THE JOURNAL OF NAVIGATION (2018), **71**, 257–280. © The Royal Institute of Navigation 2017
doi:10.1017/S0373463317000741

Generalised Theory on the Effects of Sampling Frequency on GNSS Code Tracking

Vinh T. Tran[1], Nagaraj C. Shivaramaiah[2], Thuan D. Nguyen[3], Joon W. Cheong[1], Eamonn P. Glennon[1] and Andrew G. Dempster[1]

[1](*Australian Centre for Space Engineering Research, UNSW, Australia*)
[2](*University of Colorado at Boulder, USA*)
[3](*NAVIS centre, Hanoi University of Science and Technology, Viet Nam*)
(E-mail: vinh.tran@student.unsw.edu.au)

Synchronisation of the received Pseudorandom (PRN) code and its locally generated replica is fundamental when estimating user position in Global Navigation Satellite System (GNSS) receivers. It has been observed through experiments that user position accuracy decreases if sampling frequency is an integer multiple of the nominal code rate. This paper provides an accuracy analysis based on the number of samples and the residual code phase of each code chip. The outcomes reveal that the distribution of residual code phases in the code phase range $[0, 1/n_s)$, where n_s is the number of samples per code chip, is the root cause of accuracy degradation, rather than the ratio between sampling frequency and nominal code rate. Doppler frequencies, coherent integration periods, front-end filter bandwidths and received Carrier to Noise ratios (C/N_0) also influence receiver accuracy. Also provided are a sampling frequency selection guideline and new proposed estimates of the correlation output and the Delay Locked Loop (DLL) tracking error, which can be applied to precisely model GNSS receiver baseband signal processing.

KEYWORDS

1. Sampling frequency. 2. DLL tracking error. 3. Residual code phase.

Submitted: 3 March 2017. Accepted: 3 September 2017. First published online: 28 November 2017.

1. INTRODUCTION. Global Navigation Satellite System (GNSS) receiver technology has changed dramatically since the first reception of Global Positioning System (GPS) signals, from complex and partly analogue circuits to sophisticated, small, multichannel digital receivers or fully software-defined radio architectures (Parkinson, 1996; Braasch and Van Dierendonck, 1999; Kaplan and Hegarty, 2005; Tran et al., 2017). However, navigation still relies on estimating the range from the user to at least four satellites and using multilateration. This range measurement is accomplished by synchronising the received Pseudorandom (PRN) code and a locally generated replica to estimate satellite time of transmission, which when subtracted from local time of reception and multiplied by the

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

speed of light gives the so-called pseudorange (Parkinson, 1996). Code synchronisation is performed by a Delay Locked Loop (DLL), while a Phase Lock Loop (PLL) tracks the carrier frequency and phase. The better the code alignment, the more accurate the user position.

If the sampling frequency is an integer multiple of the nominal code rate, it is the so-called commensurate sampling frequency, otherwise it is the so-called non-commensurate sampling frequency. Effects of sampling frequency on GNSS receiver code tracking were first mentioned in Thomas (1988) and detailed in Akos and Pini (2006) and Pini (2007). If the sampling frequency is commensurate, the correlation shape is distorted and receiver accuracy is degraded. However, this was only observed at four different sampling frequencies, without theoretical analysis. The conclusion, that degradation occurs when samples per chip is an integer and is eliminated when the sampling ratio is irrational, does not cover all situations in practice.

This paper provides a new approach for estimating the correlation output and DLL tracking error from a hardware receiver perspective that is based on the following key parameters: number of samples and residual code phase of each code chip, sampling frequency, Doppler frequency and coherent integration period.

The impact of the sampling frequency on the correlation output was only considered versus sampling filter bandwidth and assumed infinite samples per chip sampling (Akos and Braasch, 1996; Kaplan and Hegarty, 2005; Jang et al., 2012; Kou and Morton, 2013; Betz, 2015). The case of finite samples per chip sampling was mentioned in Akos and Pini (2006) and Hegarty (2011), but detailed theoretical analysis was not provided. It is accurately and theoretically calculated here, based on the exact number of samples per code chip. The coherent Early-Minus-Late (EML) DLL tracking error is also re-analysed to indicate clearly effects of sampling rates versus Doppler shifts, front-end filter bandwidths, and received Carrier to Noise ratios (C/N_0).

The outcomes reveal the distribution of residual code phases to be the root cause of correlator and DLL discriminator distortions, instead of the ratio between the sampling frequency and the nominal code rate. Accuracy degradation can occur when the sampling frequency is both commensurate and non-commensurate. Although only the GPS L1 Coarse Acquisition (C/A) code signal is analysed here, the results can be applied to other GNSS signals and software defined radio receivers.

This paper is organised as follows. Section 2 provides an overview of a digital GNSS receiver, mathematical models used to calculate residual code phase and correlation output. Section 3 shows the effect of sampling frequency and Doppler on the correlation and discriminator functions and provides a guideline for sampling frequency selection. DLL tracking error analyses are in Section 4 and some concluding remarks follow in Section 5.

2. DIGITAL GNSS RECEIVER OVERVIEW.

Figure 1 shows a GNSS receiver block diagram, where locally generated signals track the incoming signal by adjusting code and carrier Numerically Controlled Oscillators (NCOs). The performance of the receiver depends on the accuracy of this synchronisation.

The input to the digital part of the receiver is generally an Intermediate Frequency (IF) signal that is obtained by down-converting a Radio-Frequency (RF) signal. The front-end filter is approximated by an ideal pre-correlation filter (a rectangular Bandpass Filter (BPF)) centred at the IF frequency (f_{IF}) with two-sided bandwidth β_r. One satellite's IF signal can

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

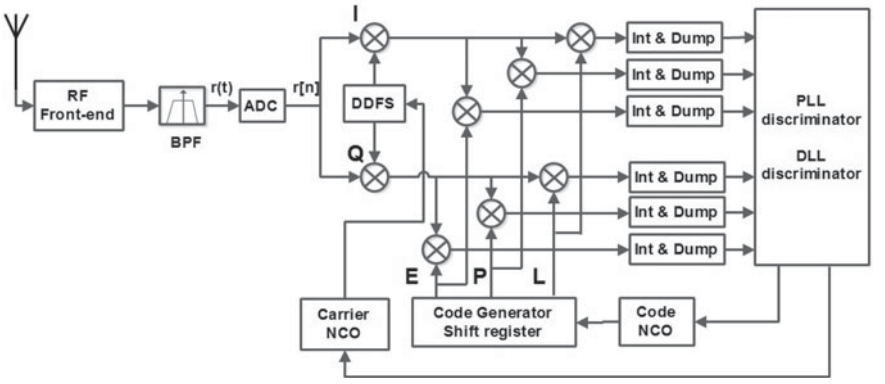


Figure 1. Block diagram of a digital GNSS receiver.

be written as:

$$r(t) = \sqrt{2C_s} \sum_{k=-\infty}^{\infty} C_{\{k\}_N} \Pi(t - kT_c(k) - \tau) d(t - \tau) cos(2\pi(f_{IF} + f_D)t + \theta) + w(t) \quad (1)$$

where C_s is the received signal power; Π is the rectangular function; $C_k = \pm 1$ is the k^{th} PRN code, $\{k\}_N$ is k modulus N, and N is the PRN code length (N = 1023 for GPS L1 C/A); $T_c(k)$ is the code chip duration that is varied by time due to the effect of the code Doppler or local oscillator drifts; $d(t - \tau) = \pm 1$ is the modulated navigation data bit; τ is the timing delay of received signal; f_{IF} and f_D denotes the IF frequency and the Doppler shift, respectively; θ is the random unknown carrier phase and $w(t)$ is the Additive White Gaussian Noise (AWGN) with zero-mean and Power Spectral Density (PSD) N_0.

The IF signal is sampled by an Analogue-to-Digital Converter (ADC) which generates a sequence of samples $r[nT_s]$, obtained by sampling $r(t)$ at the sampling frequency $f_s = 1/T_s$. The notation $r[n] = r[nT_s]$ indicates a discrete time sequence of the received signal $r(t)$:

$$r[n] = \sqrt{2C_s} \sum_{k=-\infty}^{\infty} \sum_{m=0}^{M-1} C_{\{k\}_N} \delta \left[n - m - \sum_{j=-\infty}^{j=k-1} M_j - \lfloor \tau \rfloor_{T_s} \right] d[n - \lfloor \tau \rfloor_{T_s}]$$
$$\times cos[2\pi(f_{IF} + f_D)n + \hat{\theta}] + w[n] \quad (2)$$

where δ is the Dirac delta function and $\lfloor \tau \rfloor_{T_s} = floor(\tau / T_s)$ is the delay of the received signal after sampling. M_j is the number of signal samples of the j th PRN code chip ($M_k T_s \leq T_c(j)$). M_j is dependent on the f_s/f_c ratio and the residual code phase of that code chip, with f_c being the local generated PRN code rate. $\hat{\theta}$ is the discrete time of the carrier phase. This phase discreteness is not considered as it does not affect the discussion topic. After Equation (2), $\hat{\theta}$ is replaced by θ here.

The PLL wipes off the carrier (Kaplan and Hegarty, 2005) and the DLL synchronises the local and incoming PRN codes. The signal after the Direct Digital Frequency Synthesizer (DDFS) is down-converted to baseband and is ideally contained in only the in-phase (I) channel. The DLL tracks the time delay of the incoming PRN. The baseband signal is correlated with three local replica code-taps: Early (E), Prompt (P), and Late (L), through a

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

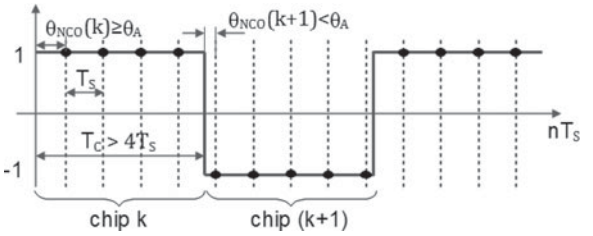


Figure 2. Residual code phases versus the number of samples per code chip with $4f_c < f_s < 5f_c$.

multiplication and integration, usually over an integer PRN code period (T_0). Discriminator feedbacks adjust the Code NCO, which fluctuates the local replica code rate to synchronise with the incoming code (Parkinson, 1996).

2.1. *Residual code phase estimation.* Estimating the pseudorange, used to calculate position, relies on the ability to locate the PRN code chip transition. The more precisely the transition is detected, the more accurate the user position. The residual code phase ($\boldsymbol{\theta_{NCO}}$) is exploited to improve the accuracy.

Let $\eta(m)$ denote the code NCO register contents at the mth clock tick, the NCO difference equation is:

$$\eta(m) = \{\eta(m-1) + f_c/f_s\}_1 \tag{3}$$

The register overflows once every $n_s = f_c/f_s$ samples on average, causing the PRN code generator to be clocked, incrementing the $(k-1)$th PRN code C_{k-1} to the kth PRN code C_k. The NCO remaining content after this overflow is the residual code phase ($\theta_{NCO}(k)$). It is equivalent to the delay between the start of the local kth chip and the first equivalent incoming signal sample or the so-called instantaneous chip phase, with $0 \le \theta_{NCO}(k) < 1/n_s$.

The number of samples per code chip, n_s, is generally a non-integer. Even if designed to be an integer, drifts in the local oscillator and Doppler shifts cause it to become a non-integer. The number of received signal samples correlated with the kth PRN code is either $\lfloor n_s \rfloor$ or $\lfloor n_s \rfloor + 1$, where the exact number is determined by a given $\theta_{NCO}(k)$ as illustrated in Figure 2 and:

$$M_k = \begin{cases} \lfloor n_s \rfloor + 1 & (0 \le \theta_{NCO}(k) < \theta_A) \\ \lfloor n_s \rfloor & (0 \le \theta_{NCO}(k) < 1/n_s) \end{cases} \tag{4}$$

where $\theta_A = 1 - \dfrac{\lfloor n_s \rfloor}{n_s}$ and $\lfloor . \rfloor$ is the floor function. The residual code phase $\theta_{NCO}(k)$ of each PRN code chip can be calculated as:

$$\theta_{NCO}(k) = \begin{cases} \theta_{NCO}(k-1) + \dfrac{\lfloor n_s \rfloor + 1}{n_s} - 1 & (0 \le \theta_{NCO}(k-1) < \theta_A) \\ \theta_{NCO}(k-1) + \dfrac{\lfloor n_s \rfloor}{n_s} - 1 & (\theta_A \le \theta_{NCO}(k-1) < 1/n_s) \end{cases} \tag{5}$$

where k is in the range $[0, T_0 N - 1]$.

2.2. *Correlation output calculation.* Assume that the PLL is locked and the IF signal is ideally down-converted to baseband. If the incoming baseband signal is correlated

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

with the local replica code for T seconds, or T0 code periods, and the navigation data bits $\mathbf{d}[\mathbf{n} - \lfloor \boldsymbol{\tau} \rfloor_{\mathbf{T_s}}]$ do not change, then the P tap correlation output is given by:

$$\begin{aligned} R(\tau) &= \frac{C_s}{N_L} \sum_{n=0}^{N_L-1} \sum_{k=0}^{T_0N-1} \sum_{m=0}^{M_k-1} C_{\{k\}_N} \delta \left[n - m - \sum_{j=0}^{j=k} M_j - \lfloor \tau \rfloor_{T_s} \right] \\ &\quad \times \sum_{k=0}^{T_0N-1} \sum_{m=0}^{M_k-1} C_{\{k\}_N} \delta \left[n - m - \sum_{j=0}^{j=k} M_j + \lfloor \theta_{NCO}(k) T_c \rfloor_{T_s} \right] \\ &= \frac{C_s}{N_L} \sum_{n=0}^{N_L-1} \sum_{k=0}^{T_0N-1} \sum_{m=0}^{M_k-1} C_{\{k\}_N} \delta \left[n - m - \sum_{j=0}^{j=k} M_j - \lfloor \tau + \theta_{NCO}(k) T_c \rfloor_{T_s} \right] \\ &\quad \times \sum_{k=0}^{T_0N-1} \sum_{m=0}^{M_k-1} C_{\{k\}_N} \delta \left[n - m - \sum_{j=0}^{j=k} M_j \right] \end{aligned} \tag{6}$$

where T_c is the nominal code chip duration and $N_L = \sum_{k=0}^{T_0N-1} : M_k$ is the total number of samples that are correlated and M_k is the number of samples of the kth PRN code chip. The correlation output, as shown in Appendix A, is:

$$R(\tau) = \frac{C_s}{N_L} \sum_{k=0}^{T_0N-1} \int_{-\beta_r/2}^{\beta_r/2} T_s G_{s_k}(f) e^{i2\pi f T_s \left(\lfloor \tau + \theta_{NCO}(k) T_c \rfloor_{T_s} \right)} df \tag{7}$$

where $-T_c \leq (\tau + \theta_{NCO}(k) T_c) \leq T_c$ and $G_{s_k}(f) = T_s \left(\frac{\sin(\pi f M_k T_s)}{\sin(\pi f T_s)} \right)^2$.

The PSD normalised to unit power of the sampling PRN code is:

$$G_s(f)_{PSD} = \frac{T_s}{N_L} \sum_{k=0}^{T_0N-1} \left(\frac{\sin(\pi f M_k T_s)}{\sin(\pi f T_s)} \right)^2 e^{i2\pi f T_s \left(\lfloor \theta_{NCO}(k) T_c \rfloor_{T_s} \right)} \tag{8}$$

It is obvious that $\theta_{NCO}(k) T_c < T_s$ thus $\lfloor \theta_{NCO}(k) T_c \rfloor_{T_s} = 0$ and

$$G_s(f)_{PSD} = \frac{M_k}{N_L} \sum_{k=0}^{T_0N-1} \frac{M_k T_s \sin^2(\pi f M_k T_s)}{M_k^2 \sin^2(\pi f T_s)} \tag{9}$$

observe that $\sum_{k=0}^{T_0N-1} : M_k = N_L$, $\lim_{f_s \to \infty} M_k T_s = T_c$, and $\lim_{f_s \to \infty} \sin(\pi f T_s) = \pi f T_s$ so:

$$\begin{aligned} \lim_{f_s \to \infty} G_s(f)_{PSD} &= \frac{1}{N_L} \sum_{k=0}^{T_0N-1} M_k \lim_{f_s \to \infty} \left(\frac{M_k T_s \sin^2(\pi f M_k T_s)}{M_k^2 \sin^2(\pi f T_s)} \right) \\ &= T_c sin\, c^2(\pi f T_c) \end{aligned} \tag{10}$$

this is the well-known PSD proposed in (Parkinson, 1996). This widely applied PSD equation is only correct when the sampling frequency is much higher than the PRN code rate.

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

3. EFFECT OF SAMPLING FREQUENCY ON CORRELATION SHAPE AND DLL DISCRIMINATOR FUNCTION. Figure 3 shows correlator shapes and EML discriminator outputs for 1 ms integrations of the incoming GPS L1 C/A signal with the P tap. Signals were received by a Nuand bladeRF front-end (Nuand-LLC, 2016) with two sampling frequencies f_s = 4·092 and 4·093 MHz. Different colour results are illustrated for random instances of 1ms integrations. Step-like distortion of both functions can be easily observed for a commensurate sampling frequency (f_s = 4·092 MHz). Oscillator drift and Doppler shifts can cause the sampling ratio to become non-integer, but not sufficiently to reduce this distortion. Distortions are absent for f_s = 4·093 MHz, a 1 kHz difference, suggesting non-commensurate sampling frequencies do not suffer the distortions. However, this is not generally correct. Figure 4 shows correlator shapes obtained from Equation (7) at several sampling rates and random values of $\boldsymbol{\theta_{NCO}(0)}$. The distortion still appears when the ratio f_s/f_c is non-commensurate.

3.1. *Effect of the sampling frequency and the integration period selection.* If Doppler shifts are zero and the sampling frequency is four-times the nominal PRN code rate, each analogue code chip always has four samples. Figure 5 shows that the same sample sequence can represent two different incoming code phases. The same correlation value is expected, despite real analogue signal alignments being different. This leads to the "staircase" correlation shapes.

Higher residual code phase density with uniform distribution in the range $[0, 1/n_s)$, gives a less distorted correlation shape, as shown in Figure 4. When f_s = 4·093, the residual code phases have 1023 different values across $[0, 1/n_s)$, with just one, two or four values for f_s = 4·092 MHz (f_s/f_c = 4), f_s = 4·6035 MHz (f_s/f_c = 4·5), or f_s = 4·3477 MHz (f_s/f_c = 4·25), respectively.

An ideal sampling frequency, therefore, must be based on the distribution of distinct residual code phases. If $f_s = M_0 f_c + \delta_{f_s}$ where $0 < \delta_{f_s} \ll f_c$, M_o is an integer, integration period is T ms (T_0 code periods), and $\theta_{NCO}(0) = 0$, from Equation (5) the number of different code phases, N_{CP}, can be estimated as:

$$N_{CP} = \frac{1 - \frac{\lfloor n_s \rfloor}{n_s}}{\frac{\lfloor n_s \rfloor + 1}{n_s} - 1} + \frac{\frac{1}{n_s} - 1 - \frac{\lfloor n_s \rfloor}{n_s}}{1 - \frac{\lfloor n_s \rfloor}{n_s}} = \frac{\delta_{f_s}}{f_c - \delta_{f_s}} + \frac{f_c - \delta_{f_s}}{\delta_{f_s}} \tag{11}$$

because $0 < \delta_{f_s} \ll f_c$, Equation (11) can be approximated as: $N_{CP} \approx \frac{f_c}{\delta_{f_s}}$.

The number of correlated code chips ($T_0 N$) is also the maximum possible number of distinct residual code phases. If $\delta_{f_s} = \frac{f_c}{T_0 N}$, the effect of correlation distortion is minimised. For instance, if the received signal is GPS L1 C/A and T = 1 ms, so $\delta_{f_s} = \pm 1$ kHz and the sampling frequency at f_s = 4·093 MHz produces the smoothest result as shown in Figure 4.

This result can also be used to estimate the minimum integration period required, e.g. if f_s = 16·3676 MHz (NordNav Rxx2 receiver), the minimum required correlation time is $T \geq$ 2·5 ms. If T = 2 ms, the distortion is much higher - as presented in Figure 6. However, these estimates assume that $0 < \delta_{f_s} \ll f_c$. When δ_{f_s} is greater, because of the random distribution, the number of distinct residual code phases cannot be directly estimated by the sampling frequency. The one-sigma coherent EML DLL error is analysed to further examine the effect of the sampling frequency in Section 4.

3.2. *Effect on the presence of Doppler and Local Oscillator (LO) clock drift.* The effect of Doppler shifts and LO drifts on the incoming signal leads to offsets of the PRN

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

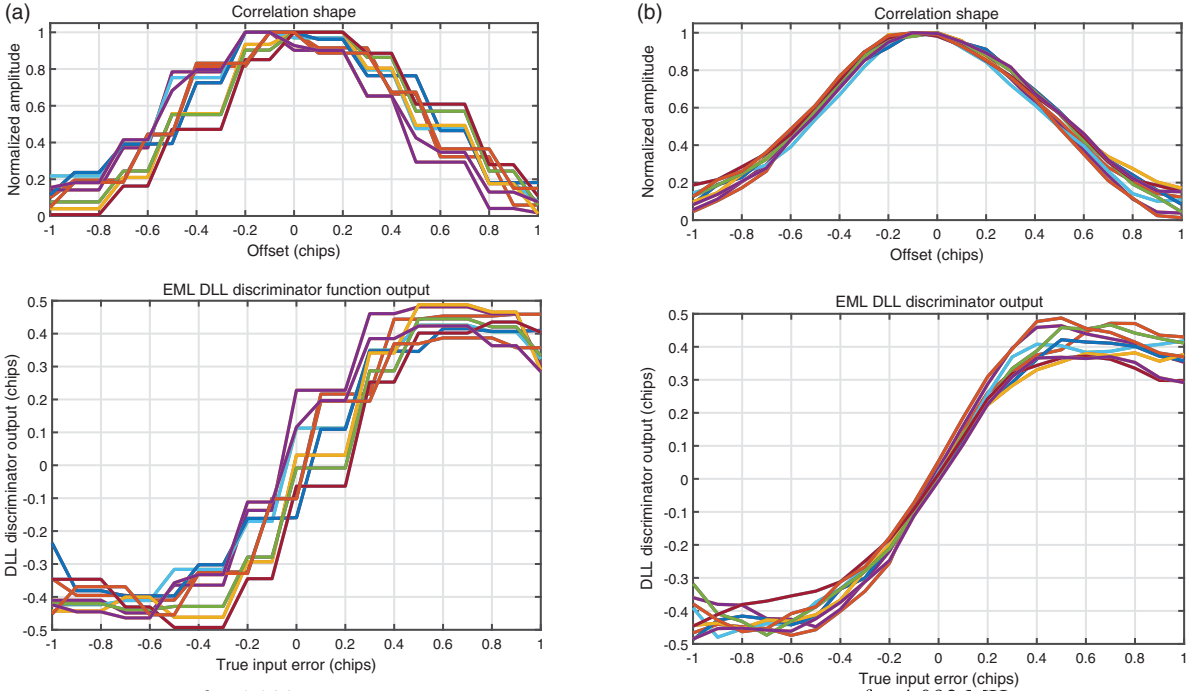


Figure 3. Normalised correlator and EML discriminator functions for different sampling frequencies. Results are obtained by correlating the incoming signal with various local generated replica signals that have the time delay from $-T_c$ to T_c with step = $10^{-2}T_c$.

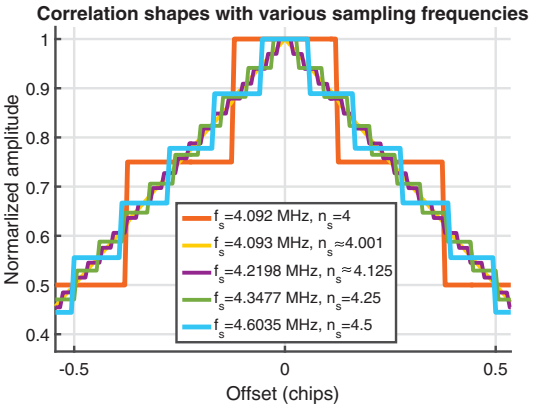


Figure 4. Correlation shapes for 1 ms integration with various sampling frequencies.

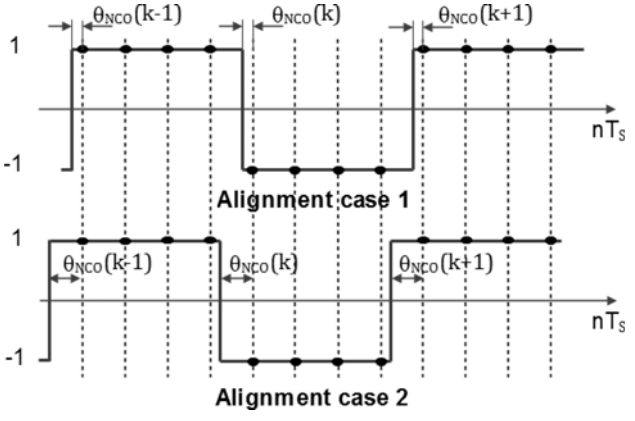


Figure 5. Ambiguous synchronisation between a local PRN code and two different incoming analogue signals of the same PRN sequence, but with slightly differing code phase offset.

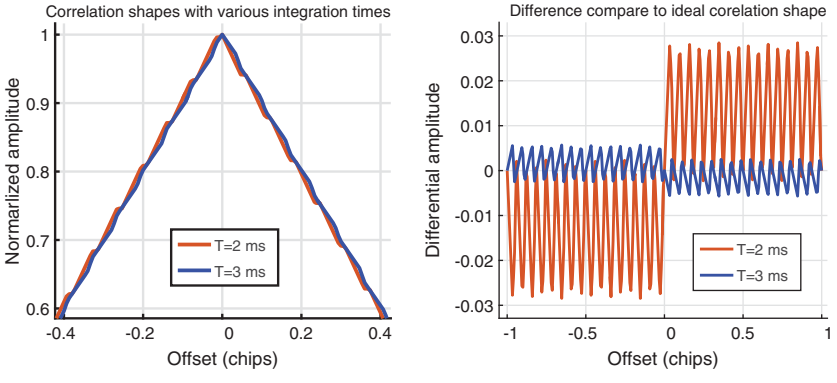


Figure 6. Correlation shapes and their errors with respect to the ideal correlation at a sampling frequency f_s = 16·3676 MHz using various coherent integration periods.

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

code rate. Consequently, the number of samples per code chip is not an integer, even if the sampling frequency is commensurate. The LO reference frequency source for a GNSS receiver is often a Temperature Compensated Crystal Oscillator (TCXO), which typically has a one second Allan deviation of 10^{-7} and frequency stability of less than one part per million (ppm). If sampling frequency is $f_s = 4{\cdot}092$ MHz, the code frequency drift is less than 4 Hz. This LO drift is very small when the Doppler shift may be inline or reverse to the LO drift. The LO drift thus can be considered as an addition to the Doppler shift. The Doppler shift is, therefore, only mentioned in the following analyses, but it actually stands for the combination of Doppler shift and LO drift.

If the Doppler frequency, f_D, is in the range $[-10, 10]$ kHz, the corresponding code frequency varies between $[-6{\cdot}5, 6{\cdot}5]$ Hz. The code Doppler frequency, δ_{f_c}, can be estimated by $\delta_{f_c} = \frac{f_D f_c}{f_{RF}}$, where f_{RF} is the carrier frequency. The code Doppler shift leads to variation of residual code phases, thus the correlator distortion will change as sampling rates drift into and away from commensurate values. The minimum Doppler frequency and coherent integration period, which flattens the correlation shape, can be evaluated by the number of distinct residual code phases. If the sampling frequency is commensurate, $f_s = M_0 f_c$, the real code rate is $\widehat{f_c} = \frac{f_c(f_{RF} - f_D)}{f_{RF}}$, with $f_D > 0$. The number of distinct code phases is:

$$N_{CP} = \frac{\frac{f_s}{\widehat{f_c}} - \frac{f_s}{f_c}}{\frac{f_s}{f_c} + 1 - \frac{f_s}{\widehat{f_c}}} + \frac{\frac{f_s}{f_c} + 1 - \frac{f_s}{\widehat{f_c}}}{\frac{f_s}{\widehat{f_c}} - \frac{f_s}{f_c}} \tag{12}$$

because $\left(\frac{f_s}{\widehat{f_c}} - \frac{f_s}{f_c}\right) \ll 1, N_{CP}$ can be approximated as:

$$N_{CP} \approx \frac{f_c \widehat{f_c}}{f_s\left(f_c - \widehat{f_c}\right)} = \frac{f_c\left(f_{RF} - f_D\right)}{f_s f_D} \tag{13}$$

The maximum number of distinct residual code phases is the number of correlated code chips. If $N_{CP} = T_o N$, the correlation shape is the smoothest. The relation between integration period T (equal to T_0 PRN code periods) and Doppler frequency (f_D), therefore, must fulfil:

$$f_D = \frac{f_c f_{RF}}{f_s T_0 N + f_c} \tag{14}$$

For example, if the received signal is GPS L1 C/A, $f_{RF} = 1575{\cdot}42$ MHz, $f_s = 4{\cdot}092$ MHz ($n_s = 4$), and integration period $T = 1$ and 10 ms, the Doppler frequency must be $f_D \geq 384{\cdot}91$ kHz and $38{\cdot}5$ kHz, respectively. This is out of most terrestrial receivers' Doppler range, $[-10, 10]$ kHz. If the Doppler is $f_D = 2$ kHz, the integration period must be $T \geq 192{\cdot}5$ ms. However, the coherent integration period should be smaller than 20 ms or 10 ms, with or without data bit synchronisation, respectively, or may be shorter due to secondary code transitions (Tran et al., 2015; Shivaramaiah et al., 2008). High Doppler frequencies, high LO drifts, and long integration periods such as tracking a data-less (Pilot) channel can help to reduce the correlation and discriminator distortions, but a typical receiver usually has a stable LO and a short integration period ($T \leq 20$ ms) for tracking, thus the effect of sampling frequency should be considered for every Doppler frequency. On the other hand, if the difference of the sampling frequency to an integer multiple of the nominal code rate, δ_{f_s}, is too small, a high Doppler frequency can lead to the distortion. For example, if

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

f_s = 4·0922 MHz (δ_{f_s} = 0·2 kHz), Doppler $f_D \approx 77$ kHz causes the number of samples per code chip to become $n_s = 4$ and the distortion thus occurs. This will be analysed in the next section.

4. COHERENT EARLY MINUS LATE DLL ERROR ANALYSIS. The selection guideline for the sampling frequency and integration period is only valid for $0 < \boldsymbol{\delta}_{f_s} \ll f_c$, as mentioned in Section 3.1. Therefore, the coherent EML DLL tracking error will be analysed to further examine effects of the sampling frequency, front-end bandwidth, coherent integration periods, and Doppler shifts in this section.

4.1. *Theoretical code tracking loop error estimate.* In this paper, the code tracking loop uses the model that was proposed by Betz and Kolodziejski (2000; 2009a; 2009b) which is re-illustrated in Figure 7. The Time Of Arrival (TOA) estimator uses an integration period of T seconds to produce an unsmoothed TOA estimate from the received signal plus noise. This estimate is used to update the previous TOA estimate. Using a code-tracking loop that acts like a TOA smoothing filter, producing smoothed TOA estimates.

According to Betz and Kolodziejski (2009a), the variance of the unsmoothed TOA estimate from the discriminator is σ_u^2, and the variance of the smoothed TOA estimate at the output of the code tracking loop, denoted as σ_s^2, is:

$$\sigma_s^2 \cong \sigma_u^2 2B_L T \tag{15}$$

where B_L is the noise bandwidth of the DLL tracking loop, T is the integration period, and $0 < B_L T \leq 0{\cdot}5$. It should be noted that a factor of (1–0·5B_LT) appears in Equation (2) of Betz and Kolodziejski (2009a). In fact, this term originates from a flaw from that work as detailed by Zhang and Zhan (2012), thus it has been omitted here.

This estimation does not accurately consider the effect of the sampling frequency. Therefore, it is re-considered here with the errors broken into two components, namely the part that is influenced by the sampling frequency (σ_1^2 and σ_2^2) and the part that is effected by the code lock loop noise filter (σ_{u1N}^2 and σ_{u2N}^2). A detailed analysis is shown in Appendix B.

The variance of the unsmoothed TOA estimate, σ_u^2, is bounded between σ_{u1}^2 and σ_{u2}^2, whose values are:

$$\sigma_{u1}^2 = \sigma_1^2 + \sigma_{u1N}^2 \tag{16}$$

$$\sigma_{u2}^2 = \sigma_2^2 + \sigma_{u2N}^2 \tag{17}$$

where:

$$\sigma_1^2 = \frac{\left(\sum_{k=0}^{T_0 N-1} \frac{1}{N_L} \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f) \sin(\pi f \Delta) \sin(\pi f \ (2\theta_{NCO}(k)\, T_c - T_s)) \cos(\pi f T_s)\, df\right)^2}{\left(\sum_{k=0}^{T_0 N-1} \frac{2\pi}{N_L} \int_{-\beta_r/2}^{\beta_r/2} f G_{s_k}(f) \sin(\pi f \Delta) \cos(2\pi f\, \theta_{NCO}(k)\, T_c)\, df\right)^2} \tag{18}$$

$$\sigma_2^2 = \frac{\left(\sum_{k=0}^{T_0 N-1} \frac{1}{N_L} \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f) \sin(\pi f \Delta) \sin(\pi f \ (2\theta_{NCO}(k) T_c - T_s)) \cos(\pi f T_s)\, df\right)^2}{\left(\sum_{k=0}^{T_0 N-1} \frac{2\pi}{N_L} \int_{-\beta_r/2}^{\beta_r/2} f G_{s_k}(f) \sin(\pi f \Delta) \cos(2\pi f\, (\theta_{NCO}(k)\, T_c - T_s))\, df\right)^2} \tag{19}$$

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

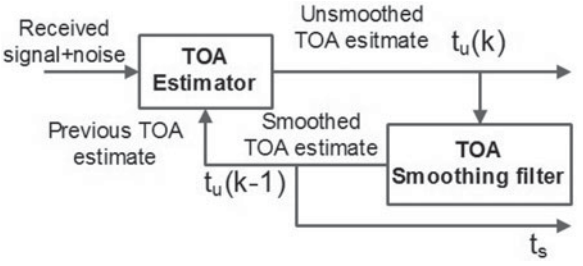


Figure 7. Representation of code tracking loop (Betz and Kolodziejski, 2009a).

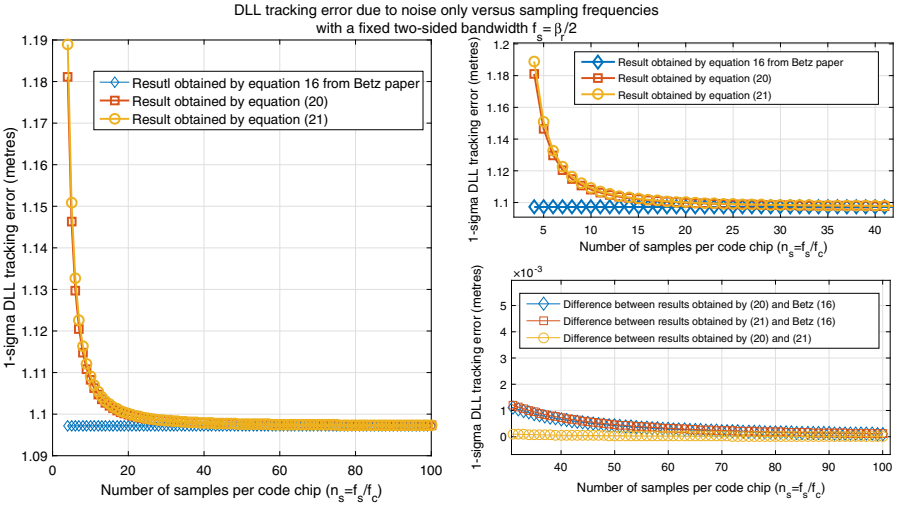


Figure 8. DLL jitter versus different sampling frequencies (step = f_c) for a GPS L1 C/A with C/N_0 = 40 dB-Hz, B_L = 0·5 Hz, T = 1 ms, and fixed BW $\beta_r = 2f_c$.

$$\sigma_{u1N}^2 = \frac{\frac{N_0}{2C_sN_L}\sum_{k=0}^{T_0N-1} - \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)\sin^2(\pi f\Delta)\,df}{\left(\sum_{k=0}^{T_0N-1}\frac{2\pi}{N_L}\int_{-\beta_r/2}^{\beta_r/2} fG_{s_k}(f)\sin(\pi f\Delta)\cos(2\pi f\,\theta_{NCO}(k)\,T_c)\,df\right)^2} \tag{20}$$

$$\sigma_{u2N}^2 = \frac{\frac{N_0}{2C_sN_L}\sum_{k=0}^{T_0N-1} - \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)\sin^2(\pi f\Delta)\,df}{\left(\sum_{k=0}^{T_0N-1}\frac{2\pi}{N_L}\int_{-\beta_r/2}^{\beta_r/2} fG_{s_k}(f)\sin(\pi f\Delta)\cos(2\pi f\,(\theta_{NCO}(k)\,T_c - T_s))\,df\right)^2} \tag{21}$$

It can be mathematically observed that when the sampling frequency is infinite, $\theta_{NCO}(k) \cong 0$ for all k and $T_s \cong 0$ the cosine value appearing in the denominator of Equations (20) and (21) is approximately one. This estimation becomes the same as the results presented in Equation (16) of Betz and Kolodziejski (2009a). The analysis comparison results presented in Figure 8 further strengthens this state.

Both σ_1^2 and σ_2^2, as shown in Equations (18) and (19), have approximately constant bias during tracking. The DLL noise filter has no effect on these values. Only the error contributed by noise σ_{u1N}^2 and σ_{u2N}^2 are filtered by the tracking loop filter. Consequently,

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

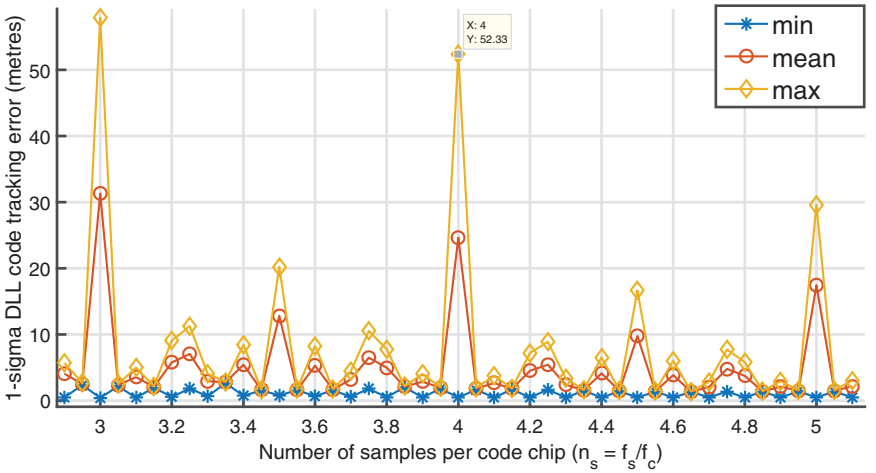


Figure 9. Upper bound and lower bound of the DLL jitter versus different sampling frequencies (step = $5*10^{-2}f_c$) for a GPS L1 C/A with C/N_0 = 45 dB-Hz, B_L = 0·5 Hz, T = 1 ms, and $\beta_r = f_s$.

the one-sigma code tracking jitter (in metres) also varies between σ_{s1} and σ_{s2}, where:

$$\sigma_{s1} = C\sqrt{\sigma_1^2 + \sigma_{u1N}^2 2B_L T} \tag{22}$$

$$\sigma_{s2} = C\sqrt{\sigma_2^2 + \sigma_{u2N}^2 2B_L T} \tag{23}$$

where C is the speed of light.

Note that σ_1^2 and σ_2^2 are constrained by the residual code phase $\theta_{NCO}(k)$, while the initial residual code phase $\theta_{NCO}(0)$ can have any random value in the range $[0, 1/n_s)$, so the code tracking loop error is altered. Figure 9 illustrates the maximum, minimum and mean values of σ_{s1} versus sampling frequencies while the residual code phase varies. Take n_s = 4 as an example, where the residual code phase is a constant for all code chips. The DLL jitter varies between the minimum value ($\approx$0·47 m), when $\theta_{NCO}(0) = 0$, and the maximum value ($\approx$52·3 m) when $\theta_{NCO}(0) \approx 1/n_s$. Since $\theta_{NCO}(0)$ is uniformly distributed in the range $[0, 1/n_s)$, the code tracking loop jitter can be approximated by the mean value of σ_{s1} or σ_{s2}. Figure 10 presents the mean value of σ_{s1} or σ_{s2} versus different sampling frequencies. The two values are approximately equal for most sampling frequencies, so for easy comparison only the mean value of σ_{s1} will be used for the remainder of the paper.

4.2. *Evaluation of theoretical results by simulated and numerical models.* In this section, the simulated and numerical models used to estimate the DLL tracking loop errors are presented. The theoretical result is calculated for GPS L1 C/A using Equation (22) with the following assumptions: the received signal has no AWGN, the front-end has an ideal filter with the two-sided bandwidth $\beta_r = f_s$ and the coherent integration period is T = 1 ms.

In the simulated model, the received pseudo-analogue signal was simulated by a sampled PRN code that was generated with frequency $f_{so} = 100f_s$. After that, it was passed through a decimator that simulates a front-end ADC with two-sided bandwidth $\beta_r \approx f_s$ and a sampling frequency f_s. This digital signal was then correlated with two locally generated E and L signals that are one code chip delayed. The desired timing offset between received and local generated signals, τ_0, was varied in the range $[-T_c, T_c]$ with step $\delta_\tau = 10^{-6}T_c$. The

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

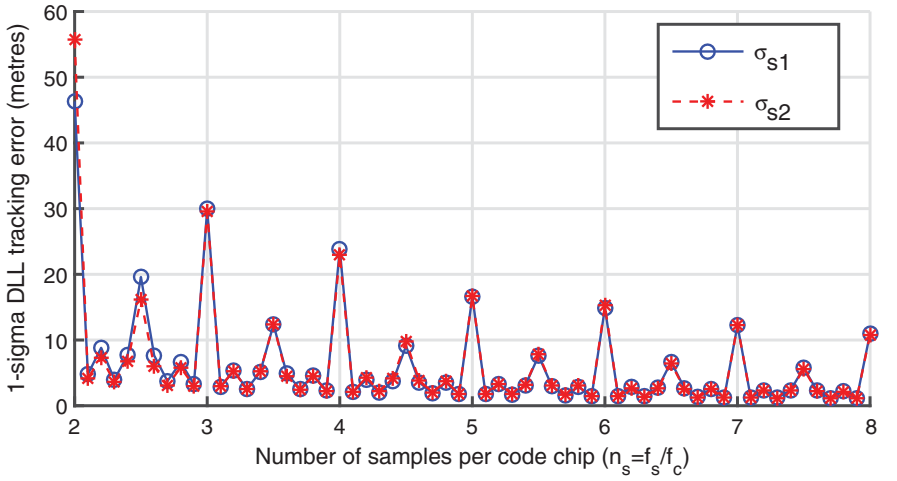


Figure 10. Mean values of two error bounds σ_{s1} and σ_{s2} versus different sampling frequencies (step = $10^{-1}f_c$) for a GPS L1 C/A with C/N_0 = 45 dB-Hz, B_L = 0·5 Hz, T = 1 ms, and $\beta_r = f_s$.

discrimination function $\frac{\mathrm{E}-\mathrm{L}}{2(\mathrm{E}+\mathrm{L})}$, is used to estimate the timing offset between the incoming signal and the local replica, τ_u. Finally, the DLL jitter (in metres) is estimated by the variance of the difference between τ_0 and τ_u as follows:

$$\sigma_s = CT_c\sqrt{var\,(\tau_u - \tau_0)} \tag{24}$$

The numerical model DLL jitter was calculated by using the same estimation model. The E and L correlator outputs were computed by Equation (7) with $\tau = -0{\cdot}5T_c$ and $0{\cdot}5T_c$, respectively. The DLL jitter $\hat{\sigma}_s$ is the mean of σ_s while the initial residual code phase $\theta_{NCO}(0)$ varies in the range $[0, 1/n_s)$.

Figure 11 illustrates the DLL tracking errors of these three models. It is clear that the three different models provide approximately the same results, for various sampling frequencies. This reinforces the correctness of the proposed theoretical model.

4.3. *Effect of Doppler and coherent integration period.* Figure 12 illustrates the DLL tracking error versus Doppler frequency, f_D, for different coherent integration periods, T of a GPS L1 C/A signal with the received signal C/N_0 = 40 dB-Hz; the sampling frequency f_s = 4·092 MHz (n_s = 4), and the ideal brick-wall ADC filter with two-sided bandwidth $\beta_r = f_s$. The presence of Doppler helps to minimise the DLL error if the sampling frequency is commensurate; however, Doppler shifts must be very high, for example $f_D \geq 38{\cdot}5$ kHz when T = 10 ms, this is out of typical Doppler range $[-10, 10]$ kHz.

If the navigation data bit has been wiped-off, the coherent integration period is not limited. Figure 13 describes the DLL tracking error versus the integration period, T, for different Doppler frequencies. Observe that, if the sampling frequency f_s = 4·092 (n_s = 4) and the Doppler frequency f_D = 2 kHz, the DLL error is a minimum when integration period $T > 193$ milliseconds. This result is equivalent to the predicted theoretical result shown in Section 3.2. Similarly, the DLL jitter is also lowest when $T \geq 40$ and 80 milliseconds while f_D = 5 and 10 kHz, respectively.

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

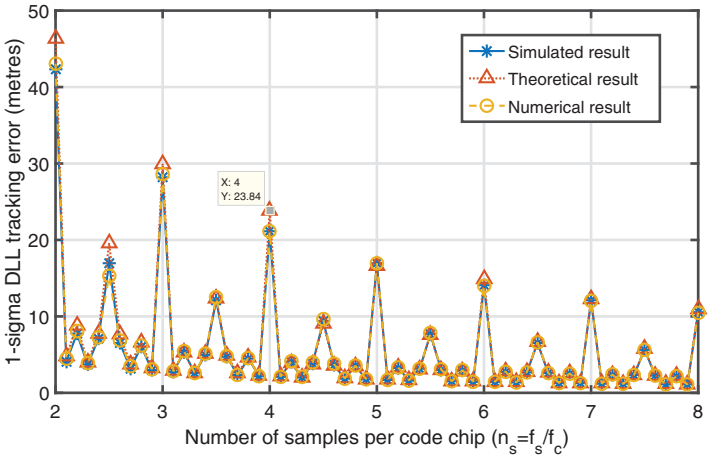


Figure 11. DLL tracking error comparison among the simulated, numerical and theoretical models (step = $10^{-1}f_c$) for a GPS L1 C/A with $T = 1$ ms, and $\beta_r = f_s$.

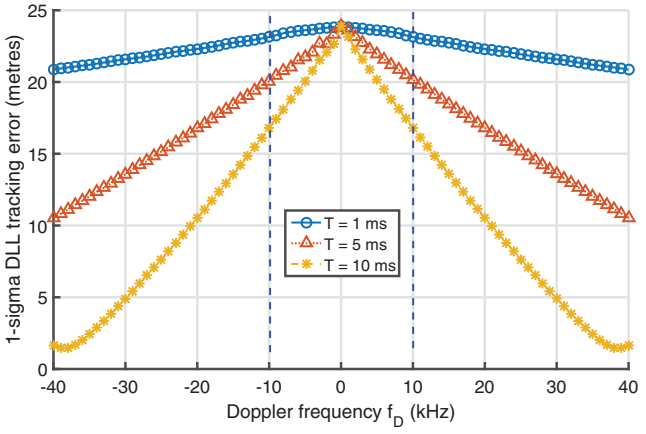


Figure 12. DLL tracking error versus Doppler frequencies f_D for different integration periods T when the sampling frequency is an integer multiple of the nominal code rate ($n_s = 4$), in which the blue dotted lines indicate the typical Doppler range.

If the sampling ratio f_s/f_c is non-integer, but the frequency difference δ_{f_s} from an integer multiple of the code rate is small, the code rate drift caused by the high Doppler frequency leads to accuracy degradation, as shown in Figure 14. This high Doppler environment does not occur on a terrestrial GNSS receiver, but may exist on a space-borne receiver, and it needs to be considered while selecting front-end sampling frequencies and integration periods.

To sum up, a very high Doppler frequency or a long coherent integration period can reduce the correlator and discriminator distortions (that is, minimise the DLL jitter) if the number of samples per code chip is an integer. However, if the sampling frequency is non-commensurate, a very high Doppler can lead to accuracy degradation. Therefore, the effect of the sampling frequency and integration periods on the receiver performance must be carefully considered for all Doppler frequencies.

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

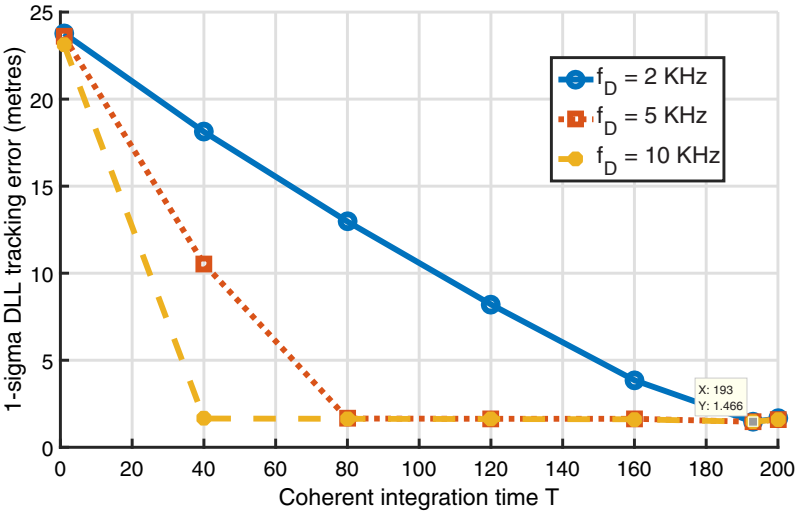


Figure 13. DLL tracking error versus integration periods T. GPS L1 C/A is used with f_s = 4·092 MHz (n_s = 4), C/N_0 = 40 dB-Hz, B_L = 0·5 Hz, T = 1 ms, and $\beta_r = f_s$.

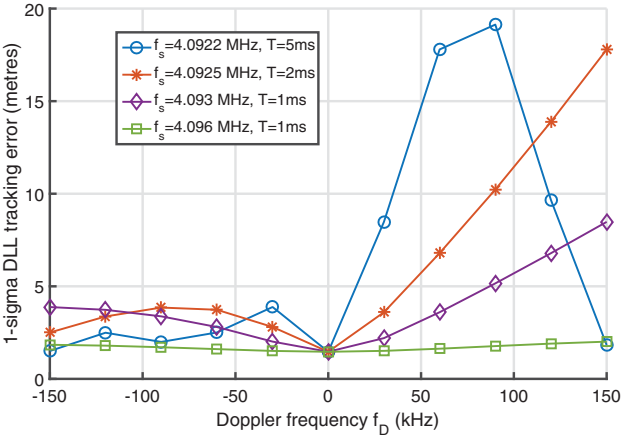


Figure 14. DLL tracking error versus Doppler frequencies f_D for different integration periods T when the sampling frequency is a non-integer multiple of the nominal code rate.

4.4. *Effect of the front-end filter bandwidth and the received* C/N_0. Figure 15 presents the DLL error versus different sampling frequencies for various C/N_0 when the two-sided bandwidth is fixed $\beta_r = 2f_c$ (for example β_r = 2·046 MHz for GPS L1 C/A signal). Observe that the one-sigma DLL tracking error is higher when the received signal is weak (C/N_0 = 10, 15, 20 dB-Hz). It is much smaller when the received C/N_0 is in the commonly received C/N_0 range (from 25 dB-Hz to 45 dB-Hz), which is equivalent to results presented in Kaplan and Hegarty (2005). The other observable trend is that the DLL error decreases when the sampling frequency increases. The more samples that are used to represent the incoming and local generated signals, the greater the accuracy at which the code chip transition can be detected. However, the DLL error is still significant at some

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

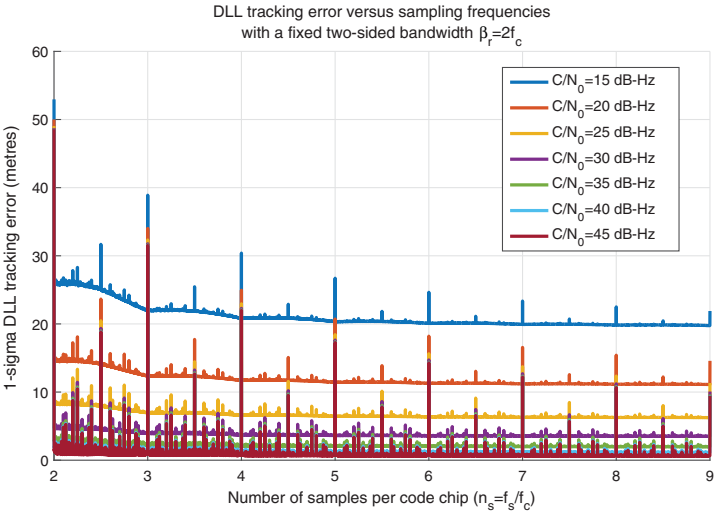


Figure 15. DLL tracking error versus different sampling frequencies (step = $10^{-3}f_c$) with a fixed front-end bandwidth $\beta_r = 2f_c$ for a GPS L1 C/A signal with $B_L = 0{\cdot}5$ Hz and $T = 1$ ms.

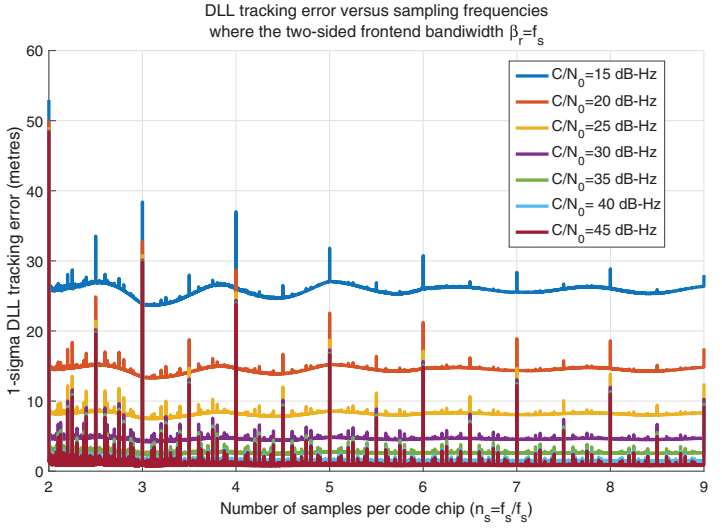


Figure 16. DLL tracking error versus different sampling frequencies (step = $10^{-3}f_c$) with front-end bandwidths $\beta_r = f_s$ for a GPS L1 C/A signal with $B_L = 0{\cdot}5$ Hz and $T = 1$ ms.

peak points due to the effect of sampling frequencies. It is considerable for every received C/N_0, but it is much more visible when the received signal is strong ($C/N_0 > 30$ dB-Hz). This error cannot be observed if the DLL tracking error estimation proposed by Betz and Kolodziejski (2009a; 2009b) is applied. Moreover, this DLL error is not only significant when the sampling rate is commensurate, but it is also considerable, when the ratio f_s/f_c is at some non-commensurate values. Therefore, the effect of the sampling frequency should be taken into account when choosing the front-end ADC sampling rate.

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

Table 1. DLL tracking error (standard deviation in metres) versus different two-sided front-end bandwidths. A live GPS L1 C/A signal is recorded by the Nuand bladeRF front-end and processed with DLL tracking parameters B_L = 0·5 Hz, T = 1 ms.

	Two-sided bandwidth (β_r)		
C/N_0(PRN)	2·2 MHz	2·7 MHz	3·6 MHz
51 dB-Hz (8)	15·51	16·36	18·13
45 dB-Hz (7)	16·55	18·08	19·76
42 dB-Hz (1)	22·97	26·25	25·48
40 dB-Hz (16)	25·74	27·63	29·15
35 dB-Hz (3)	27·33	29·63	31·34

The DLL tracking errors versus sampling frequencies with different two-sided front-end bandwidths $\beta_r = 2f_c$ and $\beta_r = f_s$ are illustrated in Figures 15 and 16, respectively. Observe that the DLL error decreases when the sampling frequency increases if the front-end bandwidth is fixed, $\beta_r = 2f_c$ (Figure 15). This is because the noise power is only governed by the front-end bandwidth, the coherent integration period, and it is not affected by the sampling frequency, as shown in Equation (20). The DLL error contributed by the noise power is thus the same when the sampling frequency increases. Consequently, the DLL error, excluding the peak impacted by the sampling frequency, is reduced when the sampling frequency is higher. However, this DLL tracking error is approximately constant if the front-end bandwidth is increased due to the sampling frequency, $\beta_r = f_s$ (Figure 16). The wider the front-end bandwidth, the greater the received signal power, but the received signal also includes AWGN so noise power is also increased. Consequently, the DLL tracking noise error fluctuates slightly, but remains unchanged in general. Because the DLL error reduction, which occurs by increasing the sampling frequency and front-end bandwidth, is approximately offset by the error increase due to the growth of the noise power, the DLL error is much more visible when the received signal power is weak ($C/N_0 < 30$ dB-Hz).

Experiments were set up to validate these conclusions. A Nuand bladeRF front-end was used to collect live GPS L1 C/A signal from a rooftop active antenna. A sampling frequency was chosen at f_s = 4·092 MHz (n_s = 4) and the signal was processed with various In-phase (I) and Quadrature (Q) front-end bandwidths of 1·5, 1·75 and 2·5 MHz, which is equivalent to the actual complex front-end bandwidth $\beta_r \approx 2{\cdot}2, 2{\cdot}7$, and 3·6 MHz, respectively. The DLL tracking error was calculated as the variance of 10,000 DLL discriminator outputs with a coherent integration period T = 1 ms and the noise bandwidth B_L = 0·5 Hz. A uBlox 6T simultaneously processed the received signal to estimate the C/N_0 of each satellite. The experimental result, as presented in Table 1, validates the previous theoretical analysis that the narrower the front-end bandwidth, the smaller the DLL tracking error. Furthermore, the DLL tracking errors, as shown in Table 1 when the C/N_0 is 45, 42, 40 and 35 dB-Hz, are matched with the theoretical results presented in Figure 15 and Figure 16 where the number of samples per code chip n_s = 4.

All in all, the effect of front-end bandwidth also needs to be considered in addition to the effect of the sampling frequency and the received C/N_0. Increasing the sampling frequency helps to reduce the DLL tracking error, and it is more effective for weak signals if and only if the front-end bandwidth is fixed and much smaller than the sampling frequency.

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

5. CONCLUSION. Sampling frequency has a strong influence on the synchronisation of the incoming and local generated signals and the code DLL tracking. The number of distinct residual code phases of each code chip and their distribution along the code phase range $[0, 1/n_s)$ is the main cause of the correlation and discriminator distortions, that was practically observed by Thomas (1988) for one specific sampling frequency. The DLL tracking error is not only considerable when the sampling frequency is commensurate, but also non-commensurate if the number of distinct residual code phases is low. The calculation of the number of distinct residual code phases in Section 3 and the theoretically proposed coherent EML DLL tracking error estimation in Section 4 further confirm this effect and can be used as a theoretical guideline for the sampling frequency and the coherent integration period selection.

The Doppler frequency can help to reduce the distortion caused by a bad sampling frequency choice; however, it is only valid if the Doppler frequency is very high, such as in a very high dynamic environment, or if the coherent integration is very long, such as when the navigation data bit transitions were predicted, or for a data-less (Pilot) channel. For a terrestrial receiver, the Doppler frequency has an insignificant effect on reducing the DLL errors because of the coherent integration that is limited by the secondary code or data bit transition.

The impact of the sampling frequency on correlation output, which is usually presented by the sampling filter bandwidth when the number of samples per code chip is assumed to be infinite, has been accurately calculated based on the number of samples per code chip in Section 2. The well-known DLL tracking error estimation proposed by Betz and Kolodziejski (2000; 2009a; 2009b) has been revised under the effect of the sampling frequency in Section 4. A more accurate estimation model has been proposed and proved mathematically (as shown in Appendix B). The previous estimation could not indicate the bias error constrained by the sampling frequency, whereas the proposed method cannot only cover the conventional DLL tracking error influenced by the noise power, but also illustrates the effect of the sampling frequency.

The front-end ADC filter bandwidth also has a strong effect on the DLL jitter. The DLL tracking error is decreased while the sampling frequency is increased and it is more efficient for weak signals ($C/N_0 < 30$ dB-Hz) if and only if the front-end bandwidth is fixed and much smaller than the sampling frequency.

REFERENCES

Akos, D. M. and Braasch, M. S. (1996). A software radio approach to global navigation satellite system receiver design. *Proceedings of the 52nd Annual Meeting of The Institute of Navigation*, 455–463.

Akos, D. M. and Pini, M. (2006). Effect of sampling frequency on GNSS receiver performance. *Navigation,* **53**(2), 85–95.

Betz, J. W. and Kolodziejski, K. R. (2000). Extended theory of Early-Late code tracking for a bandlimited GPS receiver. *Navigation*, **47**(3), 211–226.

Betz, J. and Kolodziejski, K. (2009a). Generalized theory of code tracking with an Early Late discriminator Part I: Lower bound and coherent processing. *IEEE Transactions on Aerospace and Electronic Systems,* **45**(4), 1538–1556.

Betz, J. W. and Kolodziejski, K. R. (2009b). Generalized theory of code tracking with an Early Late discriminator Part II: Noncoherent processing and numerical results. *IEEE Transactions on Aerospace and Electronic Systems*, **45**(4), 1557–1564.

Betz, J. W. (2015). *Engineering Satellite-Based Navigation and Timing: Global Navigation Satellite Systems, Signals, and Receivers*. John Wiley & Sons. URL: http://onlinelibrary.wiley.com/book/10.1002/9781119141167.

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

Braasch, M. S. and Van Dierendonck, A. J. (1999). GPS receiver architectures and measurements. *Proceedings of the IEEE*, **87**(1), 48–64.
Hegarty, C. J. (2011). Analytical model for GNSS receiver implementation losses. *Navigation*, **58**(1), 29–44.
Jang, J., Paonni, M. and Eissfeller, B. (2012). CW interference effects on tracking performance of GNSS receivers. *IEEE Transactions on Aerospace and Electronic Systems*, **48**(1), 243–258.
Kaplan, E. D. and Hegarty, C. J. (2005). *Understanding GPS: principles and applications.* Artech House.
Kou, Y. and Morton, Y. (2013). Oscillator frequency offset impact on software GPS receivers and correction algorithms. *IEEE Transactions on Aerospace and Electronic Systems.* **49**(4), 2158–2178.
Nuand-LLC. (2016). Nuand bladeRF x40 Technical Brief Description, Technical report, URL: https://www.nuand.com/bladeRF-brief.pdf.
Parkinson, B. W. (1996). *Global Positioning System: Theory and Applications, Vol. 1*. American Institute of Aeronautics and Astronautics.
Pini, M. (2007). Constraints Analysis of Radio Frequency Front-End Components: Impact on the GNSS Signal Processing, PhD thesis, Politecnico di Torino.
Shivaramaiah, N. C., Dempster, A. G. and Rizos, C. (2008). Exploiting the secondary codes to improve signal acquisition performance in Galileo receivers. *Proceedings of ION GNSS*, 1497–1506.
Thomas, J. B. (1988). *Functional Description of Signal Processing in the Rogue GPS Receiver.* National Aeronautics and Space Administration Jet Propulsion Laboratory, JPL Publication 88-15, Pasadena, California, June 1, 1988.
Tran, V. T., Shivaramaiah, N. C. and Dempster, A. G. (2015). An efficient secondary code transition cancellation correlator for fast multi-GNSS acquisition. *Proceedings of the IGNSS Symposium on GPS/GNSS*.
Tran, V. T., Shivaramaiah, N. C. and Dempster, A. G. (2017). Feasibility analysis of baseband architectures for multi-GNSS receivers. *GPS Solutions*, **21**(1), 1–11. URL: http://link.springer.com/article/10.1007/s10291-016-0542-0.
Zhang, X., & Zhan, X. (2012). Joint Determination of Precorrelation Bandwidth, Sampling Frequency and Quantization in Wideband Compass Receivers. Navigation, 59(3), 195–208.

APPENDIX A. CORRELATION OUTPUT CALCULATION

The local replica PRN can be illustrated as:

$$C_{local}(n) = \sum_{k=0}^{T_0N-1} \sum_{m=0}^{M_k-1} C_{\{k\}_N} \delta\left[n - m - N_{M_k}\right] \tag{A1}$$

where $N_{M_k} = \sum_{j=0}^{j=k} M_j$. Applying discrete time Fourier transform on the local replica PRN with $\omega = 2\pi f T_s$:

$$\begin{aligned} C(\omega) &= \sum_{n=-\infty}^{\infty} \sum_{k=0}^{T_0N-1} \sum_{m=0}^{M_k-1} C_{\{k\}_N} \delta\left[n - m - N_{M_k}\right] e^{-i\omega n} \\ &= \sum_{k=0}^{T_0N-1} C_{\{k\}_N} e^{-i\omega N_{M_k}} \sum_{m=0}^{M_k-1} e^{-i\omega m} \sum_{n=-\infty}^{\infty} \delta[n] \\ &= \sum_{k=0}^{T_0N-1} C_{\{k\}_N} e^{-i\omega k N_{M_k}} \frac{sin(M_k\omega/2)}{sin(\omega/2)} e^{-i\frac{(M_k+1)\omega}{2}} \end{aligned} \tag{A2}$$

so it can be illustrated in the frequency domain as:

$$C(f) = \sum_{k=0}^{T_0N-1} C_{\{k\}_N} e^{-i2\pi f T_s N_{M_k}} \frac{sin(M_k\pi f T_s)}{sin(\pi f T_s)} e^{-i(M_k+1)\pi f T_s} \tag{A3}$$

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

and its reverse Fourier transform is:

$$\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_k\delta\left[n-m-N_{M_k}\right] = T_s\int_{-\beta_r/2}^{\beta_r/2} C(f)\,e^{i2\pi fT_s n}df \tag{A4}$$

where β_r is the receiver front-end bandwidth.

Substituting Equation (A4) into Equation (6), yields

$$\begin{aligned}
R(\tau) &= \frac{C_sT_s}{N_L}\sum_{n=0}^{N_L-1}\sum_{k=0}^{T_0aN-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta\left[n-m-N_{M_k}-\lfloor\tau+\theta_{NCO}(k)T_c\rfloor_{T_s}\right]\int_{-\beta_r/2}^{\beta_r/2} C(f)e^{i2\pi fT_s n}df \\
&= \frac{C_sT_s}{N_L}\int_{-\frac{\beta_r}{2}}^{\beta_r/2}\sum_{n=0}^{N_L-1} C(f)\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta[n-m-N_{M_k}-\lfloor\tau+\theta_{NCO}(k)T_c\rfloor_{T_s}]e^{i2\pi fT_s n}df \\
&\cong \frac{C_sT_s}{N_L}\int_{-\beta_r/2}^{\beta_r/2}\sum_{k=0}^{T_0N-1}\left(\frac{sin(\pi fM_kT_s)}{sin(\pi fT_s)}\right)^2 C_{\{k\}_N}e^{i2\pi fT_s(N_{M_k}+\lfloor\tau+\theta_{NCO}(k)T_c\rfloor_{T_s})}\sum_{l=0}^{T_0N-1} C_{\{l\}_N}e^{-i2\pi fT_slM_l}df \\
&= \frac{C_sT_s}{N_L}\int_{-\beta_r/2}^{\beta_r/2}\sum_{k=0}^{T_0N-1}\left(\frac{sin(\pi fM_kT_s)}{sin(\pi fT_s)}\right)^2\left(\begin{array}{l}|C_{\{k\}_N}|^2e^{i2\pi fT_s(\lfloor\tau+\theta_{NCO}(k)T_c\rfloor_{T_s})} \\ +\sum_{l=0,\not=k}^{T_0N-1} C_{\{k\}_N}C_{\{l\}_N}e^{i2\pi fT_s((N_{M_k}-N_{M_l})+\lfloor\tau+\theta_{NCO}(k)T_c\rfloor_{T_s})}\end{array}\right)df \\
&= \frac{C_s}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} T_s\left(\frac{\sin(\pi fM_kT_s)}{\sin(\pi fT_s)}\right)^2 e^{i2\pi fT_s(\lfloor\tau+\theta_{NCO}(k)T_c\rfloor_{T_s})}df
\end{aligned} \tag{A5}$$

for a pseudorandom noise code sequence C_k, $k = 0, 1, 2, \ldots, N$, all misaligned chips with $k = l$ have equal probability of having +1 or −1 values. Their product sum approaches zero.

APPENDIX B. ERROR ANALYSIS FOR COHERENT EARLY MINUS LATE DLL

Define an error signal at the output of the real-part operator in the coherent early minus late TOA estimator as

$$\begin{aligned}
e(\epsilon) = \Re\Bigg\{ & R(\epsilon-\Delta/2)-R(\epsilon+\Delta/2) \\
& +\frac{\sqrt{C_s}}{N_L}\sum_{n=0}^{N_L-1} w(n)\left(\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta[n-m-kM_k-\lfloor\epsilon-\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}]\right. \\
& \left.\left.-\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta[n-m-kM_k-\lfloor\epsilon+\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}]\right)\right\}
\end{aligned} \tag{B1}$$

where $\varepsilon = t_0 - t_s$ is the code phase misalignment, t_o is the mean value of the unsmoothed TOA estimate (t_u), t_s is the smoothed TOA estimate, and Δ is the Early - Late code space ($0 < \Delta \leq T_c$).

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

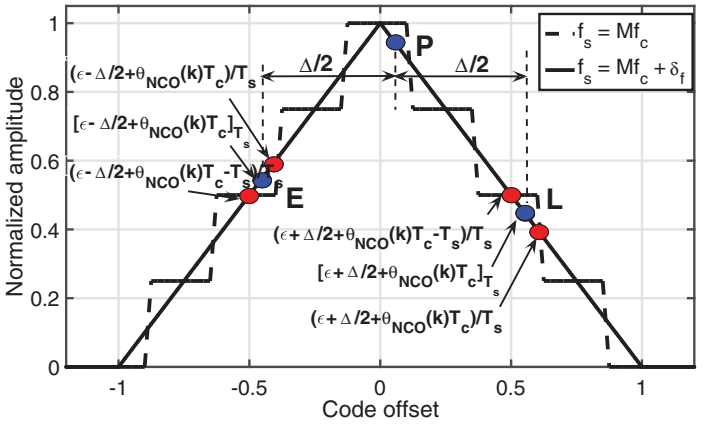


Figure B1. Early and Late approximated values based on the correlation shape.

Denote the first two terms in Equation (B1) as $e_E(\varepsilon)$ and $e_L(\varepsilon)$, respectively. Applying Equation (A5), this gives:

$$e_E(\epsilon) = \Re\left\{\frac{C_s}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)e^{i2\pi fT_s\left(\lfloor\epsilon-\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}\right)}df\right\}$$

$$= \frac{C_s}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)\cos\left(2\pi fT_s\lfloor\epsilon-\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}\right)df \quad \text{(B2)}$$

$$e_L(\epsilon) = \Re\left\{\frac{C_s}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)e^{i2\pi fT_s\left(\lfloor\epsilon+\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}\right)}df\right\}$$

$$= \frac{C_s}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)\cos\left(2\pi fT_s\lfloor\epsilon+\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}\right)df \quad \text{(B3)}$$

It is clear that $(\epsilon \pm \Delta/2 + \theta_{NCO}(k)T_c - T_s)/T_s \le \lfloor\epsilon \pm \Delta/2 + \theta_{NCO}(k)T_c\rfloor_{T_s} \le (\epsilon \pm \Delta/2 + \theta_{NCO}(k)T_c)/T_s$. Furthermore, $e_E(\varepsilon)$ and $e_L(\varepsilon)$ are respectively on the left and right slope of the correlator shape as shown in Figure B1, when ε is small.

Therefore:

$$e_E(\epsilon) \le \frac{C_s}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)\cos\left(2\pi f\ (\epsilon-\Delta/2+\theta_{NCO}(k)T_c)\right)df \quad \text{(B4)}$$

$$e_E(\epsilon) \ge \frac{C_s}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)\cos\left(2\pi f\ (\epsilon-\Delta/2+\theta_{NCO}(k)T_c-T_s)\right)df \quad \text{(B5)}$$

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

similarly, $e_L(\varepsilon)$ is:

$$e_L(\epsilon) \leq \frac{C_s}{N_L} \sum_{k=0}^{T_0 N-1} \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f) \cos\left(2\pi f \ (\epsilon + \Delta/2 + \theta_{NCO}(k)\, T_c)\right) df \tag{B6}$$

$$e_L(\epsilon) \geq \frac{C_s}{N_L} \sum_{k=0}^{T_0 N-1} \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f) \cos\left(2\pi f \ (\epsilon + \Delta/2 + \theta_{NCO}(k)\, T_c - T_s)\right) df \tag{B7}$$

Defining $e_s(\varepsilon) = e_E(\varepsilon) - e_L(\varepsilon)$, it leads to:

$$\begin{aligned} e_s(\epsilon) &\leq \frac{2C_s}{N_L} \sum_{k=0}^{T_0 N-1} \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f) \sin(\pi f \Delta) \begin{pmatrix} 2\pi f\, \epsilon \cos(2\pi f\, \theta_{NCO}(k)\, T_c) \\ + \sin(2\pi f\, \theta_{NCO}(k)\, T_c) \end{pmatrix} df \\ &\leq C_s K_1 \epsilon + C_s K_2 \end{aligned} \tag{B8}$$

and

$$\begin{aligned} e_s(\epsilon) &\geq \frac{2C_s}{N_L} \sum_{k=0}^{T_0 N-1} \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f) \sin(\pi f \Delta) \begin{pmatrix} 2\pi f\, \epsilon \cos(2\pi f\, (\theta_{NCO}(k)\, T_c - T_s)) \\ + \sin(2\pi f\, (\theta_{NCO}(k)\, T_c - T_s)) \end{pmatrix} df \\ &\geq C_s K_3 \epsilon + C_s K_4 \end{aligned} \tag{B9}$$

since ε is small and where

$$K_1 = \frac{1}{N_L} \sum_{k=0}^{T_0 N-1} \int_{-\beta_r/2}^{\beta_r/2} 4\pi f G_{s_K}(f) \sin(\pi f \Delta) \cos(2\pi f\, \theta_{NCO}(k) T_c) df \tag{B10}$$

$$K_2 = \frac{2}{N_L} \sum_{k=0}^{T_0 N-1} \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f) \sin(\pi f \Delta) \sin(2\pi f\, \theta_{NCO}(k) T_c) df \tag{B11}$$

$$K_3 = \frac{1}{N_L} \sum_{k=0}^{T_0 N-1} \int_{-\beta_r/2}^{\beta_r/2} 4\pi f G_{s_K}(f) \sin(\pi f \Delta) \cos(2\pi f\, (\theta_{NCO}(k) T_c - T_s)) df \tag{B12}$$

$$K_4 = \frac{2}{N_L} \sum_{k=0}^{T_0 N-1} \int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f) \sin(\pi f \Delta) \sin(2\pi f\, (\theta_{NCO}(k) T_c - T_s)) df \tag{B13}$$

Therefore, the error signal is:

$$e\,(t_0 - t_s) \leq C_s K_1 (t_0 - t_s) + C_s K_2 + N_w \tag{B14}$$

and

$$e\,(t_0 - t_s) \geq C_s K_3 (t_0 - t_s) + C_s K_4 + N_w \tag{B15}$$

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

with

$$N_w = \mathcal{R}\left\{\frac{\sqrt{C_s}}{N_L}\sum_{n=0}^{N_L-1} w(n)\left(\begin{array}{l}\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta\left[n-m-kM_k-\lfloor\epsilon-\Delta/2+\theta_{NCO}(k)\,T_c\rfloor_{T_s}\right]\\ -\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_K-1} C_{\{k\}_N}\delta\left[n-m-kM_k-\lfloor\epsilon+\Delta/2+\theta_{NCO}(k)\,T_c\rfloor_{T_s}\right]\end{array}\right)\right\} \tag{B16}$$

and its variance for a given t_s is:

$$var\left(e\left(t_0-t_s\right)|t_s\right) \le \lim_{t_o\to t_s}\frac{\left[C_s\left(K_1+K_3\right)\left(t_0-t_s\right)+C_s\left(K_2+K_4\right)+2N_w\right]^2}{4} \tag{B17}$$

thus

$$var(e\left(t_0-t_s\right)|t_s) \cong \frac{C_s^2(K_2+K_4)^2}{4}+E\left\{N_w^2\right\}+C_s\left(K_2+K_4\right)E\left\{N_w\right\} \tag{B18}$$

Because of the circular symmetry of $w(n)$, $E\{N_w\} \approx 0$, and

$$E\{N_w^2\} = \frac{1}{2}\frac{C_s}{(N_L)^2}\left\{\begin{array}{l}\sum_{n=0}^{N_L-1}\sum_{n=0}^{N_L-1} w(n)w^*(n)\\ \times\left(\begin{array}{l}\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta[n-m-kM_k-\lfloor\epsilon-\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}\\ -\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta[n-m-kM_k-\lfloor\epsilon+\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}]\end{array}\right)\\ \times\left(\begin{array}{l}\sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta^*[n-m-kM_k-\lfloor\epsilon-\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}]\\ \sum_{k=0}^{T_0N-1}\sum_{m=0}^{M_k-1} C_{\{k\}_N}\delta^*[n-m-kM_k-\lfloor\epsilon+\Delta/2+\theta_{NCO}(k)T_c\rfloor_{T_s}]\end{array}\right)\end{array}\right\} \tag{B19}$$

in which $\sum_{n=0}^{N_L-1} : w(n)\,w^*(n) = N_0N_L$ is the noise power of N_L samples with N_0 is the power spectral density of noise. Then following analysis, similar to that in Equation (A5), leads to:

$$\begin{aligned} E\left\{N_w^2\right\} &\cong \frac{C_sN_0}{(N_L)}\sum_{k=0}^{T_0N-1} T_s\int_{-\beta_r/2}^{\beta_r/2}\frac{\sin^2(\pi fM_kT_s)}{\sin^2(\pi fT_s)}\left(1-\cos(2\pi f\Delta)\right)df \\ &= \frac{2C_sN_0}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)\sin^2(\pi f\Delta)\,df \end{aligned} \tag{B20}$$

Therefore, the variance of the signal error for a given t_s is:

$$var(e\left(t_0-t_s\right)|t_s) \cong \frac{C_s^2(K_2+K_4)^2}{4}+\frac{2C_sN_0}{N_L}\sum_{k=0}^{T_0N-1}\int_{-\beta_r/2}^{\beta_r/2} G_{s_k}(f)\sin^2(\pi f\Delta)\,df \tag{B21}$$

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press

From the coherent early-late TOA estimator block diagram, the unsmoothed TOA estimate is:

$$t_u = t_s + \frac{e\,(t_0 - t_s)}{C_s K} \tag{B22}$$

combining with Equations (B8) and (B9) the estimated range of the unsmoothed TOA is:

$$t_s + \frac{e\,(t_0 - t_s)}{C_s K_1} \leq t_u \leq t_s + \frac{e\,(t_0 - t_s)}{C_s K_3} \tag{B23}$$

mathematically, the conditional variance of t_u for a given t_s is:

$$var(t_u|t_s) = \frac{\sigma(e\,(t_o - t_s)\,|t_s)}{C_s^2 K^2} \tag{B24}$$

The variance of the unsmoothed TOA estimator σ_u^2, therefore, varies between σ_{u1}^2 and σ_{u2}^2, where:

$$\sigma_{u1}^2 = \sigma_1^2 + \sigma_{u1N}^2 \tag{B25}$$

and

$$\sigma_{u2}^2 = \sigma_2^2 + \sigma_{u2N}^2 \tag{B26}$$

in which σ_1^2 and σ_2^2, which are respectively shown in Equations (18) and (19), are the DLL jitters that are influenced by the sampling frequency. Similarly, σ_{u1N}^2 and σ_{u2N}^2, which are respectively illustrated in Equations (20) and (21), are the DLL tracking errors that are controlled by noise power.

https://doi.org/10.1017/S0373463317000741 Published online by Cambridge University Press