Near Real-Time Signal Quality Monitoring of GNSS Signals Using a Chip Shape Deformation Metric

THESIS

Nicholas C Echeverry, Second Lieutenant, USAF

AFIT-ENG-MS-20-M-017

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-20-M-017

Near Real-Time Signal Quality Monitoring of GNSS Signals Using a Chip Shape Deformation Metric

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Nicholas C Echeverry, B.S.E.E.

Second Lieutenant, USAF

March 26, 2020

AFIT-ENG-MS-20-M-017

Near Real-Time Signal Quality Monitoring of GNSS Signals Using a Chip Shape Deformation Metric

THESIS

Nicholas C Echeverry, B.S.E.E.
Second Lieutenant, USAF

Committee Membership:

J. Addison Betances, Ph.D
Chair

Sanjeev Gunawardena, Ph.D
Member

Michael A. Temple, Ph.D
Member

Abstract

The Global Navigation Satellite System (GNSS) continues to become deeply embedded within modern civilization, and is depended on for confident, accurate navigation information. Safety of Life (SoL) applications, such as civil aviation, require precision measurements for safe operation. High precision position and timing accuracy is typically achieved using differential processing, however these systems provide limited compensation for distortions caused by multipath or faulty satellite hardware.

Signal Quality Monitoring (SQM) aims to provide confidence in a receiver's Position, Navigation, and Timing (PNT) solution and to offer timely warnings in the event that signal conditions degrade to unsafe levels. Several methods of SQM have been introduced and implemented to augment civilian SoL applications. The methods presented in this document focus on implementing effective SQM using low-cost Commercial Off-the-Shelf (COTS) equipment, a Software Defined Radio (SDR), and a typical software receiver architecture that tracks the Galileo E1C signals and the Global Positioning System (GPS) L1 Coarse-Acquisition (C/A) signals. Techniques are centered on acquiring and discriminating signal chip shapes with a goal of identifying both 1) 'clean' and 2) 'deformed' signals. Performance is measured by developing a metric which clearly differentiates the two. The demonstrated identification method is relevant to the growing significance of SQM for SoL applications while providing benefit for confidently monitoring received GNSS signal integrity without requiring specialized receiver hardware.

To my Wife.

Acknowledgements

I would like to thank Major Addison Betances for his continuous guidance and mentorship throughout this research effort. I am grateful for his provision of an engaging subject matter and direction that made this an enjoyable experience.

I would also like to thank my wife and my dogs for the daily support and company during this time, my brothers and sister for the ongoing contact, and my parents for supporting and directing my education throughout my life.

Nicholas C Echeverry

Table of Contents

Page

List of Figures

List of Tables

Near Real-Time Signal Quality Monitoring of GNSS Signals Using a Chip Shape Deformation Metric

I. Introduction

1.1 Problem Background

The Global Navigation Satellite System (GNSS) is responsible for confidently providing precise Position, Navigation, and Timing (PNT) solutions for users across the globe. Receivers operate by assuming the presence of ideal signals and are easily compromised should deviations arise. Naturally occurring distortions are induced upon signals as a result of multipath interference or malfunctioning satellite hardware. Though the latter is an unlikely occurrence, it is still a threat to PNT solution accuracy and has effects similar to signal multipath, which is a daily hazard. Signal deformations directly impact a receiver's PNT solution accuracy and should be observed such that systems are aware of the current signal conditions. Signal Quality Monitoring (SQM) seeks to provide confidence in a signal's integrity and to raise alarm should signal conditions deteriorate to unsafe levels.

1.1.1 Operational Motivation

As the GNSS continuously becomes more deeply embedded within society, it is increasingly relied upon for confidently providing precise ranging and timing information. Many civilian applications that require rigorous timing synchronization across distributed systems depend on GNSS. This includes network synchronization in communication systems, phase synchronization in electrical power grids, and time stamp-

ing in high-frequency financial transactions [2]. Precise ranging is especially required for Safety of Life (SoL) applications, with the leading example of civilian aviation. Significant positioning errors directly threaten these users' safety, and are thus intolerable. The U.S. National Space Policy [3] expresses commitment to "invest in domestic capabilities and support international activities to detect, mitigate, and increase resiliency to harmful interference to GPS." As to be discussed, several methods to achieve effective SQM exist and have been implemented for these systems. It is desired to explore alternative methods of SQM using inexpensive, Commercial Off-the-Shelf (COTS) equipment, to allow for broad application.

1.1.2 Technical Motivation

This research is prompted by the growing studies focused on Radio Frequency Distinct Native Attributes (RF-DNA) [4, 5, 6, 7]. These studies center on studying systems at the physical layer, most often by analyzing the received signal's waveform. Machine-learning algorithms have been shown to effectively classify transmission devices by the deformations they impart on signals, such that deviations from the nominal distortions may be noticed.

GNSS signals at the Earth's surface are too weak to be observed in their raw form, inspiring research centered on the chip shape domain. Introduced as the Vision Correlator (VC) by NovAtel [8, 9], like-portions of the signal are integrated to produce an averaged representation of the signal's character. Though this research does not use machine-learning to analyze GNSS signals, the chip shape domain is used to distinguish distorted signals from their nominal form.

1.2 Research Objectives

The goal of this research is to implement a functioning software GNSS receiver that is capable of monitoring signal integrity by exploiting the observable deformations in the chip shape domain.

The goal of the receiver software is to operate with a Software Defined Radio (SDR), and acquire and track GNSS signals in real-time. The purpose of tracking is to continuously compute the state of a signal, specifically, the Pseudo-Random Noise (PRN) code offset, carrier frequency offset, and carrier phase offset. For this research, the objective is to achieve carrier phase tracking and determine the signal polarity.

SQM is to be achieved by coherently integrating repeating signal transition regions to reveal an averaged waveform response called chip shape [10]. The proposed method is accomplished using parameters that are computed by signal tracking. Chip shape acquisition has the purpose of producing an in-phase average of the four possible signal transitions. Once acquired, the goal is to effectively draw conclusions on signal quality.

1.3 Document Overview

The GNSS structure and signal design, as well as research focused on SQM and signal chip shapes, will be described in Chapter II. Chapter III will detail the software receiver architecture used to collect data, track signals, and acquire and analyze signal chip shapes. Lastly, Chapter IV will exhibit the system's performance, with emphasis on how well distorted signals are able to be identified.

II. Background and Literature Review

This chapter outlines the Global Navigation Satellite System (GNSS), the need for effective Signal Quality Monitoring (SQM) techniques, and current approaches to fulfilling this need. As this thesis presents a software receiver that computes chip shapes in real-time as a means of SQM, literature regarding chip shapes will also be discussed.

2.1 Overview of GNSS

The GNSS provides users across the globe with access to accurate Position, Navigation, and Timing (PNT) solutions. The system is outlined with emphasis on the user's perspective.

2.1.1 Segment Overview

The GNSS consists of a constellation of satellites that provide timing and positioning information through the transmission of Radio Frequency (RF) signals. As described in [11], the system performance is measured by the following metrics:

- Accuracy - How close a receiver's solved PNT solution is to the true location and time.
- Integrity - The confidence that a transmitted signal is almost ideal and that a satellite will send warning should its transmission hardware malfunction.
- Continuity - The ability of the system to operate without interruption.
- Availability - How often the system meets all above criteria, lending itself useful to the users.

To provide quality service, the system must be constantly monitored and maintained. The operations of the United States' Global Positioning System (GPS) will be described here, but are also applicable to the functionality of all satellite navigation systems.

The system is composed of three unique segments (described by the GPS government website [12]): space, control, and user. The space segment consists of the actual satellite constellation which transmits navigation messages to the users, giving parameters that define the position of the Satellite Vehicles. For GPS, there are nominally 24 SVs in Medium Earth Orbit (MEO) (approximately 20,200km above Earth's surface), yet 31 are currently in place to maintain this standard [12]. The system is passive for the users, as the SVs only transmit signals which give the users everything they need to compute a PNT solution. This structure allows for the simultaneous use of satellite navigation for an unlimited amount of users [13].

The control segment exists to monitor the system, ensure that navigation messages are accurate, and shift SVs into proper orbital positions, such that errors induced by the space segment are minimal [13]. Maintenance is accomplished through a global network of ground facilities that analyze transmissions and track the SVs. Additionally, clock updates, ephemeris data, and almanac data are generated and uploaded daily to the constellation for accurate positioning information [13].

Finally there is the user segment, which is the setting for this thesis: GPS receivers are able to acquire and track the signals in order to measure pseudoranges and compute a PNT solution. The hardware includes an antenna that is designed to receive Right-Hand Circularly Polarized (RHCP) RF waves, typically using an active low-noise amplifier to lower the overall noise figure. From here, front ends must implement a bandpass filter centered near the nominal frequency, a preamplifier, downconversion to some Intermediate Frequency (IF), and digitization/quantization

(ie, Analog to Digital (A/D) conversion) [13]. Next, measurements are computed and the navigation message is decoded through the use of signal processing. The goal of the user segment is to achieve the most accurate PNT fix possible, and to realize when confidence in the solution is unacceptably low.

2.1.2 System Error Sources

As PNT solution accuracy is critical to GNSS performance, error sources have been closely studied to obtain better understanding of the system and to develop error mitigation methods.

2.1.2.1 Ionospheric Effects

The ionosphere begins roughly 50km above the Earth's surface, extending to the equivalent of several Earth radii. This portion of the atmosphere is classified by the generation of charged ions and electrons by incident solar radiation [13]. As the Sun is responsible for this section of the atmosphere, ionospheric activity is directly impacted by the solar cycle. This activity is significant to GNSS as the ionosphere directly affects RF waves, with higher ionospheric activity having more impact [14].

Relatively minor effects from the ionosphere include amplitude and phase scintillation, induced Doppler shift, and signal refraction. The most significant effect on the GNSS system is the Pseudo-Random Noise (PRN) code delay and carrier phase advance on the signal [14]. This is detrimental to the PNT solution, as pseudoranges and carrier phase measurements are altered differently for each SV, directly dependent on how long the signal propagates through the ionosphere.

Fortunately, ionospheric delay is frequency-dependent, allowing ionospheric error to be virtually removed by using a dual-frequency receiver. A majority of SVs in the GNSS constellation have yet to transmit two civilian signals on different frequencies,

so this solution is not yet universal. Several models have been introduced to be used by single-frequency users, by scaling the delay based on the elevation angle with knowledge of the known delay computed at the zenith angle [14, 13]. For most applications, however, augmentation systems are used to estimate effects for users in some localized area.

2.1.2.2 Tropospheric Effects

The troposphere composes the lower part of the atmosphere (below the ionosphere), and similarly causes RF signal delays dependent on how long the signal propagates through it. Unlike the ionosphere, tropospheric delays are caused by the refractive index of the atmosphere and act equally on different frequencies [13], removing the possibility of dual-frequency correction. As GNSS pseudoranges are computed under the assumption of free-space propagation, the slowed signal speed will directly impact PNT solutions.

About 90% of tropospheric delay is a result of the dry atmosphere, which is very predictable and can be modeled precisely. The wet component, which contributes only a small fraction to the signal delay, is composed of water vapor and is much less predictable due to varying atmospheric conditions [14, 13]. Like the ionosphere, models have been developed to remove tropospheric delay, mapping the delay at zenith angle to the elevation angle with respect to the user, but requiring input of atmospheric conditions to be accurate. Also like the ionosphere, these atmospheric delays may be corrected using augmentation systems.

2.1.2.3 Ephemeris Error

Ephemeris parameters uplinked to satellites include best curve-fit predictions of satellite positions, with residuals ranging typically from 1-6 meters [14, 13]. Though

inaccuracies directly impact pseudorange and carrier range measurements, measurement error is statistically negligible when compared to other error sources.

2.1.2.4 Receiver Noise

The receiver front-end introduces its own noise to the system, taking the form of a zero-mean gaussian probability density function due to thermal electron movement in the hardware. Quantization also introduces irreversible error, dependent on the precision of the system [14]. Most significantly, noise is independent to each receiver and resulting errors may not be removed through differential processing. In most applications, position error due to receiver noise is statistically negligible with respect to the presence of other error sources.

2.1.2.5 Multipath Error

Signal multipath, detailed more in section 2.4.1.2, introduces errors that also directly impact the PNT solution. Like receiver noise, multipath error is independent to each receiver, as it relies on the antenna's surroundings. Therefore, the error is unable to be removed by augmentation systems. The error introduced by multipath is statistically more significant than that caused by noise, making this the dominant error source in most GNSS systems [13]. This threat is difficult to model, as rapidly changing environments are unpredictable. Careful site selection, specialized antenna designs, such as choke rings, and receiver techniques such as correlator gating are often used to mitigate multipath effects [15, 16].

2.1.3 Differential Processing

Differential processing systems work in conjunction with GNSS to improve solution accuracy, as many civilian applications (such as civil air) require high precision PNT

solutions to operate. The systems operate on the principle of applying a differential solution: by using a reference station, corrections are sent to mobile receivers over a data link. These corrections take the form of measurement corrections, positional corrections, or signal integrity data [13]. Implemented systems address the needs of their application, with the most precise corrections applying to localized areas, and the less precise applying to large areas.

Typically using a single reference station, Local Area Augmentation Systems cover tens of kilometers, achieving PNT solution accuracy on the centimeter level [14]. This precision is sought after in Safety of Life (SoL) applications, using the proximity of mobile receivers and some reference station to effectively remove common errors, such as those induced by the atmosphere.

Giving corrections slightly less precise than LAAS's, a Regional Area Augmentation System (RAAS) covers an area on the scale of hundreds of kilometers, involving multiple reference stations and achieving accuracy on the decimeter level [14].

Lastly, a Wide Area Augmentation System (WAAS) covers thousands of kilometers and achieves accuracy of about a meter [14]. This level of accuracy is applicable to general applications, but also for sensitive applications such as civil aviation mid-flight, with reference stations estimating measurement errors for mobile users across the vast area. This type of augmentation system is implemented as a Satellite-Based Augmentation System (SBAS), using geo-stationary satellites to transmit the differential corrections and integrity data to the wide area. WAAS refers to the U.S. implementation, with other countries implementing their own, such as Europe's European Geostationary Navigation Overlay Service (EGNOS).

2.2 GNSS Signal Structure - GPS L1 Course-Acquisition (C/A) and Galileo E1

The signal structure is defined by publicly-available Interface Control Documents (ICDs) [17, 18], enabling PNT solutions for civilian applications.

2.2.1 Signal Modulation Overview

Signal modulation involves encoding digital symbols on a carrier wave while serving the needs of the system. For GNSS, these needs include transmitting a navigation message, providing accurate timing information, and allowing a very weak signal to be detected.

2.2.1.1 Spread-Spectrum Satellite Signals

The GNSS signal structures in this paper are characterized by their use of Direct Sequence Spread Spectrum (DSSS); a high-speed PRN code is linearly modulated as a sequence of pulses onto a signal. "Pseudo-Random" refers to the code's appearance and traits being random in nature, yet entirely deterministic. This design provides many benefits essential to the system. Most directly, it widens the signal spectrum's bandwidth considerably. Wider bandwidth allows for high resolution Time-of-Arrival (TOA) measurements, which are a necessity for computing accurate PNT solutions. Additionally, it provides significant rejection of interference, both by potential jammers that operate without knowledge of the signal structure and by other DSSS signals operating on the same frequency band [19].

The signals discussed here, both the GPS L1 C/A signal and the Galileo E1 signal, make use of the latter benefit to enable multiple signals with different PRN codes to be received at the same frequency. This structure is known as Code-Division Multiple Access (CDMA) and is possible due to the different codes being quasi-orthogonal to

each other, yielding very low interference in the codes' cross-correlation.

Though CDMA is enabled through the use of DSSS, the spectrum spreading process makes signal reception more complex due to the lowered power density on an already-weak signal and the larger bandwidth that must be observed [19]. Details on receiver processing will be included in the methodology.

2.2.1.2 GNSS Signal Modulation Schemes

The signals described here make use of Binary Phase Shift Keying (BPSK) modulation to implement DSSS, as well as any data bits to be transmitted. This digital signal scheme encodes bits, i.e. digital 1's and 0's, by shifting the signal carrier phase between the two states of 0 and π radians [13, 20]. When the signal is at baseband, i.e., its frequency response is centered about $f_c = 0$Hz, the BPSK modulation is simply seen as a $\cos(0) = 1$ or $\cos(\pi) = -1$, for binary 0 or binary 1, respectively.

A relatively recent augmentation to the standard BPSK scheme is seen in the use of Binary Offset Carrier (BOC) modulation for some GNSS signals. This scheme is achieved through the modulation of a subcarrier on the signal. In GNSS applications, this subcarrier is a square wave such that precise timing is not impacted; the subcarrier is described in [17] as

$$\mathrm{sc}(t) = \mathrm{sign}\left[\sin(2\pi R_s t)\right], \tag{1}$$

where, sign () denotes the sign of the argument, R_s refers to the rate of the subcarrier wave. The main impact of BOC modulation on the frequency spectrum is the splitting of the spectrum into two main lobes, each offset from the center frequency by the subcarrier rate. The non-centered frequency distribution yields many advantages, one being the possibility of higher pseudorange precision due to the wider bandwidth.

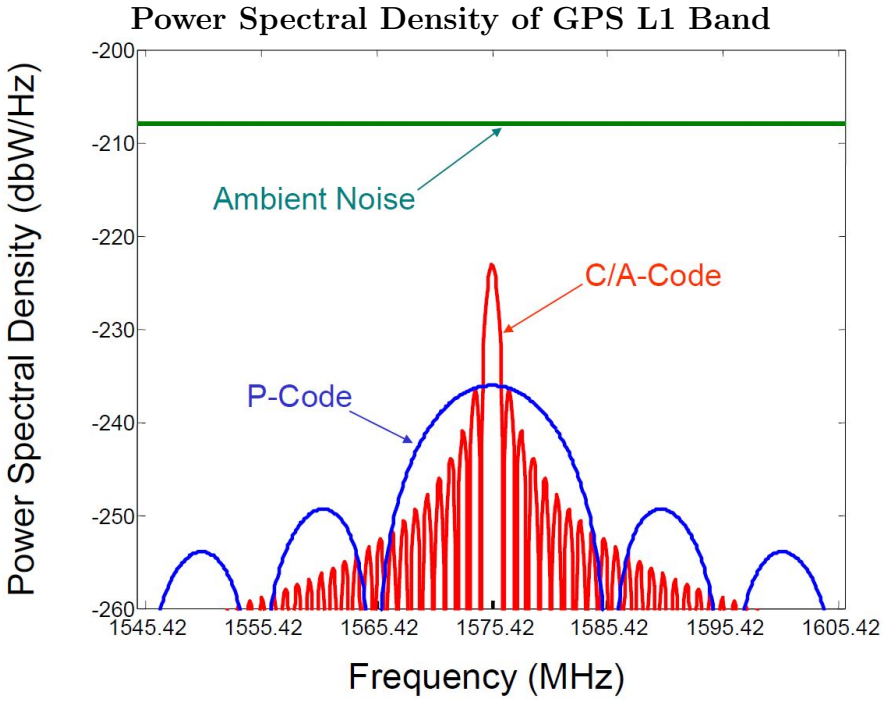


Figure 1: The ideal power spectral densities of the GPS L1 signals which make use of DSSS and how they compare to the thermal noise floor. The lobe widths are directly determined by the chipping rate of the spreading sequence; as the P-Code has a rate that is ten times faster than the C/A-code, the lobes are ten times wider. (Figure provided by [14].

2.2.2 GPS L1 C/A Signal

The GPS L1 band is the legacy satellite navigation RF link consisting of the C/A code and the Precision Code (P(Y)), each in phase quadrature with the other. The latter uses encryption as a form of anti-spoofing, with military specifications unavailable to the public. The C/A code's dual purpose is to assist in acquisition of the P(Y) code, and to enable civilian PNT applications. [18].

The L1 frequency band is centered at 1575.42MHz, with a transmitted bandwidth of 24MHz [17]. The PRN codes used are Gold codes generated by the modulo-2 addition of two subsequences [18]. Each code is 1023 chips long, transmitted at a chipping rate of 1.023 mega-chips per second (MCPS) (making each code period exactly *1ms* long).

The navigation message responsible for giving parameters which define the SV's position is transmitted at 50 bits per second (bps). That is, there are 20 periods of the PRN code per data bit, with each data bit transition synchronized with the relative epoch of the PRN code. The data message structure is detailed in [18].

All of these components describe the GPS signal on the L1-band, and are summarized in [21] by

$$\mathbf{S}_{L1}(t) = A_P \mathbf{XP}_i(t)\mathbf{D}_i(t)\cos(\omega_{L1}t + \phi) + A_C \mathbf{XG}_i(t)\mathbf{D}_i(t)\sin(\omega_{L1}t + \phi). \quad (2)$$

Seen here, there are two components in quadrature-phase with a PRN code and data modulated onto the signal. ω_{L1} refers to the frequency band's angular frequency of $2\pi(1575.42MHz)$, XP_i and XG_i refer to the spreading codes for the P(Y) code and the C/A code, respectively, and D_i refers to the data navigation message.

Because the polarity of the signal is ambiguous when first received, the data message is used to match the received polarity with that which was transmitted. Polarity is determined by the telemetry (TLM) word, which includes an 8-bit Barker word that serves as a preamble [21]. Once this sequence is found, it is known whether the receiver is in-phase or out-of-phase with the transmitted signal. Synchronizing polarity is relied upon such that data bits are not decoded as the opposite polarity, and also such that chip transitions are not confused as the opposite transition.

2.2.3 Galileo E1 Signal

The Galileo E1 band consists of two components: the data component, E1-B, and the pilot component, E1-C. The pilot component does not contain any data bit modulation, making the signal entirely deterministic and allowing for long integration times for high-sensitivity applications. Because the two components are transmitted

together, tracking one component gives all parameters needed to track the other.

Like the GPS L1-band, the Galileo E1 band is centered at 1575.42MHz, but the transmitted signal bandwidth spans a wider range of 32MHz. Each component has a primary PRN code of length 4092 chips, transmitted at a rate of 1.023 MCPS (making each code period exactly *4ms* long) [17]. The longer code length strengthens the cross-correlation performance of the codes. For the data component, there is a navigation message sent at a rate of 250 bps, such that there is one data bit sent per primary code period. For the pilot component, there is a secondary code of length 25 chips sent at 250 chips per second, such that there is likewise one secondary code chip per primary code period. The secondary code on the pilot both verifies the polarity of the signal and artificially lengthens the code such that the tiered code structure has a total code period of 100ms.

Galileo E1 is unique in the use of Composite Binary Offset Carrier (CBOC) modulation; CBOC consists of two BOC modulation schemes with different subcarrier rates, transmitted at different power levels. The Galileo E1 subcarrier rates are 1.023MHz and 6.138MHz, that is, 1- and 6-times the PRN code chipping rate. The power is allocated such that the 1.023MHz subcarrier is given 10/11 of the total power, and the 6.138MHz subcarrier is given the remaining 1/11 of the total power. The two components are combined through addition or subtraction, defined in [17] by

$$\mathbf{sc}_{E1-B}(t) = \sqrt{\frac{10}{11}}\text{sign}(\sin(2\pi(1.023MHz)t)) + \sqrt{\frac{1}{11}}\text{sign}(\sin(2\pi(6.138MHz)t)) \ , \quad (3)$$

$$\mathbf{sc}_{E1-C}(t) = \sqrt{\frac{10}{11}}\text{sign}(\sin(2\pi(1.023MHz)t)) - \sqrt{\frac{1}{11}}\text{sign}(\sin(2\pi(6.138MHz)t)) \ . \quad (4)$$

The above describes the subcarrier for the data and pilot component, respectively. These are different to minimize cross-correlation of the components. By including two BOC modulation schemes, the signal is able to be used in general applications by ignoring the high-frequency subcarrier, or in high-precision applications by taking advantage of it (thus requiring a higher front-end bandwidth).

All above components describe the Galileo E1 signal, summarized by

$$\mathbf{S}_{E1}(t) = \sqrt{\frac{1}{2}}(\mathbf{D}_{E1-B}(t)\mathbf{C}_{E1-B}(t)\mathbf{sc}_{E1-B}(t)\cos(\omega_{E1}t + \phi) \\ + \mathbf{C}_{E1-C}(t)\mathbf{sc}_{E1-C}(t)\sin(\omega_{E1}t + \phi)) \ , \tag{5}$$

where, $\mathbf{D}_{E1-B}$ refers to the data bit modulation, $\mathbf{C}_{E1-x}$ refers to the spreading code (including secondary code overlaid), $\mathbf{sc}_{E1-x}$ refers to (3) and (4), and ω_{E1} is the angular frequency of the band, $2\pi(1575.42MHz)$.

2.3 Alignment of Local Replica with Signal

Since GNSS signals have extremely low signal power once reaching the Earth's surface, the presence of the signal is only verified through correlating the signal with a local replica of the signal for some period of time. Since the replica must be aligned closely in PRN code phase and frequency, this is a two-dimensional alignment problem with different properties on each axis. The results of the in-phase and quadrature-phase dot products are given in [22], by

$$\mathbf{I}_{corr} = \frac{A}{\sqrt{2}} * N * D_i * \mathbf{R}(\tau) * \frac{\sin(\pi(\Delta' f)T_{pdi})}{\pi(\Delta' f)T_{pdi}}\cos(\Delta\phi), \tag{6}$$

$$\mathbf{Q}_{corr} = \frac{A}{\sqrt{2}} * N * D_i * \mathbf{R}(\tau) * \frac{\sin(\pi(\Delta' f)T_{pdi})}{\pi(\Delta' f)T_{pdi}}\sin(\Delta\phi), \tag{7}$$

where the only difference is in the 90-degrees phase offset given by the final term. $\mathbf{R}(\tau)$ refers to the PRN correlation function, and T_{pdi} refers to the amount of time correlation is performed for. The two channels in quadrature-phase are a necessity; as the phase alignment of the carrier changes, the overall magnitude remains constant (that is, the magnitude of $\sqrt{\mathbf{I}^2 + \mathbf{Q}^2}$). As expected, the magnitude is directly proportional to the number of points used in the dot-product, N. The dependencies, as described previously, are the frequency and code offsets, $\Delta' f$ and τ, respectively.

2.3.1 Correlation with Respect to Code Phase

As PRN codes are designed to exhibit random nature, both the correlation between two different codes and the correlation between a misaligned code with itself are very small. Ideally, an infinite sequence of random chips would result in a cross-correlation of zero, but the finite code length of GNSS signals prohibits this possibility. Even with this restriction, the correlation performance of GNSS codes yields a large spike in magnitude when identical codes are aligned, and very small magnitude when the codes are different or misaligned by more than one chip. This property is represented as the auto-correlation function, $\mathbf{R}(\tau)$, in Equations 6 and 7. As described in [22] and illustrated in Figure 2, the auto-correlation function for a period of GPS C/A PRN codes takes on discrete values of 1023, 63, -1, and -65. Correlation performance is improved with longer PRN sequences, such as those used in Galileo E1 signals.

2.3.2 Correlation with Respect to Carrier Frequency

Movement of SVs with respect to users induces a Doppler effect on each signal, creating an unknown carrier frequency offset. As correlation times are on the order of milliseconds, small frequency offsets often have low impact on the correlation magnitude.

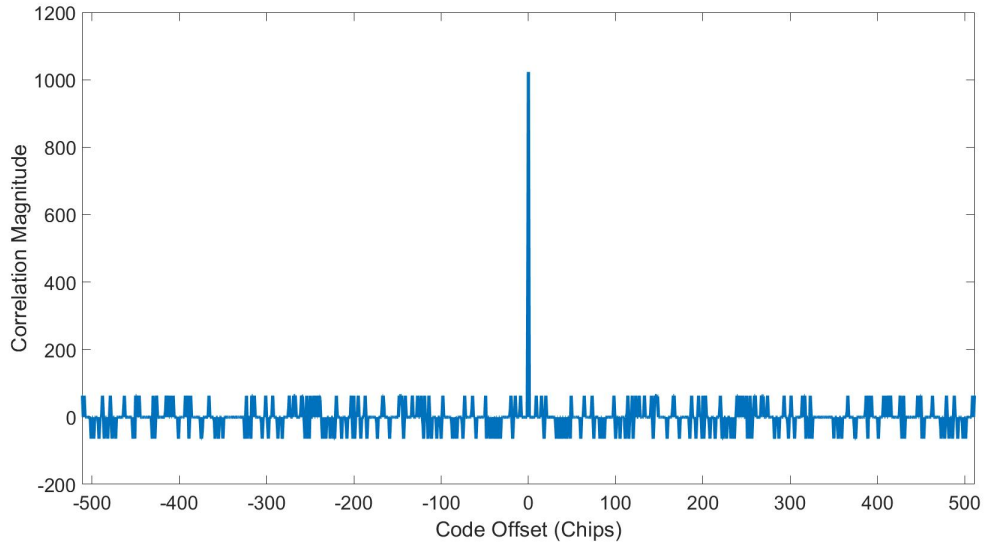


Figure 2: The auto-correlation of a single period of the GPS C/A binary code. That is, there is no frequency offset between two signal replicas and the dot product is computed for every code offset. The correlation is maximized when the codes are exactly aligned, shown more closely in Figure 3.

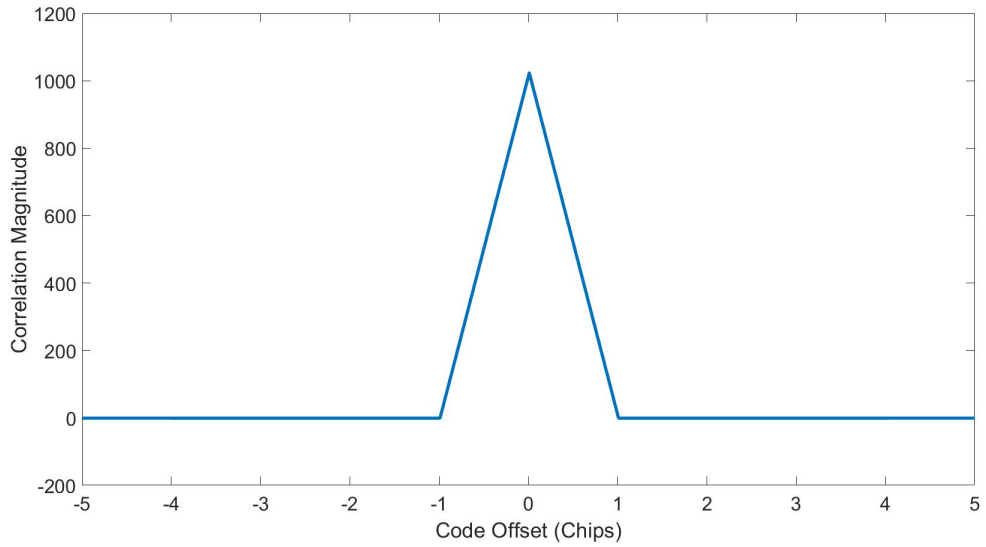


Figure 3: A subset of Figure 2, showing the correlation magnitude for code offsets close to zero. The magnitude grows when code offsets are within one chip, with the peak correlation achieved when signals are precisely aligned.

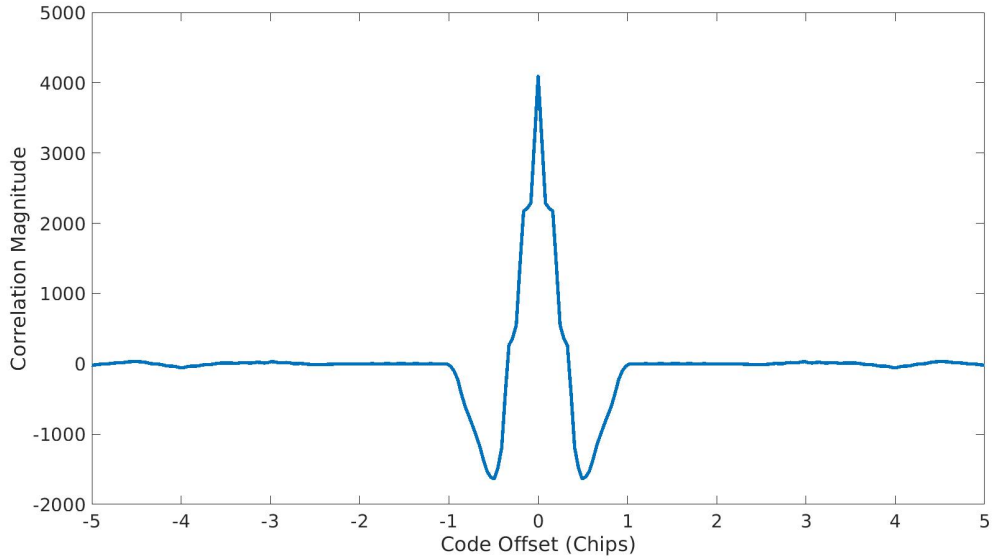


Figure 4: The auto-correlation of a Galileo E1 binary PRN sequence for low code offsets. The shape is a direct result of the subcarriers modulated on top of the PRN sequence as a part of CBOC modulation, and adds tracking benefits as it results in a sharper slope to the peak correlation magnitude.

As seen in (6) and (7), there is a *sinc* function relationship between correlation magnitude and the carrier frequency difference, with the lobe width inversely dependent on the correlation time, T_{pdi}. Correlation times typically range from *1 ms* to *20 ms* [22]; as the correlation time increases, the lobes of the *sinc* relationship grow narrower, demanding the local replica to be closer in frequency.

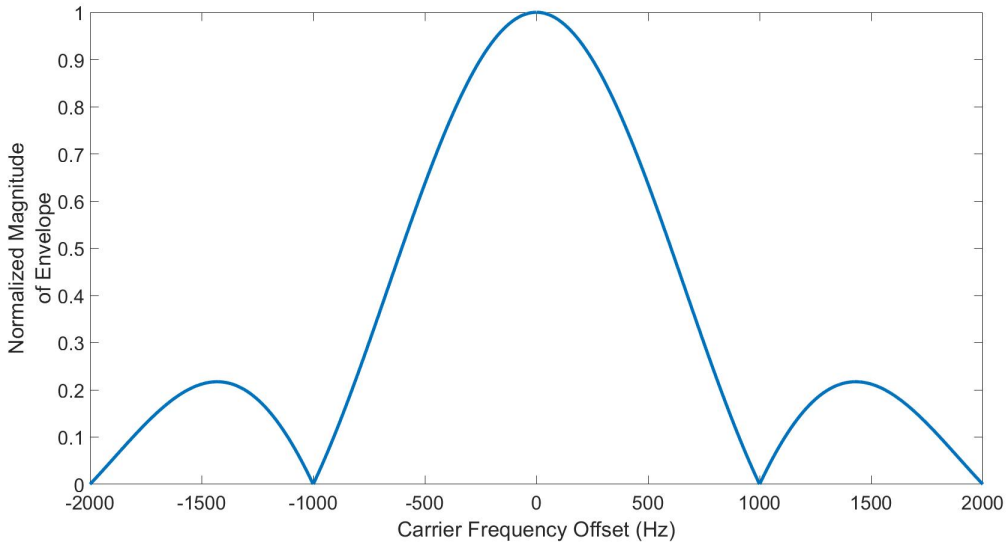


Figure 5: Wideband correlation magnitudes over a range of carrier frequency offsets between two replicas of a GPS C/A signal. Here, *1 ms* is used as the correlation time, directly determining the width of the side lobes as given in (6) and (7). The wide side lobes give this method the name of "wideband correlation" [22].

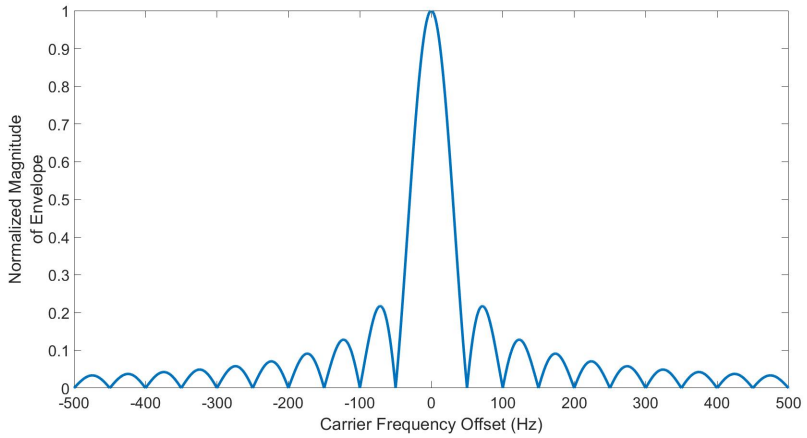


Figure 6: Narrowband correlation magnitudes over a range of carrier frequency offsets between two GPS C/A signals. Here, *20 ms* is used as the correlation time, leading to much narrower side lobes compared to Figure 6. The narrow side lobes give this method the name of "narrowband correlation" [22].

2.4 Signal Quality Monitoring of GNSS Signals

As GNSS signals become rapidly more integrated within society, there is growing dependence on accurate PNT solutions. Distortions to the signal directly impact pseudorange accuracy and introduce errors to the solution. The intent of SQM is to characterize signal integrity and confidently identify when conditions reach unsafe levels.

2.4.1 Causes of Signal Deformation

Signal deformation refers to the waveforms received at an antenna being non-ideal. The two primary causes are due to distortions induced by satellite transmission hardware and multipath.

2.4.1.1 Satellite-Induced Distortions

A GNSS receiver assumes that the signal sent at transmission is ideal, however it has been observed that distortions to the nominal signal are introduced by the satellite's signal generation hardware. Before transmission, digital distortions are introduced to the signal, causing delay (lead or lag) of rising- and falling-edge chip transitions [9, 23]. The signal structure calls for instantaneous changes in phase, which is unachievable by any transmission hardware. As a result, modulator imperfections, filtering, and bandlimiting lead to imperfections that may ultimately cause changes in the width of chips [23]. Analog deformations are also introduced, such as transition overshoot and ringing, that contribute to minor ranging errors in standalone receivers. These effects are more troubling for precision differential systems using dissimilar receivers, as the pseudorange errors may be receiver-dependent and lead to added error in the differential solution [24].

Many studies have sought to characterize satellite-induced distortions and deter-

mine nominal signal behavior, such that anomalies may be easily identified should they arise. Robust classification methods analyze the raw signal of a satellite by collecting data with high-gain parabolic antennas [23, 25]. Models are formed by observing how the received signal deviates from an ideal step response. These methods include characterizing the lead or lag of the signal's zero crossing, the overshoot of the transition, and the damping coefficient which controls the time for the signal to settle on the new level. It has also been observed that distortions in the signal directly cause distortions in receiver processing. As parabolic antennas are often impractical, studies have sought to characterize the distortions seen by processing the correlation function (typically using seven correlators), the varying tracking error for different early-late correlator spacing, or the chip-shape average (also known as the Chip Domain Observable (CDO) or the Vision Correlator (VC) by NovAtel) [8, 9].

In the worst case, satellite hardware may fail and send "unhealthy" signals that greatly deviate from the nominal. The first instance of faulty hardware was first observed in GPS satellite PRN 19 in 1993, with an additional tone added at the nominal frequency [26]. This signal had strong effects on differential GPS systems, causing positioning errors of 3-8 meters. The possibility of satellite failure is especially worrisome for SoL applications such as civil aviation, where such positioning errors are intolerable [27]. Because of this threat, the LAAS at airports that provide aircraft with accurate positioning are also tasked with monitoring signal integrity and quickly providing alerts should a signal be deemed as "unhealthy" [23, 26]. To implement such a system, it is required to have knowledge of the nominal waveform and a way of measuring the received waveform.

2.4.1.2 Signal Multipath Distortions

Signal multipath has been widely studied yet still remains a dominant source of error in satellite navigation systems. During multipath interference, a receiver collects data containing the line-of-sight (LOS) signal and a copy of the signal that has been reflected off of a nearby surface. Since multipath signals must travel further than the LOS signal, they are always delayed, weaker copies of the signal (assuming nothing hinders the LOS signal path), and have unpredictable carrier phase [16]. For long multipath signal delays, receivers tracking the LOS signal are able to resolve the multipath and have positioning solutions unaffected by the presence of the additional signal [13]. The multipath signals reflected off of surfaces nearby the receiver's antenna are of greater concern, as they arrive almost immediately after the LOS signal and produce significant distortions in the processing.

Multipath interference is a major threat to SoL applications. Most precision-navigation applications use some form of differential positioning with a reference receiver (such as an airport's LAAS); as this approach removes errors common to the receivers, it is unable to resolve error due to multipath interference as this is specific to the environment of each receiver [28]. The continued urbanization surrounding these critical applications requires new robust mitigation methods against signal multipath [29], and SQM remains a high priority to ensure unsafe measurements are avoided.

2.4.2 SQM Techniques

The need for SQM has prompted substantial research aimed at identifying signal interference within the receiver architecture. As previously stated, it has been a common practice to study the effects that signal distortions have on the signal processing. The most prominent implementation uses a plurality of correlators instead of the typical three and observes the asymmetries that may be introduced to the correlation

triangle curve [1, 8, 27]. Initially patented by Honeywell [27], the many correlators are implemented with different code offsets (with a maximum of a 1-chip offset) to outline how the signal correlates with an ideal replica. Correlators with code offsets of the same magnitude but opposite direction should theoretically be equal, so the differences in respective correlation pairs can be used to detect distortions should they pass some pre-defined threshold. Asymmetric correlation is illustrated in Figure 8, where multipath interference directly causes deformation to the GPS C/A correlation function.

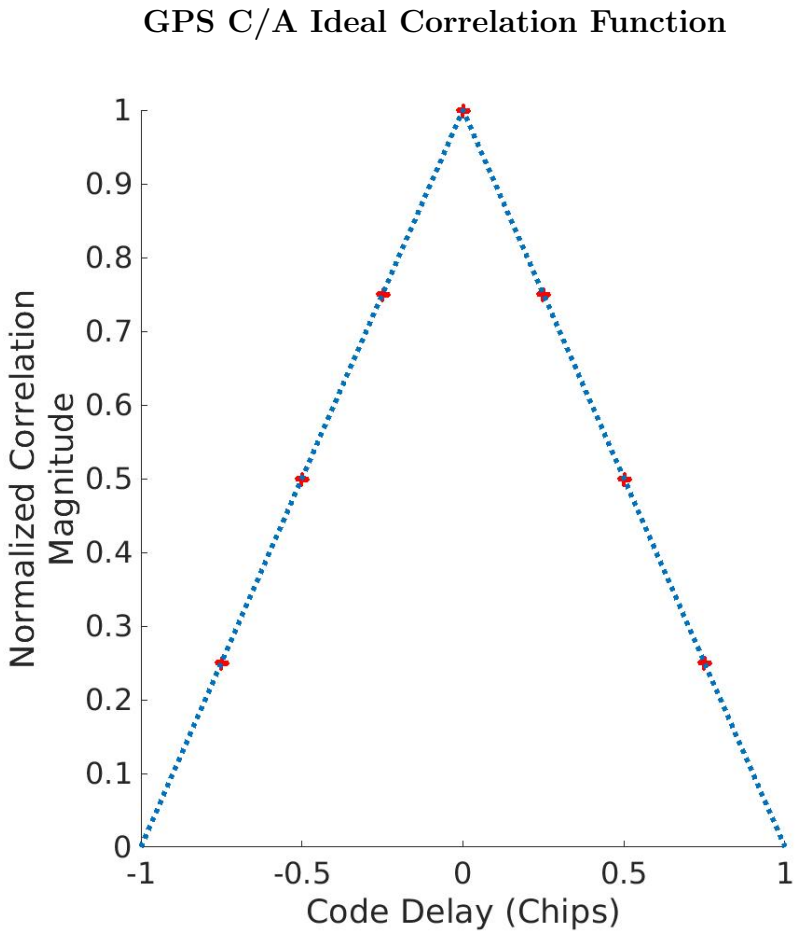


Figure 7: Correlation of an ideal GPS C/A signal with a replica. Seven discrete points are shown along the curve, along with the function's outline. The symmetric correlation function peaks when PRN codes are perfectly aligned (zero delay) and is minimized when misaligned by more than one chip.

In a similar manner, the tracking errors have been seen to predictably vary with different Early-Minus-Late (EML) correlator spacing [9]. This method of characterizing signal distortions directly applies to differential systems, as the reference (such as in WAAS) typically uses narrow correlator spacing while mobile users are using a variety of different spacing. If the varying errors aren't accounted for, additional error is introduced in the differential solution.

GPS C/A Correlation Function with Multipath Interference

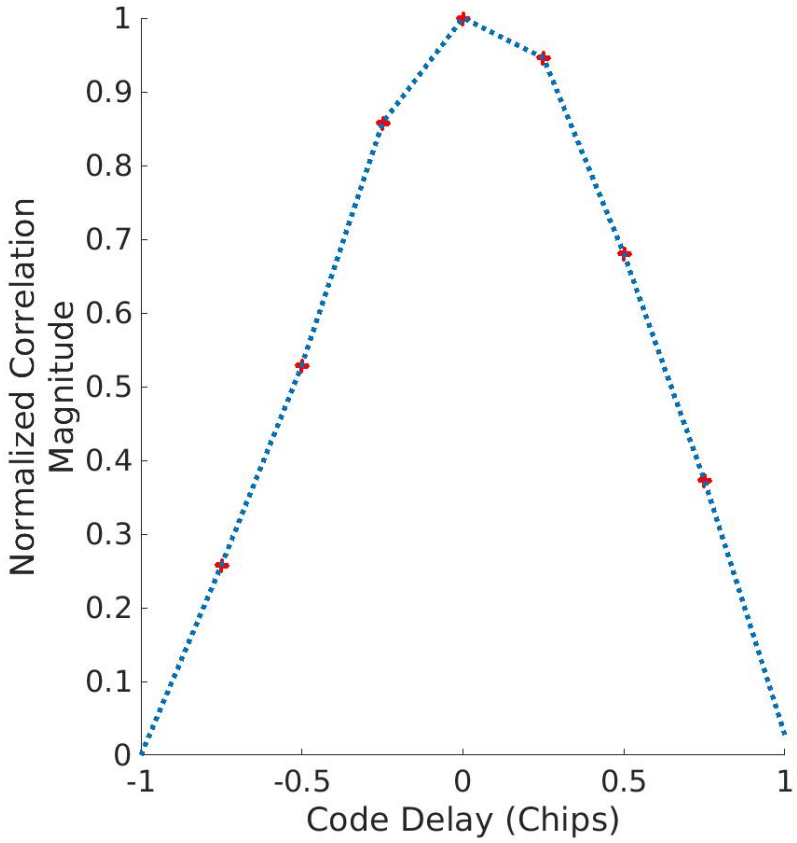


Figure 8: Correlation of a replica GPS signal with one distorted by multipath interference. Seven discrete points are shown along the curve, along with the function's outline. The deformations are obvious when compared to Figure 7, and the asymmetries lead to harmful ranging errors when used to track a signal. Asymmetries in this function may be exploited to determine when signal quality is unsafe.

2.4.3 SQM Using the Chip Shape Domain

The chip shape domain is computed by time-averaging like-portions in the signal's PRN code such that the likeness of the signal becomes observable. As the signal is added coherently, long integration causes random noise to be suppressed and greatly overpowered by the averaged signal. By assuming the signal's characteristics are constant over the integration period, the acquired chip shape then holds information directly related to the raw signal.

It has been shown that SV transmitters exhibit asymmetric deformations on signals, depending on whether the transition is rising-edge or falling-edge. These deformations are unique to each satellite due to the slight differences incurred during the manufacturing process of the signal generation hardware, and are observable in the chip shape domain [30].

As the signal deformations of interest are much more observable than the subtle differences in satellite generation hardware, it is expected to easily discern an unhealthy signal by its chip shape. To illustrate how the chip shapes differ, signals are simulated with a realistic amount of Additive White Gaussian Noise (AWGN), one with multipath present and one without. The chip shapes are shown in Figures 9, 10, and 11 with an obvious difference between the responses.

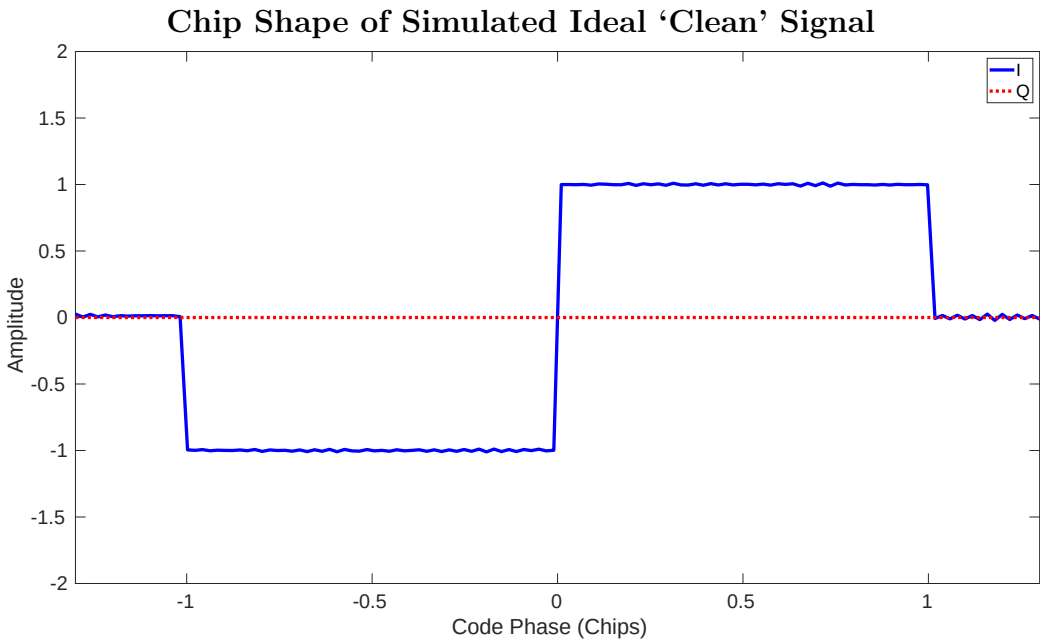


Figure 9: The chip shape of a simulated 'clean' signal. The in-phase (I) and quadrature-phase (Q) channels are shown, and this plot is achieved through the coherent integration of like-transitions.

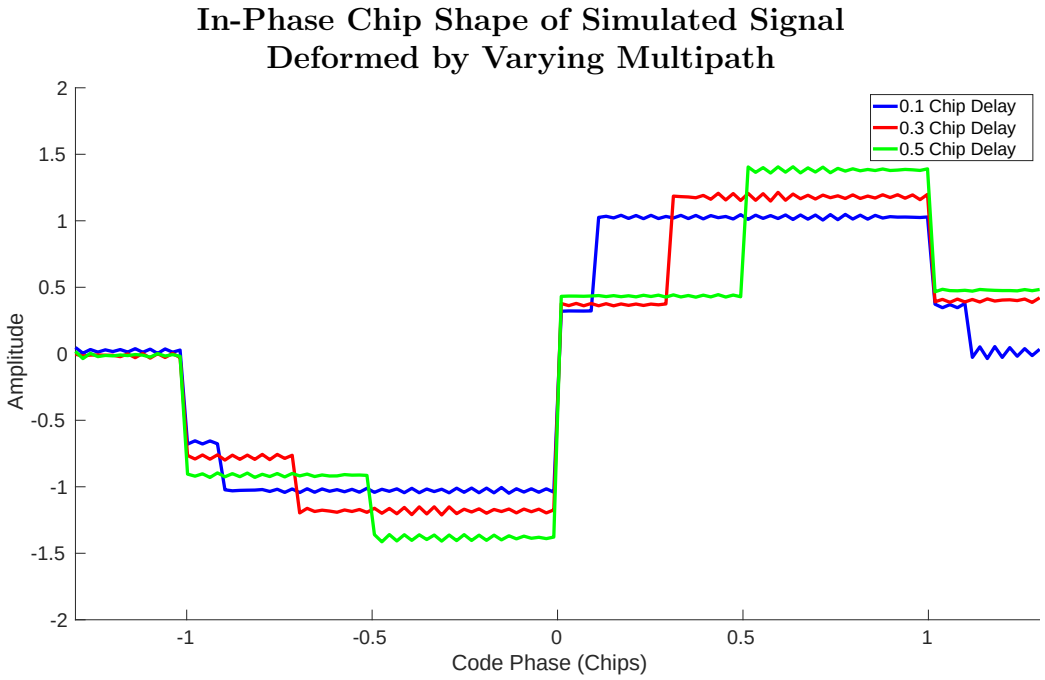


Figure 10: The in-phase chip shape of a simulated signal with multipath present. The signal is impacted by an additional signal attenuated to half power for various delays. There is a noticeable difference between these plots compared to 9, reinforcing that the chip shape domain may be used as a confident method of achieving SQM.

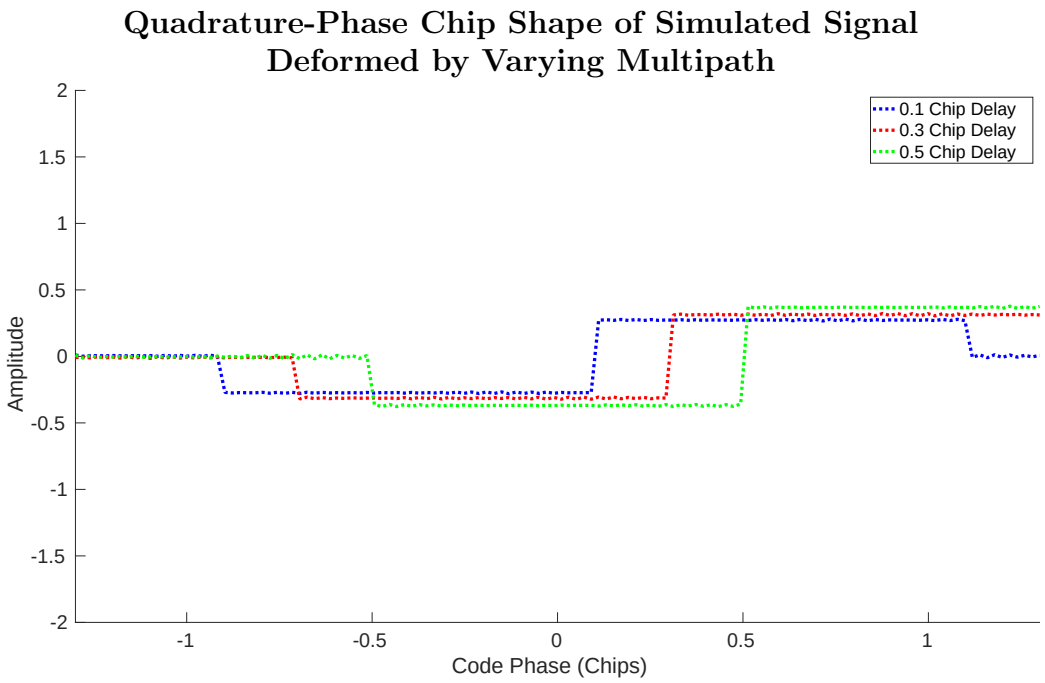


Figure 11: The quadrature-phase chip shape of a simulated signal with multipath present. The quadrature-phase counterpart to Figure 10 is shown, illustrating that both channels contain valuable information to signal deformation. This is attributed to the dependency of the delayed signal's carrier phase on the magnitude of the delay.

III. Methodology

3.1 Preamble

This chapter's goal is to describe a system that, in real-time, produces chip shapes of present Global Navigation Satellite System (GNSS) signals and uses them to draw conclusions on signal quality. In this context,"real-time" refers to processing a block of data before the next block is available. To follow the convention of typical GNSS receivers, the basic time block used is *1ms* and will be assumed throughout.

The system is split into three sections, illustrated in Figure 12: reading in live-sky data, acquiring and tracking GNSS signals in the data, and using the parameters computed while tracking to compute the chip shapes of the signals. The signal tracking process depends on the reception of live-sky data, and the chip shape computation process depends on tracking parameters, but once each process receives its respective input these processes are able to run independently. For this reason, each process is run on a parallel thread in order to minimize computation time. A circular buffer (designed in [31]) accumulates the raw data in $T_{blk} = 1ms$ blocks and passes it to the acquisition and tracking process whenever it is ready. This process returns parameters necessary to chip shape acquisition, which are accumulated on a separate circular buffer. The final process which computes chip shapes takes these parameters when available and accumulates chip transitions for some specified integration time.

The software is designed to process both the Global Positioning System (GPS) L1 Course-Acquisition (C/A) signal and the Galileo E1 signal. As the center frequency of each is centered at 1575.42MHz [17, 18], both signals are visible using the same data collection configuration.

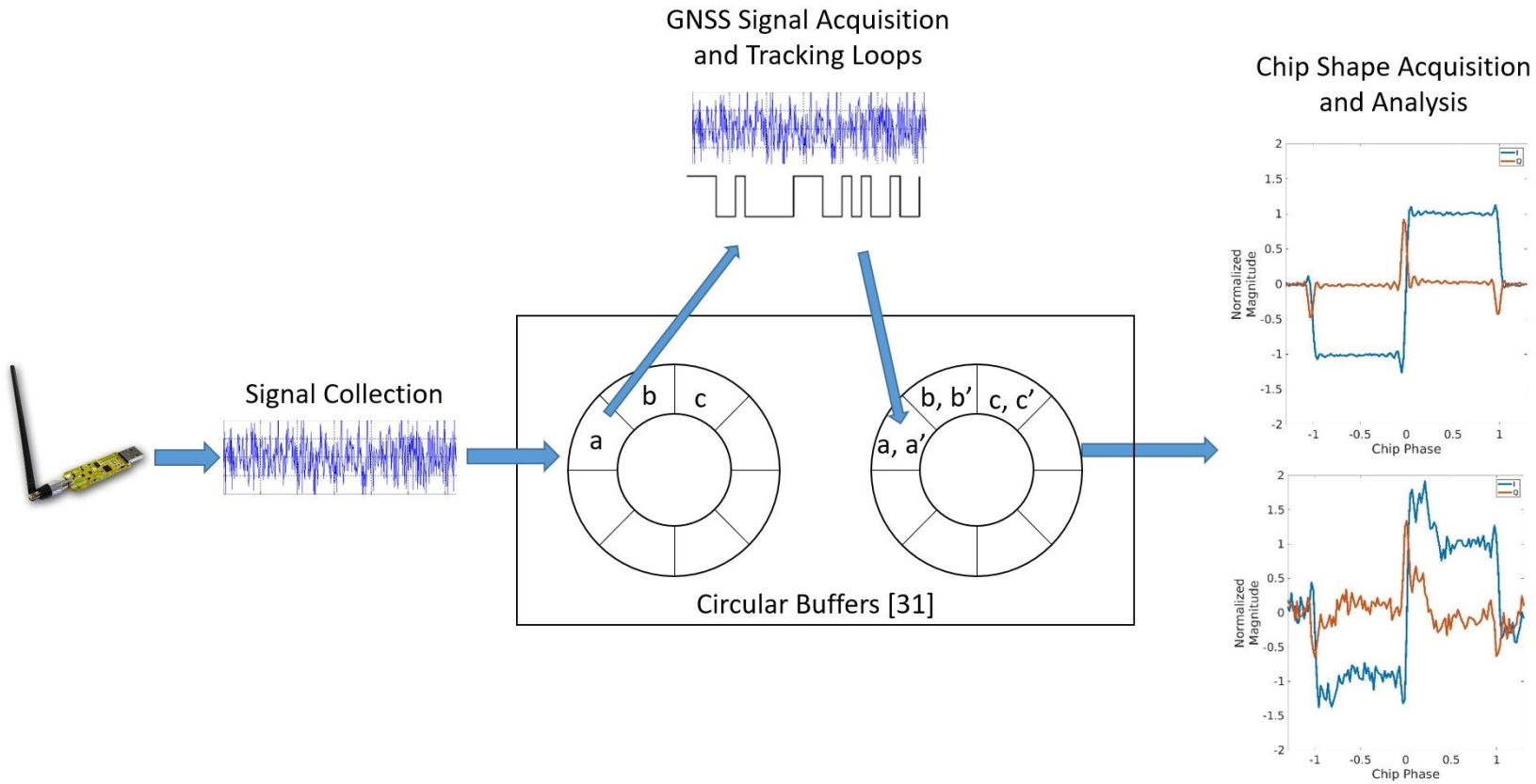


Figure 12: The diagram illustrating the system described in the methodology. This system is implemented using a Software Defined Radio (SDR) to collect live-sky data, and a C++ software architecture to process it. Circular buffers are used to hold data, as well as to save parameters related to each block of data (indicated by the the prime notation). Each component is detailed throughout the chapter. (Diagram is adapted from [31]).

3.2 Live-sky Data Collection

The first system component is responsible for the reception of live GNSS signals that makes the following signal processing possible. The process is applicable to both real-time data reception and recorded data for post-processing. This research uses both: 1) recorded data is used to differentiate between clean signals and signals with multipath present and 2) real-time reception is used to demonstrate the functionality of this Signal Quality Monitoring (SQM) method for a functioning receiver.

3.2.1 Receiver Front-End

The Radio Frequency (RF) front-end consists of all equipment used to convert live-sky signals to digital data. The antenna used for collections is the GPS portion of the *Antcom Iridium and GPS Antenna*. This component is an active antenna with a frequency range of 1562.42 – 1588.42MHz [32]. The SDR used is the Universal Software Radio Peripheral (USRP) B205mini-i by Ettus Research [33], which is able to provide up to 56MHz of instantaneous bandwidth. The USRP b205mini-i is used in conjunction with USRP Hardware Driver (UHD) libraries, which transfer collected data into a live software environment. The SDR is responsible for filtering, sampling, and quantizing analog data.

The antenna is powered using 5VDC and is connected directly to the SDR. Data is sampled as complex shorts at a sampling rate of 25 Mega-Samples per Second (MSPS), giving a front-end bandwidth of 25MHz. The SDR is tuned to a center frequency of 1575.38MHz, just 40kHz below the center band of both signals of interest.

3.2.2 Collection Environments

As the goal of the system is to determine signal quality based on chip shapes, data of both clean and distorted signals is desired. Since signal multipath is the most

prominent source of signal distortions, data is collected in both an open environment and an environment cluttered by buildings. The latter yields a high chance of signal reflections being collected, which are evident in the chip shape.

Data collection was performed at Wright-Patterson Air Force Base (WPAFB), OH, as it provides environments in open areas as well as near tall buildings. The locations are shown in Figures 13 and 14.

Figure 13: Location used to collect 'clean' GNSS signals. Here, there are minimal obstacles to interfere withlive-sky signal propagation. (Satellite image supplied by [34]).

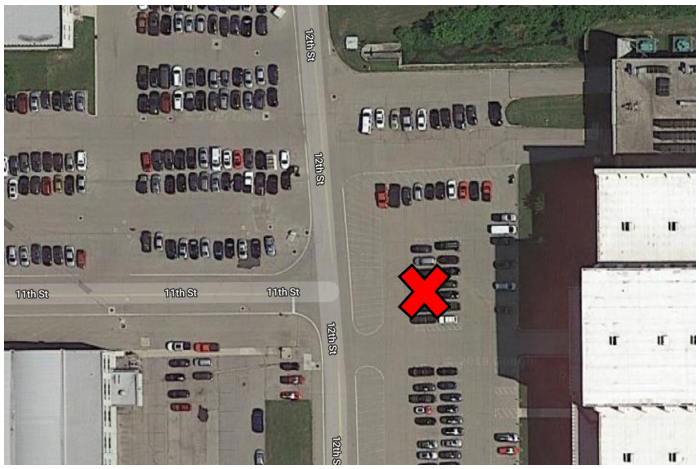


Figure 14: Location used to collect 'distorted' GNSS signals. Here, the surrounding buildings produce signal reflections that directly impact received signal responses. (Satellite image supplied by [34]).

3.3 GNSS Signal Processing and Tracking

Collected GNSS signals are digitally acquired and tracked by exploiting the known signal structure. Acquisition is used to estimate signal parameters that initialize tracking loops. The tracking loops then estimate code and carrier errors and use loop filters to apply corrections to the alignment of a local replica.

3.3.1 GNSS Signal Acquisition

The goal of GNSS signal acquisition is to search for different Pseudo-Random Noise (PRN) codes, and then to determine if they are present or not. This process is a two-dimensional search for each PRN code: satellite motion induces a Doppler effect on the signal, producing an unknown frequency offset, and by nature of the system there is an unknown PRN code offset (as the code offset will precisely determine when the signal was received.)

GNSS signal acquisition is inherently computationally expensive, as the frequency and code estimates must be accurate enough to initialize the subsequent tracking loops. For a stationary receiver, the maximum line-of-sight (LOS) velocity of a Satellite Vehicle (SV) is 800 m/s [13], corresponding to a frequency offset of roughly 4.2kHz. In order for the tracking loops (described below) to be able to pull-in the acquired signal, the frequency estimate must be within 667Hz and the code estimate must be within half of a chip to the true values (based on the $T_{blk} = 1ms$ correlation time) [22]. The search space spans N_f bins in the frequency dimension and N_c bins in the code dimension, given by

$$
\begin{aligned}
N_f &= \text{ceil}\left[8400Hz * \frac{1}{r} * \left(\frac{F_{samp} * T_{acq}}{BW}\right)\right] \ , \\
N_c &= \text{ceil}\left[L_{code} * \left(\frac{F_{samp}}{R_{chip}}\right)\right] \ ,
\end{aligned}
\tag{8}
$$

where ceil () is the ceiling of the argument, F_{samp} is the sampling rate in samples per second, T_{acq} is the duration (in seconds) of data being searched, BW is the received bandwidth in Hz, R_{chip} is the signal's chipping rate in chips per second, L_{code} is the length of the signal's PRN code (in chips), and r is a unitless scaling factor that by design may be increased to speed-up the acquisition process at the cost of precision. The scaling factor is implemented by only searching every r^{th} frequency bin.

A brute-force search would yield $N_f * N_c$ complex correlations to be computed, not including the computation of frequency shifting the signal across the N_f bins. To achieve much faster computation time, the correlation operation is translated to the frequency domain using the Fast-Fourier Transform (FFT) function provided by the *liquid-DSP* library [35]. By computing the FFT on the received data and the local replica and taking the conjugate of the latter, the element-wise multiplication of these results is the FFT of the time-domain circular correlation (assuming the local replica is designed to have the same number of points as the received data). The received data is translated to be centered about baseband (0Hz) before the acquisition process begins; after the FFT of this data is computed, frequency bins are searched by simply circularly shifting the resultant vector, where each r bin shifts are equivalent to $\left(\frac{r}{T_{acq}}\right) Hz$. Here, T_{acq} is the duration (in seconds) of the data being searched and r is the scaling factor chosen in (8). Circular shifts are implemented by indexing the array differently, which takes no computation at all. Circular shifting is a big computational improvement from frequency shifting using a complex carrier. Lastly, the Inverse Fast-Fourier Transform (IFFT) is computed to analyze the correlation

results.

As N_f frequency offsets are being searched, there are N_f resulting vectors. The maximum element from all vectors gives both the frequency and the code estimates. The resultant vector in which the maximum occurs gives the estimate of the frequency offset, based on the frequency bin width and how many circular shifts were made. The index of the maximum relates to the estimate of the code offset, where the number of samples between the maximum and the end of the vector gives the code offset estimate *in samples* of the input data. The maximum's index then relates to chips by multiplying it by the code resolution, that is, $\frac{R_{chip}}{F_{samp}}$. The code offset estimation is then given by

$$\begin{aligned} \mathbf{V} &= \text{IFFT}\left(\text{FFT}(\mathbf{data})_S * \text{FFT}(\mathbf{replica})\right) \ , \\ \text{Code Offset} &= \left[T_{acq} * F_{samp} - \left(\underset{k}{\text{argmax}}\, V(k)\right)\right] * \frac{R_{chip}}{F_{samp}} \ , \end{aligned} \tag{9}$$

where F_{samp} is the sampling rate, T_{acq} is the duration (in seconds) of data being searched, R_{chip} is the signal's chipping rate, **data** is the input data vector, S is the circular shift performed on the FFT of the data vector, and **replica** is the upsampled code vector.

3.3.1.1 Galileo E1 Signal Acquisition

GPS signals are acquired using the method described above, however small modifications are made to account for differences in the Galileo E1 signal. The PRN sequence is *4 ms* long, and most systems will not search an entire T_{acq} =*4 ms* of data, as this amount of data requires heavy computation. Instead, a smaller block of data is used (for example, T_{acq} =*2 ms* or T_{acq} =*3 ms* of data), and different portions of the PRN code are searched within this block.

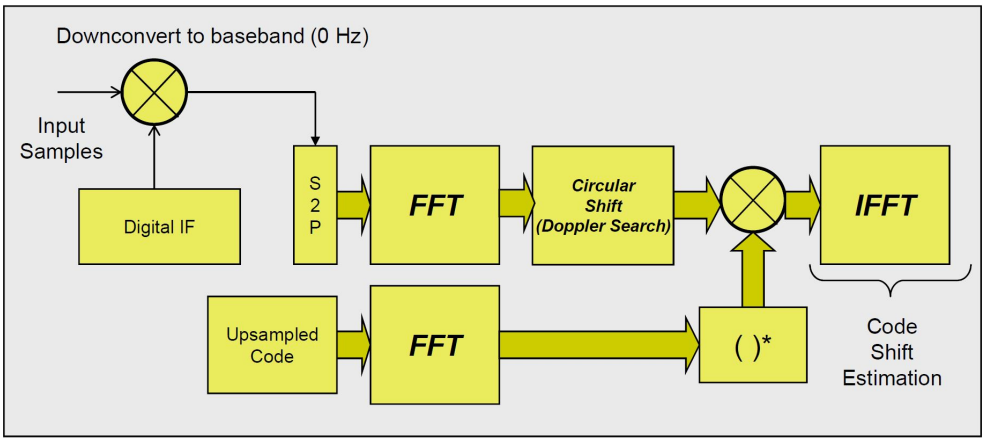


Figure 15: Outline of the acquisition process used. This process efficiently estimates the signal's frequency and code offsets. The FFT is used to search all code offsets in parallel, and circular shifts are used to effectively shift the carrier offset. Here, "S2P" means "serial-to-parallel" if the acquisition process were implemented by digital hardware. (Figure provided by [22]).

Assume $T_{acq} = 3\ ms$ of data is used to acquire a Galileo E1 Signal and divide the $4\ ms$ PRN code into four even portions. Then, it is guaranteed that one of the four portions exists in its entirety within the data. The acquisition process above is then implemented four times, with a different portion of the PRN code used in each iteration.

Since this process relies on FFT-based correlation (which operates circularly), it will produce faulty results should the data vector and the PRN being searched for be of different lengths. (That is, the chip- and frequency-resolution per FFT point needs to be identical). Therefore, the portion of the PRN code must be zero-padded to be of the same length as the data.

3.3.2 GNSS Tracking Loop

Immediately following GNSS signal acquisition, tracking loops are used to maintain frequency and code alignment of a local replica with the received signal. Due to the signal's low power, it is not observable and this local replica is representative of

the signal's alignment once it reaches the front-end's antenna.

3.3.2.1 Correlation with Local Replica

This first step of the tracking loop is also the most computationally expensive: the dot product of the received data with the local replica must be computed. This architecture uses an Early-Minus-Late (EML) code discriminator to estimate the code offset, requiring three local replicas with slightly different code offsets to be correlated with the input signal. These three replicas will be denoted as the Early, Prompt, and Late replicas, with their names referring to the code phase offset.

For a sampling rate of $F_{samp} = 25$ MSPS, this corresponds to the accumulation of 75,000 complex multiplications that must be executed for each millisecond block. The local replica's initial state is either determined by the acquisition process or the previous tracking loop iteration. The initial state determines the starting PRN code phase and the starting phase and frequency of the complex carrier.

$$\text{Early} = \sum_{idx=1}^{25000} \mathbf{data}(idx) \times \mathbf{c}(c_0 + idx \times dChip + d/2) \times \exp\left[j(\phi_0 - idx \times d\phi)\right] \quad (10)$$

$$\text{Prompt} = \sum_{idx=1}^{25000} \mathbf{data}(idx) \times \mathbf{c}(c_0 + idx \times dChip) \times \exp\left[j(\phi_0 - idx \times d\phi)\right] \quad (11)$$

$$\text{Late} = \sum_{idx=1}^{25000} \mathbf{data}(idx) \times \mathbf{c}(c_0 + idx \times dChip - d/2) \times \exp\left[j(\phi_0 - idx \times d\phi)\right] \quad (12)$$

In (10), (11), and (12), **data** refers to the complex samples received, **c** refers to the PRN code (including any subcarrier and/or overlay code), d refers to the code spacing between the Early and the Late correlator in chips, c_0 and ϕ_0 refer to the beginning code and carrier phase, respectively, $dChip$ is the magnitude of code chips per sample, and $d\phi$ is the carrier radians per sample. Notice that the difference in the three equations is solely in the code phase and not the carrier phase, as code

corrections are made independently of carrier corrections. The complex carrier may be interpreted one of two ways. First, its purpose can be seen as translating the received signal to baseband (ie, 0Hz) to be compared with the ideal baseband replica. Alternatively, it can be seen as translating the frequency of the replica to that of the received signal, yet taking the complex conjugate such that the product yields minimal quadrature amplitude if phase and frequency are aligned. For either interpretation, it is critical to begin the carrier phase where the previous iteration left off to avoid incoherent leaps in the carrier.

The above would work for a pure Direct Sequence Spread Spectrum (DSSS) signal with no data modulation or overlay code sequence, however both signals of interest include one of the two. The issue arises when the polarity of the code changes in the block of data that is received; if data with a polarity change were to be correlated with a local replica of unchanging polarity, the correlation magnitude would be drastically reduced. It is known that these potential polarity changes occur at the start of each code period, known as the epoch. Therefore, correlation before and after the epoch are separated to avoid atenuation of correlation magnitude.

This process is known as *split-sum correlation* [22]–the dot product of data *before* the epoch is added with the dot product of the previous block's data *after* the epoch. The dot product of current data *after* the epoch is then saved to be used in the next block. Split-sum correlation ensures that a full block's worth of data without a polarity change is correlated. For GPS signals, this split-sum is implemented in every block to account for the code epoch each 1023 chips. For Galileo signals, even though the code epoch occurs every four blocks, the split-sum is computed about every 1023 chips such that the correlation magnitude stays consistent. This process is illustrated in Figure 16.

As correlation is the bottleneck of each loop's iteration, it is critical to compute it

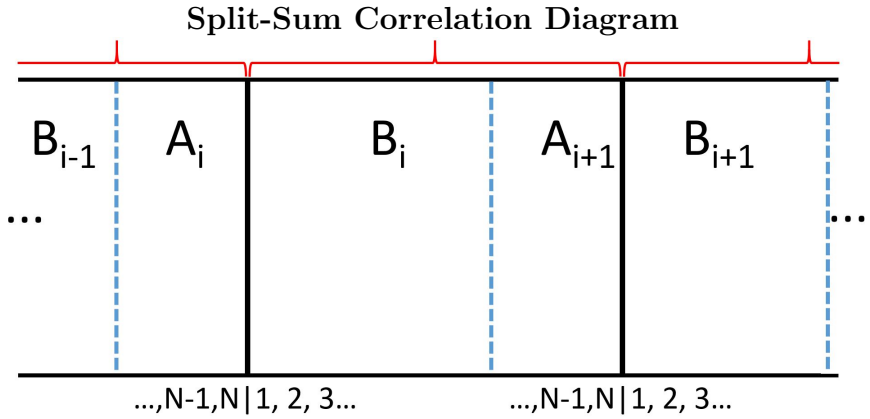


Figure 16: Split-sum correlator diagram. The organization of data reception is shown, with solid black lines representing the transmitted data blocks, and dotted blue lines the received data blocks. Split-sum correlation divides the data about the transmitted code epoch as described. A and B refer to the correlation outputs of that corresponding section. As above, A blocks are summed with B blocks of the previous index, even though they are not received within the same data block.

in the most efficient way possible. Publicly available digital signal processing library, *liquid-DSP*, is used to achieve better run-time performance. In particular, the vector dot-product object is used to implement (10), (11), and (12) as it makes use of Single Instruction for Multiple Data (SIMD) processes depending on the system's architecture [35].

3.3.2.2 Carrier and Code Discriminators

Using the output of the initial dot products, parameters of the signal are estimated to eventually provide corrections to the local replica by using a loop filter. In other words, discriminators compute how much the local replica differs from the received, in phase or in frequency.

Carrier discriminators use the output of the prompt correlator to estimate phase and frequency differences between the received signal and the local replica. In this system, each block of data is fixed at a $T_{blk} = 1ms$ duration, but the correlation outputs may be coherently added for several blocks as long as no polarity change occurs.

The total time that correlation is summed over to compute carrier discriminators is referred to as the pre-detection integration time. As described in the previous section, sudden polarity changes in the signal are expected, causing the correlation magnitude to instantly change sign. It is undesired to include this rapid phase change, as it is indicative of the modulation and not the carrier wave. For this reason, the two-quadrant arctangent is used such that a phase change of π radians has no effect on the discriminator output. The phase disciminator is then computed by (13),

$$\Delta\phi = \arctan\left(\frac{Q}{I}\right) \ , \tag{13}$$

where arctan is the two-quadrant arctangent, and Q, I refer to the quadrature-phase and in-phase components of the prompt correlation.

Similarly, the frequency discriminator is given by (14),

$$\Delta f = \frac{\arctan\left(\frac{I_{m1}*Q - Q_{m1}*I}{I_{m1}*I + Q_{m1}*Q}\right)}{T_{pdi}} \ , \tag{14}$$

where the subscript 'm1' refers to the magnitude of the component from the previous tracking iteration, and T_{pdi} refers to the pre-detection integration time used to accumulate the correlation magnitudes.

Code discriminators use the EML correlator structure to estimate the difference in code phase of the received data and the local replica. As the correlation triangle is symmetric, the difference between the early and late correlator values should ideally be zero. The error is then estimated by this difference, and normalized by the magnitude of the prompt correlator. The code discriminator is computed as

$$\Delta x = \frac{E - L}{4P} \ , \tag{15}$$

where Δx is the correction to the code phase, and E, P, L are the *magnitudes* of the

Early, Prompt, and Late correlators. In the final state of tracking, the goal is to have all tracking in the in-phase channel, meaning the quadrature channel contains only noise. At this point, E, P, and L are instead computed as the in-phase correlation magnitude to reduce system noise. Discriminating the code error and making closed-loop corrections is what composes a Delay-Locked Loop (DLL).

The correlation may be summed for even longer when computing the code discriminator. As it only requires the magnitude of correlation, the absolute value of each channel is summed over a period of time known as the after-detection integration time. This process functions even with polarity changes, as they are removed by the absolute value.

3.3.2.3 Loop Filter and Corrections

The purpose of the loop filter is to reduce noise and use the computed discriminators to respond to signal dynamics. Methods used are directly adapted from [13]. This is the final portion of the tracking loop before the closed-loop corrections are made to align the replica with the received signal. The input is determined by the type of loop being used, namely, a Phase-Locked Loop (PLL), a Frequency-Locked Loop (FLL), or a hybrid of the two.

Depending on the filter order, the input is passed through a series of multipliers and integrators which accumulate and adapt to changes in the signal. The multiplier values are outlined in [13], with a dependency only on the filter order and the chosen noise bandwidth B_0. As a FLL uses frequency discriminators, a first-order loop would respond to changes in frequency and a second-order loop would respond to second-degree changes in frequency, corresponding to velocity and acceleration of the SV. Similarly, since a PLL makes use of phase discriminators, a first-order loop responds to changes in phase, a second-order responds to changes in frequency, and a third-

order responds to second-degree changes in frequency (corresponding to position, velocity, and acceleration of the SV). A first-order FLL is typically desired at the beginning, as it pulls in frequency offset but has a mild response when compared to higher-order loops. For long-term tracking, however, a higher-order loop must be used to track acceleration motion of the SV.

The final design choice to be made is the noise bandwidth, B_0. With respect to the filter response, higher noise bandwidths lead to faster responses to changes in the signal. It is typically desired to begin with low noise bandwidths, and increase it once stable tracking is achieved. The maximum noise bandwidth is determined by the pre-detection integration time (T_{pdi}) used to calculate discriminators, described in [22] by:

$$B_0 \leq \frac{1}{2T_{pdi}} \ . \tag{16}$$

Since different parameters are desired at different stages of tracking, a state machine is required to advance the tracking once the previous stage's purpose is fulfilled. The desired end-goal is to achieve carrier-phase tracking with a narrow EML correlator spacing, illustrated in Figure 17.

3.3.2.4 Synchronizing to Signal Polarity

In the final state of tracking, the signal's code and carrier phase are being tracked. Carrier phase tracking is conveyed by correlation magnitude being entirely within the in-phase channel, while the quadrature-phase channel is minimized. Overlay sequences are modulated over both signals of interest and are seen by sharp polarity changes of the in-phase channel. For the GPS C/A signals, polarity changes are due to data bit transitions that occur every *20 ms* [18]. For the Galileo E1C signals, polarity changes are due to the secondary 25-chip long sequence with chip transitions

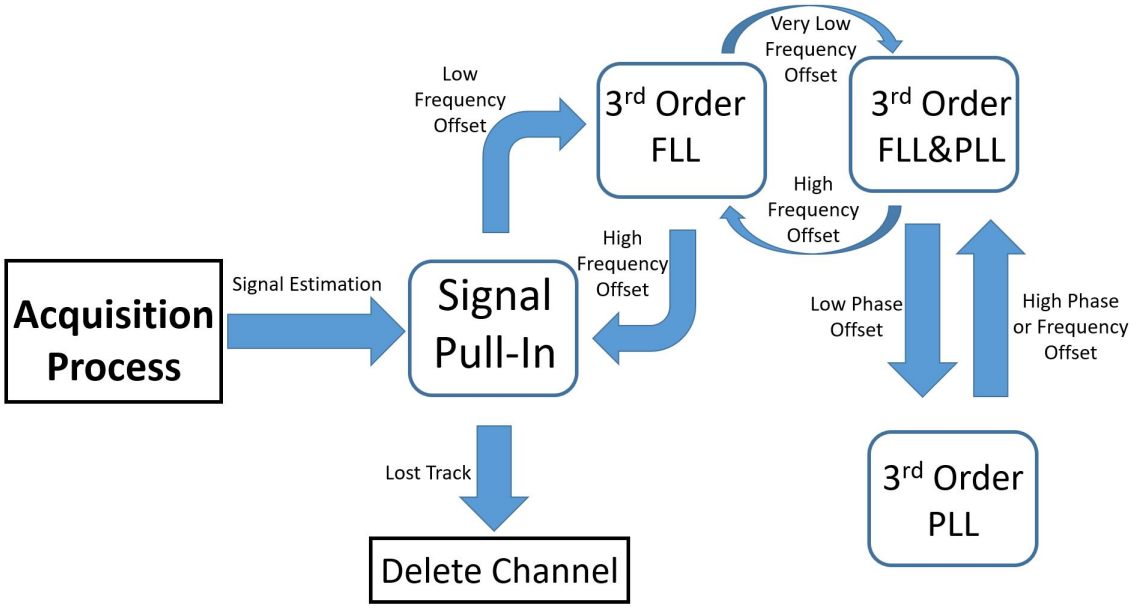


Figure 17: The flow diagram for the tracking loop controller. This figure conceptualizes the desired progression to achieve close carrier phase tracking. The specifics of each state are expanded on in Table 1.

on every code epoch, every *4 ms* [17]. To begin polarity synchronization, the sign of the in-phase channel is extracted to represent the current bit (or chip).

The GPS navigation message begins each subframe with a telemetry (TLM) word that has a fixed unique preamble that corresponds to the bit sequence of "10001011" [13]. To synchronize to polarity, bits are estimated by computing the sign of the in-phase correlation magnitude and the preamble is correlated with the previous eight bits received. (When correlating bit sequences, all 0's are represented as -1 to not suppress magnitude). A correlation magnitude of eight reveals the preamble was found, and that the replica's polarity is already synchronized with the received signal. A correlation magnitude of negative eight reveals that the preamble was found, but the replica's polarity is opposite of the signal's. In this case, a phase compensation of π radians is added to the Numerically-Controlled Oscillator (NCO) to align the local replica's polarity with the actual signal. The process of searching for the preamble continues until a correlation magnitude of eight is found.

Tracking State Parameters

Table 1: Tracking parameters for each state. This table expands on Figure 17. Here, Pdi refers to the "pre-detection integration time," which is the correlation time used to compute carrier discriminators. Similarly, Adi refers to the "after-detection integration time," which is the time used to compute the code discriminator.

State	Parameters
Signal Pull-In	FLL: 2nd Order Pdi = 1ms FLL Noise Bandwidth = 10Hz DLL: 1st Order with Carrier Aiding Adi = 1ms EML: 1 Chip Spacing, Non-Coherent Discriminator DLL Noise Bandwidth = 5Hz
3rd Order FLL	FLL: 3rd Order Pdi = 1ms FLL Noise Bandwidth = 18Hz DLL: 1st Order with Carrier Aiding Adi = 1ms EML: 0.5 Chip Spacing, Non-Coherent Discriminator DLL Noise Bandwidth = 10 Hz
3rd Order FLL&PLL	FLL/PLL: 3rd Order Pdi = 1ms FLL Noise Bandwidth = 10 Hz PLL Noise Bandwidth = 5 Hz DLL: 1st Order with Carrier Aiding Adi = 10ms EML: 0.3 Chip Spacing, Non-Coherent Discriminator DLL Noise Bandwidth = 5 Hz
3rd Order PLL	PLL: 3rd order Pdi = 1ms ; 20ms after symbol/overlay synchronization PLL Noise Bandwidth = 18 Hz [Synchronization to Overlay symbols or code] DLL: 1st Order with Carrier Aiding Adi = 200ms EML: 0.1 Chip Spacing, Coherent Discriminator DLL Noise Bandwidth = 2 Hz

The Galileo E1C secondary code is a known sequence of 25 chips. Once 25 chips are accumulated from the in-phase channel, it is circularly correlated with the actual sequence. The secondary code offset which yields a magnitude of 25 determines two things. First, it shows the current offset of the overlaying code, which is used to include the secondary code in the local replica. As the entire signal is now described, long, coherent integration of correlator outputs is now achievable on the 'pilot' component [22]. Second, the polarity is determined with the sign of the circular correlation output. As with the GPS signal described, if the sign is negative, a phase compensation of π radians is added to the local replica's NCO to align polarity.

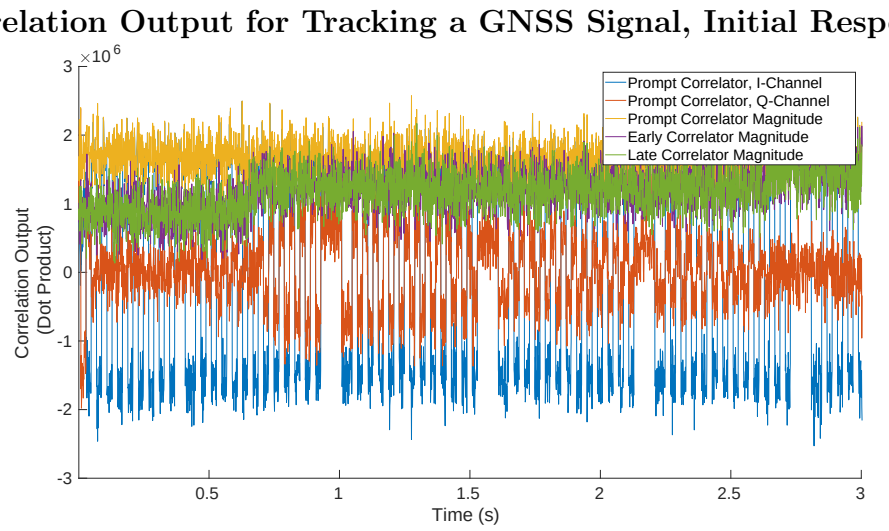


Figure 18: Correlator outputs for the first few seconds of tracking the GPS C/A PRN 1 signal. The frequency offset is observed in the initially changing magnitudes in the prompt I and Q channels. As the tracking loop pulls in the frequency offset, the I and Q magnitudes become constant (ignoring the polarity changes). As the frequency becomes stable, the phase is pulled in and the Q channel magnitude is forced to zero. The late and early correlator magnitudes are roughly equal, meaning the code is being tracked successfully. The increase in magnitude of these correlators is due to the early-late correlator spacing being made narrower.

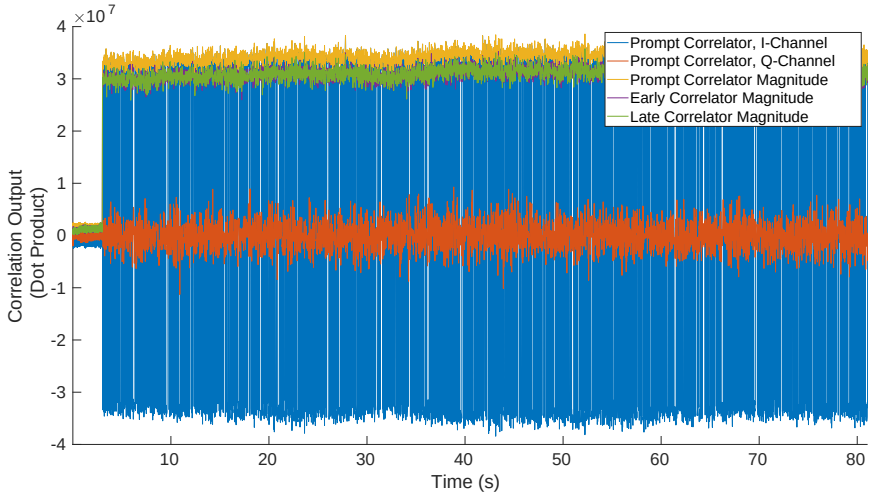


Figure 19: Correlator outputs for tracking the GPS C/A PRN 1 signal over a long period of time. Once phase is being tracked, the overlaying symbols are sychronized to and the pre-detection integration time is increased to 20ms. That is, 20ms of data are accumulated in correlation, seen by a large jump in magnitude. Finally, polarity may be determined and compensated for at this stage. At this point, each correlation output relates to a data bit. Signal tracking is now stable and may continue for long periods of time.

3.3.2.5 Tracking Indicators

Tracking indicators are computed values that are used to describe different qualities during the signal tracking process. Two indicators will be described here, namely, Phase-Lock (P-Lock) and an estimator of signal Carrier-to-Noise Ratio (CNR). These indicators are used within the receiver architecture to assist the state machine transition process. All methods described are taken from [22].

Both indicators of interest are dependent on intermediary calculations: Narrow-Band Power (NBP), Wide-Band Power (WBP), and Normalized Power (NP). NBP is achieved by coherently summing the I- and Q- channels of m correlations, then taking the magnitude of the result. NBP is described in (17), where $\mathbf{Corr}_I$ and $\mathbf{Corr}_Q$ refer to the in-phase and quadrature-phase results after correlating the data with a local replica.

$$\begin{aligned}\mathbf{NBP}_I &= \sum_{i=1}^{m} \mathbf{Corr}_I(i) \\ \mathbf{NBP}_Q &= \sum_{i=1}^{m} \mathbf{Corr}_Q(i) \\ \mathbf{NBP} &= (\mathbf{NBP}_I)^2 + (\mathbf{NBP}_Q)^2\end{aligned} \tag{17}$$

WBP is similar to NBP, however the correlation values of each block are incoherently summed by taking the magnitude before adding. WBP is calculated as

$$\mathbf{WBP} = \sum_{i=1}^{m} (\mathbf{Corr_I}(i)^2 + \mathbf{Corr_Q}(i)^2) \ . \tag{18}$$

Lastly, NP is the ratio of NBP to WBP, given by

$$\mathbf{NP} = \frac{\mathbf{NBP}}{\mathbf{WBP}} \ . \tag{19}$$

By using (17), (18), and (19), the indicators of interest may be computed. P-Lock is computed as the difference between the accumulated I- and Q-channels of m correlations, divided by the NBP itself. Using symbols from (17), P-Lock is calculated as

$$\mathbf{P\text{-}Lock} = \frac{\mathbf{NBP}_I - \mathbf{NBP}_Q}{\mathbf{NBP}} \ . \tag{20}$$

P-Lock then gives a metric of measuring how well the phase is being tracked. If all correlation is in the in-phase channel, then P-Lock will equal 1. If all is in the quadrature-phase channel, then P-Lock will equal -1. All other values lie in between these extremes.

The CNR estimate is a way of measuring signal strength at the receiver. As typical systems might measure the Signal-to-Noise Ratio (SNR), this metric is bandwidth-

dependent. CNR is a value that quantifies signal strength independent of the bandwidth used by a receiver. CNR is estimated by averaging the NP over k blocks, and using the relation defined in (21).

$$\mathbf{CNR} = \frac{\mathbf{N\bar{P}}_k - 1}{T * (m - \mathbf{N\bar{P}}_k)}$$
$$\mathbf{N\bar{P}}_k = \frac{\sum_{n=1}^{k} \mathbf{NP}(n)}{k} \tag{21}$$

Here, m is the number of blocks used to compute each NP value, T is the basic correlation time used in seconds (i.e. T_{blk} =*1ms*), and k is the number of NP values that are averaged. Results of the above computations are shown in Figures 20 and 21.

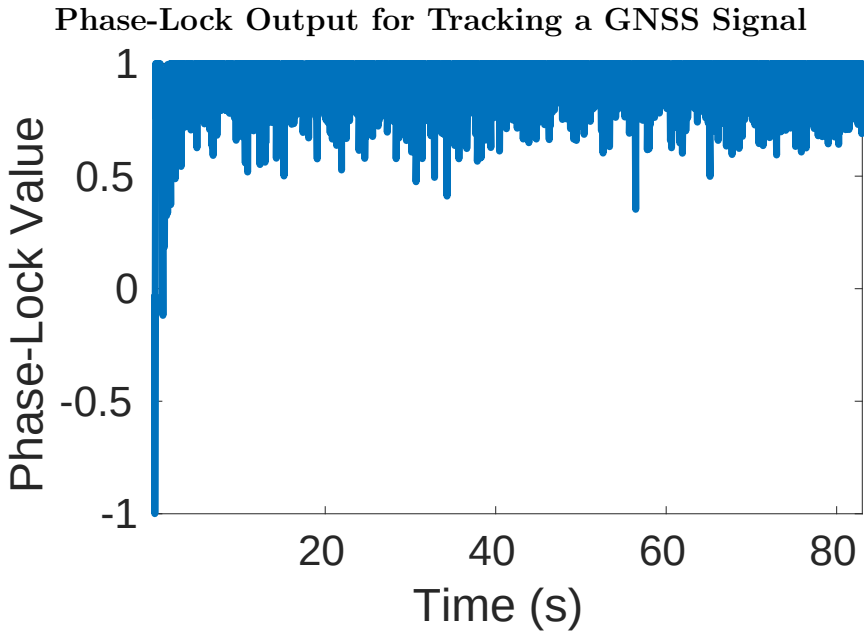


Figure 20: P-Lock output plotted against time for tracking the GPS PRN 1 C/A signal. This figure uses the same data processed in Figure 19. Initially, P-Lock rapidly varies between -1 and 1, but as the tracking loop progresses, phase is pulled-in and the P-Lock approaches 1.

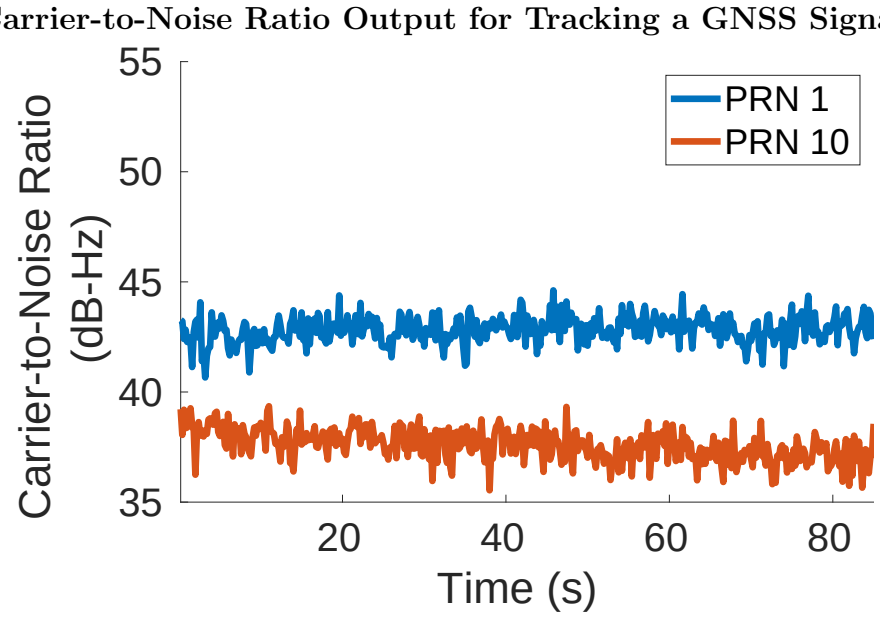


Figure 21: The CNR output for two GPS PRNs, plotted on a log scale, for the same data as in Figure 20. As the time span is a relatively short time to track a GNSS signal, the CNR remains roughly constant throughout the tracking period. The values shown in PRN 1 indicate a strong signal reception, whereas the values from PRN 10 indicate weak signal reception. As the SV leaves the antenna's LOS, it is expected for the CNR to diminish significantly. CNR is then a good measure of when a signal is no longer being tracked.

3.4 Chip Shape Analysis

Using GNSS signal chip shapes as a means of SQM is the focus of this document. The process, along with everything needed to implement it, is described below.

3.4.1 Chip Shape Acquisition

As discussed, a GNSS signal's chip shape is the time-averaged representation of a signal, acquired by summing like-portions of the code over some period of time [10]. By assuming that the signal's characteristics stay constant over some time, the chip shape then represents the signal's raw characteristics transmitted from the satellite.

In this context, the word 'transition' describes the event of a GNSS signal's code progressing to the next chip in the sequence. Since the PRN codes are bi-

nary sequences, there are four possible transitions that occur across the code period: $\{[-1,-1],[-1,1],[1,-1],[1,1]\}$. Because data bits and secondary sequences overlay the primary PRN codes, the polarity of the sequence is initially ambiguous. For this reason, chip shapes are not acquired until the signal polarity is determined and tracked by the receiver (i.e. the phase is being tracked and polarity has been compensated for).

To acquire chip shapes, several parameters are saved during tracking for each block of data. These parameters are listed below.

- Initial Code Phase - With double-precision floating-point, initial code phase describes the starting PRN chip for that block of data, including the fraction of the chip that the first data point begins on.

- Initial Carrier Phase - Also with double-precision floating-point, initial carrier phase gives the beginning phase of the carrier offset on a range of $[0, 2\pi)$. This parameter includes the phase correction made during the tracking iteration.

- Frequency Offset - The frequency offset gives the signal's carrier frequency offset due to Doppler shift and oscillator imperfections, accounting for the correction estimated during the tracking iteration.

- Polarity Before and After Epoch - If a PRN code epoch occurs within the given block of data, there is a possible polarity change during the block. These two parameters give the polarity before and after the possible epoch's polarity change, such that the precise transitions are known throughout the entire block.

Due to the design of the split-sum correlator within the tracking loop, the final parameter listed is not known until the following block of data is processed. That is, polarity after a code epoch is undetermined until the next block. When designing the receiver, tracking loops then pass on parameters for the preceding block of data.

These parameters, along with the data, are all that are needed to acquire chip shapes. As each block of data is T_{blk} =*1 ms* in duration, a buffer will accumulate the chip shape data until the desired accumulation time, then the buffer will be dumped and the process begins again. The algorithm is described in Algorithm 1 in basic terms.

The algorithm is very simple at this level of abstraction, with complexity present depending on the design. The function inputs refer to the previously listed parameters: 1) the starting code phase, c_0, 2) the starting carrier phase, ϕ_0, 3) the frequency offset, f_0, and 4) the polarities before and after the code epoch, p_A and p_B. The first step, down-conversion to baseband, is done with a simple loop by multiplying each data point by the estimated complex carrier wave. The down-conversion makes use of the initial carrier phase as well as the frequency offset to result in the signal content being purely in the in-phase channel throughout the duration of the data block. As each data block is only T_{blk} =*1ms* in duration, frequency changes within the block are assumed to be negligible, allowing for the signal to remain in-phase throughout. Determining the code transition that was transmitted is trivial given the parameters. The PRN code dictates the transition, and the current polarity of the signal determines whether the transition is reversed or not.

Algorithm 1 Chip Shape Acquisition

```
1:  function CHIP SHAPE ACQUISITION(c_0, φ_0, f_0, p_A, p_B)
2:      Wipe carrier offset from data                    ▷ Use Complex Carrier
3:      for idx ← 1, N do                    ▷ Loop through all complex samples
4:          if Code Transition Occurred then
5:              Determine transition, use relative buffer
6:              Accumulate data from previous and current chips into relative bins
7:          else
8:              Continue to Next Iteration
9:          end if
10:     end for
11: end function
```

Line 6 of Algorithm 1 brings the most freedom in design. Here, data is accumulated to result in the actual chip shapes. To optimize the accumulation, buffers are used to save data points in their corresponding time bins, then all of the bins are added to the correct chip shape buffer at each transition. The number of bins per chip is user-defined, and should be at least equal to the number of bins defined by the sampling rate as shown in (22). As the number of bins increases, the resolution increases yet so does the noise.

$$\mathrm{N}_{bins}\mathrm{N}_{bins}{}^{\mathrm{N}_{bins}{}^{\mathrm{N}_{bins}}} = \left(\frac{F_{samp}}{R_{chip}}\right) \quad (22)$$

Here, N_{bins} describes the number of time bins per chip depending on the sample rate, where F_{samp} is the sample rate and R_{chip} is the signal's chipping rate.

Data points are assigned to bins based on their code phase. Bin assignment is done by simply multiplying the fractional code phase by the number of bins per chip, implemented by

$$\text{Bin Number} = \text{floor}(N_C * [\phi - \text{floor}(\phi)]) \ , \quad (23)$$

where N_C is the number of bins per chip, and ϕ is the code phase (that is, the current chip including the fractional chip).

Once the desired accumulation time is reached, the chip shapes are normalized by two methods. First, each transition is normalized according to how many transitions occurred for that PRN. As there are ideally an equal number of each transition per PRN sequence, this is not the case. Different PRNs would lead to relative differences between transitions due to some occurring more often than others, leading to a bias between accumulated chip shape transitions. To prevent this bias, each transition is normalized by dividing by how many transitions of that type occurred during

accumulation.

Second, it is desired to have some method of normalizing power to facilitate chip shape comparisons. If accumulation time increases by a factor of ten, it is expected that the magnitude of the chip shape increases by about ten as well. It is desired to normalize the magnitude to range from -1 to 1, allowing comparison across different accumulation times and conditions. Since normalizing all transitions individually to fit inside this range would undo the normalization performed on each transition, one transition is used to normalize power for all.

Arbitrarily, transitions that remain 'high' are used to perform the power normalization here. For GPS signals, this transition is the $[1, 1]$ transition, and for Galileo signals, it is the $[-1, 1]$ transition (because of the Composite Binary Offset Carrier (CBOC) modulation). As the signal stabilizes around the center of the transition, the mean of this range is used to normalize the power for all chip shapes.

The result of this process is coherent integration of signal transitions, for as long as the designer chooses. Example results are shown in Figures 22 and 23.

3.4.2 Determining Signal Quality from Chip Shapes

The signal interference of interest introduces noticeable deformations to the signal transitions, which will be reflected in the chip shapes. It is desirable to exploit the effects of interference on chip shape using computationally-inexpensive methods. Though the majority of the chip shape is derived from the in-phase component, valuable information lies in the quadrature-phase when the signal's phase changes, as seen in Figures 22 and 23. Therefore, quadrature-phase is included in calculating the Chip Shape Response (CSR), using

$$\mathbf{CSR}_m(n) = \text{sign}\left[\mathbf{I}_m(n) \times \sqrt{\mathbf{I}_m(n)^2 + \mathbf{Q}_m(n)^2}\right], \tag{24}$$

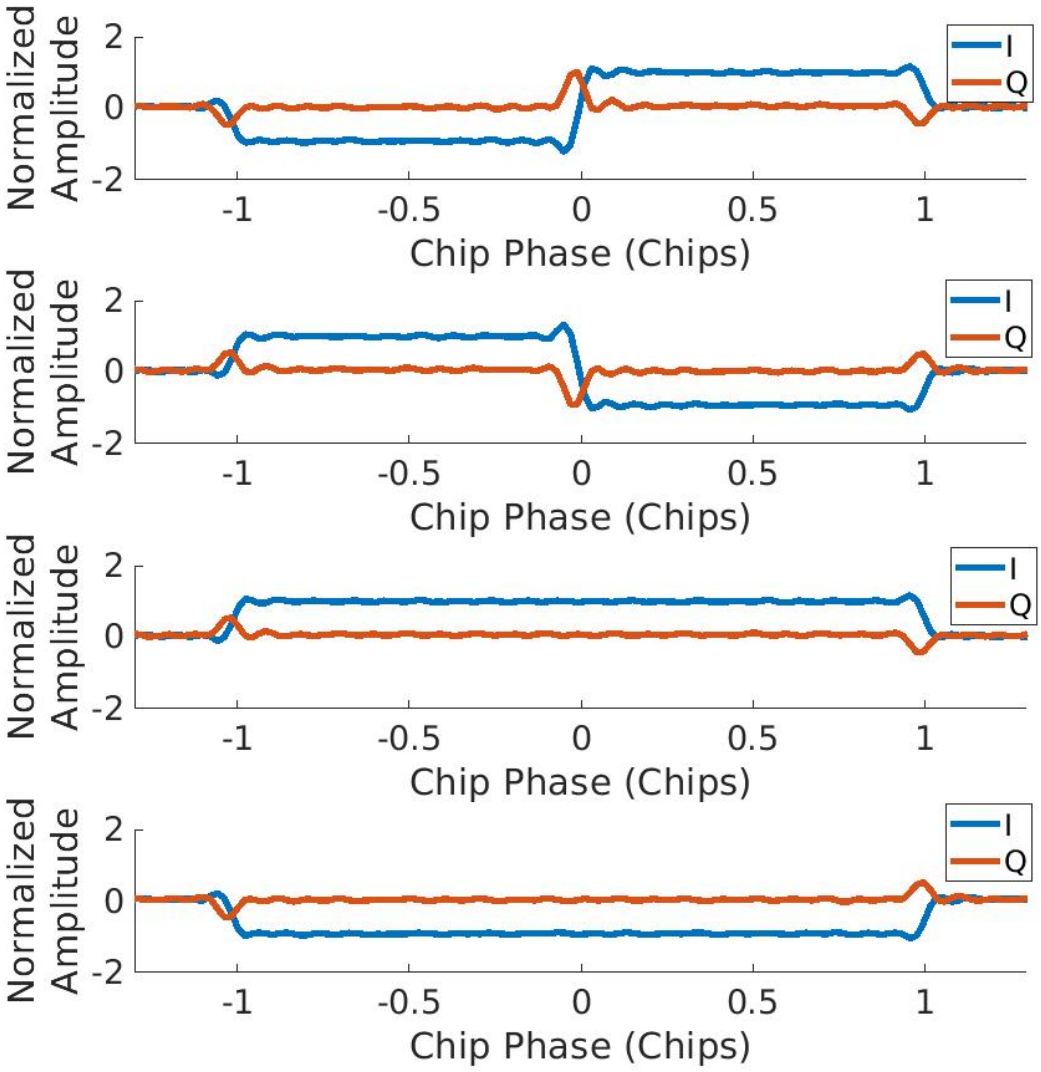


Figure 22: The four possible chip transitions for GPS PRN 1 for a 60 sec integration time and 50 bins per chip. Both in-phase (I) and quadrature-phase (Q) components are shown. Collections were in a 'clean' environment with low chances of signal interference.

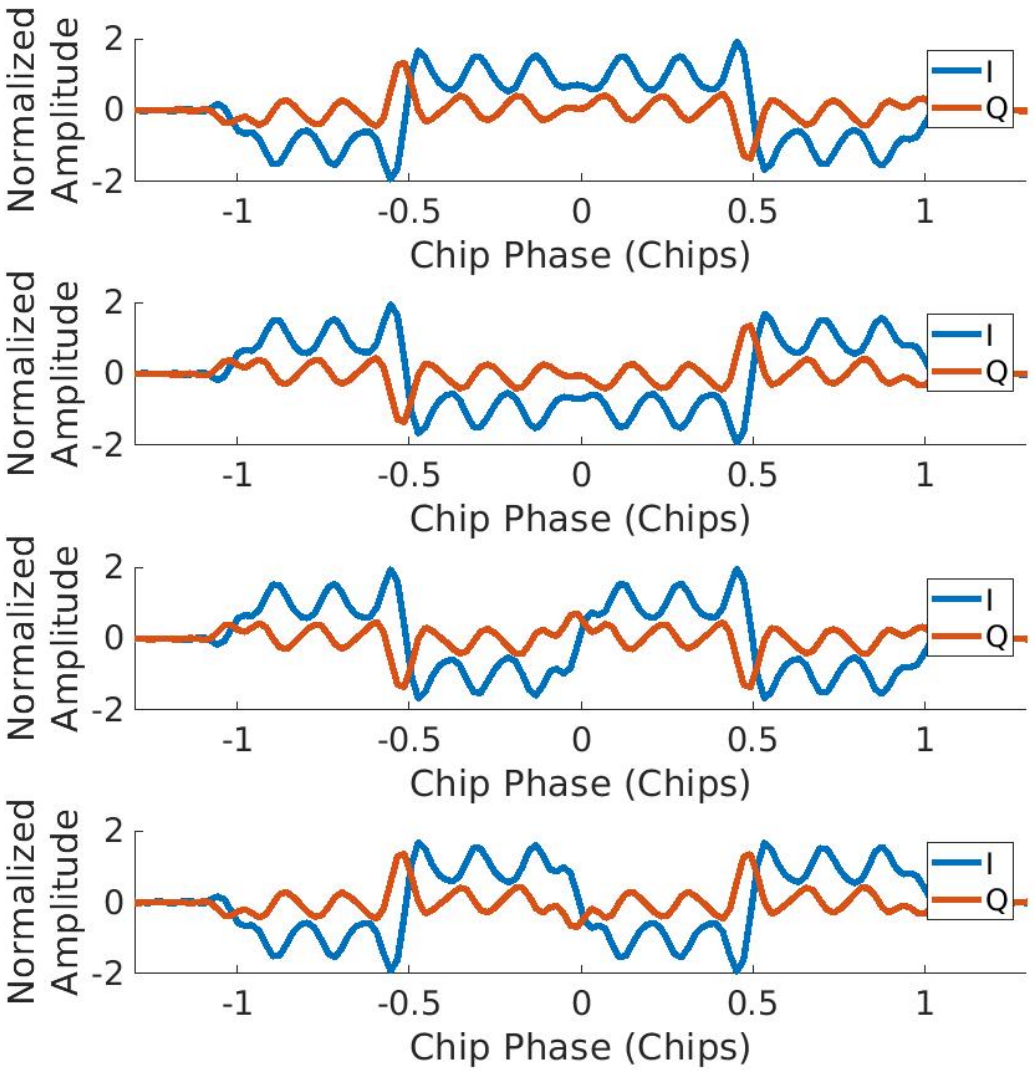


Figure 23: All four chip transitions for Galileo E1C ('pilot') 27 for a 60 sec integration time and 50 bins per chip. Collections were in a 'clean' environment, and both in-phase (I) and quadrature-phase (Q) components are shown. As the 25MHz collectionbandwidth does not fully capture the Galileo E1 band, higher-order harmonics of the high-frequency subcarrier are suppressed. Here, the signal is normalized according to the CBOC modulation scheme.

where **I** and **Q** refer to In-phase and Quadrature-phase components of the m^{th} acquired chip transition samples, and n=1, 2, ... , N_C where N_C is the total number of CSR samples in the chip shape response.

3.4.2.1 Defining the Chip Shape Deformation Metric

To determine signal quality, newly acquired chip shapes are compared to a Reference Chip Shape Response (RCSR) of the same signal. The RCSR is acquired through long chip shape integration of data collected in a 'clean' environment, using the same RF front-end. For a given chip transition m, the chip shape's deformation, $\mathbf{D}_m$, is measured as the euclidean distance between samples of the CSR of interest and RCSR samples over all points N_C, shown in (25). Though accompanying plots may show data outside of the chip transition, it is important to only include points within the transition of interest when computing these metrics, otherwise unnecessary noise is included in the results.

$$\mathbf{D}_m = \frac{1}{N_C} \sum_{n=1}^{N_C} |\mathbf{CSR}_m(n) - \mathbf{RCSR}_m(n)| \tag{25}$$

To characterize discrimination between non-distorted and distorted CSRs, the deformation of all four possible chip transitions ($\{[-1,-1],[-1,1],[1,-1],[1,1]\}$) are used to generate a single average metric called the Chip Shape Deformation Metric (CSDM), calculated as

$$\mathbf{CSDM} = \frac{1}{4} \sum_{m=1}^{4} \mathbf{D}_m \ . \tag{26}$$

The CSDM metric in (26) provides a measure of how far the received signal deviates from a non-distorted signal. As chip integration time increases for calculating the CSRs, the CSDM metric generally improves (decreases) due to increased suppression

of noise effects. However, this improvement comes at the expense of yielding a slower SQM update rate.

3.5 Multipath Simulation Design

Simulation of a GNSS signal allows the CSDM to be evaluated under ideal conditions and under conditions degraded by multipath. To capture the characteristics of each SQM method, data is simulated using ideal non-filtered signals that are not deteriorated by noise. A multipath signal is generated as a copy of the LOS signal with attenuated power and some delay. It is very significant that the delay acts on both the code and carrier, as the interaction of the two greatly impacts signal deformation. For a given delay in meters, the code and carrier delay are computed as

$$\begin{aligned} \textbf{Code Delay} &= \varepsilon \left(\frac{r_{chip}}{c} \right) \ , \\ \textbf{Carrier Delay} &= \varepsilon \left(\frac{2\pi f_c}{c} \right) \ , \end{aligned} \tag{27}$$

where ε is the delay in meters, r_{chip} is the code chipping rate, f_c is the center band frequency, and c is the speed of light. The code delay is computed in chips and the carrier delay is computed in radians.

IV. Results and Analysis

4.1 Preamble

The goal of these results is to show a differentiation between clean and distorted signals by computing metrics related to acquired chip shapes. As discussed in Chapter III, reference chip shapes acquired from data in a clean environment, using the same signals and Radio Frequency (RF) front-end, are used to determine a signal's quality. The Chip Shape Deformation Metric (CSDM) defined in section 3.4.2.1 is a direct metric of a signal's variation from some nearly-ideal reference, and the performance analysis is shown below.

4.2 Signal Quality Monitoring using the CSDM

The CSDM is implemented on real collected data sets and the performance is analyzed below. Analysis includes comparing the metric on live-sky received data in clean and multipath environments, comparing the metric to the previously-published eight-point correlator method described in [1], and comparing the CSDM across varying integration times.

4.2.1 Comparing CSDM for Clean and Distorted Signals

The CSDM will be used to discriminate clean signals and deformed signals, with the goal of maximizing the distance of metrics between the two. The reference chip shapes used throughout are acquired from a 'clean' collection, using $T_{int} = 60$ sec integration to suppress noise. All references are of the same satellite signal of interest to avoid errors induced by Pseudo-Random Noise (PRN) code biases. Both data collections to be tested differ from the one used to generate the reference, but have the same satellite constellation available.

A T_{int} = 1-second chip shape integration time is used to compute the CSDM, corresponding to a *1 Hz* update rate of signal quality. As seen in Figure 25, the CSDM metric reflects signal distortions in its magnitude when compared to Figure 24. It is also notable that data was collected by a stationary receiver, yet multipath interference caused varying levels of signal distortion over a relatively short period of time. The variations are attributed to satellite dynamics changing the multipath environment.

Figure 24 not only shows smaller values for CSDM, but much less variation due to the low-multipath environment. The non-zero values for CSDM are a result of the T_{int} = 1-second integration time and the accompanying noise. Comparing the chip shape in Figure 24 to those in Figures 22 and 23 (which use T_{int} = 60-second integration time) illustrates the dependency on long integration times to sufficiently suppress noise.

For weaker interference, it becomes more difficult to differentiate between clean and deformed signals as the CSDM metric approaches the level that is predicted of clean signals. Sensitivity to interference should then improve with longer integration time, as the chip shape of a clean signal approaches the reference and the corresponding CSDM approaches zero. Integration time then determines the update rate of signal quality and the amount of distortion that is distinguishable, to be chosen by the requirements of the system. The claim that integration time directly improves CSDM performance assumes that the integration time does not outlast the distortion.

CSDM Output and Corresponding Chip Shape for GPS PRN 1 in a 'Clean' Signal Environment

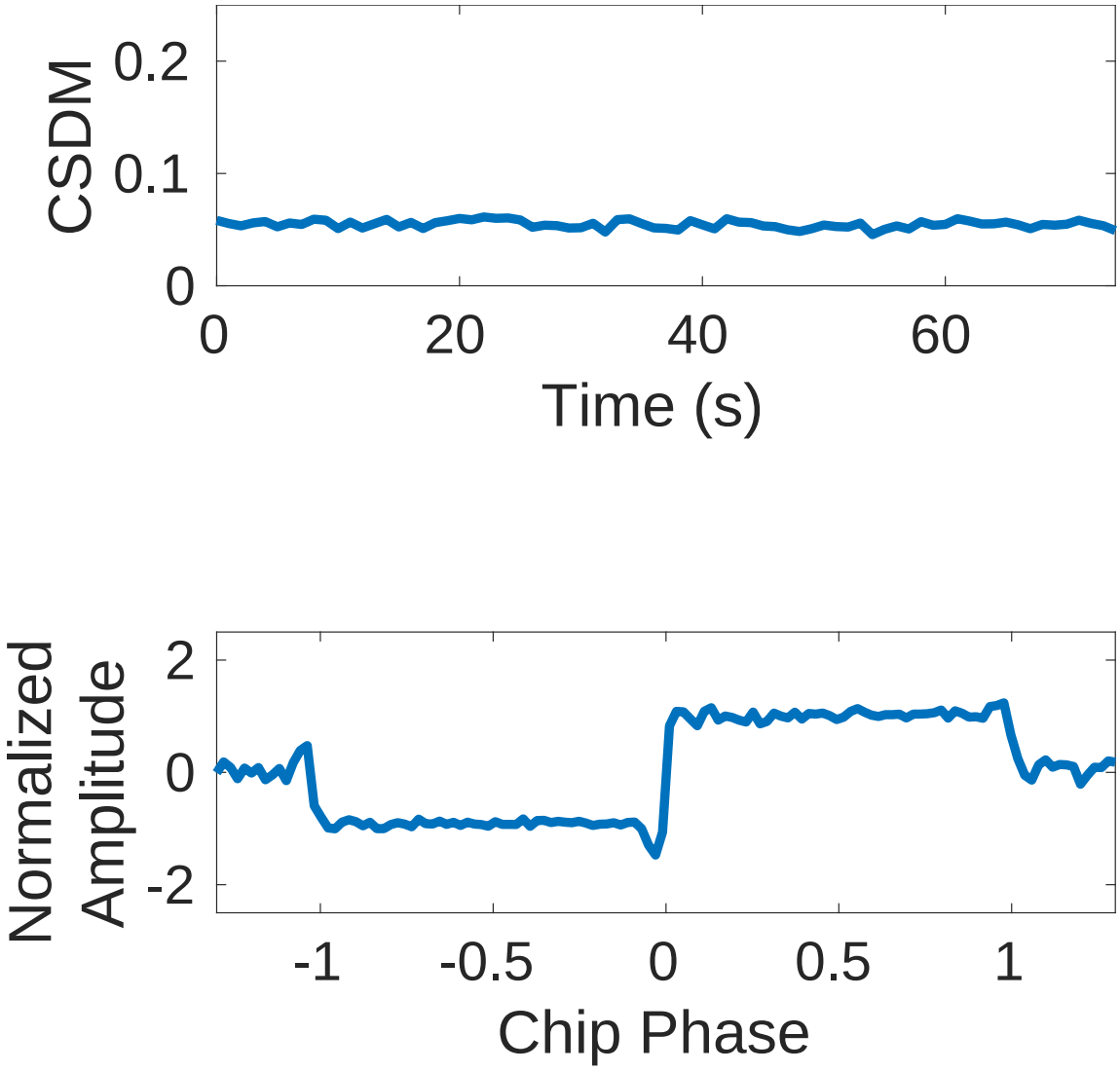


Figure 24: The CSDM and the corresponding rising-edge Chip Shape Response (CSR). These were computed for the Global Positioning System (GPS) PRN 1 signal, with data collected in the 'clean' environment. Relative to Figure 25, the absence of multipath is evident in both the lower CSDM values and minimally distorted chip shape response. The above uses 50 bins per chip and T_{int} = 1-second integration time.

CSDM Output and Corresponding Chip Shape for GPS PRN 1 C/A in a Multipath Environment

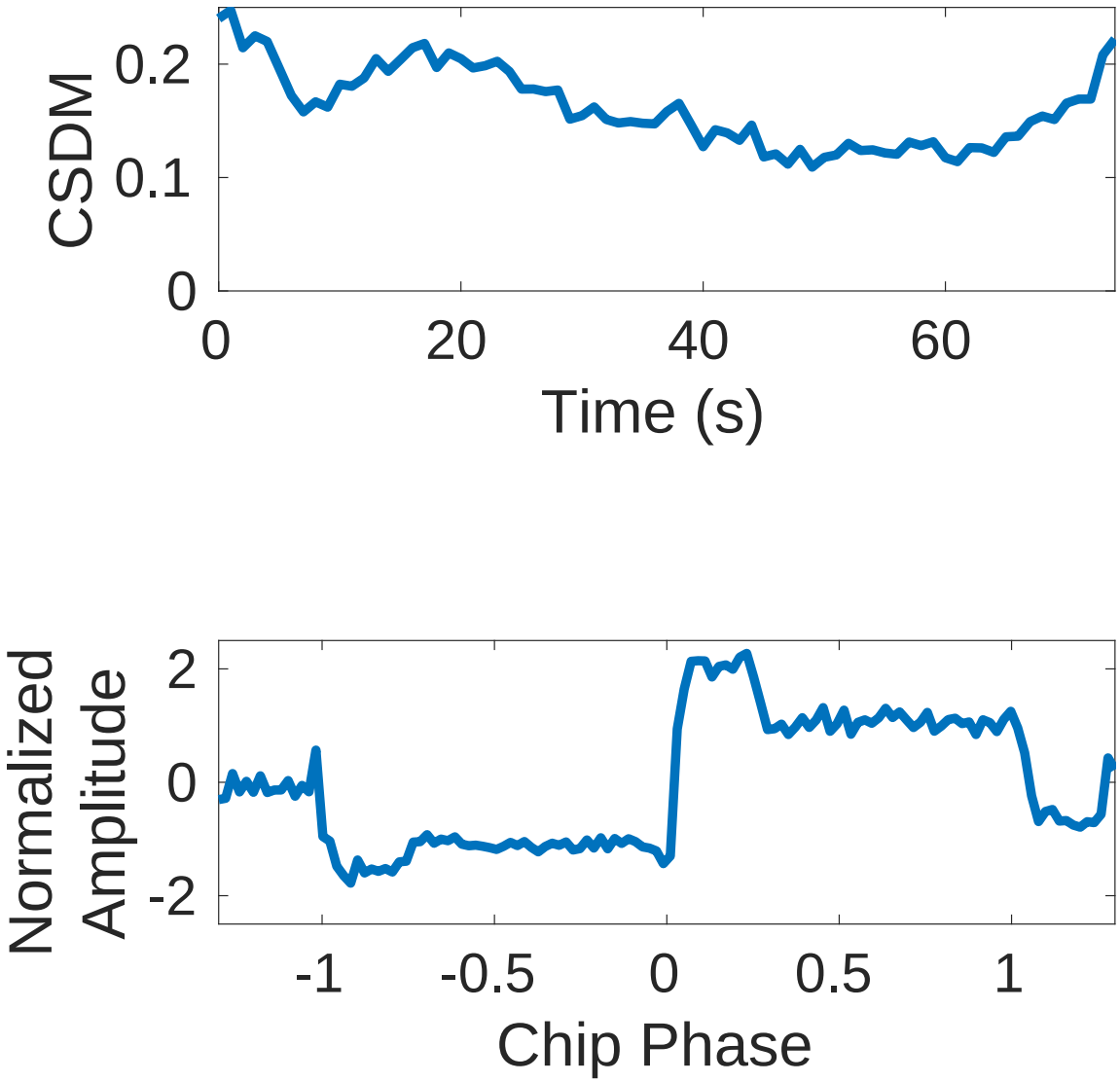


Figure 25: The CSDM and a corresponding rising-edge CSR. These were computed for the GPS PRN 1 signal with data collected in a multipath environment. Relative to Figure 24, the effect of multipath is clearly evident in both the higher CSDM values and increased distortion in the chip shape response. The above uses 50 bins per chip and T_{int} = 1-second integration time.

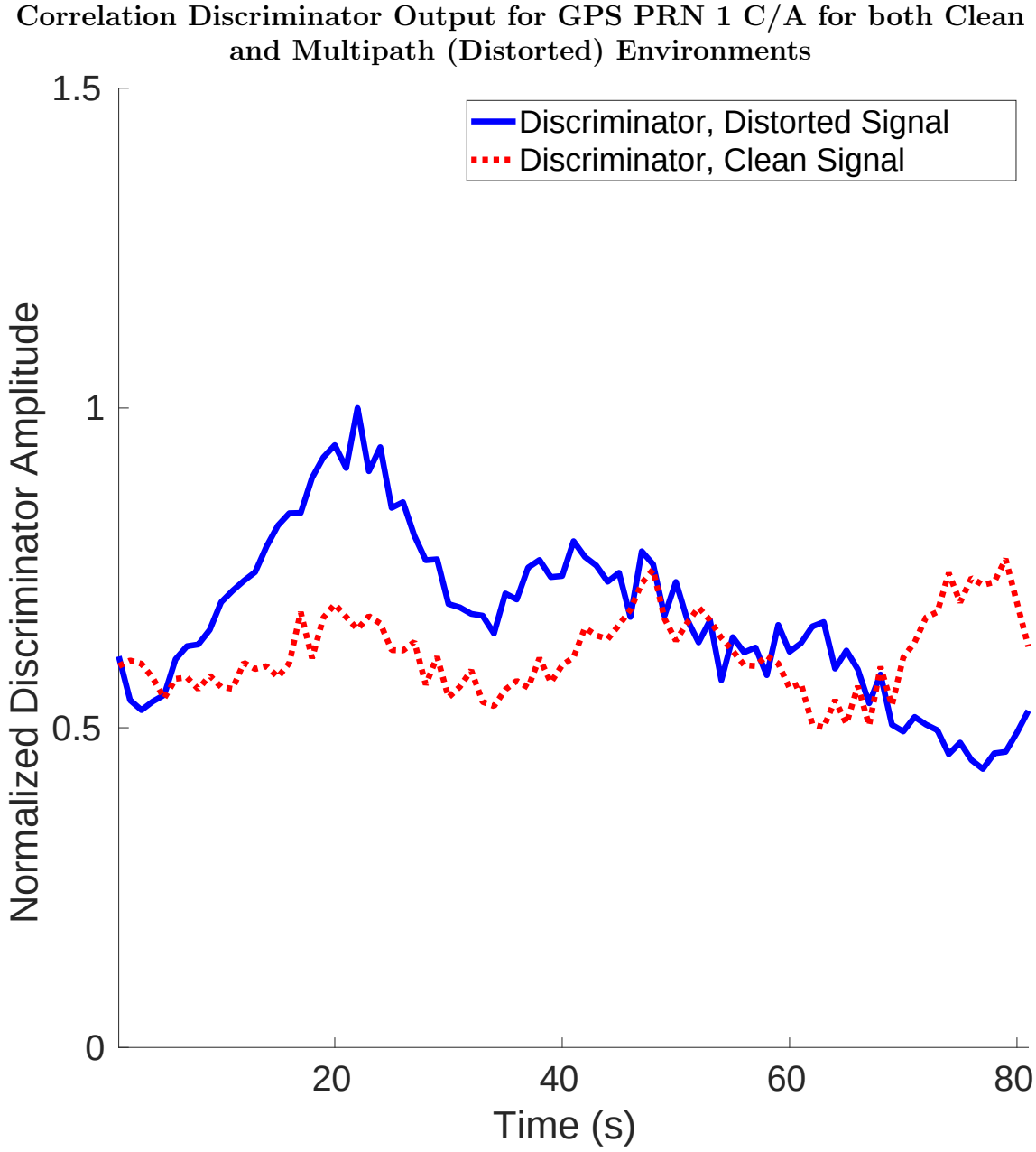


Figure 26: Computed correlation discriminator [1] performance using 8 code correlators and $T_{int} = 1$ sec integration time for live-sky data. The difference between correlators is taken, and a metric is computed using a matrix based on the ideal response. The magnitude is normalized by the peak of the metric for the deformed signal. It is seen that there is unclear distinction between the two signals for this example.

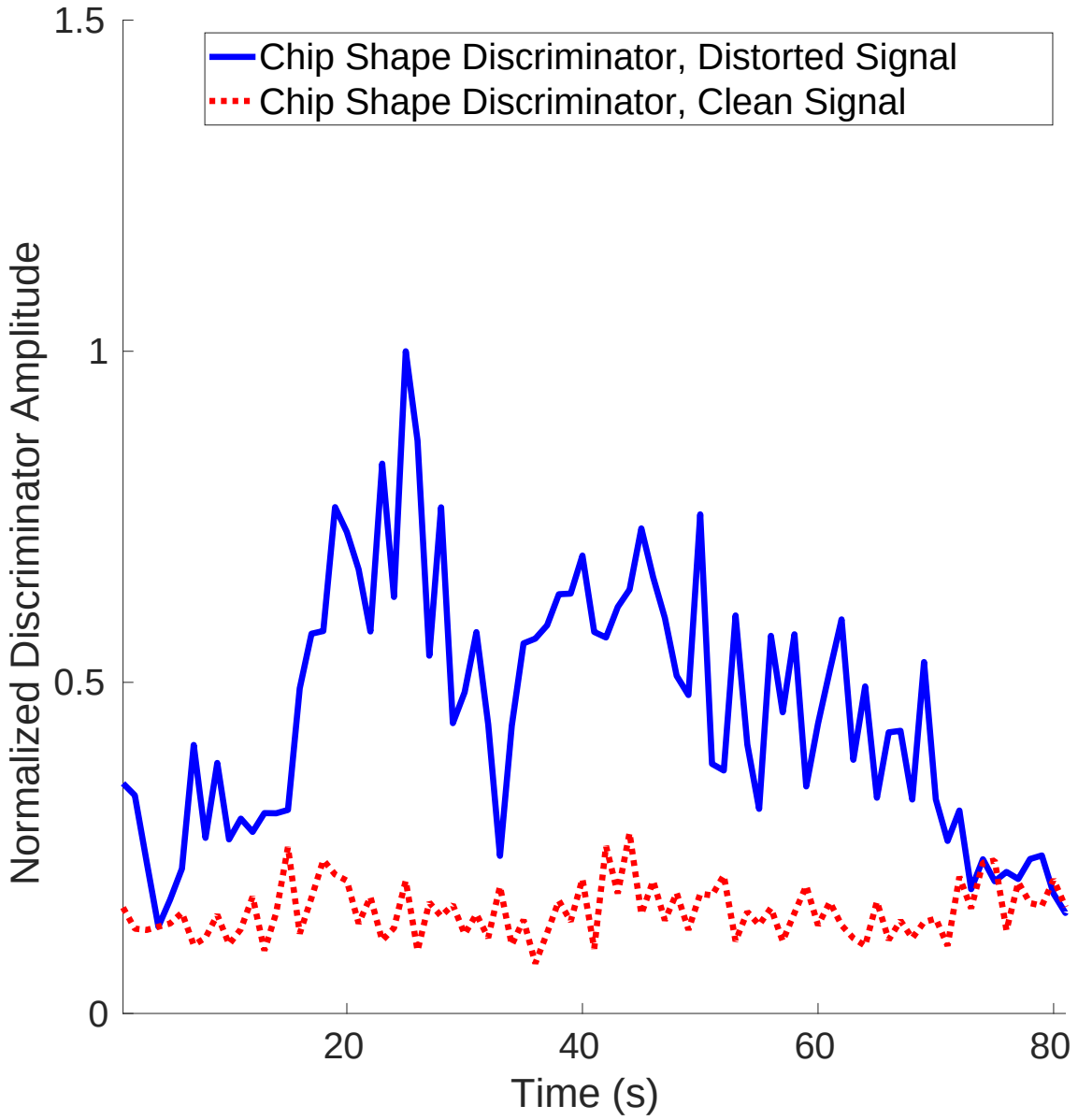


Figure 27: Computed chip shape discriminator using $T_{int} = 1$ sec integration time, 50-bins per chip, and the deformation monitor algorithm presented in [1] for live-sky data. The CSR is computed and used to calculate a metric using only the points corresponding to [0,0.125] chips. The magnitude is normalized by the peak of the metric for the deformed signal. Compared to Figure 26, there is clear distinction between a signal with deformation and one without.

4.2.2 Comparing CSDM to 8-Point Correlation Triangle

The CSDM is compared with a previously published approach presented in [1] for both live-sky data and simulated data with multipath present.

4.2.2.1 Comparing SQM Methods for Live-Sky Example

As discussed, a prominent Signal Quality Monitoring (SQM) technique is to implement eight correlators instead of the typical three, with varying spacings to outline the correlation curve. One method of determining signal quality is to compute the difference between adjacent correlation points and monitor for biases [1]. In summary, an algorithm takes a set of data points from clean collections and computes the covariance matrix with respect to the data points in the response. This is used to compute a "fusing matrix," which transforms each input response to a deformation metric.

To compare the plurality of correlators approach [1] with the CSDM method introduced here, both techniques were implemented using the *same* input data sets. One set of data is in a multipath environment, and the other is in a clean environment, such that the performance of the metrics may be compared by how they differ in each scenario. To demonstrate whether chip shapes offer more information about deformation, the algorithm presented in [1] is implemented using the plurality of correlators response as well as using the computed CSRs. Though the computed "fusing matrix" is elevation-dependent [1], the data used is over a short period of time such that this effect is negligible. Finally, as the correlation approach has a maximum delay of 0.125 chips, only the points of the CSR corresponding to [0, 0.125] chips will be used. If the entire range was used, the chip shape method would observe deformation for a wider range than the correlator approach.

Results are shown in Figures 26 and 27, showing the introduced CSDM as a com-

petitive means of achieving SQM. Compared to the plurality of correlators approach presented in [1], the CSDM establishes clear, consistent distinction between clean and deformed received Global Navigation Satellite System (GNSS) signals.

4.2.2.2 Comparing SQM Methods in a Simulation Environment

The live-sky data demonstrates an instance where the CSR method is favored over the plurality of correlators approach presented in [1], but it is desired to gain insight into how each metric performs for various signal deformation scenarios. Through simulation, the performance of the metrics are compared across various multipath signal delays to demonstrate the conditions that differentiate them. The metric is computed using the algorithm presented in [1], as described previously. It is desired to see the effects of multipath delay on each of the SQM metrics, so the delay is set to range from 0 to +1.5 chips. The delay is increased at a very slow rate such that there are several data points per carrier cycle. Arbitrarily, five points per cycle are chosen. For an integration time of $T_{int} = 20$ms, this corresponds to an increasing delay at a rate of about 1.9m/s.

To compare each SQM metric, a simulation is designed such that the multipath signal has half of the power of the line-of-sight (LOS) signal throughout. A separate simulation is generated with no multipath interference to provide a reference for each method, and the metrics are computed by how much the responses differ from the respective reference.

As the approach presented in [1] considers correlators with a maximum delay of 0.125 chips, the points from the CSR corresponding to this range will be used. Next, this range will be increased for the CSR to illustrate how the observable deformation depends on the range of points used.

From the results shown, it is concluded that the CSR provides more information

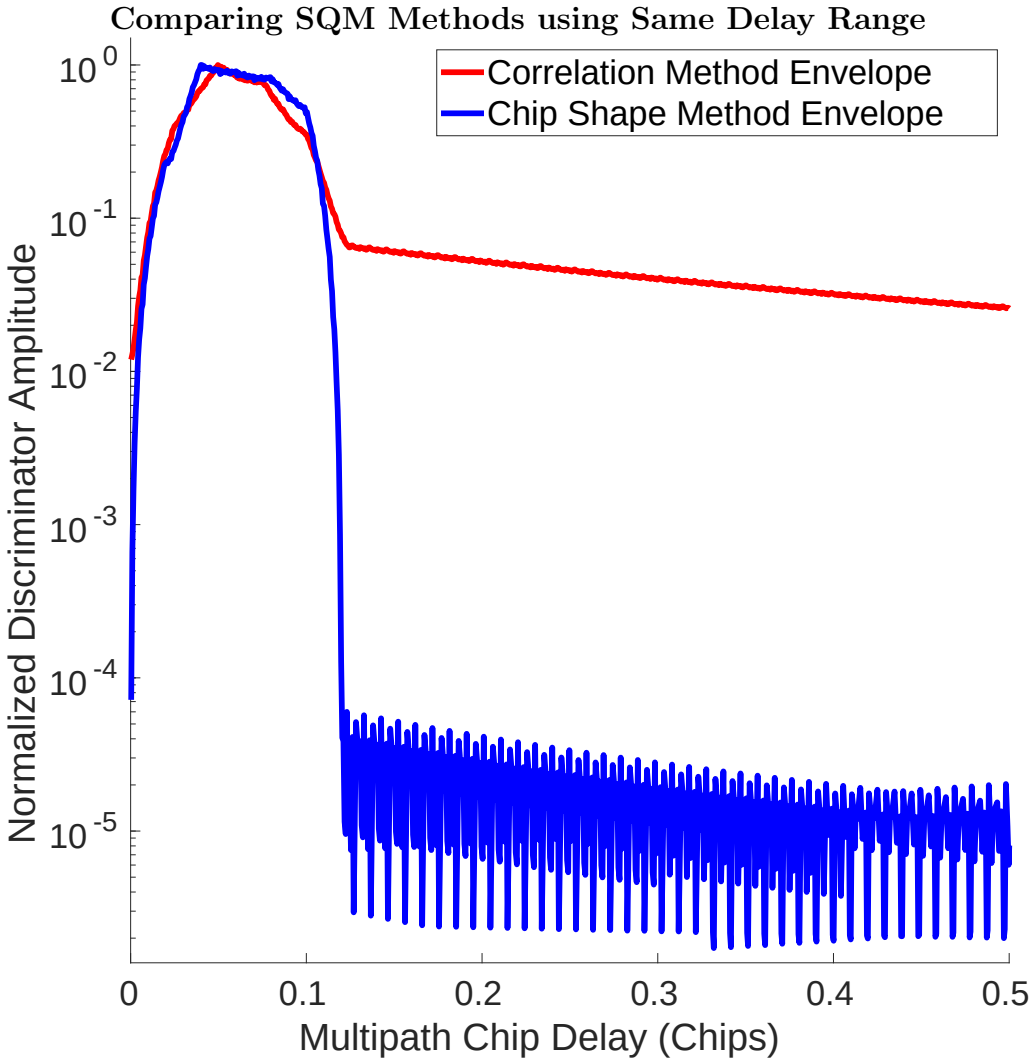


Figure 28: The plurality of correlators approach presented in [1] compared with the chip shape discriminator for increasing multipath delay. Here, the points from the CSR corresponding to [0, 0.125] chips are used to match the maximum delay in the correlator approach. Both metrics are normalized by their peak and the envelope of the curves are taken. The chip shape method provides significantly more distinction between clean signals and deformed signals, seen by the much larger range and the sharper slope to and from the peak.

than the plurality of correlators response. There is a sharp rise and fall from the specified range, showing clear distinction between a signal with and without deformation. When a larger range of points in the CSR is used, deformation is observable for a larger delay span as expected.

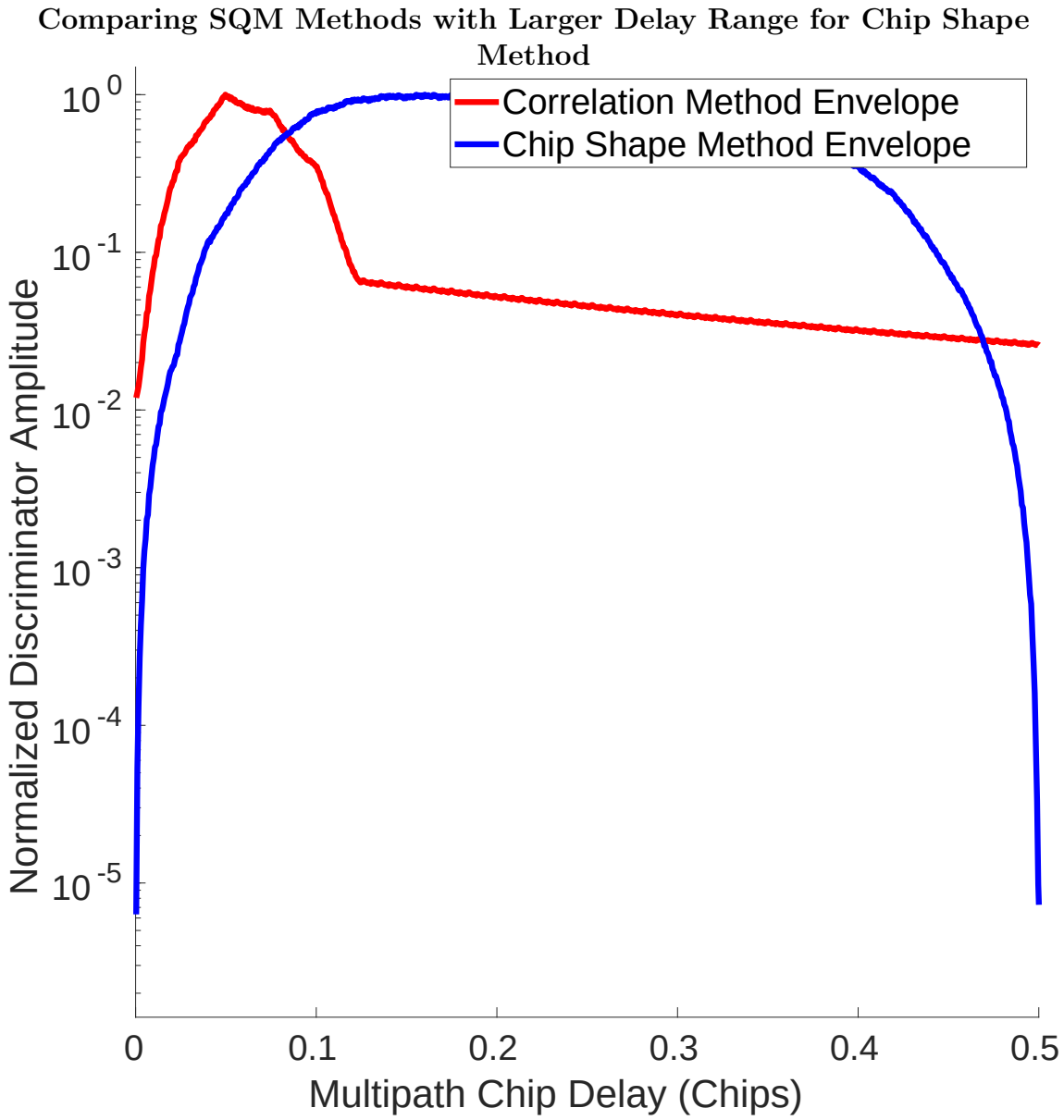


Figure 29: The plurality of correlators approach presented in [1] compared with the chip shape discriminator for increasing multipath delay, using a larger range for the chip shape method. Here, the points from the CSR corresponding to [0, 0.5] chips are used, providing more range than the correlator approach. Both metrics are normalized by their peak and the envelope of the curves are taken. Compared to Figure 28, the chip shape approach now observes deformation over a much larger delay span. This illustrates the dependency of observable deformation on the scope of points used in the CSR.

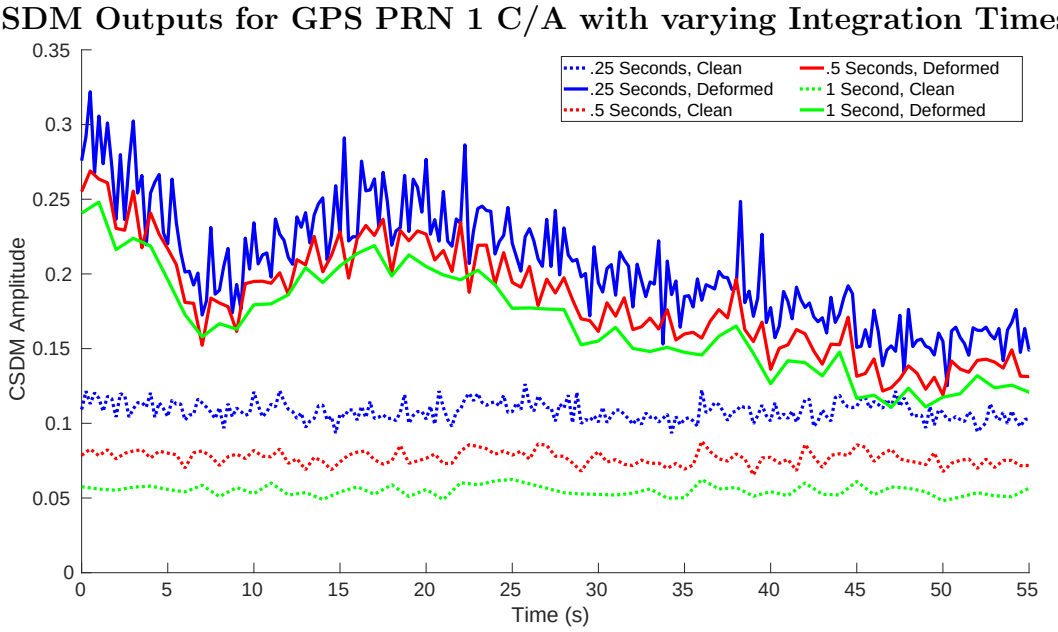
4.2.3 CSDM Dependency on Integration Time

As the update rate of signal quality is directly determined by the chip shape integration time, it is desirable to analyze how the performance is impacted as well. Here, performance is determined by how well a distorted signal is differentiated from a clean one. The effect of integration time is analyzed using a live-sky measurement with heavy multipath present, and one with negligible interference.

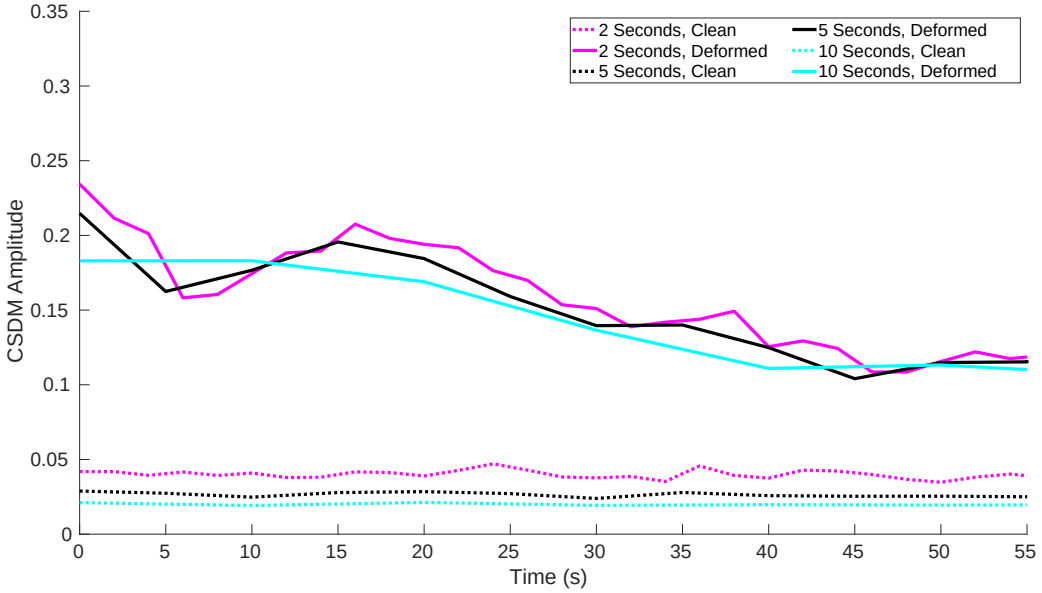
The concern with long integration times is that it may not respond quickly enough to meet the requirements of Safety of Life (SoL) applications that require immediate alerts in the case of unsafe signal quality. Furthermore, if the interference doesn't last as long as the integration time used, the effects seen in the chip shape may be subdued by averaging. Short integration times may supply a faster update rate, but are more strongly affected by noise. As noise increases, the clean and deformed chip shapes become more similar and the CSDM is not as beneficial of a metric. This relationship is explored across six integration times on the same two data sets, using the same reference chip shape.

To compare performance across different integration times, the ratio of the CSDM for a deformed signal to the CSDM for a clean signal is taken at each point. This captures the magnitude of the divide between the two, with a bigger divide signifying better identification of deformed signals.

As seen in Figures 30 and 31, longer integration times cause lower CSDM metrics in general, but give better performance of identifying deformed signals overall. The improvement is fully attributed to the suppression of noise, as the average CSDM for clean signals approaches zero. The interference that is being observed in the 'distorted' data set is present throughout in this example, even though it is changing. This interference allows the CSDM to simply improve with longer integration times. This is not universal, however, as some interference may be present for relatively short



(a) a) Integration Times of T_{int} = 0.25, 0.5, and 1-second.



(b) b) Integration Times of T_{int} = 2, 5, and 10-seconds.

Figure 30: The CSDM output using a fixed 50 bins per chip and varying integration times. The CSDM is computed for GPS PRN 1 on a set of data inflicted by heavy multipath and on one in a clean environment. As expected, the CSDM for 'clean' data approaches zero with longer integration times, as the acquired chip shapes become closer to the reference acquired through long integration. The CSDM for the 'distorted' data set also tends to decline with longer integration, fortifying the conclusion that noise contributes greatly to the magnitude of the CSDM.

periods of time, making it disadvantageous to integrate for longer.

Deformed-to-Clean CSDM Ratios for GPS PRN 1 C/A with varying Integration Times

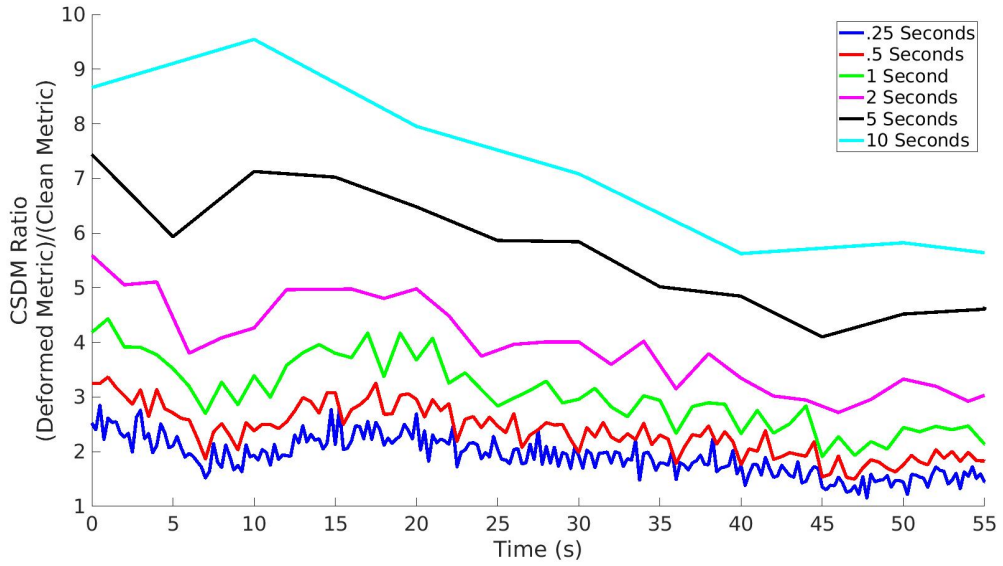


Figure 31: The ratio of the CSDM for deformed signals to the CSDM for clean signals implemented across various integration times. The ratio is computed for the data shown in Figure 30. As the data example used includes an extreme case of signal distortion, the desired conclusion to be made is about the dependence of CSDM on integration time. As seen above, the ratio increases with integration time, signifying that longer integration times allow for better discrimination between clean and deformed signals. This is attributed to the better suppression of noise that accompanies long integration.

4.3 System Runtime Performance

The system described in this document is implemented in a software environment using C++ source code, and compiled by the free, publicly available GNU compiler. The methods described are intended to encourage efficient runtime, such that this analysis may be implemented in a real-time system. Data is collected by executing the program on the computer workstation specified in Table 2.

The results shown are derived from using the same data, but filtered and down-sampled accordingly. Runtime is computed by querying the elapsed time from when data is read to when it is fully processed. Here, "fully processed" refers to a full iteration of both the tracking loops and the chip shape accumulators. Runtime will only start to be observed once chip shapes are being computed, which begins after the tracking loops synchronize to the signal polarity. Querying the execution time only once chip shapes are being computed will ignore the initial time it takes to acquire and pull-in signals, but offers insight into the long-term, steady-state performance.

For this measurement, all data is made available to the system at once such that it has the ability to execute faster than real-time if it is capable. The sample rate varies from 5 MSPS to 25 MSPS, and the number of active tracking channels ranges from one to seven. Time is queried for every 50 ms of data that is processed, and the runtime performance is computed by

Table 2: Hardware configuration of the host computer.

Computer Model:	HP Z840
Operating System:	Ubuntu 18.04.2 LTS
Processor Type:	Intel Xeon CPU E5-2687W v3 (25 MB, 3.10 GHz)
Number of Processing Cores:	10
Main Memory Type:	Micron MTA36ASF2G72PZ (DDR4 SDRAM, 16 GB, 288 RDIMM)
Total Main Memory Size:	128 GB
Secondary Memory Type:	TOSHIBA DT01ACA200 HDD (7200 RPM)
Secondary Memory Size:	2 TB

$$\textbf{Runtime Performance} = \frac{\textbf{Execution Time}}{\textbf{Time Length of Processed Data}} \ . \qquad (28)$$

For values less than or equal to one, this metric describes a system that has the capability of operating in real-time. For values greater than one, the metric describes how many times slower the system operates compared to real-time.

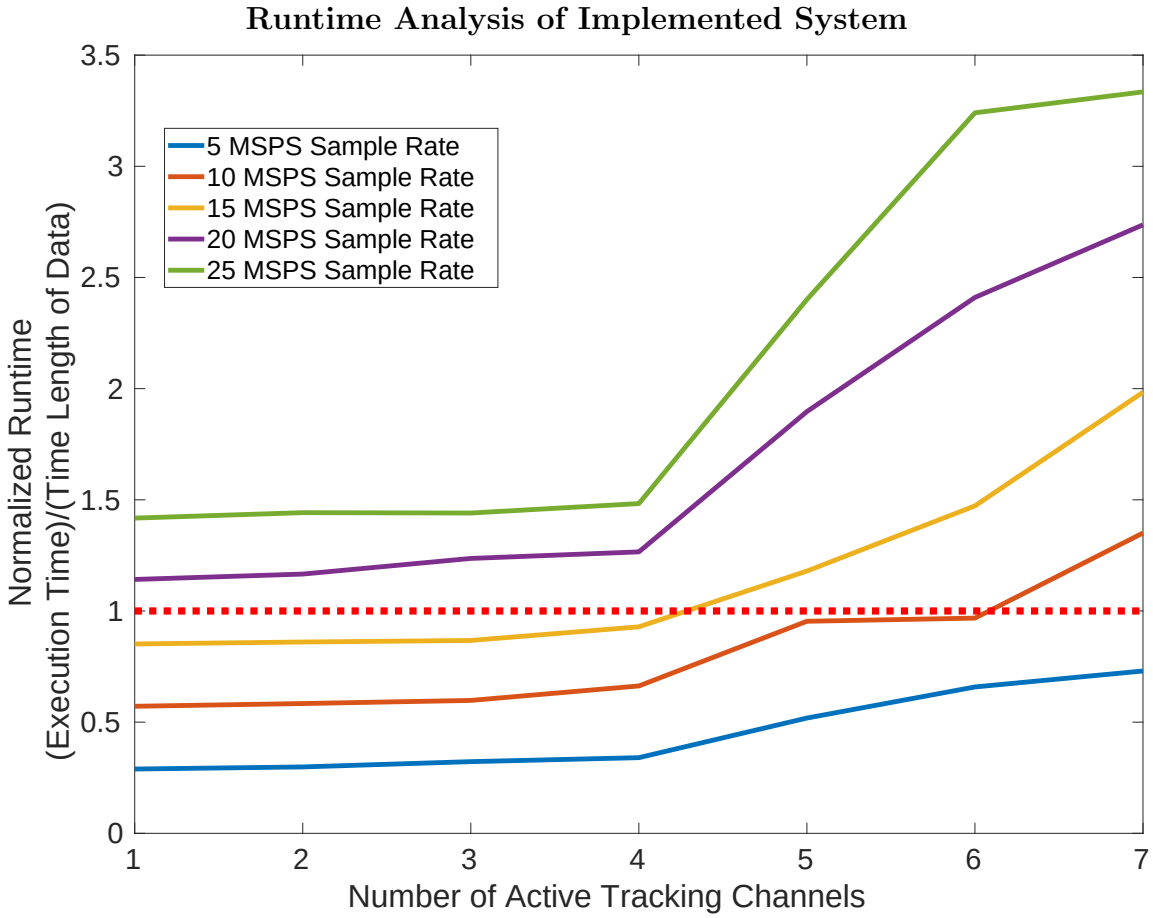


Figure 32: Runtime analysis plot of the implemented system. Execution time is measured for an increasing number of active tracking channels and for various sample rates. Data below the dashed red line denotes a system that has real-time capability. The mean run-times are shown above, but confidence intervals are not shown on this plot as the 99% intervals are all contained within 3% of the respective mean. As expected, the execution time scales with increasing sample rates. As tracking channels are run on parallel threads, execution time does not notably increase with added channels initially. However, once all available cores are being used, the execution time begins to scale with every added channel. In this system, the cores are all occupied after four channels are added, seen by the sudden change in slope.

V. Conclusions

This research demonstrated effective Signal Quality Monitoring (SQM) by analyzing and discriminating distortions seen in the chip shape domain of Global Navigation Satellite System (GNSS) signals. As multipath interference and faulty satellite hardware still remain a threat to the integrity of GNSS Safety of Life (SoL) applications, accessible SQM methods are of growing demand. The demonstration of implementing SQM to function in cooperation with a live software receiver shows the potential for wide applicability, as it does not require the use of specialized hardware to function.

The results demonstrate a successful way of discerning between clean and deformed signals by exploiting the chip shape domain. Deformations that are present on the unobservable raw signal are observable on a time-averaged representation of the signal's transitions. This property is summarized in one metric, the Chip Shape Deformation Metric (CSDM), by finding the euclidean distance between chip shapes and near-ideal references. By computing this metric on live-sky data, it is demonstrated that the CSDM may be used to successfully identify unsafe signals. The results shown are consistent with the eight-point correlation triangle apparatus described in [1], and the CSDM method presented here excels by discriminating between clean and deformed signals.

In application, CSDM performance is able to be directly improved by using longer chip shape integration time, at the cost of the signal quality update rate. It would then be desirable to use the longest integration time possible, while providing an update rate that is tolerable by the system.

5.1 Future Work

This research is supplemental to a strong foundation of works, specifically those related to Radio Frequency Distinct Native Attributes (RF-DNA) [4, 5, 6, 7]. It is desirable to continue studies focused on real-time signal tracking and chip shape acquisition, while applying the methods of RF-DNA to enhance chip shape classification. Examples for augmenting the presented research are described below.

- Using a machine-learning algorithm, acquired chip shapes can be used to model the nominal waveform received. Machine-learning will significantly increase sensitivity to distortions, directly improving monitoring performance.

- The receiver architecture may be developed further to compute position solutions, providing a means of relating signal deformations to positional error. Computing positional error would benefit systems that require a known level of position precision, as chip shape integration times may be chosen based on the sensitivity required.

- To improve runtime performance, a system could track data at a lower sample rate than the data used to compute chip shapes. Processing data at a lower sample rate could be implemented by filtering and downsampling received data, but presents the challenge of implementing a filter efficiently.

- Implementing this system on a computer workstation with a higher number of processing cores will allow for the tracking of more channels without severely impairing runtime performance. For example, the Advanced Micro Devices (AMD) Ryzen Threadripper 3990x is a cost-effective 64-core processing unit [36]. It is expected that this many cores could execute at least 24 channels without impairing runtime performance.

Bibliography

1. F. Liu, M. Brenner, and C. Y. Tang, "Signal deformation monitoring scheme implemented in a prototype local area augmentation system ground installation," in *Proceedings Institute of Navigation (ION-GNSS)*, pp. 367–380, 2006.

2. M. Graham, "Gps use in u.s. critical infrastructure and emergency communications," *United States Technical Training Institute (USTTI)*.

3. "National space policy," *Office of Space Commerce*, Jun 2010.

4. C. Dubendorfer, B. Ramsey, and M. Temple, "Zigbee device verification for securing industrial control and building automation systems," in *International Conference on Critical Infrastructure Protection*, pp. 47–62, Springer, 2013.

5. J. Lopez, N. C. Liefer, C. R. Busho, and M. A. Temple, "Enhancing critical infrastructure and key resources (cikr) level-0 physical process security using field device distinct native attribute features," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1215–1229, 2017.

6. D. R. Reising, M. A. Temple, and J. A. Jackson, "Authorized and rogue device discrimination using dimensionally reduced rf-dna fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 1180–1192, June 2015.

7. C. M. Talbot, M. A. Temple, T. J. Carbino, and J. A. Betances, "Detecting rogue attacks on commercial wireless insteon home automation systems," *Computers & Security*, vol. 74, pp. 296–307, 2018.

8. J.-B. Pagot, P. Thevenon, O. Julien, Y. Gregoire, F. Amarillo-Fernandez, and D. Maillard, "Estimation of gnss signals' nominal distortions from correlation and chip domain," 2015.

9. G. Wong, R. E. Phelts, T. Walter, and P. Enge, "Alternative characterization of analog signal deformation for gnss-gps satellites," in *Proceedings of the 2011 international technical meeting of the institute of navigation, San Diego*, 2011.

10. L. R. Weill, "Theory and applications of signal compression in gnss receivers," in *Proceedings of the Institute of Navigation (ION GNSS)*, pp. 708–719, 2007.

11. E. G. N. S. S. Agency, "What is gnss?," —Online—. Available: https://www.gsa.europa.eu/european-gnss/what-gnss.

12. GPS, "The global positioning system," —Online—. Available: https://www.gps.gov/systems/gps/.

13. E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

14. J. F. Raquet, "Gps measurement error," 2019. EENG533 Class Lecture.

15. M. Irsigler, G. W. Hein, and B. Eissfeller, "Multipath performance analysis for future gnss signals," in *Proc., ION NTM*, 2004.

16. B. Townsend and P. Fenton, "A practical approach to the reduction of pseudorange multipath errors in a l1 gps receiver," in *Proceedings of the 7th International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City, UT, USA*, Citeseer, 1994.

17. G. S. Authority, "Galileo signal in space interface control document for open service," 2008. —Online—. Available: https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo-OS-SIS-ICD.pdf.

18. G. P. S. Directorate, "Systems engineering integration interface specification is-gps-200," 2019. —Online—. Available: https://www.gps.gov/technical/icwg/IS-GPS-200K.pdf.

19. M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread spectrum communications handbook*, vol. 2. McGraw-Hill, Inc, 1994.

20. B. Sklar, *Digital Communications: Fundamentals and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

21. J. Spilker Jr, "Gps signal structure and performance characteristics," *Navigation*, vol. 25, no. 2, pp. 121–146, 1978.

22. S. Gunawardena, "Gnss receiver design," 2019. EENG633 Class Lecture.

23. R. E. Phelts, T. Walter, and P. Enge, "Characterizing nominal analog signal deformation on gnss signals," in *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS), Savannah, GA*, pp. 1343–1350, 2009.

24. S. Gunawardena, M. Carroll, J. Raquet, and F. Van Graas, "High-fidelity signal deformation analysis of live sky galileo e1 signals using a chipshape software gnss receiver," *Proceedings of ION GNSS+, Tampa, Florida*, pp. 3325–3334, 2015.

25. G. Wong, R. E. Phelts, T. Walter, and P. Enge, "Characterization of signal deformations for gps and waas satellites," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation*, 2010.

26. P. Enge, R. Phelts, and A. Mitelman, "Detecting anomalous signals from gps satellites," *ICAO, GNSS/P, Toulouse, France*, 1999.

27. M. A. Brenner, "Apparatus for navigation satellite signal quality monitoring," Jan. 17 2006. US Patent 6,987,820.

28. M. S. Braasch, "Gps multipath model validation," in *Proceedings of Position, Location and Navigation Symposium - PLANS '96*, pp. 672–678, April 1996.

29. S. Gunawardena, J. Raquet, and M. Carroll, "Correlator beamforming for multipath mitigation in high-fidelity gnss monitoring applications," *Navigation*, vol. 66, no. 1, pp. 169–183, 2019.

30. J. R. Sanjeev Gunawardena and F. van Graas, "Chip transition-edge based signal tracking for ultra-precise gnss monitoring applications," *A Satellite with Personality*, 2015.

31. A. King, "Discriminating GNSS Chip Shapes using RF-DNA Techniques. Draft Master's Thesis," *Air Force Institute of Technology*, 2019.

32. Iridium, "Antcom iridium and gps antenna," 2019. —Online—. Available: https://www.iridium.com/products/antcom-iridium-gps-antenna/.

33. E. Research, "Usrp b205mini-i," —Online—. Available: https://www.ettus.com/all-products/usrp-b205mini-i/.

34. Google, "Google maps," 2019. —Online—. Available: https://www.google.com/maps/.

35. J. D. Gaeddert, "Liquid dsp - software-defined radio digital signal processing library," —Online—. Available: http://liquidsdr.org/.

36. "Amd ryzen™ threadripper™ 3990x processor," *AMD*.

Acronyms

A/D Analog to Digital. 6

AWGN Additive White Gaussian Noise. 25

BOC Binary Offset Carrier. 11, 14, 15

bps bits per second. 13, 14

BPSK Binary Phase Shift Keying. 11

C/A Course-Acquisition. vii, ix, x, 10, 12, 13, 16, 17, 19, 23, 28, 41, 44, 45, 47

CBOC Composite Binary Offset Carrier. 14, 18, 52

CDMA Code-Division Multiple Access. 10, 11

CDO Chip Domain Observable. 21

CNR Carrier-to-Noise Ratio. 45, 46, 47, 48

COTS Commercial Off-the-Shelf. iv, 2

CSDM Chip Shape Deformation Metric. viii, 55, 56, 57, 58, 63, 64, 67, 68, 69, 73

CSR Chip Shape Response. 52, 55, 59, 60, 63, 64, 65, 66

DLL Delay-Locked Loop. 40

DSSS Direct Sequence Spread Spectrum. 10, 11, 12, 37

EGNOS European Geostationary Navigation Overlay Service. 9

EML Early-Minus-Late. 24, 36, 39, 41

FFT Fast-Fourier Transform. 33, 35

FLL Frequency-Locked Loop. 40, 41

GNSS Global Navigation Satellite System. iv, vii, viii, 1, 2, 3, 4, 6, 7, 8, 10, 11, 15, 16, 20, 28, 30, 31, 32, 35, 48, 56, 64, 73

GPS Global Positioning System. iv, vii, ix, x, 5, 12, 13, 14, 16, 17, 19, 21, 23, 28, 30, 34, 37, 41, 42, 44, 45, 47, 48, 52, 59, 60

ICDs Interface Control Documents. 10

IF Intermediate Frequency. 5

IFFT Inverse Fast-Fourier Transform. 33

LAAS Local Area Augmentation System. 9, 21, 22

LOS line-of-sight. 22, 32, 48, 56, 64

MCPS mega-chips per second. 12, 14

MEO Medium Earth Orbit. 5

MSPS Mega-Samples per Second. 30, 36

NBP Narrow-Band Power. 45, 46

NCO Numerically-Controlled Oscillator. 42, 44

NP Normalized Power. 45, 46, 47

P-Lock Phase-Lock. 45, 46, 47

PLL Phase-Locked Loop. 40

PNT Position, Navigation, and Timing. iv, 1, 4, 5, 6, 7, 8, 9, 10, 12, 20

PRN Pseudo-Random Noise. ix, x, 3, 6, 10, 12, 13, 14, 15, 16, 18, 21, 25, 32, 33, 34, 35, 36, 44, 45, 47, 48, 49, 50, 51, 57

RAAS Regional Area Augmentation System. 9

RCSR Reference Chip Shape Response. 55

RF Radio Frequency. 4, 5, 6, 7, 12, 30, 55, 57

RF-DNA Radio Frequency Distinct Native Attributes. 2, 74

RHCP Right-Hand Circularly Polarized. 5

SBAS Satellite-Based Augmentation System. 9

SDR Software Defined Radio. iv, 3, 29, 30

SIMD Single Instruction for Multiple Data. 38

SNR Signal-to-Noise Ratio. 46

SoL Safety of Life. iv, 2, 9, 21, 22, 67, 73

SQM Signal Quality Monitoring. iv, vii, 1, 2, 3, 4, 20, 22, 25, 26, 30, 48, 56, 63, 64, 73

SV Satellite Vehicle. 5, 6, 13, 16, 25, 32, 40, 41, 48

TLM telemetry. 13, 42

TOA Time-of-Arrival. 10

UHD USRP Hardware Driver. 30

USRP Universal Software Radio Peripheral. 30

VC Vision Correlator. 2, 21

WAAS Wide Area Augmentation System. 9, 24

WBP Wide-Band Power. 45, 46

WPAFB Wright-Patterson Air Force Base. 31

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
19–03–2020	Master's Thesis	Sept 2018 — Mar 2020

4. TITLE AND SUBTITLE

Signal Quality Monitoring of GNSS Signals
Using a Chip Shape Deformation Metric

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Nicholas C. Echeverry, 2Lt

5d. PROJECT NUMBER

JON# 19G211

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-20-M-017

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory, AFMC
Attn: Dr. Vasu Chakravarthy
2241 Avionics Circle, Bldg 620
Wright-Patterson AFB OH 45433-7765
Email: vasu.chakravarthy@us.af.mil, Comm: 937-713-4026

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RYWE

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Global Navigation Satellite System continues to become deeply em-bedded within modern civilization, and is depended on for confident, accurate navigation information. High precision position and timing accuracy is typically achieved using differential processing, however these systems provide limited compensation for distortions caused by multipath or faulty satellite hardware. Signal Quality Monitoring (SQM) aims to provide confidence in a receiver's Position, Navigation, and Timing solution and to offer timely warnings in the event that signal conditions degrade to unsafe levels. The methods presented in this document focus on implementing effective SQM using low-cost Commercial Off-the-Shelf equipment, a Software Defined Radio, and a typical software receiver architecture that tracks the Galileo E1C signals and the Global Positioning System L1 Coarse-Acquisition signals. Techniques are centered on acquiring and discriminating signal chip shapes with a goal of identifying both 1) 'clean' and 2) 'deformed' signals. The demonstrated identification method is relevant to the growing significance of SQM for SoL applications while providing benefit for confidently monitoring received GNSS signal integrity without requiring specialized receiver hardware.

15. SUBJECT TERMS

Global Navigation Satellite System; Global Positioning System; Signal Quality Monitoring; Chip Shape

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj. J. Addison Betances, AFIT/ENG
U	U	U	UU	95	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636; joan.betancesjorge@afit.edu