Performance and Complexity Comparison of Adaptive Loop-Bandwidth Tracking Techniques

Iñigo Cortés, Jon Ander Iñiguez de Gordoa, 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 analyzes the performance and complexity of state-of-the-art adaptive scalar tracking techniques used in modern digital global navigation satellite system (GNSS) receivers. Ideally, a tracking channel should be able to adjust to both noisy and dynamic environments for optimal performance. Precision and robustness define the performance of the tracking. The difference between the square root of the Cramér-Rao bound (CRB) and the average tracking jitter at the discriminator's output determines the precision of the tracking, whereas the speed of the response specifies the robustness of the tracking in high dynamic scenarios. The amount of operations required to implement a robust tracking technique indicates the complexity of the algorithm. The fast adaptive bandwidth, the fuzzy logic, and the loop-bandwidth control algorithm adaptive tracking techniques are first analyzed and evaluated in a software receiver. Second, these techniques are 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 loop-bandwidth control algorithm's advantage over adaptive loop-bandwidth techniques while preserving good tracking performance and low complexity.**

***Index Terms*—Global navigation satellite system (GNSS), adaptive scalar tracking loop (A-STL), fast adaptive bandwidth (FAB), fuzzy logic, loop-bandwidth control algorithm (LBCA).**

I. Introduction

In global navigation satellite system (GNSS) receivers, it is necessary to maintain the lock onto the satellite signals in order to decode the navigation message and determine the pseudo-range to a given satellite [1]. Locking onto the signal is referred to as tracking. A GNSS receiver must utilize the best possible means to track the signals, since it leads to a better pseudo-range estimation, and in turn, to a more precise and reliable position, velocity, and time (PVT) solution. However, noise, receiver dynamics, and multipath effects make this a challenging task.

A scalar tracking loop (STL) handles the tracking of each received GNSS signal fully independently. It synchronizes to the carrier frequency (frequency locked loop (FLL)), carrier phase (phase locked loop (PLL)) and code phase (delay locked loop (DLL)). 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 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.

978-1-7281-6455-7/20/$31.00 ©2020 IEEE

Conventional STLs present a trade-off between the ability to stay locked in the presence of dynamics and the amount of filtered noise to achieve better precision. Dynamics stress the tracking loop, whereas noise reduces the tracking performance. The former is characterized by changes in position, velocity, acceleration, jerk, or even higher orders of change, and the latter arises due to the thermal noise, the oscillator phase noise, and other noise sources. The precision of the tracking, the dynamics, and the capability to stay in lock depend on the loop type, the loop order, the integration time and the loop bandwidth.

Variable loop-bandwidth tracking techniques give a solution to this problem by adapting the loop-bandwidth of the STL depending on the scenario conditions [3]–[7]. This paper evaluates the performance and complexity of different state-of-the-art variable loop-bandwidth tracking techniques. First, Section II presents background on STL, followed with a detailed description of the implemented variable loop-bandwidth tracking techniques in Section III. Section IV presents the results of the implemented techniques in an open software interface GNSS hardware receiver. Finally, Section V draws conclusions and indicates future work.

II. Scalar Tracking Loop

This section presents an introduction to standard STLs. Fig. 1 shows the general model of an STL. The output of the

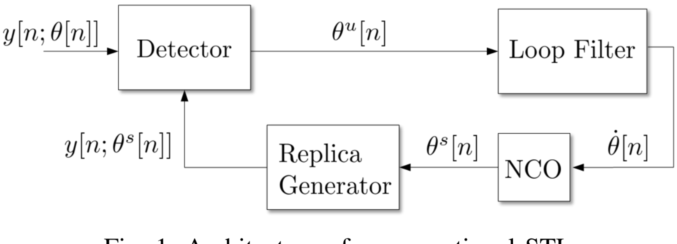


Fig. 1: Architecture of a conventional STL

Authorized licensed use limited to: FhI fur Integrierte Schaltungen Angewandte Elek. Downloaded on February 05,2021 at 09:10:51 UTC from IEEE Xplore. Restrictions apply.

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 loop-bandwidth affects the time of response of the loop-filter. Noise-suppressing tracking needs smaller loop-bandwidths, whereas larger loop bandwidths are required to react faster against signal dynamics. 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. 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. Before the discriminator block, the signal is passed through an integration and dump (IAD) module in order to sample and integrate the signal.

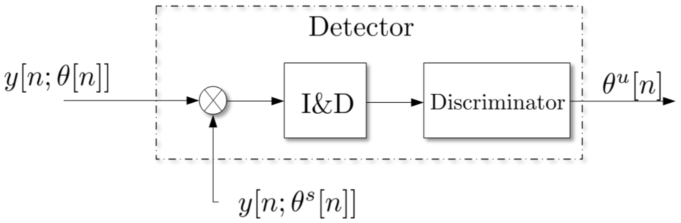


Fig. 2: Architecture of the detector

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 sensitive against signal dynamics. The product between the loop-bandwidth and the integration time T_i sets the response time of the STL.

III. Variable Loop Bandwidth Tracking Techniques

Variable loop-bandwidth tracking techniques set a connection between the loop-bandwidth and time-varying scenario conditions. In dynamic scenarios, a fast loop response with a large loop-bandwidth is preferred in order to follow the dynamics, whereas in stationary scenarios a noise-rejecting low bandwidth is appropriate. This category adapts the loop-bandwidth depending on the noise and signal dynamics of the tracking channel. Two techniques from this category are presented below.

A. Fast Adaptive Bandwidth

The fast adaptive bandwidth (FAB) tracking technique estimates permanently the input signal parameters (thermal noise, phase noise or steady state error (SSE)) of the STL using accurate models [8]. The combination of these models creates a loop-bandwidth dependent cost function $c[n;B[n]]]$. Setting the first derivative of the cost function to zero with respect to the loop-bandwidth $B[n]$, leads to an estimated optimal loop-bandwidth B_opt:

$$\frac{\partial c[n;B[n]]}{\partial B[n]} = 0 \rightarrow B_\text{opt} \tag{1}$$

The algorithm updates the current loop-bandwidth. The three-sigma rule-of-thumb of the tracking loop error is a common cost function [9]. The noisy output of the discriminator makes it difficult to measure the SSE correctly (even more for higher order loops). One solution is to accumulate the noisy measurements for a large interval [3] or to implement a dynamic error filter [4]. Since the poles of the loop filter are directly related to the loop-bandwidth, a cost function can be used to estimate the optimum pole of the loop filter [5]. An abrupt change of the estimated optimal loop-bandwidth B_opt may create tracking instabilities. Therefore, different solutions are required. An empirical scaling factor for the dynamic stress estimator and a constrained loop-bandwidth can solve this problem [4]. Another solution is to smooth the loop-bandwidth update using the Newton-Raphson method and a low pass filter (LPF) [5].

This paper presents a three-sigma rule-of-thumb based FAB. Fig. 3 shows the structure of this method. The cost function of the presented FAB algorithm is the jitter of the smoothed error σ^s_θ:

$$\sigma^s_\theta[n;B[n]] = \sigma^s_\text{thermal}[n;B[n]] + \frac{\theta_\text{sse}[n;B[n]]}{3} \tag{2}$$

where σ^s_thermal is the thermal noise and θ_sse is the dynamic stress error.

Since σ^s_θ is loop-bandwidth dependent, the first order derivative of σ^s_θ regarding the loop-bandwidth can be performed. By equaling the derivative to zero, the calculated optimal loop-bandwidth B_opt for a 3rd order Costas PLL is [9]:

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

B_opt is dependent on the C/N_0, the integration time T_i and the line-of-sight (LOS) jerk dynamics $\frac{\partial^3 R}{\partial t^3}$. The constant η is the relation coefficient between the noise loop-bandwidth and the natural frequency of the tracking loop. For a 3rd order PLL, η equals to 0.7845.

The integration time T_i is selected to have a constant value of 20 ms. The C/N_0 estimator updates each tracking epoch. The dynamic stress estimator filters first the discriminator's output. The filter is a first order infinite impulse response (IIR) with a cut-off frequency f_co of 1 Hz. Therefore, an effective exponential moving average of 1 s of data is achieved. Second, it performs the third derivative of the filtered discriminator's

Authorized licensed use limited to: FhI fur Integrierte Schaltungen Angewandte Elek. Downloaded on February 05,2021 at 09:10:51 UTC from IEEE Xplore. Restrictions apply.

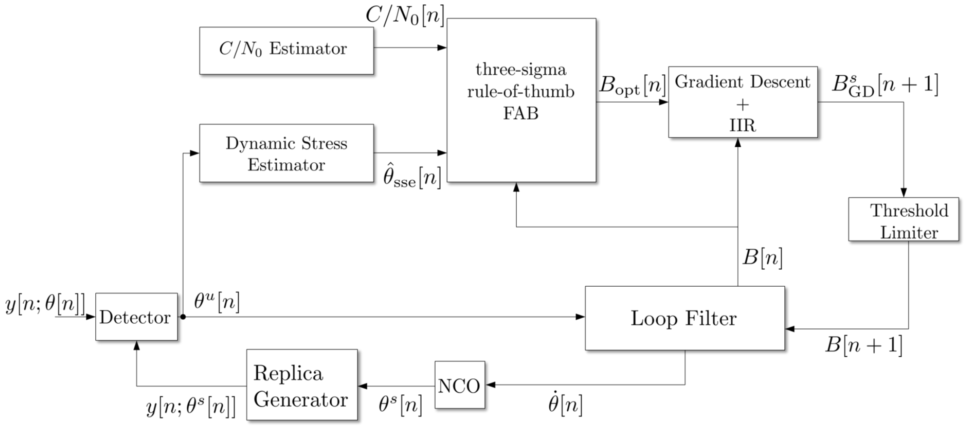


Fig. 3: Architecture of the 3-sigma rule-of-thumb based FAB

output μ^u_θ in order to achieve an estimated jerk measurement $\hat{\theta}_{\text{sse}}$:

$$\hat{\theta}_{\text{sse}}[n] = \frac{\mu^u_\theta[n] - 3\mu^u_\theta[n-1] + 3\mu^u_\theta[n-2] - \mu^u_\theta[n-3]}{\Delta t^3} \tag{4}$$

where Δt is the time of filtered data by the mentioned first order IIR filter, 1 s. Since the unsmoothed error is normalized, $\hat{\theta}_{\text{sse}}$ is in units of $(1/\text{s}^3)$. Next, the LOS jerk dynamics $\frac{\partial^3 R}{\partial t^3}$ (deg/s^3) is calculated as the following:

$$\frac{\partial^3 R}{\partial t^3}[n] = 360 \left(\frac{B[n]}{\eta}\right)^3 \hat{\theta}_{\text{sse}}[n] \tag{5}$$

Since abrupt loop-bandwidth changes may result in tracking instabilities, the gradient descent method and a first order IIR filter is used to smooth the update. The gradient descent method updates the actual loop-bandwidth $B[n]$ gradually:

$$B_{\text{GD}}[n] = B[n] + T_{\text{i}} \frac{B_{\text{opt}}[n] - B[n]}{|\Delta B_{\text{opt}}[n]|} \tag{6}$$

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

$$\Delta B_{\text{opt}}[n] = B_{\text{opt}}[n-1] - B_{\text{opt}}[n] \tag{7}$$

The updated loop-bandwidth $B_{\text{GD}}[n]$ depends on the actual loop-bandwidth $B[n]$, the estimated optimum loop-bandwidth $B_{\text{opt}}[n]$ and the previous estimated loop-bandwidth $B_{\text{opt}}[n-1]$. To avoid instabilities, $|\Delta B_{\text{opt}}[n]|$ is hard set to the integration time T_{i} if it is less than the latter.

Next, a LPF smooths $B_{\text{GD}}[n]$, achieving $B^s_{\text{GD}}[n]$. Finally, the filtered loop-bandwidth is passed through a threshold limiter:

$$B[n+1] = \begin{cases} 18 & \text{if } B^s_{\text{GD}}[n] > 18 \\ B^s_{\text{GD}}[n] & \text{if } 4 \geq B^s_{\text{GD}}[n] \leq 18 \\ 4 & \text{if } B^s_{\text{GD}}[n] < 4 \end{cases} \tag{8}$$

A number of conclusions of the FAB implementation can be drawn before showing the results in the next section. First, the speed of the SSE estimation $\hat{\theta}_{\text{sse}}$ determines the speed of the algorithm to react against signal dynamics. Second, the FAB algorithm does not consider other sources of signal dynamics such as clock drift and low order transient dynamics. Hence, in a static scenario the B_{opt} would tend erroneously to zero due to the fact that $\hat{\theta}_{\text{sse}}$ is negligible and other dynamic sources are not included. Third, the complexity of the algorithm seems to be significant due to the seventh root in Equation (3) and the number of divisions to be performed.

B. Fuzzy Logic

Compared to the FAB technique, fuzzy logic based tracking techniques simplify significantly the complexity of the control algorithm. The presented fuzzy logic algorithm is based on [7]. Fig. 4 shows the implemented fuzzy logic algorithm structure. The standard deviation σ^u_θ and the absolute mean $|\mu^u_\theta|$ of the discriminator's output are the inputs for this algorithm. The normalization stage normalizes the estimated noise $\bar{N}$ and the estimated dynamics $\bar{D}$:

$$\bar{N} = \frac{\sigma^u_\theta}{\sigma^u_\theta + |\mu^u_\theta|} \tag{9}$$

$$\bar{D} = \frac{|\mu^u_\theta|}{\sigma^u_\theta + |\mu^u_\theta|} \tag{10}$$

Next, the *zero* f^{ZO}, the *positive-small* f^{PS} and the *positive-large* f^{PL} fuzzy functions weight each normalized estimation. These fuzzy functions are characterized as follows:

$$f^{\text{ZO}}[n;\psi[n]] = \begin{cases} \frac{T^\psi_{\text{fuzzy}} - \psi[n]}{T^\psi_{\text{fuzzy}}} & \text{if } 0 \leq \psi[n] \leq T^\psi_{\text{fuzzy}} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$f^{\text{PS}}[n;\psi[n]] = \begin{cases} \frac{\psi[n]}{T^\psi_{\text{fuzzy}}} & \text{if } 0 \leq \psi[n] \leq T^\psi_{\text{fuzzy}} \\ \frac{1-\psi[n]}{1-T^\psi_{\text{fuzzy}}} & \text{if } T^\psi_{\text{fuzzy}} \leq \psi[n] \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$f^{\text{PL}}[n;\psi[n]] = \begin{cases} \frac{\psi[n] - T^\psi_{\text{fuzzy}}}{1-T^\psi_{\text{fuzzy}}} & \text{if } T^\psi_{\text{fuzzy}} \leq \psi[n] \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$\psi[n]$ is the input estimation ($\bar{N}$ or $\bar{D}$) and T^ψ_{fuzzy} is the function threshold that defines the regions of these three fuzzy functions. The function threshold of $\bar{D}$ and $\bar{N}$ are:

$$T^{\bar{D}}_{\text{fuzzy}} = D^{opt} \tag{14}$$

Authorized licensed use limited to: FhI fur Integrierte Schaltungen Angewandte Elek. Downloaded on February 05,2021 at 09:10:51 UTC from IEEE Xplore. Restrictions apply.

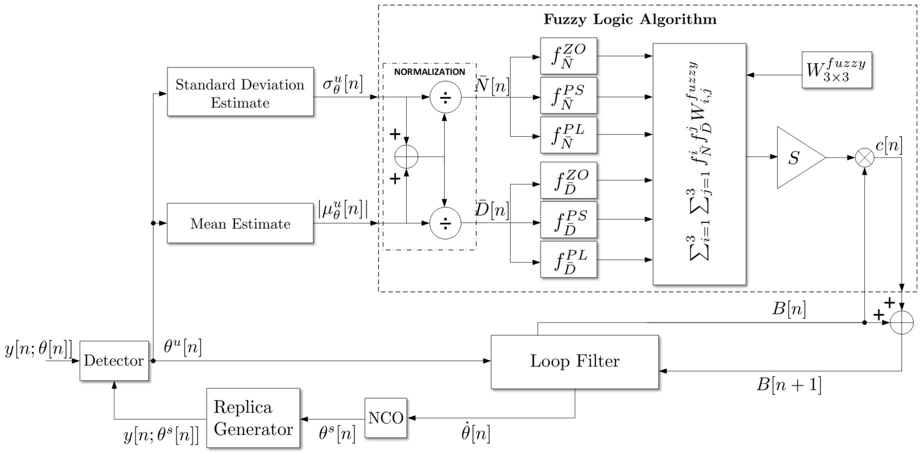


Fig. 4: Architecture of implemented fuzzy logic technique

$$T^{\bar{N}}_{\text{fuzzy}} = 1 - T^{\bar{D}}_{\text{fuzzy}} \tag{15}$$

where D^{opt} is the optimal normalized dynamics in order to achieve best tracking performance. The three-sigma rule-of-thumb is considered to determine this parameter. For a 3rd order PLL the optimal normalized dynamics D^{opt} is $1/7$ [10]. Fig. 5 shows the fuzzy functions for $\bar{D}$ and $\bar{N}$. An interesting remark is the symmetry between the weighted estimates since $\bar{D} = 1 - \bar{N}$:

$$f^{\text{ZO}}[n;\bar{D}[n]] = f^{\text{PL}}[n;\bar{N}[n]] \tag{16}$$

$$f^{\text{PS}}[n;\bar{D}[n]] = f^{\text{PS}}[n;\bar{N}[n]] \tag{17}$$

$$f^{\text{PL}}[n;\bar{D}[n]] = f^{\text{ZO}}[n;\bar{N}[n]] \tag{18}$$

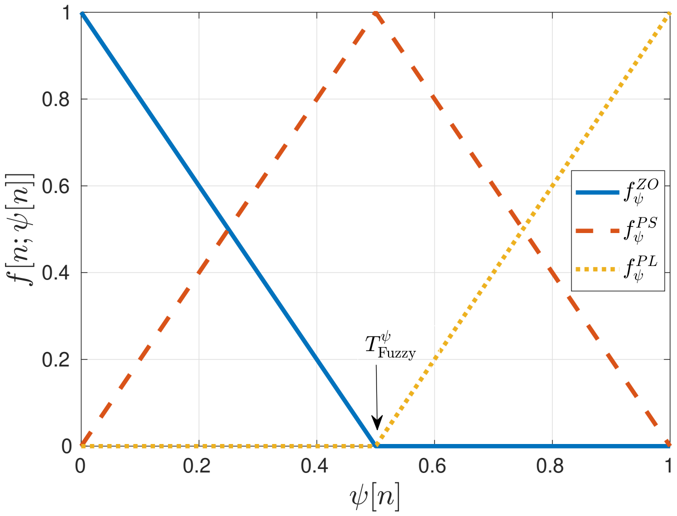


Fig. 5: Fuzzy functions for normalized estimates $\bar{D}$ and $\bar{N}$

The fuzzy-weighted estimates are combined together with a weighting fuzzy matrix $W^{\text{fuzzy}}_{3\times3}$:

$$P[n] = \sum_{i=1}^{3}\sum_{j=1}^{3} f^i[n;\bar{N}[n]] f^j[n;\bar{D}[n]] W^{\text{fuzzy}}_{i,j} \tag{19}$$

where f^1, f^2 and f^3 are f^{Z0}, f^{PS} and f^{PL}, respectively.

Table I shows the values of $W^{\text{fuzzy}}_{3\times3}$ that are empirically set.

		$\bar{D}$		
		ZO	PS	PL
$\bar{N}$	ZO	0	0.5	0.75
	PS	-0.25	0	0.5
	PL	-0.5	-0.25	0

TABLE I: Values of the fuzzy Matrix $W^{\text{fuzzy}}_{3\times3}$

The resultant value $P[n]$ is scaled by S and multiplied by the current loop-bandwidth $B[n]$, achieving the final control signal $c[n]$:

$$c[n;P[n];B[n]] = P[n]SB[n] \tag{20}$$

Finally, the loop-bandwidth is updated:

$$B[n+1] = B[n] + c[n;P[n];B[n]] \tag{21}$$

Two examples are addressed. If the normalized dynamics $\bar{D}$ tends to zero, the normalized noise $\bar{N}$ goes to one. Therefore, $f^{\text{ZO}}[n;\bar{D}[n]]$ and $f^{\text{PL}}[n;\bar{N}[n]]$ tend to one and the remain fuzzy functions are zero. Then, $P[n]$ results as:

$$P[n] = W^{\text{fuzzy}}_{3,1} = -0.5 \tag{22}$$

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

In the opposite case, if $\bar{D}$ tends to one, the normalized noise $\bar{N}$ goes to zero. In such a case, $P[n] = 0.75$ and the loop-bandwidth increases together with the control signal. The bigger the loop-bandwidth, the bigger the update. This can lead to instabilities due to abrupt changes of the loop-bandwidth.

Authorized licensed use limited to: FhI fur Integrierte Schaltungen Angewandte Elek. Downloaded on February 05,2021 at 09:10:51 UTC from IEEE Xplore. Restrictions apply.

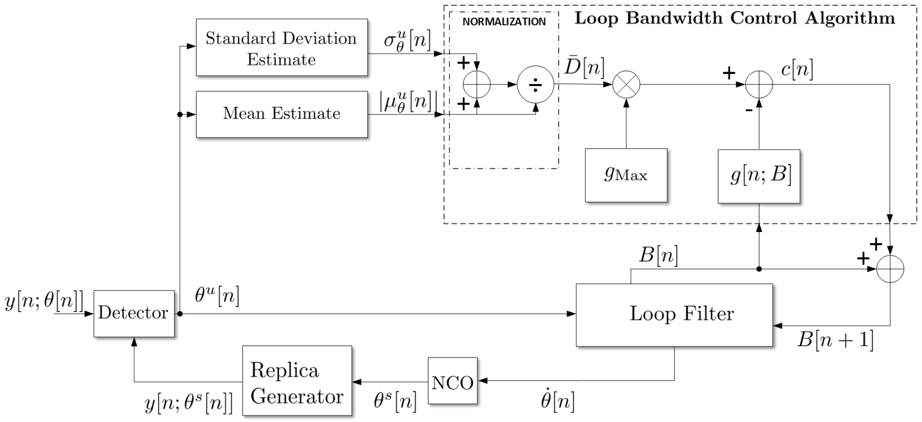


Fig. 6: Architecture of loop-bandwidth control algorithm (LBCA)

C. Loop-Bandwidth Control Algorithm

Similar to the fuzzy logic method, the LBCA [10] uses the discriminator's statistics to adapt the loop-bandwidth. However, it uses a loop-bandwidth dependent sigmoid-based weighting function to combine these values. Fig. 6 shows the structure of the LBCA.

The inputs of the algorithm are the absolute values of the mean and the standard deviation estimates of the discriminator's output. The former is interpreted as the dynamics and the latter as the noise of the tracking channel. The signal dynamic estimate is normalized $\bar{D}$. Next, $\bar{D}$ is scaled by the maximum value of the weighting function g_{Max} and it is combined with the weighting function $g[n;B]$. The weighting function is a linear combination of sigmoid functions and it is shaped considering the mentioned optimal normalized dynamics D^{opt}. Finally, the difference between the estimated weighted dynamics and the estimated weighted noise is performed, achieving the update that is added to the current loop-bandwidth. The control value $c[n;B[n]]$ used is:

$$c[n;B[n]] = 0.01\cdot\bar{D} - \begin{bmatrix}0.002\\0.008\end{bmatrix}^T \begin{bmatrix}\text{Sig}\left((B[n]-3)\right)\\ \text{Sig}\left(5(B[n]-16)\right)\end{bmatrix} \quad (23)$$

IV. Experiment and Results

The GOOSE platform from Fraunhofer IIS is a GNSS receiver with an open software interface [11]. The tracking stage is partially implemented in hardware and software. The correlators and the NCOs are implemented in a field-programmable gate array (FPGA), whereas the software part includes the discriminators, the loop filters and the adaptive tracking algorithms. Each tracking channel contains a 2nd order FLL, a 3rd order Costas PLL and a 2nd order DLL with PLL assisted DLL (PAD) enabled. The FLL is first enabled in order to track and refine the acquired Doppler frequency. Once a stable lock is achieved, the transition to the PLL is done, passing through a FLL assisted PLL (FAP). The purpose is to evaluate the performance and the complexity of each variable loop-bandwidth tracking technique implemented in the PLL of this receiver against simulated scenarios with different dynamics and noise levels. These simulations use GPS L1 C/A signals and the integration time T_{i} is 20 ms. The test set-up is the same as with previous studies [10], where a Spirent GSS9000 radio-frequency constellation simulator (RFCS) is used to generate controlled scenarios. These scenarios are either static or have receiver dynamics. The duration of the simulation is 20 minutes and 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 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. Then, each 30 seconds the C/N_0 level is reduced 4 dB until reaching the desired level. For instance, 3.5 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 16 minutes of the simulation are considered.

A. Performance of Variable Loop-Bandwidth Tracking Techniques

In order to define the quality factor to determine the tracking performance of an adaptive tracking technique, two parameters are considered: the Cramér-Rao bound (CRB) and the one-sigma rule threshold of the unsmoothed error $\sigma_{\theta^u}^{th}$. The CRB indicates the minimum error variance of an unbiased estimator. Considering the PLL as a time of arrival (ToA) unbiased estimator [12], the resulting square-root CRB of θ^u is:

$$\sigma_{LB}^u = \left(\frac{\lambda}{2\pi}\right)\sqrt{\frac{1}{2T_{\text{i}}C/N_0}\left(1+\frac{1}{2T_{\text{i}}C/N_0}\right)} \quad (24)$$

where λ is the wavelength of the GNSS signal.

From the three-sigma rule of thumb, the jitter of the smoothed error σ_θ^s must be less than a conservative threshold in order to ensure an stable tracking [1]. This upper threshold is also applied to σ_θ^u. Since an arctangent discriminator is used, the one-sigma rule threshold in meters is:

$$\sigma_{\theta^u}^{th} = \frac{\Omega}{12}\times\frac{\lambda}{2\pi} = \frac{\pi}{12}\times\frac{\lambda}{2\pi} = \frac{\lambda}{24} \quad (25)$$

Authorized licensed use limited to: FhI fur Integrierte Schaltungen Angewandte Elek. Downloaded on February 05,2021 at 09:10:51 UTC from IEEE Xplore. Restrictions apply.

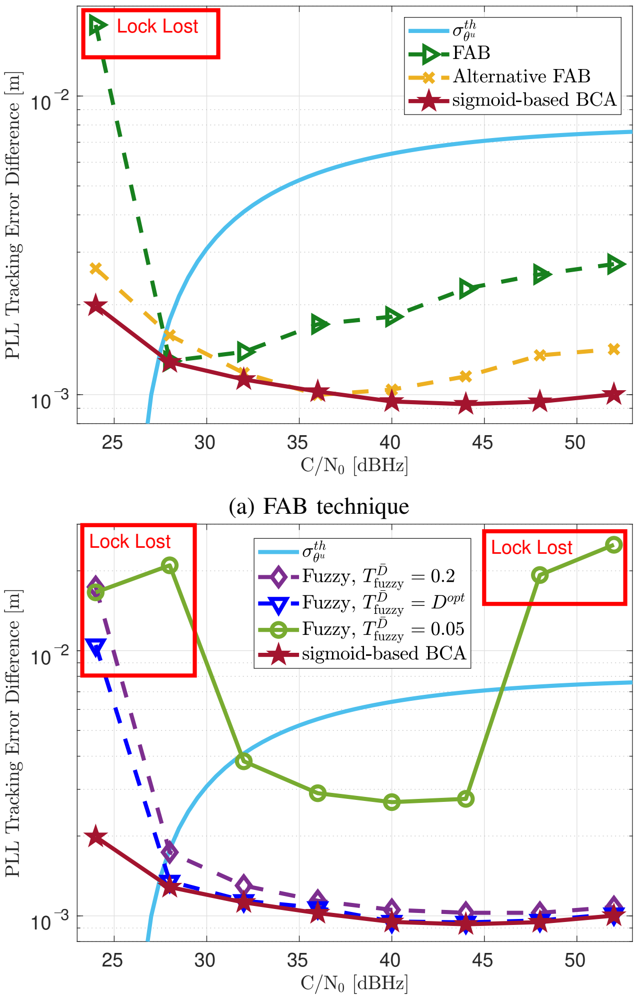
where Ω is the phase pull-in range in radians. The $1/12$ factor in (25) is included because the one-sigma rule threshold is one-third of the three-sigma rule threshold and one-fourth of Ω is selected to have a conservative threshold.

The tracking performance in a static scenario is defined as the difference between the average σ_{θ}^{u} and the lower bound standard deviation σ_{LB}^{u}. Fig. 7 shows the tracking error difference of the FAB and the fuzzy logic techniques for different C/N_0 in a static scenario. The one-sigma rule threshold $\sigma_{\theta^u}^{th}$ in meters is also included. Values greater than $\sigma_{\theta^u}^{th}$ mean that the lock of the tracking is likely lost and under sever cycle slips.

Fig. 7a compares the tracking performance of the FAB and the sigmoid-based LBCA. As expected, the presented FAB decreases the bandwidth every epoch, since the estimated jerk dynamics are negligible. The final loop-bandwidth tends to 4 Hz, because of the threshold limiter. An alternative FAB is implemented, adding each iteration 4 Hz to the estimated optimal loop-bandwidth. There is some improvement, but one can observe in Figure 8a that the loop-bandwidth has almost a constant value of 7 Hz. Therefore, bad results of the FAB are achieved.

The interesting part comes in the comparison between the fuzzy logic and the LBCA. Fig. 7b shows the tracking performance of both algorithms. This graph includes three fuzzy logic configurations with different function thresholds $T_{\text{fuzzy}}^{\bar{D}}$. The bigger $T_{\text{fuzzy}}^{\bar{D}}$, the more sensitive is the algorithm against noise and less against dynamics. The best performance is achieved when the function threshold of $\bar{D}$ tends to the optimal normalized dynamics D^{opt}. The performance is very close to the LBCA, with the difference that the LBCA can still track at 24 dBHz.

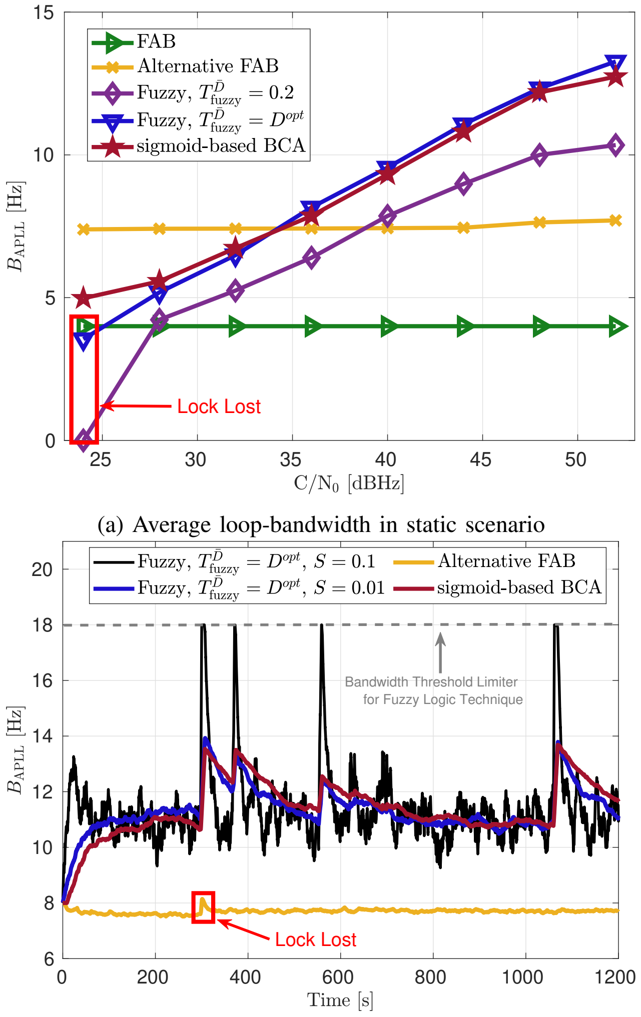
Fig. 8 presents the variation of the loop-bandwidth for different tracking techniques. Fig. 8a shows that, in a static scenario, the average loop-bandwidth using the fuzzy logic with $T_{\text{fuzzy}}^{\bar{D}} = D^{\text{opt}}$ is the closest to the LBCA method. The average loop-bandwidth of the fuzzy logic with $T_{\text{fuzzy}}^{\bar{D}} = 0.05$



(a) FAB technique

(b) Fuzzy logic technique with $S = 0.01$

Fig. 7: Comparison of the tracking error difference in static scenario for different tracking techniques



(a) Average loop-bandwidth in static scenario

(b) Loop-bandwidth in dynamic scenario at 44 dBHz

Fig. 8: Loop-bandwidth variation for different variable loop-bandwidth tracking techniques

Authorized licensed use limited to: FhI fur Integrierte Schaltungen Angewandte Elek. Downloaded on February 05,2021 at 09:10:51 UTC from IEEE Xplore. Restrictions apply.

does not give much information, since the bandwidth tends to more than 20 Hz, making the loop filter unstable. Therefore, it is not included.

The same dynamic scenario with a maximum jerk of $7.8\,g/s$ is used as in [10]. In contrast to static scenarios, the quality factor to indicate the performance of the tracking technique in dynamic scenarios is the speed to react against signal dynamics. Moreover, another parameter to consider is the ability of the algorithm to reach to the minimum necessary loop-bandwidth that can deal against dynamics. This is due to the fact that a bigger loop-bandwidth can introduce noise in the tracking loop and this can lead to a loss of lock, particularly, at low C/N_0. Fig. 8b shows the change of the loop-bandwidth in the dynamic scenario for the different tracking methods. The FAB reacts very slow against dynamics, losing at the end the lock. The fuzzy logic with a scale factor $S = 0.01$ presents instead a good performance similar to the LBCA. In the case of the fuzzy logic with $S = 0.1$, a threshold limiter is added because the updated loop-bandwidth increases to values bigger than 18 Hz. This fuzzy logic reacts faster against signal dynamics, but with the drawback of being highly unstable at high loop-bandwidths.

B. Complexity of the Variable Loop-Bandwidth Tracking Techniques

One method to quantify the complexity of an adaptive tracking technique is to measure the time each algorithm takes to be performed. However, this approach depends on how good the software implementation of the algorithm is and it does not bring an objective comparison. Another option is to count the required mathematical operations. This method is chosen because it gives a fair complexity comparison between algorithms. Table II shows the amount of additions, multiplications and divisions required for each case.

	FAB	Fuzzy Logic	LBCA
Additions	12	23	10
Multiplications	60	32	8
Divisions	6	4	1

TABLE II: Complexity of the Adaptive Loop-Bandwidth Tracking Techniques

In the case of the FAB algorithm, the seventh root is approximated by the derivation of the Newton-Raphson method [13], and considering the result in the fifth iteration. The sigmoid-based LBCA is approximated using a piecewise linear approximation [14]. The algorithm with less complexity is the LBCA, whereas the FAB presents the highest complexity due to the estimation of the optimal loop-bandwidth.

V. Conclusion

This paper compares the performance and complexity between different state-of-the-art adaptive loop-bandwidth tracking techniques. The algorithms are implemented on the GOOSE GNSS receiver platform. The FAB presents poor tracking performance and is the most complex. The fuzzy logic shows interesting results and it is a clear opponent for the LBCA regarding the tracking performance. The LBCA has superior tracking performance with the least amount of complexity.

Future research includes the study of an improved FAB algorithm with a cost function that is based on a sum of weighted estimated statistics from the discriminator's output [15]. Moreover, a future objective is to implement the Kalman filtering (KF)-based STL in GOOSE, measure the tracking performance and complexity, and compare it to the presented techniques.

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] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] I. Cortes, J. R. Van der Merwe, A. Rügamer, and W. Felber, "Adaptive loop-bandwidth control algorithm for scalar tracking loops," in *Proceedings of IEEE/ION PLANS*, 2020.

[11] 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.

[12] 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.

[13] K. Atkinson, *An Introduction to Numerical Analysis*. Wiley, 1989.

[14] 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.

[15] 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.

Authorized licensed use limited to: FhI fur Integrierte Schaltungen Angewandte Elek. Downloaded on February 05,2021 at 09:10:51 UTC from IEEE Xplore. Restrictions apply.