Received: 27 March 2022 | Revised: 13 August 2022 | Accepted: 24 October 2022

DOI: 10.33012/navi.570

ORIGINAL ARTICLE

Multi-Parameter Adaptive Notch Filter (MPANF) for Enhanced Interference Mitigation

Johannes Rossouw van der Merwe | **Iñigo Cortés** | **Fabio Garzia** | **Alexander Rügamer** | **Wolfgang Felber**

Satellite Based Positioning Systems Department, Fraunhofer IIS, Germany

Correspondence
Johannes Rossouw van der Merwe
Nordostpark 84, 90411 Nuremberg, Germany
Email: johannes.roussouw.vandermerwe@iis.fraunhofer.de

Abstract

Interference signals degrade global navigation satellite system (GNSS) performance and must be mitigated. Chirp signals can be mitigated with an adaptive notch filter (ANF), but the dynamic behavior limits performance. An ANF determines the instantaneous frequency and removes interference with a notch filter. However, there are several limitations. In this article, we propose a multi-parameter adaptive notch filter (MPANF) approach that significantly enhances conventional ANFs. First, it uses an loop-bandwidth control algorithm (LBCA) to alter the loop bandwidth of an frequency-locked loop (FLL)-based adaptation algorithm to facilitate superior tracking agility-to-precision trade-off. Second, it dynamically adjusts the notch depth to switch on interference mitigation or pass the signal through. Third, it modifies the notch width to accommodate tracking stability and optimize interference signal suppression to GNSS signal removal. The presented MPANF exhibits superior performance against chirp signals, including faster response to jump discontinuities.

Keywords

adaptive notch filter, ANF, FLL, FMCW, frequency-locked loop, frequency-modulated continuous wave, interference, LBCA, loop-bandwidth control algorithm, mitigation, MPANF, multi-parameter adaptive notch filter

1 | INTRODUCTION

Global navigation satellite system (GNSS) receivers are vulnerable to interference signals (Dovis, 2015). Frequency-modulated continuous waves (FMCWs) are common interference signals that transmit a constant signal but vary their instantaneous frequency over time. Chirp signals, also known as swept-frequency signals, are popular FMCW signals (Mitch et al., 2011) and are often used for privacy protection devices (PPDs; van der Merwe et al., 2018). Chirps change their center frequency linearly over time, then periodically jump back to the starting frequency. If a receiver mitigates an FMCW interference signal, it may regain robustness and reliability.

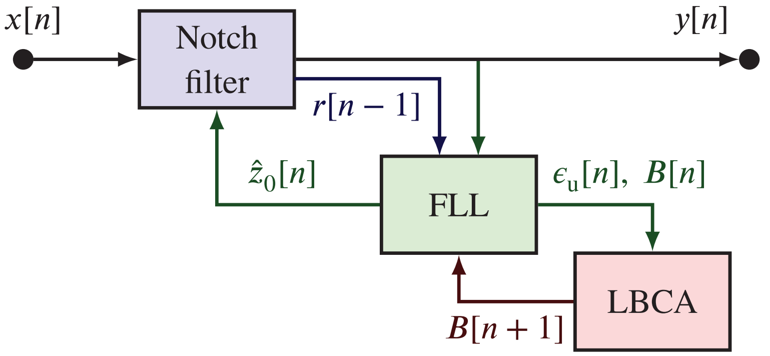

FIGURE 1 Architecture of the AFLL-ANF, with its three main parts: notch filter, FLL, the LBCA, and their interfaces

Adaptive notch filters (ANFs) have shown good interference mitigation capabilities, and they estimate the instantaneous frequency of an FMCW signal, then create a notch filter to surgically remove the interfering signal (Borio et al., 2014; Friedlander & Smith, 1984; Gamba & Falletti, 2018, 2019; Musumeci et al., 2016). The least-mean-squares (LMS; Wendel et al., 2016) or frequency-locked loop (FLL; Gamba & Falletti, 2018) methods are commonly used for adaption algorithms. The chirp dynamics and frequency jump discontinuities limit the adaption algorithm performance of the ANF. Fine tuning the adaptive algorithm can optimize the ANF performance for a given interference (Qin et al., 2019). However, it does not generalize the performance for the diversity of interference signals (van der Merwe et al., 2018) and requires reoptimization for each new signal. An improvement is to use the adaptive FLL ANF (AFLL-ANF; van der Merwe et al., 2021). It uses the loop-bandwidth control algorithm (LBCA) to adapt the loop bandwidth of an FLL. A critical tuning parameter of the FLL is the loop bandwidth, which is a trade-off between the agility and noise suppression of the FLL. The LBCA allows the adaptation algorithm to respond more agilely to jump discontinuities and improve tracking precision when the interference signal is stable. The architecture of the AFLL-ANF is shown in Figure 1.

The AFLL-ANF solves several issues with fixed ANFs (van der Merwe et al., 2021) and performs well in most cases. However, it does not outperform all static loops if correctly tuned for a specific interference (Qin et al., 2019). More complex ANF methods such as using a Kalman filter (Kang et al., 2018; Panchalard et al., 2006) or the cardinalized probability hypothesis density (CPHD) approach (Kim et al., 2019) further improve ANFs but are complex, require significant resources to implement, and restrict real-time operation. Therefore, there is a need for an improved AFLL-ANF, but not for excessive complexity.

This article proposes a multi-parameter adaptive notch filter (MPANF) that extends the AFLL-ANF. The MPANF adapts the loop bandwidth (agility-versus-precision trade-off), the notch width (filter suppression versus filter isolation), and notch depth (filter on versus pass-through). It uses three staggered notch filters to achieve superior mitigation capability. The primary goal is to improve the mitigation performance for different chirp types, including pulsed chirp signals. The secondary goal is to limit excessive resource requirements, which are approximately 2.5-times more complex than an AFLL-ANF. Figure 2 shows the architecture of the MPANF.

The contribution of this article is twofold. First, it presents a thorough analysis of the notch filter, including the characterization of the notch width and the transient

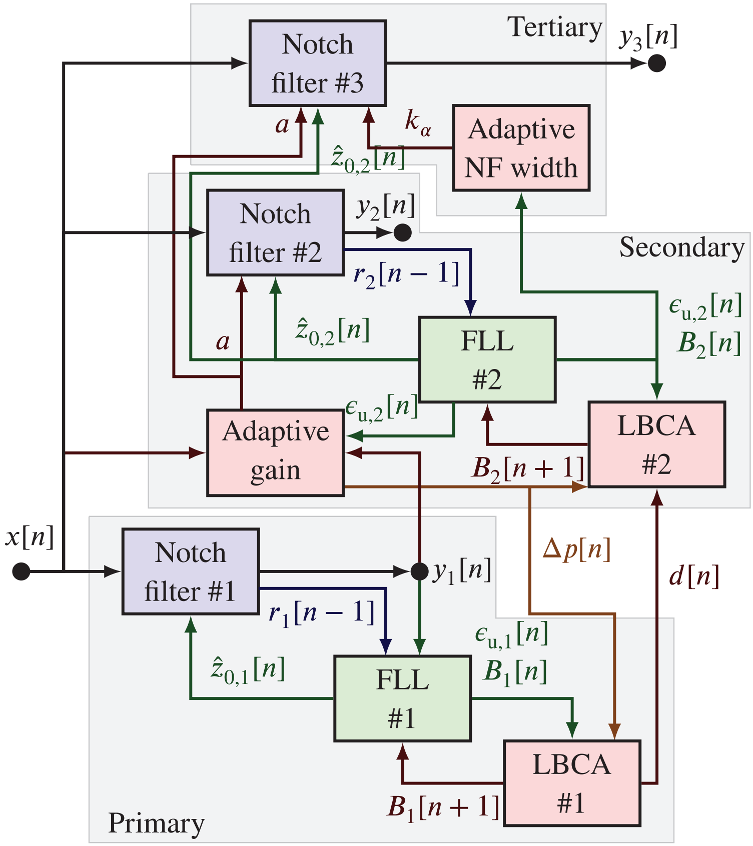

FIGURE 2 Architecture of the MPANF with its subcomponents and interfaces: The interfaces are indexed according to the filter stage (i.e., primary, secondary, or tertiary notch filter).

effects on the instantaneous transfer function in Section 5. Second, it presents approaches to adapt the notch width (Section 5) and notch depth (Section 4). To the authors' knowledge, such adaptation approaches have not yet been considered in the literature. Finally, the theoretical adaptation methods are compared with the AFLL-ANF and static ANFs (i.e., fixed loop bandwidth ANFs). The results show that the MPANF is superior to previous methods in various chirp signals. Furthermore, the methods are also tested against pulsed chirps, which increase the mitigation challenge as the ANF should be switched off if no interference is present, but it often results in tracking instabilities.

The rest of the article is structured as follows: Section 2 introduces conventional ANFs. The LBCA algorithm is described in Section 3. Section 4 demonstrates methods of dynamically switching the notch filter on and off. Section 5 analyzes the notch width and presents methods to steer it. To illustrate the different adaption techniques, Section 6 exhibits two case studies. Section 7 describes a Monte-Carlo simulation showing the results. Finally, Section 8 draws some conclusions and suggests future research.

2 | ADAPTIVE NOTCH FILTERS

This section presents the standard FLL-based ANF. The architecture is shown in Figure 3 with the two main components: the notch filter and the FLL.

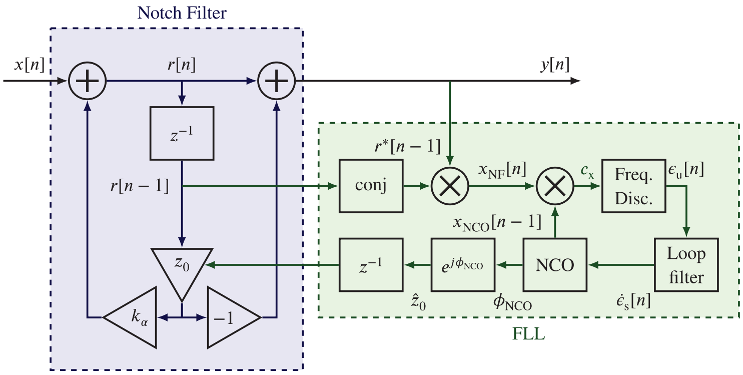

FIGURE 3 Flow diagram of the ANF with the notch filter and the FLL

2.1 | Notch Filter

A notch filter removes a narrow portion of the spectrum. Several implementation methods exist, but the most popular is a digital first-order infinite impulse response (IIR) filter (Dovis, 2015). This filter type is simple to implement, requires few resources, and has a deep notch characteristic compared with finite impulse response (FIR) filter implementations. However, it results in a nonlinear phase response and may become unstable. The transfer function, $H_0(z)$, for a digital first-order IIR ANF processor is (Borio et al., 2014):

$$H_0(z) = \frac{Y(z)}{X(z)} = \frac{1 - \hat{z}_0[n]z^{-1}}{1 - k_\alpha \hat{z}_0[n]z^{-1}} \tag{1}$$

where k_α is the pole contraction factor, and $\hat{z}_0[n]$ is a complex value that relates to the estimated notch frequency, such that:

$$\hat{z}_0[n] = a \cdot e^{2\pi \, j \cdot \hat{f}_0[n] T_s} \tag{2}$$

where $\hat{f}_0[n]$ is the instantaneous notch frequency, T_s is the sample interval of the digital system, and a is the amplitude of the complex estimate $\hat{z}_0$.

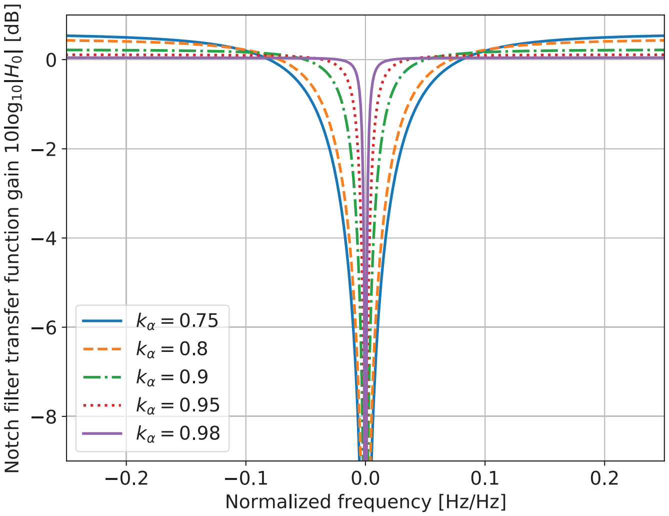

FIGURE 4 Magnitude of the transfer function for different k_α values with $a = 1$

The pole contraction factor, k_α, determines the width of the notch filter relative to the sample rate of the digital system. It must be in $k_\alpha \in [0, 1]$ to ensure filter stability (Borio et al., 2014). However, it is usually selected to be in $k_\alpha \in [0.7, 0.99]$ to limit the notch width (i.e., so that the notch is not too wide and ineffective) and for numerical stability (i.e., if the notch is too narrow, limited bit width in fixed-point signal processing may result in a value temporarily exceeding $k_\alpha > 1$). Figure 4 shows the magnitude response of the transfer function (Equation [1]) for different pole contraction factors, k_α and $\hat{z}_0[n] = 1$, and Figure 5 shows the phase response.

The amplitude a of the complex estimate $\hat{z}_0$ determines the notch depth. Only if $a = 1$ will the notch have a spectral null at the center frequency, $\hat{f}_0[n]$. If the amplitude a is zero, the notch filter reverts to an all-pass filter. Therefore, the amplitude a allows the filter to be switched on or off. Figure 6 shows the magnitude response of the transfer function for amplitudes a and $\hat{f}_0[n] = 0$, and Figure 7 shows the phase response.

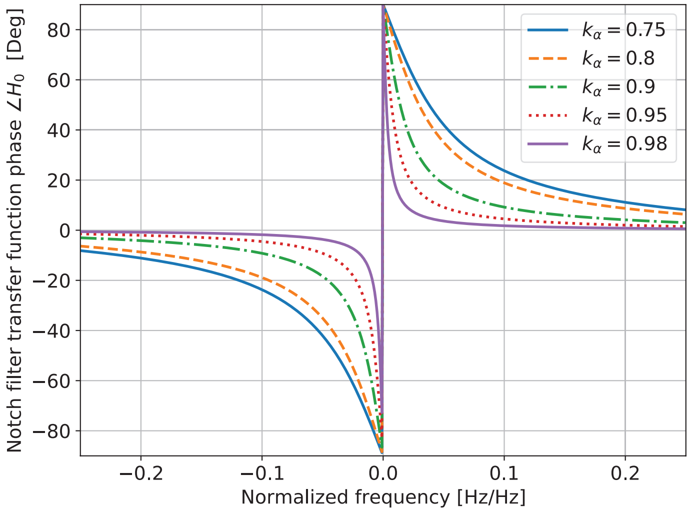

FIGURE 5 Phase of the transfer function for different k_α values with $a = 1$

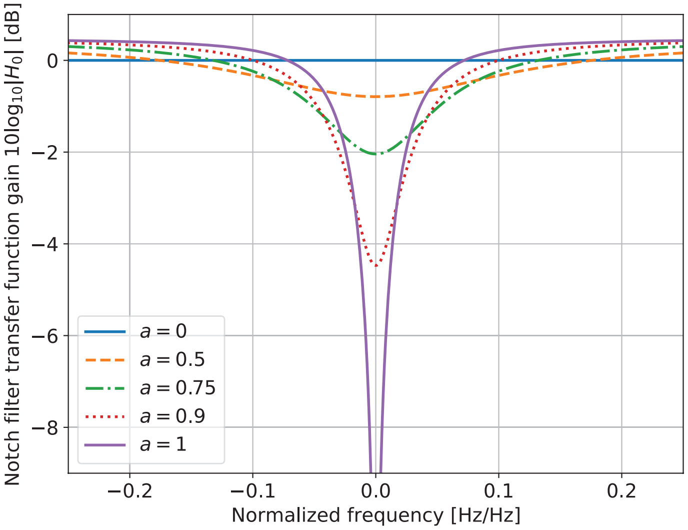

FIGURE 6 Magnitude of the transfer function for different a values with $k_\alpha = 0.8$

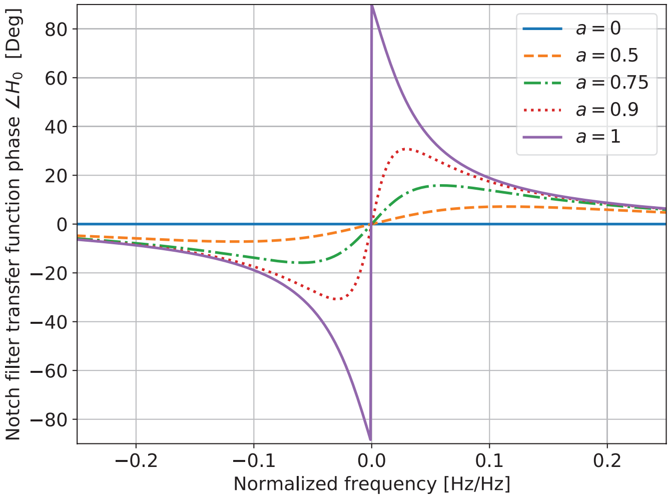

FIGURE 7 Phase of the transfer function for different a values with $k_\alpha = 0.8$

The adaptation algorithm alters the $\hat{z}_0[n]$ value for each sample offset n. The notch filter is separated into two parts to estimate the correct value for $\hat{z}_0[n]$: The feedback part is called the IIR or auto-regressive part, and the feed-forward part is called the FIR or moving average part (Borio et al., 2008). First, the IIR part is implemented with a feedback loop:

$$r[n] = x[n] + k_\alpha \hat{z}_0[n-1]r[n-1] \tag{3}$$

with $r[n]$ being the output of the IIR part. The IIR part adds the pole to the transfer function, $\boldsymbol{H}_0(z)$. The pole results in a gain increase around the estimated frequency $\hat{f}_0$, and improves the adaptation sensitivity near the current estimate. Therefore, this part is responsible for the width of the notch. Secondly, the FIR part is implemented with the feed-forward loop:

$$y[n] = r[n] - \hat{z}_0[n-1]r[n-1] \tag{4}$$

The FIR part adds a null to the transfer function, $\boldsymbol{H}_0(z)$. As such, this part controls the null location for the filter. The previous output of the IIR loop, $r[n-1]$, in conjunction with the complete filter output, $y[n]$, is used as the base metric, $x_{\mathrm{NF}}[n]$, to adapt the loop filter (Borio et al., 2008):

$$x_{\mathrm{NF}}[n] = r^*[n-1] \cdot y[n] \tag{5}$$

where $(\cdot)^*$ is the complex conjugate of the signal.

There are several adaptation algorithms, including LMS (Borio et al., 2008; Wendel et al., 2016), FLL (Gamba & Falletti, 2018), Kalman filtering (Kang et al., 2018; Panchalard et al., 2006), and any of these aided with machine learning (Abbasi et al., 2020). However, this article focuses on FLL-based methods, as they are simple to use with the LBCA. The LBCA also synergizes with Kalman filters (Cortés et al., 2022), which is an exciting topic for future research.

Finally, ANFs apply to frequency sparse signals like FMCW signals. However, other interference signal waveforms, e.g., band-limited noise, are not appropriate, requiring other mitigation techniques (Dovis, 2015; van der Merwe et al., 2021). Furthermore, the performance will degrade if two simultaneous signals are received (van der Merwe et al., 2020). This study is limited to chirp-like FMCW signals.

2.2 | Frequency-Locked Loop

An FLL has many applications (Talbot, 2012), but for GNSS, it is most known to estimate and remove the carrier in the signal tracking stage (Kaplan & Hegarty, 2017). FLLs are extensively used for ANFs (Gamba & Falletti, 2018) since they can track an FMCW signal's instantaneous frequency.

An FLL consists of four main parts: a correlator, the frequency discriminator, a loop filter, and a numerically controlled oscillator (NCO), as shown in Figure 3. The correlator compares the base metric, $x_{\mathrm{NF}}[n]$, representing the measured signal to an estimated signal, $x_{\mathrm{NCO}}[n]$:

$$c_{\mathrm{x}}[n] = x_{\mathrm{NF}}[n] \times x^{*}_{\mathrm{NCO}}[n-1] \tag{6}$$

where $c_{\mathrm{x}}[n]$ is the correlator output, $x_{\mathrm{NF}}[n]$ is the measured value from the notch filter, and $x_{\mathrm{NCO}}[n-1]$ is the previous estimate from the NCO. Next, the discriminator estimates the instantaneous frequency difference between the measured and received signals.

Several discriminator functions exist, but for the presented implementation, a four-quadrant arc-tangent frequency discriminator is used (Kaplan & Hegarty, 2017):

$$\Delta c_{\mathrm{x}}[n] = c^{*}_{\mathrm{x}}[n-1] \times c_{\mathrm{x}}[n] \tag{7}$$

$$\epsilon_{\mathrm{u}}[n] = \frac{f_s}{2\pi}\mathrm{atan2}(\Im\{\Delta c_{\mathrm{x}}[n]\}, \Re\{\Delta c_{\mathrm{x}}[n]\}) \tag{8}$$

where $\Delta c_{\mathrm{x}}[n]$ is the product of the input signal with its past conjugated, $\epsilon_{\mathrm{u}}[n]$ is the un-smoothed frequency error in Hertz relative to the sample frequency, f_s, atan2(·, ·) is the four-quadrant arc-tangent function, $\Re\{\cdot\}$ is the real operator, and $\Im\{\cdot\}$ is the imaginary operator.

The loop filter smooths out the frequency error by suppressing noise. It is implemented as an IIR digital filter and has many design parameters to consider. The filter order and the loop bandwidth, B, directly characterize the FLL performance. A low filter order tends to be more stable and has less overshooting but has limited performance in high dynamic changes, whereas a high filter order can accommodate increased dynamics but is more prone to overshooting and instability (Gardner, 2005). The filter order is selected as a fixed value during the design phase. A large loop bandwidth, B, results in an agile FLL with large errors and is suitable for fast-changing signals (Cortés et al., 2020). A small loop bandwidth, B, results in a slow response but improves noise suppression, which is more suitable for stationary and noisy signals. Altering the loop bandwidth facilitates a dynamics-to-noise trade-off and is discussed in more detail in the subsequent section. The loop filter is modeled as a simple linear filter:

$$\dot{\epsilon}_{\mathrm{s}}[n] = h_{\mathrm{LF}}[n] * \epsilon_{\mathrm{u}}[n] \tag{9}$$

where $\dot{\epsilon}_{\mathrm{s}}[n]$ is the smoothed frequency error rate, $h_{\mathrm{LF}}[n]$ is the impulse response of the filter, and $*$ denotes the convolution operation. For the final implementation, a standard second-order loop filter based on the bilinear transform is selected (Kaplan & Hegarty, 2017):

$$h_{\mathrm{LF}}[n] = \omega_0 \frac{\left(\frac{T_{\mathrm{s}}\omega_0}{2} + \sqrt{2}\right) + \left(\frac{T_{\mathrm{s}}\omega_0}{2} - \sqrt{2}\right) z^{-1}}{1 - z^{-1}} \tag{10}$$

where T_{s} is the sample time of the filter, ω_0 is the natural frequency of the filter and relates to the loop bandwidth (also referred to as the noise bandwidth) as:

$$B = 0.53 \cdot \omega_0 \tag{11}$$

where B is the loop bandwidth.

The NCO generates a replica signal and determines the instantaneous frequency of the loop. The NCO acts like an integrator:

$$\omega_{\mathrm{NCO}} = 2\pi T_{\mathrm{s}} \cdot \epsilon_{\mathrm{s}}[n] = 2\pi T_{\mathrm{s}} \cdot \left(\epsilon_{\mathrm{s}}[n-1] + \dot{\epsilon}_{\mathrm{s}}[n]\right) \tag{12}$$

where ϵ_{s} is the smoothed error related to the NCO frequency, ω_{NCO}. The estimated instantaneous frequency of the notch is determined based on the NCO output and can be used for the notch filter as described in Equation (2):

$$\hat{z}_0[n] = \mathrm{e}^{\,\mathrm{j}\omega_{\mathrm{NCO}}[n]} \tag{13}$$

Finally, the last integration step is needed in the NCO to predict the signal for the next cycle:

$$\phi_{\mathrm{NCO}}[n] = \phi_{\mathrm{NCO}}[n-1] + \omega_{\mathrm{NCO}}[n] \tag{14}$$

$$x_{\mathrm{NCO}}[n] = \mathrm{e}^{\,\mathrm{j}\phi_{\mathrm{NCO}}[n]} \tag{15}$$

where $\phi_{\mathrm{NCO}}[n]$ is the instantaneous phase of the NCO.

An alternative FLL ANF approach is the exponential discriminator (Gamba & Falletti, 2018), which smooths, stabilizes, and improves estimates. However, it introduces additional delays, which impede the agility of adaptive FLL methods. Hence, it is not considered for the remainder of the article.

2.3 | Adaptive Notch Filter Improvements

The ANF cannot immediately jump with the interference signal (except in post-processing, where it can be compensated for), resulting in a large portion of the interference signal bleeding through (Wendel et al., 2016). A method to counter this is to add a pulse blanker immediately after the ANF. It is effective, especially considering that this transition period is typically only a couple of microseconds (Borio, 2016; Wendel et al., 2016).

Multipole notch filters are computationally more complex but improve the notch suppression capabilities (Borio et al., 2008). Alternatively, a multi-stage approach may also be used to mitigate multiple signals, but it has several limitations (van der Merwe et al., 2020).

The ANF is known to degrade the signal if there is no interference or the interference-to-noise ratio (INR) is too low (Falletti et al., 2020). A solution is to include a detector to switch the ANF on or off (Borio et al., 2008; Falletti et al., 2020). However, an improved approach is presented in Section 4.

3 | LOOP BANDWIDTH CONTROL ALGORITHM

This article applies adaptive loop techniques deployed in adaptive scalar tracking loops (A-STLs) to alter the loop bandwidth of the FLL. The adaptive loop responds faster against jump discontinuities by temporarily increasing the loop bandwidth before settling back to a smaller loop bandwidth once it is locked on the chirp signal. In recent publications (Cortés et al., 2020; Cortés et al., 2020, 2021), the LBCA has shown promising results for robust tracking in GNSS receiver tracking loops. The LBCA was initially developed for phase-locked loops (PLLs), but it is extended to adapt an FLL in this article.

The AFLL-ANF combines the LBCA with the ANF for superior performance (van der Merwe et al., 2021). A similar suggestion to use the LBCA for ANF was made by Dimc et al. (2021). The LBCA for the AFLL-ANF additionally includes an early warning jump detector to detect and adapt to the frequency jump discontinuities more quickly.

3.1 | Standard LBCA

The LBCA (Cortés et al., 2020) adapts the loop bandwidth, B, based on the statistics of the discriminator output, $\epsilon_{\mathrm{u}}[n]$. The algorithm achieves this by combining sigmoid-based weighting functions of the normalized bandwidth B_{N}. Figure 8 shows the structure of the LBCA.

First, the absolute mean, $\left|\mu_{\epsilon}^{u}\right|$, and the standard deviation, σ_{ϵ}^{u}, of the discriminator output are estimated. The absolute mean, $\left|\mu_{\epsilon}^{u}\right|$, is interpreted as the dynamics or bias of the estimate and the standard deviation, σ_{ϵ}^{u}, as the estimation noise. Second, the normalized dynamics, $\tilde{D}$, are estimated as:

$$\tilde{D}[n] = \frac{\left|\mu_{\epsilon}^{u}[n]\right|}{\left|\mu_{\epsilon}^{u}[n]\right| + \sigma_{\epsilon}^{u}[n]} \tag{16}$$

Third, at the core of the LBCA, the normalized dynamics $\tilde{D}$ are combined with a weighting function, $g[n, B_{\mathrm{N}}]$. The difference between the normalized dynamics estimate, $\tilde{D}$, which is scaled by the maximum value of the weighting function, g_{Max}, and the weighting function, $g[n, B_{\mathrm{N}}]$, determines the update of the normalized bandwidth, B_{N}. This update divided by the sampling period represents the control signal, $c[n]$, in Hz:

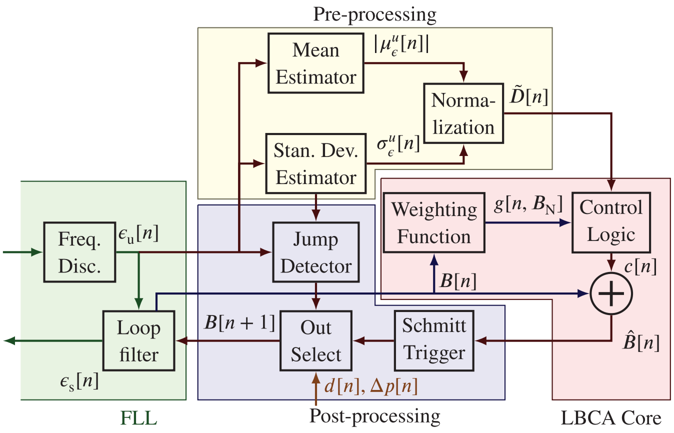

FIGURE 8 Architecture of the LBCA, including the preprocessing and postprocessing stages

$$c[n] = \left(g_{\text{Max}} \cdot \tilde{D}[n] - g[n, B_{\text{N}}]\right) / T_{\text{s}} \tag{17}$$

$$c[n] = \frac{g_{\text{Max}}}{T_{\text{s}}}\left(\tilde{D}[n] - \frac{g[n, B_{\text{N}}]}{g_{\text{Max}}}\right) \tag{18}$$

Finally, the addition of the control signal $c[n]$ into the current loop bandwidth $B[n]$ determines the estimated loop bandwidth, $\hat{B}[n]$:

$$\hat{B}[n] = B[n] + c[n] \tag{19}$$

The estimated loop bandwidth, $\hat{B}[n]$, goes through a Schmitt trigger to reduce possible switching instabilities.

$$B[n+1] = \begin{cases} \hat{B}[n] & \text{if} \left|\hat{B}[n] - B[n]\right| \geq T_{\text{schmitt}} \\ B[n] & \text{otherwise} \end{cases} \tag{20}$$

where T_{schmitt} is the Schmitt trigger threshold set to 10 kHz.

The weighting function, $g[n, B_{\text{N}}]$, directly specifies the adaptation performance of the LBCA. It is a linear combination of K normalized positive sigmoid functions:

$$g[n, B_{\text{N}}] = \sum_{k=1}^{K} w_k \text{Sig}_k\left(S_k\left(B_{\text{N}}[n] - P_k\right)\right) \tag{21}$$

$$= \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}^T \begin{bmatrix} \text{Sig}_1\left(S_1\left(B_{\text{N}}[n] - P_1\right)\right) \\ \vdots \\ \text{Sig}_K\left(S_K\left(B_{\text{N}}[n] - P_K\right)\right) \end{bmatrix} \tag{22}$$

where P_{k} is the shift parameter, S_{k} is the horizontal scaling, w_{k} is the vertical scaling, and $(\cdot)^T$ is the transpose. The sigmoid function, $\text{Sig}(x)$, is defined as (Domingos, 2015):

$$\text{Sig}(x) = \frac{1}{1 + e^{-x}} \tag{23}$$

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

$$g_{\text{Max}} = \sum_{k=1}^{K} w_k \tag{24}$$

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

$$\left|c[n]\right| \leq g_{\text{Max}} / T_{\text{s}} \tag{25}$$

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

Figure 9 shows the shape of the normalized weighting function used for the AFLL-ANF as a linear combination of two sigmoid functions and is defined as:

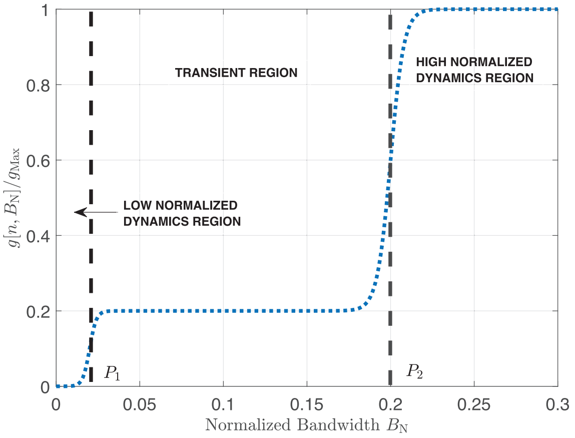

FIGURE 9 Normalized weighting function of the LBCA for the MPANF

$$g\left[n,B_{\mathrm{N}}\right]=\begin{bmatrix}w_1\\w_2\end{bmatrix}^T\begin{bmatrix}\mathrm{Sig}\left(S_1\left(B_{\mathrm{N}}-P_1\right)\right)\\\mathrm{Sig}\left(S_2\left(B_{\mathrm{N}}-P_2\right)\right)\end{bmatrix} \tag{26}$$

$$=\begin{bmatrix}0.002\\0.008\end{bmatrix}^T\begin{bmatrix}\mathrm{Sig}\left(500\left(B_{\mathrm{N}}-0.02\right)\right)\\\mathrm{Sig}\left(250\left(B_{\mathrm{N}}-0.2\right)\right)\end{bmatrix} \tag{27}$$

$$g_{\mathrm{Max}}=\sum_{n=1}^{2}w_n=0.01 \tag{28}$$

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

Finally, the additional inputs of the jump detector, $d[n]$, and power ratio, $\Delta p[n]$, in Figure 8 are external early-warning triggers and are presented in Sections 3.2 and 4, respectively.

3.2 | Jump Detector

A jump detector works as an early warning for a frequency jump discontinuity. The mean and standard deviation estimators of the LBCA are too slow for quick reaction, which motivates the early detection.

The jump detector takes an absolute mean of the current and previous discriminator outputs as a decision metric, $d[n]$:

$$d[n]=\left|\frac{\epsilon_{\mathrm{u}}[n]+\epsilon_{\mathrm{u}}[n-1]}{2}\right| \tag{29}$$

Using the last two values, it has a reaction delay of one sample. If the metric $d[n]$ is larger than three times the standard deviation of the discriminator, $\sigma_{\epsilon}^{u}[n]$ (i.e., less than 99% chance of randomly triggering), it forces the FLL to use the largest possible loop bandwidth, B_{Max}:

$$B\left[n+1\right]=\begin{cases}B_{\mathrm{Max}} & \text{if} \quad d\left[n\right]\geq\lambda_{\mathrm{d}}\\B\left[n+1\right] & \text{otherwise}\end{cases} \tag{30}$$

$$\lambda_{\mathrm{d}} = 3 \cdot \sigma_{\epsilon}^{u}[n] \tag{31}$$

This approach is simple yet has shown effective results (van der Merwe et al., 2021). Furthermore, the conservative threshold makes it improbable to trigger during noisy scenarios. It results in enhanced stability of the LBCA. Once the trigger occurs, the LBCA will adjust the loop bandwidth to an optimal value. A maximum bandwidth relating to the sample rate, f_{s}, is selected for the rest in this article:

$$B_{\mathrm{Max}} = \frac{f_{\mathrm{s}}}{4} = 5\,\mathrm{MHz} \tag{32}$$

4 | NOTCH FILTER DEPTH ADAPTATION

An ANF is capable of mitigating an interference signal, emphasizing its use. However, if no interference is present, it may disrupt the GNSS signal. Borio and Gioia (2021) demonstrated that an ANF causes significant pseudorange biases that motivate the need to bypass the ANF if no interference is present. Gamba and Falletti (2019) considered methods to detect when chirp interference is present and either uses an ANF to mitigate the interference or bypasses the ANF. This approach results in a hard switching between the mitigation and bypass that may lead to phase disruptions in the signal. The superior interference detection method is the power ratio (Gamba & Falletti, 2019). It uses the instantaneous power of the input signal, $p_x[n]$, to determine the average input signal power $\bar{p}_x[n]$:

$$p_x[n] = x[n]x^*[n] \qquad \bar{p}_x[n] = \frac{1}{K}\sum_{k=0}^{K-1} p_x[n-k] \tag{33}$$

where $x[n]$ is the complex input signal and K is the number of samples used to calculate the average power. Similarly, the instantaneous power, $p_y[n]$, of the output signal, $y_1[n]$ (i.e., after the primary ANF mitigation) is also averaged $\bar{p}_y[n]$:

$$p_y[n] = y_1[n]y_1^*[n] \qquad \bar{p}_y[n] = \frac{1}{K}\sum_{k=0}^{K-1} p_y[n-k] \tag{34}$$

Finally, the power ratio between the input power, $\bar{p}_x[n]$, and the output power, $\bar{p}_y[n]$, is determined:

$$\Delta p[n] = 10\log_{10}\left(\frac{\bar{p}_x[n]}{\bar{p}_y[n]}\right) \tag{35}$$

If the ANF successfully mitigated an FMCW signal, then the power ratio, $\Delta p[n]$, tends to be a large value. Conversely, if no interference is present, the power ratio, $\Delta p[n]$, tends to be zero. It forms the basis for FMCW detection (Gamba & Falletti, 2019).

This approach can be mapped to the notch depth, a, by extending it to a smooth transposition. A simple approach is to use a sigmoid function, defined by two thresholds, λ_1 and λ_2:

$$P_{\mathrm{p}} = \frac{\lambda_1 + \lambda_2}{2} \qquad S_{\mathrm{p}} = \frac{2\pi}{\lambda_2 - \lambda_1} \tag{36}$$

$$a'[n] = \mathrm{Sig}\left(S_\mathrm{p}\left(\Delta p[n] - P_\mathrm{p}\right)\right) \tag{37}$$

where P_p is the position bias of the sigmoid function, S_p is the sigmoid scaling value, and $a'[n]$ is the estimated gain. The thresholds for the MPANF are empirically tuned to be:

$$\lambda_1 = 1 \quad [\mathrm{dB}] \qquad \lambda_2 = 4 \quad [\mathrm{dB}] \tag{38}$$

Figure 10 shows the mapping of the power ratio, $\Delta p[n]$, to the estimated gain, $a'[n]$.

The issue with this approach is the delay of the mean power estimate. The first solution uses a first-order IIR filter to estimate the mean power to minimize the delay, similar to what is used in the LBCA (Cortés et al., 2020). In addition, an early detector must immediately switch the interference signal on when an interference signal is detected. Kurtosis is an excellent indicator for detecting changes (Caviedes & Gurbuz, 2002). Therefore, the kurtosis, $\kappa_\epsilon[n]$, of the unsmoothed error from ϵ_u suffices as an early detector to switch on the filter. The kurtosis, $\kappa_\epsilon[n]$, relates to the fourth-order statistical mean and requires the mean $\overline{\epsilon_\mathrm{u}}[n]$ and variance $\sigma_\epsilon^2[n]$:

$$\overline{\epsilon_\mathrm{u}}[n] = \mathrm{E}\left[\epsilon_\mathrm{u}[n]\right]_K = \frac{1}{K}\sum_{k=0}^{K-1}\epsilon_\mathrm{u}[n-k] \tag{39}$$

$$\sigma_\epsilon^2[n] = \mathrm{E}\left[\left(\epsilon_\mathrm{u}[n] - \overline{\epsilon_\mathrm{u}}[n]\right)^2\right]_K \tag{40}$$

$$= \mathrm{E}\left[\left(\epsilon_\mathrm{u}[n]\right)^2\right]_K - \overline{\epsilon_\mathrm{u}}^2[n] \tag{41}$$

$$\kappa_\epsilon[n] = \frac{\mathrm{E}\left[\left(\epsilon_\mathrm{u}[n] - \overline{\epsilon_\mathrm{u}}[n]\right)^4\right]_K}{\sigma_\epsilon^2[n]} \tag{42}$$

$$= \frac{\mathrm{E}\left[\epsilon_\mathrm{u}^4[n]\right]_K - \overline{\epsilon_\mathrm{u}}^4[n]}{\sigma_\epsilon^2[n]} \tag{43}$$

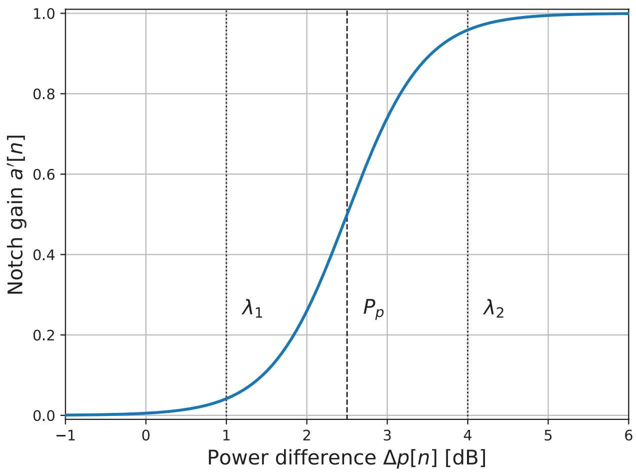

FIGURE 10 Power delta to notch filter gain mapping

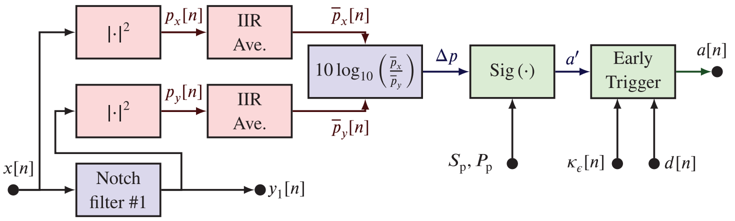

FIGURE 11 Adaptive gain flow diagram

The final notch filter gain, $a[n]$, is either the estimated gain, $a'[n]$, or forced to be on:

$$a[n] = \begin{cases} 1 & \text{if } \kappa_\epsilon[n] > \lambda_k \text{or } d[n] \geq \lambda_d \\ 1 & \text{if } a'[n] \geq 0.998 \\ a'[n] & \text{otherwise} \end{cases} \tag{44}$$

where λ_k is the kurtosis threshold, λ_d is the jump-detector threshold (Equation [31]), and $d[n]$ is the jump-detector decision metric (Equation [29]). The bootstrapping by 0.998 limits switch bouncing. This approach allows the ANF to immediately switch on if an interference signal is present but takes longer to switch off. The kurtosis threshold is selected as $\lambda_k = 2.75$ for the MPANF. Figure 11 shows the flow diagram to calculate the notch depth, a. The same notch depth a is used to adapt the secondary and the tertiary notch filters in Figure 2.

In Figure 2, a low-pass IIR filter is used instead of a mean operator (Equations [33] and [34]), for implementation efficiency.

Another significant issue with an absent interference is that the ANF with an LBCA tends to push the bandwidth down. This results in a slow response when the interference appears. However, the power ratio can be reused to force the loop bandwidth of the primary, $B_1[n+1]$, and secondary, $B_2[n+1]$, LBCA to its maximum value, B_{Max}:

$$B_1[n+1] = \begin{cases} B_{\text{Max}} & \text{if } d[n] \geq \lambda_d \text{or } \Delta p[n] < \lambda_R \\ B_1[n+1] & \text{otherwise} \end{cases} \tag{45}$$

$$B_2[n+1] = \begin{cases} B_{\text{Max}} & \text{if } d[n] \geq \lambda_d \text{or } \Delta p[n] < \lambda_R \\ B_2[n+1] & \text{otherwise} \end{cases} \tag{46}$$

where the reset threshold is set to $\lambda_R = 0 \quad [\text{dB}]$. This approach allows the MPANF to dynamically switch on and off depending on whether an interference is present or not.

5 | NOTCH FILTER WIDTH ADAPTATION

The width of the notch filter directly impacts mitigation capabilities. However, optimizing it is difficult, and previous research focused on parametric sweeps (Qin et al., 2019). This section aims to provide a quantitative analysis of the notch filter width and present methods to adapt the width.

5.1 | Ideal Notch Width

To understand the effect of the k_α on the notch filter width, let:

$$\hat{z}_0[n]=1 \qquad z^{-1}=e^{-2jx} \tag{47}$$

$$x=-\frac{\omega T_s}{2}=-\frac{\pi f_c}{f_s} \tag{48}$$

The transfer function is simplified (see Appendix A):

$$H_0=\frac{1-\hat{z}_0[n]z^{-1}}{1-k_\alpha\hat{z}_0[n]z^{-1}}=\frac{1}{\frac{k_\alpha+1}{2}-j\frac{1-k_\alpha}{2\tan(x)}} \tag{49}$$

The required power suppression, L, is defined as:

$$L=\left|\frac{1}{H_0}\right|^2 \tag{50}$$

Then the single-sided notch width is derived as (see Appendix B):

$$f_c=\frac{-f_s}{\pi}\tan^{-1}\left(\frac{1-k_\alpha}{\sqrt{4L-(k_\alpha+1)^2}}\right) \tag{51}$$

Therefore, if k_α is defined and the cutoff frequency f_c for a specific power loss L can be determined. Figure 12 shows the cutoff frequency relative to a 20-MHz sampled notch filter for $L=3$ dB and $L=6$ dB suppression. Usually, 3 dB is selected as a design point for digital filters (Ifeachor & Jervis, 2002), but for the remainder of this article, 6 dB is selected to enhance suppression capabilities.

Equation (51) is practical for analyzing the filter. However, it does not facilitate the steering of the pole-contraction factor, k_α. Therefore, it needs to be reformulated as a function of the cutoff frequency and power loss L (See Appendix C):

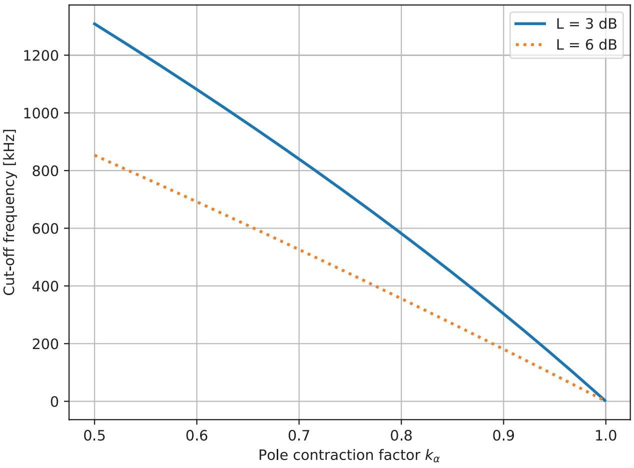

FIGURE 12 Cutoff frequency for a specified pole-contraction factor estimation error for a 20-MHz sampled notch filter

$$k_\alpha = \cos(2x) - \sqrt{4L\sin^2(x) - \sin^2(2x)} \tag{52}$$

This expression gives a theoretical function but is not practical for firmware systems due to the trigonometric functions and square root. Through a second-order Taylor approximation, the function is optimized and simplified for the constraints $k_\alpha \in [0.75, 1]$ and $x < 0.1$:

$$k_\alpha \approx 1 - \left(2\sqrt{L-1}\right)x - 2x^2 \tag{53}$$

The benefit is that the coefficient $\sqrt{L-1}$ can be precalculated, making it practical. Figure 13 shows the approximation (Equation [53]) compared with the theoretical value (Equation [52]) for a 20-MHz sampled notch filter. The difference is not visible, and Figure 14 shows the error. The approximation error is below 0.1% for the entire range, proving that it is an accurate approximation.

Finally, through back substitution:

$$k_\alpha \approx 1 - 2\sqrt{L-1}\left(\frac{\pi f_c}{f_s}\right) - 2\left(\frac{\pi f_c}{f_s}\right)^2 \tag{54}$$

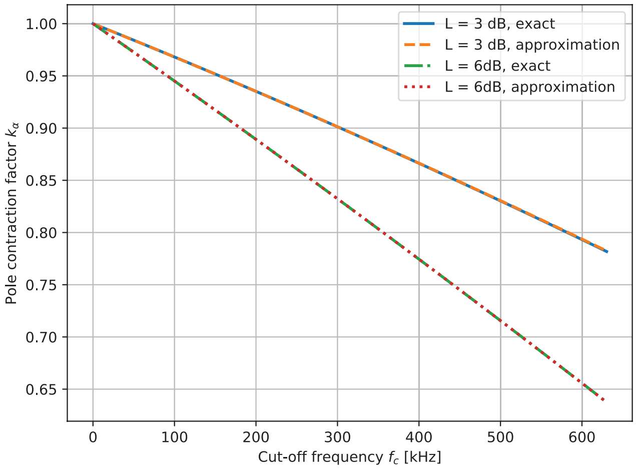

FIGURE 13 Pole-contraction factor estimation for a 20-MHz sampled notch filter

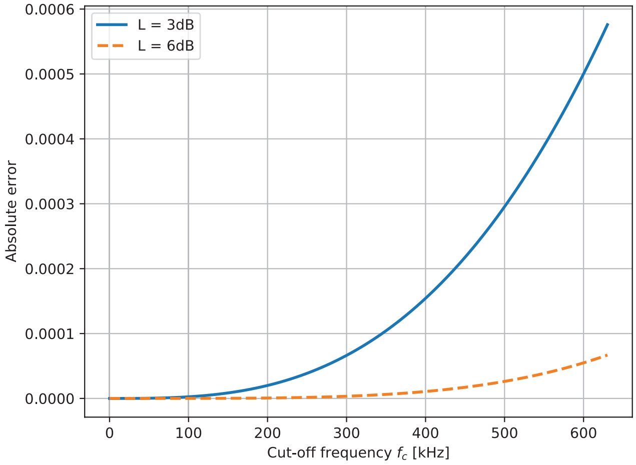

FIGURE 14 Pole-contraction factor estimation error for a 20-MHz sampled notch filter

5.2 | Notch Width Transient Effects

When two successive k_α values are not the same, the transfer is different (See Appendix D):

$$H_0(z) = \frac{1+(k_{\alpha 1}-1)\hat{z}_0\, z^{-1} - k_{\alpha 1}\,\hat{z}_0^2\, z^{-2}}{1-k_{\alpha 1}k_{\alpha 0}\,\hat{z}_0^2\, z^{-2}} \tag{55}$$

$$= \frac{\left(1-\hat{z}_0\, z^{-1}\right)\left(1+k_{\alpha 1}\hat{z}_0\, z^{-1}\right)}{\left(1-\sqrt{k_{\alpha 1}k_{\alpha 0}}\,\hat{z}_0\, z^{-1}\right)\left(1+\sqrt{k_{\alpha 1}k_{\alpha 0}}\,\hat{z}_0\, z^{-1}\right)} \tag{56}$$

where it is assumed that the notch frequency (i.e., $\hat{z}_0$) is constant, the current pole-contraction factor, $k_{\alpha 1}$, and the previous one, $k_{\alpha 0}$, are static. It makes several assumptions but still facilitates an analysis of how the notch filter behaves with an instantaneous change.

Figure 15 shows the zero-pole plot for Equation (56), assuming $\hat{z}_0 = 1$. It considers four cases:

- Static wide notch with $k_{\alpha 1} = k_{\alpha 0} = 0.7$ (red)
- Static narrow notch with $k_{\alpha 1} = k_{\alpha 0} = 0.95$ (blue)
- Notch closing from wide $k_{\alpha 0} = 0.7$ to narrow $k_{\alpha 1} = 0.95$ (green)
- Notch opening from narrow $k_{\alpha 0} = 0.95$ to wide $k_{\alpha 1} = 0.95$ (yellow)

As expected, the two static settings only have a single null and pole on the right side of the complex plane (note that all nulls at $+1$ overlap). However, the transitioning filter has an additional null and pole on the plane's left side. Further, both transitioning filters have poles, p_1 and p_2, that are mirrored and between the other two pole-contraction settings:

$$p_1 = -p_2 = \sqrt{k_{\alpha 1}k_{\alpha 0}} = \sqrt{0.95 \cdot 0.7} = 0.815 \tag{57}$$

However, where the two transitioning filters differ is the null placement. When the notch is closing (k_α increases), the null is on the outside of the pole. It causes a temporary second notch to form on the opposite side of the spectrum. Therefore, it potentially filters more of the signal out, but as it is a relatively narrow bandwidth, the impact on the gain is not as severe. When the notch is

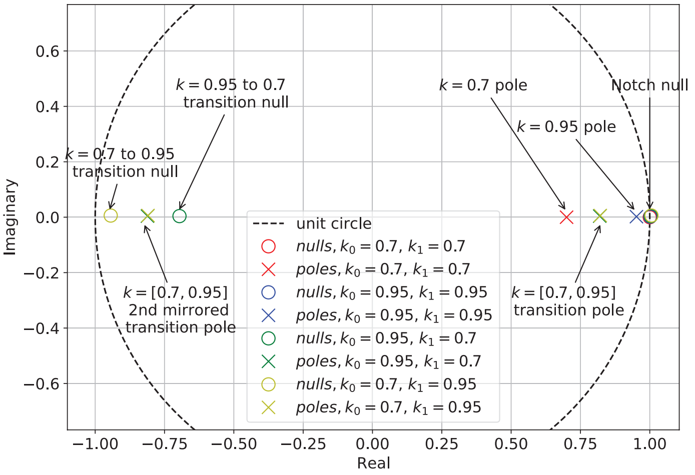

FIGURE 15 Zero-pole plot on the complex plane for different pole-contraction factor changes from k_0 to k_1 with $a = 1$

opening (k_α decreases), the null is on the inside of the pole. It causes a band-pass filter response on the opposite side of the filter, potentially resulting in instabilities. While testing the notch width steering, it was determined that this effect resulted in spikes in the output signal that caused severe degradation. Figure 16 confirms these effects with a magnitude spectral plot. Further, Figure 17 indicates that adverse phase effects also occur at the opposite points.

This analysis shows that several adverse effects occur when the pole-contraction factor, k_α, is changed. These effects are more problematic when the notch opens (i.e., k_α decreases) and need to be addressed in algorithm design. However, when the notch closes (i.e., k_α increases), the effects are negligible. Therefore, a strategy would be to dynamically close the notch but reset the notch filter when it is increased.

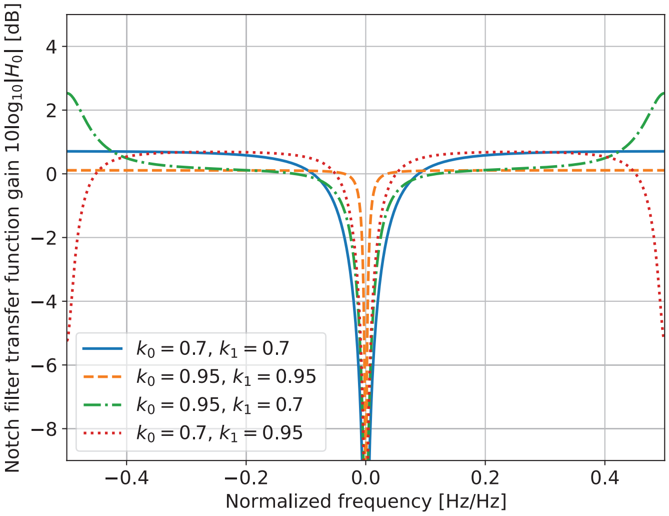

FIGURE 16 Magnitude of the transfer function for different pole-contraction factor changes from k_0 to k_1 with $a = 1$

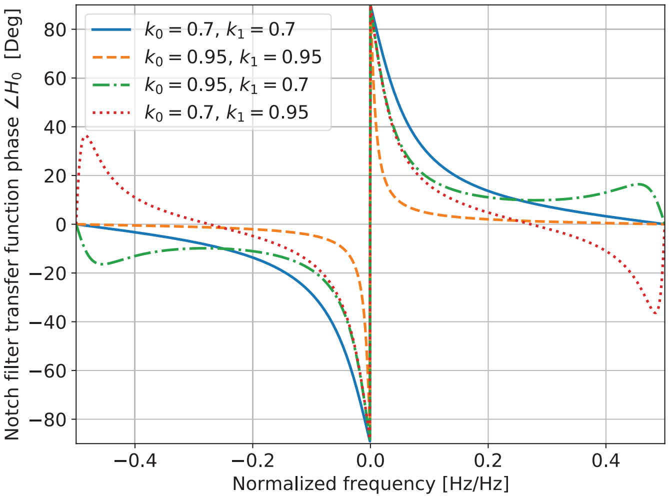

FIGURE 17 Phase of the transfer function for different pole-contraction factor changes from k_0 to k_1 with $a = 1$

5.3 | Notch Width Steering

Altering the notch width could provide some benefits to a notch filter. Ideally, the notch should be as wide as possible to ensure that the interference signal is completely removed but as narrow as possible to limit the spectral loss of the GNSS signals. Therefore, the required instantaneous interference bandwidth is needed for design. There are two factors to consider regarding the interference bandwidth. The first factor is the purity of the sine wave: Theoretically, the FMCW has an instantaneous frequency spectrum of a Dirac function. However, the frequency and amplitude variations change result in a broader impure instantaneous bandwidth. It can be characterized by tremendous effort, but a simple answer is to assume that suitable notch width is not available. It is considered outside the scope of the article and is a potential avenue for future research. The second factor is the estimation accuracy, which is the error between the estimated instantaneous frequency and the actual instantaneous frequency. The estimation accuracy, $\sigma_{\epsilon,s}$, is the accuracy of the NCO estimate and relates to the standard deviation, $\sigma^u_{\epsilon,2}$, of the secondary ANF (See Figure 2) discriminator output from the FLL (Betz & Kolodziejski, 2009). As the standard deviation, $\sigma^u_{\epsilon,2}$, of the discriminator output is already determined by the secondary LBCA, the value is reused:

$$\sigma^2_{\epsilon,s} = \frac{2B_2[n]}{f_\mathrm{s}} \cdot \left(\sigma^u_{\epsilon,2}\right)^2 \tag{58}$$

where the $B[n]/f_\mathrm{s}$ factor scales the noise to the output of the NCO, and the factor two translates the single-sided loop bandwidth, $B[n]$, to the two-sided bandwidth. An additional factor two is added as a safety margin that is empirically determined, and it compensates for the sine wave impurity:

$$\sigma^2_{\epsilon,b} = 2\sigma^2_{\epsilon,s} \tag{59}$$

The final estimation accuracy, $\sigma^2_{\epsilon,b}$, determines the theoretical approximated pole-contraction factor, k'_α, from Equation (54):

$$k'_\alpha = 1 - 2\sqrt{L-1}\left(\frac{\pi\sigma_{\epsilon,b}[n]}{f_s}\right) - 2\left(\frac{\pi\sigma_{\epsilon,b}[n]}{f_s}\right)^2 \tag{60}$$

An excellent operational choice for the suppression is $L = 4$ (i.e., 6 dB), and with the 20-MHz sample rate, it simplifies to:

$$k'_\alpha = 1 - 5.44\times10^{-7}\cdot\sigma_{\epsilon,b}[n] - 4.93\times10^{-14}\cdot\sigma^2_{\epsilon,b}[n] \tag{61}$$

If the pole-contraction factor, k_α, is too high, the filter can become unstable with numerical approximations as a pole is too close to the unit circle (e.g., see Figure 15). If it is too low, the notch is too wide, resulting in significant spectral distortion. Therefore, a limiter is added to ensure a lower bound for notch width and an upper bound for numerical stability:

$$k_{\alpha,e}[n] = \begin{cases} k'_\alpha & \text{if } 0.75 < k'_\alpha < 0.99 \text{ and } \sigma_{\epsilon,b}[n] \le \dfrac{f_s}{2} \\ 0.75 & \text{if } k'_\alpha \le 0.75 \text{ or } \sigma_{\epsilon,b}\left[n\right] \ge \dfrac{f_s}{2} \\ 0.99 & \text{if } k'_\alpha \ge 0.99 \end{cases} \tag{62}$$

where k'_α is the theoretical value and $k_{\alpha,e}\left[n\right]$ is the practical value. Excessive switching has some adverse effects as shown in Figures 16 and 17. Therefore, an asymmetric Schmitt trigger is included in determining the final pole-contraction factor:

$$k_\alpha\left[n+1\right]=\begin{cases}k_{\alpha,e}\left[n\right] & \text{if } k_\alpha\left[n\right]-k_{\alpha,e}\left[n\right]<-\lambda_{k,u}\\ k_{\alpha,e}\left[n\right] & \text{if } k_\alpha\left[n\right]-k_{\alpha,e}\left[n\right]>\lambda_{k,d}\\ k_\alpha\left[n\right] & \text{otherwise}\end{cases} \tag{63}$$

where the step-up threshold is selected as $\lambda_{k,u}=0.005$, and the step-down threshold is selected as $\lambda_{k,d}=0.1$. Further, to limit the instabilities highlighted in Figure 15, the tertiary notch filter is reset when the pole-contraction factor, $k_\alpha[n]$, decreases:

$$\left.\begin{array}{l}y_3[n]=0\\ r_3[n]=0\\ r_3[n-1]=0\end{array}\right\}\text{if } k_\alpha[n+1]<k_\alpha[n] \tag{64}$$

It creates a temporary zero in the output, but this relatively small price ensures filter stability. Further, the asymmetric Schmitt trigger limits excessive resets. Therefore, this approach is useful to alter the notch, despite several additional steps to ensure operational stability.

6 | CASE STUDIES

Two case studies are presented to demonstrate the behavior of the MPANF. A linear chirp signal with a 15-MHz chirp bandwidth and 20-us repetition rate is simulated at a 15-dB INR and 20-MHz sample rate. The chirp has the first jump discontinuity at timestamp 0 us, and the MPANF is initialized 1 ms prior. Figure 18 shows the spectrogram of the continuous chirp signal.

Figure 19 shows the notch frequency estimated from the primary FLL, the secondary FLL, and the actual instantaneous frequency simulated. Between −1 us and 0 us, both FLLs have precise tracking. The secondary FLL immediately responds to the jump discontinuity, as it uses the jump detector of the primary. It demonstrates the improved response time of the secondary FLL in the MPANF. After the jump

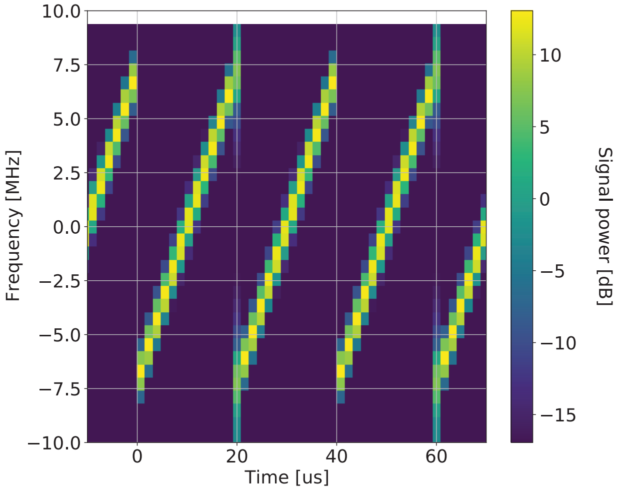

FIGURE 18 Spectrogram of a continuous chirp with 12-MHz bandwidth and 20-us repetition rate

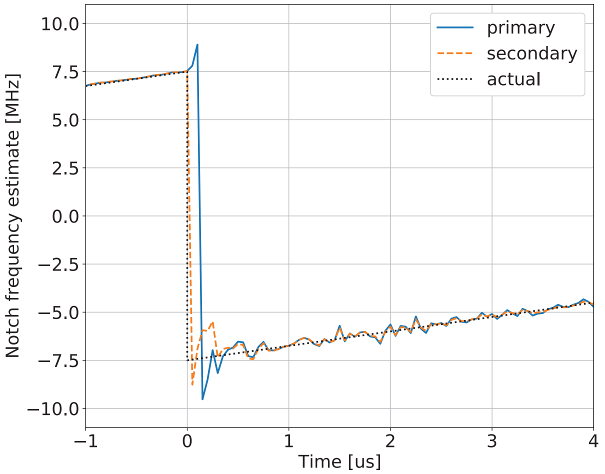

FIGURE 19 Frequency estimate of the primary and secondary FLLs compared with the instantaneous frequency for a continuous chirp

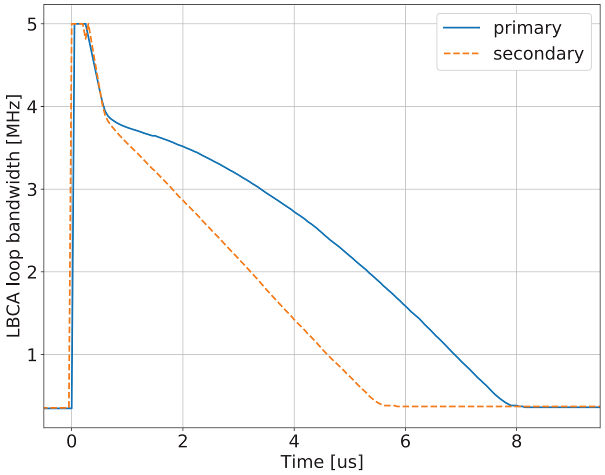

FIGURE 20 Loop bandwidth estimate for the primary and secondary LBCA with a continuous chirp

from 1 us to 4 us, both FLLs are near the correct frequency but noisier than before the discontinuity. It indicates that loop bandwidths are increased to facilitate a fast response but are not yet decreased for accurate tracking.

Figure 20 shows the loop bandwidths set by the primary and secondary LBCA. The secondary LBCA immediately increases to the maximum set bandwidth when the chirp jumps, whereas the primary is delayed with one sample. The jump detector and loop-bandwidth estimation predict the loop bandwidth for the following loop filter closure, causing one sample delay. Furthermore, as the secondary LBCA responds more quickly, the unsmoothed frequency error $\epsilon_{\mathrm{u}}[n]$ is reduced, resulting in a faster settling time. It is visible between 1 us to 8 us. The jump detector demonstrates that the early trigger from the primary to secondary LBCA significantly enhances overall tracking capability.

Figure 21 shows the notch depth (i.e., amplitude a) and notch width (i.e., pole-contraction factor k_α) steering. The notch depth is constantly and correctly set to the maximum value of $a = 1$. The notch width sets to the minimum value of $k_\alpha = 0.75$ each time the chirp has a jump discontinuity (0 us, 20 us, and 40 us). The wider notch is more effective when the frequency estimate is less precise and allows for a faster pull-in. The notch width decreases as the FLL pulls in and locks

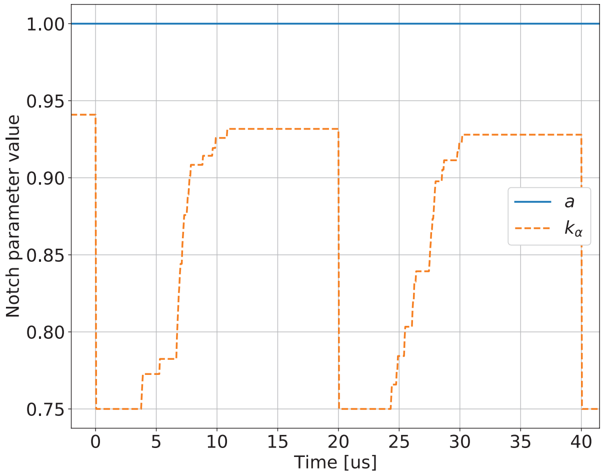

FIGURE 21 Amplitude and pole-contraction factor estimated with a continuous chirp

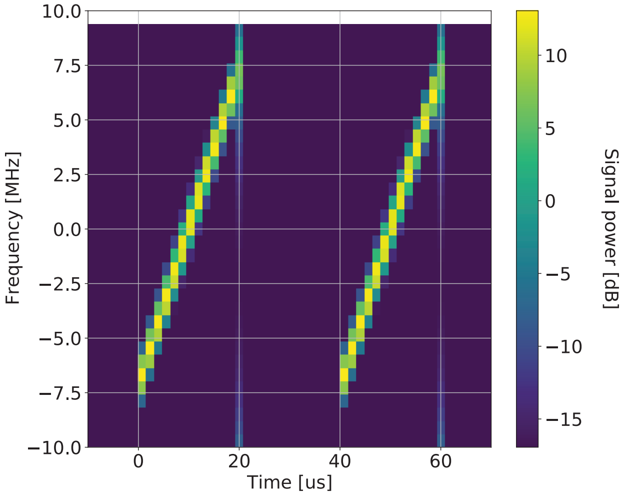

FIGURE 22 Spectrogram of a pulsed chirp with a 15-MHz bandwidth and 20-us repetition rate

onto the correct frequency and finally settles at a high value of $k_\alpha \approx 0.94$. In this example, the minimum notch width for $k_\alpha \approx 0.99$ is not achieved, showing that the notch width is noise limited by the FLL tracking. It demonstrates the notch filter ability to adapt to the current tracking precision.

The first case study illustrated the performance when a chirp is constantly present. Hence, the roles of the loop bandwidth and the notch width are clear, but the role of the notch depth is not. In the second case study, the chirp is pulsed. Pulsed chirp signals represent signals commonly used for radar systems (Bernfeld, 1984). However, such signals also represent cases when the chirp bandwidth exceeds the bandwidth of a front-end filter (Borio, 2018). The generated signal transmits a single chirp cycle of 20 us, then switched off for 20 us. Figure 22 shows the spectrogram of the pulsed chirp signal.

Figure 23 shows the frequency estimates for the primary and secondary FLLs. Between 0 us and 20 us, both accurately track the chirp signal. However, when the interference signal is switched off between 20 us and 40 us, both lose lock and have random estimates. Initially, both FLLs seem to track a signal between 20 us and 23 us, but it is caused by the delays of the internal IIR estimates and the current low loop bandwidth. However, at 23 us, the LBCA resets and forces the loop bandwidth to the highest value. Between 23 us and 40 us, there is no signal to track, and the FLL is aimlessly tracking noise, resulting in random frequency estimates.

Figure 24 presents the loop bandwidth estimates of the primary and secondary LBCAs. When the chirp is present (0 us to 20 us), both estimates start at the maximum bandwidth and decrease to the minimum bandwidth. Both have high bandwidths near the maximum when the chirp is absent (20 us to 40 us). There is a delay in response between 20 us to 23 us, as was observed with the frequency estimates. In the pulsed chirp case, the difference between the primary and secondary LBCAs estimates is negligible compared with the continuous chirp case.

Figure 25 shows the notch depth (i.e., amplitude a) and notch width (i.e., pole-contraction factor k_α) steering. It demonstrates detecting the interference and switching on and off automatically. The notch depth is at a maximum ($a = 1$) when the chirp is present and switched off ($a = 0$) when the interference is absent. The transition delay of switching the notch filter off is visible between 20 us and 23 us. The notch width follows a similar pattern to the continuous case when the chirp is present, but it is forced to the maximum width ($k_\alpha = 0.75$) when interference is not present.

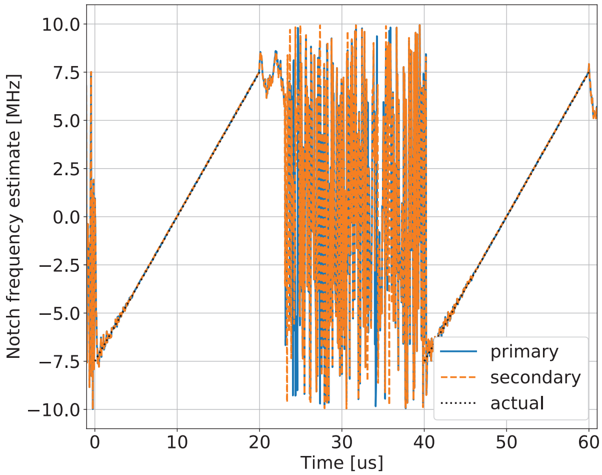

FIGURE 23 Frequency estimate of the primary and secondary FLLs compared with the actual instantaneous frequency for a pulsed chirp of Figure 22

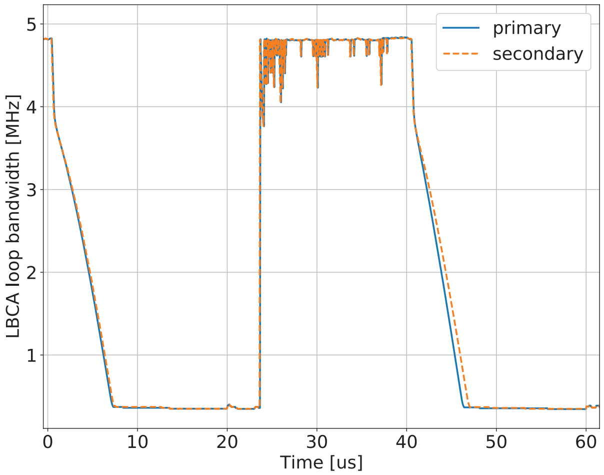

FIGURE 24 Loop bandwidth estimate for the primary and secondary LBCA with a pulsed chirp

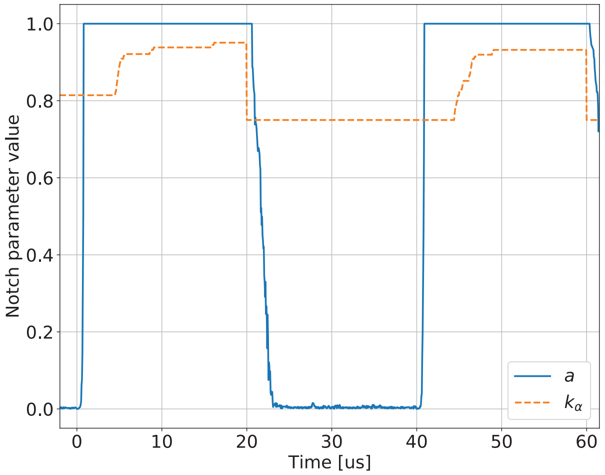

FIGURE 25 Amplitude and pole-contraction factor estimated with a continuous chirp

7 | MONTE-CARLO SIMULATION

7.1 | Simulation Setup and Design

The results focus on the impact on GNSS signal processing. In a previous study, the frequency estimation capabilities of the AFLL-ANF were investigated (van der Merwe et al., 2021) but will be omitted for the MPANF as similar results are expected. Figure 26 provides an overview of the simulation setup, consisting of three parts. First, the signal generation creates a GNSS signal, noise, and chirp interference at a sample rate of 20 MHz. The GNSS signal generator creates six GPS L1-C/A signals, and the noise generator ensures a carrier-to-noise density ratio (C/N_0) of 48 dBHz for all satellites. The gain of the chirp FMCW signal is scaled to achieve different INRs to facilitate comparison for different interference powers. The chirp signals are selected to have a bandwidth of 5 MHZ, which relates to a high interference impact with a jamming restive quality factor of $Q_\mathrm{j} = 5.05$ (Kaplan & Hegarty, 2017; van der Merwe et al., 2018) and has a high impact on the GNSS signals. The chirp rates, T, are varied for the tests and include:

$$T \in [10, 20, 50, 100, 200, 1000]\,\mathrm{us} \tag{65}$$

Second, the interference mitigation stage removes the chirp interference. It consists of an ANF to remove most of the signal and is followed by a pulse blanker to remove any vestigial interference components. Several ANFs filters are evaluated: the proposed MPANF (see Figure 2), the AFLL-ANF (van der Merwe et al., 2021; see Figure 1), and various static FLL-based ANFs with fixed loop bandwidth settings (see Figure 3). It facilitates a direct comparison of the adaptive FLL to fixed settings.

The fixed loop bandwidth settings are:

$$B \in [10, 100, 500, 800, 1000, 1500, 2000]\,\mathrm{kHz} \tag{66}$$

A pole-contraction factor of $k_\alpha = 0.9$ is selected for static ANFs and the AFLL-ANF. The pulse blanker is implemented as a memory-less pulse blanker with a conservative threshold related to the known variance of the noise generator:

$$y[n] = \begin{cases} 0 & \text{if} \quad |x[n]| \geq 3 \cdot \sigma_\mathrm{n} \\ x[n] & \text{otherwise} \end{cases} \tag{67}$$

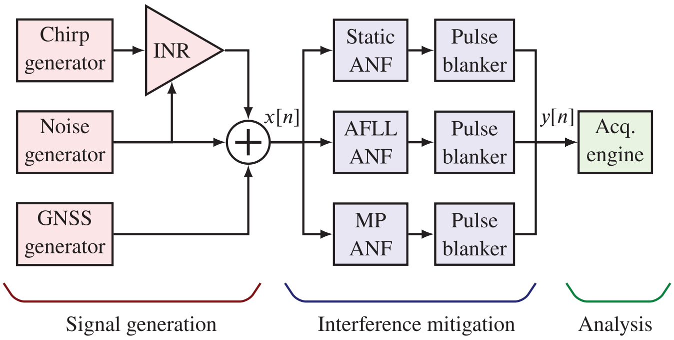

FIGURE 26 Flow diagram of the test setup

where $x[n]$ is the input signal, $y[n]$ is the output signal, and σ_n is the standard deviation of the noise generator signal.

Third, the results are analyzed and presented. The alpha mean, α_{mean}, from GNSS acquisition processing provides superior insight into GNSS performance with mitigation (Dovis, 2015). The acquisition engine uses 1 ms coherent integration and incoherently accumulates 10 epochs, resulting in 10 ms of integration gain. A Doppler search space of ±5 kHz with a Doppler resolution of 455 Hz is selected. The alpha mean, α_{mean}, is the ratio between the correlation power for the correct code offset and Doppler to the mean correlation power from the acquisition engine:

$$\alpha_{\mathrm{mean}} = 10 \cdot \log_{10}\left(\frac{x_c^2}{\frac{1}{NM}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} x^2\left[n,m\right]}\right) \tag{68}$$

where x_c^2 is the correlation power at the correct value and $x^2[n,m]$ is the correlation power for the n-th code-phase and m-th Doppler bin, N is the number of code phases tested, and M is the number of Doppler bins. The alpha mean, α_{mean}, provides a prediction of the GNSS processing capability of the signal. The higher the value, the more reliable acquisition is, as the probability of detection increases while that of false detection decreases. Negative values indicate that correct acquisition is not possible.

Other methods to evaluate the performance include the C/N_0 (van der Merwe et al., 2018) or the position accuracy (Borio & Gioia, 2021). However, these require longer data sets and more processing, making them more processing intensive. Such approaches are planned for future research.

A Monte-Carlo simulation evaluates the MPANF and AFLL-ANF to static ANFs. The simulation consists of 90 runs (relatively small for statistical analysis) for each INR setting. Each run does full acquisition processing for each of the six satellites, resulting in 540 α_{mean} measurements per INR setting.

A limitation of presenting all the results is that the plots quickly become cluttered. Therefore, secondary statistical analysis is performed. The maximum and the median for the static ANFs are determined:

$$\alpha_{\mathrm{mean}}^{(\mathrm{max})}(\mathrm{INR}) = \max_{B} \alpha_{\mathrm{mean}}(B,\mathrm{INR}) \tag{69}$$

$$\alpha_{\mathrm{mean}}^{(\mathrm{median})}(\mathrm{INR}) = \operatorname*{median}_{B} \alpha_{\mathrm{mean}}(B,\mathrm{INR}) \tag{70}$$

where $\alpha_{\mathrm{mean}}^{(\mathrm{max})}(\mathrm{INR})$ is the maximum alpha mean for all static ANFs for a given INR and represents the best performance, $\alpha_{\mathrm{mean}}^{(\mathrm{median})}(\mathrm{INR})$ is the median and represents the average performance for a poorly tuned FLL bandwidth, and B is the fixed loop bandwidth of the static FLLs (see Equation [66]). Next, the best-tuned static ANF is determined as the one with the smallest square loss relative to the maximum:

$$\Delta\alpha_{\text{mean}}(B,\text{INR}) = \alpha_{\text{mean}}(B,\text{INR}) - \alpha_{\text{mean}}^{(\max)}(\text{INR}) \tag{71}$$

$$B_{\text{opt}} = \arg\min_{B} \sum_{\text{INR}} \left(\Delta\alpha_{\text{mean}}(B,\text{INR})\right)^2 \tag{72}$$

where $\Delta\alpha_{\text{mean}}(B,\text{INR})$ is the delta in decibels to the maximum value, and B_{opt} is the optimal bandwidth from the finite selection of static bandwidths. The maximum, $\alpha_{\text{mean}}^{(\max)}(\text{INR};$ absolute best), median $\alpha_{\text{mean}}^{(\text{median})}(\text{INR};$ average), best static ANF $\alpha_{\text{mean}}(B_{\text{opt}}, \text{INR};$ best tuned), and no mitigation (worst performance) values are plotted to give a comparison for the MPANF.

7.2 | Results

7.2.1 | Selected Cases

Figure 27 shows the results for a continuous chirp with 50 us repetitions without a trailing pulse blanker. The ANFs have a peak at 10 dB but then degrade. This is contributed to the bleed-through after the chirp discontinuity, as shown by Wendel et al. (2016).

Figure 28 shows the same chirp but includes a trailing pulse blanker this time. In this case, the mitigation is significantly improved as it suppresses the signal bleed-through at the jump discontinuity and shows the remarkable synergy between an ANF and pulse blanker (PB). The MPANF has superior performance above 10 dB INR for this interference signal. The AFLL-ANF is second best and is better than static ANFs. It shows the benefit of the adaptive loop bandwidth approaches for such interference signals. All mitigation methods have a dip between −5 and 10 dB INR, which is the *switch-on effect*. Below these INRs, the chirp is too low-power to be tracked, and the ANFs only respond to noise; above, the chirp is high-power and clear to detect and track; but in-between, the ANFs struggle to detect and track the interference, resulting in performance

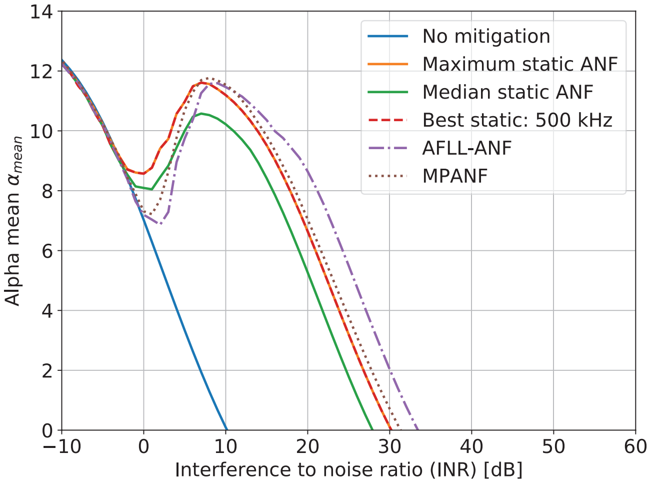

FIGURE 27 Alpha mean α_{mean} for a 50-us continuous chirp without pulse blanker

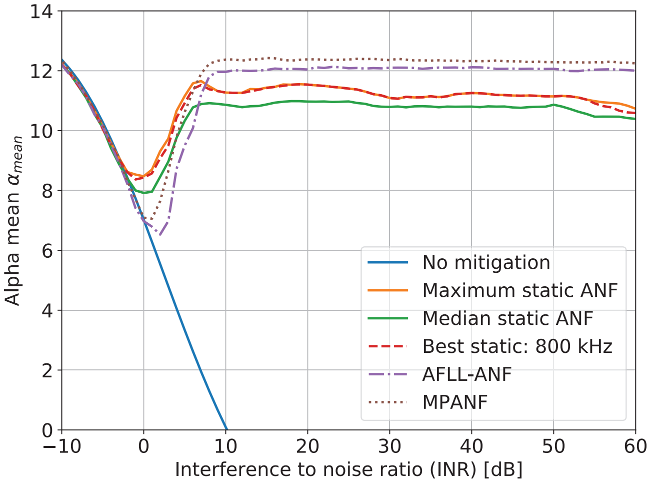

FIGURE 28 Alpha mean α_{mean} for a 50-us continuous chirp with pulse blanker

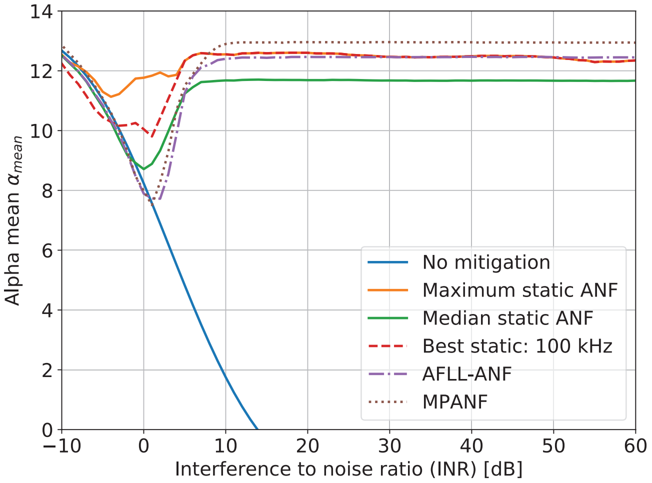

FIGURE 29 Alpha mean α_{mean} for a 1-ms continuous chirp with pulse blanker

degradation. The switch-on effect is most extreme for the MPANF and AFLL-ANF and is contributed to by the estimation complexity of these algorithms resulting in more susceptibility in partial tracking.

Figure 29 shows a much slower chirp with a repetition rate of 1 ms. Only the MPANF is superior in this scenario, and the AFLL-ANF performs similarly to the best static ANF. The best static setting is 100 kHz compared with 800 kHz in Figure 28, illustrating that the optimal performance depends on the scenario. It emphasizes the benefit of adaptive loop bandwidth methods to acclimate to diverse scenarios and alleviate tuning requirements. The switch-on effect is less severe for the slower chirp.

Figures 28 and 29 show that, for continuous chirps, the MPANF improves the AFLL-ANF, but only marginally. Hence, the benefit of the additional complexity for the MPANF is not evident for such signals.

Figure 30 shows the results for a pulsed chirp with 50-us repetitions with a pulse blanker. The MPANF shows superior performance, including a shallow dip during the switch-on stage. However, the AFLL-ANF has significantly reduced performance and is comparable to the median static ANF. It shows the limitation of the AFLL-ANF to a pulsed signal.

Figure 31 shows the results for a pulsed chirp with 1-ms repetition with a pulse blanker. Similar to Figure 30, the MPANF shows superior performance, and the AFLL-ANF severely degrades. Further, the difference between the best-tuned static ANF and the MPANF is smaller, as observed in Figures 28 and 29.

Additional results are available in Appendix E for the interested reader.

7.2.2 | MPANF Compared With Maximum Static

Figures 27 to 30 only show selected cases. However, performance analysis over a larger range of chirp repetitions is required.

Figure 32 shows the difference between the MPANF to the maximum static ANF for continuous chirps. In the legend, the mean, μ, in decibels also determines the average difference with the static max ANF for INRs ≥ 10 dB. The loss during the switch-on phase is below 5 dB. Most chirps improve for INR values above 10 dB; the only exception is the 20-us chirp. A hypothesis is that the LBCA adaption rate may oscillate, resulting in reduced performance. Fine tuning the weighting function, $g[n, B_N]$, may solve this issue and is suggested as a future investigation.

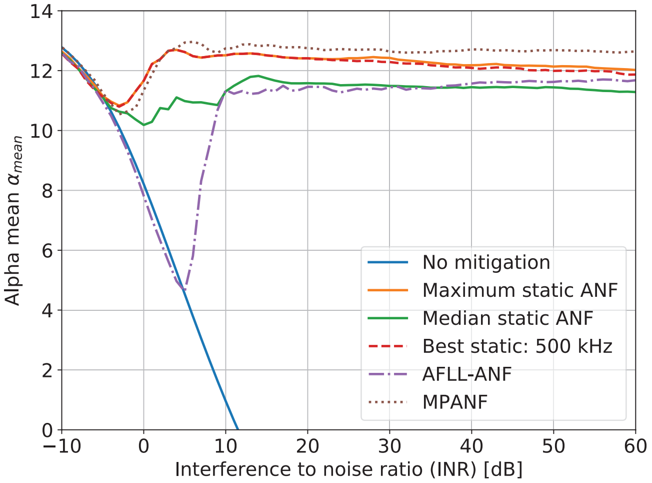

FIGURE 30 Alpha mean α_{mean} for a 50-us-pulsed chirp with pulse blanker

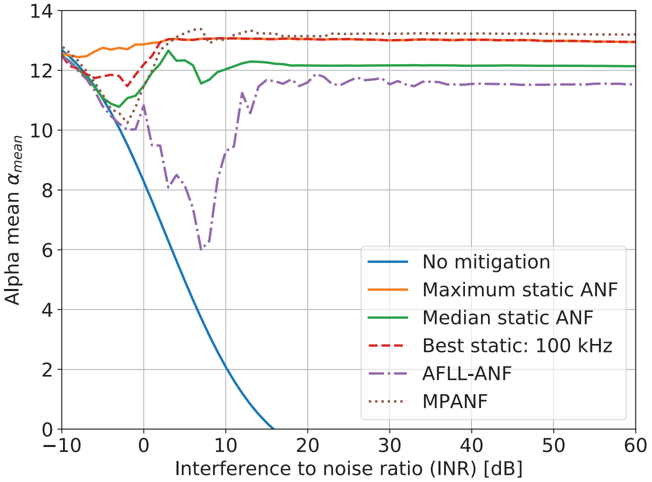

FIGURE 31 Alpha mean α_{mean} for 1-ms-pulsed chirp with pulse blanker

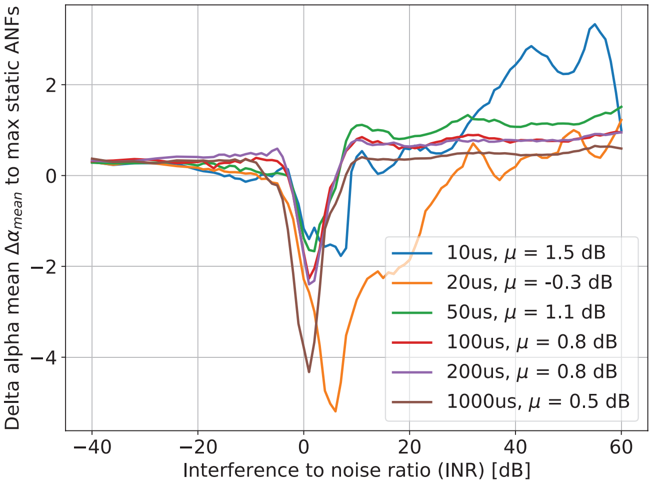

FIGURE 32 Delta alpha mean, $\Delta\alpha_{\text{mean}}$, for continuous chirps with pulse blanker for the MPANF

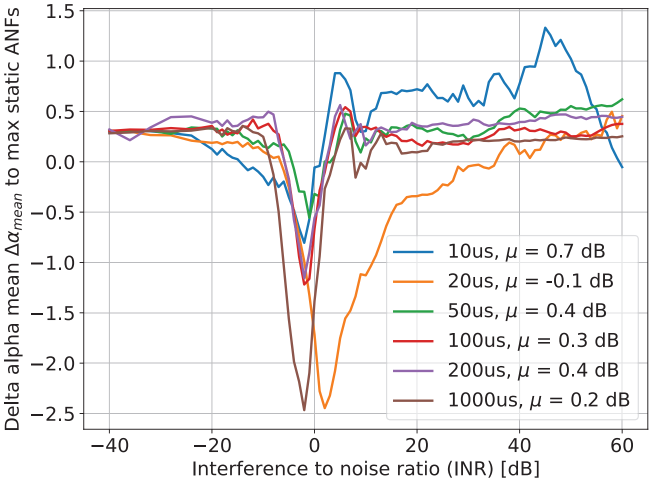

FIGURE 33 Delta alpha mean, $\Delta\alpha_{\text{mean}}$, for pulsed chirps with pulse blanker for the MPANF

The average delta alpha mean, μ, ranges from −0.3 to 1.5 dB, indicating that the MPANF provides superior performance for most chirps compared with the maximum achievable performance with static ANFs.

Figure 33 shows the difference between the MPANF and the maximum static ANF for pulsed chirps. Similar observations to Figure 32 are made. However, the delta alpha mean, $\Delta\alpha_{\text{mean}}$, has a smaller absolute range and mean range.

7.2.3 | Summarized Performance

Table 1 summarizes the mean performance of the MPANF, AFLL-ANF, median static ANF, and the best static ANF for every continuous chirp evaluated. These values are determined through the same process as Figures 32 and 33. The last row shows the average over all chirps. Similarly, Table 2 shows the performance of the pulsed chirps. In both tables, the MPANF has an average increase exceeding 0.3 dB compared with the maximum. As it is the only ANF that increases in both scenarios, it emphasizes its performance benefit to various chirp types and conditions. The AFLL-ANF increases with 0.22 dB for continuous chirps but decreases

TABLE 1
Performance Summary and Comparison With Continuous Chirps With Pulse Blanker

Rep. T [us]	MPANF μ [dB]	AFLL μ [dB]	median μ [dB]	Best μ [dB]	B_{opt} [kHz]
10	+1.5	+0.9	−5.1	−0.3	2000
20	−0.3	−1.4	−0.8	−0.0	1500
50	+1.1	+0.8	−0.4	−0.0	800
100	+0.8	+0.6	−0.5	−0.0	500
200	+0.8	+0.4	−0.7	−0.0	500
1000	−0.5	−0.0	−0.8	−0.0	100
Av.	+0.73	+0.22	−1.38	−0.05	—

TABLE 2
Performance Summary and Comparison With Pulsed Chirps With Pulse Blanker

Rep. T [us]	MPANF μ [dB]	AFLL μ [dB]	median μ [dB]	Best μ [dB]	B_{opt} [kHz]
10	+0.7	−0.4	−1.3	−0.4	2000
20	−0.1	−0.3	−0.6	−0.1	1000
50	+0.4	−0.8	−0.8	−0.1	500
100	+0.3	−1.6	−0.9	−0.0	500
200	+0.4	−1.5	−1.1	−0.0	500
1000	+0.2	−1.6	−0.9	−0.0	100
Av.	+0.32	−1.03	−0.93	−0.1	—

with 1.03 dB for pulsed chirps, reaffirming the benefit only to certain interference signal types. All other ANFs have average losses, emphasizing the performance of the adaptive loop bandwidth techniques.

Tables 1 and 2 show that the MPANF is the superior ANF notch filter, even outperforming all static notch filters. However, the improvement over the maximum alpha mean is less than 1.5 dB for all cases. After that, the performance-to-complexity trade-off is questionable: Significant complexity is added for marginal benefits. However, the best-tuned static ANF does not always yield optimal performance, indicating that circumventing manual fine tuning already justifies the additional complexity. Nevertheless, practical application and hardware implementation of the MPANF are cardinal for future research.

Further, when comparing the results of Figures 27 and 28, it is clear that the best value for performance is the inclusion of a pulse blanker after the notch filter. Thereafter, design considerations between static ANF, AFLL-ANF, and MPANF architectures can be considered, with the MPANF exhibiting the best performance. However, these architectures are still significantly simpler than a Kalman filter (Kang et al., 2018; Panchalard et al., 2006) or CPHD approaches (Kim et al., 2019).

8 | CONCLUSION

An MPANF architecture was presented in this article that adapts the notch loop bandwidth, notch width, and notch depth of an ANF. It uses three consecutive notch filters to estimate the parameters.

In Monte-Carlo simulations, the MPANF and other ANFs were evaluated against continuous and pulsed chirps with different chirp repetition rates. The results show that the MPANF outperforms the best-tuned static ANFs, the best combination of static ANFs, and the AFLL-ANF. Furthermore, it also performs well against pulsed chirps, compared with the degradation in the AFLL-ANF.

This article presented the first design and software implementation of the MPANF. Future research includes investigating practical application and hardware implementation capabilities. Furthermore, evaluation against other types of FMCW signals is also proposed to determine the limits of this architecture. Lastly, several optimizations of the architecture can also be considered, e.g., improving the degradation during the switch-on stage.

AUTHOR CONTRIBUTIONS

Conceptualization, J.R.M. and I.C.; methodology, J.R.M.; software, J.R.M. and I.C.; validation, J.R.M., I.C., F.G., and A.R.; formal analysis, J.R.M. and I.C.; investigation, J.R.M. and I.C.; resources, F.G. and A.R.; data curation, J.R.M.; writing–original draft preparation, J.R.M. and I.C.; writing–review and editing, J.R.M., I.C., F.G., and A.R.; visualization, J.R.M. and I.C.; supervision, F.G.; project administration, A.R.; and funding acquisition, W.F. and A.R.

CONFLICT OF INTEREST

The authors declare no potential conflicts of interest.

REFERENCES

Abbasi, M., Mosavi, M. R., & Reazei, M. J. (2020, September). GPS continues wave jamming canceller using an ANF combined with an artificial neural network. *2020 8th Iranian Joint Congress on Fuzzy and intelligent Systems (CFIS)*, Mashhas, Iran. https://doi.org/10.1109/CFIS49607.2020.9238700

Bernfeld, M. (1984, April). Chirp Doppler radar. *Proceedings of the IEEE*, *72*(4), 540–541. https://doi.org/10.1109/PROC.1984.12890

Betz, J. W., & Kolodziejski, K. R. (2009, October). Generalized theory of code tracking with an early-late discriminator part I: lower bound and coherent processing. *IEEE Transactions on Aerospace and Electronic Systems*, *45*(4), 1538–1556. https://doi.org/10.1109/TAES.2009.5310316

Borio, D. (2016). Swept GNSS jamming mitigation through pulse blanking. *2016 European Navigation Conference (ENC)*, Helsinki, Finland. https://doi.org/10.1109/EURONAV.2016.7530549

Borio, D. (2018, May). Sub-band robust GNSS signal processing for jamming mitigation. *2018 European Navigation Conference (ENC)*, Gothenburg, Sweden. https://doi.org/10.1109/EURONAV.2018.8433250

Borio, D., Camoriano, L., & Presti, L. L. (2008). Two-pole and multi-pole notch filters: a computationally effective solution for GNSS interference detection and mitigation. *IEEE Systems Journal*, *2*(1), 38–47. https://doi.org/10.1109/JSYST.2007.914780

Borio, D., & Gioia, C. (2021, March). GNSS interference mitigation: a measurement and position domain assessment. *NAVIGATION*, *68*(1), 93–114. https://doi.org/10.1002/navi.391

Borio, D., O'Driscoll, C., & Fortuny, J. (2014). Tracking and mitigating a jamming signal with an adaptive notch filter. *Inside GNSS*, *9*(2), 67–73. https://www.insidegnss.com/auto/marapr14-WP.pdf

Caviedes, J., & Gurbuz, S. (2002, December). No-reference sharpness metric based on local edge kurtosis. *Proc. of the International Conference on Image Processing*, Rochester, NY. https://doi.org/10.1109/ICIP.2002.1038901

Cortés, I., Iñiguez de Gordoa, J. A., van der Merwe, J. R., Rügamer, A., & Felber, W. (2020, June). Performance and complexity comparison of adaptive loop-bandwidth tracking techniques. *2020 International Conference on Localization and GNSS (ICL-GNSS)*, Tampere, Finland. https://doi.org/10.1109/ICL-GNSS49876.2020.9115543

Cortés, I., van der Merwe, J. R., Lohan, E. S., Nurmi, J., & Felber, W. (2022). Performance evaluation of adaptive tracking techniques with direct-state Kalman filter. *Sensors*, *22*(2), 420. https://doi.org/10.3390/s22020420

Cortés, I., van der Merwe, J. R., Nurmi, J., Rügamer, A., & Felber, W. (2021). Evaluation of adaptive loop-bandwidth tracking techniques in GNSS receivers. *Sensors*, *21*(2), 502. https://doi.org/10.3390/s21020502

Cortés, I., van der Merwe, J. R., Rügamer, A., & Felber, W. (2020). Adaptive loop-bandwidth control algorithm for scalar tracking loops. *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Portland, OR. https://doi.org/10.1109/PLANS46316.2020.9110182

Dimc, F., Pavlovčič-Prešeren, P., & Bažec, M. (2021). Robustness against chirp signal interference of on-board vehicle geodetic and low-cost GNSS receivers. *Sensors*, *21*(16), 5257. https://doi.org/10.3390/s21165257

Domingos, P. (2015). *The master algorithm: how the quest for the ultimate learning machine will remake our world*. Basic Books, Inc.

Dovis, F. (2015). *GNSS interference threats and countermeasures*. Artech House.

Falletti, E., Gamba, M. T., & Pini, M. (2020). Design and analysis of activation strategies for adaptive notch filters to suppress GNSS jamming. *IEEE Transactions on Aerospace and Electronic Systems*, *56*(5), 3718–3734. https://doi.org/10.1109/TAES.2020.2982301

Friedlander, B., & Smith, J. (1984, March). Analysis and performance evaluation of an adaptive notch filter. *IEEE Transactions on Information Theory*, *30*(2), 283–295. https://doi.org/10.1109/TIT.1984.1056887

Gamba, M. T., & Falletti, E. (2018, December). Performance analysis of FLL schemes to track swept jammers in an adaptive notch filter. *2018 9th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Noordwijk, Netherlands. https://doi.org/10.1109/NAVITEC.2018.8642663

Gamba, M. T., & Falletti, E. (2019). Performance comparison of FLL adaptive notch filters to counter GNSS jamming. *2019 International Conference on Localization and GNSS (ICL-GNSS)*, Nuremberg, Germany. https://doi.org/10.1109/ICL-GNSS.2019.8752751

Gardner, F. (2005). *Phaselock techniques* (3rd Ed.). Wiley.

Ifeachor, E. C., & Jervis, B. W. (2002). *Digital signal processing: a practical approach*. Prentice Hall.

Kang, C. H., Kim, S. Y., & Park, C. G. (2018). An adaptive Kalman filtering approach to Fourier analysis for estimating various chirp-type GNSS interference frequencies. *NAVIGATION*, *65*(1), 3–13. https://doi.org/10.1002/navi.217

Kaplan, E. D., & Hegarty, C. J. (2017). *Understanding GPS: Principles and applications* (3rd ed.). Artech House.

Kim, S. Y., Kang, C. H., & Park, C. G. (2019). Frequency tracking and mitigation method based on CPHD filter and adaptive multiple linear Kalman notch filter for multiple GNSS interference. *NAVIGATION*, *66*(4), 803–830. https://doi.org/10.1002/navi.332

Mitch, R. H., Dougherty, R. C., Psiaki, M. L., Powell, S. P., O'Hanlon, B.W., Bhatti, J. A., & Humphreys, T. E. (2011, September). Signal characteristics of civil GPS jammers. *Proc. of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2011),* Portland, OR, 1907–1919. https://www.ion.org/publications/abstract.cfm?articleID=9740

Musumeci, L., Curran, J. T., & Dovis, F. (2016). A comparative analysis of adaptive notch filtering and wavelet mitigation against jammers interference. *NAVIGATION*, *63*(4), 533–550. https://doi.org/10.1002/navi.167

Panchalard, R., Koseeyaporn, J., & Wardkein, P. (2006, June). State-space Kalman adaptive IIR notch filter. *2006 International Conference on Communications, Circuits and Systems*, Guilin, China. https://doi.org/10.1109/ICCCAS.2006.284619

Qin, W., Dovis, F., Gamba, M. T., & Falletti, E. (2019). A comparison of optimized mitigation techniques for swept-frequency jammers. *Proc. of the 2019 International Technical Meeting of the Institute of Navigation*, Reston, VA, 233–247. https://doi.org/10.33012/2019.16691

Talbot, D. B. (2012). *Frequency acquisition techniques for phase locked loops*. Wiley. https://doi.org/10.1002/9781118383285

van der Merwe, J. R., Cortés Vidal, I., Garzia, F., Lohan, E. S., Nurmi, J., & Felber, W. (2021, November). Resilient interference mitigation with adaptive frequency locked loop based adaptive notch filtering. *2021 Navigation,* Edinburgh, United Kingdom. https://www.researchgate.net/publication/356254217_Resilient_Interference_Mitigation_with_Adaptive_Frequency_Locked_Loop_based_Adaptive_Notch_Filtering

van der Merwe, J. R., Garzia, F., Rügamer, A., Cortés Vidal, I., & Felber, W. (2020, November). Adaptive notch filtering against complex interference scenarios. *2020 European Navigation Conference (ENC)*, Dresden, Germany. https://doi.org/10.23919/ENC48637.2020.9317518

van der Merwe, J. R., Rügamer, A., Garzia, F., Felber, W., & Wendel, J. (2018, April). Evaluation of mitigation methods against COTS PPDs. *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Monterey, CA. https://doi.org/10.1109/PLANS.2018.8373470

van der Merwe, J. R., Garzia, F., Rügamer, A., & Felber, W. (2021). Advanced and versatile signal conditioning for GNSS receivers using the high-rate DFT-based data manipulator (HDDM). *NAVIGATION*, *68*(4), 779–797. https://doi.org/10.1002/navi.441

Wendel, J., Schubert, F. M., Rügamer, A., & Taschke, S. (2016, September). Limits of narrowband interference mitigation using adaptive notch filters. *Proc. of the 29th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, Portland, OR, 286–294. https://doi.org/10.33012/2016.14799

How to cite this article: van der Merwe, J. R., Cortés, I., Garzia, F., Rügamer, A., & Felber, W. (2023). Multi-Parameter adaptive notch filter (MPANF) for enhanced interference mitigation. *NAVIGATION, 70*(2). https://doi.org/10.33012/navi.570

APPENDIX

A | Transfer Function

Let:

$$\hat{z}_0[n]=1 \qquad z^{-1}=\mathrm{e}^{-2\,\mathrm{j}x} \tag{A1}$$

$$x=-\frac{\omega T_s}{2}=-\frac{\pi f_c}{f_s} \tag{A2}$$

Then the transfer function can be simplified to:

$$H_0=\frac{1-\hat{z}_0[n]z^{-1}}{1-k_\alpha\,\hat{z}_0[n]z^{-1}}=\frac{1-\mathrm{e}^{-2\,\mathrm{j}x}}{1-k_\alpha\mathrm{e}^{-2\,\mathrm{j}x}} \tag{A3}$$

$$=\frac{\mathrm{e}^{-\,\mathrm{j}x}(\mathrm{e}^{\,\mathrm{j}x}-\mathrm{e}^{-\,\mathrm{j}x})}{(1-k_\alpha)-k_\alpha\mathrm{e}^{-\,\mathrm{j}x}(\mathrm{e}^{\,\mathrm{j}x}-\mathrm{e}^{-\,\mathrm{j}x})} \tag{A4}$$

$$=\frac{\mathrm{e}^{-\,\mathrm{j}x}\,2\,\mathrm{j}\sin(x)}{(1-k_\alpha)-k_\alpha\mathrm{e}^{-\,\mathrm{j}x}\,2\,\mathrm{j}\sin(x)} \tag{A5}$$

$$=\frac{1}{k_\alpha+\dfrac{1-k_\alpha}{\mathrm{e}^{-\,\mathrm{j}x}\,2\,\mathrm{j}\sin(x)}} \tag{A6}$$

$$=\frac{1}{k_\alpha-\mathrm{j}\dfrac{1-k_\alpha}{2\sin(x)}\mathrm{e}^{\,\mathrm{j}x}} \tag{A7}$$

$$=\frac{1}{k_\alpha-\mathrm{j}\dfrac{1-k_\alpha}{2\sin(x)}(\cos(x)+\mathrm{j}\sin(x))} \tag{A8}$$

$$=\frac{1}{\dfrac{k_\alpha+1}{2}-\mathrm{j}\dfrac{1-k_\alpha}{2\tan(x)}} \tag{A9}$$

B | CUTOFF FREQUENCY

The suppression on a power scale is:

$$L=\left|\frac{1}{H_0}\right|^2=\left(\frac{k_\alpha+1}{2}\right)^2+\left(\frac{1-k_\alpha}{2\tan(x)}\right)^2 \tag{B1}$$

Through simplification, the final notch width can be determined:

$$4L = (k_\alpha + 1)^2 + \frac{(1-k_\alpha)^2}{\tan^2(x)} \tag{B2}$$

$$\tan^2(x) = \frac{(1-k_\alpha)^2}{4L-(k_\alpha+1)^2} \tag{B3}$$

$$x = \tan^{-1}\left(\frac{1-k_\alpha}{\sqrt{4L-(k_\alpha+1)^2}}\right) \tag{B4}$$

$$f_c = \frac{-f_s}{\pi}\tan^{-1}\left(\frac{1-k_\alpha}{\sqrt{4L-(k_\alpha+1)^2}}\right) \tag{B5}$$

C | POLE-CONTRACTION FACTOR

Using (B1):

$$4L = (k_\alpha + 1)^2 + \frac{(1-k_\alpha)^2}{\tan^2(x)} \tag{C1}$$

$$\begin{aligned} 4L\sin^2(x) &= (\sin^2(x)+\cos^2(x)) \\ &\quad +2k_\alpha(\sin^2(x)-\cos^2(x)) \\ &\quad +k_\alpha^2(\sin^2(x)+\cos^2(x)) && \text{(C2)} \\ &= 1-2k_\alpha\cos(2x)+k_\alpha^2 && \text{(C3)} \end{aligned}$$

$$0 = k_\alpha^2 - 2\cos(2x)k_\alpha + 1 - 4L\sin^2(x) \tag{C4}$$

$$k_\alpha = \frac{2\cos(2x) \pm \sqrt{4\cos^2(2x) - 4(1-4L\sin^2(x))}}{2} \tag{C5}$$

$$= \cos(2x) \pm \sqrt{\cos^2(2x) - 1 + 4L\sin^2(x)} \tag{C6}$$

$$= \cos(2x) \pm \sqrt{4L\sin^2(x) - \sin^2(2x)} \tag{C7}$$

Only the negative sign results in a value for $k_\alpha \leq 1$, therefore:

$$k_\alpha = \cos(2x) - \sqrt{4L\sin^2(x) - \sin^2(2x)} \tag{C8}$$

D | NOTCH WIDTH STEP

Starting from the base transfer function:

$$H_0 = \frac{1-\hat{z}_0\, z^{-1}}{1-k_\alpha\, \hat{z}_0\, z^{-1}} \tag{D1}$$

Rewrite in the time domain:

$$y[n] = k_\alpha\, \hat{z}_0 y[n-1] + x[n] - \hat{z}_0 x[n-1] \tag{D2}$$

Assume that the notch frequency (i.e., $\hat{z}_0$) is constant and that the pole-contraction factors change. The current pole-contraction factor, $k_{\alpha 1}$, and the previous $k_{\alpha 0}$ can be written as separate equations:

$$y[n] = k_{\alpha 1}\, \hat{z}_0 y[n-1] + x[n] - \hat{z}_0 x[n-1] \tag{D3}$$

$$y[n-1] = k_{\alpha 0}\, \hat{z}_0 y[n-2] + x[n-1] - \hat{z}_0 x[n-2] \tag{D4}$$

Combining these two functions:

$$y[n] = k_{\alpha 0} k_{\alpha 1} \hat{z}_0^2\, y[n-2] + x[n] + (k_{\alpha 1} - 1)\, \hat{z}_0\, x[n-1] - k_{\alpha 1}\, \hat{z}_0^2\, x[n-2] \tag{D5}$$

Finally, the new transfer function can be determined:

$$H_0(z) = \frac{1 + (k_{\alpha 1} - 1)\hat{z}_0\, z^{-1} - k_{\alpha 1} \hat{z}_0^2\, z^{-2}}{1 - k_{\alpha 1} k_{\alpha 0}\, \hat{z}_0^2\, z^{-2}} \tag{D6}$$

$$= \frac{(1-\hat{z}_0\, z^{-1})\ (1 + k_{\alpha 1} \hat{z}_0\, z^{-1})}{\left(1 - \sqrt{k_{\alpha 1} k_{\alpha 0}}\, \hat{z}_0\, z^{-1}\right)\left(1 + \sqrt{k_{\alpha 1} k_{\alpha 0}}\, \hat{z}_0\, z^{-1}\right)} \tag{D7}$$

E | SUPPLEMENTAL RESULTS

This section presents supplemental results for 10-us and 20-us chirp repetition intervals.

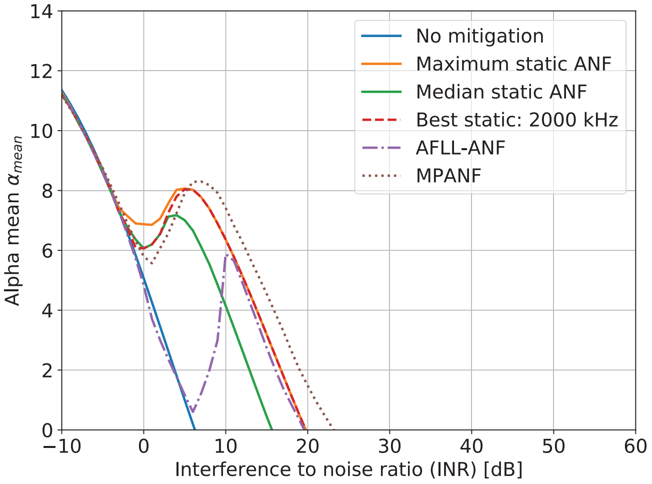

FIGURE E1 10-us continuous chirp without pulse blanker

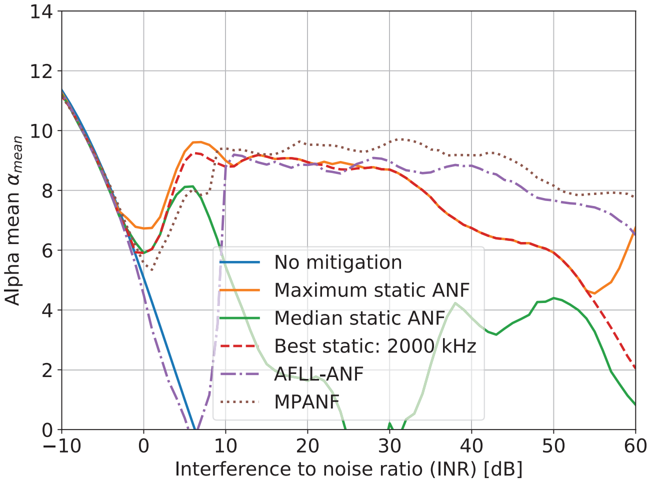

FIGURE E2 10-us continuous chirp with pulse blanker

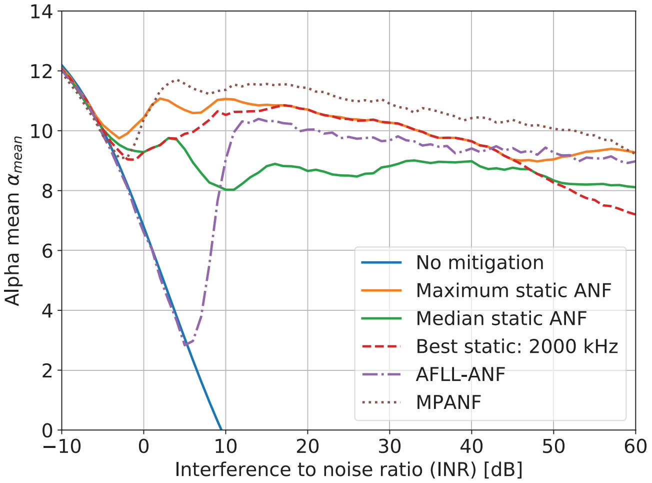

FIGURE E3 10-us-pulsed chirp with pulse blanker

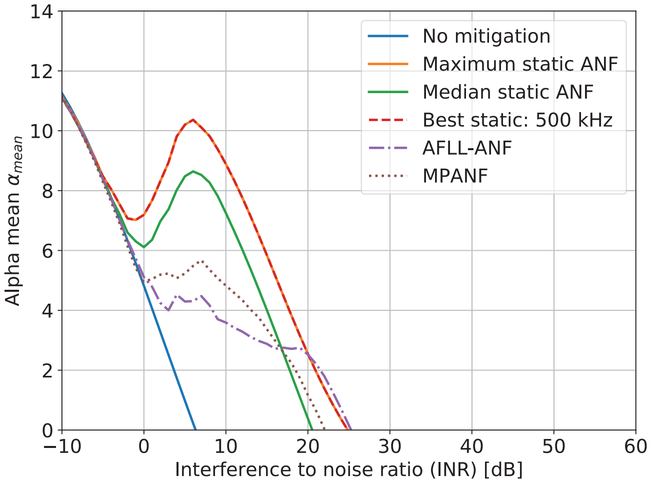

FIGURE E4 20-us continuous chirp without pulse blanker

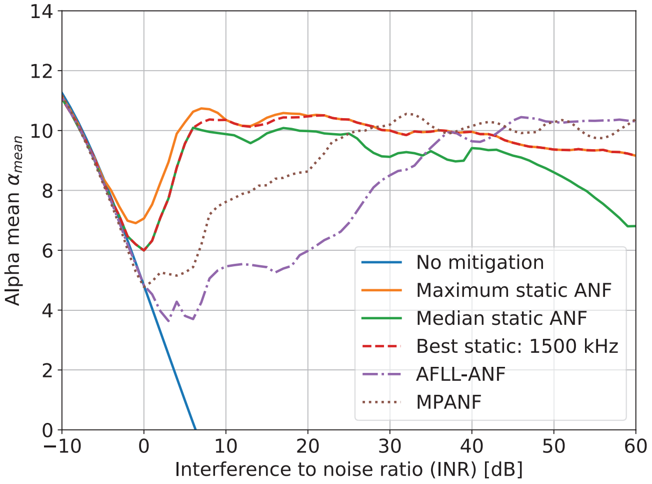

FIGURE E5 20-us continuous chirp with pulse blanker

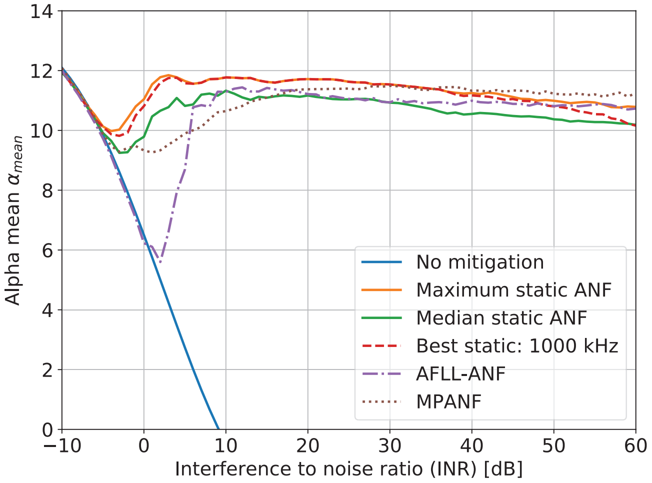

FIGURE E6 20-us-pulsed chirp with pulse blanker