

Development of a Real-Time Software-Defined Receiver for Broadband LEO PNT

W. Jeremy Morrison*, Todd E. Humphreys*

**Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin*

ABSTRACT

This paper offers a first look at the development of a software defined radio (SDR) capable of processing signals from low-Earth-orbit (LEO) constellations for time-of-arrival and frequency-of-arrival measurements in real time. LEO signals offer greater potential accuracy and resilience to jamming and spoofing than traditional global navigation satellite system (GNSS) signals, due to their high power and wide bandwidth. SDRs have historically enabled rapid innovation cycles for traditional GNSS. However, the large bandwidth of LEO mega-constellation signals have been prohibitive for using “pure” SDRs—those that perform correlation on a general purpose processor—to process these signals in real-time. This paper showcases the development of LEONARD, the first pure SDR capable of real-time LEO broadband processing. Differences from the real-time tracking of GNSS signals are discussed and preliminary tracking results are shown.

I. INTRODUCTION

Besides revolutionizing global communications, recently launched broadband low-Earth-orbit (LEO) mega-constellations are poised to revolutionize global positioning, navigation, and timing (PNT). Compared to traditional global navigation satellite systems (GNSS), they offer higher power, much wider bandwidth, more rapid multipath decorrelation, and the possibility of stronger authentication and zero-age-of-ephemeris, all of which will enable greater accuracy and greater resilience against jamming and spoofing [1]–[11].

Today, much of the world’s infrastructure relies on accurate PNT. Traditional GNSS-derived PNT has exhibited a lack of robustness with the advent of widespread targeted interference, due to the low power and relatively narrow bandwidth of GNSS signals. LEO mega-constellations are now poised to become the new suppliers of precise and resilient PNT, either as GNSS augmentations or stand-alone services. Table I details possible strategies for exploiting LEO signals for PNT.

Recent results of pseudorange-based LEO broadband PNT indicate the Fused and Network-Aided approaches are capable of sub-meter accuracy, exhibit a high degree of resistance to multipath, and require much less time to a solution fix than stand-alone opportunistic methods [9]–[11].

The key to such astonishing performance is the extremely wide bandwidths available from broadband LEO communications satellites. For example, each Starlink channel spans 240 MHz [5] and each OneWeb channel spans 230 MHz [26]. A

single Starlink or OneWeb channel dwarfs the bandwidth of all traditional GNSS services combined!

The results reported in [7]–[11] and elsewhere all depend on post-processing of captured LEO broadband signals, typically with code written in a scripting language such as Python or Matlab. Such post-processing is practical and effective for exploring new techniques and feasibility analyses, yet many real-world applications require real-time or near real-time PNT solutions.

So far as we are aware there currently exist no real-time software-defined radio (SDR) receivers for broadband LEO PNT that have been reported in the open literature. Yet such a receiver would be of immense value for further research and development of this emerging field of broadband LEO PNT. SDR would be ideal for this task, as it would enable rapid innovation cycles, just as it has for traditional GNSS [27].

Given the extremely wide bandwidths involved, it has been unclear whether a “pure” SDR—one whose full pipeline, including correlation, takes place on a general-purpose processor—would be capable of processing broadband LEO signals. To capture a full 240-MHz Starlink channel, the SDR would need to process complex samples arriving at a rate greater than 240 Msps. Fortunately, generating the time of arrival measurements needed for pseudorange-based PNT, while benefiting in precision from high bandwidth, does not require the capture of a full channel.

Aside from the increased bandwidth, processing LEO signals differs from processing traditional GNSS signals in two key ways. First, the satellite vehicle (SV) movement, as seen from the Earth, is much more dynamic, and SVs use highly directive phased arrays. These factors imply that an SDR tracking LEO signals must spend a significant amount of resources on repeated acquisition of signals, akin to the LEO network entry problem presented in [28]. Second, the signal structure of commercial broadband LEO signals is not as well known or predictable as GNSS signals are. Most of the data carried by these signals, in nominal operations with customers, appear random due to encryption and error correction coding. These random parts of the signals cannot be used for correlation without further processing or network aiding; only known and deterministic pilots or synchronization symbols can be used as such. To use the random data symbols of the signals, the incoming signals must first be decoded using estimation methods as presented in [9], putting further computational strain on the SDR and requiring high SNR.

This paper presents our initial development of the LEO Navigation Assimilation Radio Device (LEONARD). It is our

TABLE I: LEO PNT Trade Space from [10]

	Dedicated [12]–[14]	Fused [4], [15]–[17]	Network-Aided Opportunistic [8], [11], [18]–[20]	Stand-Alone Opportunistic [21]–[25]
Description	LEO constellation or hosted payloads dedicated solely to PNT.	Fuses a secondary PNT service with a primary communications service.	Like stand-alone opportunistic, but a network of reference stations provides corrections.	Exploits unmodified signals from communications SVs. Public ephemerides. No network aiding.
Marginal deployment cost	high: constellation of SVs	low: uses communications hardware and signals	high: network of reference stations	very low
Potential availability	mid-term	near-term	near-term for local coverage	immediate
Potential accuracy [m]	< 1	< 1	< 1	< 100
Time to fix [s]	< 10	< 10	< 10	< 1000
Dependency on traditional GNSS	@SVs in near-term	@SVs in near-term	@reference stations in near-term	@SVs in some cases

vision to make LEONARD capable of tracking live LEO broadband signals and producing from them a stream of precise time-of-arrival (TOA) measurements. To exemplify a LEO broadband communication signal we chose the Starlink Ku-band downlink signal due to Starlink’s position as industry leader. The following sections introduce the Starlink signal, detail some of the differences between tracking traditional GNSS signals and Starlink signals, and show LEONARD’s preliminary performance on real-world data.

II. LEO COMMUNICATIONS SIGNAL STRUCTURE: STARLINK

We choose Starlink signals to demonstrate the real-time processing of LEO signals, since their constellation is the largest and their signal structure is well understood through prior research. Each of Starlink’s eight OFDM channels span a 240-MHz bandwidth and each frame slot—the period during which a frame could be sent—spans 1/750 seconds. Every frame consists of 302 OFDM symbols, most of which are filled with unpredictable user-specific data. The first two symbols in a frame are termed the primary synchronization sequence (PSS) and the secondary synchronization sequence (SSS) [5]. While it has been shown that other persistent features in the Starlink signal exist [7], their exact form has not been published at the time of writing. A diagram of the frame structure can be seen in Fig. 1.

Every frame slot is either occupied with a transmitted frame or devoid of signal. The density of transmitted frames occupying frame slots varies based on customer demand near the receiver used to capture the signal and, though often exhibiting patterns, is generally not predictable.

III. FROM TRACKING GNSS SIGNALS TO TRACKING STARLINK SIGNALS

LEONARD inherits its internal structure from the science-grade GRID SDR developed at UT Austin over the 15 years. GRID is a PNT-dedicated real-time SDR, that has until now

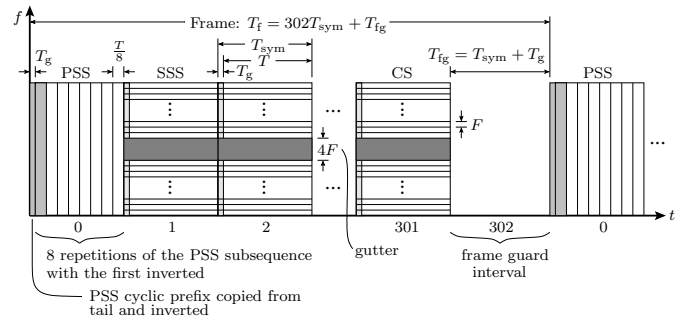


Fig. 1: A diagram showing the structure of the Starlink downlink frame as discovered in [5]. The exact values of the various time periods can be seen in table II.

only been used to process CDMA-based GNSS signals, whose structure differs significantly from the OFDM-based Starlink signals. While the origins and computational efficiency of GRID lie in bit-wise parallel correlation operations, it has been expanded to correlate binary codes with up to 16-bit quantized RF data.

In the following subsections we will discuss the changes implemented in LEONARD to track OFDM-based signals as opposed to GNSS signals. We will also discuss tracking methods and their feasibility given a Starlink-like LEO signal.

A. The Local Code Replica

Traditional GNSS signals are CDMA based—they are spread across their allocated spectrum using ranging codes. These ranging codes are the local code replicas used for correlation in an SDR, and can be generated from one or two pseudo-random binary sequences. Hence, GRID’s internal architecture had never been designed to generate and correlate codes like the Starlink SSS, whose time-domain representation is not binary. LEONARD could have generated a binary form of the SSS and accepted the commensurate quantization losses, but we decided, looking towards other signal types and yet-

TABLE II: Starlink Downlink Signal Parameters from [5]

Name	Parameter	Value	Units
Channel bandwidth	F_s	240	MHz
Number of subcarriers in F_s	N	1024	
Number of cyclic prefix intervals	N_g	32	
Frame period	T_f	1/750	s
Frame guard interval	T_{fg}	$68/15 = 4.5\overline{33}$	μs
Number of non-zero symbols per frame	N_{sf}	302	
Useful OFDM symbol interval	$T = N/F_s$	$64/15 = 4.2\overline{66}$	μs
Symbol guard interval	$T_g = N_g/F_s$	$2/15 = 0.1\overline{33}$	μs
OFDM symbol duration	$T_{sym} = T + T_g$	4.4	μs
Subcarrier spacing	$F = F_s/N$	234375	Hz
Center frequency of i th channel	F_{ci}	$10.7 + F/2 + 0.25(i - 1/2)$	GHz
Channel spacing	F_δ	250	MHz
Width of guard band between channels	F_g	10	MHz

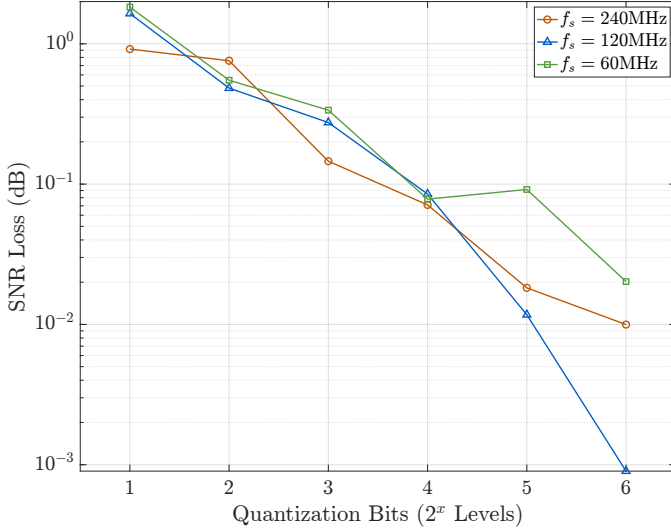


Fig. 2: Quantization of the code replica (PSS+SSS) and its impact on post-correlation SNR. This plot was generated by correlating the quantized local replica against primary beam's frame from captured data at a different IQ sampling rates. Quantization losses are less than 2 dB.

to-be-uncovered synchronization sequences, to build a new code generator that can generate finely-quantized codes. The quantization losses for Starlink's PSS and SSS combination are seen in Fig. 2, ranging from binary to near-lossless 6-bit quantization.

Another difference from traditional GNSS arises in the code sampling process. When generating the local code replica for GNSS signals the SDR is essentially sampling the infinite sinc-squared power spectrum of binary sequences. If the sampling rate for the code replica is at least that of the mainlobe of the sinc-squared spectrum, then the distortion of the code replica due to aliasing of the sidelobes is minimal. Hence, a logical choice for GNSS code replicas is to skip the alias-filtering operation usually implemented when downsampling a signal.

When generating the Starlink code replica however, the SDR is sampling a signal with a flat 240 MHz bandwidth power spectrum. If sampling at less than the full rate, this leaves high-power spectrum outside the sampling bandwidth that, if not filtered out before sampling, aliases into the code

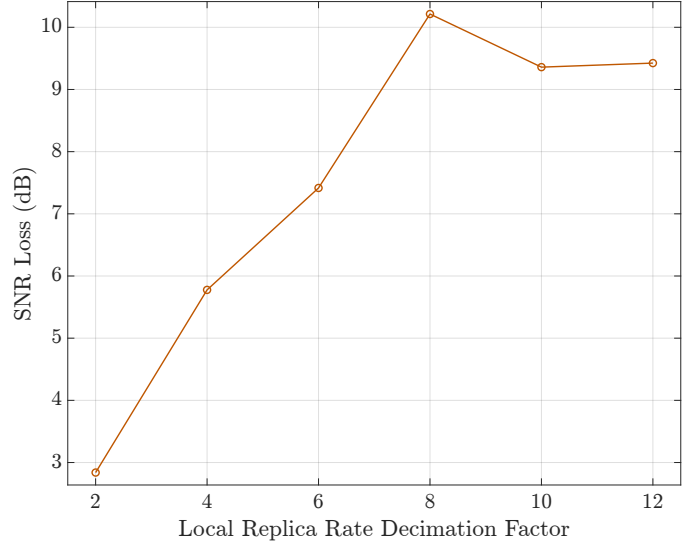


Fig. 3: Post-correlation SNR loss as a function of local replica rate decimation factor. This plot was generated by correlating filtered and unfiltered code replicas against a primary beam's frame from captured data and differencing the SNR. Decimation factors higher than 8 do not result in increased SNR losses, as the correlation peak is roughly the same magnitude as the noise floor of the frame.

replica. To maximize SNR it would be ideal to apply a matched filter to the code replica, mimicking the filter in the RF frontend used to capture the incoming data. The SNR losses when not using an anti-aliasing filter in the code replica generation is portrayed in Fig. 3 for several decimation factors of the full 240 MHz bandwidth.

If the sampling rate is constant throughout a data capture, then the filtering of the local replica need only be done during the SDR setup phase, which does not affect its run-time performance. Since the code for Starlink is not periodically concatenated in the signal, as is the case for GNSS signals, some distortion at the edges will be introduced by filtering, but a non-causal forward-backward filter should minimize such distortions and ensure the filter does not impart a delay to the code replica that needs to be accounted for in downstream processing operations.

B. Starlink Timing Properties

Traditional GNSS signals are generated by high-quality clocks onboard the satellites and the carrier is phase locked to the code at the time of transmission, only brought out of lock by ionospheric code-carrier divergence, which can be modeled and accounted for in receivers. By contrast, Starlink signals are generated by TCXO-quality oscillators and exhibit less well-behaved timing — three separate clocks are apparent in the signal.

As discussed in Section II, the Starlink signal is not continuously transmitting. The first clock, titled the frame clock, governs the transmission times of frames. Its somewhat erratic timing properties have been detailed in [29] and include oscillatory modes, large excursions, and adjustments on a 1-Hz cadence. Understanding the behavior of the frame clock and being able to track it are crucial for real-time Starlink SDR operations, since tracking its offset is much like tracking the code-phase of traditional GNSS signals.

The second clock is the carrier clock, driving the generation of the carrier signal used to modulate the baseband OFDM waveform. This clock is well-behaved in the short-term, but, like the frame clock, can also exhibit 1-Hz adjustments. The carrier clock is only observable when a transmitted frame is received and the carrier phase is often discontinuous between subsequent frames, making carrier phase tracking challenging. After accounting for the frequency offset due to Doppler and barring 1-Hz adjustments, the carrier clock exhibits TCXO-quality timing properties.

The third clock is the OFDM symbol clock, governing the start times of OFDM symbols within a frame. Offsets in this clock's frequency have the effect of dilating or compressing the frame duration. Preliminary analysis indicates it seems to be short-term stable over subsequent frames.

Expanding on the model presented in [29], Fig. 4 shows the three clocks and their outputs t_c , t_f , and t_s . An underlying base oscillator, disciplined to GNSS time by the correction signal s_d , is modeled, whose output t_d drives all three clocks. The frame and symbol clocks might be software-based clocks that depend on the underlying OFDM symbol processing.

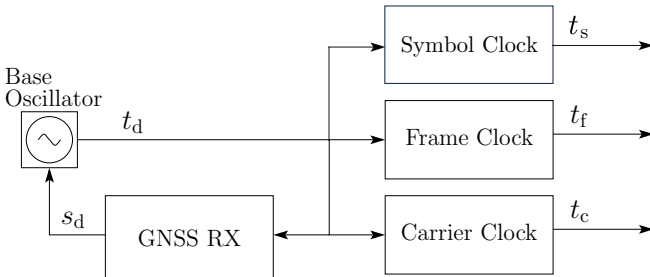


Fig. 4: Three clocks are apparent in the Starlink signal: the carrier, frame, and symbol clocks. This model shows a base GNSS disciplined oscillator driving the three clocks.

C. Tracking Loops

In the foregoing subsection the Starlink signal is modeled with three clocks. To unlock the signal's full PNT potential,

all three offsets of these clocks should be tracked in a receiver. Let $F_{c,rx}$, $F_{f,rx}$, and $F_{sym,rx}$ be the received carrier, frame slot, and in-frame OFDM symbol frequencies and let β , β_f , and β_s be their offsets from the nominal frequencies. The received frequencies are related to the frequency offsets as follows.

$$F_{c,rx} = \frac{F_c}{1 + \beta} \approx F_c(1 - \beta) \quad (1)$$

$$F_{f,rx} = \frac{F_f}{1 + \beta_f} \approx F_f(1 - \beta_f) \quad (2)$$

$$F_{sym,rx} = \frac{F_{sym}}{1 + \beta_s} \approx F_{sym}(1 - \beta_s) \quad (3)$$

Note that the carrier frequency offset β can also be thought of as the negative normalized Doppler frequency.

$$\beta = -\frac{F_D}{F_c} \quad (4)$$

When using as code replica only the synchronization sequences at the start of the frame it is not necessary to track the symbol frequency offset β_s , since the time shift within the 8.8 μs period that the sequences span is negligible. However, for code replicas that span a large fraction of the Starlink frame, tracking the symbol frequency offset β_s would be beneficial.

Unlike GRID's GNSS tracking structure, which couples a phase locked loop (PLL) and a delay locked loop (DLL), LEONARD implements a frequency locked loop (FLL) for tracking β and DLL for tracking β_f . For traditional GNSS the carrier phase of the signal is continuous across code boundaries, but the Starlink carrier phase is not continuous across subsequent frames, limiting the coherent integration to one frame period. The coherent integrating over the known PSS and SSS symbols provide noisy phase and frequency estimates at the start of each transmitted frame. These estimates then are propagated to the next transmitted frame start, often multiple frame slots in the future, making phase tracking impossible due to inevitable cycle slipping. In contrast an FLL, once locked, is much more robust to frequency changes over empty slot frames: The large frequency tracking null-to-null mainlobe beamwidth, about 57 kHz [30], corresponds to tens of seconds of Doppler change for a LEO satellite at zenith.

Tracking the frame frequency offset β_f with a DLL is challenging for Starlink signals, given the large timing excursions and frequent adjustments to the frame clock. While a single clock is used to generate both the carrier and code in traditional GNSS, Starlink's clocks for these evolve independently, meaning the carrier- and code-tracking systems, the FLL and DLL respectively, cannot be coupled. The correlation peak tracked by the DLL is one IQ sampling period wide and the DLL must remain locked to that peak during tracking. If the timing deviation from one transmitted frame to the next is greater than the sampling period, then a DLL with an early-minus-late architecture is likely to slip off of the peak. Consider sampling the full 240 MHz bandwidth of the Starlink channel: The correlation peak would be only 4.17 ns wide. During large oscillations and excursions in the frame clock the DLL is in danger of slipping off the peak. Large swaths of empty frame slots exacerbate the danger. This makes

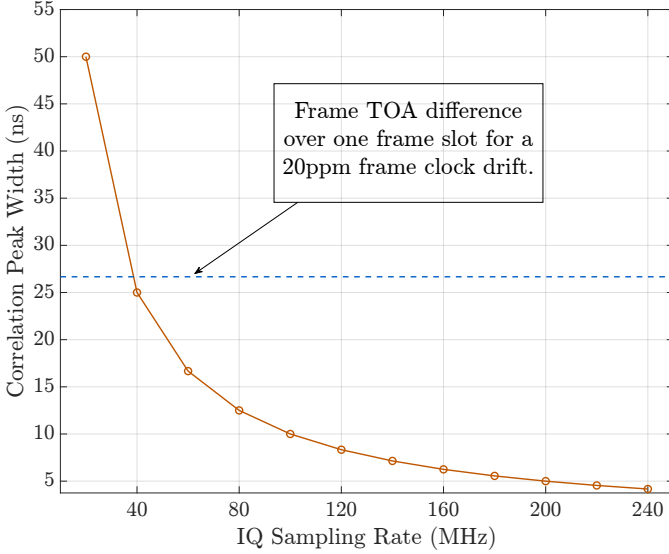


Fig. 5: Correlation peak width in nanoseconds vs. the received IQ sampling rate in MHz. The DLL within LEONARD must be able to set early and late code replica taps on the correlation peak to track the signal, without slipping off the peak due to instability in the frame clock.

tracking the code phase at high sampling rates problematic. Lower sampling rates do not stress the DLL as much, but result in worse TOA measurements than the high sampling rates. In Fig. 5 the correlation peak widths for several sampling rates are illustrated.

Initializing the frame clock DLL also requires care. Since the frame and carrier clocks are decoupled, the frequency estimate of acquiring one Starlink frame alone cannot be used to initiate the DLL. Consider a frame rate that differs from the nominal 750 Hz by 20 ppm. The next transmitted frame start, with the optimistic assumption of occupying the next frame slot, would differ from the nominal frame start by 27 ns! Clearly this would throw the DLL off its tracking peak, especially if there are empty frame slots before the next transmitted frame. Hence, initialization of the DLL has to occur with a β_f parameter estimated from at least two acquired frames, such that the DLL can predict the next frame start time with greater accuracy.

To successfully track the Starlink signals with a DLL one must trade off some TOA accuracy (sampling rate) for increased DLL stability, or move to more sophisticated DLL architectures as in [31], where the authors implement an adaptive DLL architecture, using up to 400 taps to track the frame clock's offset. Given the tradeoff of correlation-peak-width to DLL tracking stability, one might want to implement a super-resolution TOA technique akin to the one used in [9].

IV. PRELIMINARY TRACKING RESULTS

In this section we show preliminary results with 0.5 seconds of real-world data. The 16-bit quantized RF data were captured using a high-gain parabolic dish at a 55 MHz IQ rate. Figure 6 shows the correlation of the Doppler-adjusted PSS+SSS code replica with the first few frame slots in the capture. We

attempted to track the primary beam for the duration of the capture. Within data captures the primary beam often has a roughly repeating pattern of transmitted frames, which is 2-3 occupied frame slots separated by 4-5 empty frame slots in this case.

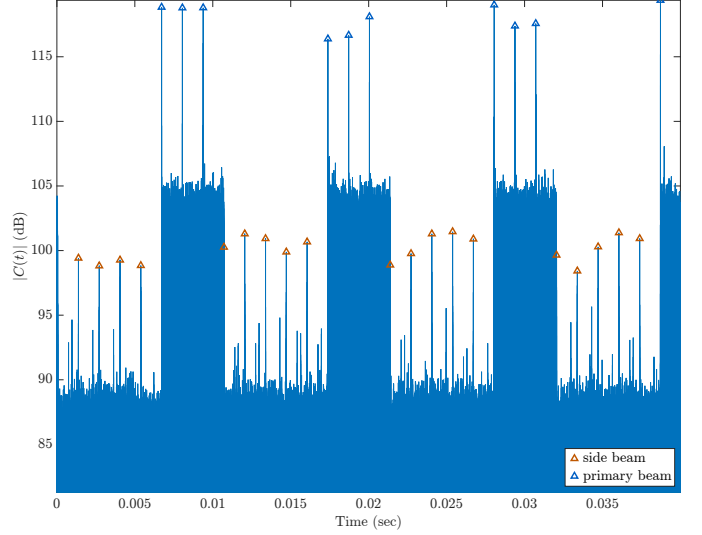


Fig. 6: MATLAB correlation of the PSS+SSS local code replica with the first frame slots in the captured data. Multiple correlation peak trains (beams) are visible and the primary beam and dominant side beam are marked.

For the reasons explained in Section III, it was unclear if the early-minus-late DLL architecture inherited from GRID would be sufficient to track the erratic behavior of the Starlink frame clock. To test this, the FLL and DLL within LEONARD were initialized with β and β_f values corresponding to the primary beam, found from a comprehensive analysis tool written in MATLAB. Further, since the frequency is quasi-stationary over a half-second interval, the FLL tracking was turned off, isolating the DLL as the only active component during the tracking.

Unfortunately, a second-order canonical early-minus-late DLL does not seem to be able to cope with the Starlink frame clock jitter, even with capture-specific tailoring of the DLL bandwidth. Figure 7 shows the prompt correlator output over the data period. As the DLL struggles to keep locked to the correlation peak the magnitude of the prompt correlation sways until the DLL slips off of the peak around the 0.35 second mark.

A more encouraging result is the time it took to process the data—the 0.5 seconds of tracking processed in 0.18 seconds, showing that LEONARD can support the RF operations necessary for real-time tracking of signals at a 55+ MHz IQ sampling rate.

Development of LEONARD will continue. We plan to develop a robust DLL able to track the frame clock offset, create an acquisition routine, implement super-resolution TOA techniques, and ultimately generate real-time PNT solutions.

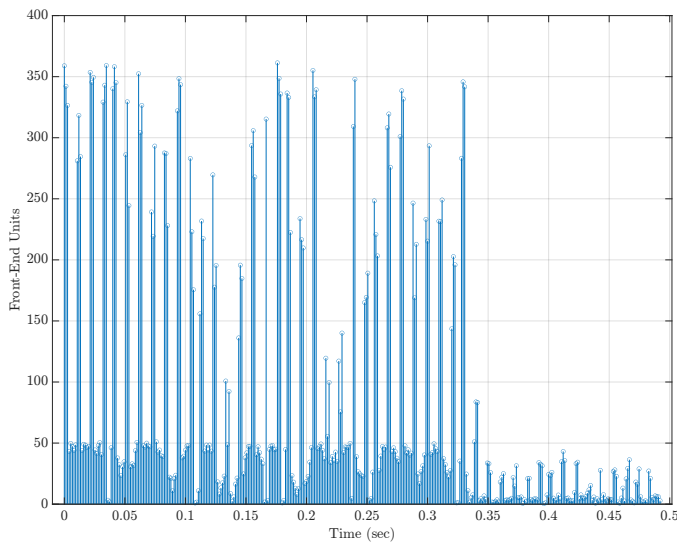


Fig. 7: Output of the prompt correlation tap for each frame slot within the 0.5 seconds of data. The magnitude of the prompt correlation varies as the frame clock DLL struggles to stay locked onto the correlation peak, ultimately slipping off of the main peak around 0.35 seconds.

REFERENCES

- [1] T. G. Reid, A. M. Neish, T. Walter, and P. K. Enge, "Broadband LEO constellations for navigation," *NAVIGATION*, vol. 65, no. 2, pp. 205–220, 2018.
- [2] N. Jardak and Q. Jault, "The potential of LEO satellite-based opportunistic navigation for high dynamic applications," *Sensors*, vol. 22, no. 7, p. 2541, 2022.
- [3] T. E. Humphreys, "Interference," in *Springer Handbook of Global Navigation Satellite Systems*. Springer International Publishing, 2017, pp. 469–503.
- [4] P. A. Iannucci and T. E. Humphreys, "Fused low-Earth-orbit GNSS," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2022.
- [5] T. E. Humphreys, P. A. Iannucci, Z. M. Komodromos, and A. M. Graff, "Signal structure of the Starlink Ku-band downlink," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–16, 2023.
- [6] W. Stock, R. T. Schwarz, C. A. Hofmann, and A. Knopp, "Survey on opportunistic PNT with signals from LEO communication satellites," *IEEE Communications Surveys & Tutorials*, 2024.
- [7] S. Kozhaya, S. Joe, and Z. M. Kassas, "Unveiling Starlink for PNT," *NAVIGATION*, vol. 72, no. 1, 2025.
- [8] S. Shahcheraghi, J. Saroufim, and Z. M. Kassas, "Acquisition, Doppler tracking, and differential LEO-aided IMU navigation with uncooperative satellites," *IEEE Transactions on Aerospace and Electronic Systems*, 2025.
- [9] W. Qin, Z. M. Komodromos, S. C. Morgan, and T. E. Humphreys, "Maximum likelihood time of arrival and Doppler estimation for precise Starlink-based PNT," in *Proceedings of the IEEE/ION PLANS Meeting*, Salt Lake City, UT, 2025.
- [10] S. C. Morgan, Z. M. Komodromos, W. Qin, Z. L. Clements, A. M. Graff, W. J. Morrison, and T. E. Humphreys, "A mock implementation of fused LEO GNSS," in *Proceedings of the IEEE/ION PLANS Meeting*, Salt Lake City, UT, 2025.
- [11] Z. M. Komodromos, S. C. Morgan, Z. L. Clements, W. Qin, W. J. Morrison, and T. E. Humphreys, "Network-aided pseudorange-based LEO PNT from OneWeb," in *Proceedings of the IEEE/ION PLANS Meeting*, Salt Lake City, UT, 2025.
- [12] W. Van Uytsel, T. Janssen, M. Weyn, and R. Berkmans, "A technical overview of current "new space" LEO-PNT initiatives and their application potential," in *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2024, pp. 1–6.
- [13] E. Rubinov, "FrontierSI State of the Market Report LEO PNT 2024 Edition," Jan. 2025, <https://frontiersi.com.au/wp-content/uploads/2025/01/FrontierSI-State-of-Market-Report-LEO-PNT-2024-Edition-v1.1.pdf>.
- [14] N. S. Miller, J. T. Koza, S. C. Morgan, S. M. Martin, A. Neish, R. Grayson, and T. Reid, "SNAP: A Xona Space Systems and GPS software-defined receiver," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2023, pp. 897–904.
- [15] P. A. Iannucci and T. E. Humphreys, "Economical fused LEO GNSS," in *Proceedings of the IEEE/ION PLANS Meeting*, 2020.
- [16] A. M. Graff and T. E. Humphreys, "Ziv-Zakai-optimal OFDM resource allocation for time-of-arrival estimation," *IEEE Transactions on Wireless Communications*, vol. 24, no. 8, pp. 6886–6901, 2025.
- [17] —, "OFDM-based positioning with unknown data payloads: Bounds and applications to LEO PNT," *IEEE Transactions on Wireless Communications*, 2025, submitted for review.
- [18] Z. M. Kassas, N. Khairallah, and S. Kozhaya, "Ad astra: Simultaneous tracking and navigation with megaconstellation LEO satellites," *IEEE Aerospace and Electronic Systems Magazine*, 2024.
- [19] J. Saroufim and Z. M. Kassas, "Ephemeris and timing error disambiguation enabling precise LEO PNT," *IEEE Transactions on Aerospace and Electronic Systems*, 2025.
- [20] S. Zhou, R. Yang, Y. Li, X. Zhan, and H. Qin, "Iridium TOA estimation and positioning based on carrier tracking and beam decoding," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–23, 2025.
- [21] Z. Tan, H. Qin, L. Cong, and C. Zhao, "Positioning using IRIDIUM satellite signals of opportunity in weak signal environment," *Electronics*, vol. 9, no. 1, p. 37, 2019.
- [22] M. L. Psiaki, "Navigation using carrier Doppler shift from a LEO constellation: TRANSIT on steroids," *NAVIGATION*, vol. 68, no. 3, pp. 621–641, 2021.
- [23] M. Neinavaie, J. Khalife, and Z. M. Kassas, "Acquisition, Doppler tracking, and positioning with Starlink LEO satellites: First results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 2606–2610, 2022.
- [24] B. McLemore and M. L. Psiaki, "Navigation using Doppler shift from LEO constellations and INS data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, pp. 4295–4314, 2022.
- [25] N. Jardak, R. Adam, and Q. Jault, "Leveraging multi-LEO satellite signals for opportunistic positioning," *IEEE Access*, 2024.
- [26] Z. M. Komodromos, Z. L. Clements, and T. E. Humphreys, "Signal parameter estimation and demodulation of the OneWeb Ku-Band downlink," 2025, in preparation.
- [27] T. Pany, D. Akos, J. Arribas, M. Z. H. Bhuiyan, P. Closas, F. Dovis, I. Fernandez-Hernandez, C. Fernández-Prades, S. Gunawardena, T. Humphreys, Z. M. Kassas, J. A. L. Salcedo, M. Nicola, M. L. Psiaki, A. Rügamer, Y.-J. Song, and J.-H. Won, "GNSS software defined radio: History, current developments, and standardization efforts," *NAVIGATION*, vol. 71, no. 1, 2024.
- [28] S. Morgan and T. E. Humphreys, "HOOC-EM: Fast beam sweeping for LEO mega-constellation customer terminals," in *Proceedings of the ION GNSS+ Meeting*, 2024.
- [29] W. Qin, A. M. Graff, Z. L. Clements, Z. M. Komodromos, and T. E. Humphreys, "Timing properties of the Starlink Ku-band downlink," *IEEE Transactions on Aerospace and Electronic Systems*, 2025, submitted for review.
- [30] Z. M. Komodromos, W. Qin, and T. E. Humphreys, "Signal simulator for Starlink Ku-Band downlink," in *Proceedings of the ION GNSS+ Meeting*, 2023, pp. 2798–2812.
- [31] C. Lichtenberger, F. Binder, O. Picchi, F. Menzione, and T. Pany, "Using a block processing discriminator for precise tracking of Starlink signals," in *Proceedings of the 38th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2025)*, 2025.