Received: 30 May 2025 | Revised: 19 May 2026 | Accepted: 7 July 2026

DOI: 10.33012/navi.788

ORIGINAL ARTICLE

Vector Tracking Loop Jitter

Susmita Bhattacharyya

[1] Department of Aerospace Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721302, India

Correspondence
Email: susmita@aero.iitkgp.ac.in

Abstract

The vector loop architecture is an advanced signal tracking algorithm for global navigation satellite system (GNSS) receivers. It has long been the subject of much research for its ability to address GNSS vulnerability. This paper presents a methodology to obtain the total jitter of the noncoherent vector tracking architecture using its state-space (SS) formulation, referred to as SS jitter, and relates the SS jitter to that determined with transfer function (TF) models. Thus, a theoretical framework for evaluating the performance of noncoherent vector loops is developed. In this regard, vector TFs are derived in the z-domain without decoupling the delay- and frequency-locked loops, unlike previous work. Two different noise equivalent bandwidths from the z-domain TF are also calculated. A close agreement between the SS and TF jitters is demonstrated, proving their equivalence. A metric to represent the collective performance of vector loops for loss-of-lock detection is identified using dynamically allocated bounds on the jitter. The steady-state jitter performance at different noise levels in the absence of dynamic stress error is analyzed in detail. Finally, using high-dynamic simulations of the Global Positioning System and Navigation with Indian Constellation signals, it is shown that the proposed metric holds promise for detecting loss of lock in the presence of jerk.

Keywords

loss of lock, noise equivalent bandwidth, state-space jitter, transfer function jitter, vector tracking loops

1 | INTRODUCTION

The vector tracking architecture is an advanced receiver algorithm developed to address the vulnerability of global navigation satellite system (GNSS) signals (Spilker Jr., 1995). Since its introduction in the 1980s (Copps et al., 1980), extensive research has been carried out to explore its potential for improving GNSS robustness against signal degradation, outages, and high platform dynamics (Lashley et al., 2009). Different variants of this tracking method with enhanced capabilities have also been proposed in the literature, including joint vector tracking (Liu et al., 2011), maximum likelihood vector architecture (Weill, 2010), differential vector phase-locked loops (Brewer & Raquet, 2016), and multi-receiver vector tracking (Ng & Gao, 2017). The superiority of the vector receiver over the traditional scalar one, however, comes at the price of increased computational

Licensed under CC-BY 4.0 © 2026 Institute of Navigation

complexity. Therefore, recent work has focused on reducing the computational load without sacrificing performance (Mu & Long, 2017; Abedi & Mosavi, 2022; Gómez et al., 2023), thus paving the way for real-time vector loop implementations. Multi-GNSS vector tracking receivers have also been studied in the literature for improved performance (Shytermeja, 2017; Tabatabaei et al., 2017).

Traditionally, the performance of a scalar tracking loop architecture is quantitatively evaluated by its total jitter (Ward et al., 2006). The jitter is determined by the sum of the 3-σ tracking error due to wideband and colored noises and dynamic stress error arising from abrupt changes in platform motion. Ward et al. (2006) studied different sources of jitter in detail for both conventional phase- and frequency-locked loops using appropriate transfer function (TF) models. With the help of the total jitter, Razavi et al. (2008) explained the relative importance of different tracking parameters in systematically designing a scalar loop. Jitter was also shown to be a reliable metric for loss-of-lock detection if the scalar tracking loop bandwidth (BW) is not too low. Yang et al. (2017) and Mute and Bhattacharyya (2023) analyzed how the tracking loop jitter varies in adaptive scalar loops whose parameters are adaptively chosen based on carrier-to-noise ratios (C/N_0) and platform dynamics (or line-of-sight jerk).

While the jitter of scalar tracking loops has been well studied, a similar treatment is lacking in the context of more complex vector architecture. Lashley et al. (2009) provided a methodology to calculate the vector jitter considering thermal noise, clock noise, and dynamic stress error for a noncoherent vector loop with 20-ms measurement update intervals. In their approach, the calculation of jitter due to noise involved updating the state error covariance using a two-step process. The dynamic stress error was shown to be derived from the state error difference between two separate Kalman filters, one of which had knowledge of the dynamics and, therefore, accurately modeled the change in acceleration and the other did not. As channels are coupled in vector tracking mode, the maximum jitter among all channels was compared against the threshold for loss-of-lock analysis. Simulation results showed that during an 8-g maneuver at 16 dB-Hz, the velocity estimates of the vector loops began to diverge. The jitters of the majority of channels also exceeded the threshold, thereby correctly predicting loss of lock. However, the performance was not validated at different C/N_0 ratios against specific platform jerk levels. To obtain the vector loop jitter with TF models similar to that of its scalar counterpart, appropriate TFs of the time-varying vector tracking architecture are needed. While Bhattacharyya (2018) described s-domain TF models and noise equivalent BWs of vector loops under certain assumptions, to the best of the author's knowledge, there is little work on vector jitter calculation using TFs.

This paper determines the total jitter of a noncoherent vector architecture with both state-space (SS) and TF formulations. The major contributions of this work are as follows. First, a theoretical framework for evaluating the performance of vector loops is presented in terms of SS- and TF-based tracking jitters. More precisely, *a priori* and *a posteriori* jitters based on the theoretical SS model, termed SS jitters, are obtained. Then, z-domain TF models of the vector loops are determined without decoupling the delay- and frequency-locked loops, unlike the work by Bhattacharyya (2018). The current TF formulation removes all other key assumptions of the previous work, is more generic in nature, and has a much simpler implementation. Two different noise equivalent BWs, one each for measurement noise and process noise, are also calculated. A close agreement between the SS and TF jitters is established to demonstrate their equivalence. Second, with the help of the SS jitter, a practical loss-of-lock detection algorithm is developed for the noncoherent vector loops. The SS jitters are initially obtained at the measurement update intervals. To reduce the computational load, the measurement updates of

the noncoherent vector loops are often performed at an interval larger than the coherent integration time, which is usually 10 ms for carrier frequency tracking and 20 ms for code phase (Bhattacharyya & Gebre-Egziabher, 2010). Therefore, intermediate bounds on the jitter are calculated every 20 ms in order to apply existing thresholds for loss-of-lock detection. A suitable metric for representing the collective performance of all channels is also identified. The proposed metric is used to detect loss of lock in the presence of jerk in a high-dynamic simulation. It should be noted that the simulations are performed with the Global Positioning System (GPS) L1 and Navigation with Indian Constellation (NavIC) L5 signals. Thus, both geometric and frequency diversities are attained in a single scenario, which demonstrates the effectiveness of the proposed methodology for different constellations as well as different signals.

The remainder of this paper is organized as follows. First, the approach to calculating the SS jitter is described. Then, TF models are introduced, and noise equivalent BWs are determined. Using this, the TF jitter is obtained. Subsequently, an appropriate metric is proposed for loss-of-lock detection with the help of the SS jitter. Next, simulation studies show the equivalence between the SS and TF jitters, and the vector jitter performance is analyzed for different noise and jerk levels. Finally, conclusions are provided.

2 | VECTOR LOOP JITTER

In this paper, a noncoherent vector tracking architecture is considered for GPS and NavIC constellations (see Figure 1) to determine the vector loop jitter. This architecture consists of vector delay-locked loops (VDLLs) for code phase tracking and vector frequency-locked loops (VFLLs) for carrier frequency tracking. The state transition and linearized measurement models of the vector navigation filter, which uses an extended Kalman filter (EKF) implementation, are as follows:

$$\underline{\varkappa}_\ell = \mathbf{F}\underline{\varkappa}_{\ell-1} + \underline{w}_{\ell-1} \tag{1}$$

$$\underline{z}_\ell = \mathbf{H}_\ell \Delta\underline{\varkappa}_\ell + \underline{\zeta}_\ell \tag{2}$$

The vector $\Delta\underline{\varkappa}_\ell$ at t_ℓ is given by $(\underline{\varkappa}_\ell - \hat{\underline{\varkappa}}_\ell^-)$. The true state vector is $\underline{\varkappa}$, and its *a priori* estimate is $\hat{\underline{\varkappa}}^-$. $\underline{\varkappa}$ comprises the user position, velocity, and acceleration in the Earth-centered Earth-fixed (ECEF) coordinate frame as well as GPS and NavIC receiver clock biases, drifts, and drift rates; thus, this vector has 15 elements. It should be noted that the position estimation error in the ECEF frame can later be converted to the local navigation frame to compute the vertical and horizontal position errors. In the filter, $\hat{\underline{\varkappa}}^-$ and the *a posteriori* estimate of $\underline{\varkappa}$, $\hat{\underline{\varkappa}}^+$, are propagated from a time instant t'_{k-1} to t'_k by multiplying the matrix, $\mathbf{F}'$, which is given as follows:

$$\mathbf{F}' = \mathbf{I}_{5\times5} \otimes \begin{bmatrix} 1 & \Delta t' & ((\Delta t')^2)/2 \\ 0 & 1 & \Delta t' \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $\otimes$ denotes the Kronecker tensor product and $\mathbf{I}_{5\times5}$ is a 5×5 identity matrix. t'_k increments every 20 ms; thus, $\Delta t' = \left(t'_k - t'_{k-1}\right) = 20$ ms. In contrast, the subscript ℓ in Equation (1) indicates the time instants for measurement updates

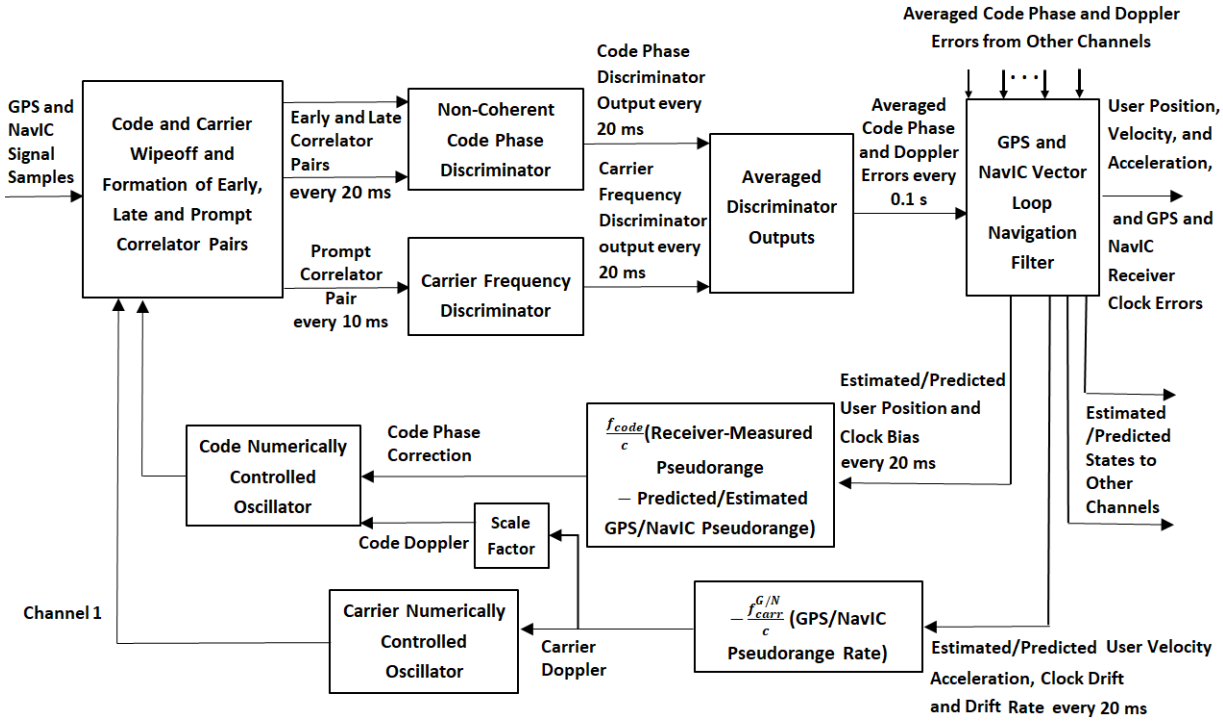

FIGURE 1 Noncoherent vector loop architecture

Appropriate variables at the channel and navigation levels are indicated in the figure. f_{code} and $f_{carr}^{G/N}$ are the code chipping rate and GPS/NavIC carrier frequency, respectively, and c is the speed of light.

(i.e., t_ℓ). In the implementation of this study, t_ℓ increments every 0.1 s (i.e., $\Delta t = \left(t_{\ell+1} - t_\ell\right) = 0.1$ s). It is apparent that $t_\ell = t'_k$ when $k = 5\ell$ (with $\ell = 1, 2, \ldots$). Hence, $\mathbf{F} = (\mathbf{F}')^5$. The process noise vector, $\underline{w}_{\ell-1}$, is assumed to be added to obtain the true states every t_ℓ, as shown in Equation (1). $\underline{w}_{\ell-1}$ has zero mean and error covariance $\mathbf{Q}$. $\underline{z}_\ell$ is the measurement innovation vector at t_ℓ, and $\underline{\zeta}_\ell$ is the measurement noise vector at t_ℓ with zero mean and covariance $\mathbf{W}_\ell$. $\underline{w}$ and $\underline{\zeta}$ are unknown vectors; only their statistical properties (i.e., mean and covariance) can be predicted. $\underline{z}_\ell$ is obtained from five 20-ms code phase (early-minus-late envelope) and carrier frequency (atan2) discriminator outputs averaged over 0.1 s. It should be noted that in the beginning, prior to the transition from scalar to vector loops, the coherent integration time of the correlators is synchronized with data bit transitions. The data bit boundaries are continuously tracked afterward by counting 20 1-ms coarse acquisition (C/A)-code periods. Thus, the two 10-ms prompt correlator pair intervals of the atan2 carrier frequency discriminator are ensured not to fall across data bit boundaries. The measurement model matrix, $\mathbf{H}_\ell$, relates $\underline{z}_\ell$ to $\Delta\underline{\varkappa}_\ell$, as defined in Appendix A.1. Interested readers can also refer to the supplementary material of Mute and Bhattacharyya (2023) for a detailed discussion of the measurement model and code and carrier numerically controlled oscillator corrections of the dual-constellation GPS L1 and NavIC L5 vector tracking architecture.

If $\underline{e}_\ell^+$ is the state estimation error after a measurement update (i.e., $\underline{e}_\ell^+ = (\underline{\varkappa}_\ell - \underline{\hat{\varkappa}}_\ell^+)$), then it can be propagated to the next time step as follows (Simon, 2006):

$$\underline{e}_{\ell+1}^+ = (\mathbf{I} - \mathbf{K}_{\ell+1}\mathbf{H}_{\ell+1})\mathbf{F}\underline{e}_\ell^+ + (\mathbf{I} - \mathbf{K}_{\ell+1}\mathbf{H}_{\ell+1})\underline{w}_\ell - \mathbf{K}_{\ell+1}\underline{\zeta}_{\ell+1} \tag{4}$$

where $\mathbf{K}_{\ell+1}$ is the Kalman gain at $t_{\ell+1}$. From Equation (4), the propagation of the *a priori* state error before a measurement update, $\underline{e}_{\ell}^{-}\left(=\Delta\underline{\varkappa}_{\ell}\right)$, is given as follows:

$$\underline{e}_{\ell+1}^{-}=(\mathbf{F}-\mathbf{K}'_{\ell}\mathbf{H}_{\ell})\underline{e}_{\ell}^{-}+\underline{w}_{\ell}-\mathbf{K}'_{\ell}\underline{\zeta}_{\ell} \tag{5}$$

where $\mathbf{K}'_{\ell}=\mathbf{F}\mathbf{K}_{\ell}$. The vector loop jitter is defined for the VDLL and VFLL, using the *a priori* and *a posteriori* state estimation errors, $\underline{e}_{\ell+1}^{-}$ and $\underline{e}_{\ell+1}^{+}$. This is because the code phase and carrier frequency of a channel are aligned with the incoming code phase and carrier frequency based on the estimated or predicted state vector of the vector navigation filter, as shown in Figure 1. Therefore, the state estimation/prediction error at the navigation level will reflect as code phase and carrier frequency alignment errors at the channel level. Hence, the *a posteriori* (and *a priori*) code phase and carrier Doppler errors together can be written as $\underline{\epsilon}_{e,\,\ell+1}=\mathbf{H}_{\ell+1}\underline{e}_{\ell+1}^{+}\left(\text{and }\underline{\epsilon}_{p,\,\ell+1}=\mathbf{H}_{\ell+1}\underline{e}_{\ell+1}^{-}\right)$. The vectors $\underline{\epsilon}_{e,\,\ell+1}$ and $\underline{\epsilon}_{p,\,\ell+1}$ are of dimension $2n\times 1$, where n is the number of visible satellites used for positioning. The code phase errors are given by the first n elements, and the carrier Doppler errors are obtained from the next n elements. The total *a posteriori* VDLL (or VFLL) jitter of a channel comprises the 3-σ *a posteriori* code phase (or carrier Doppler) estimation error due to random effects or noise plus the transient/steady-state error due to abrupt changes in platform dynamics (i.e., the dynamic stress error, which is treated as a deterministic quantity). This is an extension of the definition provided by Ward et al. (2006) for scalar tracking loops. The total *a priori* jitter is defined in a similar manner. As the vector navigation filter estimates the user position, velocity, and acceleration, the dynamic stress error primarily arises from the user jerk. In this paper, jitter is studied after the initial transients of the navigation filter have died down.

To calculate the vector loop jitter for each channel, the effects of the process and measurement noises should be reflected in $\underline{\epsilon}_{e,\,\ell+1}$ through $\underline{e}_{\ell+1}^{+}$ (and in $\underline{\epsilon}_{p,\ell+1}$ through $\underline{e}_{\ell+1}^{-}$). For this purpose, the matrix pair $(\mathbf{H}_{\ell+1},(\mathbf{I}-\mathbf{K}_{\ell+1}\mathbf{H}_{\ell+1})\mathbf{F})$ must be observable in the case of $\underline{\epsilon}_{e,\,\ell+1}$, where $\mathbf{H}_{\ell+1}$ is the measurement model matrix and $(\mathbf{I}-\mathbf{K}_{\ell+1}\mathbf{H}_{\ell+1})\mathbf{F}$ is the system model matrix (Kailath, 1980). This is proved in Appendix A.2, assuming that $(\mathbf{H}_{\ell+1},\mathbf{F})$ is observable for a stable vector navigation filter and $\mathbf{F}$ is invertible (valid for the adopted model). It is also proved that the matrix pairs $((\mathbf{I}-\mathbf{K}_{\ell+1}\mathbf{H}_{\ell+1})\mathbf{F},(\mathbf{I}-\mathbf{K}_{\ell+1}\mathbf{H}_{\ell+1}))$ and $((\mathbf{I}-\mathbf{K}_{\ell+1}\mathbf{H}_{\ell+1})\mathbf{F},\mathbf{K}_{\ell+1})$ are reachable or stabilizable with regard to the noises $\underline{w}_{\ell}$ and $\underline{\zeta}_{\ell+1}$ for $\underline{e}_{\ell+1}^{+}$, respectively, where the first matrix of a pair is the system model matrix and the second one is the coefficient matrix for the input vector (see Appendix A.3–A.4 for derivations). It should be noted that the matrix pairs written as before are the standard representations of matrices in control theory for defining the system observability and reachability/stabilizability. While the latter is not essential in the usual sense for jitter calculation, for completeness, the proof in the Appendix justifies that some modes of $\underline{e}_{\ell+1}^{+}$ are affected by noise. Hence, $\underline{\epsilon}_{e,\,\ell+1}$ would reflect the effects of $\underline{w}_{\ell}$ and $\underline{\zeta}_{\ell+1}$ on $\underline{e}_{\ell+1}^{+}$ through observability. The observability of $\epsilon_{p,\,\ell+1}$ is shown as part of the proof in Appendix A.2. The reachability criteria can be easily proved following the approach of the Appendix. Observability and stabilizability will also be assessed later in the context of the dynamic stress error. Next, the mathematical expression of the total SS jitter for vector loops is provided.

2.1 | State-Space Jitter

Simon (2006) showed that for a given Kalman gain matrix, the steady-state estimation error covariance can be written as the sum of the covariance due to process noise only and the covariance due to measurement noise only. That is, after the initial transients die down, the estimation error covariance is determined by the process and measurement noises. Therefore, if $\Sigma^+_{p,\ell}$ and $\Sigma^+_{m,\ell}$ denote the *a posteriori* error covariances only for $\underline{w}$ and $\underline{\zeta}$, respectively, then the *a posteriori* error covariance, Σ^+_ℓ, after the initial transients subside, is $\left(\Sigma^+_{p,\ell}+\Sigma^+_{m,\ell}\right)$. The same can also be written for the *a priori* error covariance, Σ^-_ℓ. Thus, two jitters due to noise (called noise jitters later) can be calculated, one each with Σ^-_ℓ and Σ^+_ℓ. These terms are called the *a priori* and *a posteriori* SS noise jitters, respectively, in this paper. The *a posteriori* SS noise jitter for channel i in the VDLL is given by the following:

$$\underline{\sigma}^+_{VDLL_{noise},\ell}(i)=\sqrt{\mathbf{G}^+_\ell(i,i)} \tag{6}$$

where $\mathbf{G}^+_\ell=\mathbf{H}_\ell\Sigma^+_\ell\mathbf{H}^T_\ell$ and $i=1,\ldots,n$. n is the number of satellites used for positioning at epoch ℓ and can change with time. $\underline{\sigma}^+_{VDLL_{noise},\ell}$ represents the VDLL noise jitter vector for all channels, whereas its argument denotes the i-th channel. For the VFLL, the *a posteriori* noise jitter of channel η is given as follows:

$$\underline{\sigma}^+_{VFLL_{noise},\ell}(\eta)=\sqrt{\mathbf{G}^+_\ell(\eta,\eta)},\eta=n+1,\ldots,2n \tag{7}$$

Similarly, using Σ^-_ℓ, the *a priori* noise jitters for the VDLL and VFLL, respectively, are as follows:

$$\underline{\sigma}^-_{VDLL_{noise},\ell}(i)=\sqrt{\mathbf{G}^-_\ell(i,i)} \quad \text{and} \quad \underline{\sigma}^-_{VFLL_{noise},\ell}(\eta)=\sqrt{\mathbf{G}^-_\ell(\eta,\eta)} \tag{8}$$

where $\mathbf{G}^-_\ell=\mathbf{H}_\ell\Sigma^-_\ell\mathbf{H}^T_\ell$. Following the definition after Equation (5), the 1-σ total jitter of a channel is given by the sum of the noise jitter and one-third of the dynamic stress error, which separately accounts for the effect of the nonzero mean of $\underline{w}$. It should be noted that $\underline{w}$ is assumed to have zero mean, and its covariance, $\mathbf{Q}$, is only considered in the previous noise jitter calculation. Because the user acceleration is modeled as constant in the vector navigation filter, user jerk will result in a dynamic stress error. In this paper, the final jitter for loss-of-lock detection is determined later using both *a priori* and *a posteriori* jitters. It should be noted that a separate consideration is made for the contributions of process noise (or $\mathbf{Q}$) in the jitter and of dynamic stress error arising from jerk. The diagonal elements of $\mathbf{Q}$ corresponding to the user acceleration may have comparatively higher values than other elements so that the filter appropriately learns from its measurements during dynamics. However, the actual jerk need not be the same as what is expected from $\mathbf{Q}$. Hence, the dynamic stress error must be included. While considering the full $\mathbf{Q}$ in the noise jitter along with the dynamic stress error may result in an overbound of the total jitter, simulation results will show that this does not lead to a false detection of loss of lock. This is partially due to the fact that the contribution of the diagonal acceleration terms of $\mathbf{Q}$ to $\mathbf{G}^-_\ell$ via the addition of $\mathbf{Q}$ in Σ^-_ℓ is zero because the respective columns of $\mathbf{H}_\ell$ for the acceleration are zero (see Appendix A.1). Considering only the clock part of $\mathbf{Q}$ to separately update the noise jitter in a two-step process following the work of Lashley et al. (2009) would require fairly accurate knowledge of the clock parameters. This may not always be possible, and

parameters may change with time. Thus, as a conservative approach, the full **Q** is preferred in this paper to overbound the noise jitter.

The dynamic stress error is calculated at each epoch for a given user jerk profile, instead of finding the steady-state error for a constant jerk level. This approach is used to account for real scenarios in which the jerk need not remain constant and can change with time. The calculation of the dynamic stress error is described next. The *a priori* state error due to nonzero jerk, $\Delta\underline{\breve{x}}^{-}_{dyn,\ell}$, before the measurement update at ℓ is as follows:

$$\Delta\underline{\breve{x}}^{-}_{dyn,\ell} = \mathbf{F}\Delta\underline{\breve{x}}^{+}_{dyn,\ell-1} + \mathbf{B}\underline{u}_{jerk,\ell-1} \tag{9}$$

where $\Delta\underline{\breve{x}}^{+}_{dyn,\ell-1}$ is the cumulative state error due to previous jerks after the measurement update at $\ell-1$ and is a zero vector when $\ell=1$. $\mathbf{B}\underline{u}_{jerk,\ell-1}$ is the error introduced to the modeled user position, velocity, and acceleration at epoch ℓ due to user jerk, $\underline{u}_{jerk,\ell-1}$, at epoch $\ell-1$. $\underline{u}_{jerk,\ell-1}$ has three components given by $\left[ux_{jerk_{\ell-1}}\ uy_{jerk_{\ell-1}}\ uz_{jerk_{\ell-1}}\right]^T$, which are the ECEF *x*-, *y*-, and *z*-components of jerk, respectively. The first nine rows of **B** are calculated over the time interval Δt (= 0.1 s) as follows:

$$\mathbf{B}(1:9,\ :) = \mathbf{I}_{3\times3} \otimes \begin{bmatrix} \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}^T \tag{10}$$

The last six rows of **B** have zero elements corresponding to the GPS and NavIC receiver clock errors. Following Equations (4) and (5), the state error, $\Delta\underline{\breve{x}}^{+}_{dyn,\ell}$, after the measurement update and $\Delta\underline{\breve{x}}^{-}_{dyn,\ell}$ are as follows:

$$\Delta\underline{\breve{x}}^{+}_{dyn,\ell} = (\mathbf{I}-\mathbf{K}_\ell\mathbf{H}_\ell)\mathbf{F}\Delta\underline{\breve{x}}^{+}_{dyn,\ell-1} + (\mathbf{I}-\mathbf{K}_\ell\mathbf{H}_\ell)\mathbf{B}\underline{u}_{jerk,\ell-1} \tag{11}$$

$$\Delta\underline{\breve{x}}^{-}_{dyn,\ell} = \mathbf{F}(\mathbf{I}-\mathbf{K}_\ell\mathbf{H}_\ell)\Delta\underline{\breve{x}}^{-}_{dyn,\ell-1} + \mathbf{B}\underline{u}_{jerk,\ell-1} \tag{12}$$

The *a priori* dynamic stress error is $\underline{\epsilon}^{-}_{dyn,\ell} = \mathbf{H}_\ell\Delta\underline{\breve{x}}^{-}_{dyn,\ell}$, and the *a posteriori* dynamic stress error is $\underline{\epsilon}^{+}_{dyn,\ell} = \mathbf{H}_\ell\Delta\underline{\breve{x}}^{+}_{dyn,\ell}$. Because $(\mathbf{H}_\ell, (\mathbf{I}-\mathbf{K}_\ell\mathbf{H}_\ell)\mathbf{F})$ is observable and $((\mathbf{I}-\mathbf{K}_\ell\mathbf{H}_\ell)\mathbf{F}, (\mathbf{I}-\mathbf{K}_\ell\mathbf{H}_\ell))$ is stabilizable (see Appendix A.2 and A.3), the effect of $\mathbf{B}\underline{u}_{jerk,\ell-1}$ is reflected in $\underline{\epsilon}^{+}_{dyn,\ell}$. A similar justification holds for $\underline{\epsilon}^{-}_{dyn,\ell}$. Using Equation (8) and the dynamic stress error, the total *a priori* 1-σ SS jitters of a VDLL channel i and VFLL channel η before a measurement update are as follows:

$$\underline{\sigma}^{-}_{VDLL,\ell}(i) = \underline{\sigma}^{-}_{VDLL_{noise},\ell}(i) + \frac{|\underline{\epsilon}^{-}_{dyn,\ell}(i)|}{3} \tag{13}$$

$$\underline{\sigma}^{-}_{VFLL,\ell}(\eta) = \underline{\sigma}^{-}_{VFLL_{noise},\ell}(\eta) + \frac{|\underline{\epsilon}^{-}_{dyn,\ell}(\eta)|}{3} \tag{14}$$

where $|.|$ represents the absolute value. The "–" in the superscript of the preceding equations is replaced with "+" after a measurement update to represent the total *a posteriori* 1-σ SS jitters. It should be noted that these jitters are determined in this paper using theoretical models only. Next, the vector loop jitter is determined from its TF models.

2.2 | Transfer Function Jitter

Traditionally, the jitter is calculated for a scalar tracking loop based on its TF model. Following this approach, the TF jitter is determined for the vector architecture and is later used to validate the SS jitter. Becaue TFs require constant parameters, first, piecewise constant models of vector loops are developed over time intervals, each of duration T_{sub}, without decoupling the VDLL and VFLL. It should be noted that even when there is no dynamic stress error (i.e., for a static user or constant-velocity/acceleration user motion), the measurement model matrix, **H**, would change with time due to user-satellite relative motion, making the system model time-varying. **K** may also change owing to the adaptive calculation of **W** based on C/N_0. Therefore, piecewise constant models are required for the TF formulations and subsequent jitter calculations. An algorithm to adaptively calculate T_{sub} for a model is given in Appendix A.5. Over each interval of T_{sub} duration, **H, W,** and **K** are assumed to be constant. It should be noted that the minimum value of T_{sub} is 0.1 s, which is the measurement update interval. In this case, a TF model would last only 0.1 s. Using Equations (4) and (5), $\underline{e}^{+}_{\ell+1}$ and $\underline{e}^{-}_{\ell+1}$ for a given piecewise constant model q of duration T_{sub} reduce to the following:

$$\underline{e}^{+}_{\ell+1} = (\mathbf{I} - \mathbf{K}_q\mathbf{H}_q)\mathbf{F}\underline{e}^{+}_{\ell} + (\mathbf{I} - \mathbf{K}_q\mathbf{H}_q)\underline{w}_{\ell} - \mathbf{K}_q\underline{\zeta}_{\ell+1} \tag{15}$$

$$\underline{e}^{-}_{\ell+1} = (\mathbf{F} - \mathbf{K}'_q\mathbf{H}_q)\underline{e}^{-}_{\ell} + \underline{w}_{\ell} - \mathbf{K}'_q\underline{\zeta}_{\ell} \tag{16}$$

where $\mathbf{K}_q$ and $\mathbf{H}_q$ are the optimal Kalman gain and actual **H** matrices at the beginning of T_{sub} for the model q, respectively. These matrices are kept constant until a new piecewise constant model $(q+1)$ starts. The optimal **K** implies the Kalman gain obtained using the EKF equations with the actual time-varying model of the vector navigation filter. ℓ indicates the indices for the time instants of measurement updates, as before. Because the parameters of the preceding equations are constant, it is possible to find their z-transforms with respect to $\underline{w}$ and $\underline{\zeta}$ as follows:

$$\underline{e}^{+}_{q}(z) = \underline{e}^{+}_{1,q}(z) + \underline{e}^{+}_{2,q}(z) \quad \text{and} \quad \underline{e}^{-}_{q}(z) = \underline{e}^{-}_{1,q}(z) + \underline{e}^{-}_{2,q}(z) \quad \text{where} \tag{17}$$

$$\underline{e}^{+}_{1,q}(z) = (z\mathbf{I} - (\mathbf{I} - \mathbf{K}_q\mathbf{H}_q)\mathbf{F})^{-1}(\mathbf{I} - \mathbf{K}_q\mathbf{H}_q)\underline{w}(z), \quad \text{and}$$

$$\underline{e}^{+}_{2,q}(z) = -(z\mathbf{I} - (\mathbf{I} - \mathbf{K}_q\mathbf{H}_q)\mathbf{F})^{-1}\mathbf{K}_q z\underline{\zeta}(z)$$

$$\underline{e}^{-}_{1,q}(z) = (z\mathbf{I} - (\mathbf{F} - \mathbf{K}'_q\mathbf{H}_q))^{-1}\underline{w}(z), \quad \text{and} \quad \underline{e}^{-}_{2,q}(z) = -(z\mathbf{I} - (\mathbf{F} - \mathbf{K}'_q\mathbf{H}_q))^{-1}\mathbf{K}'_q\underline{\zeta}(z)$$

The preceding two equations provide the z-transforms, $\underline{e}^{+}_{q}(z)$ and $\underline{e}^{-}_{q}(z)$, of $\underline{e}^{+}$ and $\underline{e}^{-}$, respectively, for the piecewise constant model q. $\underline{e}_{1,q}(z)$ and $\underline{e}_{2,q}(z)$ represent the z-transform components with respect to $\underline{w}$ and $\underline{\zeta}$, respectively. The z-transforms of $\underline{\epsilon}_e$ and $\underline{\epsilon}_p$ (see after Equation (5) for a definition), $\underline{\epsilon}_e(z)$ and $\underline{\epsilon}_p(z)$, are $\mathbf{H}_q\underline{e}^{+}_{q}(z)$ and $\mathbf{H}_q\underline{e}^{-}_{q}(z)$, respectively. Because the z-transform is valid for a finite step size of 0.1 s, it is more appropriate for vector loops than the Laplace transform. The presented formulation does not require the decoupling of VDLL and VFLL, unlike the work of Bhattacharyya (2018), and also removes other key assumptions of the previous work. The z-domain TFs are used next to find the noise equivalent BWs and jitter.

2.2.1 | Noise Equivalent Bandwidths

The TF matrices relating $\underline{w}(z)$ and $\underline{\zeta}(z)$ to $\underline{\epsilon}_e(z)$ are denoted by $\mathbf{G}_{e_{1,q}}(z)$ and $\mathbf{G}_{e_{2,q}}(z)$, respectively. Similarly, the matrices for $\underline{\epsilon}_p(z)$ are $\mathbf{G}_{p_{1,q}}(z)$ and $\mathbf{G}_{p_{2,q}}(z)$. Using the previous two equations and the paragraph after them, the mathematical expressions of these matrices are summarized as follows:

$$\begin{aligned}\mathbf{G}_{e_{1,q}}(z) &= \mathbf{H}_q(z\mathbf{I}-(\mathbf{I}-\mathbf{K}_q\mathbf{H}_q)\mathbf{F})^{-1}(\mathbf{I}-\mathbf{K}_q\mathbf{H}_q),\\ \mathbf{G}_{e_{2,q}}(z) &= -z\mathbf{H}_q(z\mathbf{I}-(\mathbf{I}-\mathbf{K}_q\mathbf{H}_q)\mathbf{F})^{-1}\mathbf{K}_q\end{aligned} \tag{18}$$

$$\begin{aligned}\mathbf{G}_{p_{1,q}}(z) &= \mathbf{H}_q(z\mathbf{I}-(\mathbf{F}-\mathbf{K}'_q\mathbf{H}_q))^{-1},\\ \mathbf{G}_{p_{2,q}}(z) &= -\mathbf{H}_q(z\mathbf{I}-(\mathbf{F}-\mathbf{K}'_q\mathbf{H}_q))^{-1}\mathbf{K}'_q\end{aligned} \tag{19}$$

The dimensions of $\mathbf{G}_{e_{1,q}}(z)$ and $\mathbf{G}_{p_{1,q}}(z)$ are $2n\times 15$, and those of $\mathbf{G}_{e_{2,q}}(z)$ and $\mathbf{G}_{p_{2,q}}(z)$ are $2n\times 2n$. Any arbitrary TF matrix, $\mathbf{D}(z)$, of dimension $m\times r$ can be written as follows:

$$\mathbf{D}(z)=\begin{bmatrix} D_{11}(z) & D_{12}(z) & \cdots & D_{1r}(z)\\ D_{21}(z) & D_{22}(z) & \cdots & D_{2r}(z)\\ & \vdots & & \\ D_{m1}(z) & D_{m2}(z) & \cdots & D_{mr}(z)\end{bmatrix} \tag{20}$$

where $D_{i\eta}$ is the TF element in the i-th row and η-th column. Its one-sided noise equivalent BW is given by the following:

$$\mathrm{BW}^{i\eta}=\frac{1}{2\pi}\int_0^{\pi} D_{i\eta}(e^{j\omega})(D_{i\eta}(e^{j\omega}))^{*}\,d\omega \tag{21}$$

where $(D_{i\eta}(e^{j\omega}))^{*}$ is the complex conjugate transpose of $D_{i\eta}(e^{j\omega})$. $\mathrm{BW}^{i\eta}$ is the BW of a unity gain boxcar filter that will pass the same noise power as that of $D_{i\eta}(e^{j\omega})$. Note that the direct current gain is not divided in the preceding equation, as different TF elements need not have the same gain (Bhattacharyya, 2018). Thus, $\mathrm{BW}^{i\eta}$ is given by the square of the H_2 norm of $D_{i\eta}(e^{j\omega})$ divided by 2 (i.e., $\mathrm{BW}^{i\eta}=\|D_{i\eta}\|^2/2$). In MATLAB, the noise equivalent BWs are calculated for a discrete step size of 0.1 s. Algorithm 1 calculates the BWs for a generic TF matrix, $\mathbf{D}(z)=\mathbf{P}(z\mathbf{I}-\mathbf{A}_1)^{-1}\mathbf{B}_1$, of dimension $m\times r$, where $\mathbf{P}$, $\mathbf{A}_1$, and $\mathbf{B}_1$ are matrices of appropriate dimensions.

Because there are two input vectors (process noise, $\underline{w}$, and measurement noise, $\underline{\zeta}$), BWs are determined for each input vector using the corresponding TF matrix. These BWs are called the process noise and measurement noise BWs, respectively. The $\mathrm{BW}^{i\eta}$ terms for $\mathbf{G}_{e_{1,q}}(z), \mathbf{G}_{e_{2,q}}(z), \mathbf{G}_{p_{1,q}}(z)$, and $\mathbf{G}_{p_{2,q}}(z)$ are denoted by $\mathrm{BW}^{i\eta}_{e_{1,q}}$, $\mathrm{BW}^{i\eta}_{e_{2,q}}$, $\mathrm{BW}^{i\eta}_{p_{1,q}}$, and $\mathrm{BW}^{i\eta}_{p_{2,q}}$, respectively. Next, the TF noise jitter and dynamic stress error are calculated.

ALGORITHM 1
Noise BW calculation

for $\eta = 1{:}r$ % column index of the TF matrix, $\mathbf{D}(z) = \mathbf{P}(z\mathbf{I} - \mathbf{A}_1)^{-1}\mathbf{B}_1$
  % Find the zeros, poles, and gains of the TF matrix, $\mathbf{D}(z)$, from its state-space representation
  % for the η-th input
  [zero, pole, gain] = ss2zp($\mathbf{A}_1$, $\mathbf{B}_1$, $\mathbf{P}$, zero matrix of dimension $m \times r$, η); % see MATLAB help
  % for details regarding "ss2zp"
  for $i = 1{:}m$ % row index of the TF matrix, $\mathbf{D}(z)$
    % Find the TF element, $D_{i\eta}$, in the i-th row and η-th column in pole-zero form for a
    % discrete step size of 0.1 s
    $D_{i\eta}$ = zpk(zero(:, i), pole, gain(i), 0.1); % zpk(.) finds the pole-zero form
    % Calculate the noise BW of $D_{i\eta}$ as $\mathrm{BW}^{i\eta} = ||D_{i\eta}||^2/2$
    $\mathrm{BW}^{i\eta}$ = (norm($D_{i\eta}$) × norm($D_{i\eta}$))/2; % norm(.) calculates the H_2 norm of the TF, $D_{i\eta}$
  end
end

2.2.2 | *Noise Jitter and Dynamic Stress Error*

Considering both process and measurement noise BWs, the *a posteriori* noise jitters of a VDLL channel i and VFLL channel η for the q-th TF model are given as follows:

$$\sigma^{+}_{VDLL,\,noise_{tf}}(i,q) = \left(2\sum_{l=1}^{15}\mathrm{BW}^{il}_{e_{1,q}}\mathbf{Q}(l,l) + 2\sum_{l=1}^{2n}\mathrm{BW}^{il}_{e_{2,q}}\mathbf{W}_q(l,l)\right)^{1/2}, \quad i = 1,\ldots,n \tag{22}$$

$$\sigma^{+}_{VFLL,\,noise_{tf}}(\eta,q) = \left(2\sum_{l=1}^{15}\mathrm{BW}^{\eta l}_{e_{1,q}}\mathbf{Q}(l,l) + 2\sum_{l=1}^{2n}\mathrm{BW}^{\eta l}_{e_{2,q}}\mathbf{W}_q(l,l)\right)^{1/2}, \quad \eta = (n+1),\ldots,2n \tag{23}$$

where $\mathbf{Q}$ is of dimension 15×15. $\mathbf{W}_q$ is the measurement error covariance of the q-th TF model obtained adaptively based on C/N_0 when the q-th model starts and is held constant until the start of the next model. It is evident that unlike scalar tracking, the noise jitter of a channel depends on the input noises of all channels. The *a priori* noise jitters of the same VDLL and VFLL channels are as follows:

$$\sigma^{-}_{VDLL,\,noise_{tf}}(i,q) = \left(2\sum_{l=1}^{15}\mathrm{BW}^{il}_{p_{1,q}}\mathbf{Q}(l,l) + 2\sum_{l=1}^{2n}\mathrm{BW}^{il}_{p_{2,q}}\mathbf{W}_q(l,l)\right)^{1/2}, \quad i = 1,\ldots,n \tag{24}$$

$$\sigma^{-}_{VFLL,\,noise_{tf}}(\eta,q) = \left(2\sum_{l=1}^{15}\mathrm{BW}^{\eta l}_{p_{1,q}}\mathbf{Q}(l,l) + 2\sum_{l=1}^{2n}\mathrm{BW}^{\eta l}_{p_{2,q}}\mathbf{W}_q(l,l)\right)^{1/2}, \quad \eta = (n+1),\ldots,2n \tag{25}$$

These equations can be derived in the following way. If $\mathbf{D}(e^{j\omega})$ is a TF matrix and $\mathbf{R}$ is a diagonal input noise covariance, then the output noise variance of channel i is $\mathbf{C}(i,i)$, where $\mathbf{C}$ is as follows:

$$\mathbf{C} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathbf{D}(e^{i\omega})\mathbf{R}(\mathbf{D}(e^{j\omega}))^{*}\,d\omega \tag{26}$$

The measurement noise covariance, $\mathbf{W}_q$, is assumed to be a diagonal matrix, determined based on C/N_0 and discriminator functions. The process noise covariance, $\mathbf{Q}$, may have an off-diagonal element because of the receiver clock error,

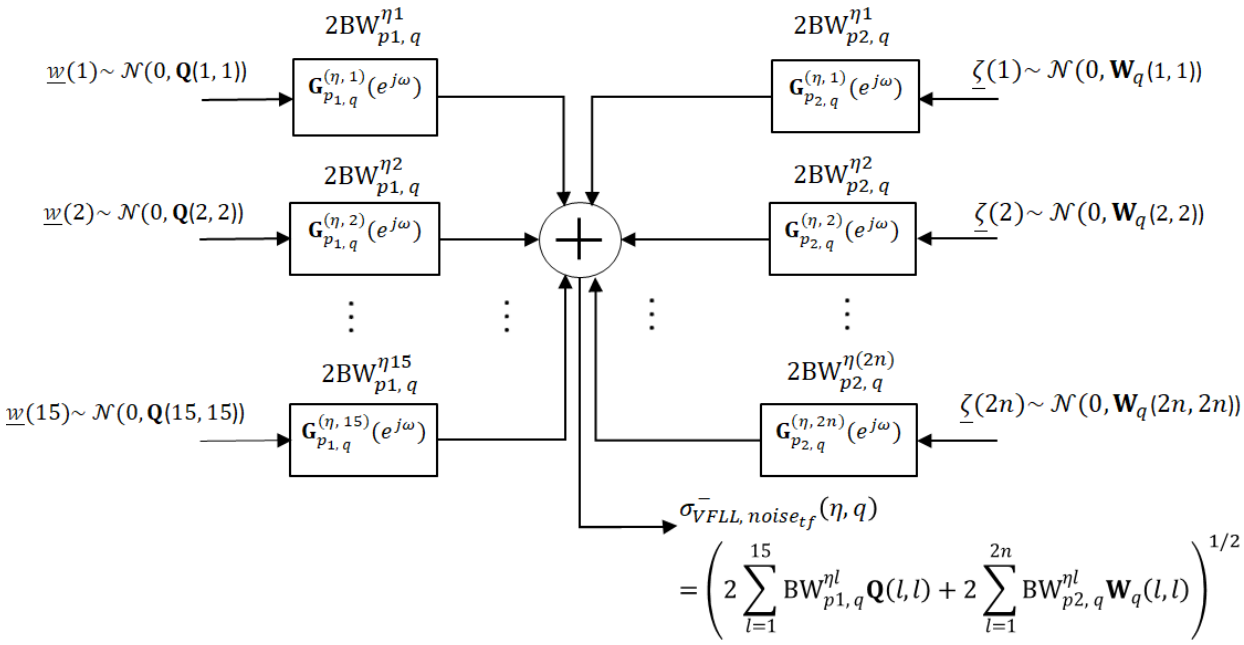

FIGURE 2 Graphical representation of noise propagation from input to output through the q-th TF model for the *a priori* VFLL noise jitter in channel η
The superscripts of $\mathbf{G}_{p_{1,q}}$ and $\mathbf{G}_{p_{2,q}}$ denote the row and column indices. $\mathcal{N}(\ ,\ )$ represents Gaussian distribution, with the mean and variance shown in parentheses.

which is given by $2\pi^2 h_{-2}c^2(\Delta t)^2/2$ (Brown & Hwang, 2012) ($\sim 1.77\times10^{-4}$ m^2/s where, h_{-2} denotes the h_{-2} parameter of clock noise, having a numerical value equal to 2×10^{-20} for a low-quality temperature compensated crystal oscillator (TCXO) (Curran et al., 2012a), $\Delta t = 0.1$ s, and c is the speed of light). However, the effect of this term is not considered in the TF noise jitter, as it is small. The off-diagonal terms of $\mathbf{Q}$ for the user position, velocity, and acceleration are considered to be zero in this paper, thus allowing Equations (22)–(25) to be valid. A diagonal $\mathbf{Q}$ is also reasonable for an update interval of 0.1 s (Groves, 2013). Although neglecting the off-diagonal elements of $\mathbf{Q}$ does not compromise performance for the simulation studies in this paper, these elements will be accounted for in future work. Figure 2 depicts how input noises $\underline{w}$ and $\underline{\zeta}$ propagate through the q-th TF model and contribute to the *a priori* noise jitter at the output of VFLL channel η. It should be noted that the TF noise jitter provides steady-state values. It is not possible to capture any transient effects in this method. The dynamic stress error is determined next. Unlike the noise jitter, this formulation considers transient effects. Transients can arise from changes in TF models and user motion.

To calculate the dynamic stress error, it is assumed that the user jerk profile (i.e., $\underline{u}_{jerk}$) is known for every measurement update interval of 0.1 s and remains constant over 0.1 s. The state errors due to jerk before and after measurement updates, $\Delta\underline{\varkappa}^{-}_{dyn,tf}$ and $\Delta\underline{\varkappa}^{+}_{dyn,tf}$, are propagated every 0.1 s through the q-th TF model of duration T_{sub} using the following equations:

$$\Delta\underline{\varkappa}^{+}_{dyn,tf,\ell} = (\mathbf{I}-\mathbf{K}_q\mathbf{H}_q)\mathbf{F}\Delta\underline{\varkappa}^{+}_{dyn,tf,\ell-1} + (\mathbf{I}-\mathbf{K}_q\mathbf{H}_q)\mathbf{B}\underline{u}_{jerk,\ell-1} \tag{27}$$

$$\Delta\underline{\varkappa}^{-}_{dyn,tf,\ell} = (\mathbf{F}-\mathbf{K}'_q\mathbf{H}_q)\Delta\underline{\varkappa}^{-}_{dyn,tf,\ell-1} + \mathbf{B}\underline{u}_{jerk,\ell-1} \tag{28}$$

Considering $\mathbf{B}\underline{u}_{jerk}$ to be the input vector, the TF matrices for $\Delta\underline{\varkappa}^{-}_{dyn,tf}$ and $\Delta\underline{\varkappa}^{+}_{dyn,tf}$ are $\mathbf{G}^{-}_{dyn,q}(z)=(z\mathbf{I}-(\mathbf{F}-\mathbf{K}'_q\mathbf{H}_q))^{-1}$ and $\mathbf{G}^{+}_{dyn,q}(z)=(z\mathbf{I}-(\mathbf{I}-\mathbf{K}_q\mathbf{H}_q)\mathbf{F})^{-1}(\mathbf{I}-\mathbf{K}_q\mathbf{H}_q)$, respectively. $\Delta\underline{\varkappa}^{-}_{dyn,tf,\ell}$ is calculated at each time instant ℓ as the sum of the impulse response of $(z\mathbf{I}-(\mathbf{F}-\mathbf{K}'_q\mathbf{H}_q))^{-1}$ due to $\Delta\underline{\varkappa}^{-}_{dyn,tf,\ell-1}$ (see the first term on the right-hand side of Equation (28)) and the step response of $(z\mathbf{I}-(\mathbf{F}-\mathbf{K}'_q\mathbf{H}_q))^{-1}$ over 0.1 s due to $\mathbf{B}\underline{u}_{jerk,\ell-1}$ only. $\Delta\underline{\varkappa}^{+}_{dyn,tf,\ell}$ is determined in a similar manner. Then, the corresponding dynamic stress errors, $\underline{\epsilon}^{-}_{dyn,tf,\ell}$ and $\underline{\epsilon}^{+}_{dyn,tf,\ell}$, are obtained by multiplying $\mathbf{H}_q$ by $\Delta\underline{\varkappa}^{-}_{dyn,tf,\ell}$ and $\Delta\underline{\varkappa}^{+}_{dyn,tf,\ell}$, respectively.

The total *a posteriori* 1-σ TF jitters of VDLL channel i and VFLL channel η after a measurement update for the q-th TF model are as follows:

$$\underline{\sigma}^{+}_{VDLL,tf,\ell}(i,q)=\underline{\sigma}^{+}_{VDLL,noise_{tf}}(i,q)+\frac{|\underline{\epsilon}^{+}_{dyn,tf,\ell}(i)|}{3} \tag{29}$$

$$\underline{\sigma}^{+}_{VFLL,tf,\ell}(\eta,q)=\underline{\sigma}^{+}_{VFLL,noise_{tf}}(\eta,q)+\frac{|\underline{\epsilon}^{+}_{dyn,tf,\ell}(\eta)|}{3} \tag{30}$$

Likewise, the *a priori* jitter can be calculated before measurement updates. Vector loop jitter thresholds and a collective metric to represent overall performance are described next. The thresholds and metric are then used in simulation studies to theoretically predict the conditions that would lead to loss of lock.

3 | COLLECTIVE JITTER METRIC FOR LOSS-OF-LOCK DETECTION

In the previous formulations, the *a priori* and *a posteriori* vector loop jitters (both SS and TF) are calculated every 0.1 s, considering thermal noise and receiver clock and dynamic stress errors. In this section, a loss-of-lock detection mechanism is designed using the SS jitter described in Section 2.1. Following the work of Ward et al. (2006), the 3-σ jitter threshold for loss-of-lock detection is $d/2$ for code phase tracking, where d is the early–late correlator spacing (= 1 chip in this paper). For carrier frequency tracking, the threshold is $1/(4T)$ for the atan2 discriminator, where T is the coherent integration time. These values are proposed as conservative rule-of-thumb thresholds in the work by Ward et al. (2006), presumably to provide a safe margin from the noise-free pull-in range of the respective discriminators (Curran et al., 2012b; Curran, 2015). If the jitter exceeds its threshold, it is likely that the noisy discriminator output will also cross its pull-in range. It is important to note that the prescribed jitter threshold for carrier frequency tracking explicitly depends on T. In the vector loop implementation of this paper, T is 20 ms for the VDLL and 10 ms for the VFLL, as two 10-ms prompt correlator pairs are needed to form the carrier frequency discriminator outputs. Hence, for loss-of-lock detection, vector loop jitters of all channels are calculated every 20 ms so that they are obtained at the discriminator and code/carrier update rate, which is also the prediction step size of the navigation filter, and the aforementioned threshold can be applied in a manner similar to that of scalar tracking. In addition, the operations of channels are coupled in vector tracking. Thus, a collective metric accounting for the jitters of all channels is required. For this purpose, appropriate jitter bounds are proposed next.

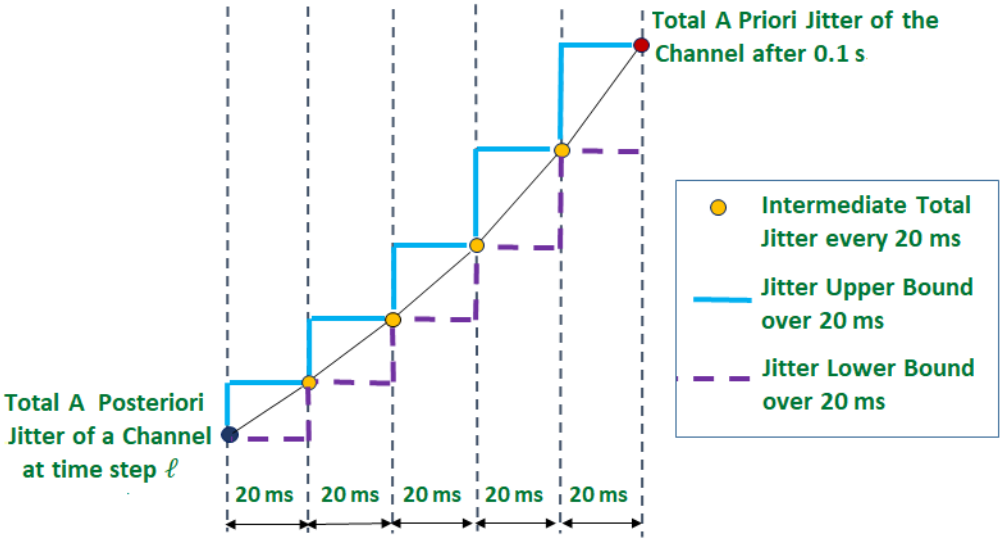

FIGURE 3 Illustration of jitter lower and upper bounds for a channel over 20-ms intervals For a decreasing trend, the bounds will be interchanged. That is, there are two bounds: one corresponds to the beginning of a 20-ms interval and the other corresponds to the end.

Figure 3 illustrates the concept of jitter lower and upper bounds for a given channel. As shown in the figure, the time interval Δt (= 0.1 s) is divided into five sub-intervals of 20 ms each. The jitter is calculated every 20 ms using an approach similar to that for determining the SS jitter discussed earlier. The thick solid horizontal step represents the jitter upper bound for a channel over a 20-ms interval, whereas the dashed horizontal step indicates the lower bound. The predicted jitter is expected to be contained within the two bounds. Algorithm 2 describes the calculation of the jitter bounds every 20 ms over a 0.1-s interval.

The final jitter upper bound of a channel every 0.1 s is the average of five jitter upper bounds, each corresponding to the end of 20 ms, following the convention of Figure 3. Similarly, the final lower bound is the average of five lower bounds, each at the beginning of 20 ms. The reason for averaging the bounds is that if the majority of the five upper bounds over 0.1 s cross the threshold, their average is likely to cross the threshold as well. The mean of the averaged bounds across all VFLL channels and across all VDLL channels is also determined. First, the mean is separately calculated over all GPS channels and all NavIC channels, and then the two means are averaged. The averaged mean is called the mean jitter upper or lower bound for the VFLL (or VDLL). The mean bound is considered a collective jitter metric, reflecting overall vector loop performance, and is compared against the jitter threshold mentioned earlier for loss-of-lock detection every 0.1 s. The effectiveness of the mean jitter upper bound for detecting loss of lock will be shown in Section 4. It should be noted that for a decreasing jitter over 0.1 s (i.e., the reverse trend of Figure 3), the bound at the beginning of 20 ms becomes the upper bound. Because it would not be known if the jitter increases or decreases over a 0.1 s interval, both bounds at the beginning and end of 20 ms are calculated. The overall sequence of operations for calculating the SS and TF jitters and the jitter metric is described in Appendix A.6. In this paper, these terms are calculated offline. Simulation results are presented next.

ALGORITHM 2
Calculation of jitter bounds every 20 ms

% Calculation of noise jitter every 20 ms between time steps ℓ and $\ell + 1$

% Perform a measurement update of the *a priori* state error covariance:

$\mathbf{\Sigma}_{\ell}^{+} = (\mathbf{I} - \mathbf{K}_{\ell}\mathbf{H}_{\ell})\mathbf{\Sigma}_{\ell}^{-}(\mathbf{I} - \mathbf{K}_{\ell}\mathbf{H}_{\ell})^{T} + \mathbf{K}_{\ell}\mathbf{W}_{\ell}\mathbf{K}_{\ell}^{T}$;

$\mathbf{\Sigma}_{temp,1} = \mathbf{\Sigma}_{\ell}^{+}$; % Save $\mathbf{\Sigma}_{\ell}^{+}$ to a temporary variable for jitter upper bound calculation

$\mathbf{\Sigma}_{temp,2} = \mathbf{\Sigma}_{\ell}^{+}$; % Save $\mathbf{\Sigma}_{\ell}^{+}$ to another temporary variable for jitter lower bound calculation

for index = 1:5

  % Propagation of state error covariance over 20 ms for the upper bound (following the

  % convention of Figure 3)

  $\Sigma_{temp,1} = \mathbf{F}'\Sigma_{temp,1}(\mathbf{F}')^{T}$;

  $\mathbf{\Sigma}_{temp,3} = \Sigma_{temp,1}$ + (index)*$\mathbf{Q}$/5; % One-fifth of the process noise covariance $\mathbf{Q}$ is added every

  % 20 ms

  $\mathbf{G}_{temp} = \mathbf{H}_{\ell}\mathbf{\Sigma}_{temp,3}\mathbf{H}_{\ell}^{T}$; % Project $\mathbf{\Sigma}_{temp,3}$ to code phase and Doppler domains

  temp = diag($\mathbf{G}_{temp}$); % Save the diagonal terms to an array, temp

  noise_jitter_ub(:, index) = sqrt(temp); % 1-σ noise jitter upper bounds of all channels at

  % the end of every 20 ms

  % Propagation of state error covariance for the lower bound

  $\mathbf{\Sigma}_{temp,3} = \mathbf{\Sigma}_{temp,2}$ + (index − 1)*$\mathbf{Q}$/5;

  $\mathbf{\Sigma}_{temp,2} = \mathbf{F}'\Sigma_{temp,2}(\mathbf{F}')^{T}$; % For the next iteration

  $\mathbf{G}_{temp} = \mathbf{H}_{\ell}\mathbf{\Sigma}_{temp,3}\mathbf{H}_{\ell}^{T}$; % Project $\mathbf{\Sigma}_{temp,3}$ to code phase and Doppler domains

  temp = diag($\mathbf{G}_{temp}$); % Save the diagonal terms to an array, temp

  noise_jitter_lb(:, index) = sqrt(temp); % 1-σ noise jitter lower bounds of all channels at

  % the beginning of every 20 ms

end

% Calculation of dynamic stress error every 20 ms

% Calculate $\Delta\underline{\varkappa}_{dyn,\ell}^{+}$ and $\Delta\underline{\varkappa}_{dyn,\ell+1}^{-}$ using Equations (11) and (12), respectively

% Calculate *a posteriori* and *a priori* dynamic stress errors at ℓ and $\ell+1$, respectively

$\underline{\epsilon}_{dyn,\ell}^{+} = \mathbf{H}_{\ell}\Delta\underline{\varkappa}_{dyn,\ell}^{+}$ % *A posteriori* dynamic stress error

$\underline{\epsilon}_{dyn,\ell+1}^{-} = \mathbf{H}_{\ell}\Delta\underline{\varkappa}_{dyn,\ell+1}^{-}$ % *A priori* dynamic stress error; $\mathbf{H}$ does not change substantially

% over 0.1 s

% Assuming a linear variation of the error over 0.1 s, a slope is calculated. Note that the

% absolute value is taken:

$$\text{slope} = \frac{|\underline{\epsilon}_{dyn,\ell+1}^{-}|/3 - |\underline{\epsilon}_{dyn,\ell}^{+}|/3}{0.1};$$

% Calculate the 1-σ total jitters of both VDLL and VFLL channels at the beginning and end

% of every 20 ms

for index = 1:5

  % 1-σ upper bounds

  % Note that the last iteration gives a jitter identical to the total *a priori* jitter after 0.1 s

  % (i.e., Equations (13) and (14))

  jitter_ub(:, index) = noise_jitter_ub(:, index) + ($|\underline{\epsilon}_{dyn,\ell}^{+}|/3$ + slope*0.02*index);

  % 1-σ lower bounds (Note that the first iteration gives the total *a posteriori* 1-σ jitter at ℓ)

  jitter_lb(:, index) = noise_jitter_lb(:, index) + ($|\underline{\epsilon}_{dyn,\ell}^{+}|/3$ + slope*0.02*(index − 1));

end

$\mathbf{\Sigma}_{\ell+1}^{-} = \mathbf{\Sigma}_{temp,1} + \mathbf{Q}$ % *a priori* state error covariance for time step $(\ell+1)$

4 | SIMULATION STUDIES

GPS L1 and NavIC L5 signals were simulated for a high-dynamic unmanned aerial vehicle (UAV) trajectory using Accord's GNSS simulator (Accord Software and Systems, 2024). The generated signals were recorded by an IFEN intermediate-frequency (IF) data recorder with a 20-MHz sampling frequency. The recorded IF data were post-processed by a software vector receiver with the models discussed in the beginning of Section 2. The simulator introduced ionospheric and tropospheric delays using the Klobuchar and RTCA98 models, respectively. The receiver removed these errors using the Klobuchar and Hopfield models. The UAV trajectory, skyplot, and variations in jerk magnitudes are depicted in Figure 4. The jerk can be as high as 16 g/s, lasting approximately 20 s at two separate instants during the maneuver. Other jerk magnitudes are also labeled in the figure. While the signals were simulated at 42–44 dB-Hz, artificial white noise was injected into the recorded IF data to create low-C/N_0 environments. Thus, low C/N_0 along with high dynamics creates scenarios ideal for vector loop jitter analysis.

The measurement noise covariance, $\mathbf{W}$, is adaptively determined based on C/N_0. Two sets of process noise covariance, denoted by $\mathbf{Q}_1$ and $\mathbf{Q}_2$, are considered in this paper for loss-of-lock analysis with different jerks. For the user position, velocity, and acceleration states along the ECEF x-, y-, and z-axes, the covariances are diag([0.5 0.2 8 0.5 0.2 8 0.7 0.2 8]) (used for all C/N_0 values) and diag([0.5 0.2 0.1 0.5 0.2 0.5 0.7 0.2 0.2]) (used only for C/N_0 values of 30 dB-Hz or lower), respectively, where diag() is a diagonal matrix. The first

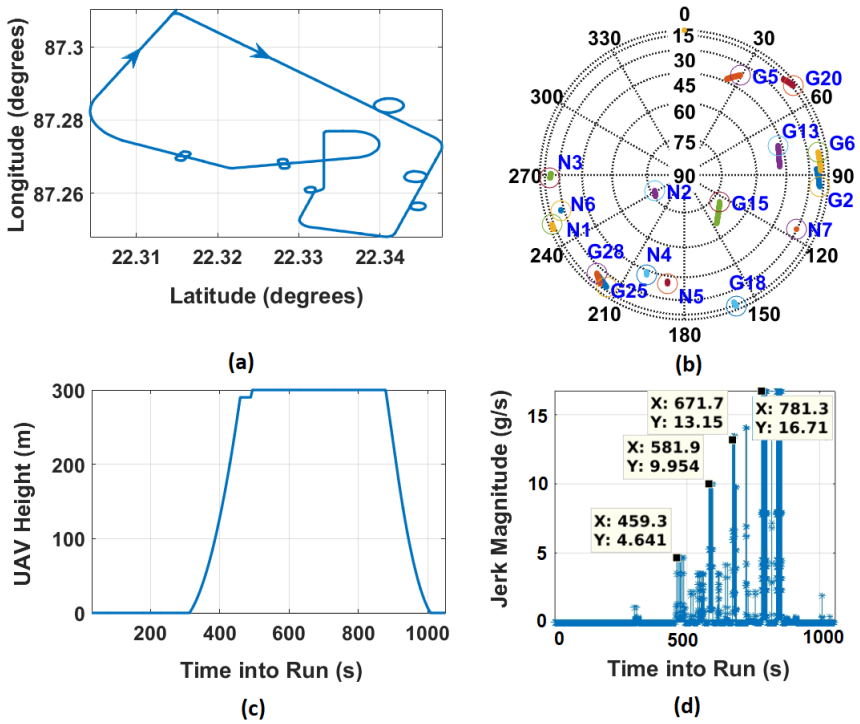

FIGURE 4 Simulated user trajectory, skyplot, and user jerk magnitude
"G" denotes GPS, and "N" denotes NavIC satellites in the skyplot. Circles indicate the end of a satellite trace.

three elements are used for the x-components of the user position, velocity, and acceleration in m^2, $(m/s)^2$, and $(m/s^2)^2$, respectively; the next three are for the y-components, and the last three are for the z-components. Once set, these elements are not changed with C/N_0 or dynamics. For $\mathbf{Q}_1$, the clock bias and drift states are calculated using the h_0 and h_{-2} parameters of a TCXO (Brown & Hwang, 2012). The portion of the $\mathbf{Q}_1$ matrix for the receiver clock bias (in m^2), drift (in $(m/s)^2$), and drift rate (in $(m/s^2)^2$) (same for both GPS and NavIC) is $\mathbf{Q}_{clk} = \begin{bmatrix} q_\varphi \Delta t + q_\omega (\Delta t)^3/3 & q_\omega (\Delta t)^2/2 & 0 \\ q_\omega (\Delta t)^2/2 & q_\omega (\Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Here, $q_\varphi = \frac{h_0}{2} c^2$ and $q_\omega = 2\pi^2 h_{-2} c^2$, the numerical values of the h_0 and h_{-2} clock parameters are 1×10^{-21} and 2×10^{-20}, respectively, and c is the speed of light. For $\mathbf{Q}_2$, at a mean C/N_0 (i.e., the average of all channel values) of 30 dB-Hz, $\mathbf{Q}_{clk}$ is taken as diag([3 2.5 1]) to adequately track dynamics. It is not changed until 21 dB-Hz. For a mean C/N_0 at or below 20 dB-Hz, $\mathbf{Q}_{clk}$ of $\mathbf{Q}_2$ is considered diag([3 1.5 1]) to stabilize the filter even without high dynamics. A larger $\mathbf{Q}_{clk}$ is chosen for $\mathbf{Q}_2$ than that for $\mathbf{Q}_1$ to compensate for small acceleration terms that are not adequate during high dynamics. It should be noted that $\mathbf{Q}_1$ (except the clock bias and drift terms of $\mathbf{Q}_{clk}$) and $\mathbf{Q}_2$ are chosen by trial and error during the initial filter tuning process to enable the vector loops to track the user motion trajectory in Figure 4. Both sets of $\mathbf{Q}$ ensure that the filter learns from its measurements and that the *a posteriori* position, velocity, and acceleration errors are bounded by the respective 3-σ bounds at steady state in the absence of user jerk across the different levels of C/N_0 considered in this paper. For $\mathbf{Q}_1$, the acceleration terms are kept high to account for propagation model inaccuracies during high dynamics with jerk, and the position and velocity terms have small, nominal values. For $\mathbf{Q}_2$, the acceleration terms are small and non-ideal. These non-ideal acceleration terms and those for the receiver clock in $\mathbf{Q}_2$ result in a loss of lock at different jerks. While $\mathbf{Q}_1$ is shown to better handle dynamics than $\mathbf{Q}_2$, the latter effectively demonstrates that jitter can detect loss of lock at different jerks in the same UAV simulation. The results also show how the total jitter changes with two different $\mathbf{Q}$ matrices, and for both cases, the loss-of-lock detection by the jitter is validated by the divergence of estimated states in the simulation.

Before loss-of-lock analysis, however, it is important to compare the SS and TF jitters and to show that jitter can predict the steady-state errors in the code and carrier channels in the presence of noise alone. To this end, the SS jitter is first validated with the TF jitter after initial transients of the filter have died down with $\mathbf{Q}_1$ process noise covariance. The validation results of both the total *a priori* and *a posteriori* SS and TF jitters are shown in Table 1, where the mean ± 3-σ values of the two jitter differences across all VDLL and VFLL channels and time steps are reported for different ranges of C/N_0. It should be noted that the C/N_0 ratios are reduced from the default range of (42–44) dB-Hz at 126.1 s into the run and then restored at 1026.1 s. The scenario is run multiple times, each time with one set of C/N_0 noted in the table. The difference between the SS and TF jitters of a channel is expressed as a percentage of the TF jitter before the mean and standard deviation have been calculated. For a fair comparison, it is ensured that both the SS and TF jitters use the same piecewise constant models (see the beginning of Section 2.2); the jitters are then calculated theoretically but using the $\mathbf{K}$, $\mathbf{H}$, and $\mathbf{W}$ matrices obtained from the vector loop runs corresponding to each model. The jitter difference is calculated every T_{sub} s (i.e., at the end of a constant model) so that the transient effects due to the model change captured by the SS jitter are reduced. The results in the table indicate that both the SS noise jitter and dynamic

TABLE 1
Mean ± 3-σ Value of the Difference between the SS and TF Jitters, Calculated over All Channels and Time Steps
The difference is expressed as a percentage of the corresponding TF jitter.

Range of C/N_0 (dB-Hz)	Mean ± 3-σ value of the difference between the total *a posteriori* SS and TF jitters (%)	Mean ± 3-σ value of the difference between the total *a priori* SS and TF jitters (%)
42–44	0.02484 ± 0.7139	0.0139 ± 1.0486
38.6–40	0.0177 ± 1.2878	0.01095 ± 1.69
33.5–35	0.0511 ± 3.761	0.03657 ± 4.456
28.5–31	0.0732 ± 4.71	0.0568 ± 5.203

stress error are in good agreement with their TF counterparts, thus establishing an equivalence between two differently calculated jitters and validating the use of the SS jitter. The larger differences at lower C/N_0 can be attributed to the following factors: 1. The transition from high C/N_0 to relatively low C/N_0 at 126.1 s and vice versa at 1026.1 s results in transients, which take some time to subside. The TF noise jitter captures steady-state values, but the SS noise jitter reflects transients. 2. T_{sub} is primarily 0.1 s at low C/N_0, indicating that the TF models are changed just after 0.1 s. While changes can cause minor transients, these effects would not be evident in the TF noise jitter.

Having established an equivalence between the SS and TF jitters, the steady-state jitter performance is analyzed at different noise levels in the absence of dynamic stress error with 16 visible satellites. For this analysis, the duration of data between 330 s and 450 s into the run is considered when the UAV moves with a small constant acceleration (i.e., no jerk) and the filter has reached steady state. More precisely, it is evaluated if the jitter is able to predict the code phase and Doppler error variations. The selected duration is sufficiently long to study the steady-state performance after C/N_0 has been reduced.

For a stringent comparison, the code phase and carrier Doppler errors of a channel at the time of reception, t_r, are determined differently from that defined after Equation (5), which forms the basis for the jitter calculation. If ρ_{tr} is the true pseudorange of a channel at t_r and ρ_{pr} is the predicted pseudorange of the vector receiver, these quantities are given by the following:

$$\rho_{tr} = c(t_r - t_t^{\text{true}}) \tag{31}$$

$$\rho_{pr} = c(t_r - t_t^{\text{pred}}) \tag{32}$$

where t_t^{true} is the true time of transmission and t_t^{pred} is its receiver-predicted value. The difference between the two pseudoranges is as follows:

$$\rho_{tr} - \rho_{pr} = c(t_t^{\text{pred}} - t_t^{\text{true}}) \tag{33}$$

Because the transmitted code of a GNSS satellite gives a fine component of the signal transmission time, the code phase alignment error of the receiver results in an error in the calculation of the time. Thus, the *a priori* code phase error of the vector receiver in meters would be $(\rho_{tr} - \rho_{pr})$. This error is expected to be bounded by the total *a priori* 3-σ code phase jitter, whose 1-σ value is given by Equation (13) after conversion to meters. It should be noted that the dynamic stress error component can be neglected in the absence of jerk. Similarly, the *a priori* version of Equation (29) should be considered for the TF jitter. The *a priori* carrier Doppler

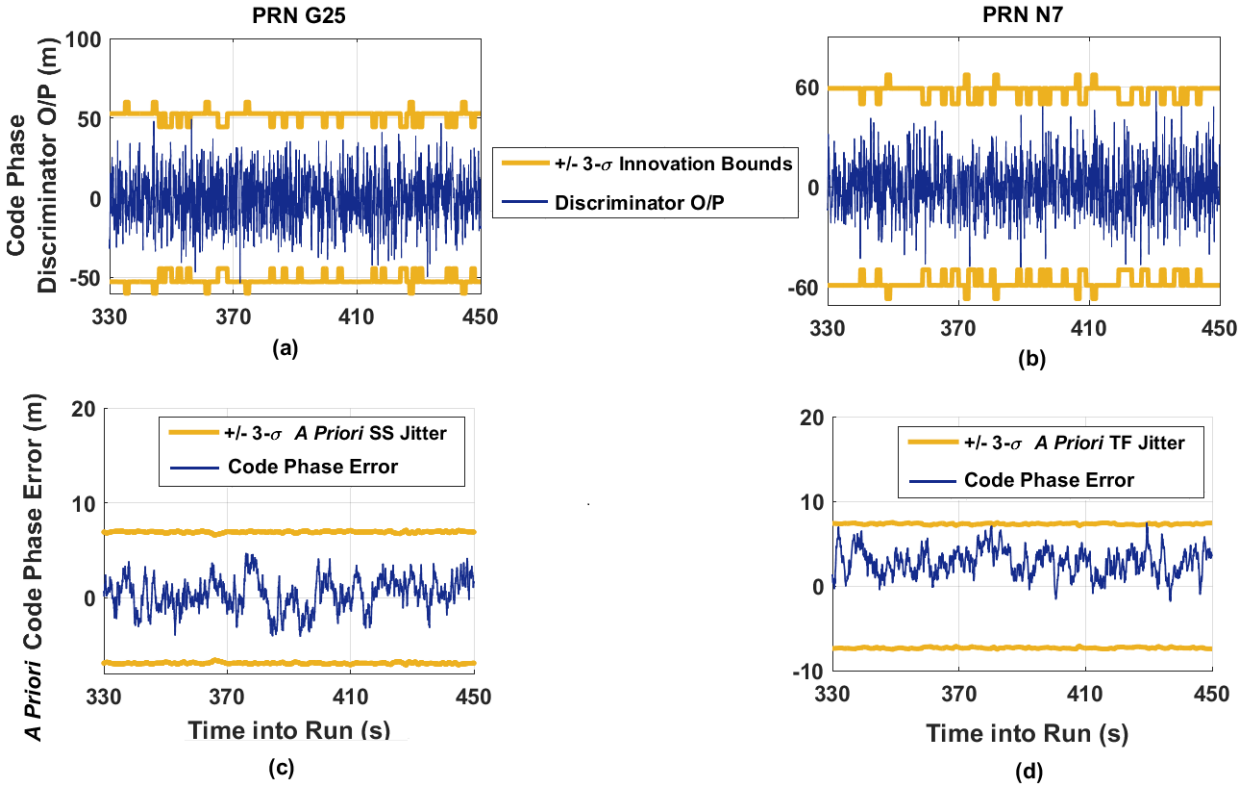

FIGURE 5 Steady-state code phase jitter performance in two channels at a mean C/N_0 of 30 dB-Hz
Code phase discriminator outputs (O/P) are also shown along with $\pm 3\text{-}\sigma$ bounds. Panel (c) depicts the SS jitter, and panel (d) presents the TF jitter, as obtained from their respective vector loop runs.

error of a channel is given as the difference between the receiver-predicted Doppler and true Doppler, which should be bounded by the 3-σ value of the total jitter given by Equation (14). Likewise, the *a priori* TF jitter can be calculated following Section 2.2.

For a satellite, the true pseudorange measurement and its rate (needed for the Doppler calculation) are obtained from the simulator log files every 0.1 s, considering all error sources except the receiver clock bias and drift. Because the clock terms are not known, their estimates by the vector receiver at high C/N_0 ratios of (42–44) dB-Hz are assumed to be true and added to the measurements derived from the simulator files at appropriate times after extrapolation. The code phase and carrier Doppler errors of select GPS and NavIC satellites calculated as explained before, alongside the total *a priori* SS and TF jitters, are presented in Figures 5 and 6 for a mean C/N_0 of 30 dB-Hz. Both figures use the $\mathbf{Q}_1$ process noise covariance. The code phase and carrier frequency discriminator outputs, which are the receiver's noisy estimates of the code and carrier errors, are also illustrated along with their respective $\pm$ 3-σ bounds. Because the discriminator outputs are the innovation vector of the navigation filter, their bounds are determined from the diagonal elements of the innovation covariance matrix. In both figures, panel (c) shows the SS jitter whereas panel (d) shows the TF jitter.

For the results in panels (b) and (d) of Figures 5 and 6, the piecewise constant models discussed in the beginning of Section 2.2 are used because they are needed for calculating the TF jitter. For panels (a) and (c), the original, time-varying vector loop implementation is used, as calculating the SS jitter does not require piecewise

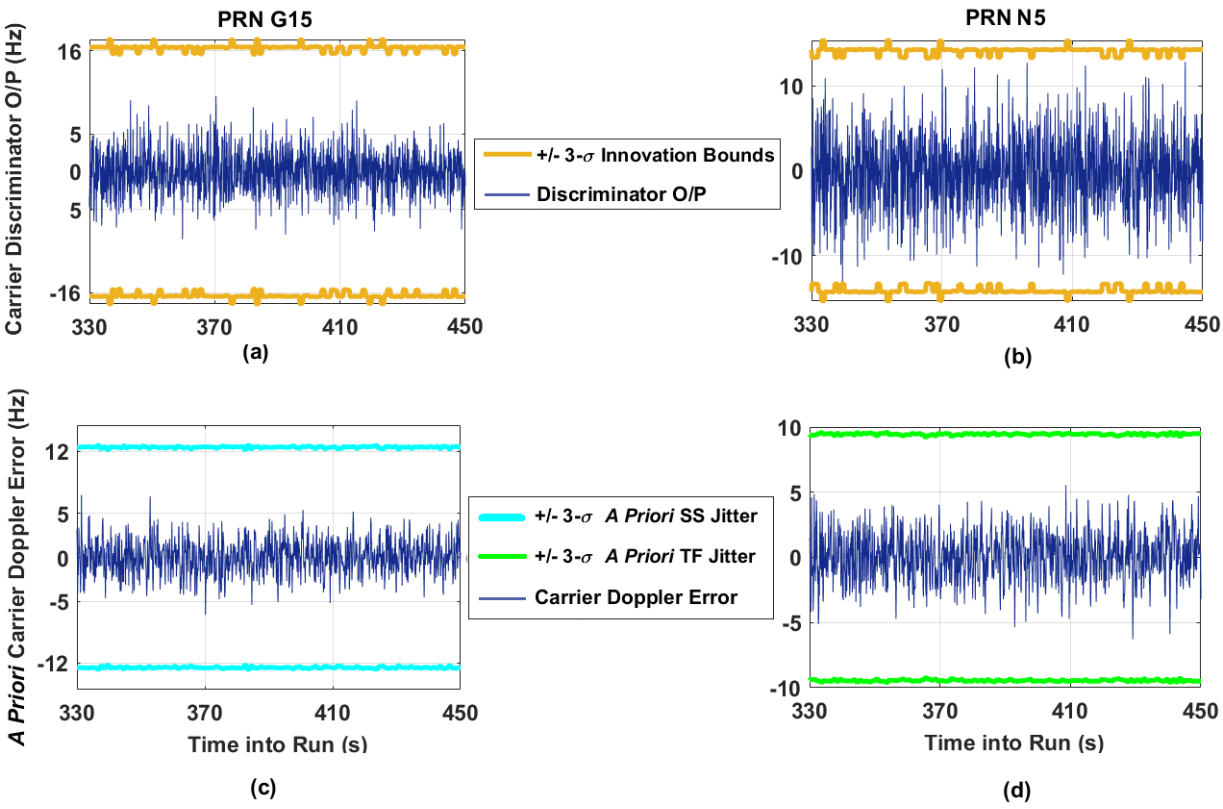

FIGURE 6 Steady-state carrier Doppler jitter performance in two channels at a mean C/N_0 of 30 dB-Hz
Carrier frequency discriminator outputs are also shown along with $\pm 3\text{-}\sigma$ innovation bounds. Panel (c) depicts the SS jitter, and panel (d) presents the TF jitter, as obtained from their respective vector loop runs.

constant models. Both figures show that the total *a priori* SS and TF jitters bound the corresponding code phase and carrier Doppler errors in their respective vector loop runs. Although not shown, other channels offer similar performance.

The metric proposed in Section 3 (i.e., the mean jitter upper bound of the VDLL/VFLL) is expected to capture the average vector loop behavior over an interval of 0.1 s. Therefore, it is imperative to determine whether this metric can predict the code and carrier channel errors. For this purpose, Figure 7(a) presents the maximum and minimum code phase errors among all channels before and after the measurement updates (i.e., the maximum and minimum considering the *a posteriori* and *a priori* errors over a 0.1-s interval) at 30 dB-Hz and the corresponding $\pm 3\text{-}\sigma$ VDLL jitter metric. Panel (b) of the same figure illustrates these errors for the carrier channels. It is evident that the jitter metric is a good indicator of the code and carrier channel performance. Finally, Figure 8 shows the carrier channel behavior for a mean C/N_0 of 23.1 dB-Hz for both $\mathbf{Q}_1$ and $\mathbf{Q}_2$ process noise covariances. While the carrier Doppler error sometimes exceeds the jitter metric in panel (a), the overall performance is acceptable. The code phase error plots, although omitted for brevity, are mostly bounded by their respective $3\text{-}\sigma$ jitter metrics. It is important to note that the $3\text{-}\sigma$ jitter metric represents the collective performance of all vector loop channels on average; hence, sporadic level crossings may occur at some time instants.

The previous analysis indicates that jitter can predict the code and carrier channel errors in the presence of noise. Next, the loss-of-lock performance of the jitter is studied in the presence of jerk. For this purpose, the proposed metric with the

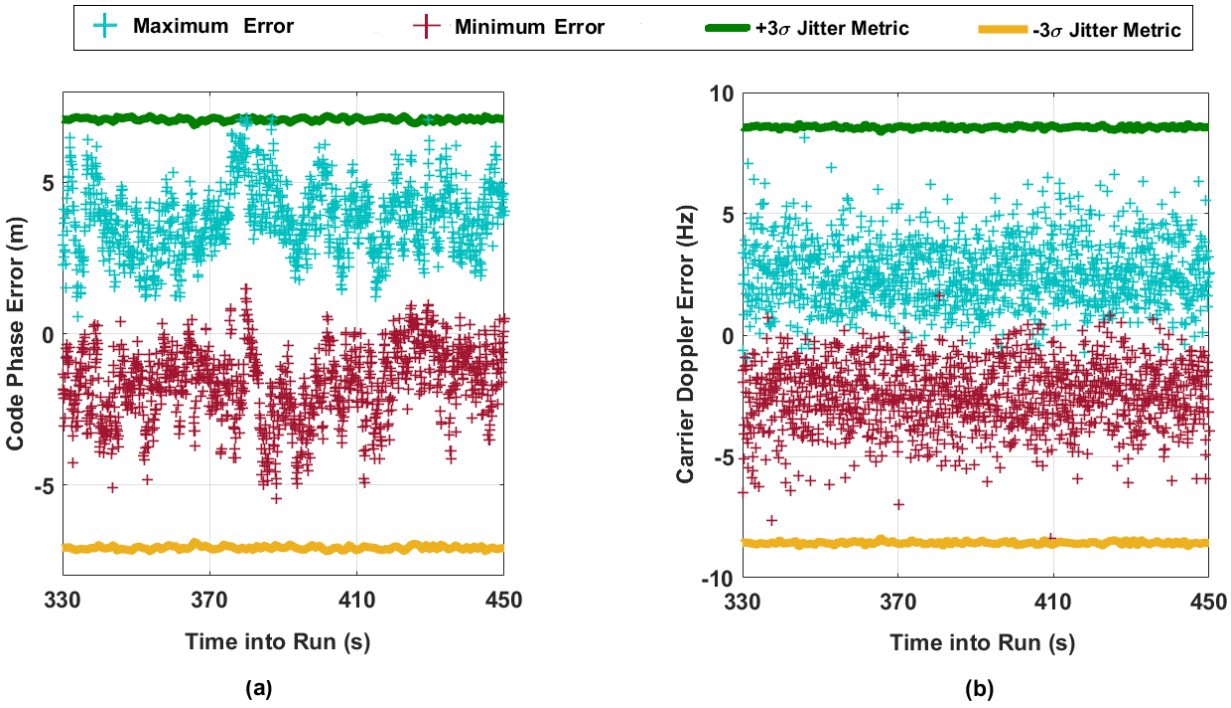

FIGURE 7 Comparison of ±3-σ jitter metric (or mean jitter upper bound) against code phase and carrier Doppler errors at 30 dB-Hz between 330 and 450 s into the run with $\mathbf{Q}_1$ process noise covariance

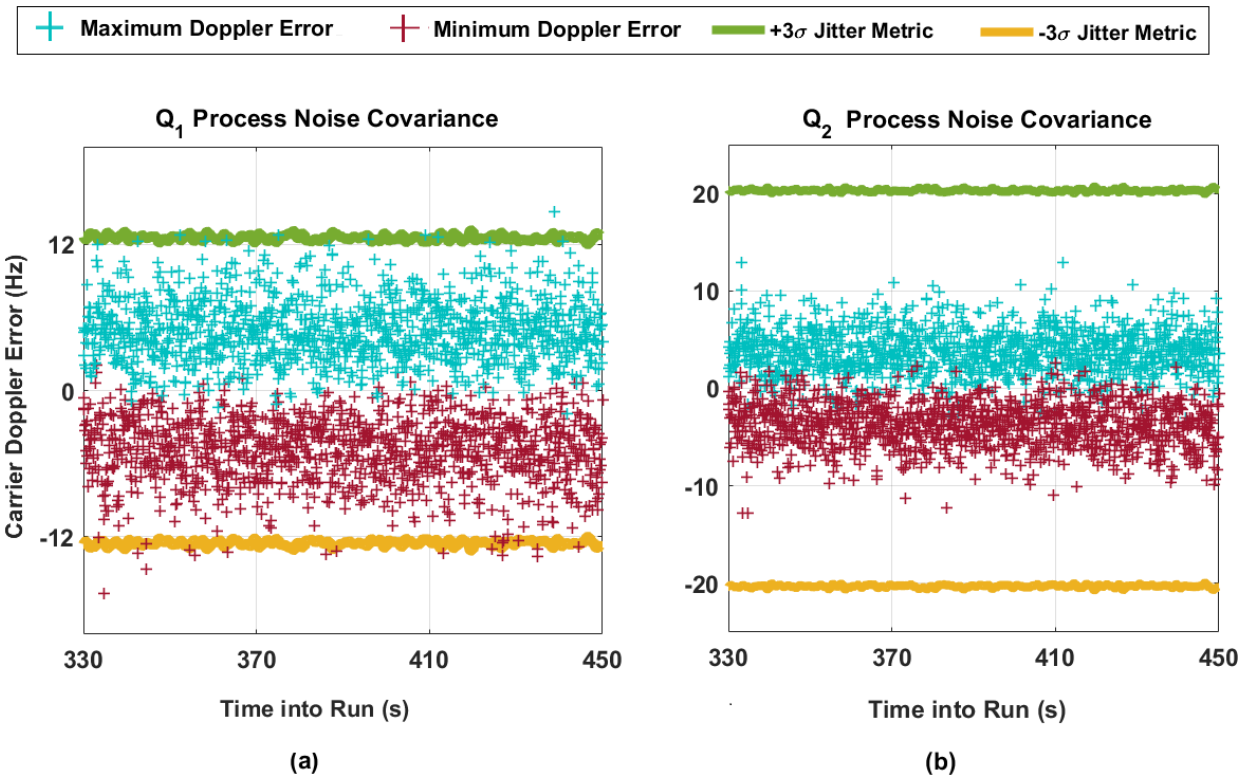

FIGURE 8 Comparison of ±3-σ VFLL jitter metric (or mean VFLL jitter upper bound) against carrier Doppler errors at 23 dB-Hz between 330 and 450 s into the run

SS jitter in Section 3 is shown in Figure 9 for the VFLL at different C/N_0 ratios. Artificial white noise is injected into the recorded IF data from 126.1 s into the run in panels (a) and (c) and from 76.1 s in panels (b) and (d) to reduce the C/N_0 from the default range of (42–44) dB-Hz. This results in an abrupt increase in the jitter before 200 s, as shown in the figure. The C/N_0 is restored after 1000 s. The mean of the reduced C/N_0 values over all channels and time steps before loss of lock is indicated at the top of each panel. It should be noted that each panel shows results for a different run executed independently of the others. The solid dark line in the figure represents the 1-σ jitter threshold, $1/(12T)$, for loss-of-lock detection, which is 8.33 Hz for $T = 10$ ms. The label "State Divergence Onset" indicates the time instant when one of the vector loop states during the simulation starts to oscillate/diverge, meaning its error just crosses the 3-σ bound of the filter. This is eventually followed by error growth in all states. The marker in the figure, on the other hand, indicates the time instant when the jitter metric crosses its threshold by a sufficient margin. Figures 9(a) and (b) show results for $\mathbf{Q}_1$ process noise covariance whereas panels (c) and (d) show those for $\mathbf{Q}_2$. In (c), the mean jitter upper

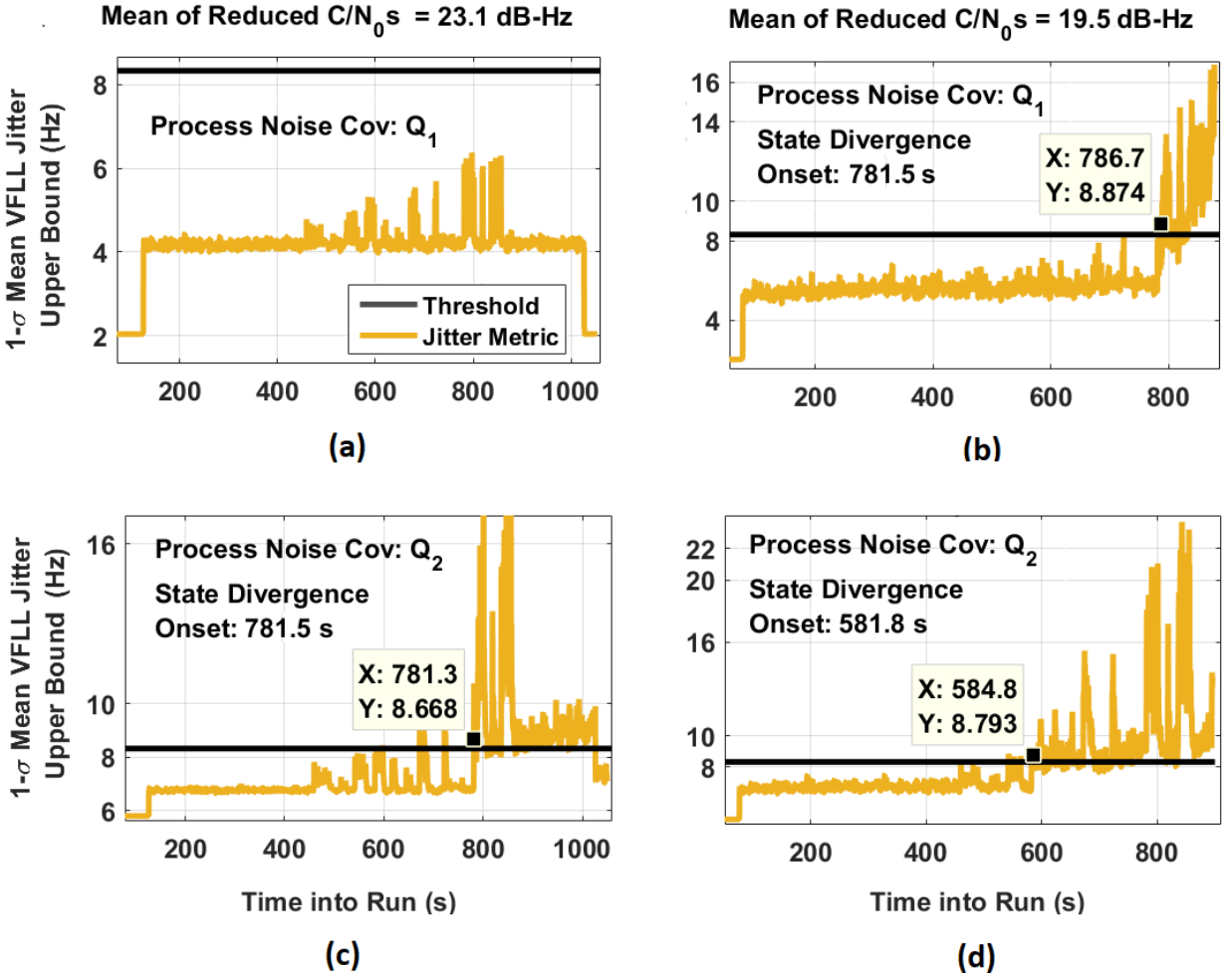

FIGURE 9 Mean VFLL jitter upper bound for different mean C/N_0 ratios
Initial epochs with transients are not considered. Although not shown in the figure, the mean jitter lower bound (i.e., the mean of bounds calculated at the beginning of 20-ms intervals in Figure 3) exceeds the threshold for the first time at 789.3 s, 781.5 s, and 594.3 s in (b), (c), and (d), respectively. The abrupt change in jitter before 200 s and after 1000 s is due to the decrease in C/N_0 from (42–44) dB-Hz and the increase afterward.

bound crosses its threshold just before the onset of the actual state error growth. In (b) and (d), the threshold is crossed at 5.2 s and 3 s after the onset, respectively. In contrast, panel (a) with $\mathbf{Q}_1$ process noise covariance does not exhibit any loss of lock, as the vector loops work for the entire duration in this case. Thus, Figure 9 illustrates that the proposed jitter metric holds promise to represent the collective vector loop performance and could be a good indicator of loss of lock. Although the jitter lower bound is not shown, the time instants when the mean bound crosses the threshold are provided in the figure caption. A comparison with Figure 4(d) indicates that loss of lock occurs during 16.71 g/s in panels (b) and (c) of the figure and during 9.954 g/s in panel (d). It is evident that vector loops offer better dynamic performance with $\mathbf{Q}_1$ than with $\mathbf{Q}_2$. However, the choice of $\mathbf{Q}_2$ allows for the demonstration of loss of lock at two different jerk levels. It is important to note that the x-axis limits are less than 1000 s in panels (b) and (d) because the jitter is plotted until the vector loop algorithm has a minimum number of satellites for positioning. Jitter calculation is also not valid after loss of lock when the position and velocity errors are sufficiently large because the linearized model of the vector loops does not hold. Therefore, plotting the jitter after a few hundred seconds of continuous loss of lock does not provide any meaningful information, as is evident in panel (c) of the figure. In this case, after indicating loss of lock for 244.7 s, the jitter metric drops below the threshold when C/N_0 is restored to (42–44) dB-Hz, although the reduced value is larger than its previous level for the same C/N_0. A careful inspection also reveals that after restoring C/N_0, the individual channel jitters exhibit large variations over time in comparison with the earlier smooth profile. The actual vector loop simulation, on the other hand, has diverged significantly in 244.7 s, and in the remaining few seconds, the EKF navigation filter does not show any tendency to recover from the divergence.

5 | CONCLUSION

In this paper, the SS jitter and TF jitter of the noncoherent vector architecture were presented for a theoretical evaluation of the tracking loop performance. In this context, z-domain TF models of the time-varying vector architecture were developed without decoupling the VDLL and VFLL. Noise equivalent BWs were also calculated. The results show that the SS and TF jitters are in close agreement, thus proving their equivalence. Appropriate jitter bounds were proposed for measurement update intervals greater than 20 ms so that existing loss-of-lock thresholds could be applied. A metric to represent the collective performance of all channels was identified for loss-of-lock detection. The steady-state jitter performance in the absence of dynamic stress error was analyzed in detail. Then, high-dynamic simulations of GPS L1 and NavIC L5 signals were used to showcase the potential of the proposed metric to detect loss of lock at different jerk levels. In future work, extensive simulations with different user trajectories will be performed to assess the jitter performance. The feasibility of applying the user jerk calculated from the estimated user acceleration will be evaluated to obtain the total jitter in real time. The off-diagonal elements of the process noise covariance matrix will be accounted for in the TF jitter calculation. A better metric to represent the collective performance may also be identified. In addition, adaptive determination of the process noise covariance matrix may be implemented (Chen et al., 2024). Jitter is theoretically determined in this paper. A suitable methodology to estimate it will be developed in future work.

CONFLICT OF INTEREST

The author declares no conflicts of interest.

REFERENCES

Abedi, A. A., & Mosavi, M. R. (2022). Low computational-complexity vector tracking for low-cost GNSS receivers. *Measurement*, *192*, 1–14. https://doi.org/10.1016/j.measurement.2022.111171

Accord Software and Systems. (2024). *Accord's GNSS simulator*. https://www.accord-soft.com/GNSS_simulator_simac6.php

Anderson, B. D. O., & Moore, J. B. (2005). Time-invariant filters. In *Optimal filtering* (pp. 62–89). Dover Publications.

Bhattacharyya, S. (2018). Vector loop transfer functions and noise bandwidths. *NAVIGATION*, *65*(1), 55–72. https://doi.org/10.1002/navi.219

Bhattacharyya, S., & Gebre-Egziabher, D. (2010). Development and validation of parametric models for vector tracking loops. *NAVIGATION*, *57*(4), 275–295. https://doi.org/10.1002/j.2161-4296.2010.tb01783.x

Brewer, J., & Raquet, J. (2016). Differential vector phase locked loop. *IEEE Transactions on Aerospace and Electronic Systems*, *52*(3), 1046–1055. https://doi.org/10.1109/TAES.2016.140160

Brown, R. G., & Hwang, P. Y. C. (2012). Kalman filter applications to the GPS and other navigation systems. In *Introduction to random signals and applied Kalman filtering* (pp. 324–328). John Wiley & Sons, Inc.

Chen, Z., Biggie, H., Ahmed, N., Julier, S., & Heckman, C. (2024). Kalman filter auto-tuning with consistent and robust Bayesian optimization. *IEEE Transactions on Aerospace and Electronic Systems*, *60*(2), 2236–2250. https://doi.org/10.1109/TAES.2024.3350587

Copps, E.M., Geier, G.J., Fidler, W.C., & Grundy, P.A., (1980). Optimal processing of GPS signals. *NAVIGATION*, *27*(3), 171–182. https://doi.org/10.1002/j.2161-4296.1980.tb01417.x

Curran, J. T. (2015). Enhancing weak-signal carrier phase tracking in GNSS receivers. *International Journal of Navigation and Observation*, *2015*(1), 1–15. https://doi.org/10.1155/2015/295029

Curran, J. T., Lachapelle, G., & Murphy, C. C. (2012a). GNSS PLL design conditioned on thermal and oscillator phase noise. *IEEE Transactions on Aerospace and Electronic Systems*, *48*(1), 180–196. https://doi.org/10.1109/TAES.2012.6129629

Curran, J. T., Lachapelle, G., & Murphy, C. C. (2012b). Improving the design of frequency lock loops for GNSS receivers. *IEEE Transactions on Aerospace and Electronic Systems*, *48*(1), 850–868. https://doi.org/10.1109/TAES.2012.6129674

Gómez, M. A., Solera-Rico, A., Valero, E., Lázaro, J. A., & Fernández-Prades, C. (2023). Enhancing GNSS receiver performance with software-defined vector carrier tracking for rocket launching. *Results in Engineering*, *19*, 1–15. https://doi.org/10.1016/j.rineng.2023.101310

Groves, P. D. (2013). GNSS: User equipment processing and errors. In *Principles of GNSS inertial and multisensor integrated navigation systems* (pp. 399–417). Artech House.

Kailath, T. (1980). State space descriptions - some basic concepts. In *Linear systems* (pp. 79–145). Prentice-Hall, Inc.

Lashley, M., Bevly, D. M., & Hung, J. Y. (2009). Performance analysis of vector tracking algorithms for weak GPS signals in high dynamics. *IEEE Journal of Selected Topics in Signal Processing*, *3*(4), 661–673. https://doi.org/10.1109/JSTSP.2009.2023341

Liu, J., Cui, X., Chen, Q., & Lu, M. (2011, January). Joint vector tracking loop in a GNSS receiver. In *Proceedings of the 2011 International Technical Meeting of the Institute of Navigation*, (pp. 1025–1032). https://www.ion.org/publications/pdf.cfm?articleID=9550

Mu, R., & Long, T. (2017). Design and implementation of vector tracking loop for high-dynamic GNSS receiver. *Sensors*, *21*(16:5629), 1–19. https://doi.org/10.3390/s21165629

Mute, D., & Bhattacharyya, S. (2023). A robust signal tracking algorithm for GPS and NAVIC constellations. *GPS Solutions*, *27*(3), 1–20. https://doi.org/10.1007/s10291-023-01444-2

Ng, Y., & Gao, G. X. (2017). GNSS multireceiver vector tracking. *IEEE Transactions on Aerospace and Electronic Systems*, *53*(5), 2583–2593. https://doi.org/10.1109/TAES.2017.2705338

Razavi, A., Gebre-Egziabher, D., & Akos, D. M. (2008). Carrier loop architectures for tracking weak GPS signals. *IEEE Transactions on Aerospace and Electronic Systems*, *44*(2), 697–710. https://doi.org/10.1109/TAES.2008.4560215

Shytermeja, E. (2017, December). *Design and performance of a GNSS single-frequency multi-constellation vector tracking architecture for urban environments* [Doctoral dissertation, Université de Toulouse].

Simon, D. (2006). Additional topics in Kalman filtering. In *Optimal state estimation: Kalman, H_∞, and nonlinear approaches* (pp. 298–329). Wiley-Interscience.

Spilker Jr., J. J. (1995). Fundamentals of signal tracking theory. In B. W. Parkinson & J. J. Spilker Jr. (Eds.), *Global Positioning System: Theory and applications* (Vol. 1, pp. 245–327). AIAA.

Tabatabaei, A., Mosavi, M. R., Shahhoseini, H. S., & Borre, K. (2017). Vectorized and federated software receivers combining GLONASS and GPS. *GPS Solutions*, *21*(3), 1331–1339. https://doi.org/10.1007/s10291-017-0615-8

Ward, P., Betz, J. W., & Hegarty, C. J. (2006). Satellite signal acquisition, tracking, and data demodulation. In E. D. Kaplan & C. J. Hegarty (Eds.), *Understanding GPS principles and applications* (pp. 153–241). Artech House.

Weill, L. R. (2010, September). A high performance code and carrier tracking architecture for ground-based mobile GNSS receivers. In *Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS)* (pp. 3054–3068). https://www.ion.org/publications/abstract.cfm?articleID=9413

Yang, R., Ling, K. V., Poh, E. K., & Morton, Y. (2017). Generalized GNSS signal carrier tracking-part II: Optimization and implementation. *IEEE Transactions on Aerospace and Electronic Systems*, *53*(4), 1798–1811. https://doi.org/10.1109/TAES.2017.2674198

How to cite this article: Bhattacharyya, S. (2026). Vector tracking loop jitter. *NAVIGATION*, *73*. https://doi.org/10.33012/navi.788

A | APPENDIX

In this section, the measurement model matrix of the vector navigation filter is defined. Then, the observability and reachability properties of some matrices are proven. Finally, the algorithms for T_{sub} and vector jitter calculations are described.

A.1 | Measurement Model Matrix of the Vector Navigation Filter

The measurement model matrix of the vector navigation filter at time t_ℓ is $\mathbf{H}_\ell = \begin{bmatrix} \mathbf{H}_{x,\ell} & \mathbf{H}_{y,\ell} & \mathbf{H}_{z,\ell} & \mathbf{H}^G_{clk,\ell} & \mathbf{H}^N_{clk,\ell} \end{bmatrix}$. Its components are as follows:

$$\mathbf{H}_{\xi,\ell} = \begin{bmatrix} \tilde{u}^\xi_{1,\ell}\frac{f_{code}}{c} & \cdots & \tilde{u}^\xi_{n_G+1,\ell}\frac{f_{code}}{c} & \cdots & \tilde{u}^\xi_{n,\ell}\frac{f_{code}}{c} & [\mathbf{0}]_{1\times n} \\ & [\mathbf{0}]_{1\times n} & & \tilde{u}^\xi_{1,\ell}\frac{f^G_{carr}}{c} & \cdots & \tilde{u}^\xi_{n,\ell}\frac{f^N_{carr}}{c} \\ & & & [\mathbf{0}]_{1\times 2n} & & \end{bmatrix}^T_{3\times 2n}, \xi = x, y, z$$

$$\mathbf{H}^G_{clk,\ell} = \begin{bmatrix} -\frac{f_{code}}{c} \cdots -\frac{f_{code}}{c} & [\mathbf{0}]_{1\times n_N} & [\mathbf{0}]_{1\times n} \\ [\mathbf{0}]_{1\times n} & -\frac{f^G_{carr}}{c} \cdots -\frac{f^G_{carr}}{c} & [\mathbf{0}]_{1\times n_N} \\ & [\mathbf{0}]_{1\times 2n} & \end{bmatrix}^T_{3\times 2n}$$

$$\mathbf{H}^N_{clk,\ell} = \begin{bmatrix} [\mathbf{0}]_{1\times n_G} & -\frac{f_{code}}{c} \cdots -\frac{f_{code}}{c} & [\mathbf{0}]_{1\times n} \\ [\mathbf{0}]_{1\times n_G} & [\mathbf{0}]_{1\times n} & -\frac{f^N_{carr}}{c} \cdots -\frac{f^N_{carr}}{c} \\ & [\mathbf{0}]_{1\times 2n} & \end{bmatrix}^T_{3\times 2n}$$

where $[\mathbf{0}]$ denotes a matrix or vector of zero elements. $\tilde{u}^\xi_{i,\ell}$ is equal to the x-, y- or z-component of the line-of-sight vector in the ECEF frame at t_ℓ for the i-th

satellite, with $\xi = x, y, z$. n_G is the number of visible GPS satellites, and n_N is the number of visible NavIC satellites at t_ℓ (i.e., $n_G + n_N = n$). Columns 1–3 of $\mathbf{H}^G_{clk,\ell}$ and $\mathbf{H}^N_{clk,\ell}$ correspond to the receiver clock bias, drift, and drift rate for GPS and NavIC, respectively. f_{code} denotes the transmitted C/A-code chipping rate, which is the same for both constellations. f^G_{carr} and f^N_{carr} are the GPS and NavIC carrier frequencies, respectively. c is the speed of light.

Next, the observability and reachability properties of some matrices are proven. It should be noted that the subscript ℓ denoting the time instants in the matrices is dropped for ease of representation.

A.2 | Proof that (H, (I − KH)F) is Observable

First, it is proved that $(\mathbf{H}, \mathbf{F}(\mathbf{I} - \mathbf{KH}))$ is observable. It has been assumed that $(\mathbf{H}, \mathbf{F})$ of the vector navigation filter is observable. Therefore, $(\mathbf{F}^T, \mathbf{H}^T)$ is reachable. Using the reachability property (i.e., the Popov–Belevitch–Hautus eigenvector tests (Kailath, 1980)), it can be stated that, for any left eigenvector $\underline{v}^T$ of $\mathbf{F}^T$ for the eigenvalue λ, $\underline{v}^T\mathbf{F}^T = \lambda\underline{v}^T$ and $\underline{v}^T\mathbf{H}^T \neq 0$. Let $\underline{u}^T$ be a left eigenvector of $(\mathbf{F}(\mathbf{I}-\mathbf{KH}))^T$ such that $\underline{u}^T\mathbf{H}^T = 0$. If the corresponding eigenvalue of $(\mathbf{F}(\mathbf{I}-\mathbf{KH}))^T$ is γ, then using $\underline{u}^T\mathbf{H}^T = 0$, one can write the following:

$$\underline{u}^T(\mathbf{F}(\mathbf{I}-\mathbf{KH}))^T = \underline{u}^T\mathbf{F}^T = \gamma\underline{u}^T \tag{34}$$

This relation implies that $\underline{u}^T$ is a left eigenvector of $(\mathbf{F})^T$ and $\underline{u}^T\mathbf{H}^T = 0$, which contradicts the reachability criterion of $(\mathbf{F}^T, \mathbf{H}^T)$ stated before. Thus, for a left eigenvector $\underline{u}^T$ of $(\mathbf{F}(\mathbf{I}-\mathbf{KH}))^T, \underline{u}^T\mathbf{H}^T \neq 0$. That is, $((\mathbf{F}(\mathbf{I}-\mathbf{KH}))^T, \mathbf{H}^T)$ is reachable or $(\mathbf{H}, \mathbf{F}(\mathbf{I}-\mathbf{KH}))$ is observable.

Next, it is shown that $(\mathbf{H}, (\mathbf{I}-\mathbf{KH})\mathbf{F})$ is observable (or $(((\mathbf{I}-\mathbf{KH})\mathbf{F})^T, \mathbf{H}^T)$ is reachable). Let $\underline{\omega}^T$ be a left eigenvector of $((\mathbf{I}-\mathbf{KH})\mathbf{F})^T$ for the eigenvalue α. Then, the following relation holds:

$$\underline{\omega}^T((\mathbf{I}-\mathbf{KH})\mathbf{F})^T = \alpha\underline{\omega}^T \tag{35}$$

By post-multiplying $\mathbf{F}^T$ on either side of the preceding equation and writing $\underline{\omega}^T\mathbf{F}^T$ as $\underline{\omega}_1^T$, one obtains $\underline{\omega}_1^T(\mathbf{I}-\mathbf{KH})^T\mathbf{F}^T = \alpha\underline{\omega}_1^T$. Thus, $\underline{\omega}_1^T$ is a left eigenvector of $(\mathbf{F}(\mathbf{I}-\mathbf{KH}))^T$. Because $((\mathbf{F}(\mathbf{I}-\mathbf{KH}))^T, \mathbf{H}^T)$ has been proved to be reachable, $\underline{\omega}_1^T\mathbf{H}^T \neq 0$. Hence, $\mathbf{H}\underline{\omega}_1 \neq 0$. Next, it is shown that $\mathbf{H}\underline{\omega} \neq 0$. Because $\underline{\omega}^T$ is a left eigenvector of $((\mathbf{I}-\mathbf{KH})\mathbf{F})^T$, proving $\mathbf{H}\underline{\omega} \neq 0$ (or, $\underline{\omega}^T\mathbf{H}^T \neq 0$) would imply that $(((\mathbf{I}-\mathbf{KH})\mathbf{F})^T, \mathbf{H}^T)$ is reachable or $(\mathbf{H}, (\mathbf{I}-\mathbf{KH})\mathbf{F})$ is observable.

From the expression of $\mathbf{F}$ after Equation (3), it is evident that $\mathbf{F}$ is invertible. Therefore, $\mathbf{H}\underline{\omega} = \mathbf{H}(\mathbf{F})^{-1}\underline{\omega}_1$. As $\mathbf{H}\underline{\omega}_1 \neq 0, \underline{\omega}_1 \in \mathcal{R}_{\mathbf{H}^T}$ (the range space of $\mathbf{H}^T$), which is orthogonal to the null space of $\mathbf{H}$, $\mathcal{N}_{\mathbf{H}}$. Using the mathematical expression of $\mathbf{F}$, it can be shown that $\underline{\omega}_1^T(\mathbf{F})^{-1}\underline{\omega}_1$ cannot be zero for a nonzero $\underline{\omega}_1$, meaning $\mathbf{F}^{-1}$ will not project $\underline{\omega}_1$ from $\mathcal{R}_{\mathbf{H}^T}$ to $\mathcal{N}_{\mathbf{H}}$. Therefore, $\mathbf{H}\underline{\omega} \neq 0$, and $(\mathbf{H}, (\mathbf{I}-\mathbf{KH})\mathbf{F})$ is observable.

A.3 | Proof that ((I − KH)F, (I − KH)) is Stabilizable

Because $(\mathbf{H}, \mathbf{F})$ of the vector navigation filter is observable, there exists a $\mathbf{K}'$ such that the eigenvalues of $(\mathbf{F}-\mathbf{K}'\mathbf{H})$ are less than one (Anderson & Moore, 2005).

Alternatively, it can be stated that by suitably selecting $\mathbf{K}'$, one can arbitrarily place the eigenvalues of $(\mathbf{F}-\mathbf{K}'\mathbf{H})$. Because $\mathbf{F}$ is invertible by definition, for each $\mathbf{K}'$, there exists a unique $\mathbf{K}$ such that $\mathbf{K}=(\mathbf{F})^{-1}\mathbf{K}'$. Therefore, for any $\mathbf{K}'$, $(\mathbf{F}-\mathbf{K}'\mathbf{H})$ is identical to $\mathbf{F}(\mathbf{I}-\mathbf{KH})$, where $\mathbf{K}=(\mathbf{F})^{-1}\mathbf{K}'$. It can be easily shown that the eigenvalues of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ are identical to those of $\mathbf{F}(\mathbf{I}-\mathbf{KH})$. Thus, the eigenvalues of $\mathbf{F}(\mathbf{I}-\mathbf{KH})$ or $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ can be arbitrarily placed by selecting a suitable $\mathbf{K}$.

Let $\underline{v}^T$ be a left eigenvector of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ such that $\underline{v}^T(\mathbf{I}-\mathbf{KH})=0$. Hence, $\underline{v}^T(\mathbf{I}-\mathbf{KH})\mathbf{F}=0$, which implies that the corresponding eigenvalue of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ is zero. Therefore, $\underline{v}^T(\mathbf{I}-\mathbf{KH})\neq 0$ for any left eigenvector, $\underline{v}^T$, of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ for a nonzero eigenvalue. Thus, if $\mathbf{K}$ is selected such that the eigenvalues of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ are nonzero, $((\mathbf{I}-\mathbf{KH})\mathbf{F},(\mathbf{I}-\mathbf{KH}))$ is reachable. If $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ has a zero eigenvalue, then $((\mathbf{I}-\mathbf{KH})\mathbf{F},(\mathbf{I}-\mathbf{KH}))$ is stabilizable, implying that any uncontrollable modes of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ are stable. This is because the eigenvalues of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ can be arbitrarily placed or made less than one by selecting a suitable $\mathbf{K}$ (see the preceding paragraph). This is true for a stable, working vector navigation filter. Although stabilizability is not required for the calculation of vector loop jitter, the proof in this section indicates that $\underline{e}^+$ is affected by the process noise input in Equation (4), as the eigenvalues of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ cannot all be zero; hence, some modes of $\underline{e}^+$ would be controllable or affected by the process noise.

A.4 | Proof that ((I − KH)F, K) is Reachable

Let $\underline{v}^T$ be a left eigenvector of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ for the eigenvalue γ such that $\underline{v}^T\mathbf{K}=0$. Hence, $\underline{v}^T\mathbf{F}=\gamma\underline{v}^T$, which implies that γ is also an eigenvalue of $\mathbf{F}$. Following the first paragraph of Appendix A.3, the eigenvalues of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ can be arbitrarily placed by selecting a suitable $\mathbf{K}$. For an asymptotically stable navigation filter (i.e., when vector loops are operational with a good number of satellites being tracked), $\mathbf{K}$ should be such that the eigenvalues of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$ are less than one. However, the eigenvalues of $\mathbf{F}$ are fixed and equal to one. Therefore, $\underline{v}^T\mathbf{K}\neq 0$ for any left eigenvector, $\underline{v}^T$, of $(\mathbf{I}-\mathbf{KH})\mathbf{F}$. Consequently, $((\mathbf{I}-\mathbf{KH})\mathbf{F},\mathbf{K})$ is reachable.

A.5 | Algorithm for T_{sub} Calculation

The algorithm for calculating T_{sub}, over which $\mathbf{H}$, $\mathbf{W}$, and $\mathbf{K}$ matrices are assumed to be constant, is given in this section. A new model is started (i.e., the duration of T_{sub} for the existing TF model is determined) if one of the following happens: 1) the satellite visibility changes; 2) the 2-norm of the difference between the actual time-varying $\mathbf{H}$ matrix and that of the existing model exceeds a certain threshold; 3) the 2-norm of the difference between the optimal $\mathbf{K}$ (i.e., the $\mathbf{K}$ obtained using the EKF equations with the actual time-varying model of the filter) and that of the existing piecewise constant model exceeds a certain threshold. To determine whether the thrid condition arises, the optimal $\mathbf{K}$ is computed for each measurement update interval. The algorithm for the T_{sub} calculation is described next.

ALGORITHM 3
T_{sub} calculation

```
if(first measurement update)
   tf_counter = 1; % TF model counter is one at the beginning
end
vis_flag = 0; % Set visibility flag to zero at each epoch. If satellite visibility
% changes (i.e., a new satellite comes or a satellite sets/is blocked), a new model
% should be started. To indicate this, vis_flag is later set to one, when needed.
% Set the maximum 2-norm of the difference between the actual H and that of the
% existing TF model
delta_H_th = 0.01;
% Set the maximum 2-norm of the difference between the optimal K matrix and that of
% the existing TF model (delta_K_th) based on C/N0. The values of delta_K_th are selected
% by trial and error. It should be noted that lower values are also allowed, as the minimum
% T_sub is fixed to 0.1 s. A larger T_sub results in a TF model being valid for a longer duration,
% but performance should not degrade due to a larger T_sub.
if(average C/N0 over all channels > 30)
   delta_K_th = 0.5;
else if(average C/N0 > 20)
   delta_K_th = 0.1;
else
   delta_K_th = 0.001;
end
if(a new satellite has come or one or more satellites have set/are blocked)
   vis_flag = 1; % As a change in visibility may create transients, a new model should
   % be started
end
if(vis_flag == 1 || number of measurement updates == 20) % The first 19 0.1-s intervals are
% skipped to avoid large transients
   T_sub(tf_counter) = 0.1; % Start a new model
   % Make K, H, and W of this model equal to the optimal K and actual H and W at the
   % current time, respectively
   tf_counter = tf_counter + 1; % Increment the model counter to point to the next model
else if(number of measurement updates > 20)
   delta_K = Optimal K - constant K of existing model; % Find delta_K
   delta_H = actual H - constant H of existing model; % Find delta_H
   if(norm(delta_K) < delta_K_th and norm(delta_H) < delta_H_th)
      T_sub(tf_counter-1) = T_sub(tf_counter-1) + 0.1; % Increment the duration of the
      % existing model, and retain the previous K, H, and W
   else
      T_sub(tf_counter) = 0.1; % Start a new model
      % Make K, H, and W of this model equal to the optimal K and actual H and W at
      % the current time, respectively
      tf_counter = tf_counter + 1; % Increment the model counter to point to the next model
   end
end
```

A.6 | Algorithm for Overall Sequence of Operations for Jitter Calculation

ALGORITHM 4
Overall sequence of operations for the calculation of SS and TF jitters and jitter metric

```
% Obtain all H, W, and K matrices and T_sub values from the vector loop run
tf_index = 1; % TF model index is initialized to one
loop_count = 0.1; % Iteration counter has a step size of 0.1
for t = 0:0.1:total duration % A time loop iteration is performed every 0.1 s
  % Calculation of the total a priori and a posteriori SS jitters and jitter metric for
  % loss-of-lock detection
  % Execute Algorithm 2 to calculate the five jitter lower and upper bounds over a 0.1-s
  % interval between any two time steps ℓ and (ℓ+1). Note that the first of the five lower
  % bounds is the total a posteriori SS jitter at ℓ, and the last of the five upper bounds is the
  % total a priori SS jitter at (ℓ+1) for an increasing trend in jitter (see Figure 3).
  % Calculate the jitter metric for loss-of-lock detection (i.e., the mean of bounds, as
  % discussed in Section 3)

  % Calculation of the total a priori and a posteriori TF jitters
  if(loop_count == 0.1) % Execute only once for a given TF model
    % Consider appropriate H, W, and K matrices for the TF model, which would last for
    % T_sub(tf_index)
    % Using Algorithm 1, calculate the process and measurement noise BWs for the TFs in
    % Equations (18) and (19)
  end
  if(loop_count ~= T_sub(tf_index))
    loop_count = loop_count + 0.1; % Increment loop_count for the duration of
    % T_sub(tf_index) (≥ 0.1)
  else
    loop_count = 0.1; % Reset loop_count to 0.1 to re-calculate the BWs as a new TF model
    % would start in the next iteration
    tf_index = tf_index + 1; % Increment tf_index by one to point to the next TF model
  end
  % For the dynamic stress error at ℓ, determine the impulse response of
  % (zI−(I−K_q H_q)F)^{-1} after 0.1 s due to Δx⁺_{dyn,tf,ℓ−1} in Equation (27) (and similarly due
  % to Δx⁻_{dyn,tf,ℓ−1} in Equation (28)). Use the MATLAB function "impulse" for
  % this purpose.
  % Determine the step response of (zI−(I−K_q H_q)F)^{-1}(I−K_q H_q) after 0.1 s due to
  % B u_{jerk,ℓ−1} in Equation (27) (and similarly in Equation (28)), and add this step response
  % to the corresponding impulse response to determine the dynamic stress error
  % Use Equations (22), (23), (29), and (30) to calculate the total a posteriori TF jitter.
  % Similarly, calculate the a priori TF jitter.
end
```