Compass-M1 Broadcast Codes in E2, E5b, and E6 Frequency Bands

Grace Xingxin Gao, Alan Chen, Sherman Lo, David De Lorenzo, Todd Walter, and Per Enge, *Fellow, IEEE*

Abstract—With the launch of the Compass-M1 satellite on 14 April 2007, China is set to become the latest entrant into global navigation satellite systems (GNSS). Understanding the interoperability and integration of the Chinese Compass with the current GNSS, namely the U.S. Global Positioning System (GPS), the European Galileo, and the Russian GLONASS, requires knowing and understanding its signal structures—specifically its pseudorandom noise (PRN) codes and code structures. Moreover, the knowledge of the code is a prerequisite for designing receivers capable of acquiring and tracking the satellite. More important is determining if the signal may degrade performance of the current GNSS in the form of interference. Finally, we are eager to learn from the code and signal design of our Chinese colleagues. For this research, we set up a 1.8-m dish antenna to collect the broadcast Compass-M1 signals. Even with the dish antenna, the received signal is still weak and buried in thermal noise. We then apply signal processing and are able to extract the PRN code chips out of the noise in all three frequency bands. The PRN codes are thousands of bits long. In addition, we find that the Compass-M1 PRN codes in all frequency bands are Gold codes. We also derive the Gold code generators to represents thousands of code chips with fewer than a hundred bits. Finally, we implement these codes in our software receiver to verify and validate our analysis.

Index Terms—Global navigation satellite system (GNSS), pseudorandom noise codes, spread spectrum.

I. INTRODUCTION

THE Beidou or Compass navigation satellite system (CNSS) is China's entry into the realm of global navigation satellite systems (GNSS). The current design plans for 30 medium earth orbit (MEO) satellites and five geostationary orbit (GEO) satellites. The MEO satellites will operate in six orbital planes to provide global navigation coverage [1]. Compass will share many features in common with U.S. GPS and European Galileo systems, providing the potential for low cost integration of these signals into a GPS/Galileo/Compass receiver. These commonalities include multiple frequencies, signal structures, and services.

Statements from Chinese sources indicate that the system will provide at least two services: an open civilian service and a

TABLE I
COMPASS FREQUENCIES AND MODULATION. BPSK(K) DEFINES A BPSK MODULATION WITH A CHIP RATE OF $K \times 1.023$ MHz. FOR EXAMPLE, BPSK(2) DENOTES 2.046-MHz CHIP RATE

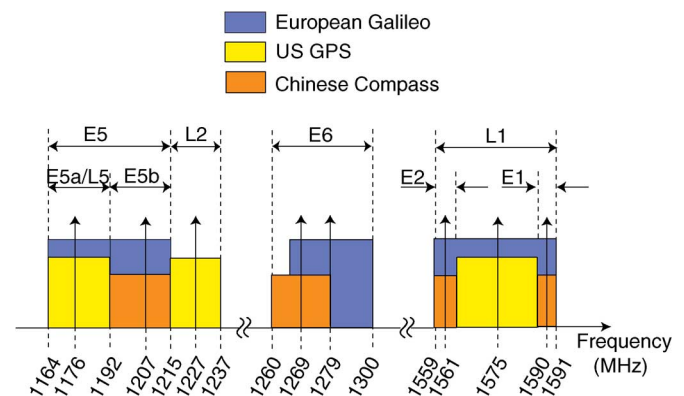| Frequency band	Center frequency (MHz)	Modulation type (inphase, quadrature)
E1	1589.74	BPSK(2), BPSK(2)
E2	1561.10	BPSK(2), BPSK(2)
E6	1268.52	BPSK(10), BPSK(10)
E5b	1207.14	BPSK(2), BPSK(10)

Fig. 1. Frequency occupation of GPS, Galileo, and Compass.

higher precision military/authorized user service [1]. According to International Telecommunication Union (ITU) filings by China, Compass will broadcast in four frequency bands known as E1, E2, E6, and E5b [2]. Table I provides center frequency and modulation information on the signals in each of these bands. Fig. 1, taken from [2], shows the overlap in frequency of the Compass signals with those of GPS and Galileo. Like GPS and Galileo, the Compass navigation signals are code division multiple access (CDMA) signals. They use binary phase shift keying (BPSK) [3]. BPSK(K) defines a BPSK modulation with a chip rate of $K \times 1.023$ MHz. For example, BPSK(2) denotes 2.046-MHz chip rate. Further, our observations and analysis indicate that the codes from the current Compass-M1 are derived from Gold codes.

The Compass-M1 satellite represents the first of the next generation of Chinese navigation satellites and differs significantly from China's previous Beidou navigation satellites. Those earlier satellites were considered experimental, and were developed for two-dimensional positioning using the radio determination satellite service (RDSS) concept pioneered by Geostar [4]. Compass-M1 is also China's first MEO navigation satellite. Previous Beidou satellites were geostationary and only provided coverage over China. The global implications of this satellite and the new GNSS it represents makes the satellite

of great interest to navigation experts. The rapid manner in which researchers have already trained their instruments onto the satellite proves this point. For example, Grelier *et al.* from the Centre National d'Etudes Spatiales (CNES, the French space agency) published an informative overview of their observations of the Compass-M1 signals a month after its launch [2]. The interest has resulted in significant basic information on the Compass-M1 satellite. Observations by CNES, ourselves, and other researchers indicate that the current satellite is only broadcasting on three of the frequencies (E2, E6, E5b). To the best of the authors' knowledge, no observations of Compass E1 broadcasts have yet been made. It also appears that the Compass satellite is not continuously broadcasting navigation signals on the other three frequencies; we have occasionally observed unmodulated or continuous wave (CW) signals in those bands.

The similarity in frequency, signal structure, and services with GPS and Galileo makes Compass a tantalizing prospect for GNSS users. These similarities could allow for the addition of Compass to an integrated GNSS receiver without additional expensive hardware or processing. As such, great motivation exists for understanding Compass and how it may be properly and cost-effectively integrated into a GNSS receiver. Conversely, the signals may pose a source of interference and degrade the performance of Galileo, because Compass overlays the Galileo signals in the E2, E5b, and E6 bands. Hence, understanding the signal design and modulation is important in order to determine the Compass system's properties for interoperability and interference. The first step toward this latter goal is to determine the Compass codes. This will help to develop prototype GPS/Galileo/Compass receivers and help identify ways to best use the new signals together with other planned or existing GNSS signals.

Apart from these basic observations of the Compass-M1 signal structure, little information has been published on the actual codes. We presented our research results of Compass codes and the code structures in [5]. Initial observations including signal spectra and secondary codes, but not PRN codes were reported in [2]. This paper will present our approach to decoding the Compass codes and deriving the code generators in sufficient detail for others to replicate our work. The outline of the paper is as follows. Section II describes our 1.8-m dish antenna and the data collection equipment. Section III describes the process of decoding the E6 code, including carrier wipeoff, code period determination, Doppler offset wipeoff, secondary code determination, solving Doppler ambiguity, zeroing initial phase, and determining the start of the code. With these signal processing steps, the PRN codes are decoded up to overall polarity. Section IV proves that these codes are Gold codes. The code generators are derived and the overall polarity ambiguity of the PRN code is resolved. Section V presents the results for E2 and E5b codes. These codes and code generators are validated by implementing them in our software receiver to track the Compass-M1 satellite in Section VI.

II. DATA COLLECTION METHOD

There are three data collection methods that we can use to determine the codes for Compass-M1. The first method is to use

Fig. 2.   Stanford GNSS Monitor Station (SGMS) antenna.

Fig. 3.   Portable ground station setup.

a high-gain directional antenna such as the 47 m Stanford Research Institute (SRI) dish to measure signals from the satellite. Such an antenna provides enough gain so that the satellite signals will exceed the noise floor and the code chips can be read directly. For example, we used this method at Stanford University to verify the Galileo In-Orbit Validation Element A (GIOVE-A) code [6]. GIOVE-A is the first test satellite of the Galileo system, and was launched in December 2005 [7]. The drawback of using the SRI dish is that usage of the dish is expensive and takes weeks to arrange.

A second method is to use a moderate gain directional antenna and process the data collected by that instrument. The GPS Laboratory at Stanford University has a 1.8-m steerable parabolic reflector antenna with an L band feed [8]. This system, termed the Stanford GNSS Monitor Station (SGMS), provides about 27-dB gain over a standard patch antenna. The antenna and ground station are seen in Figs. 2 and 3. While the SGMS antenna does not provide enough gain to push the code chips above the noise, it allows for determination of the chips without the need for significant processing (in comparison to the third method). The SGMS allowed us to determine the GIOVE-A L1

BOC(1,1) bits with 98% accuracy in the first days of its transmission [6]. The code chips in error were corrected by assuming that they derive from a family of linear codes. Data collected using the SGMS also allowed for the determination of the rest of the GIOVE-A codes [8].

A third method is to collect data using a good omnidirectional antenna. Compared with directional parabolic antennas, an omnidirectional antenna receives not only the Compass-M1 signal, but also other GNSS satellite signals, which act as interference. The smaller antenna gain requires averaging over longer intervals. Obtaining the code chips needs additional signal conditioning, signal processing and *a priori* knowledge and assumptions about the signal structure. The benefit of this method is that it can be done with readily available equipment. Cornell University determined the GIOVE-A L1 BOC(1,1) code in this way [9].

Since we have the SGMS and wanted to determine the Compass-M1 codes quickly, we used the second method. The rest of the paper will discuss how we determined the Compass-M1 E2, E5b and E6 inphase channel codes. We only show the derivation of E6 code, because the decoding process for E2 and E5b codes is similar. Moreover, the E6 code is the most difficult to decode due to its high chip rate of 10.23 MHz. In comparison, the chip rate of E2 and E5b codes is only 2.046 MHz. Data was collected from the SGMS using an Agilent 89600 vector signal analyzer (VSA). Both inphase (I) and quadrature (Q) channel data were collected and the data were nominally mixed to baseband using the center frequencies indicated in Table I.

III. DETERMINING THE E6 CODE

The Compass-M1 signals travel 21 550 km to reach the Earth. The received signals are highly attenuated due to path loss. With an omnidirectional antenna, the received signal power is on the order of 10^{-16} W, assuming a transmit power of 30 W. Even with the 1.8 m dish antenna and high-quality low-noise amplifiers (LNA), the received signal-to-noise ratio (SNR) is still roughly 65 to 70 dB-Hz and the SNR in the signal noise equivalent bandwidth is about -5 to 0 dB. The pseudorandom noise (PRN) code chips are buried in noise and not directly visible in the time domain. Low SNR is the main challenge in extracting the Compass-M1 PRN code sequences.

In order to decode the PRN code sequences, we need to process the data to boost the signal above the noise floor. The main idea is to accumulate the signal to suppress the noise. Our approach is to stack multiple periods of the PRN sequence. We decoded Galileo GIOVE-A broadcast codes by using this method [6], [8], [9]. Discovering secondary code or navigation bit information is mandatory for this method. At least 100 ms of the Compass data have to be recorded. In this paper, we explain the decoding of the Compass-M1 codes by this approach [5].

A. Compass Signal Modeling

Before decoding the Compass code, we first develop a model for the signal. As the Compass system is a code division spread spectrum system and is similar to the GPS system, we model the Compass signals based on our understanding of the GPS signals [3], [10]. The received signal contains a
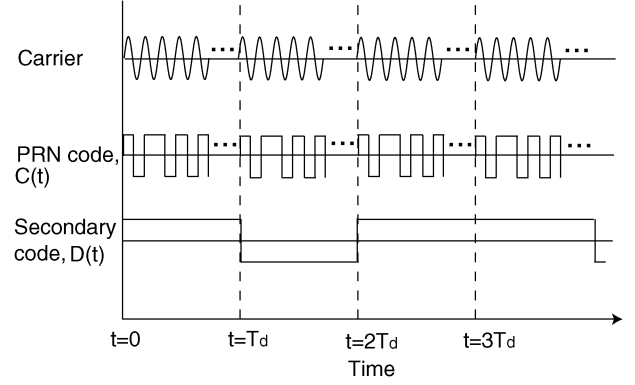

Fig. 4. Example of relationship between the PRN code signal, the secondary code, and the carrier. The secondary code duration is the same as the PRN code period.

spread PRN code, which is a binary ± 1 vector denoted as $\mathbf{c} = (c_0, c_1, c_2, \ldots, c_{N_c-1})$.

Each chip in the PRN code $\mathbf{c}$ has a rectangular shape:

$$W_c(t) = \begin{cases} 1, & 0 < t \le T_c \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where T_c is the duration of a PRN chip.

The PRN signal $C(t)$ is therefore the periodic binary phase shift keyed version of $\mathbf{c}$. $C(t)$ is periodic with the period, $T_d = N_c T_c$. In other words

$$C(t) = C(t - lT_d) \tag{2}$$

for all integers l. One period of $C(t)$ is

$$C(t) = \sum_{i=0}^{N_c-1} c_i W_c(t - iT_c) \tag{3}$$

where N_c is the number of chips in the PRN code sequence.

The PRN code is modulated with either navigation data or a secondary code or both. Since the goal of this paper is decoding the PRN code, rather than studying the secondary code or navigation data, we simplify our terminology in the following way. We denote the secondary code or navigation data or their product as the binary ± 1 vector $\mathbf{d} = (d_0, d_1, d_2, \ldots)$, and refer to it as the secondary code. We assume that the secondary code bit duration is the same as the PRN period T_d, and this also handles the case that bit duration is an integer multiple of T_d. Then we express the navigation signal waveform as

$$D(t) = \sum_{i=0}^{\infty} d_i W_d(t - iT_d) \tag{4}$$

where $W_d(t)$ is a rectangular window function with width T_d,

$$W_d(t) = \begin{cases} 1, & 0 < t \le T_d \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The modulated signal, $D(t)C(t)$, is carried with a nominal carrier frequency f_c. Since the satellite is moving, the carrier is affected by a Doppler frequency offset f_D and an initial phase θ. The relationship between the PRN code signal, the secondary code and the carrier is shown in Fig. 4.
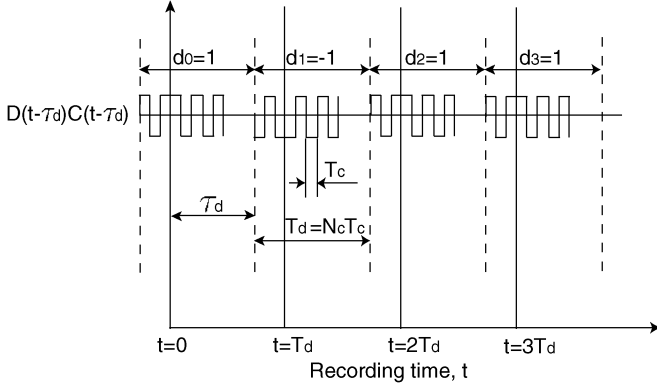
Fig. 5. Relationship of T_c, T_d and τ_d with the recorded data. T_c is the PRN code chip duration. N_c is the number of bits per PRN code period. The PRN code period is the same as the secondary code bit duration, $T_d = N_c T_C$. The code phase offset τ_d arises from the lack of synchronization between the recorded data and the start of the PRN code. In other words, τ_d represents the start of the first complete period of the PRN code.

The received signal is written as

$$\sqrt{P} D(t - \tau_d) C(t - \tau_d) \exp j \left(2\pi(f_c + f_D)t + \theta \right) + n(t) \quad (6)$$

where τ_d is the code phase offset arising from the lack of synchronization between the recorded data and the start of the PRN code. The signal power is P, and $n(t)$ captures noise and interference. For decoding, the SNR matters, not the absolute signal power. Hence, we normalize by the signal power, and the noise becomes $n_0(t) = n(t)/\sqrt{P}$.

So, the received signal $s(t)$ is written as

$$D(t - \tau_d) C(t - \tau_d) \exp j \left(2\pi(f_c + f_D)t + \theta \right) + n_0(t). \quad (7)$$

In fact, there are two signals: inphase and quadrature. The inphase signal contains a short PRN code for civilian use, while the quadrature signal has a long PRN code for military or authorized use. Since we are only interested in the short civilian code, the signal model only shows the inphase code. We take the quadrature signal as interference and include it in $n_0(t)$.

We now define the code phase τ_d in (6) and (7) formally. First we set $t = 0$ to represent the start of the recording. Then τ_d is the start of the first complete period of the PRN code. Clearly, τ_d is less than the secondary code bit duration; that is $0 \leq \tau_d < T_d$.

The relationship of T_c, T_d and τ_d with the recorded data is illustrated in Fig. 5. For simplicity, the carrier, Doppler offset, and noise are not shown in the figure.

In order to reveal the PRN code from the received signal in (7), we need to strip off the secondary code $D(t)$, wipe off Doppler offset f_D, zero the initial phase shift θ and adjust the code phase offset τ_d. The decoding diagram is shown in Fig. 6. The processing flow follows the numbered arrows. The rest of this section explains the process of Fig. 6 in detail for the Compass E6 code. A similar methodology is employed in the E2 and E5b bands. We use the E6 signal as an example, because it is more challenging to decode in the 20 MHz bandwidth E6 band than the 4-MHz bandwidth E2 or E5b bands.
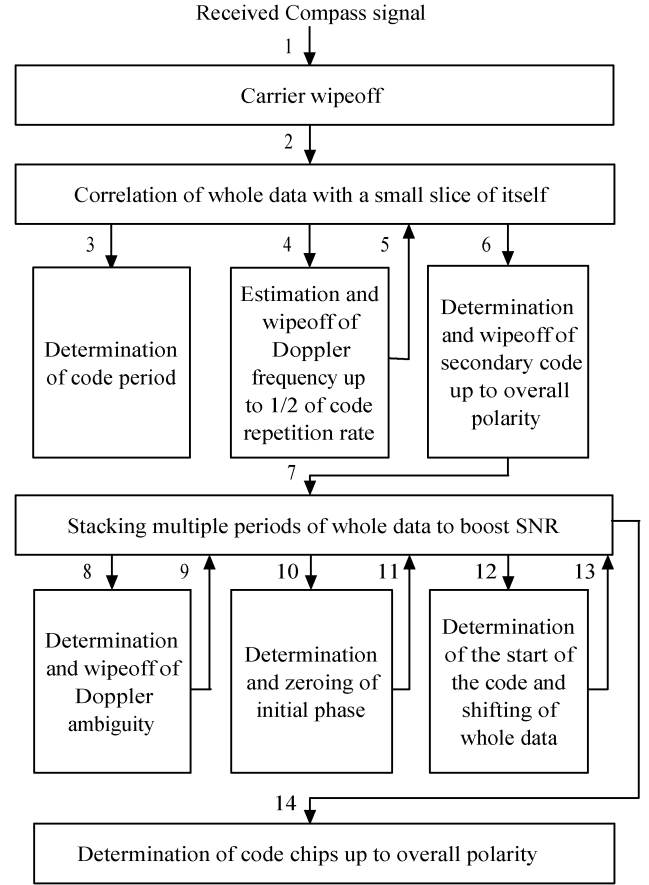

Fig. 6. Decoding block diagram.

B. Carrier Wipeoff

The first step is carrier wipeoff. In our setup, the VSA downconverts the incoming signal to the baseband. The baseband signal is written as

$$s_b(t) = D(t - \tau_d) C(t - \tau_d) \exp j(2\pi f_D t + \theta) + n_b(t) \quad (8)$$

where $n_b(t)$ is the baseband noise.

Any error between the VSA frequency and the true carrier is treated as part of the Doppler offset and will be wiped off at a later stage.

C. Correlation of Whole Data With a Small Slice of Itself

Although the Compass signal is buried in noise, some of its characteristics can be obtained by correlation, namely code period, Doppler offset, and navigation bits. Correlation is a signal processing technique commonly used in communication and radar systems [11]. We first correlate the whole data with a small slice of itself.

The small slice of the baseband data is denoted as

$$\tilde{s}_b(t) = \begin{cases} s_b(t), & 0 < t \leq t_0 \\ 0, & t > t_0 \end{cases} \quad (9)$$

where t_0 is the duration of the slice of data. It is important that this slice of data does not contain the boundary of a code period. That is, we need to guess $t_0 < T_d$. On the other hand, the slice

should be large enough to make the correlation peaks discoverable. Assuming -5 to 0 dB as a rough estimate of the received SNR and a margin of 15 dB, a correlation gain of at least 20 dB is required, which translates to at least 100 chips in the small slice, or $t_0 > 0.01$ ms. If the PRN code period is known, a good value of t_0 would be half of the code period, because one out of two consecutive slices of data will not contain a period boundary. If the code period is unknown, it may be necessary to try a few guesses of t_0 and adjust the time origin $t = 0$ to satisfy these constraints.

The correlation function is

$$\int_{-\infty}^{\infty} s_b(t+\tau)\tilde{s}_b^*(\tau)d\tau = \int_{-\infty}^{\infty} s_b(\tau)\tilde{s}_b^*(\tau-t)d\tau$$

$$= \int_{t}^{t+t_0} s_b(\tau)s_b^*(\tau-t)d\tau \quad [12]. \quad (10)$$

Assuming no noise $(n_b(t) \simeq 0)$, we substitute (8) into (10) to express the correlation as

$$\int_{t}^{t+t_0} [D(\tau-\tau_d)C(\tau-\tau_d)\exp j(2\pi f_D\tau+\theta)]$$

$$\cdot [D(\tau-t-\tau_d)C(\tau-t-\tau_d)$$

$$\cdot \exp -j(2\pi f_D(\tau-t)+\theta)]\,d\tau$$

$$= \exp j2\pi f_D t \int_{t}^{t+t_0} [D(\tau-\tau_d)C(\tau-\tau_d)$$

$$\cdot D(\tau-t-\tau_d)C(\tau-t-\tau_d)]\,d\tau. \quad (11)$$

The beauty of this correlation result is twofold. First, initial phase θ inside the integral cancels out, due to the multiplication of the signal and the shifted version of its conjugate. Thus, it is not necessary to consider carrier phase when processing the correlation results up to step 6 as shown in the decoding block diagram in Fig. 6. Although θ does not show in the correlation results, it still exists in the received signal, and needs to be wiped off in step 10. Second, the Doppler offset is moved out of the integral. As a result, the Doppler offset becomes a visible modulation on the correlation peaks.

The expression of the correlation result has two parts, a sinusoidal envelope and an integral. The integral

$$\int_{t}^{t+t_0} [D(\tau-\tau_d)C(\tau-\tau_d)D(\tau-t-\tau_d)C(\tau-t-\tau_d)]\,d\tau \quad (12)$$

is the correlation between $s_b'(t) = D(t-\tau_d)C(t-\tau_d)$ and its small slice, $\tilde{s}_b'(t)$,

$$\tilde{s}_b'(t) = \begin{cases} s_b'(t), & 0 < t \leq t_0 \\ 0, & t > t_0. \end{cases} \quad (13)$$

With respect to (8), $s_b'(t)$ is the noiseless version of $s_b(t)$ with no Doppler offset or initial phase. The evaluation of (12) is illustrated in Fig. 7. Positive correlation peaks occur whenever
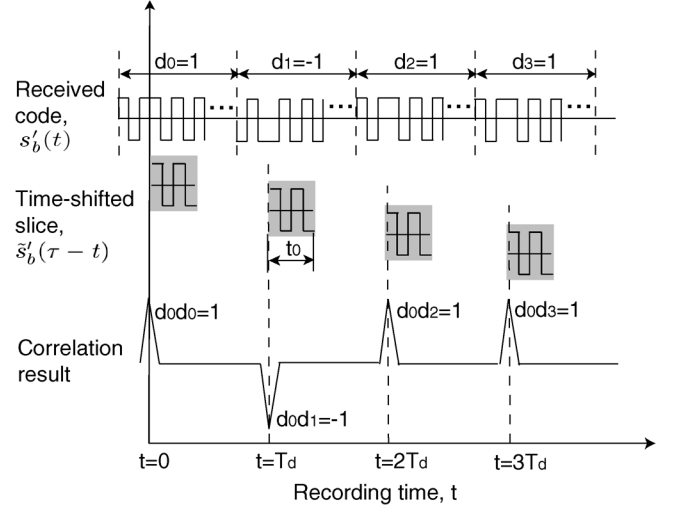

Fig. 7. Correlation of the received code with time-shifted replicas. Correlation peaks occur whenever the time-shifted slice aligns with similar versions of itself or flipped versions. The ith correlation peak polarity is determined by the secondary code bit of the small slice d_0 and the secondary code bit of the corresponding segment d_i. The triangular correlation peak width is twice the PRN code chip duration.

the time-shifted slice $\tilde{s}_b'(\tau - t)$ aligns with similar versions itself in $s_b'(t)$, and negative ones whenever it aligns with flipped versions of itself. These alignments happen at multiples of the PRN code period. In other words, peaks of height $d_0 d_i$ occur at $t = iT_d$, $i = 0, 1, 2, \ldots$.

Therefore, the correlation peak heights in (11) are

$$d_0 d_i \exp j2\pi f_D t \quad (14)$$

at $t = iT_d$, $i = 0, 1, 2, \ldots$.

The inphase and quadrature channels of the correlation results are shown in Figs. 8 and 9. In agreement with (14), the correlation is a string of equally-spaced peaks, modulated by a sinusoidal envelope. The correlation results are normalized by the amplitude at $t = 0$. The first correlation peak is higher than the rest, because the received signal contains both the PRN code and noise. The first peak matches in both code and noise, while others match in code only.

For this experiment, the sampling rate is 46.08 MHz. We oversample the received signal to mitigate the effect of sampling bit transitions. We choose the data length to be 100 ms. The data length needs to be long enough to provide enough processing gain to eventually boost the signal above the noise, but cannot be too long, because of satellite and VSA clock drift. We choose the duration of the small slice t_0 to be 0.5 ms. Again, the small slice needs to be long enough to maximize the processing gain, but should not be too long so as to reach the code period boundaries. In practice, achieving a good enough choice of these parameters requires some iteration, followed by validation after the code period is determined.

D. Determination of Code Period

Since we aim to stack multiple periods of the signal to boost SNR, we now determine the code period. According to Fig. 7, the correlation peaks occur at multiples of the code period. Measuring the inter-peak interval in Figs. 8 and 9 yields the code
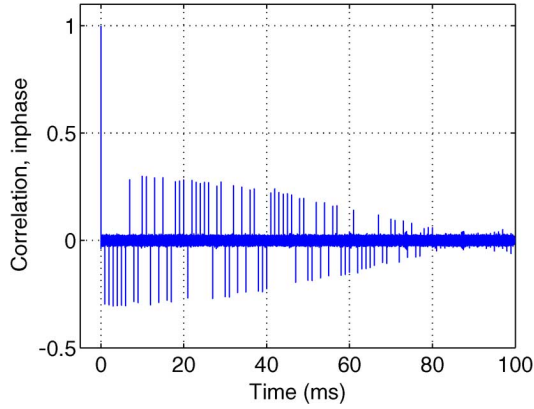
Fig. 8.   Correlation of whole data with a small slice of itself, as a result of (11), inphase channel. The correlation results are normalized by the amplitude at $t = 0$. The first correlation peak is higher than the rest, because the received signal contains both the PRN code and noise. The first peak is a result of both code and noise matched, while others are code only. Measuring the inter-peak interval yields the code period.
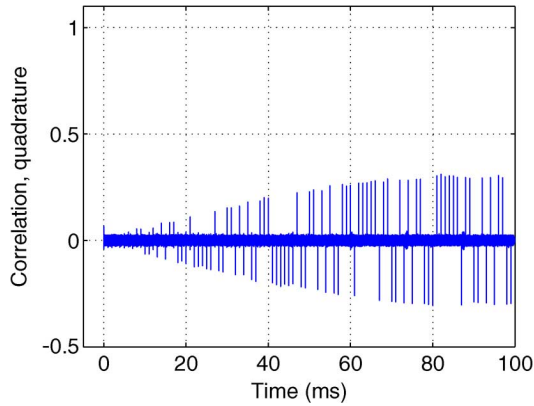

Fig. 9.   Correlation of whole data with a small slice of itself, as a result of (11), quadrature channel.

period to be $T_d = 1$ ms, by averaging the number of samples between the correlation peaks and dividing by the sampling rate.

The code period of 1 ms validates choosing the whole data length to be 100 ms and the small slice to be 0.5 ms. This is because 100 ms of data is 100 periods of the code and provides 20-dB processing gain. Among two consecutive 0.5-ms slices, at least one of them is guaranteed to have no code period boundary, because the small slice duration is half the code period.

E. Estimation and Wipeoff of Doppler Frequency up to Half of the Code Repetition Rate

In Figs. 8 and 9, the heights of the peaks vary due to the Doppler envelope, which is $\exp j2\pi f_D t$ in (14). For MEO satellites, the Doppler, in a range of $[-5000 \text{ Hz}, 5000 \text{ Hz}]$, causes fast variation of correlation peaks and received signal amplitudes in both inphase and quadrature channels, and thus must be removed. The next step is to estimate the Doppler offset f_D and wipe it off from the received baseband signal. One way of compensating Doppler offset is by accurately calculating satellite speed from the almanac. This approach requires a stable and precise oscillator in the receiver. We use pure signal processing instead.

Note that the Doppler estimation is based on the correlation peaks $T_d = 1$ ms apart. This is equivalent to estimating a signal with $1/T_d = 1$ kHz sampling rate. Furthermore, we are unsure of the true polarity of the Doppler envelope; i.e., the polarity of the $d_0 d_i$ coefficient in (14). Changes of f_D by even multiples of $1/2T_d$ or changes of f_D by odd multiples of $1/2T_d$ coupled with sign changes of $d_0 d_i$ produce identical output peaks. Therefore, the envelope of the peaks aliases at a multiple of $1/2T_d$. In other words, the estimated Doppler $\hat{f}_D$ has an ambiguity equal to a multiple of half of the sampling rate. In this case, $T_d = 1$ ms, so the Doppler ambiguity is a multiple of $1/2T_d = 500$ Hz. The true Doppler offset equals the Doppler estimated based on correlation peaks plus ambiguity adjustment; i.e., $f_D = \hat{f}_D + k(1/2T_d)$, for some integer k. We will use signal processing methods to resolve the $k(1/2T_d)$ Doppler ambiguity later once multiple periods of data are stacked.

We search for $\hat{f}_D$ in the frequency range $[-250 \text{ Hz}, 250 \text{ Hz}]$ with a step of 0.01 Hz. We choose the value that minimizes the standard deviation of correlation peak heights after Doppler compensation. The received baseband signal then becomes

$$s_b(t) \exp(-j2\pi \hat{f}_D t) = s_b(t) \exp\left(-j2\pi\left(f_D - k\frac{1}{2T_d}\right)t\right). \tag{15}$$

Substituting (8) into (15), the noiseless part then becomes

$$D(t - \tau_d)C(t - \tau_d)\exp j\left(\pi k\frac{1}{T_d}t + \theta\right). \tag{16}$$

Figs. 10 and 11 show the correlation results in inphase and quadrature channels after compensating for the estimated Doppler frequency offset. As expected, the peaks in the inphase channel now have uniform heights, although varying polarities. There are no peaks in the quadrature channel, because as demonstrated in (11), the correlation operation zeros the initial phase of the signal. The initial phase still exists, but does not appear in the correlation result.

F. Determination and Wipeoff of Secondary Code up to Overall Polarity

After the estimated Doppler offset is wiped off, the correlation peak polarities from (14) become

$$d_0 d_i \tag{17}$$

at $t = iT_d$, $i = 0, 1, 2, \ldots$.

This reflects the polarities of the positive and negative correlation peaks in Fig. 10. If we assume $d_0 = 1$, the peak signs reveal the secondary code **d** to be $[1, -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, 1, -1, 1, -1, \ldots]$. Assuming $d_0 = -1$ flips the sign of all the data, including the primary PRN code **c**. The sign flipping problem will be solved after the code generator polynomial is derived in Section IV.

We can read off the secondary code up to overall polarity ambiguity, and write it in time domain as

$$d_0 D(t) = d_0 \sum_{i=0}^{\infty} d_i W_d(t - iT_d). \tag{18}$$

We generate the secondary code waveform with a sampling rate of 46.08 MHz and length of 100 ms, matching the received
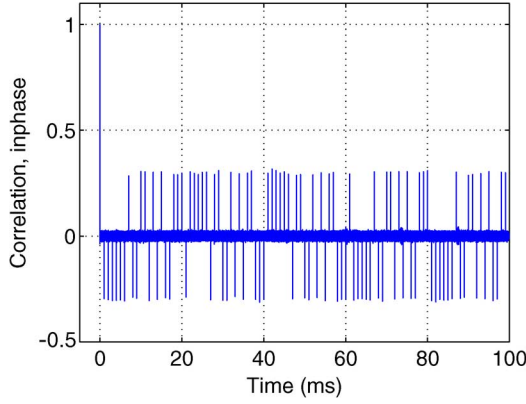
Fig. 10. Correlation after wiping off estimated Doppler offset, inphase channel. As expected, the peaks in the inphase channel now have uniform heights, although varying polarities. The correlation peak polarities follow (17).
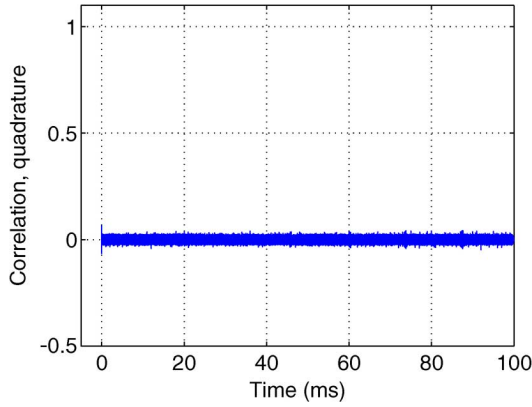

Fig. 11. Correlation after wiping off estimated Doppler offset, quadrature channel. There are no peaks in the quadrature channel, because as demonstrated in (11), the correlation operation zeros the initial phase of the signal. The initial phase still exists, but does not appear in the correlation result.

signal. We then wipe off the secondary code in the Doppler compensated baseband signal by point-wise multiplication with the secondary code waveform. Since the start of the PRN code period τ_d is yet unknown, the local secondary code replica $d_0 D(t)$ is not aligned with the secondary code of the received signal, as shown in Fig. 12. For simplicity, Fig. 12 does not show the Doppler residual. Now in each period of the received signal, the segment before the boundary of a PRN code period has the same overall polarity, while the segment after does not, as shown in Fig. 12. Since $d_i^2 = 1$, the result after secondary code demodulation is

$$
\begin{cases}
d_0 C(t - \tau_d) \\
\quad \times \exp j\left(\pi k \frac{1}{T_d} t + \theta\right), & \text{for } iT_d \le t < iT_d + \tau_d \\
d_0 d_i d_{i+1} C(t - \tau_d) \\
\quad \times \exp j\left(\pi k \frac{1}{T_d} t + \theta\right), & \text{for } iT_d + \tau_d \le t < (i+1)T_d
\end{cases}
$$
(19)

where $i = 0, 1, 2, \ldots$. In each period, the first τ_d duration of data no longer contains secondary code except for the overall polarity d_0. So for these segments of data, the secondary code is wiped off.
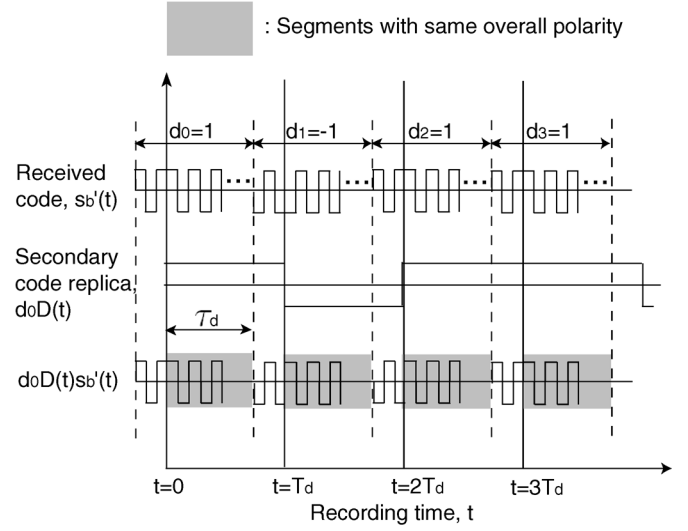

Fig. 12. Secondary code wipeoff. Because the recording time is not synchronized with the start of a complete PRN code period, only the secondary code of the signal segments in the shaded area is wiped off. The transition point is the start of the PRN code period τ_d which will be determined later.
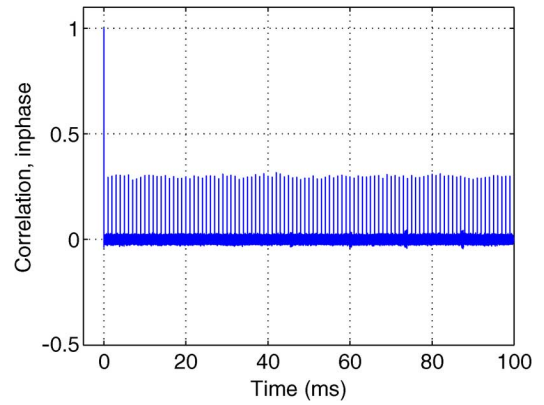

Fig. 13. Correlation after wiping off estimated Doppler offset and secondary code, inphase channel. The all-positive correlation peaks verify that the shaded segments in Fig. 12 have the same overall polarity, and the secondary code is wiped off. Multiple periods of the signal are ready to be stacked.

We do not show the experimental result associated with (19) because the processed signal is still buried in noise and is invisible. However, the correlation peaks are above the noise, as shown in Fig. 13. The peaks are generated by correlating the secondary code demodulated data of 100 ms with a 0.5-ms slice of itself. The all-positive correlation peaks verify that the shaded segments in Fig. 12 have the same overall polarity, and the secondary code is demodulated.

G. Stacking Multiple Periods of the Whole Data

So far, the code period has been determined, and part of each code period has the same overall polarity. We are ready to stack multiple periods of the data together to boost the signal above the noise floor. With respect to our model in (19), d_0 is a constant and $C(t - \tau_d)$ repeats for each code period T_d. If k (associated with Doppler ambiguity) is an even number, then $\exp j(\pi k(1/T_d)t + \theta)$ also repeats every time duration T_d. This means that the first τ_d duration of each period of the signal is the same. This motivates us to stack M periods of the signal
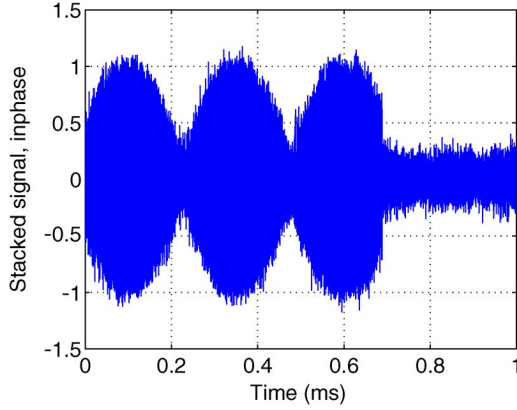
Fig. 14. Multiple periods of signal stacked, inphase channel. Different from the previous figures of correlation results, the time range is now 1 ms. The signal stack shows two parts. The first part corresponds to the shaded segments in Fig. 12. The signal energy is boosted $M = 100$ times because of the coherent stacking. The sinusoidal envelope in the first part is consistent with (19) and reflects the Doppler ambiguity, which will be discussed in the next subsection. The second part looks like noise due to the noncoherent stacking.
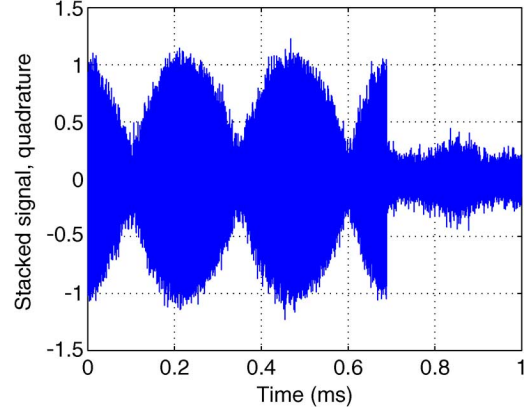

Fig. 15. Multiple periods of signal stacked, quadrature channel. The signal exist in both inphase and quadrature channels because of the initial carrier phase, which will be zeroed in a later subsection.

together. If we assume additive white noise, then the SNR is increased M times. The signal after stacking becomes

$$Md_0C(t - \tau_d)\exp j\left(\frac{\pi k}{T_d}t + \theta\right), \quad \text{for } 0 \le t < \tau_d \quad (20)$$

and

$$d_0C(t-\tau_d)\exp j\left(\frac{\pi k}{T_d}t + \theta\right)\sum_{i=0}^{M-1}d_id_{i+1}, \quad \text{for } \tau_d \le t < T_d. \quad (21)$$

If k is an odd number, the signals in different periods cancel out, and the stacking result appears to be all noise. If this happens, we modulate each period of data with $(-1)^i$, $i = 0, 1, 2, \ldots$, and restack.

The above analysis is based on continuous signals. Our data are discrete with sampling frequency of 46.08 MHz. Since the PRN code period is not an integer multiple of the sample period, the number of samples per PRN code period will vary by 1 sample. We pad an additional sample of value 0 at the end of the period if necessary to ensure each period has the same number of samples. Then we stack the multiple periods by point-wise addition. The zero padding does not significantly distort the final PRN code waveform due to oversampling For the case of the Compass E6 signal, there are 4.5 samples per PRN code chip.

Figs. 14 and 15 show the inphase and quadrature channels of the $T_d = 1$ ms stack of the baseband signal with Doppler compensated and secondary code wiped off. $M = 100$ code periods are stacked in the two figures, and the signal amplitude is normalized. Different from the previous figures of correlation results, the time range is now 1 ms.

The signal stack has two parts as shown in Figs. 14 and 15. The first part appears to have a sinusoidal envelope, while the second part looks like noise. The first part corresponds to the sum of the shaded segments in Fig. 12, described by (20), and shows the PRN code energy boosted M times. The code

$C(t - \tau_d)$ is modulated by the initial secondary code bit d_0 and a sinusoidal wave $\exp j((\pi k/T_d)t + \theta)$. The second part of the stacked signal is described by the sum of the terms in (21). If we assume the secondary code to consist of independent identically distributed (i.i.d.) random variables with zero mean, then $\sum_{i=0}^{M-1} d_id_{i+1}$ also has zero mean. This second part of the signal stack is thus dominated by noise.

H. Determination and Wipeoff of Doppler Ambiguity

We solve the Doppler ambiguity problem by estimating the frequency of the sinusoidal envelope in the first part of the signal stack. As argued in Section III-E, the Doppler ambiguity is a multiple of $1/2T_d = 500$ Hz. In this case, there are two sinusoidal cycles in 1 ms. So the Doppler ambiguity is 2000 Hz; i.e., $k = 4$.

After the Doppler ambiguity is resolved, the Doppler is wiped off completely by multiplying the demodulated signal with a 2000-Hz sinusoid of length 100-ms sampled at 46.08 MHz. The noiseless part of the signal becomes

$$D(t - \tau_d)C(t - \tau_d)\exp j\theta. \quad (22)$$

We then restack the signal. The inphase and quadrature parts of the 1-ms stack of the data are shown in Figs. 16 and 17. Now the signal envelope becomes constant.

I. Determination and Zeroing of Initial Phase

Figs. 16 and 17 show signals in both inphase and quadrature channels, even though there is only one PRN code of interest. The reason is the initial phase shift θ, which needs to be zeroed. Recall that θ was canceled out of the correlation in (11) but not yet removed from the signal itself. We generate the I-Q plot by plotting inphase values versus quadrature values in a 2-D domain. Each complex time-domain sample becomes a dot in the I-Q plot. Fig. 18 shows the I-Q plot of the 1-ms stack before phase adjustment. The angle between the axis of the two clusters of dots and the horizontal axis reveals the initial phase. We expect to see two clusters of dots separated horizontally if the initial phase is zero. Hence, we adjust the initial phase shift θ
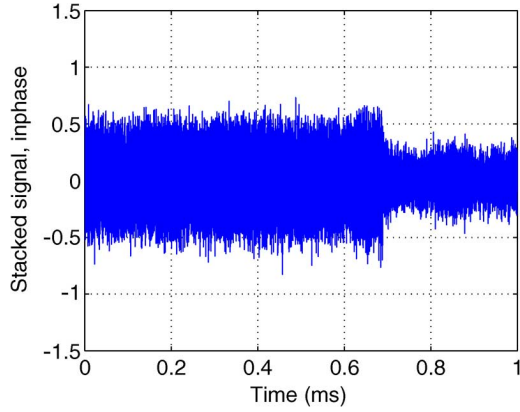
Fig. 16. Stacked signal with no Doppler ambiguity, inphase channel. Since the Doppler is completely removed, the signal envelope becomes constant.
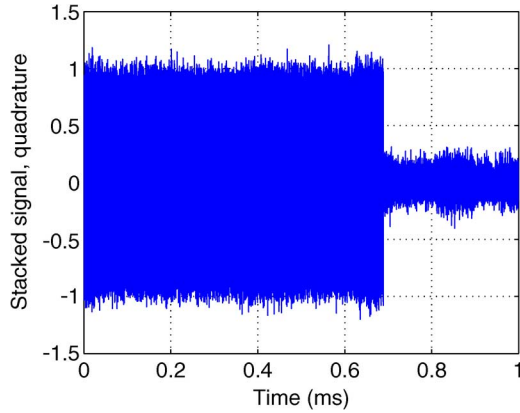

Fig. 17. Stacked signal with no Doppler ambiguity, quadrature channel. Again the signal energy is split between the inphase and quadrature channel due to the initial carrier phase, which will be discussed in the next subsection.
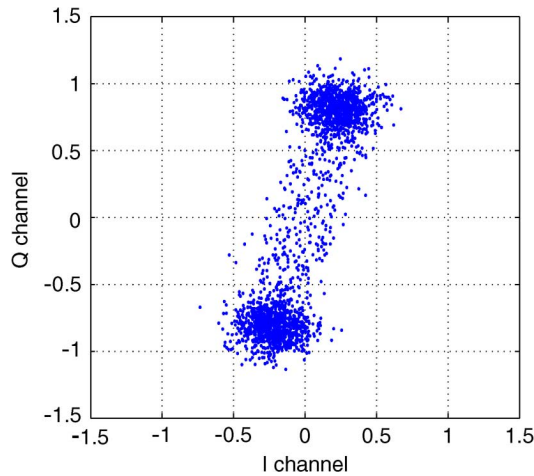

Fig. 18. The I-Q plot of the stacked signal before phase adjustment. We generate the I-Q plot by plotting inphase values versus quadrature values in a 2-D domain. Each complex time-domain sample becomes a dot in the I-Q plot. The angle between the axis of the two clusters of dots and the horizontal axis reveals the initial phase.

so that the axis of the time-domain scatter points is aligned with the inphase axis, as shown in Fig. 19.
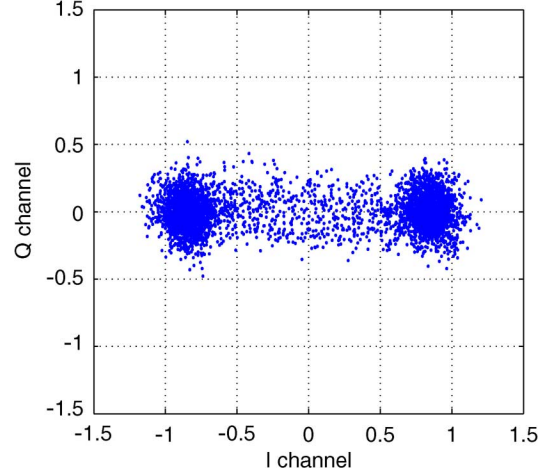

Fig. 19. The I-Q plot of the stacked signal after phase adjustment. As expected, the axis of the time-domain scatter points is aligned with the inphase axis.
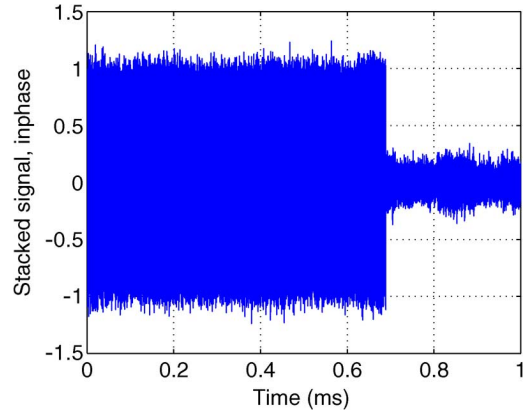

Fig. 20. Stacked signal with initial phase adjusted, inphase channel. After the initial phase is set to zero, the signal energy concentrates in the inphase channel; while the quadrature channel has no signal but noise and interference. Again, the signal stack has two parts, because the start of the data recording is not aligned with the that of the PRN code period, and only the first part is coherently stacked. The transition point reveals the boundary of a PRN code period.

Figs. 20 and 21 show the inphase and quadrature part of the 1-ms stack of baseband signal with complete Doppler wipeoff, secondary code wipeoff and initial phase adjustment. After the initial phase is set to zero, the signal energy concentrates in the inphase channel; while the quadrature channel has no signal but noise and interference. Again, the signal stack has two parts, because the start of the data recording is not aligned with that of the PRN code period, as indicated in (19). Only the first part is coherently stacked. The transition point reveals the boundary of a PRN code period.

J. Determination of the Start of the Code and Shifting the Whole Data Set

The next step is to determine the start of the code τ_d. According to (20) and (21), τ_d is the transition point in the stacked signal from boosted PRN code to noise. If we zoom in, the transition region of the stacked signal is shown in Fig. 22. We roughly estimate τ_d by reading the number of samples from the start to the transition point. We then shift the whole 100-ms data
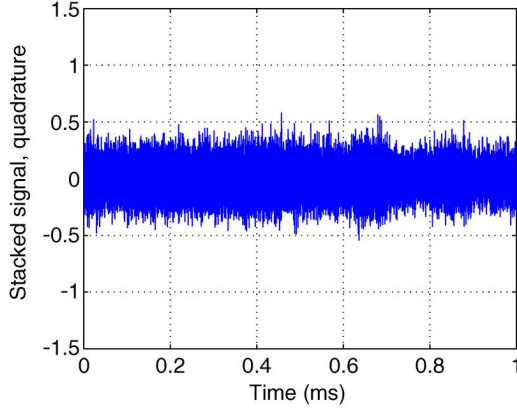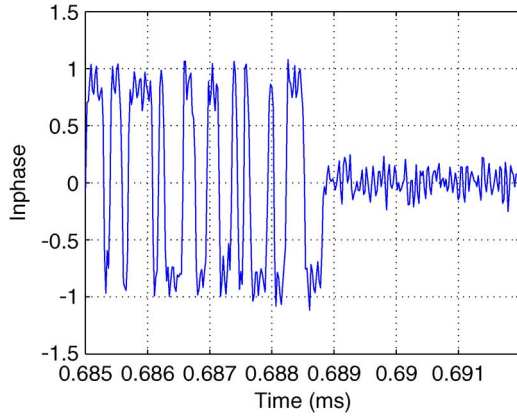
Fig. 21.   Stacked signal with initial phase adjusted, quadrature channel.

Fig. 23.   Stacked signal with code start determined, zoom in. The PRN code waveform is revealed.

Fig. 22.   Stacked signal with initial phase adjusted, zoom in. The transition point reveals a PRN code period boundary, or the start of the PRN period τ_d.
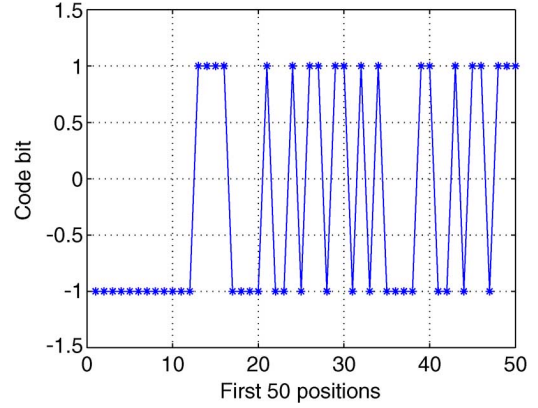
Fig. 24.   First 50 bits of E6 I-channel PRN code.

to the rough estimate of τ_d, restack, and determine τ_d to accuracy of 1 data sample. This accuracy is sufficient to read off code due to oversampling.

We then shift the whole data to the start of the PRN code. The noiseless part of the signal now becomes

$$d_0 C(t). \tag{23}$$

K. Determination of Code Chips up to Overall Polarity

With the start of the PRN code sequence determined, we restack the whole data into one code period. The individual code chips are now visible over the noise. Fig. 23 shows the first 5 μs. The whole code sequence is thus revealed.

After downsampling, the code bits are obtained. The E6 I-channel PRN code is 10 230 bits long and lasts for one 1 ms. Fig. 24 shows the first 50 bits of the code. There is a low probability of error associated with this step, but the code can be verified by repeating the entire process with another segment of data or by finding a consistent generator for the code.

Note that the overall polarity of the PRN code is ambiguous, because the sign of the first bit of the secondary code d_0 is not determined yet. This may cause the sign of the whole PRN code sequence to flip. In the next section, we will derive the simplest possible code generator, which will take into account the sign ambiguity issue.
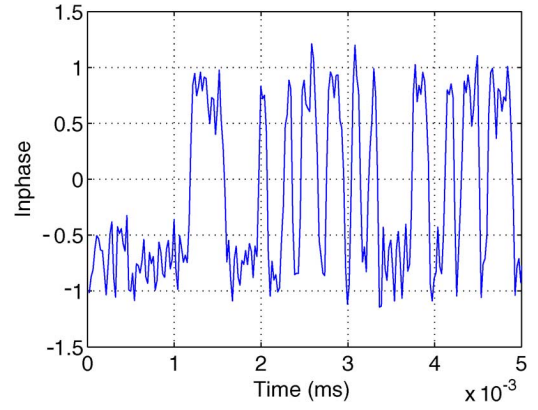
IV. DERIVING CODE GENERATORS

With the code sequence obtained, we can implement this PRN sequence in a software receiver for acquisition and tracking. However, we would also like to study the code structure, which will help us understand the effects of this code on other signals in the frequency band. This motivates us to seek the underlying code generator.

Furthermore, determining the PRN code generator helps minimize the code representation. This dramatically reduces the memory requirement of the receivers. Storing thousands of bits in receivers is expensive in terms of flash memory and even more expensive in digital signal processing (DSP) units.

We consider linear codes as likely candidates for the Compass code design, because they have good correlation performance, and can be generated by linear feedback shift registers (LFSRs), which require only tens of bits to specify [13].

The schematic of an LFSR is shown in Fig. 25 [14]–[16]. Its outputs are linear combinations of the previous bits. In other words, the entire output sequence u_i is completely determined by its tap weights $(a_1, \ldots, a_N)$ and initial state $(u_1, \ldots, u_N)$. The LFSR arithmetic is modulo 2.

The algorithm in Fig. 26 searches for a linear code representation. It loops through values of N, the length of the LFSR, until tap weights consistent with the demodulated code sequence are
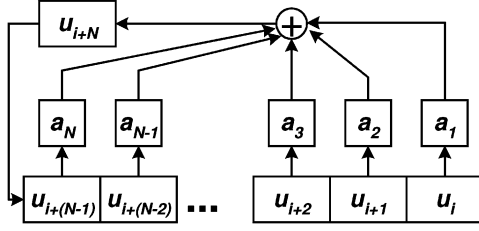
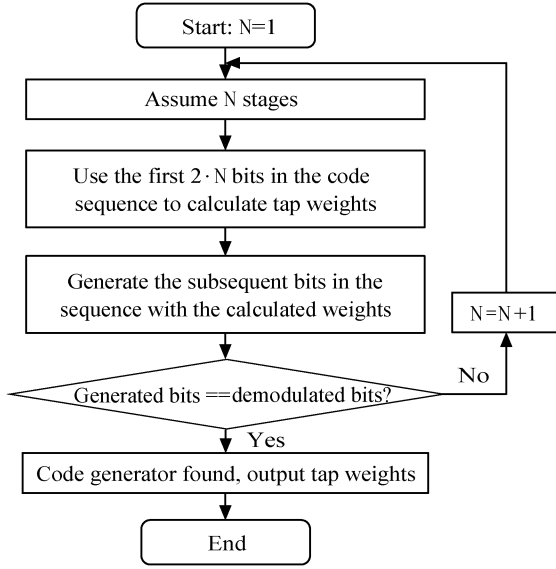Fig. 25. Linear feedback shift register (LFSR).

Fig. 26. Search algorithm for linear code representation.

found. For example if $N = 10$, we can form the following ten equations with ten unknown tap weights, $a_1, \ldots, a_{10}$:

$$\begin{cases} u_{11} = a_{10} \cdot u_{10} \oplus a_9 \cdot u_9 \oplus \cdots \oplus a_1 \cdot u_1 \\ u_{12} = a_{10} \cdot u_{11} \oplus a_9 \cdot u_{10} \oplus \cdots \oplus a_1 \cdot u_2 \\ \vdots \\ u_{20} = a_{10} \cdot u_{19} \oplus a_9 \cdot u_{18} \oplus \cdots \oplus a_1 \cdot u_{10} \end{cases} \quad (24)$$

The above set of equations is solved modulo 2 to obtain the tap weights. The subsequent bits in the sequence are then generated. If the generated bits and the demodulated bits match, the code generator is then found. If not, we increment N and repeat. To accommodate possible chip estimation errors, we define a successful match to be a 95% agreement between the generated bits and the demodulated bits.

Our method of finding the shortest LFSR for a given output sequence is closely related to the Berlekamp–Massey algorithm in [17], [18]. The Berlekamp–Massey algorithm is more efficient because it speeds up the solution of the set of equations by reusing the results of previous solutions. For deriving short LFSRs (tens of bits), the time savings are not significant. Our method only requires us to find a string of correctly demodulated bits of length equal to twice the length of the LFSR. Discrepancies in the rest of the sequence are allowed. Our algorithm is more robust to errors. More importantly, our algorithm is able to handle a concatenated code, which consists of multiple segments with different generators. If the generated bits match the

TABLE II
Code Generator Polynomials and Initial States for Generating the First 8190 Bits of the Compass E6 I-Channel Code

E6_head I channel code	
Polynomial_1	$X^{13} + X^{12} + X^{10} + X^9 + X^7 + X^6 + X^5 + X + 1$
Initial State_1	[1 1 1 1 1 1 1 1 1 1 1 1 0]
Polynomial_2	$X^{13} + X^4 + X^3 + X + 1$
Initial State_2	[1 1 1 1 1 1 1 1 1 1 1 1 1]

TABLE III
Code Generator Polynomials and Initial States for Generating Bits 8191–10230 (Last 2040 Bits) of the Compass E6 I-Channel Code

E6_tail I channel code	
Polynomial_1	$X^{13} + X^{12} + X^{10} + X^9 + X^7 + X^6 + X^5 + X + 1$
Initial State_1	[1 1 1 1 1 1 1 1 1 1 1 1 1]
Polynomial_2	$X^{13} + X^4 + X^3 + X + 1$
Initial State_2	[1 1 1 1 1 1 1 1 1 1 1 1 1]

decoded ones but only up to a point, then a transition point has been revealed. We use the remaining bits to find another generator using the algorithm in Fig. 26 until we reach either another transition point, or the end. The E6 code turns out to be a concatenated code.

With our algorithm, the Compass-M1 E6 code is proven to be linear. Moreover, the E6 code is composed of segments from two codes, denoted as E6_head and E6_tail. E6_head provides the first 8190 bits of the code sequence. E6_tail contains the 8191st bit to the 10 230th bit in the sequence. Both E6_head and E6_tail are generated by 26th-order LFSRs. The 26th-order polynomials can be further factorized into two 13th-order polynomials. This means that both the head and tail parts of the E6 code sequence can be generated by modulo 2 summing the outputs of pairs of 13 stage LFSRs. The initial states of the 13-stage LFSRs are calculated by brute-force search. In fact, the head and tail parts share the same two 13th-order polynomials, which form the preferred pair of a Gold code [11], [19]. The 13-order Gold codes for the head and tail parts share identical code generator polynomials. The only difference between the codes is the last bit of the initial state of one of the code generators.

The code generators and initial conditions for the E6_head and E6_tail sequence are presented in Tables II and III, respectively.

Note that there is a sign ambiguity in the PRN code, as mentioned in the previous section. The overall polarity of the demodulated PRN code sequence can be flipped. Furthermore, the code generator operates over $\{0,1\}$. When mapping to the digital signal output, it is possible to map from $\{0,1\}$ into either $\{-1, 1\}$ or $\{1, -1\}$. Considering the PRN code sign ambiguity, we derive the simplest code generators for E6, E2 and E5b codes as follows. If the PRN code polynomial can be factorized into $1 + X$ multiplied by another factor, then we simplify by removing the $1 + X$ factor from the polynomial. This is because the polynomial $1 + X$ generates a sequence of all ones. If it is added modulo 2 to the code sequence, all the resultant code bits flip sign.

After obtaining the primary PRN codes and their generators, we demodulate the secondary code as well, which turns

TABLE IV
Code Generator Polynomials and Initial States for
Generating the Compass E2 I-Channel Code

E2 I channel code	
Polynomial_1	$X^{11} + X^{10} + X^9 + X^8 + X^7 + X + 1$
Initial State_1	[0 1 0 1 0 1 0 1 0 1 0]
Polynomial_2	$X^{11} + X^9 + X^8 + X^5 + X^4 + X^3$ $+ X^2 + X + 1$
Initial State_2	[0 0 0 0 0 0 1 1 1 1 1]

out to be a 20-bit Neuman-Hofman code with the following sequence:$[-1-1-1-1-11-1-111-11-11-1-1111-1]$, as defined in [20]. The 20-bit Neuman–Hofman code of Compass-M1 E6 band is the same as that used in GPS L5 signals.

The code sequence sign ambiguity of the previous section can be solved after deriving the PRN code polynomials. If the PRN code polynomial can be factorized into $1 + X$ multiplied by another factor, then the derived code should be flipped to remove the $1 + X$ factor from the polynomial. Otherwise, the derived code is correct. This is because the polynomial $1 + X$ generates a sequence of all ones. If it is added modulo 2 to the code sequence, all the resultant code bits flip sign.

Now we have solved the sign ambiguity for the secondary code sequence. The secondary code turns out to be a 20-bit Neuman–Hofman code with the following sequence:$[-1-1-1-1-11-1-111-11-11-1-1111-1]$, as defined in [20]. The 20-bit Neuman–Hofman code of Compass-M1 E6 band is the same as that used in GPS L5 signals.

V. Decoding the E2 and E5b Codes

We decoded the E2 and E5b codes using the technique described above for decoding the E6 code. The E2 signal uses BPSK(2) modulation. The Compass E2 I-channel primary code is 1 ms long and has 2046 bits. The E2 Q-channel has a long military/authorized service code that is not studied in this paper. Our analysis has proven that the E2 short code is linear and can be generated by a 22nd-order LSFR. The 22nd-order LSFR polynomial can be further factorized into two 11th-order polynomials. This indicates that the Compass E2 I-channel PRN code is an 11–stage Gold code.

The code generator polynomials and initial states are shown in Table IV. The PRN code generator schematic is shown in Fig. 27.

Two signals occupy the E5b band: one modulated with BPSK(2) in the I-channel and one with BPSK(10) in the Q-channel. The BPSK(2) code is a 1-ms short code. The BPSK(10) code is a long military/authorized service code, also not studied in this paper.

The Compass E5b short code turns out to be the same as the E2 code. Many observers have noted that the E2 I-channel and E5b BPSK(2) codes are identical [2], [5]. This is also verified through the acquisition and tracking of the E5b I-channel signal using the E2-derived I-channel code.

Furthermore, the E2 and E5b I-channel also have a 20-bit secondary code sequences identical to the one used in E6 band.
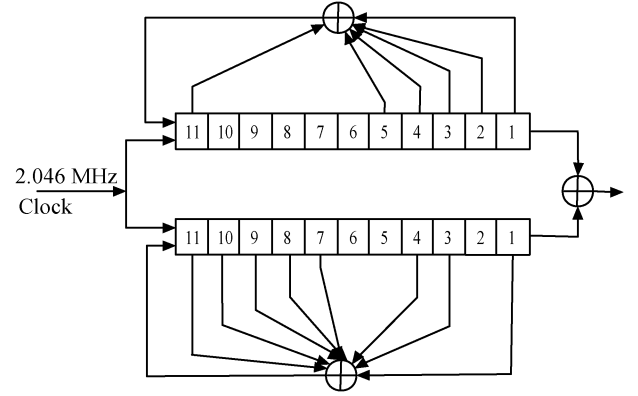

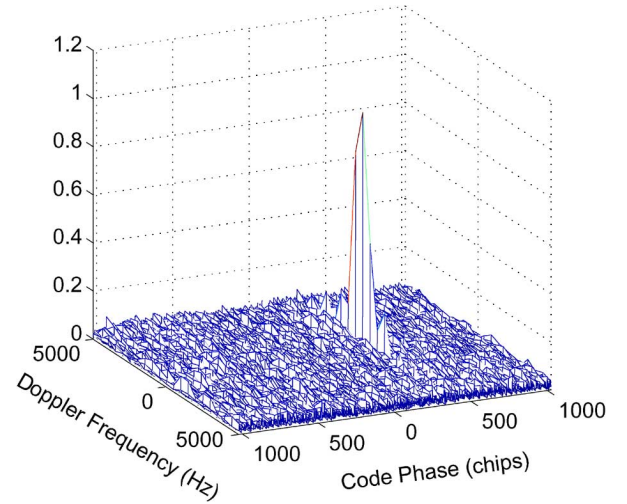Fig. 27. Code generator schematic of the Compass E2 I-channel signal

Fig. 28. Acquisition plot of Compass-M1 E2 I-channel.

VI. Verification of Compass Codes

We verify the decoded codes by acquiring and tracking the signals from the Compass-M1 satellite with a software GNSS receiver. We have developed a multi-signal all-in-view GNSS software receiver implemented in MATLAB (in collaboration with University of Colorado, Boulder, and Aalborg University) [21]. We implemented the Compass-M1 codes in this jointly developed receiver. Raw Compass data collected by the SGMS is loaded into our software receiver to test the efficacy of the derived codes.

Acquisition is implemented as a parallel code-phase search using FFT-based processing. Several milliseconds of data may be combined to increase weak-signal sensitivity or to provide more accurate estimates of carrier Doppler frequency, although at a tradeoff in execution time. The 3-D acquisition plot in Fig. 28 shows the normalized correlation function output of the Compass-M1 E2 signal as a function of code phase on one axis and carrier Doppler frequency on the other axis. A small amount of averaging (2 ms) is used. We read the code phase and Doppler estimate based on the location of the main peak in the code phase and Doppler domain. For brevity, we only show the acquisition and tracking results for the E2 channel.
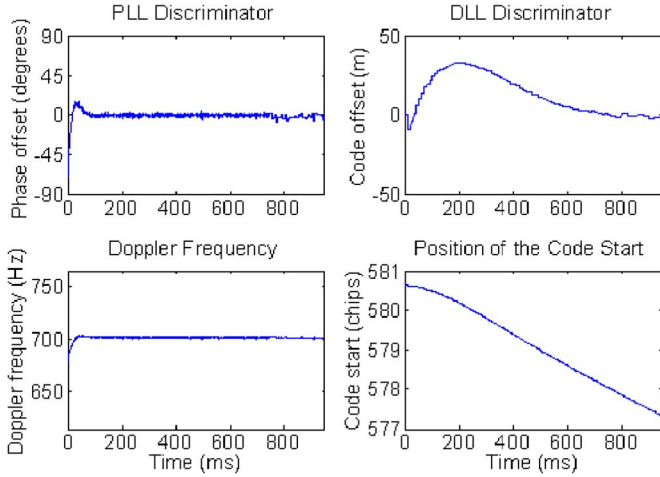
Fig. 29.   Tracking results of Compass-M1 E2 I-channel.

TABLE V
SUMMARY OF COMPASS-M1 BROADCAST CODE RESULTS (I-CHANNEL ONLY)

Frequency band	Modulation type	Primary code period	Code generators	Secondary code period
E2	BPSK(2)	1 ms	11 stage Gold code	20 ms
E5b	BPSK(2)	1 ms	11 stage Gold code	20 ms
E6	BPSK(10)	1 ms	Two 13 stage Gold codes concatenated	20 ms

NV, a Belgium-based company, were able to track Compass-M1 E2 and E5b signals in real-time through a hemispherical antenna [22].

VII.   CONCLUSION

This paper described in detail the decoding of the Compass-M1 broadcast I-channel codes in all frequency bands, namely E2, E5b, and E6 bands. We not only extracted the code bits, but also derived the code generators. All three PRN primary codes are linear codes with a period of 1 ms. The E2 and E5b codes are identical, a truncated 11th-order Gold code of 2046 bits. The E6 code has 10 230 bits, a concatenation of two Gold code segments. Both segments are truncated 13th-order Gold codes with the same code polynomials but different initial states. The E2, E5b, and E6 primary codes are modulated with 20-bit Neuman–Hofman codes as secondary codes. The secondary codes in the three frequency bands are identical.

Table V provides a summary of the codes that we have currently determined.

We implemented the derived codes in our software receiver and successfully acquired and tracked the real broadcast Compass-M1 signals. This validates our decoding results. We are excited at the prospect of decoding and tracking many more new satellite signals.

Immediately after acquisition, the code-phase and carrier frequency estimates are used to initialize the code and carrier numerically-controlled oscillators (NCOs). The receiver refines the estimates of carrier frequency, carrier phase, and code phase through a succession of tracking modes, where the PLL and DLL noise bandwidths are successively reduced.

The tracking output in Fig. 29 shows four subplots as follows, each as a function of elapsed tracking time along the horizontal axis:

- upper-left: PLL discriminator output in degrees;
- upper-right: DLL discriminator output in meters ($150 \, \text{m} = 1$ chip);
- Lower-left: carrier Doppler frequency estimate;
- Lower-right: code-phase estimate with respect to the receiver's on-board millisecond counter.

Since one of the tracking objectives is the estimation of the secondary code length and sequence, integration times are kept to 1 ms for all tracking modes (the length of the primary spreading code sequence). This is because carrier polarity may change at each millisecond, and this sequence is unknown until the secondary decoding has occurred.

All tracking outputs converge, such as phase offset, code offset, and Doppler frequency. The PLL converges quickly. However, the DLL discriminators take a bit longer to settle to roughly zero offset. This is caused by the acquisition algorithm estimating the code phase to the nearest sample, while there are only 2.5 samples per chip due to the choice of sampling rate. The result is that our estimate may be off by as much as a quarter of a chip. This is confirmed by the plot as our estimate is never greater than 0.25 chip (about 40 m) during convergence. The Doppler frequency is locked at 700 Hz as shown in the lower-left plot in Fig. 29.

For brevity, we only show the acquisition and tracking results for E2 channel.

Our Compass PRN codes have also been implemented in hardware and independently verified by receiver companies. For example, W. Dewilde *et al.* from Septentrio Satellite Navigation

REFERENCES

[1] "China to Build Global Satellite Navigation System." [Online]. Available: http://www.english.people.com.cn/200704/16/eng20070416_366845.html

[2] T. Grelier, J. Dantepal, A. Delatour, A. Ghion, and L. Ries, "Initial observations and analysis of Compass MEO satellite signal," *Inside GNSS*, pp. 39–43, May/Jun. 2007.

[3] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*.   Lincoln, MA: Ganga-Jamuna, 2006.

[4] "Compass Satellite Navigation Experimental System (BeiDou-1)." [Online]. Available: http://www.sinodefence.com/strategic/space-craft/beidou1.asp

[5] G. X. Gao, A. Chen, S. Lo, D. De Lorenzo, and P. Enge, "GNSS over China, the Compass MEO satellite codes," *Inside GNSS*, pp. 36–43, Jul./Aug. 2007.

[6] G. X. Gao, J. Spilker, T. Walter, P. Enge, and A. R. Pratt, "Code generation scheme and property analysis of broadcast Galileo L1 and E6 signals," in *Proc. ION GNSS Conf.*, Fort Worth, TX, Sep. 2006.

[7] "Successful launch of the Galileo Satellite: GIOVE-A, the first European navigation satellite in space," [Online]. Available: http://www.galileoju.com/doc/8541%20Press%20Release%20Successful%20Launch%20of%20GIOVE%20A.pdf

[8] G. X. Gao, D. S. De Lorenzo, A. Chen, S. C. Lo, D. M. Akos, T. Walter, and P. Enge, "Galileo broadcast E5 codes and their application to acquisition and tracking," in *Proc. ION NTM Conf.*, San Diego, CA, Jan. 2007.

[9] M. L. Psiaki, T. E. Humphreys, S. Mohiuddin, S. P. Powell, A. P. Cerruti, and P. M. Kintner, Jr., "Searching for Galileo," in *Proc. ION GNSS Conf.*, Fort Worth, TX, Sep. 2006.

[10] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and Applications*, 2nd ed. Norwood, MA: Artech House, 2006.

[11] S. W. Golomb and G. Gong, *Signal Design for Good Correlation*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[12] B. Porat, *Digital Processing of Random Signals: Theory and Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[13] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory and Applications*. Reston, VA: Amer. Inst. Aeronaut. Astronaut. (AIAA), 1996.

[14] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Commun.*. Reading, MA: Addison-Wesley, 1995.

[15] S. Glisic and B. Vucetic, *Spread Spectrum CDMA Systems for Wireless Communications*. Norwood, MA: Artech House, 1997.

[16] R. M. Rao and S. Dianat, *Basics of Code Division Multiple Access (CDMA)*. Bellingham, WA: SPIE Press, 2005.

[17] E. R. Berlekamp, *Algorithmic Coding Theory*. New York: McGraw-Hill, 1968.

[18] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, Jan. 1969.

[19] M. A. Abu-Rgheff, *Introduction to CDMA Wireless Communications*. New York: Academic, 2007.

[20] J. Spilker, *Digital Communications by Satellite*, ser. Information Theory. Englewood Cliffs: Prentice-Hall, 1977.

[21] D. S. De Lorenzo, "Navigation Accuracy and Interference Rejection for GPS Adaptive Antenna Arrays," Ph.D. dissertation, Stanford Univ., Stanford, CA, 2007.

[22] W. D. Wilde, F. Boon, J.-M. Sleewaegen, and F. Wilms, "More compass points: Tracking China's MEO satellite on a hardware receiver," *Inside GNSS*, pp. 44–48, Jul./Aug. 2007.

Grace Xingxin Gao received the B.S. degree in mechanical engineering and the M.S. degree in electrical engineering, both from Tsinghua University, Beijing, China, in 2001 and 2003, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2008.

She is a Research Associate in the GPS Laboratory, Stanford University. Her current research interests include GNSS signal and code structures, GNSS receiver architectures, and interference mitigation.

Alan Chen received the S.B. degree in aeronautics and astronautics from the Massachusetts Institute of Technology, Cambridge, and the M.S. degree from the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA. He currently pursuing the Ph.D. degree in the Department of Aeronautics and Astronautics, Stanford University.

His current research interest involves unexploded ordnance, sensor fusions, autonomous helicopters, and GNSS signals.

Sherman Lo received the Ph.D. degree in aeronautics and astronautics from Stanford University, Stanford, CA.

He is a Research Associate at the Stanford University GPS Research Laboratory managing the assessment of Loran for civil aviation and also works on a variety of GNSS-related topics.

Dr. Lo received the Institute of Navigation (ION) Early Achievement Award and the International Loran Association (ILA) President's Award.

David De Lorenzo received the Ph.D. degree in aeronautics and astronautics from Stanford University, Stanford, CA, with thesis research on adaptive antenna arrays, their ability to reject radio-frequency interference, and their impact on GPS measurement errors.

He is a Research Associate at the Stanford University GPS Research Laboratory. He has previously worked for Lockheed Martin and for the Intel Corporation.

Todd Walter received the Ph.D. degree from Stanford University, Stanford, CA, in 1993.

He is a Senior Research Engineer in the Department of Aeronautics and Astronautics, Stanford University. He is currently developing Wide Area Augmentation System (WAAS) integrity algorithms and analyzing the availability of the WAAS signal.

Dr. Walter is a fellow of the ION.

Per Enge (S'78–M'92–SM'02–F'04) received the Ph.D. degree from the University of Illinois, Urbana-Champaign.

He is a Professor of aeronautics and astronautics at Stanford University, Stanford, CA, where he is the Kleiner–Perkins, Mayfield, Sequoia Capital Professor in the School of Engineering. He directs the GPS Research Laboratory, which develops satellite navigation systems based on the Global Positioning System (GPS). He has been involved in the development of Federal Aviation Administration's GPS Wide Area Augmentation System (WAAS) and Local Area Augmentation System (LAAS) for the FAA.

Prof. Enge received the Kepler, Thurlow, and Burka Awards from the Institute of Navigation. He is a Member of the National Academy of Engineering and a Fellow of the Institution of Navigation.