

## Article

# Theoretical Upper and Lower Limits for Normalized Bandwidth of Digital Phase-Locked Loop in GNSS Receivers

Young-Jin Song <sup>1</sup>, Thomas Pany <sup>2</sup> and Jong-Hoon Won <sup>3,\*</sup>

<sup>1</sup> Autonomous Navigation Laboratory, Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea; yj.song@inha.edu

<sup>2</sup> Institute of Space Technology and Space Applications, University of Federal Armed Forces Munich, 85577 Neubiberg, Germany; thomas.pany@unibw.de

<sup>3</sup> Department of Electrical Engineering, Inha University, Incheon 22212, Republic of Korea

\* Correspondence: jh.won@inha.ac.kr; Tel.: +82-32-860-7406

**Abstract:** Determining the loop noise bandwidth and the coherent integration time is essential and important for the design of a reliable digital phase-locked loop (DPLL) in global navigation satellite system (GNSS) receivers. In general, designers set such parameters approximately by utilizing the well-known fact that the DPLL is stable if the normalized bandwidth, which is the product of the integration time and the noise bandwidth, is much less than one. However, actual limit points are not fixed at exactly one, and they vary with the loop filter order and implementation method. Furthermore, a lower limit on the normalized bandwidth may exist. This paper presents theoretical upper and lower limits for the normalized bandwidth of DPLL in GNSS receivers. The upper limit was obtained by examining the stability of DPLL with a special emphasis on the digital integration methods. The stability was investigated in terms of z-plane root loci with and without the consideration of the computational delay, which is a delay induced by the calculation of the discriminator and the loop filter. The lower limit was analyzed using the DPLL measurement error composed of the thermal noise, oscillator phase noise, and dynamic stress error. By utilizing the carrier-to-noise density ratio threshold which indicates the crossing point between the measurement error and the corresponding threshold, the lower limit of the normalized bandwidth is obtained.

**Keywords:** coherent integration time; global navigation satellite system (GNSS) receiver; loop noise bandwidth; normalized bandwidth; phase-locked loop (PLL); signal tracking loop; stability analysis



**Citation:** Song, Y.-J.; Pany, T.; Won, J.-H. Theoretical Upper and Lower Limits for Normalized Bandwidth of Digital Phase-Locked Loop in GNSS Receivers. *Sensors* **2023**, *23*, 5887. <https://doi.org/10.3390/s23135887>

Academic Editor: Chris Rizos

Received: 15 May 2023

Revised: 20 June 2023

Accepted: 23 June 2023

Published: 25 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Several traditional signal tracking loop architectures, such as phase-locked loops (PLLs) for carrier phase tracking, are widely used as engineering standards in modern digital global navigation satellite system (GNSS) receivers. These architectures ensure optimal carrier phase tracking loop performance by minimizing phase noise jitter at a steady state while constraining the transient error for specified changes in the signal phase at a fixed amount [1,2].

One solution is offered by the variational method for optimization using a Lagrangian multiplier. An optimum solution for PLLs in the continuous-time domain is given by a function of the loop noise bandwidth, which can be interpreted as normal filter coefficients such as decay ratio, damping ratio, and natural frequency [1]. The same method (minimization method) can be applied in the discrete-time domain by directly employing the z-transform [3]. Finally, the filter order  $n$ , the noise bandwidth  $B$ , and the coherent integration time  $T$  (i.e., the loop update interval), which determine the number of integrators and the multiplier coefficients of the digital PLL (DPLL) and its response for different signal dynamics and noise statistics, should be chosen to achieve the design for specific user requirements [2]. The optimal solution for the design parameters varies with the expected environmental conditions of the receiver. For example, dynamic users such as receivers

within an aircraft and a missile are benefit from a large  $B$  and small  $T$  because of the large gain (to alleviate the dynamic stress) and faster update rate. On the contrary, in the case of the static but weak signal conditions such as indoor users can employ the small  $B$  and large  $T$  to raise the post-correlation signal-to-noise ratio while suppressing the noise effect.

A software implementation of this based on a simple numerical integrator for the PLL of a pseudo-noise spread spectrum receiver to accommodate a given set of dynamics specifications was previously presented [4]. A systematic survey of the theoretical and experimental work on this topic for the period between 1960 and 1980 can be found in [5].

Typically, longer  $T$  in correlators, which are equivalent to longer loop update intervals in loop filters, provide the additional processing gain required for the detection and processing of signals in harsh environments, specifically weak indoor signals and integrated navigation systems [6]. The benefits of the longer  $T$  using the aiding technique are multipath mitigation in the Doppler domain, cross-correlation protection by data wipe-off, and reduced squaring loss. The use of ancillary sensors with relevant processing facilities was proposed to extend the  $T$  to several seconds by providing better-matched replica signals [7]. However, there is no further processing gain beyond a certain  $T$  because of the spatial and temporal decorrelation of the received signal, the residual instability of the receiver oscillator, and data bit transitions, all of which result in a narrowing of the bandwidth [8]. The limits on the  $T$  due to the spatial decorrelation of multipath signals were derived and experimentally verified using a generalized multipath scattering model [6]. The results showed that the processing gain obtained from longer  $T$  saturates after the antenna has moved a certain distance, which is typical for indoor propagation.

In addition, one of the dominant error sources in PLLs is dynamic stress error, which is a barrier to further reducing the loop bandwidth  $B$ . In particular, removing the dynamic stress for a moving platform receiver using external aiding information would be a logical first step towards decreasing the  $B$  and enhancing the tracking sensitivity in weak signal environments. Therefore, narrower bandwidths are known to be a good solution in weak signal processing because they suppress noise effects. However, both longer  $T$  and narrower  $B$  are less robust to signal dynamic stress.

Investigation of the trade-off between the sensitivity and the loop update interval to choose an optimal  $T$  was performed for a signal tracking Kalman filter with a special emphasis on the discrete nature of the loop, where the carrier phase measurements were asynchronous with the loop update [9].

Furthermore, designing a DPLL relies heavily on transforming a continuous-time system into a discrete-time system, performed by Laplace to a  $z$ -domain mapping. However, if the loop update interval is large relative to the most significant time constant of the system, the discrete-time system may not be an accurate representation of the corresponding continuous-time system. That is, the larger loop update interval makes the system less stable, and the updated output no longer gives an accurate measure of the output [10].

This means that there is another type of factor, similar to transforming continuous-time systems into discrete data systems. The DPLL will be equivalent to its analog counterpart only if the product  $BT$  (the so-called normalized loop bandwidth) is close to zero [5,11]. For sufficiently small values of  $BT$  (e.g.,  $BT < 0.1$ ), this can provide an adequate starting point for the analysis and design of DPLLs. However, as  $BT$  increases, the deviation of the equivalent loop noise bandwidth and the closed-loop pole locations of the designed DPLL from the desired ones in the continuous-time domain eventually leads to the instability of the loop. Therefore, both the limit on  $T$  and  $B$  should be considered together [11,12].

Another method (controlled-root method) was proposed to derive the loop filter constants from the loop roots based on the direct physical meaning of the loop noise bandwidth and the root-specific decay rate or damping. In this method, the deviation of discrete-time systems from continuous-time systems for large values of  $BT$  is avoided. However, the maximum achievable  $BT$  for a stable loop is limited to 0.4 for a rate-only feedback-type numerically controlled oscillator (NCO) [11,12]. Note that this condition is satisfied for most communication systems because their  $BT$  value remains within this range.

However, for some new applications in GNSSs, such as weak signal tracking,  $T$  values larger than one navigation data bit duration (e.g., 20 ms) are preferred for high sensitivity after the navigation data bit transition is solved correctly. Moreover, the use of pilot signal components in new GNSSs provides a chance to use the longer  $T$ , which will inevitably require larger  $BT$  values.

In this study, although two of the DPLL design methods (minimization, controlled-root) are introduced above, another DPLL design approach, designing the filter in the analog domain and then transforming it to the digital domain by using three types of digital transform methods (step-invariant, impulse-invariant, and bilinear), is used. As it is simpler than designing directly in the digital domain or applying controlled-root formulation, it is still widely employed as an engineering standard in many GNSS receiver designs, especially for high sensitivity. In addition, the computational delay (also called as transport lag), a time delay due to the calculation process for the loop update, is assumed not to exist for the first time of the analysis and is considered later.

The paper concentrates exclusively on the theoretical upper and lower limits of  $BT$ . The approach taken here for the upper limit is to choose a value of  $BT$  that satisfies the stability condition based on root loci in the  $z$ -domain. Each digital transform method is analyzed for the utilization of the loop filters and the NCOs. In addition, the lower limits of  $BT$  are obtained considering the effect of the  $BT$  on the oscillator quality and dynamic stress of the DPLL. The contribution of this paper is to provide the baseline for the DPLL designers by suggesting the upper and lower limits as requirements.

The remainder of this paper is organized as follows. In Section 2, PLLs are briefly reviewed with a special emphasis on each digital transformation method for the loop filter and the NCO, considering the effect of the computational delay on the transfer function. Section 3 describes the stability problem encountered near the upper limit of  $BT$  values for different digital transform methods. The stability of the different digital transformation methods is tested in terms of root loci,  $BT$  margins, Bode plots, and step responses, with and without the assumption of the computational delay. Section 4 derives the lower limits of  $BT$  for a few typical scenarios in consideration of the oscillator quality and the dynamic stress of the receiver. In Section 5, the stability issue is numerically tested using a GNSS software receiver and the paper is concluded in Section 6.

## 2. Review of Phase-Locked Loops

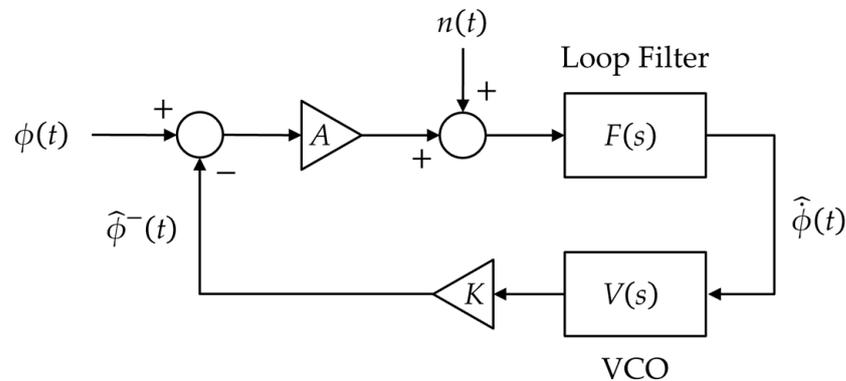
This section presents a review of the results widely used in PLL design. Specifically, the mathematical models for PLLs are reviewed in the continuous-time and discrete-time domains, with a special emphasis on each digital integration method implementing the loop filter and the NCO.

### 2.1. Continuous Model

Using the assumptions that the phase error is small enough and that the input noise is uncorrelated with the incoming waves, the linearized model for a PLL in the continuous-time domain is presented as shown in Figure 1. The device is composed of a phase detector (discriminator), a loop filter  $F(s)$ , and a voltage-controlled oscillator (VCO)  $V(s)$  with a discriminator gain  $A$  and VCO gain  $K$ , where  $s = \sigma + j\omega$  denotes the Laplace operator. In the figure,  $\phi(t)$  symbolizes the measured input phase at time  $t$ ,  $\hat{\phi}(t)$  and  $\hat{\phi}^-(t)$  represent the estimates of the phase rate and phase, respectively, and  $n(t)$  denotes the noise.

The purpose of the loop filter is to reduce the influence of the noise to produce an accurate estimate of the original signal. First-/second-/third-order loop filters (selected depending on user requirements) are widely used in GNSS receivers. The loop filter should be designed to satisfy the filter design criteria, namely minimizing VCO phase noise jitter due to noise effects (i.e., minimization of the root-mean-square noise error), and maintaining the transient error in the VCO phase caused by specified changes in the signal phase (i.e., the minimization of the transient error). These criteria are effectively met

using the variational method for optimization via the Lagrangian multiplier method for continuous-time PLLs [1] and discrete-time PLLs [3].



**Figure 1.** Linearized phase-locked loop (PLL) model in the continuous-time domain.

The VCO can be uniquely specified using a simple analog integrator represented by  $1/s$  (the Laplace transform of an integrator in the continuous-time domain):

$$V(s) = 1/s \quad (1)$$

The closed-loop transfer function of the continuous-time PLL  $H(s)$  is given by:

$$H(s) = \frac{AKV(s)F(s)}{1 + AKV(s)F(s)} \quad (2)$$

Typically, the feedback loop gain  $AK$  is assumed to be 1 (i.e., unity gain). Then, the transfer function in (2) can be rewritten as:

$$H(s) = \frac{V(s)F(s)}{1 + V(s)F(s)} = \frac{F(s)}{s + F(s)} \quad (3)$$

Table 1 presents a summary of first-/second-/third-order tracking loops and their characteristics. The filter order and the loop filter natural radian frequency  $\omega_0$  (which is obtained from  $B$ ) are determined in the design stage depending on the user requirements.  $a_i$  and  $b_i$  represent the filter coefficients of the  $i$ -th-order loop filters [2].

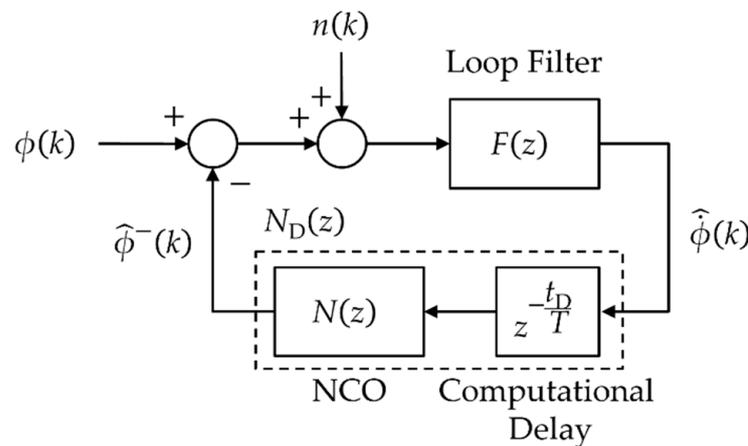
**Table 1.** Summary of loop filters.

Loop Order	Loop Filter Transfer Function, $F_i(s)$	Closed-Loop Transfer Function, $H_i(s)$	Typical Filter Values	Sensitive to	Application Examples
First	$F_1(s) = \omega_0$	$H_1(s) = \frac{\omega_0}{s + \omega_0}$	$\omega_0 = 4B$	Velocity stress	Aided code tracking loops
Second	$F_2(s) = \frac{a_2\omega_0 s + \omega_0^2}{s}$	$H_2(s) = \frac{a_2\omega_0 s + \omega_0^2}{s^2 + a_2\omega_0 s + \omega_0^2}$	$\omega_0 = 1.89B$ $a_2 = \sqrt{2}$	Acceleration stress	Aided or low-dynamics phase-locked loop, unaided frequency-locked loop
Third	$F_3(s) = \frac{b_3\omega_0 s^2 + a_3\omega_0^2 s + \omega_0^3}{s^2}$	$H_3(s) = \frac{b_3\omega_0 s^2 + a_3\omega_0^2 s + \omega_0^3}{s^3 + b_3\omega_0 s^2 + a_3\omega_0^2 s + \omega_0^3}$	$\omega_0 = 1.27B$ $a_3 \approx 1.1$ $b_3 \approx 2.4$	Jerk stress	Unaided phase-locked loop

## 2.2. Discrete Model

A block diagram for a DPLL with unity gain is shown in Figure 2. In the figure,  $F(z)$  represents the transfer function of an arbitrary loop filter, and  $N(z)$  is the transfer function of the NCO, where both are digital equivalents of  $F(s)$  and  $V(s)$ , respectively, and can be affected by the characteristics of the particular digital implementation of the analog

integrator. Additionally, the computational delay is inserted between the loop filter and the NCO to model the effect of the inevitable delay due to the calculation processes of the discriminator and loop filter. The ideal case that does not have such a delay can be represented by substituting  $t_D = 0$ . Finally, the computational delay is combined with  $N(z)$  to form the delay-included NCO transfer function  $N_D(z)$  later.



**Figure 2.** Linearized digital PLL (DPLL) model in the discrete-time domain (assuming unity gain;  $AK = 1$ ).

Without considering the computational delay, the closed-loop transfer function of the DPLL  $H(z)$  is given by:

$$H(z) = \frac{N(z)F(z)}{1 + N(z)F(z)} \quad (4)$$

which can be rewritten in polynomial form as:

$$H(z) = \frac{\beta_m z^m + \beta_{m-1} z^{m-1} + \dots + \beta_0}{\alpha_n z^n + \alpha_{n-1} z^{n-1} + \dots + \alpha_0} = \frac{\sum_{j=0}^m \beta_j z^j}{\sum_{i=0}^n \alpha_i z^i} \quad (5)$$

where  $\alpha_i$  and  $\beta_j$  are filter coefficients to the  $n$ -th order polynomial in the denominator and the  $m$ -th polynomial in the numerator, respectively, and  $m$  and  $n$  are the number of zeros and poles, respectively ( $n \geq m$ ;  $n =$  the order of the loop).

There are three primary categories of methods for obtaining a discrete equivalent of a continuous transfer function because of the discrete nature of the system, as described in the following sections [13].

### 2.2.1. Hold Equivalent

The hold equivalence method is to design a discrete system with an input consisting of samples of input signals, which has an output that approximates the output of the continuous-time transfer function whose input is the input signals in the continuous-time domain. This design is accomplished by acquiring a sequence of samples and extrapolating or holding them to produce a continuous signal. Depending on the order of the holding approximation, there are several variations in this method, such as zero-order holding and first-order holding. However, this method is only used to model a continuous system, because it relies on the time response of the system being unable to faithfully reproduce the frequency response, whereas filter design of the loop filter used in this work is based on their frequency response characteristics.

### 2.2.2. Pole-Zero Mapping

The idea behind the pole-zero mapping method is to use a set of heuristic rules to locate the poles and zeros using the map  $z = e^{sT}$ . Then, the gain of a  $z$ -transform is set to describe a discrete equivalent transfer function that approximates the given transfer function in the  $s$ -plane.

### 2.2.3. Numerical Integration

To obtain an equivalent discrete model of a PLL that retains the frequency response of the original continuous-time PLL, designers typically apply a numerical integration method (also known as the  $z$ -transform method). The fundamental concept of this method is to represent the given continuous-time transfer function as a differential equation and to derive a difference equation by replacing  $s$  in the continuous transfer function with a function of  $z$ .

Table 2 summarizes the three types of numerical integration methods that are of interest in this paper, including their  $z$ -domain and discrete-time domain expressions, in addition to some differently used terminology in different fields. These methods are step-invariant (SI), impulse-invariant (II), and bilinear (BL) rules. A detailed digital implementation of these methods for first-/second-/third-order PLLs is presented in the Appendix A.

**Table 2.** Digital equivalences to the analog integrator.

<b><math>s</math>-Domain: Continuous-Time Domain Expression</b>	<b><math>z</math>-Domain</b>	<b>Discrete-Time Domain Expression</b>	<b>Rule</b>	<b>Terminology</b>
$\frac{1}{s} \leftrightarrow y(t) = \int x(t)dt$	$\frac{1}{s} = \frac{T}{z-1}$	$y_{\text{SI}}(k) = y(k-1) + T \cdot x(k-1)$	Step-invariant	Forward rule, zero-order holder
	$\frac{1}{s} = \frac{Tz}{z-1}$	$y_{\text{II}}(k) = y(k-1) + T \cdot x(k)$	Impulse-invariant	Backward rule, box car
	$\frac{1}{s} = \frac{T}{2} \frac{z+1}{z-1}$	$y_{\text{BL}}(k) = y(k-1) + \frac{T}{2}[x(k) + x(k-1)]$	Bilinear	Trapezoid, Tustin

Given a continuous-time transfer function  $H(s)$ , a discrete equivalent  $H(z)$  can easily be found in these methods by substituting  $s$  with its counterpart equation for  $z$ , which can itself be derived from the equations in Table 2. Therefore, each of the approximations in Table 2 can be viewed as a mapping function from the  $s$ -plane to the  $z$ -plane. To make the system stable in graphical interpretation, the region of the stable poles in the  $s$ -plane (i.e., the left half of the  $s$ -plane) should be mapped onto the inside of the unit circle in the  $z$ -plane [13].

Table 3 lists the resulting  $\alpha_i$  and  $\beta_j$  in (5) for the first-/second-/third-order DPLLs for various combinations of digital representations of the integrators in the loop filter  $F(z)$  and NCO  $N(z)$ . Note that all the systems are a function of  $\omega_0 T$ , such that  $BT$  determines the overall filter response.

### 2.3. Computational Delay

The previous section derived the coefficients of the closed-loop transfer functions in the discrete domain without considering the computational delay. The computational delay or transport lag is a time delay between the end of the integration process and the new NCO input (i.e., phase rate estimate), which is incurred by the calculation process at the discriminator and the loop filter. The computational delay  $t_D$  exists in hardware and real-time software receivers without an appropriate buffering capability to eliminate this delay and should be considered in the stability analysis [14].

**Table 3.** Closed-loop filter coefficients for different digital representations for the integrator in a loop filter and a numerically controlled oscillator (NCO) (zero computational delay assumed).

Closed-Loop Transfer Function: $H(s) = \frac{N(s)F(s)}{1+N(s)F(s)}$	F(s) Transform Method	N(s) Transform Method		
		Step-Invariant $s \leftarrow \frac{z-1}{T}$	Impulse-Invariant $s \leftarrow \frac{z-1}{Tz}$	Bilinear $s \leftarrow \frac{2}{T} \frac{z-1}{z+1}$
First-order loop $H_1(s) = \frac{\omega_0}{s+\omega_0}$ $\omega_0 = 4B$	-	$\beta_0 = T\omega_0$ $\alpha_1 = 1$ $\alpha_0 = T\omega_0 - 1$	$\beta_1 = T\omega_0$ $\beta_0 = 0$ $\alpha_1 = T\omega_0 + 1$ $\alpha_0 = -1$	$\beta_1 = T\omega_0$ $\beta_0 = T\omega_0$ $\alpha_1 = T\omega_0 + 2$ $\alpha_0 = T\omega_0 - 2$
	Step-invariant $s \leftarrow \frac{z-1}{T}$	$\beta_1 = a_2T\omega_0$ $\beta_0 = T^2\omega_0^2 - a_2T\omega_0$ $\alpha_2 = 1$ $\alpha_1 = a_2T\omega_0 - 2$ $\alpha_0 = T^2\omega_0^2 - a_2T\omega_0 + 1$	$\beta_2 = a_2T\omega_0$ $\beta_1 = T^2\omega_0^2 - a_2T\omega_0$ $\beta_0 = 0$ $\alpha_2 = a_2T\omega_0 + 1$ $\alpha_1 = T^2\omega_0^2 - a_2T\omega_0 - 2$ $\alpha_0 = 1$	$\beta_2 = a_2T\omega_0$ $\beta_1 = T^2\omega_0^2$ $\beta_0 = T^2\omega_0^2 - a_2T\omega_0$ $\alpha_2 = a_2T\omega_0 + 2$ $\alpha_1 = T^2\omega_0^2 - 4$ $\alpha_0 = T^2\omega_0^2 - a_2T\omega_0 + 2$
Second-order loop $H_2(s) = \frac{a_2\omega_0s + \omega_0^2}{s^2 + a_2\omega_0s + \omega_0^2}$ $\omega_0 = 1.89B$ $a_2 = \sqrt{2}$	Impulse-invariant $s \leftarrow \frac{z-1}{Tz}$	$\beta_1 = T^2\omega_0^2 + a_2T\omega_0$ $\beta_0 = -a_2T\omega_0$ $\alpha_2 = 1$ $\alpha_1 = T^2\omega_0^2 + a_2T\omega_0 - 2$ $\alpha_0 = -a_2T\omega_0 + 1$	$\beta_2 = T^2\omega_0^2 + a_2T\omega_0$ $\beta_1 = -a_2T\omega_0$ $\beta_0 = 0$ $\alpha_2 = T^2\omega_0^2 + a_2T\omega_0 + 1$ $\alpha_1 = -a_2T\omega_0 - 2$ $\alpha_0 = 1$	$\beta_2 = T^2\omega_0^2 + a_2T\omega_0$ $\beta_1 = T^2\omega_0^2$ $\beta_0 = -a_2T\omega_0$ $\alpha_2 = T^2\omega_0^2 + a_2T\omega_0 + 2$ $\alpha_1 = T^2\omega_0^2 - 4$ $\alpha_0 = -a_2T\omega_0 + 2$
	Bilinear $s \leftarrow \frac{2}{T} \frac{z-1}{z+1}$	$\beta_1 = T^2\omega_0^2 + 2a_2T\omega_0$ $\beta_0 = T^2\omega_0^2 - 2a_2T\omega_0$ $\alpha_2 = 2$ $\alpha_1 = T^2\omega_0^2 + 2a_2T\omega_0 - 4$ $\alpha_0 = T^2\omega_0^2 - 2a_2T\omega_0 + 2$	$\beta_2 = T^2\omega_0^2 + 2a_2T\omega_0$ $\beta_1 = T^2\omega_0^2 - 2a_2T\omega_0$ $\beta_0 = 0$ $\alpha_2 = T^2\omega_0^2 + 2a_2T\omega_0 + 2$ $\alpha_1 = T^2\omega_0^2 - 2a_2T\omega_0 - 4$ $\alpha_0 = 2$	$\beta_2 = T^2\omega_0^2 + 2a_2T\omega_0$ $\beta_1 = 2T^2\omega_0^2$ $\beta_0 = T^2\omega_0^2 - 2a_2T\omega_0$ $\alpha_2 = T^2\omega_0^2 + 2a_2T\omega_0 + 4$ $\alpha_1 = 2T^2\omega_0^2 - 8$ $\alpha_0 = T^2\omega_0^2 - 2a_2T\omega_0 + 4$

Table 3. Cont.

Closed-Loop Transfer Function: $H(s) = \frac{N(s)F(s)}{1+N(s)F(s)}$	F(s) Transform Method	N(s) Transform Method		
		Step-Invariant $s \leftarrow \frac{z-1}{T}$	Impulse-Invariant $s \leftarrow \frac{z-1}{Tz}$	Bilinear $s \leftarrow \frac{2}{T} \frac{z-1}{z+1}$
Third-order loop $H_3(s) = \frac{b_3\omega_0 s^2 + a_3\omega_0^2 s + \omega_0^3}{s^3 + b_3\omega_0 s^2 + a_3\omega_0^2 s + \omega_0^3}$ $\omega_0 \approx 1.2B$ $a_3 \approx 1.1$ $b_3 \approx 2.4$	Step-invariant $s \leftarrow \frac{z-1}{T}$	$\beta_2 = b_3 T \omega_0$ $\beta_1 = a_3 T^2 \omega_0^2 - 2b_3 T \omega_0$ $\beta_0 = T^3 \omega_0^3 - a_3 T^2 \omega_0^2 + b_3 T \omega_0$ $\alpha_3 = 1$ $\alpha_2 = b_3 T \omega_0 - 3$ $\alpha_1 = a_3 T^2 \omega_0^2 - 2b_3 T \omega_0 + 3$ $\alpha_0 = T^3 \omega_0^3 - a_3 T^2 \omega_0^2 + b_3 T \omega_0 - 1$	$\beta_3 = b_3 T \omega_0$ $\beta_2 = a_3 T^2 \omega_0^2 - 2b_3 T \omega_0$ $\beta_1 = T^3 \omega_0^3 - a_3 T^2 \omega_0^2 + b_3 T \omega_0$ $\beta_0 = 0$ $\alpha_3 = b_3 T \omega_0 + 1$ $\alpha_2 = a_3 T^2 \omega_0^2 - 2b_3 T \omega_0 - 3$ $\alpha_1 = T^3 \omega_0^3 - a_3 T^2 \omega_0^2 + b_3 T \omega_0 + 3$ $\alpha_0 = -1$	$\beta_3 = b_3 T \omega_0$ $\beta_2 = a_3 T^2 \omega_0^2 - b_3 T \omega_0$ $\beta_1 = T^3 \omega_0^3 - b_3 T \omega_0$ $\beta_0 = T^3 \omega_0^3 - a_3 T^2 \omega_0^2 + b_3 T \omega_0$ $\alpha_3 = b_3 T \omega_0 + 2$ $\alpha_2 = a_3 T^2 \omega_0^2 - b_3 T \omega_0 - 6$ $\alpha_1 = T^3 \omega_0^3 - b_3 T \omega_0 + 6$ $\alpha_0 = T^3 \omega_0^3 - a_3 T^2 \omega_0^2 + b_3 T \omega_0 - 2$
	Impulse-invariant $s \leftarrow \frac{z-1}{Tz}$	$\beta_2 = T^3 \omega_0^3 + a_3 T^2 \omega_0^2 + b_3 T \omega_0$ $\beta_1 = -a_3 T^2 \omega_0^2 - 2b_3 T \omega_0$ $\beta_0 = b_3 T \omega_0$ $\alpha_3 = 1$ $\alpha_2 = T^3 \omega_0^3 + a_3 T^2 \omega_0^2 + b_3 T \omega_0 - 3$ $\alpha_1 = -a_3 T^2 \omega_0^2 - 2b_3 T \omega_0 + 3$ $\alpha_0 = b_3 T \omega_0 - 1$	$\beta_3 = T^3 \omega_0^3 + a_3 T^2 \omega_0^2 + b_3 T \omega_0$ $\beta_2 = -a_3 T^2 \omega_0^2 - 2b_3 T \omega_0$ $\beta_1 = b_3 T \omega_0$ $\beta_0 = 0$ $\alpha_3 = T^3 \omega_0^3 + a_3 T^2 \omega_0^2 + b_3 T \omega_0 + 1$ $\alpha_2 = -a_3 T^2 \omega_0^2 - 2b_3 T \omega_0 - 3$ $\alpha_1 = b_3 T \omega_0 + 3$ $\alpha_0 = -1$	$\beta_3 = T^3 \omega_0^3 + a_3 T^2 \omega_0^2 + b_3 T \omega_0$ $\beta_2 = T^3 \omega_0^3 - b_3 T \omega_0$ $\beta_1 = -a_3 T^2 \omega_0^2 - b_3 T \omega_0$ $\beta_0 = b_3 T \omega_0$ $\alpha_3 = T^3 \omega_0^3 + a_3 T^2 \omega_0^2 + b_3 T \omega_0 + 2$ $\alpha_2 = T^3 \omega_0^3 - b_3 T \omega_0 - 6$ $\alpha_1 = -a_3 T^2 \omega_0^2 - b_3 T \omega_0 + 6$ $\alpha_0 = b_3 T \omega_0 - 2$
	Bilinear $s \leftarrow \frac{2}{T} \frac{z-1}{z+1}$	$\beta_2 = T^3 \omega_0^3 + 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0$ $\beta_1 = 2T^3 \omega_0^3 - 8b_3 T \omega_0$ $\beta_0 = T^3 \omega_0^3 - 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0$ $\alpha_3 = 4$ $\alpha_2 = T^3 \omega_0^3 + 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0 - 12$ $\alpha_1 = 2T^3 \omega_0^3 - 8b_3 T \omega_0 + 12$ $\alpha_0 = T^3 \omega_0^3 - 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0 - 4$	$\beta_3 = T^3 \omega_0^3 + 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0$ $\beta_2 = 2T^3 \omega_0^3 - 8b_3 T \omega_0$ $\beta_1 = T^3 \omega_0^3 - 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0$ $\beta_0 = 0$ $\alpha_3 = T^3 \omega_0^3 + 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0 + 4$ $\alpha_2 = 2T^3 \omega_0^3 - 8b_3 T \omega_0 - 12$ $\alpha_1 = T^3 \omega_0^3 - 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0 + 12$ $\alpha_0 = -4$	$\beta_3 = T^3 \omega_0^3 + 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0$ $\beta_2 = 3T^3 \omega_0^3 + 2a_3 T^2 \omega_0^2 - 4b_3 T \omega_0$ $\beta_1 = 3T^3 \omega_0^3 - 2a_3 T^2 \omega_0^2 - 4b_3 T \omega_0$ $\beta_0 = T^3 \omega_0^3 - 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0$ $\alpha_3 = T^3 \omega_0^3 + 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0 + 8$ $\alpha_2 = 3T^3 \omega_0^3 + 2a_3 T^2 \omega_0^2 - 4b_3 T \omega_0 - 24$ $\alpha_1 = 3T^3 \omega_0^3 - 2a_3 T^2 \omega_0^2 - 4b_3 T \omega_0 + 24$ $\alpha_0 = T^3 \omega_0^3 - 2a_3 T^2 \omega_0^2 + 4b_3 T \omega_0 - 8$

The computational delay can be modeled as an additional delay unit for the amount of  $t_D$  located before the NCO [14], which results in the modification of the NCO transfer function  $N(z)$  as:

$$N_D(z) = z^{-\frac{t_D}{T}} N(z) \quad (6)$$

where the subscript D means the computational delay included. Utilizing the modified NCO transfer function  $N_D(z)$ , the closed-loop transfer function  $H(z)$  of (4) is modified to contain the computational delay as follows:

$$H_D(z) = \frac{z^{-\frac{t_D}{T}} N(z) F(z)}{1 + z^{-\frac{t_D}{T}} N(z) F(z)} = \frac{N_D(z) F(z)}{1 + N_D(z) F(z)} \quad (7)$$

By assuming that the new NCO input is obtained from the integration results at the previous epoch of the loop filter, as a typical example, the  $t_D$  becomes  $T$ . Then, the  $N_D(z)$  contains unit delay as a computational delay as follows:

$$N_D(z) = z^{-1} N(z) \quad (8)$$

and is used for the stability analysis of the DPLLs in case of computational delay are considered.

### 3. Stability Analysis for Upper Limit

In practice, DPLLs are a type of sampled-data loop that is never unconditionally stable; high gain loops always result in instability because of the inherent computational delay, which constitutes a major potential drawback of this type of system. Moreover, a stability problem occurs for DPLLs when the loop bandwidth is insufficiently small relative to the loop update interval. Therefore, the upper and lower limits on the value of  $BT$  that ensure the stability margin and the gain margin of the DPLLs should be determined. The upper limits for DPLLs are obtained from the stability analysis in this Section.

#### 3.1. Root Loci

As is well known in control theory, the stability of linear time-invariant systems in the continuous-time domain can be determined by determining the location of the roots of the characteristic equation of the system. For bounded-input-bounded-output stability, the roots of the characteristic equation (i.e., the poles of the system) must all lie in the left half of the  $s$ -plane. Therefore, the stability of the system can be determined by checking the root locus diagram, which is a pictorial representation of the poles of the closed-loop transfer function as a function of the loop gain.

The root locus diagrams for discrete-data systems are constructed in the  $z$ -plane using essentially the same properties as those of the continuous-data systems in the  $s$ -plane, except that the relationship between the root locations and stability must be made concerning the unit circle  $|z| = 1$  in the  $z$ -plane [10]. Here, the loci of roots are defined when only  $BT$  varies instead of the loop gain  $K$  (unity gain is assumed for the discriminator gain  $A$ ) to test the stability of a DPLL as a function of  $BT$ . The closed-loop transfer function of the DPLL in (7) is then rewritten as:

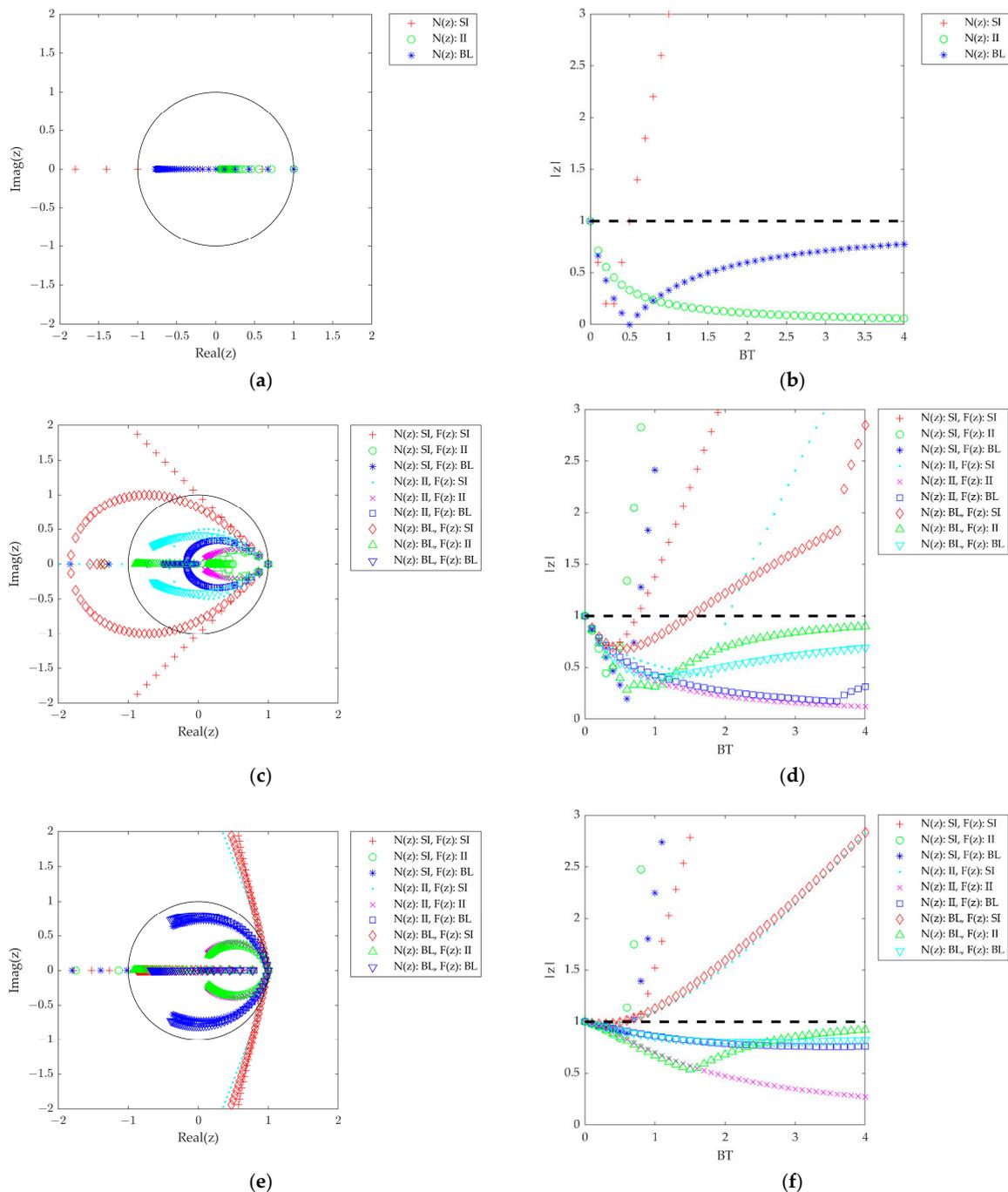
$$H_D(z) = \frac{(z - z_m)(z - z_{m-1}) \cdots (z - z_0)}{(z - p_n)(z - p_{n-1}) \cdots (z - p_0)} = \frac{\prod_{j=0}^m (z - z_j)}{\prod_{i=0}^n (z - p_i)} \quad (9)$$

where  $p_i$  and  $z_j$  represent the poles and zeroes of the system, respectively.

It is obvious that as  $B$  increases, the roots of the closed-loop transfer functions in the  $s$ -domain deviate from the origin, but still stay in the left half of the  $s$ -plane for all values (i.e., the system is unconditionally stable). However, for the second- or higher-order systems, such pole deviation causes oscillation of their step responses.

The left-side figures of Figure 3 illustrate the same effect in the  $z$ -plane for the first-/second-/third-order closed loops as described above but with different numerical integration methods for  $N(z)$  and  $F(z)$ , and with the assumption of the zero computational

delays. The trajectory for all cases begins at  $z = 1$  when  $BT = 0$  and then moves to the inside or outside of the unit circle as  $BT$  increases. The presence of roots at  $|z| = 1$  causes the step response of the system to oscillate with a constant amplitude, and the system becomes marginally stable. Thus, the DPLL will be unstable for  $|z| > 1$ , even if its counterpart in the continuous-time domain is unconditionally stable. Therefore, the value of  $BT$  that makes the system marginally stable is the upper limit of available  $BT$  values for the stable DPLL. The limit is defined as  $BT_{osc}$  which indicates the  $BT$  when  $|z| = 1$  (i.e.,  $BT$  at the crossing point between the root trajectory and the unit circle) and is the point that makes the system marginally stable, so the step response begins to oscillate at this point.



**Figure 3.** Root location of the DPLL closed-loop transfer functions with three different integration methods for  $N(z)$  and  $F(z)$  ( $t_D = 0$ ). (a) First-order root loci. (b) First-order root magnitude. (c) Second-order root loci. (d) Second-order root magnitude. (e) Third-order root loci. (f) Third-order root magnitude.

The right-side figures of Figure 3 show  $|z|$  for the same case as discussed above with respect to  $BT$ . As mentioned earlier,  $|z|$  should be less than 1 for a stable system. For the first-order DPLL in Figure 3b, the root loci for SI are outside the unit circle in the  $z$ -domain for  $BT > 0.5$  (i.e.,  $BT_{osc} = 0.5$ ), whereas those for BL and II converge to values on the unit circle ( $|z| = 1$ ) and the origin ( $|z| = 0$ ), respectively, as  $BT$  increases. A similar phenomenon is observed in the second- and third-order cases. Therefore, three types of DPLL systems are identified in terms of system stability while varying  $BT$  as follows (see Table 4):

1. Type A—The poles drift outside the unit circle in the  $z$ -domain as  $BT$  increases. The system is stable only when  $BT \leq BT_{osc}$ , otherwise it is unstable;
2. Type B—All of the poles are within the unit circle but approach  $|z| = 1$  as  $BT$  increases. The instability of the system increases as  $BT$  increases;
3. Type C—All of the poles of the system are located inside the unit circle. The system is unconditionally stable, even for large  $BT$  values.

**Table 4.** Three types of DPLL systems regarding stability.

Type	Descriptions	Remarks
A	$ z  \leq 1$ only when $BT \leq BT_{osc}$ , otherwise $ z  > 1$	Conditionally stable only when $BT \leq BT_{osc}$
B	$ z  \leq 1$ for all $BT$ s, but $ z  \rightarrow 1$ as $BT$ increases	The instability increases as $BT$ increases
C	$ z  < 1$ for all $BT$ s, and $ z  \sim 0$ as $BT$ increases	Unconditionally stable

Figure 4 presents the same plots as Figure 3, but the unit delay is included in the NCO transfer function as (8) for the assumption of the computational delay (i.e.,  $t_D = T$  assumed) in this case. Unlike the zero computational delay case, regardless of the used digital integration method, all the root loci magnitude exceed  $|z| = 1$  as the  $BT$  increases. For instance, as can be seen in Figure 4b, the II and BL of the first-order DPLL, which are stable for all  $BT$ s with zero computational delays, are now stable only at  $BT \leq 0.51$  (i.e.,  $BT_{osc} = 0.51$ ) because of the computational delay. Similar results can be observed for the second- and third-order DPLLs. Therefore, it is viewed that the computational delay in the NCO degrades the stability of the DPLLs. Consequently, such DPLLs are conditionally stable.

Table 5 summarizes the stability conditions of the  $BT$  for different integration methods in the loop filter and the NCO. For the zero computational delays, the II method plays an important role in making the system unconditionally stable (Type C) and the SI method forces the system to be conditionally stable (i.e., Type A). However, as discussed above, if the unit computational delay is included in the stability analysis, all the methods are conditionally stable (i.e., Type A) regardless of the filter order or integration method. Furthermore, all the  $BT_{osc}$  for  $t_D = T$  are located within 0.25 and 0.75 which does not exceed 1, so the results are matched with the well-known rule-of-thumb  $BT$  threshold ( $BT \approx 1$ ) for the stable operation of the DPLL.

**Table 5.** Marginal stability conditions of the  $BT$  for different integration methods in the loop filter and NCO.

Filter Order	NCO, $N(z)$	Loop Filter, $F(z)$	Zero Computational Delay ( $t_D = 0$ )		Unit Computational Delay ( $t_D = T$ )	
			$BT_{osc}^*$	Type	$BT_{osc}$	Type
First	SI	-	0.51	A	0.26	A
	II	-	No limit	C	0.51	A
	BL	-	$ z  \sim 1$ when $BT \gg 1$	B	0.51	A

Table 5. Cont.

Filter Order	NCO, $N(z)$	Loop Filter, $F(z)$	Zero Computational Delay ( $t_D = 0$ )		Unit Computational Delay ( $t_D = T$ )	
			$BT_{osc}^*$	Type	$BT_{osc}$	Type
Second	SI	SI	0.75	A	0.27	A
		II	0.55	A	0.25	A
		BL	0.75	A	0.27	A
	II	SI	2.05	A	0.75	A
		II	No limit	C	0.55	A
		BL	$ z  \sim 1$ when $BT \gg 1$	B	0.75	A
BL	SI	1.5	A	0.41	A	
	II	$ z  \sim 1$ when $BT \gg 1$	B	0.43	A	
	BL	$ z  \sim 1$ when $BT \gg 1$	B	0.44	A	
Third	SI	SI	0.53	A	0.38	A
		II	0.58	A	0.29	A
		BL	0.70	A	0.33	A
	II	SI	0.57	A	0.53	A
		II	No limit	C	0.58	A
		BL	$ z  \sim 1$ when $BT \gg 1$	B	0.70	A
BL	SI	0.53	A	0.51	A	
	II	$ z  \sim 1$ when $BT \gg 1$	B	0.49	A	
	BL	$ z  \sim 1$ when $BT \gg 1$	B	0.60	A	

\* Values for  $BT$  should be equal to or less than this value for stability.

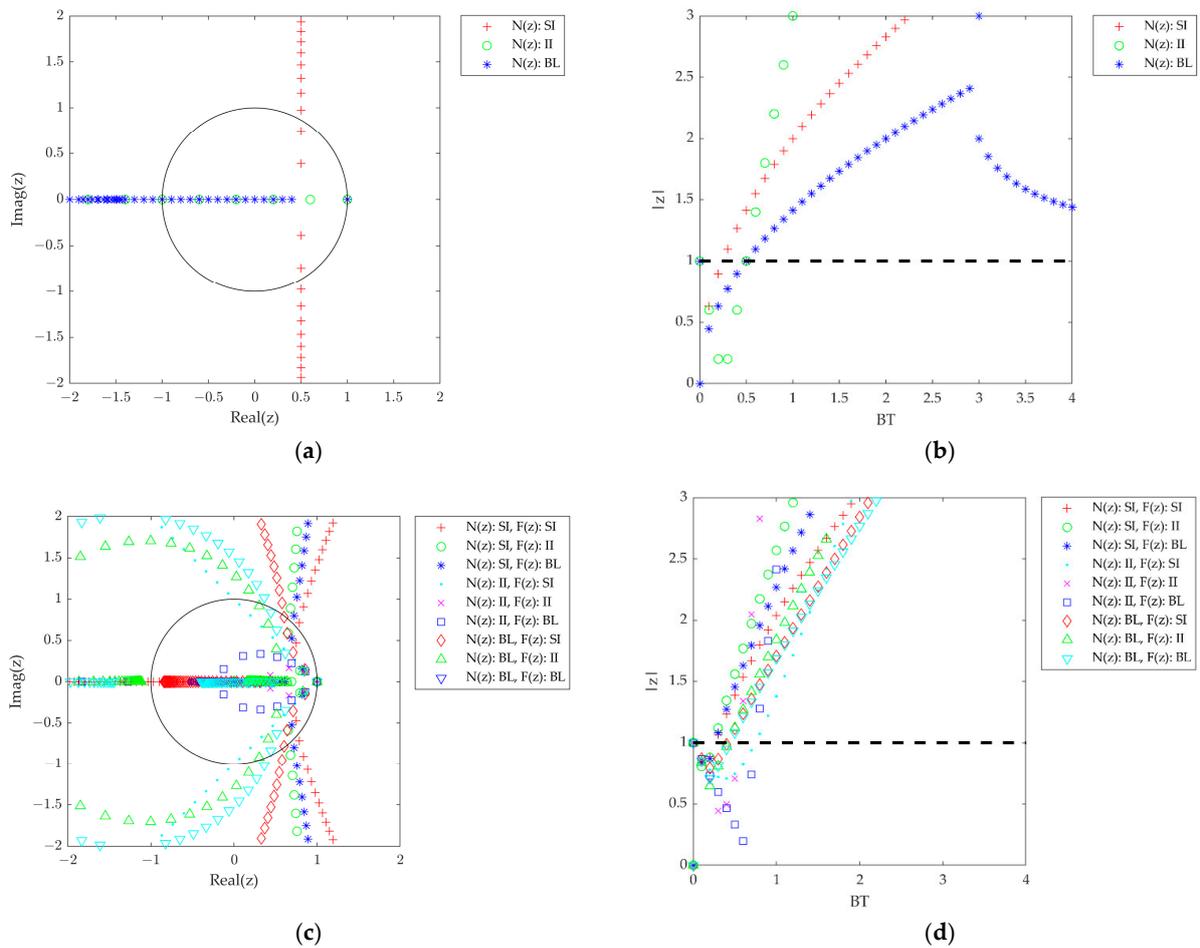
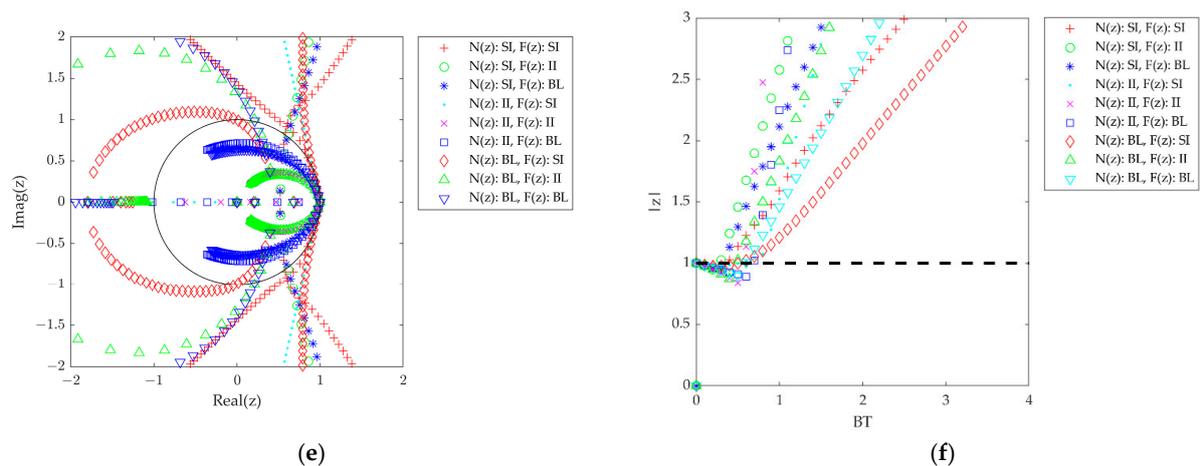


Figure 4. Cont.



**Figure 4.** Root location of the DPLL closed-loop transfer functions with three different integration methods for  $N(z)$  and  $F(z)$  ( $t_D = T$ ). (a) First-order root loci. (b) First-order root magnitude. (c) Second-order root loci. (d) Second-order root magnitude. (e) Third-order root loci. (f) Third-order root magnitude.

### 3.2. BT Margin

In system theory, the gain margin is the ratio of the maximum loop gain with stability to the loop gain at the design point and is used to determine the region of stability. Typically, the gain margin is obtained using the ratio of the minimum damping parameter that produces instability to the minimum damping parameter at the design point because the loop gain is proportional to the damping parameter. A similar concept is utilized to determine the region of stability of a system that has a unity gain for different values of  $BT$ .

Defining the normalized bandwidth margin as the ratio of the maximum  $BT$  value with stability to the  $BT$  value at the design point, the margin can be calculated as follows:

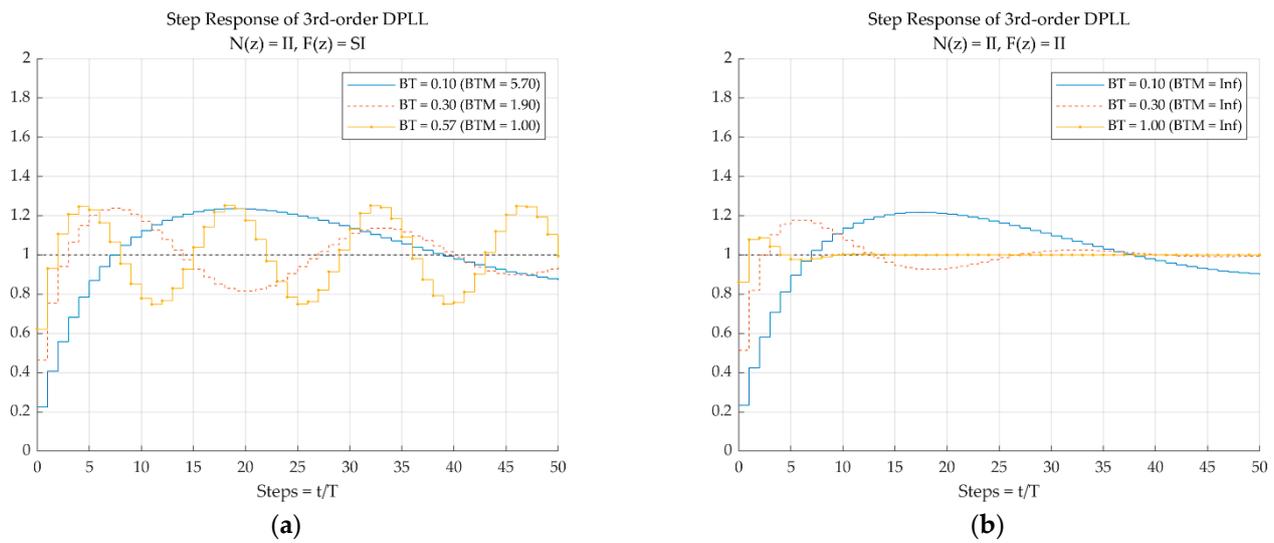
$$BTM = \frac{BT_{osc}}{BT} \quad (10)$$

Note that the normalized bandwidth margin is inversely proportional to  $BT$  and that for the same value of  $B$  within the stable region (i.e.,  $BT < BT_{osc}$ ), a larger loop update interval makes the system reduce the capacity of the sampled output to accurately measure the true output.

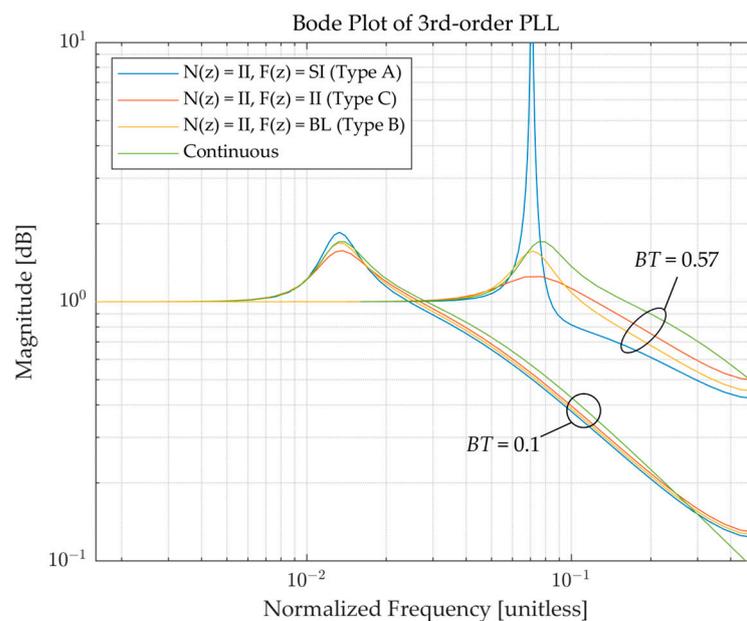
### 3.3. Step and Frequency Responses

Figure 5 shows an example of the step response of the third-order closed loop function of DPLLs with the numerical integration combination of  $N(z) = II$  and  $F(z) = SI$  [Type A in Figure 5a] and II [Type C in Figure 5b] for three  $BT$  values. Since all the DPLL types are the same when the computational delay is included, zero computational delays are assumed here to observe the behaviors of the different types. The Type A system oscillates rapidly as  $BT$  increases and becomes unstable when  $BT = 0.57$ , which is the value of  $BT_{osc}$  for the II-SI combination when  $t_D = 0$  (see Table 5), whereas the Type C system remains stable, even for  $BT = 1$ .

Figure 6 presents the Bode plots of the same case as discussed above with the addition of  $F(z) = BL$  (Type B) and its continuous-time domain counterpart for small and large  $BT$ . When  $BT$  is sufficiently small relative to  $BT_{osc}$ , the frequency responses of the DPLLs accurately represent the system, and little distortion is observed for all numerical integration methods. However, the distortion causes the DPLLs to oscillate at maximum amplitudes at larger  $BT$  when  $BT$  approaches the system's resonant frequency  $BT_{osc}$ . This clarifies the fact that the design parameter for the PLL is the bandwidth of the loop  $B$ ; however, the normalized bandwidth  $BT$  in DPLL does not reflect the true noise equivalent bandwidth of the loop as  $BT$  increases, and at the same time, the deviation of the actual bandwidth of the digital system is affected by the type of numerical integration method.



**Figure 5.** Step responses of the third-order closed-loop transfer function of the DPLL with different numerical integrations for  $N(z)$  and  $F(z)$  for different  $BT$  ( $t_D = 0$ ). (a) Type A— $N(z) = II$  and  $F(z) = SI$  (b) Type C— $N(z) = II$  and  $F(z) = II$ .



**Figure 6.** Bode plots of the third-order closed-loop transfer functions of the PLL with different numerical integrations for  $N(z)$  and  $F(z)$  for different  $BT$  ( $t_D = 0$ ).

#### 4. $BT$ Lower Limit

As revealed from the previous section, the  $BT$  upper limits exist for each numerical integration method for the DPLL, where  $BT$ s larger than such a limit cannot assure the stable operation of the DPLL. Similarly, lower limits can exist that the  $BT$ s lower than the limit fails to track the carrier phase successfully. The reason for this is that the gain (i.e.,  $BT$ ) of the tracking loop is too small to catch up with the dynamics induced by the receiver dynamic stress and the oscillator phase noise. Therefore, the  $BT$  lower limits can be drawn by analyzing the DPLL measurement error with its threshold for stable tracking. Here, the third-order DPLL is considered for the sake of simplicity, however, a similar approach can be applied to first-/second-order DPLLs analogously.

#### 4.1. Measurement Error

The DPLL measurement error is the standard deviation of the estimated phase error in the DPLL, and the thermal noise contributes a huge portion of it. Nevertheless, the oscillator phase noise and dynamic stress should be taken into account if a low-quality oscillator is used as the clock source, or the receiver faces large dynamics. The oscillator phase noise induced by the vibration is not considered in this work for the simplicity of analysis. The standard deviation of each error source is obtained individually and merged later to form the overall measurement error which can be compared with the threshold.

##### 4.1.1. Thermal Noise

The thermal noise of the Costas PLL which is used for the phase tracking of the data channel (i.e., phase transition existing channel) is modeled as [2]:

$$\sigma_t = \frac{180}{\pi} \sqrt{\frac{B}{C/N_0} \left(1 + \frac{1}{2TC/N_0}\right)} \text{ [deg]} \quad (11)$$

where  $C/N_0$  is the carrier-to-noise density ratio expressed in a linear scale [Hz].

##### 4.1.2. Allan Deviation Phase Noise

An ideal oscillator is represented as a sinusoidal wave, but a realistic oscillator suffers from phase modulation which has a random characteristic. The phase modulation results in frequency instability which can be described using the Allan deviation. Finally, the phase noise of the third-order DPLL induced by the Allan deviation is given by [15]:

$$\sigma_A = \frac{180}{\pi} \sqrt{2\pi^2 f_c^2 \left( \frac{\pi^2 h_{-2}}{3\omega_0^3} + \frac{\pi h_{-1}}{3\sqrt{3}\omega_0^2} + \frac{h_0}{6\omega_0} \right)} \text{ [deg]} \quad (12)$$

where  $f_c$  is the carrier frequency [Hz] and  $h_{-2}$ ,  $h_{-1}$ ,  $h_0$  are the clock parameters which are determined by the quality of the oscillator. As can be observed from the equation,  $\sigma_A$  is inversely proportional to the  $\omega_0$ , which is proportional to the  $B$ . That is, DPLLs with sufficient  $B$  can effectively suppress the clock jitter, otherwise, the phase tracking performance is determined by the quality of the oscillator. In this study, two types of oscillators are utilized for the phase noise analysis, namely a temperature-compensated crystal oscillator (TCXO) and oven-controlled crystal oscillator (OCXO). Clock parameters are obtained using the model provided by [15], which are listed in Table 6.

**Table 6.** Clock parameters for the calculation of the Allan deviation oscillator phase noise.

Oscillator Type	$h_0$ [s]	$h_{-1}$ [-]	$h_{-2}$ [1/s]
Temperature-compensated crystal oscillator (TCXO)	$1.00 \times 10^{-21}$	$1.00 \times 10^{-20}$	$2.00 \times 10^{-20}$
Oven-controlled crystal oscillator (OCXO)	$2.51 \times 10^{-26}$	$2.51 \times 10^{-23}$	$2.51 \times 10^{-22}$

##### 4.1.3. Dynamic Stress Error

The DPLL suffers from the dynamic stress error when the receiver-satellite line-of-sight (LOS) range varies rapidly, for example, if the receiver moves fast. The endurable order of the dynamic stress is related to the order of the DPLL as described in Table 1. The dynamic stress error of the DPLL is incurred by the dynamic component that has higher order than the trackable component of the DPLL and can be generally modeled as [2]:

$$\theta_e = \frac{d^n R / dt^n}{\omega_0^n} \text{ [deg]} \quad (13)$$

where  $n$  is the loop filter order and  $d^n R/dt^n$  is the LOS dynamic stress [deg/s <sup>$n$</sup> ] of the  $n$ -th order DPLL. The typically used second-/third-order DPLLs for the GNSS receiver suffer from the  $d^2 R/dt^2$  and  $d^3 R/dt^3$ , which are the acceleration stress and jerk stress, respectively.

#### 4.1.4. Overall Measurement Error

The overall DPLL measurement error is composed of the thermal noise, oscillator phase noise, and dynamic stress error as follows [2]:

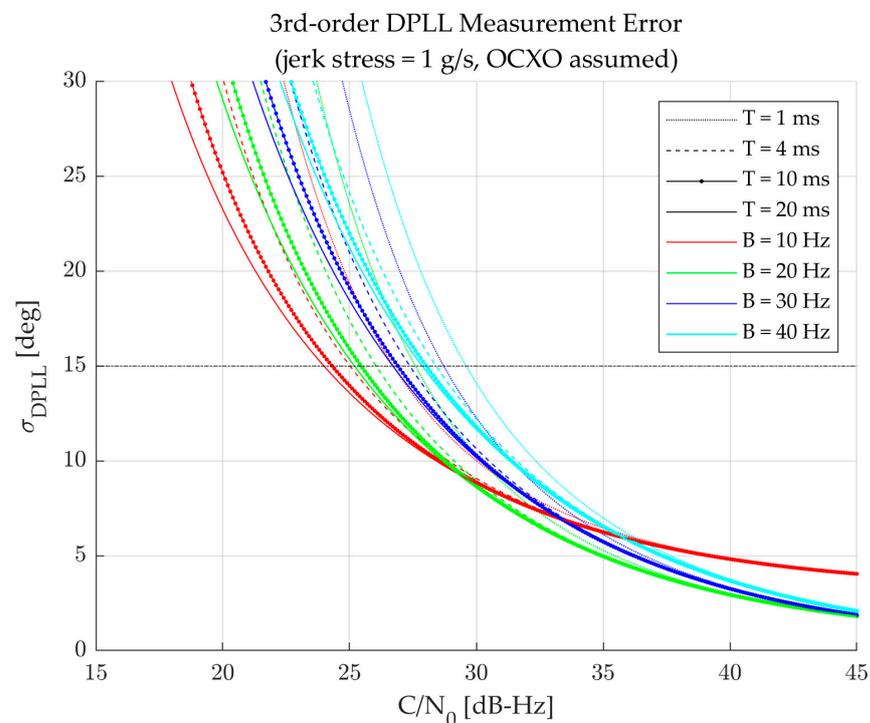
$$\sigma_{\text{DPLL}} = \sqrt{\sigma_t^2 + \sigma_A^2} + \frac{\theta_e}{3} \leq 15 \text{ deg} \quad (14)$$

and can be calculated by substituting (11) to (13) into (14). The  $3\sigma_{\text{DPLL}}$  must not exceed the rule-of-thumb DPLL tracking threshold (i.e., 45 deg), which is obtained by the 1/4 of the pull-in range (i.e., 180 deg) of the Costas PLL discriminator, consequently,  $\sigma_{\text{DPLL}}$  should be less than 15 deg to not to loss lock [2].

#### 4.2. Lower Limit

Since the  $\sigma_t$  in (11) is a function of the  $C/N_0$ , the resulting  $\sigma_{\text{DPLL}}$  in (14) varies with the given  $C/N_0$ . Figure 7 illustrates the  $\sigma_{\text{DPLL}}$  curves for the  $C/N_0$  for various  $B$  and  $T$  values, as an example. Obviously, the measurement error increases as the  $C/N_0$  decreases, and eventually crosses the measurement error threshold line (i.e., horizontal line at 15 deg). As the DPLL cannot operate above the threshold line, the  $C/N_0$  at the crossing point indicates the minimum  $C/N_0$  that the specific DPLL can track, which can be defined as a  $C/N_0$  threshold for the DPLL as:

$$C/N_{0,\text{thresh}} = \arg \min_{C/N_0 \in \mathbb{R}} |\sigma_{\text{DPLL}} - 15| \quad (15)$$



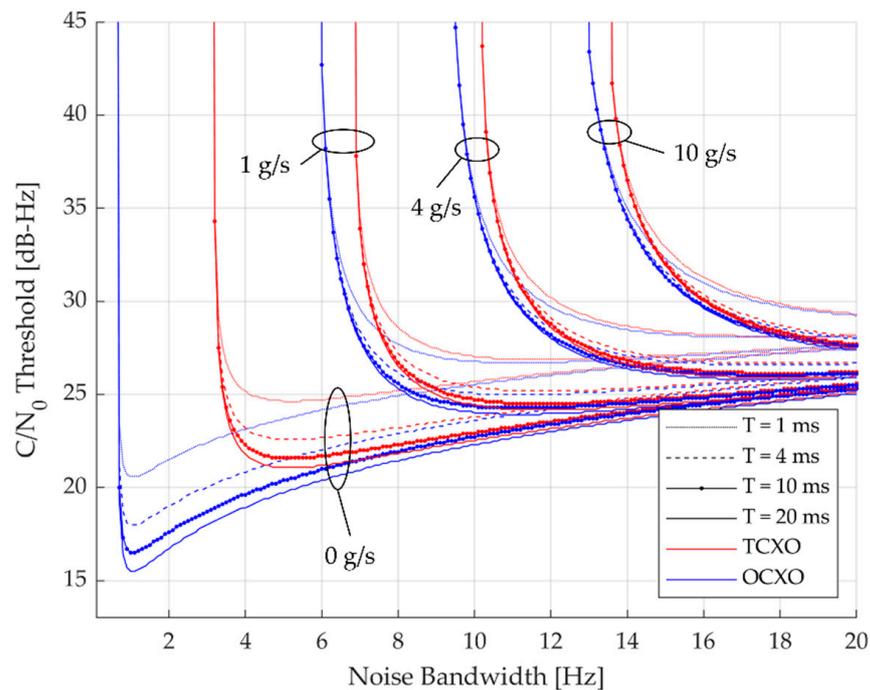
**Figure 7.** Third-order DPLL measurement error curves for the  $C/N_0$  for various  $B$  and  $T$  (jerk stress of 1 g/s and OCXO assumed).

The  $C/N_0$  thresholds of third-order DPLL for each measurement error parameter are calculated to form the  $C/N_0$  threshold curves presented in Figure 8. As the  $B$  becomes narrower, the  $C/N_0$  threshold declines smoothly, and at some point, it rapidly rises. Such

vertical lines represent the lower limits of the  $B$ , which means DPLLs that have  $B$ s smaller than this limit have insufficient gain to work well in that condition. Note that, as can be observed from the figure, the lower limit is dependent on the  $B$ , not  $BT$ , and  $T$  has a negligible impact on the lower limits. Therefore, the lower limits with respect to the  $B$  are induced first and the  $T$  is multiplied later to form the  $BT$  that is the main interest of this paper. The  $BT$  lower limit is represented as follows:

$$\begin{aligned} BT_{\text{low}} &= T \times B_{\text{min}} \\ &= T \times \arg \max_{B \in (0, \infty)} \left| \frac{\partial}{\partial B} (C/N_{0, \text{thresh}}) \right| \end{aligned} \quad (16)$$

where  $B_{\text{min}}$  is the minimum  $B$  [Hz] that the DPLL can operate at specific environmental conditions. In the case of no dynamic stress (0 g/s case), the  $B$  lower limits of TCXO and OCXO deviate from each other because all the phase noise observed as a dynamic component at the receiver is dominated by the oscillator phase noise. Clearly, the  $B$  limit of the OCXO is smaller than TCXO, which verifies that the OCXO has better clock performance than TCXO. One of the reasons why the high-quality oscillator (e.g., OCXO) is preferred for the application of weak signal processing can be inferred from the figure, as the OCXO can provide tracking capability down to approximately 15 dB-Hz of  $C/N_0$  even for  $T = 20$  ms. When the dynamic stress exists, the two oscillators have similar effects on the lower limit, and as the dynamics get stronger, the limits grow because the required gains increase as well.

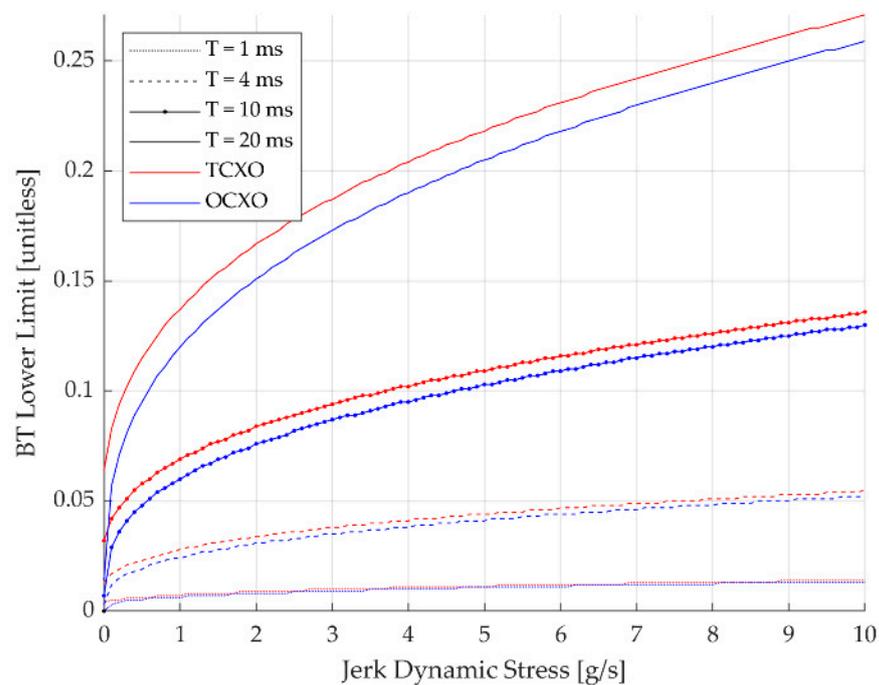


**Figure 8.**  $C/N_0$  threshold curves for  $B$  of third-order DPLL using TCXO and OCXO.

The obtained  $BT$  lower limits of third-order DPLL are arranged in Table 7. Figure 9 shows the  $BT$  lower limits for various jerk dynamic stresses visually. It can be observed that shorter  $T$  is beneficial in the dynamic environment since the  $T = 1$  ms have nearly the same  $BT$  limits for all jerk values. Additionally, OCXO has more advantages than TCXO when a longer  $T$  is used, as the difference between the  $BT$  limits of the oscillators becomes larger for the longer  $T$ .

**Table 7.** BT lower limits of third-order DPLL.

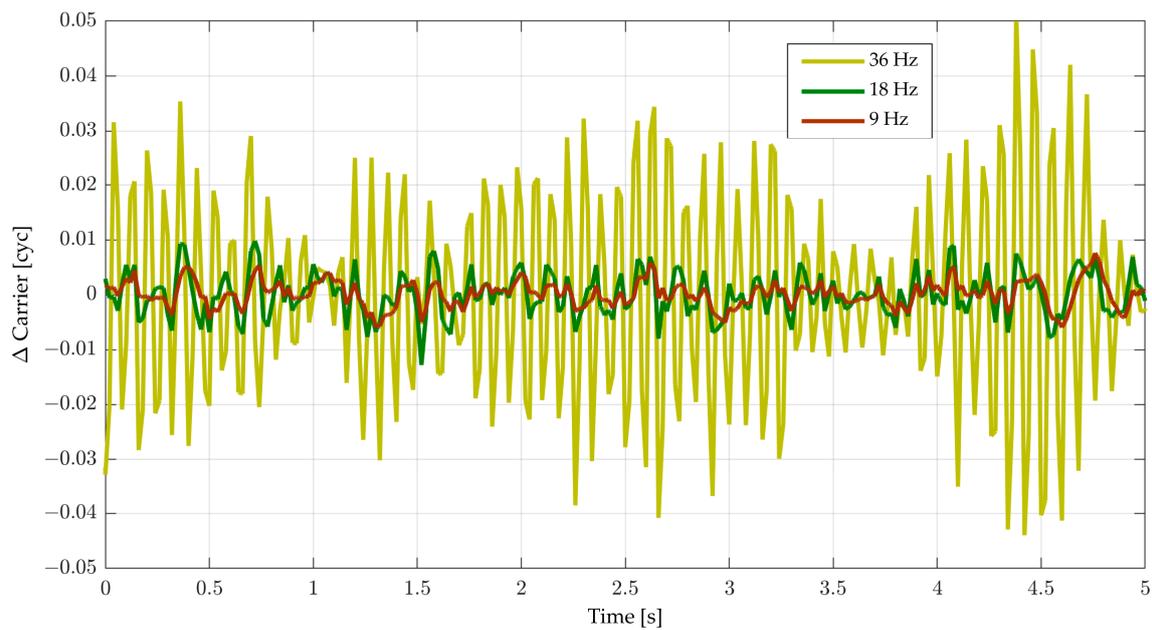
Jerk Stress [g/s]	Oscillator Type	BT Lower Limit			
		$T = 1$ ms	$T = 4$ ms	$T = 10$ ms	$T = 20$ ms
0	TCXO	0.004	0.013	0.032	0.064
	OCXO	<0.001	0.003	0.007	0.014
1	TCXO	0.007	0.028	0.069	0.137
	OCXO	0.006	0.024	0.060	0.120
4	TCXO	0.011	0.041	0.102	0.204
	OCXO	0.010	0.038	0.095	0.190
10	TCXO	0.014	0.055	0.136	0.271
	OCXO	0.013	0.052	0.130	0.259

**Figure 9.** BT lower limits of third-order DPLL for jerk dynamic stresses.

## 5. Implementation and Caveats

Focusing on the actual implementation of a second-order DPLL, the step-invariant model for the NCO and loop filter is commonly used. This model is implemented on a GNSS software receiver. For a simulated binary phase shift keying with a chipping rate of 1.023 MHz pilot signal with a constant Doppler shift, the carrier phase tracking error is shown in Figure 10. The largest bandwidth (36 Hz) results in  $BT = 0.72$ , which is near the stability limit ( $BT_{osc} = 0.75$ ) when  $t_D = 0$ . In this case, oscillations are visible, but carrier tracking remains stable.

At first glance, the implementation of a DPLL seems straightforward; however, two important flaws should be taken care of. Therefore, they are considered and implemented in the receiver to see the effects on the phase tracking performance of the DPLL.



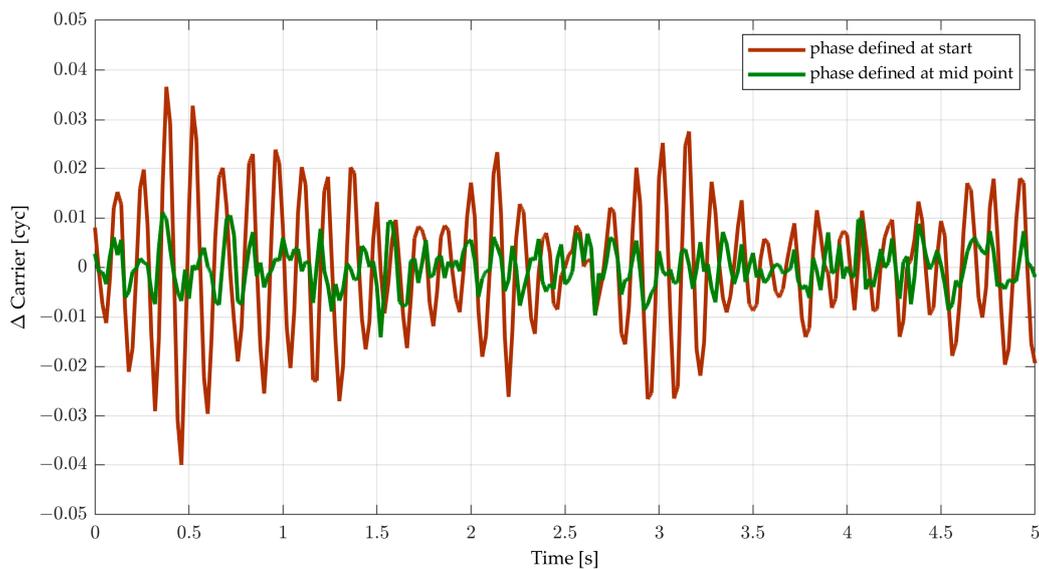
**Figure 10.** Carrier phase tracking error for  $T = 20$  ms, a second-order DPLL with SI/SI numerical integration,  $t_D = 0$ ,  $C/N_0 = 47.7$  dB-Hz and various bandwidths.

### 5.1. Incorrect Carrier Phase Reference Epoch

The correlator integrates correlation results from  $t$  to  $t + T$  and dumps it at  $t + T$ , then the phase detector (discriminator) estimates the error between the received true phase and the estimated phase in the NCO at  $t + T$ . The loop filter predicts the next phase and phase rate for  $t + 2T$  and controls the NCO accordingly. However, the phase error obtained at  $t + T$  actually indicates the phase error at  $t + T/2$ . This is because the integrated correlator outputs are the mean values between  $t$  and  $t + T$  and the resulting phase error is also the mean phase error during the integration period because it is calculated using the mean correlation results. Since the NCO rate is constant over that period and the incoming phase rate is likewise typically assumed to be constant for  $T$  in the digital loop, the true and estimated phase and the phase error change linearly in that duration. Consequently, the mean phase error reasonably represents the phase error for the midpoint in the integration process (which is at  $t + T/2$  in this case).

The reference epoch of the estimated phase can be defined differently by each receiver designer. Defining the phase to represent the start point of the integration has an effect such as the insertion of a delay for  $T/2$  (half-sample delay). In this work, for the simplicity of the analysis to easily assess the upper and lower limits of  $BT$ , it is assumed that the carrier phase is defined at the midpoint of the integration interval without any delay, which is an ideal case. This definition is reasonable because the possible Doppler error does not affect the carrier phase discriminator value, and the delay effect on carrier measurement can be compensated later at the measurement extraction module in the baseband process of receivers.

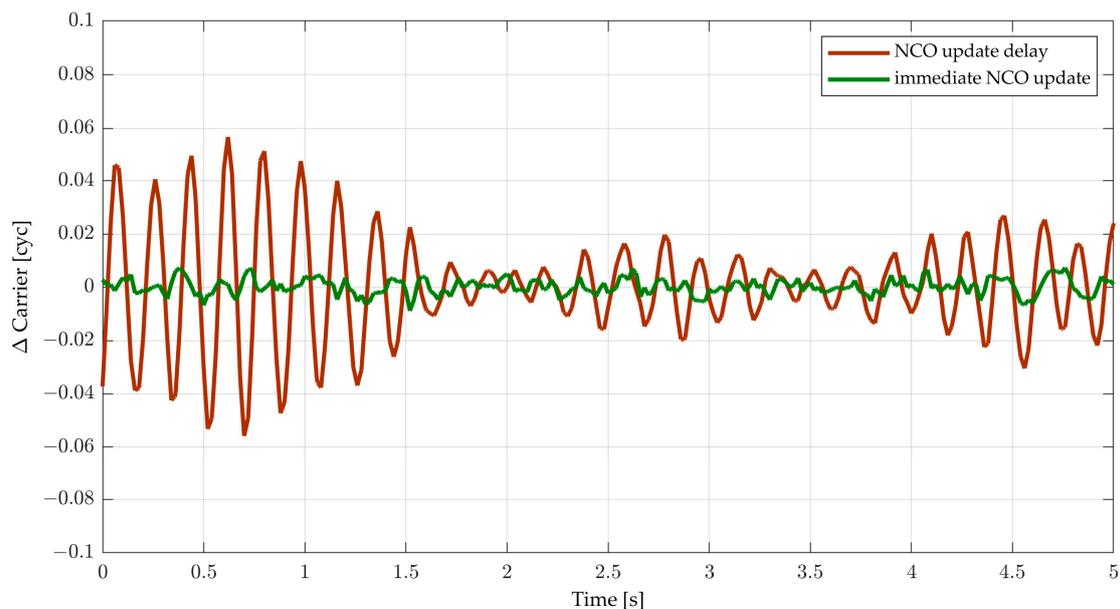
On the other hand, a real-world implementation of a DPLL might use a carrier phase value defined at the beginning of the integration interval for convenience. In this case, a Doppler error affects the carrier phase discriminator output (i.e., the Doppler-based phase error is the Doppler error multiplied by  $T/2$ ). If this assumption is used (without accounting for the Doppler error) in combination with the digitization scheme presented here, the tracking stability is degraded. This phenomenon is shown in Figure 11. In fact, the stability drops for a second-order PLL with the SI/SI scheme from  $BT = 0.75$  to  $BT \sim 0.4$  (determined on an empirical basis) when  $t_D = 0$ .



**Figure 11.** Carrier phase tracking error for  $T = 20$  ms, a 20 Hz second-order DPLL with SI/SI numerical integration,  $t_D = 0$ ,  $C/N_0 = 47.7$  dB-Hz, and two different carrier phase reference epochs.

### 5.2. Computational Delay

The computation of the discriminator and the loop filter consumes a certain amount of time which is defined as the computational delay as described earlier. Proper buffering of the incoming signal samples can provide the necessary delay to allow an NCO update for the next integration interval. If buffering is not possible, the NCO update must eventually be delayed by one integration period (i.e.,  $t_D = T$ ). The effect of the computational delay is shown in Figure 12 for the second-order DPLL with SI/SI integration, using  $BT = 0.26$  which has nearly zero margins when  $t_D = T$  ( $BT_{osc} = 0.27$ ). As expected, the result of the NCO update delay case oscillates while the immediate update case does not. One can observe the actual degradation of stability by the computational delay. Furthermore, the result verifies that the obtained  $BT$  upper limit by root loci in Section 3 is correct.



**Figure 12.** Carrier phase tracking error for  $T = 20$  ms, a 13 Hz second-order DPLL with SI/SI numerical integration,  $C/N_0 = 47.7$  dB-Hz, and immediate NCO update ( $t_D = 0$ ) and an NCO update delay of 20 ms ( $t_D = T$ ).

## 6. Conclusions

In this study, the upper and lower limits of the  $BT$  for the DPLL were deduced theoretically and presented. For the upper limit, the effects of using digital integration methods on the stability of the DPLLs were investigated using the root loci. Stability problems in sampled-data loops occurred when  $BT$  was not sufficiently small, such that the sampled-data loops no longer represented their counterparts in the continuous-time domain, which typically occurs in modern digital GNSS receivers. The computational delay inherent in the discriminator and loop filter was considered in the analysis. All the types of DPLLs become conditionally stable when the unit delay is inserted as the computational delay, while the types vary for the order and integration method when zero computational delays are assumed.

The lower limits were obtained by analyzing the DPLL measurement error and the corresponding threshold. The thermal noise, oscillator phase noise using Allan deviation, and dynamic stress error were taken into account to constitute the measurement errors. The  $C/N_0$  threshold was defined as the  $C/N_0$  at the crossing point between the measurement error and threshold. By observing the  $C/N_0$  thresholds for varying  $B$ , the point that the  $C/N_0$  threshold increases rapidly existed, which is the lower limit. The lower limits are affected by the oscillator quality and dynamic stress because some amount of gain is required for the DPLL to track quickly varying phase and/or phase rate stably.

Issues related to the actual implementation of the DPLL were suggested with the simulation results. As an example, simulation results using second-order DPLL with SI/SI integration were presented. The phase error oscillates as  $BT$  approaches the deduced upper limits (for both cases with and without the computational delay), which verifies the previous analysis since it behaves as expected. For the numerical analysis of the lower limits, an accurate modeling of the oscillator phase noise on the sampled signal is needed and the dynamic stress, which has a dominant role in determining the lower limits, does not have random characteristics, so it is hard to observe the effect of it using the Monte Carlo simulation. Therefore, such detailed simulation and verification have remained as future works.

**Author Contributions:** Conceptualization, Y.-J.S., T.P. and J.-H.W.; methodology, Y.-J.S., T.P. and J.-H.W.; software, Y.-J.S. and J.-H.W.; validation, Y.-J.S., T.P. and J.-H.W.; formal analysis, Y.-J.S. and J.-H.W.; investigation, Y.-J.S., T.P. and J.-H.W.; resources, Y.-J.S., T.P. and J.-H.W.; data curation, Y.-J.S. and J.-H.W.; writing—original draft preparation, Y.-J.S. and J.-H.W.; writing—review and editing, T.P. and J.-H.W.; visualization, Y.-J.S., T.P. and J.-H.W.; supervision, J.-H.W.; project administration, J.-H.W.; funding acquisition, J.-H.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2020R1A2C2013091).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

A simple integrator in the continuous-time domain and its corresponding numerical implementations using SI, II, and BL are given by:

$$y(t) = \int_{t=0}^T x(t)dt \quad (\text{A1})$$

and:

$$\begin{aligned} y_{\text{SI}}(k) &= y(k-1) + T \cdot x(k-1) \\ y_{\text{II}}(k) &= y(k-1) + T \cdot x(k) \\ y_{\text{BL}}(k) &= y(k-1) + \frac{T}{2} [x(k) + x(k-1)] \end{aligned} \quad (\text{A2})$$

where  $x$  and  $y$  are the input and output of the integrator, respectively, as in Table 2, and the subscripts represent each numerical integration method.

Let us consider a second-order PLL, for example. Figure A1 shows the block diagram of the linearized model of the second-order PLL in the continuous-time domain, where the analog integrator is represented by  $1/s$  [2]. In the figure,  $\varepsilon(t)$  represents the discriminator output at  $t$ , which is obtained by subtracting the phase estimate at the previous loop update epoch,  $\hat{\phi}^-(t)$ , from the phase measurement at  $t$ ,  $\phi(t)$ . Similarly, the discriminator output is calculated in the discrete-time domain as follows:

$$\varepsilon(k) = \phi(k) - \hat{\phi}^-(k) = \phi(k) - \hat{\phi}(k-1) \quad (\text{A3})$$

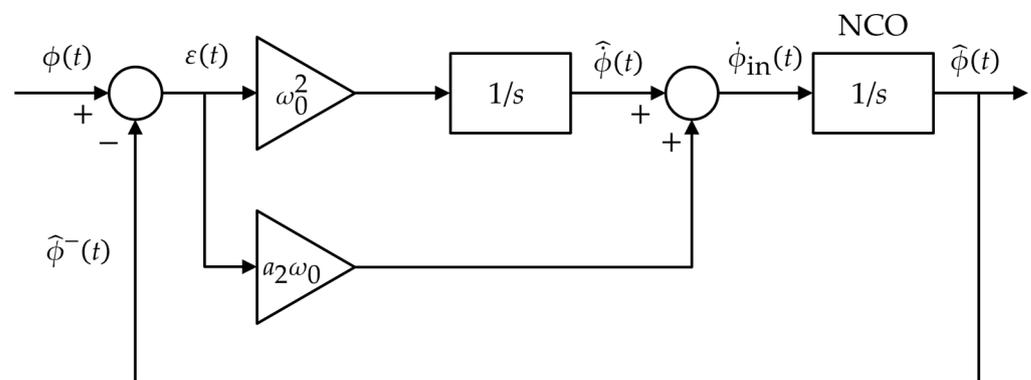


Figure A1. Second-order PLL.

The discriminator output is multiplied by the multiplier coefficients, which are based on the natural frequency of the system (i.e.,  $\omega_0$ ). The output is then processed as shown in the figure.

The digital implementation of the integrator in  $F(z)$  is obtained by applying (A2) to  $\hat{\phi}$ :

$$\begin{aligned} \hat{\phi}_{\text{SI}}(k) &= \hat{\phi}(k-1) + T \cdot \omega_0^2 \varepsilon(k-1) \\ \hat{\phi}_{\text{II}}(k) &= \hat{\phi}(k-1) + T \cdot \omega_0^2 \varepsilon(k) \\ \hat{\phi}_{\text{BL}}(k) &= \hat{\phi}(k-1) + \frac{T}{2} \cdot \omega_0^2 [\varepsilon(k) + \varepsilon(k-1)] \end{aligned} \quad (\text{A4})$$

The control signal of the NCO, which is the estimate of the phase rate, is obtained by the summation of  $\hat{\phi}(k)$  and the output from the component with  $a_2\omega_0$ :

$$\dot{\phi}_{\text{in}}(k) = \hat{\phi}(k) + a_2\omega_0\varepsilon(k) \quad (\text{A5})$$

where  $a_2$  is the filter coefficient of the second-order PLL.

The digital implementation of  $N(z)$  is obtained by applying (A2) and (A5) to  $\hat{\phi}$ :

$$\begin{aligned}
 \hat{\phi}_{\text{SI}}(k) &= \hat{\phi}(k-1) + T \cdot \dot{\hat{\phi}}_{\text{in}}(k-1) \\
 &= \hat{\phi}(k-1) + T \cdot \hat{\phi}(k-1) + T \cdot a_2 \omega_0 \varepsilon(k-1) \\
 \hat{\phi}_{\text{II}}(k) &= \hat{\phi}(k-1) + T \cdot \dot{\hat{\phi}}_{\text{in}}(k) \\
 &= \hat{\phi}(k-1) + T \cdot \hat{\phi}(k) + T \cdot a_2 \omega_0 \varepsilon(k) \\
 \hat{\phi}_{\text{BL}}(k) &= \hat{\phi}(k-1) + \frac{T}{2} [\dot{\hat{\phi}}_{\text{in}}(k) + \dot{\hat{\phi}}_{\text{in}}(k-1)] \\
 &= \hat{\phi}(k-1) + \frac{T}{2} [\hat{\phi}(k) + \hat{\phi}(k-1)] + \frac{T}{2} \cdot a_2 \omega_0 [\varepsilon(k) + \varepsilon(k-1)]
 \end{aligned} \tag{A6}$$

Similarly, the digital implementation of  $F(z)$  and  $N(z)$  for the first-/third-order cases can be obtained. Finally, the digital implementation of the overall DPLL for the combinations of the three integration methods is listed in Table A1.

**Table A1.** DPLL implementations for three numerical integration methods.

Filter Order	Type	Type	DPLL Implementations
First	$N(z)$	SI	$\hat{\phi}_{\text{SI}}(k) = \hat{\phi}(k-1) + T \cdot \omega_0 \varepsilon(k-1)$
		II	$\hat{\phi}_{\text{II}}(k) = \hat{\phi}(k-1) + T \cdot \omega_0 \varepsilon(k)$
		BL	$\hat{\phi}_{\text{BL}}(k) = \hat{\phi}(k-1) + \frac{T}{2} \cdot \omega_0 [\varepsilon(k) + \varepsilon(k-1)]$
Second	$F(z)$	SI	$\hat{\phi}_{\text{SI}}(k) = \hat{\phi}(k-1) + T \cdot \omega_0^2 \varepsilon(k-1)$
		II	$\hat{\phi}_{\text{II}}(k) = \hat{\phi}(k-1) + T \cdot \omega_0^2 \varepsilon(k)$
		BL	$\hat{\phi}_{\text{BL}}(k) = \hat{\phi}(k-1) + \frac{T}{2} \cdot \omega_0^2 [\varepsilon(k) + \varepsilon(k-1)]$
Third	$N(z)$	SI	$\hat{\phi}_{\text{SI}}(k) = \hat{\phi}(k-1) + T \cdot \hat{\phi}(k-1) + T \cdot a_2 \omega_0 \varepsilon(k-1)$
		II	$\hat{\phi}_{\text{II}}(k) = \hat{\phi}(k-1) + T \cdot \hat{\phi}(k) + T \cdot a_2 \omega_0 \varepsilon(k)$
		BL	$\hat{\phi}_{\text{BL}}(k) = \hat{\phi}(k-1) + \frac{T}{2} [\hat{\phi}(k) + \hat{\phi}(k-1)] + \frac{T}{2} \cdot a_2 \omega_0 [\varepsilon(k) + \varepsilon(k-1)]$
Third	$F(z)$	SI	$\hat{\phi}_{\text{SI}}(k) = \hat{\phi}(k-1) + T \cdot \omega_0^3 \varepsilon(k-1) \hat{\phi}_{\text{SI}}(k) =$ $\hat{\phi}(k-1) + T \cdot \hat{\phi}_{\text{SI}}(k-1) + T \cdot a_3 \omega_0^2 \varepsilon(k-1)$
		II	$\hat{\phi}_{\text{II}}(k) = \hat{\phi}(k-1) + T \cdot \omega_0^3 \varepsilon(k) \hat{\phi}_{\text{II}}(k) = \hat{\phi}(k-1) + T \cdot \hat{\phi}_{\text{II}}(k) + T \cdot a_3 \omega_0^2 \varepsilon(k)$
		BL	$\hat{\phi}_{\text{BL}}(k) = \hat{\phi}(k-1) + \frac{T}{2} \cdot \omega_0^3 [\varepsilon(k) + \varepsilon(k-1)] \hat{\phi}_{\text{BL}}(k) =$ $\hat{\phi}(k-1) + \frac{T}{2} [\hat{\phi}_{\text{BL}}(k) + \hat{\phi}_{\text{BL}}(k-1)] + \frac{T}{2} \cdot a_3 \omega_0^2 [\varepsilon(k) + \varepsilon(k-1)]$
Third	$N(z)$	SI	$\hat{\phi}_{\text{SI}}(k) = \hat{\phi}(k-1) + T \cdot \hat{\phi}(k-1) + T \cdot b_3 \omega_0 \varepsilon(k-1)$
		II	$\hat{\phi}_{\text{II}}(k) = \hat{\phi}(k-1) + T \cdot \hat{\phi}(k) + T \cdot b_3 \omega_0 \varepsilon(k)$
		BL	$\hat{\phi}_{\text{BL}}(k) = \hat{\phi}(k-1) + \frac{T}{2} [\hat{\phi}(k) + \hat{\phi}(k-1)] + \frac{T}{2} \cdot b_3 \omega_0 [\varepsilon(k) + \varepsilon(k-1)]$

$a_i, b_i$ : filter coefficients for the  $i$ -th order PLL.

## References

- Jaffe, R.; Rehtin, E. Design and Performance of Phase-Lock Circuits Capable of Near-Optimum Performance Over a Wide Range of Input Signal and Noise Levels. *IRE Trans. Inf. Theory* **1955**, *1*, 66–76. [\[CrossRef\]](#)
- Kaplan, E.D.; Hegarty, C. *Understanding GPS Principles and Applications*, 2nd ed.; Artech House: Norwood, MA, USA, 1996.
- Gupta, S.C. On Optimum Digital Phase-Locked Loops. *IEEE Trans. Commun. Technol.* **1968**, *16*, 340–344. [\[CrossRef\]](#)
- Cahn, C.; Leimer, D.K.; Marsh, C.L.; Huntowski, F.; Larue, G.D. Software Implementation of a PN Spread Spectrum Receiver to Accommodate Dynamics. *IEEE Trans. Commun.* **1977**, *25*, 832–840. [\[CrossRef\]](#)
- Lindsey, W.C.; Chie, C.M. A Survey of Digital Phase-Locked-Loops. *Proc. IEEE* **1981**, *69*, 410–431. [\[CrossRef\]](#)
- Broumandan, A.; Nielsen, J.; Lachapelle, G. Coherent Integration Time Limit of a Mobile Receiver for Indoor GNSS Applications. *GPS Solut.* **2012**, *16*, 157–167. [\[CrossRef\]](#)
- Pany, T.; Riedl, B.; Winkel, J.; Wörz, T.; Schweikert, R.; Niedermeier, H.; Lagrasta, S.; López-Risueno, G.; Jiménez-Banos, D. Coherent Integration Time: The Longer, The Better. *Inside GNSS* **2009**, *4*, 52–61.

8. Watson, R.; Lachapelle, G.; Klukas, R. Testing Oscillator Stability as a Limiting Factor in Extreme High-Sensitivity GPS Applications. In Proceedings of the European Navigation Conference, Manchester, UK, 8–10 May 2006.
9. O'Driscoll, C.; Petovello, M.; Lachapelle, G. Choosing the Coherent Integration Time for Kalman Filter-based Carrier Phase Tracking of GNSS Signals. *GPS Solut.* **2011**, *15*, 345–356. [[CrossRef](#)]
10. Kuo, B.C. *Automatic Control Systems*, 6th ed.; Prentice-Hall: Englewood Cliffs, NJ, USA, 1991.
11. Stephens, S.A.; Thomas, J.B. Controlled-root formulation for digital phase-locked loops. *IEEE Trans. Aerosp. Electron. Syst.* **1995**, *31*, 78–95. [[CrossRef](#)]
12. Kazemi, P.L. Optimum Digital Filters for GNSS Tracking Loops. In Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008), Savannah, CA, USA, 16–19 September 2008; pp. 2304–2313.
13. Franklin, G.F.; Powell, J.D.; Workman, M.L. *Digital Control of Dynamic Systems*, 2nd ed.; Addison-Wesley Publishing Company: Reading, MA, USA, 1990.
14. Ward, P.W.; Fuchser, T.D. Stability criteria for GNSS receiver tracking loops. *Navig. J. Inst. Navig.* **2014**, *61*, 293–309. [[CrossRef](#)]
15. Irsigler, M.; Eissfeller, B. PLL Tracking Performance in the Presence of Oscillator Phase Noise. *GPS Solut.* **2002**, *5*, 45–57. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.