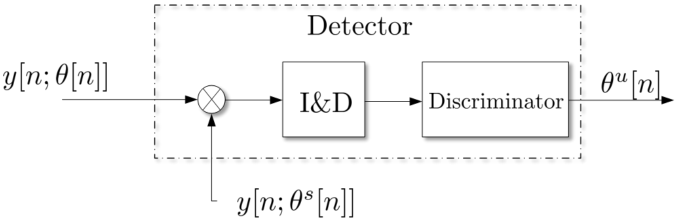


Fig. 2: Architecture of the detector

A. Theoretical Tracking Performance

The most significant error sources of an STL are the smoothed thermal noise σ^s_thermal and the dynamic stress error

θ_{sse}. The three-sigma rule-of-thumb is applied to the STL in order to have a moderate jitter in the tracking channel [15]:

$$\sigma_{\theta}^{s} = \sigma_{\text{thermal}}^{s} + \frac{\theta_{\text{sse}}}{3} \leq \sigma_{\theta^{s}}^{th} \tag{1}$$

where $\sigma_{\theta^s}^{th}$ is the one-sigma rule threshold. Values of σ_{θ}^{s} greater than $\sigma_{\theta^s}^{th}$ mean that the lock of the tracking is most probably lost. This upper threshold depends on the type of discriminator. For instance, the one-sigma rule threshold in meters for a two-quadrant arctangent discriminator is:

$$\sigma_{\theta^{s}}^{th} = \frac{\Omega}{12} \times \frac{\lambda}{2\pi} = \frac{\pi}{12} \times \frac{\lambda}{2\pi} = \frac{\lambda}{24} \tag{2}$$

where λ is the wavelength of the GNSS signal and Ω is the phase pull-in range in radians. Consider that the one-sigma rule threshold is one-third of the three-sigma rule threshold. In addition, one-fourth of the phase pull-in range is selected to have a conservative threshold. That is the reason of including the $1/12$ factor in (2).

The smoothed thermal noise jitter in meters of a p^{th} order Costas PLL with an arctangent discriminator is [15]:

$$\sigma_{\text{thermal}}^{s} = \frac{\lambda}{2\pi}\sqrt{\frac{B}{C/N_0}\left(1 + \frac{1}{2T_{\text{i}}C/N_0}\right)} \tag{3}$$

In addition, the dynamic stress error in meters can be expressed as:

$$\theta_{\text{sse}} = \left(\frac{\mu}{B}\right)^{p} \cdot \frac{\partial^{p}R}{\partial t^{p}} \tag{4}$$

where $\partial^{p}R/\partial t^{p}$ is the maximum line-of-sight (LOS) steady state error dynamics and μ is the relation coefficient between the noise loop-bandwidth B and the natural frequency of the tracking loop.

The Cramér-Rao bound (CRB) indicates the minimum error variance of an unbiased estimator [16]. In a stationary scenario, the CRB of the smoothed error in a Costas PLL CRB_{PLL}^{s} is the thermal noise variance:

$$CRB_{\text{PLL}}^{s} = (\sigma_{\text{thermal}}^{s})^{2} \tag{5}$$

Considering the STL as a time of arrival (ToA) unbiased estimator, the relationship between the smoothed and unsmoothed error jitter is [17]:

$$\sigma_{\theta}^{s} = \sigma_{\theta}^{u}\sqrt{2BT_{\text{i}}(1 - 0.5BT_{\text{i}})} \tag{6}$$

If the loop-bandwidth B tends to zero, then it can be further simplified:

$$\lim_{B \to 0} \sigma_{\theta}^{s} = \sigma_{\theta}^{u}\sqrt{2BT_{\text{i}}}\ : \tag{7}$$

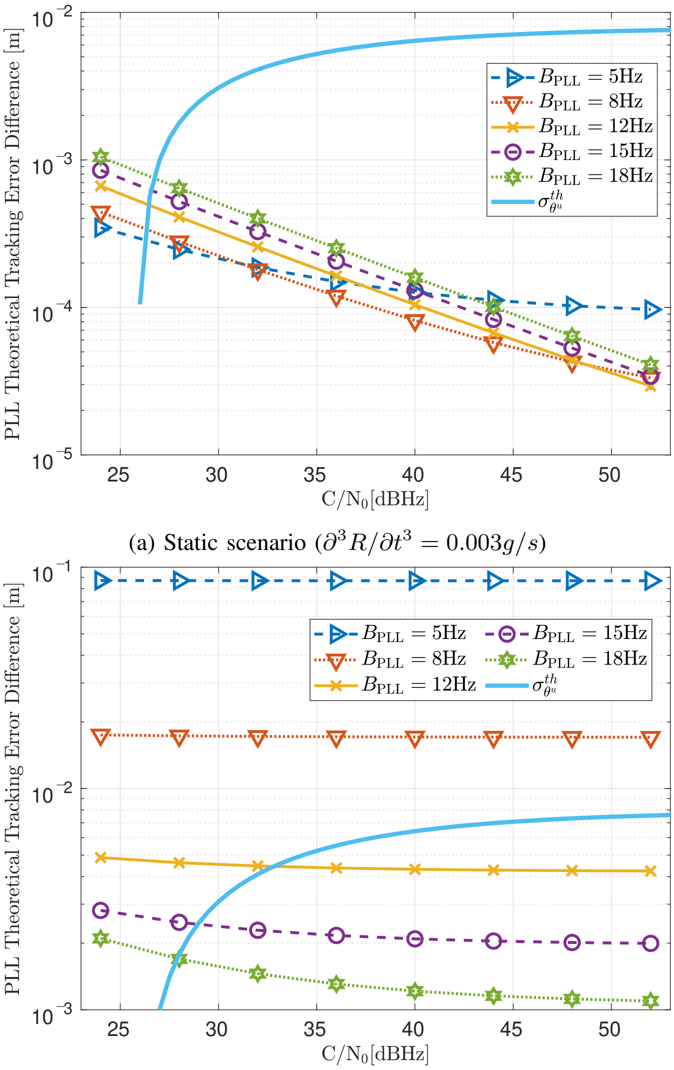
The resulting CRB of the unsmoothed error is:

$$CRB_{\text{PLL}}^{u} = \left(\frac{\lambda}{2\pi}\right)^{2}\left[\frac{1}{2T_{\text{i}}C/N_0}\left(1 + \frac{1}{2T_{\text{i}}C/N_0}\right)\right] \tag{8}$$

CRB_{PLL}^{u} indicates the minimum variance at the output of the discriminator and is independent on the loop-bandwidth.

In order to decide if the tracking is most probably lost or not, the same conservative one-sigma rule threshold used for the smoothed error is applied to the unsmoothed error:

$$\sigma_{\theta^{u}}^{th} = \sigma_{\theta^{s}}^{th} \tag{9}$$



(a) Static scenario ($\partial^3 R/\partial t^3 = 0.003g/s$)

(b) High dynamic scenario ($\partial^3 R/\partial t^3 = 3g/s$)

Fig. 3: Theoretical tracking error difference of a 3^{th} order PLL

where $\sigma_{\theta^u}^{th}$ is the one-sigma rule threshold of θ^{u}.

Fig. 3 shows a first theoretical approach of the tracking error difference in a 3^{rd} order Costas PLL for different C/N_0 and loop-bandwidth levels. The GNSS signal is GPS L1 C/A and the integration time T_{i} is set to 20 ms. A static scenario Fig. 3a and a high dynamic scenario Fig. 3b are shown. These graphs present the difference between the standard deviation of the unsmoothed error σ_{θ}^{u} and the square root of CRB_{PLL}^{u}. The one-sigma rule threshold $\sigma_{\theta^u}^{th}$ in meters is also included.

Since it is not probable to have an ideal static scenario due to unpreventable dynamic sources (e.g. clock drift), the static scenario includes small dynamics ($\partial^3 R/\partial t^3 = 0.003g/s$) in order to have a closer approximation to real scenarios.

The theoretical optimal normalized dynamics $\bar{D}_{\text{theory}}$ is noteworthy to mention, since it is an important parameter to consider for the loop-bandwidth control algorithm (LBCA). $\bar{D}_{\text{theory}}$ is the ratio between the dynamics stress error and the

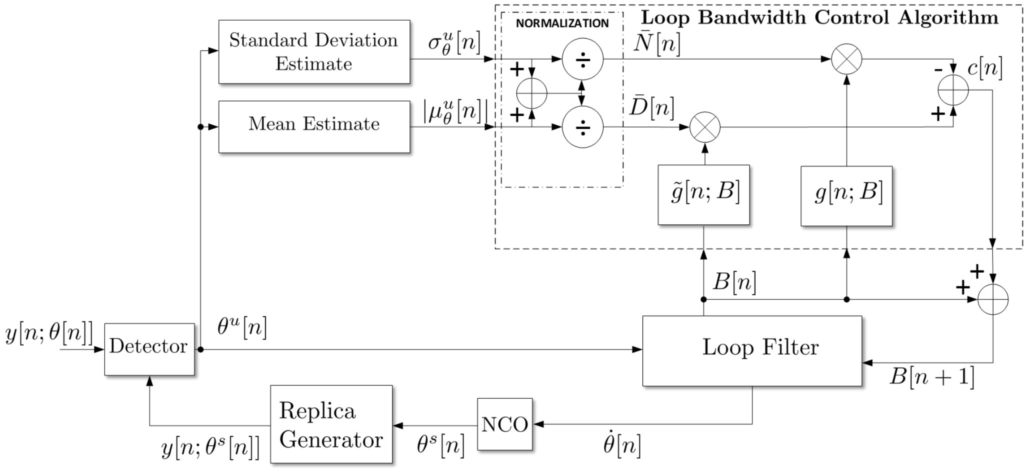


Fig. 4: Architecture of LBCA

three-sigma thermal noise:

$$\bar{D}_{\text{theory}} = \frac{\theta_{\text{sse}}}{3\sigma^s_{\text{thermal}} + \theta_{\text{sse}}} \tag{10}$$

For a Costas PLL, at the optimal loop-bandwidth B_{opt}, the normalized dynamics depends only on the loop order p:

$$\lim_{B \to B_{\text{opt}}} \bar{D}_{\text{theory}} = \bar{D}^{opt}_{\text{theory}} = \frac{1}{1+2p} \tag{11}$$

This information is considered later to determine the weighting function of the LBCA.

III. Loop-Bandwidth Control Algorithm

The time of response of an STL determines the tracking performance. A slow response time is desired in a noisy scenario, whereas a fast response is preferred in a high dynamic one. The proposed algorithm modifies the time of response of an STL depending on the estimated signal dynamics and noise statistics. The selected parameter that defines the time of response of an STL is the equivalent noise bandwidth. Therefore, this algorithm is called *loop-bandwidth control algorithm.*

A. Algorithm Description

Fig. 4 shows the diagram of the LBCA for a particular tracking channel. The proposed LBCA estimates the next loop-bandwidth $B[n+1]$, based upon the current loop-bandwidth $B[n]$ and a control signal $c[n]$:

$$B[n+1] = B[n] + c[n] \tag{12}$$

The control signal $c[n]$ is a weighted difference between the estimated normalized dynamics $\bar{D}$ and normalized noise statistics $\bar{N}$ of the channel. The coefficients $\tilde{g}$ and g are loop-bandwidth dependent functions that weight $\bar{D}$ and $\bar{N}$, respectively:

$$c[n] = \tilde{g}\left[n; B[n]\right] \bar{D} - g\left[n; B[n]\right] \bar{N} \tag{13}$$

The inputs of this algorithm are the absolute mean $|\mu_\theta^u[n]|$ and the standard deviation σ_θ^u of the unsmoothed error. These are related with the signal dynamics and noise. A first order low-pass infinite impulse response (IIR) filter estimates these statistics. The coefficients α and β determine the speed of the estimation. In the case of the absolute mean the IIR filter is applied directly to the discriminator's output θ^u:

$$\mu_\theta^u[n] = \alpha \cdot \mu_\theta^u[n-1] + \beta \cdot \theta^u[n] \tag{14}$$

In the case of the standard deviation σ_θ^u, the power of the unsmoothed error r must be estimated first:

$$r[n] = \alpha \cdot r[n-1] + \beta \cdot \theta^u[n]^2 \tag{15}$$

$$\sigma_\theta^u[n] = \sqrt{r[n] - \mu_\theta^u[n]^2} \tag{16}$$

The normalized bandwidth B_{IIR} sets the coefficients of the IIR filter. B_{IIR} varies according to the integration time of the correlators T_{i} and the desired cut-off frequency f_{co}:

$$B_{\text{IIR}} = \frac{f_{\text{co}} \cdot T_{\text{i}}}{2} \tag{17}$$

The gain of the IIR filter is unitary:

$$\alpha + \beta = 1 \tag{18}$$

As mentioned, the coefficients are a function of the selected normalized bandwidth B_{IIR} [18]:

$$\alpha = 2 - \cos\left(\pi B_{\text{IIR}}\right) - \sqrt{\left(2 - \cos\left(\pi B_{\text{IIR}}\right)\right)^2 - 1} \tag{19}$$

The normalization of $|\mu_\theta[n]|$ and $\sigma_\theta[n]$ are $\bar{D}$ and $\bar{N}$:

$$\bar{D} = \frac{|\mu_\theta^u[n]|}{|\mu_\theta^u[n]| + \sigma_\theta^u[n]} \tag{20}$$

$$\bar{N} = \frac{\sigma_\theta^u[n]}{|\mu_\theta^u[n]| + \sigma_\theta^u[n]} \tag{21}$$

The weights $\tilde{g}$ and g are positive functions and bandwidth dependent. In addition, they are related by the following expression:

$$\tilde{g}\left[n; B[n]\right] = g_{\text{Max}} - g\left[n; B[n]\right] \tag{22}$$

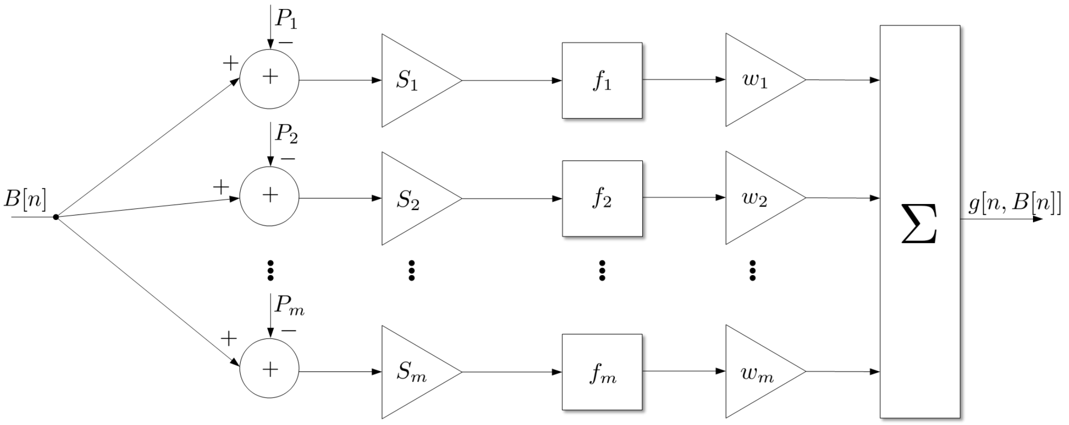


Fig. 5: Architecture of the weighting function

Equation (13) can be further developed:

$$c[n] = g_{\text{Max}}\bar{D} - g\,[n; B[n]]\,(\bar{N} + \bar{D}) \tag{23}$$

where g_{Max} represents the maximum value of g. Since $\bar{N} + \bar{D}$ equals to 1,

$$c[n] = g_{\text{Max}}\bar{D} - g\,[n; B[n]]\,. \tag{24}$$

The weighting function $g\,[n; B[n]]$ determines the impact of the estimators in the adaptive algorithm. Fig. 5 presents the structure of $g\,[n; B[n]]$. It consists of a linear combination of m normalized, positive and monotone increasing functions:

$$g\,[n; B[n]] = \sum_{k=0}^{m} w_k f_k\,(S_k(B[n] - P_k)) \tag{25}$$

Each function f_k is shifted by P_k, vertically scaled by w_k and horizontally scaled by S_k. The summation of each w_k leads to the maximum value of $g\,[n; B[n]]$, g_{Max}:

$$g_{\text{Max}} = \sum_{k=0}^{m} w_k \tag{26}$$

This parameter determines the speed of the algorithm to reach the optimum loop-bandwidth. The higher g_{Max}, the faster the algorithm reacts to dynamics but also the more sensible it becomes to the estimators' noise.

The tendency of $g\,[n; B[n]]$ is presented in two different scenarios. In a noisy stationary scenario the signal dynamics are negligible ($\bar{D} \approx 0$) and the theoretical optimal loop-bandwidth would tend to zero, since the noise must be filtered as much as possible. Therefore, the loop-bandwidth must decrease until it reaches zero. Applying the LBCA in such a scenario, the control signal becomes a negative function:

$$c[n] = -g\,[n; B[n]] \tag{27}$$

Consequently, the loop-bandwidth decreases every time the loop is closed. Once the loop-bandwidth tends to zero, the control signal must stop being negative. This means that the weighting function must tend to zero when the loop-bandwidth also tends to zero.

An opposite scenario is a low noisy scenario with high dynamics. In this case, the normalized dynamic estimate predominates ($\bar{D} \approx 1$). The theoretical optimal loop-bandwidth must tend to the highest possible loop-bandwidth in order to react as fast as possible to signal dynamics. In such scenario, the control signal is a positive function and increases the loop-bandwidth every time the loop is closed:

$$c[n] = g_{\text{Max}} - g\,[n; B[n]] \tag{28}$$

Due to the increase of the loop-bandwidth, the weighting function increases and, in turn, the control signal decreases. The loop-bandwidth reaches its maximum value when the weighting function tends to g_{Max}.

Supposing that the estimation error is negligibly small, the updated loop-bandwidth reaches to a *point of equilibrium*. The *point of equilibrium* is achieved when the control signal $c[n]$ tends to zero and no longer changes the bandwidth. The control signal is zero when the weighting function $g\,[n; B[n]]$ reaches a value that balances the dynamics and the noise estimations.

B. Extension of the Algorithm

The LBCA can be further extended updating the loop-bandwidths of L tracking channels:

$$\mathbf{B}_{L\times 1}[n+1] = \mathbf{B}_{L\times 1}[n] + \mathbf{c}_{L\times 1}[n] \tag{29}$$

$\mathbf{B}_{L\times 1}$ is the loop-bandwidth vector and $\mathbf{c}_{L\times 1}$ the control vector. The latter updates the loop filter's noise bandwidth for each tracking channel:

$$\mathbf{c}_{L\times 1} = \tilde{\mathbf{g}}_{L\times L}\bar{\mathbf{D}}_{L\times 1} - \mathbf{g}_{L\times L}\bar{\mathbf{N}}_{L\times 1} \tag{30}$$

where $\tilde{\mathbf{g}}_{L\times L}$ and $\mathbf{g}_{L\times L}$ are diagonal matrices:

$$\tilde{\mathbf{g}}_{L\times L} = diag(\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_L) \tag{31}$$

$$\mathbf{g}_{L\times L} = diag(g_1, g_2, \ldots, g_L) \tag{32}$$

The performance of the algorithm depends on the family of functions used to determine the weighting function. The next section presents a particular case of this generic algorithm showing a specific family of normalized sigmoid functions.

C. Sigmoid-based Weighting Function

Considering the defined mathematical model of an LBCA, a particular case implementation using a sigmoid-based weighting function is presented. A sigmoid function is defined as:

$$\mathrm{Sig}(x) = \frac{1}{1+\mathrm{e}^{-x}} \tag{33}$$

The properties of the sigmoid function provide great advantages in the LBCA. First, the sigmoid function is bounded to [0,1]:

$$0 \leq \mathrm{Sig}(x) \leq 1 \tag{34}$$

Second, the reverse function of the sigmoid is:

$$\mathrm{Sig}(-x) = 1 - \mathrm{Sig}(x) \tag{35}$$

Third, the derivative of the sigmoid function being biased by P, horizontally scaled by S and vertically scaled by w is:

$$w\,\frac{\partial \mathrm{Sig}(S(x-P))}{\partial x} = w\,S\,\mathrm{Sig}(S(x{-}P))\mathrm{Sig}(S(P{-}x)) \tag{36}$$

The sigmoid-based weighting function has the following expression:

$$g_{\mathrm{Sig}}\left[n; B[n]\right] = \sum_{k=0}^{m} w_k Sig_k(S_k(B[n]-P_k)) \tag{37}$$

Considering the first property of the sigmoid, the sigmoid-based weighting function is bounded to $[0, g_{\mathrm{max}}]$:

$$0 \leq g_{\mathrm{Sig}}\left[n; B[n]\right] \leq g_{\mathrm{max}} \tag{38}$$

Due to the second property, the weighting function used for the signal dynamics estimation $\tilde{g}_{\mathrm{Sig}}\left[n; B[n]\right]$ can be expressed as:

$$\tilde{g}_{\mathrm{Sig}}\left[n; B[n]\right] = g_{\mathrm{Sig}}\left[n; -B[n]\right] \tag{39}$$

In addition, the derivative of g_{Sig} is:

$$\frac{\partial g_{\mathrm{Sig}}\left[n; B[n]\right]}{\partial B[n]} = \sum_{k=0}^{m} w_k \frac{\partial Sig_k(S_k(B[n]-P_k))}{\partial B[n]} \tag{40}$$

The slope of the weighting function depends on the horizontal scaling S_k and vertical scaling w_k and determines how fast the weighting function changes. The bigger the slope, the faster the change of the weighting function.

The control signal using the sigmoid-based weighting function can be represented as:

$$c[n] = \sum_{k=0}^{m} w_k \bar{D} - \sum_{k=0}^{m} w_k Sig_k(S_k(B[n]-P_k)) \tag{41}$$

The fundamental step of this algorithm is to choose the appropriate parameters (w_k, S_k, P_k) of the sigmoid function. An empirical approach determines these parameters. First, the theoretical optimum loop-bandwidth of the STL is calculated using a loop-bandwidth dependent cost function. This function is based on a theoretical characterization of the signal dynamics and noise of the tracking channel [15]. Second, a real characterization of the STL implemented in a GNSS receiver must be done. The theoretical model and the real results are compared in order to know how much is the difference between them. Finally, according to the analysis done, the number of sigmoid functions m, the vertical scaling w_k, the horizontal scaling S_k and the biases P_k are chosen.

The following section compares the theoretical tracking performance of the PLL presented in Section II with the tracking performance of the PLL implemented in a GNSS with an open software interface. This comparison serves to achieve a correct configuration of the LBCA. Then, the tracking performance of the PLL using the LBCA is compared with the traditional PLL.

IV. Results

The GOOSE platform from Fraunhofer IIS is a GNSS Receiver with an open software interface [19]. Its tracking stage contains a 3rd order PLL and a 2nd order DLL. In addition, the PLL assisted DLL (PAD) is enabled. The purpose is to evaluate the performance of the LBCA implemented in the PLL of this receiver against simulated scenarios with different dynamics and noise levels. These simulations use GPS L1 C/A signals. The tracking correlators use a maximum integration time T_i of 20 ms.

First, the characterization of the PLL in the GOOSE platform is performed and it is compared with the theoretical analysis. Finally, the sigmoid-based weighting function is selected and the performance of the LBCA in a stationary and a high dynamic scenario is presented.

Fig. 6 shows the setup in which the GOOSE platform receives synthetic data from a GNSS signal simulator. A Python script located in the user computer is used to perform automated tests. The script configures first the scenario of the GNSS simulator through transmission control protocol (TCP). Second, the simulator runs and the user computer accesses the GOOSE platform through secure shell (SSH) in order to run the tracking application. Then, the GOOSE platform acquires and tracks the GNSS signal. Finally, the user computer receives the tracking information from the GOOSE platform and the data is stored and evaluated.

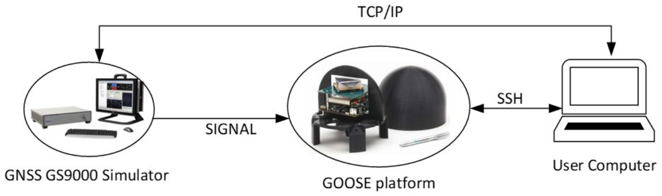


Fig. 6: Test setup with the signal simulator and receiver platform

A. PLL Tracking Performance

To evaluate correctly the algorithm, the tracking performance of the conventional PLL implemented in GOOSE is analyzed. The duration of the simulation is 20 minutes. For a particular PLL loop-bandwidth B_{PLL}, the simulation repeats for different C/N_0 levels. In this case, the simulation repeats 8 times, from 24 dBHz to 52 dBHz with a step size of 4 dB.

The same method is done for different B_{PLL} values. The loop-bandwidth of the DLL B_{DLL} remains constant for all the simulations with a loop-bandwidth of 0.1 Hz.

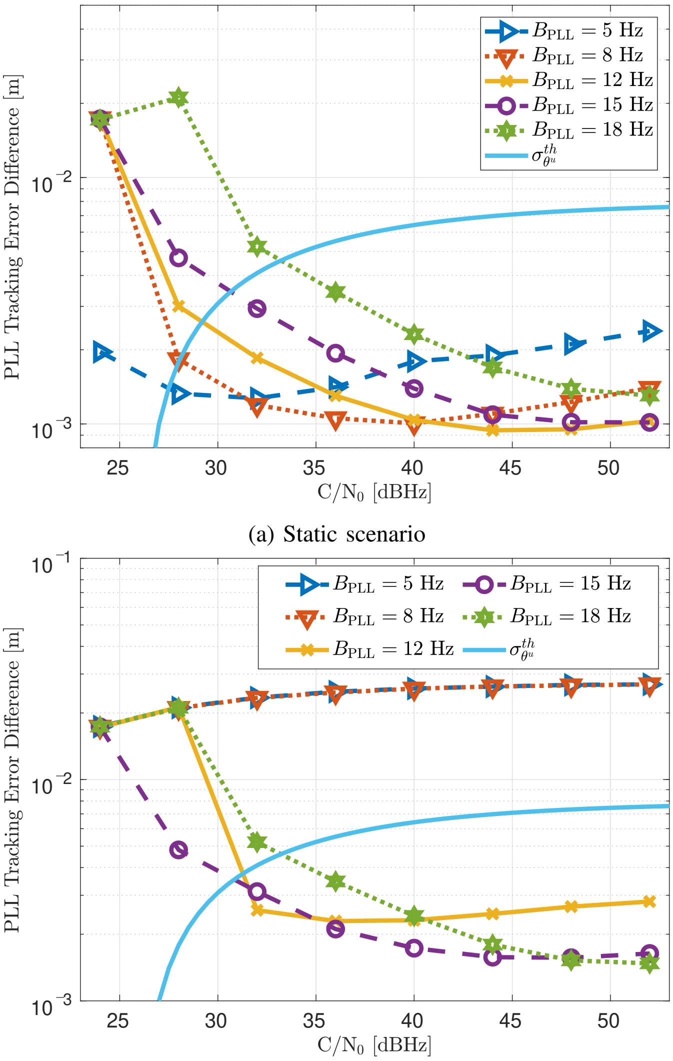
Since the sensitivity of the acquisition is lower than the sensitivity of the tracking, the simulation starts always at the highest C/N_0 level, 52 dBHz. The C/N_0 level is reduced 4 dB each 30 seconds until reaching the desired level. For instance, 5 minutes are necessary to reach a C/N_0 level of 24 dBHz. Therefore, in order to assure that the measured tracking performance is reliable, the last 5 minutes of the simulation are considered.

The CRB at the output of the NCO is the minimum variance of the smoothed error (3) and determines the performance of the PLL. However, since it depends on the loop-bandwidth, it is not a reliable parameter to determine the performance of the PLL in variable loop-bandwidth tracking techniques. In addition, since the NCO of the GOOSE platform is implemented in hardware, it is not possible to collect data of the smoothed error from the software. Therefore, the CRB of the unsmoothed error CRB^{u}_{PLL} is chosen to evaluate the tracking performance. In contrast to the CRB of the smoothed error, CRB^{u}_{PLL} does not depend on the loop-bandwidth (8). Moreover, the unsmoothed error is easily available from software.

Fig. 7 shows the tracking performance of the conventional 3^{rd} order PLL in a static and a dynamic simulated scenario. For each simulation a five minutes average of the C/N_0 is performed. The standard deviation of the arctangent discriminator's output is also calculated. The latter parameter is converted to meters and it indicates the tracking error jitter. These graphs show the difference between the obtained tracking error jitter and the square root of CRB^{u}_{PLL} for different C/N_0 and loop-bandwidths. The one-sigma threshold of the unsmoothed error is also added since it indicates if the tracking is in lock or not.

In the stationary scenario the GOOSE platform receives a GPS L1 C/A signal from a visible satellite located near the zenith. The tracking error difference in this scenario is presented in Fig. 7a. In this scenario, although the receiver is static, low signal dynamics are still present due to the movement of the satellites, the clock jitter, the clock drift and non-linearities of the receiver's frontend. Therefore, the static scenario still has low dynamics present. The real signals have more degradation than in theory. This is due to the fact that the theory takes only the thermal noise into account and not other error sources. Even having an error bias in the obtained results, the tendency is similar to the theoretical one. The lower the loop-bandwidth is, the lower the tracking error at low C/N_0. However, at higher C/N_0 levels a higher loop-bandwidth is adequate in order to have minimum tracking error.

The dynamic scenario receives a GPS L1 C/A signal from a visible satellite located near the horizon. This low-elevation satellite is more sensitive to receiver dynamics. In this scenario, the receiver is in a vehicle that has a high-dynamic trajectory. The tracking error difference for this dynamic scenario is presented in Fig. 7b. Fig. 8 shows the line-of-sight steady state error dynamics for this scenario. The maximum jerk is $\partial^3\mathrm{R}/\partial\mathrm{t}^3 = 7.8\mathrm{g/s}$. Setting a PLL loop-bandwidth of 5 Hz and 8 Hz, the tracking loop is not able to follow the high signal dynamics in any C/N_0 level. However, for higher loop-bandwidths, it is still possible to track the signal. The obtained results presents a coherent behavior comparing to the theoretical values.



(a) Static scenario

(b) High dynamic scenario

Fig. 7: Tracking error difference of a 3^{rd} order Costas PLL

B. Adaptive PLL Tracking Performance

The LBCA is implemented in the PLL of the GOOSE platform. First, the configuration of the IIR estimators is described. Then, the parameters of the sigmoid-based weighting function are selected based on the theoretical normalized dynamics. Finally, the initial configuration of the GOOSE platform is indicated and the obtained results are shown.

The estimators for the adaptive phased locked loop (APLL) uses an IIR filter with a cut-off frequency f_{co} of 5 Hz. Therefore, the effective exponential moving average of 200 ms of data is achieved. Hence, $\alpha = 0.85$ and $\beta = 0.15$. This

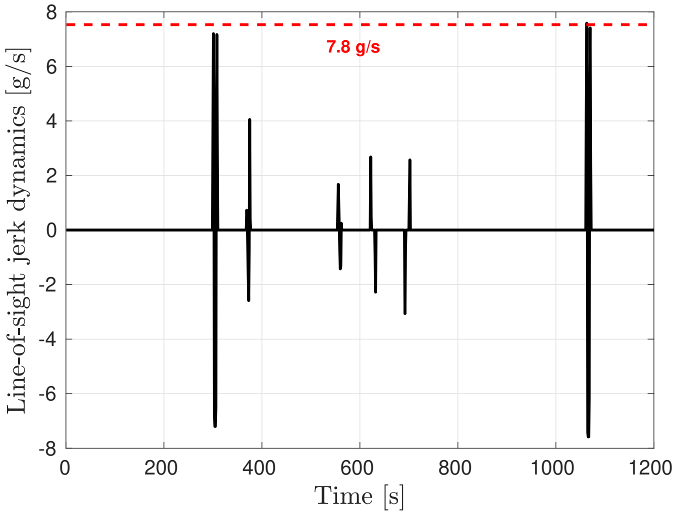


Fig. 8: LOS jerk dynamics of high dynamic scenario

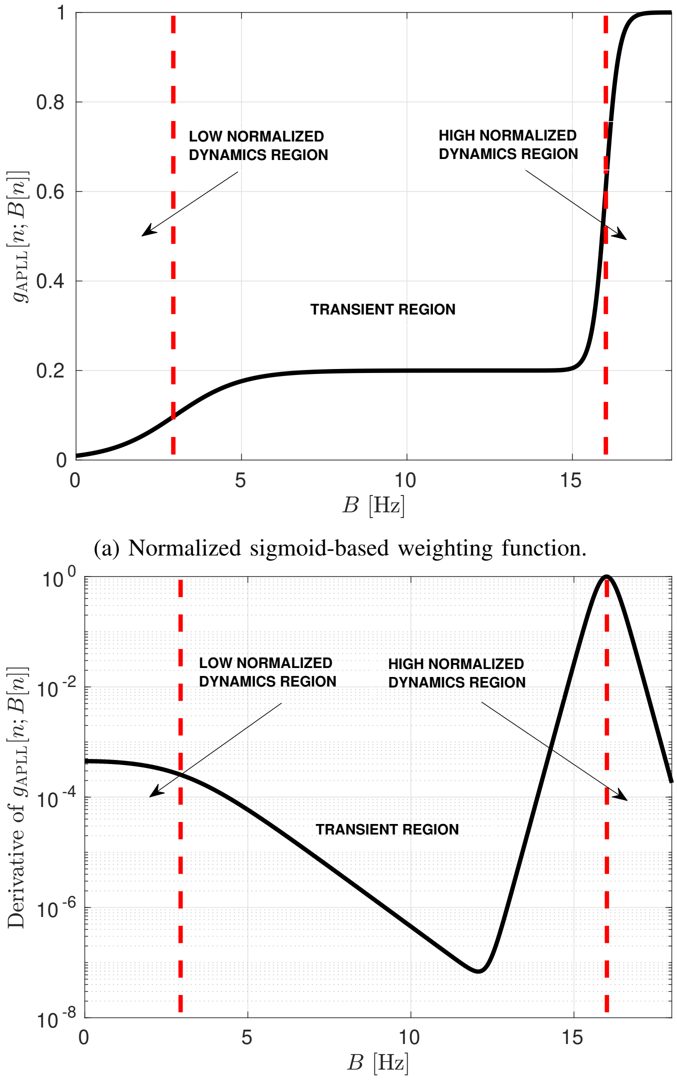
IIR filter estimates the mean $\mu_\theta^u[n]$ and the standard deviation $\sigma_\theta^u[n]$ of the unsmoothed phase error $\theta^u[n]$.

The sigmoid-based weighting function of the adaptive PLL $g_{\text{APLL}}[n; B]$ is a linear combination of two sigmoid functions:

$$g_{\text{APLL}}[n; B[n]] = \begin{bmatrix} 0.002 \\ 0.008 \end{bmatrix}^T \begin{bmatrix} \text{Sig}\left((B[n] - 3)\right) \\ \text{Sig}\left(5(B[n] - 16)\right) \end{bmatrix} \tag{42}$$

Fig. 9 shows the sigmoid-based weighting function. The shape of the function is presented in Fig. 9a. This function is distributed into three regions. The first region, $0-3$Hz, is the *low normalized dynamic region*. For a low C/N_0 level and low signal dynamics, the normalized dynamic $\bar{D}$ tends to zero. Then, the control signal is always negative (27), the loop-bandwidth decreases and, in turn, the weighting function tends to the low normalized dynamic region. The second region is the *transient region*, $3-16$Hz. The function in this region has practically a constant value. This value is chosen taking into account the theoretical optimal normalized dynamics $\bar{D}_{\text{theory}}^{opt}$ for a PLL (11). $\bar{D}_{\text{theory}}^{opt}$ gives a theoretical value based on the dynamic stress error and thermal noise. Since it is a 3$^{\text{rd}}$ order PLL, the $\bar{D}_{\text{theory}}^{opt}$ is $1/7$. This means that the optimal loop-bandwidth is located when 14.28% of normalized dynamics is achieved. It is probable that this value does not give the optimal normalized real dynamics $\bar{D}_{\text{real}}^{opt}$, since additional dynamics and noise sources are not considered. After some testing and calibration, the value of the $\bar{D}_{\text{real}}^{opt}$ is chosen to be around 0.18 and 0.2. Then, the optimal loop-bandwidth is located when $18-20\%$ of normalized dynamics is achieved. The third region, $16-18$Hz, is the *high normalized dynamic region*. For a high C/N_0 level and high signal dynamics, the normalized dynamic $\bar{D}$ tends to one. Then, the control signal is a positive value Eq. (28), the loop-bandwidth increases and, in turn, the weighting function tends to this region.

The derivative of the sigmoid weighting function in Fig. 9b indicates how fast the function changes in each region. The higher changes are located between regions and this leads to constraining the weighting function in the transient region.



(a) Normalized sigmoid-based weighting function.

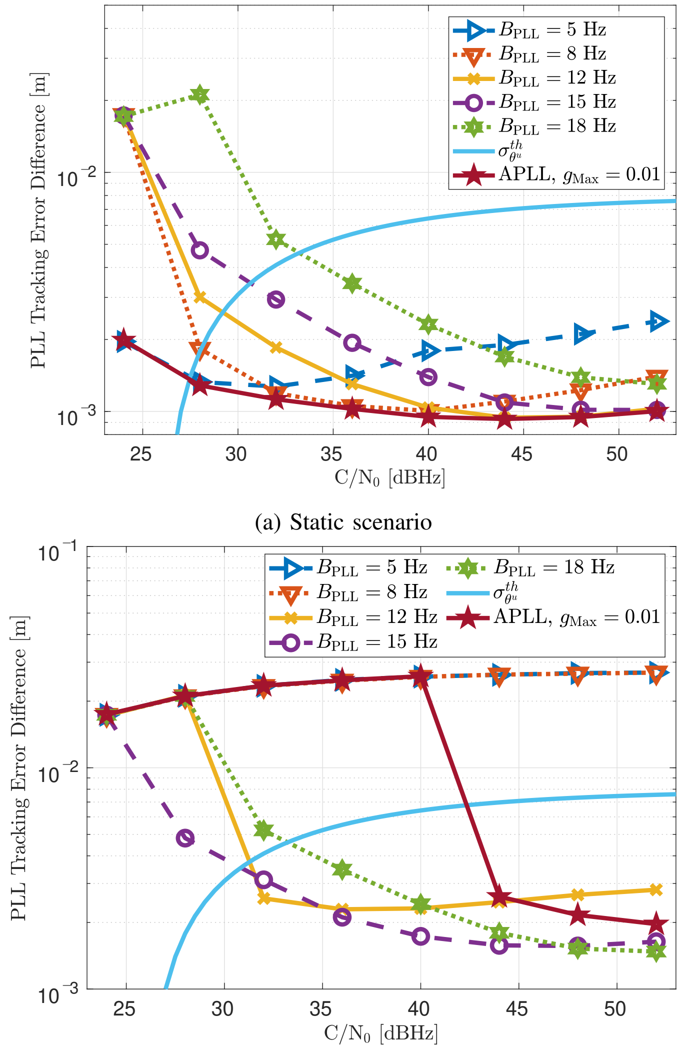
(b) First derivative of the sigmoid-based weighting function

Fig. 9: Sigmoid-based weighting function for the APLL

Since g_{Max} equals to 0.01, the maximum loop-bandwidth update is ±0.01 Hz for every integration interval T_{i}. If the signal dynamics and noise statistics are estimated without any uncertainty, a larger g_{Max} would be considered to reach to the optimum loop-bandwidth as fast as possible. This would also lead to a higher sensitivity against dynamics. However, the estimators used are very noisy and a high g_{Max} can lead to instabilities.

To evaluate the performance of the APLL, the same method as described for the PLL characterization is followed. The initial loop-bandwidth of the APLL is 8 Hz and the LBCA starts when T_{i} is 20 ms. Fig. 10 shows the tracking performance of the APLL compared to the traditional PLL in the same scenarios mentioned previously.

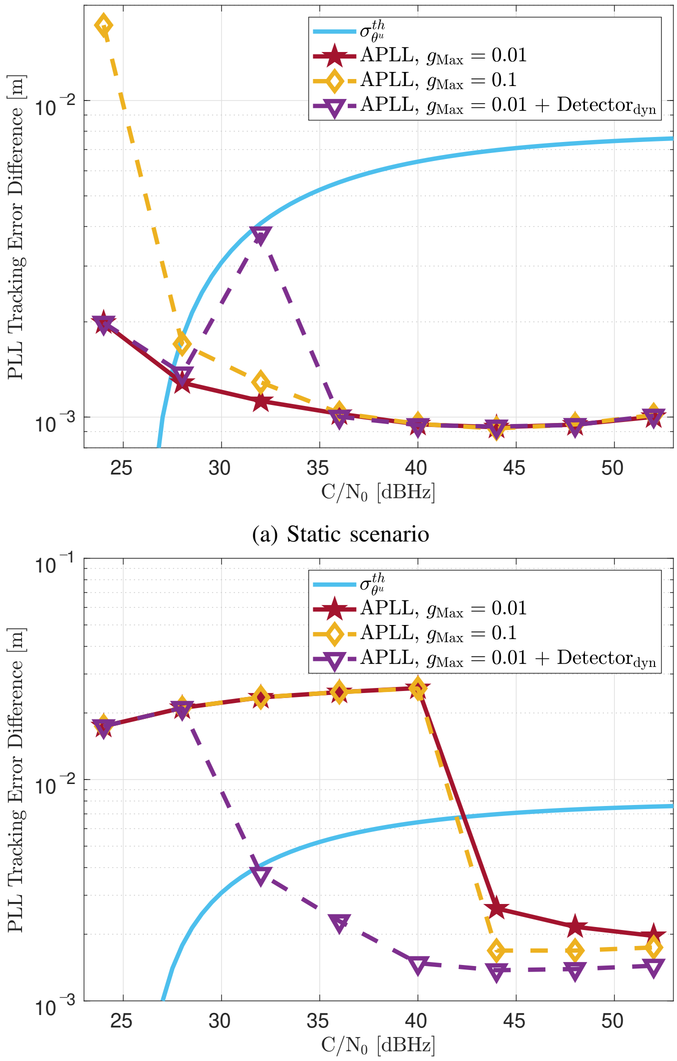
In the low dynamics scenario Fig. 10a, the adaptive LBCA



(a) Static scenario

(b) High dynamic scenario

Fig. 10: Tracking error difference comparison between PLL and APLL



(a) Static scenario

(b) High dynamic scenario

Fig. 11: Tracking error difference for different APLL settings

achieves the minimum tracking error for each C/N_0. However, in the high dynamics scenario Fig. 10b the algorithm becomes unstable under 44 dBHz. This is due to a slow response of the algorithm. The maximum value of the weighting function g_{Max} is low and consequently less sensitive to react against high signal dynamics. Another reason is that the estimators are too slow to detect high dynamic transients. More information about signal dynamics and noise statistics of the channel can improve significantly the performance of the algorithm. For instance, different tracking stages of the tracking loop or the output of a frequency discriminator can be considered to give more information to the LBCA.

Fig. 11 shows the effect of g_{Max}. In a stationary scenario Fig. 11a, a lower g_{Max} leads to a more stable update of the loop-bandwidth in low C/N_0 levels. However, a higher g_{Max} provides a better tracking performance in high dynamic scenarios Fig. 11b. A provisional solution is also presented using a *signal dynamics detector*. It is a threshold detector that defines if the tracking is getting lost due to high dynamics. The following conditions are considered:

- $C/N_0 > 30$dBHz
- $|\theta^u| - |\mu_\theta^u| > 2\sigma_{\theta^u}^{th}/3$
- $|\theta^u| - |\sigma_\theta^u| > \sigma_{\theta^u}^{th}/3$

If all the conditions are met, the detector removes one sigmoid function of the weighting function. Then, the weighting function becomes:

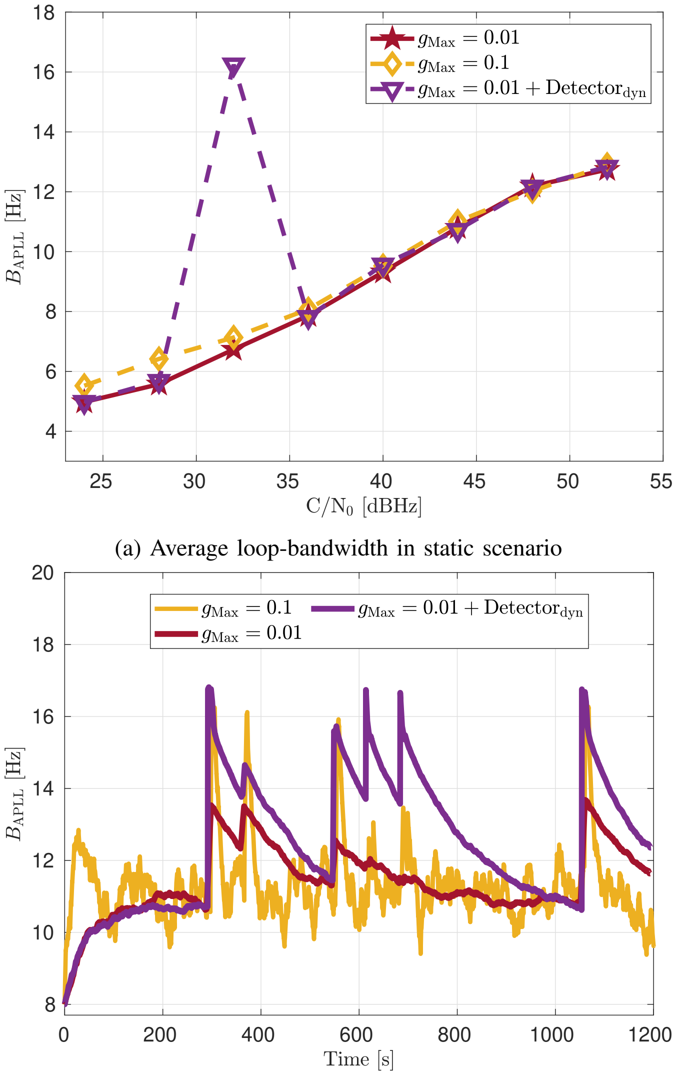
$$g_{\text{Detector}}[n; B[n]] = \text{Sig}\left(5(B[n] - 16)\right) \tag{43}$$

The transient region decreases to zero and g_{Max} is set to one. Hence, once high signal dynamics are detected, the threshold detector is activated and the weighting function changes in order to respond fast against high dynamics.

Fig. 12 shows the variation of the loop-bandwidth for each configuration of the sigmoid-based weighting function. For the stationary scenario, the average loop-bandwidth for each

C/N_0 is measured. Fig. 12a shows that a higher g_{Max} presents a noisier loop-bandwidth as the C/N_0 decreases. The *signal dynamics detector* stops working correctly at 32 dBHz. At this C/N_0 level, the detector assumes that there is dynamics while only noise is present. For that reason, this solution is only enabled for a C/N_0 higher than 30 dBHz.

Fig. 12b shows the variation of the loop-bandwidth in the high dynamic scenario at 45 dBHz. The LBCA responds slower against high signal dynamics using a low g_{Max} such as 0.01. However, the response is faster increasing g_{Max} to 0.1. This comes at the cost of a noisier loop-bandwidth update. The aforementioned provisional solution uses a low g_{Max} and the *signal dynamics detector*. If dynamics are detected, the weighting function is changed in order to respond as fast as possible. Once the presence of dynamics disappears, the detector is disabled and the weighting function goes to its previous state.



(a) Average loop-bandwidth in static scenario

(b) Loop-bandwidth in high dynamic scenario at 45 dBHz

Fig. 12: Loop-bandwidth variation for different APLL settings

V. Conclusion

This paper presents an adaptive LBCA for STLs. First, the description of the STL and its theoretical tracking performance is necessary to understand the foundation of the algorithm. Then, a detailed description of the algorithm shows the main structure, an extension, and a particular case example with sigmoid-based weighting functions. Finally, the implementation of the LBCA in a 3$^{\mathrm{rd}}$ order PLL proves the stability and superior performance of the algorithm in a stationary scenario. Moreover, it also shows the algorithm limitations in high dynamic scenarios.

In a high dynamic scenario, the algorithm is stable above 40 dBHz. If the maximum value of the sigmoid weighting function increases, the algorithm can react faster to dynamics, and the tracking performance increases. However, this comes at the risk of making unstable the algorithm. A provisional solution is presented using a switch to detect dynamics. If the lock is getting lost, the transient region of the weighting function is decreased, being sensitive against dynamics.

Future research includes adding more information about the signal dynamics and noise of the tracking channel. The presented algorithm takes only the statistics of the discriminator's output. The statistics of the different stages of the tracking loop, their respective order of derivatives and integrals can be considered to add more information. Moreover, in the case of a PLL, the output of the FLL discriminator can additionally benefit by adding more information of the signal dynamics.

The LBCA updates the time of response of a traditional STL. This algorithm can be also applied to update the time of response in KF-based tracking techniques, reducing significantly the complexity of the KF.

Optimizing the weighting function could benefit the algorithm's performance. Regarding the complexity of the algorithm, a sigmoid-based weighting function introduces much computational burden. However, it can be significantly reduced using a piece-wise linear approximation [20] [21].

The algorithm presents a good trade-off between simplicity and performance. Further, current results show promising adaption capability.

Acknowledgments

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

References

[1] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and Applications*, 2nd ed., 2006.

[2] F. M. Gardner, *Phaselock Techniques*, 3rd ed. Wiley, 2005.

[3] J. A. López-Salcedo, J. A. D. Peral-Rosado, and G. Seco-Granados, "Survey on robust carrier tracking techniques," *IEEE Communication Surveys & Tutorials*, vol. 12, no. 2, pp. 670 – 688, January 2014.

[4] T. Humphreys, M. Psiaki, P. J. Kintner, and B. Ledvina, "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)*, September 2005, pp. 156 – 167.

[5] P. L. Kazemi and C. O'Driscoll, "Comparison of assisted and stand-alone methods for increasing coherent integration time for weak GPS signal tracking," in *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, September 2008, pp. 1730 – 1740.

[6] S. Skone, G. Lachapelle, D. Yao, W. Yu, and R. Watson, "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)*, September 2005, pp. 1126 – 1137.

[7] K. Muthuraman, "Tracking techniques for GNSS data/pilot signals," Ph.D. dissertation, University of Calgary, January 2010.

[8] F. Legrand, C. Macabiau, J. Issler, L. Lestarquit, and C. Mehlen, "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)*, September 2000, pp. 2346 – 2356.

[9] F. Legrand and C. Macabiau, "Results of the implementation of the fast adaptive bandwidth lock loops on a real GPS receiver in a high dynamics context," *International Symposium on GNSS*, May 2001.

[10] S. Ugazio, L. L. Presti, and M. Fantino, "Design of real time adaptive DPLLs for generic and variable doppler frequency," in *Proceedings of the 2011 International Conference on Localization and GNSS (ICL-GNSS)*, June 2011, pp. 169 – 174.

[11] Y. W. Yuan Chen, Haixin Zheng, "Adaptive bandwidth PLL design based on fuzzy logic control," in *2011 4th IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, November 2011, pp. 645–648.

[12] X. Sun, H. Qin, and J. Niu, "Comparison and analysis of GNSS signal tracking performance based on kalman filter and traditional loop," *WSEAS Transactions on Signal Processing*, vol. 9, pp. 99 – 108, January 2013.

[13] J.-H. Won, "A novel adaptive digital phase-lock-loop for modern digital GNSS receivers," *IEEE Communication Letters*, vol. 18, no. 1, pp. 46–49, January 2014.

[14] M. Meurer and F. Antreich, *Springer Handbook of Global Navigation Satellite Systems - Signals and Modulation*, May 2017, pp. 91 – 119.

[15] D.-J. Jwo, "Optimisation and sensitivity analysis of GPS receiver tracking loops in dynamic environments," in *IEE Proc. - Radar, Sonar and Navigation*, August 2001, pp. 241 – 250.

[16] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, ser. Prentice-Hall Signal processing series. Prentice Hall, 1993, vol. 1.

[17] J. W. Betz and K. R. Kolodziejski, "Generalized theory of code tracking with an early-late discriminator Part I: Lower bound and coherent processing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, pp. 1538 – 1556, October 2009.

[18] J. Strydom, J. de Witt, and J. Cilliers, "High range resolution X-band urban radar clutter model for a DRFM-based hardware in the loop radar environment simulator," in *2014 International Radar Conference*, October 2014, pp. 1–6.

[19] M. Overbeck, F. Garzia, A. Popugaev, O. Kurz, F. Forster, W. Felber, A. S. Ayaz, S. Ko, and B. Eissfeller, "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)*, September 2015.

[20] C. Alippi and G. Gajani, "Simple approximation of sigmoidal functions: realistic design of digital neural networks capable of learning," in *IEEE International Symposium on Circuits and Systems*, vol. 3, July 1991, pp. 1505 – 1508.

[21] O. Cetin, F. Temurtas, and S. Gulgonul, "An application of multilayer neural network on hepatitis disease diagnosis using approximations of sigmoid activation function," *Dicle Medical Journal / Dicle Tip Dergisi*, vol. 42, pp. 150 – 157, June 2015.