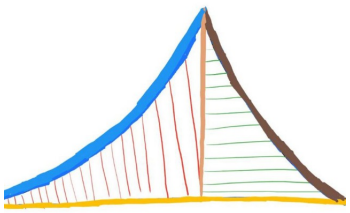
NAVIC STANDARD SIMULATION

Through Python

G. V. V. Sharma



Contents

List of Figures

List of Tables

Chapter 1

Introduction

NavIC (an acronym for 'Navigation with Indian Constellation') is the operational name for Indian Regional Navigation Satellite System (IRNSS), developed independently and indigenously by Indian Space Research Organization (ISRO). The objective of this autonomous regional satellite navigation system is to provide accurate real-time positioning and timing services to users in India and a region extending upto $1,500$ km (930 mi) around it.

NavIC is designed with a constellation of 7 satellites and a network of ground stations operating 24 x 7. Three satellites of the constellation are placed in geostationary orbit and four satellites are placed in inclined geosynchronous orbit. The ground network consists of control centre, precise timing facility, range and integrity monitoring stations, two-way ranging stations, etc.

NavIC provides two levels of service, the "standard positioning service", which is open for civilian use, and a "restricted service" (an encrypted one) for authorised users (including the military). NavIC has a theoritical positional accuracy of 5m - 20m for general users and 0.5m for military purposes.

This book describes the NavIC standards simulation using Python code. Chapter 2 provides NavIC system overview, various frequency bands used and various services provided by NavIC. Chapter 3 describes the implementation details of Transmitter module. Chapter 4 details about various channeling parameters used in simulation. Chapter 5 elaborately describes the implementation details of Receiver module. Chapter 6 details out key results from the simulation.

1.1. Scope of simulation

The scope of the simulation is limited to

1. sending baseband signal (without a carrier) thorugh transmitter module, mixing it with channel modelling module and verifying that the same baseband signal is received at the output of the receiver module.
2. baseband signals for L5 and S bands (L1 band is out of scope)
3. only SPS services signal (RS signal is out of scope)

Chapter 2

NavIC System Overview

2.1. The Frequency Bands

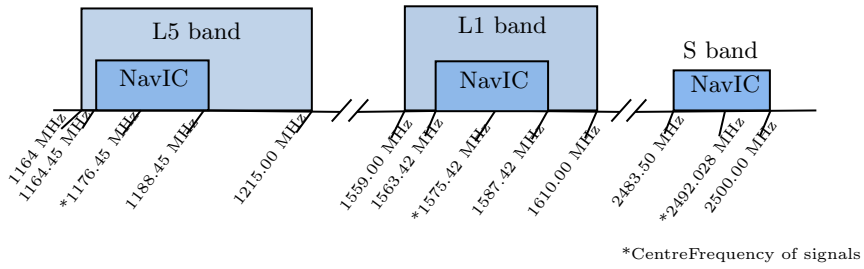


Figure 2.1: Frequency bands of NavIC Signals

Bands	Carrier Frequency	Bandwidth	Usage
L1	1575.42 Mhz	24 Mhz	for low power devices
L5	1176.45 Mhz	24 Mhz	navigation and positioning
S	2492.028 Mhz	16.5 Mhz	SBAS and messaging

Table 2.2: NavIC frequency bands

Satellite communication utilizes multiple frequency bands to accommodate different types of communication services and addresses various technical considerations. Here are some reasons why multiple frequency bands are used in satellite communication:

Spectrum Allocation: The electromagnetic spectrum is divided into various frequency bands to allocate different services and applications. This division ensures that different systems can operate without interfering with each other. By utilizing multiple frequency bands, satellite communication can effectively coexist with other wireless services and minimize interference issues.

Signal Propagation Characteristics: Different frequency bands exhibit unique propagation characteristics. Lower frequency bands, such as L band, have better signal penetration through obstacles and are less affected by atmospheric conditions, making them suitable for applications where signal reliability is crucial. Higher frequency bands, such as Ku band or Ka band, offer larger bandwidths and higher data transmission rates, making them ideal for applications requiring high-speed data transfer.

Bandwidth and Capacity: Different frequency bands offer varying bandwidths, and by utilizing multiple bands, satellite communication systems can increase overall capacity. This allows for the simultaneous transmission of multiple signals, accommodating a wide range of services such as television broadcasting, voice communication, internet access, and data transfer.

Frequency Reuse and Interference Mitigation: Satellite systems employ frequency reuse techniques to maximize the utilization of the available frequency spectrum. By using different frequency bands, satellite operators

can reuse frequencies in different geographical areas without causing interference. This allows for efficient utilization of the limited spectrum resources.

Regulation and International Coordination: The allocation and usage of frequency bands are regulated by international bodies and national spectrum management organizations. These regulations help ensure efficient spectrum utilization, prevent interference between different systems, and promote global coordination and compatibility of satellite communication services.

In summary, the use of multiple frequency bands in satellite communication enables efficient spectrum utilization, accommodates different services, and addresses various technical considerations such as signal propagation, bandwidth, capacity, and interference mitigation. By leveraging the advantages offered by different frequency ranges, satellite systems can provide reliable, high-speed communication services to a wide range of applications and users.

The seven satellites in the NavIC constellation so far use two frequencies for providing positioning data — the L5 and S bands. This was because India hadn't received the International Telecommunication Union authorisation for using the L1 and L2 frequency bands, which are widely used worldwide for navigation services. The new satellites NVS-01 onwards, meant to replace these satellites, will also have L1 band. L1 is an interoperable frequency and can be used across all chipsets(of mobile devices), provided they use our signal architecture.

2.1.1. L-band

The L band offers several advantages for wireless communication systems, including a balance between signal propagation characteristics and antenna size. It provides good signal penetration through various atmospheric conditions, vegetation, and even some obstacles. These properties make it suitable for applications such as satellite communication, navigation systems, and mobile networks.

Satellite communication is one of the significant applications of the L band. Satellites in geostationary orbits often utilize this frequency range for broadcasting television signals, as well as for maritime, navigation and aeronautical communications. The L band allows for reliable and efficient transmission over long distances, making it a valuable resource for global connectivity. Because of satellites' increased use, number and size, congestion has become a serious issue in the lower frequency bands.

L1 and L5 are specific frequencies within the L band that are used in Global Navigation Satellite Systems (GNSS), such as GPS (Global Positioning System), NavIC and Galileo. These frequencies play a crucial role in providing accurate positioning, navigation, and timing information.

2.1.1.1. L1

L1 refers to the first frequency within the L band used by GNSS. In NavIC, the L1 frequency is centered around 1575.42 MHz. The L1 signal carries the primary navigation message and is used for standard positioning and timing applications. It is widely used in various sectors, including transportation,

surveying, and consumer applications like personal navigation devices and smartphones.

2.1.1.2. L5

L5, on the other hand, is an additional frequency introduced in modernized GNSS systems like GPS and Galileo. In NavIC, the L5 frequency is centered around 1176.45 MHz. It was introduced to provide improved accuracy, integrity, and resistance to interference. The L5 signal carries more precise and reliable positioning information, making it particularly useful in critical applications that require high levels of accuracy, such as aviation, surveying, and scientific research. The L1 frequency offers broad coverage and compatibility with legacy systems, while the L5 frequency provides more precise positioning and improved resistance to interference. The combination of these frequencies allows for more reliable and accurate navigation solutions, benefiting a wide range of industries and applications.

2.1.2. S-band

The S band is another frequency range within the electromagnetic spectrum, located between the L band and the C band. It spans a frequency range of approximately 2 to 4 GHz. In NavIC, S band frequency is centred around 2492.028 MHz. The S band finds applications in various fields, including communication, radar systems, satellite broadcasting, and scientific research.

One of the primary uses of the S band is in satellite communication. Satel-

lites in geostationary orbits often utilize S band frequencies for uplink and downlink communication with ground stations. The S band provides a good balance between antenna size and data capacity, making it suitable for broadcasting television signals, voice communication, and data transmission. However, the higher frequency bands typically give access to wider bandwidths, but are also more susceptible to signal degradation due to 'rain fade' (the absorption of radio signals by atmospheric rain, snow or ice).

2.2. NavIC Architecture

The NavIC architecture is as shown in Fig 2.2. It mainly consists of

1. Space segment
2. Ground segment
3. User segment

2.2.1. Space segment

Space segment consists of a constellation of 7 satellites. Three satellites of the constellation are placed in geostationary orbit, at 32.5°E, 83°E and 129.5°E respectively, and four satellites are placed in inclined geosynchronous orbit with equatorial crossing of 55°E and 111.75°E respectively, with inclination of 29° (two satellites in each plane).

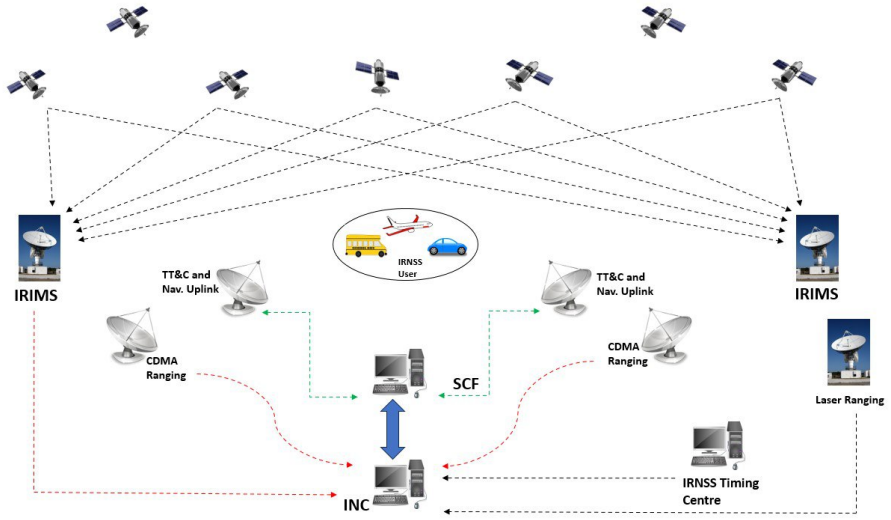


Figure 2.2: NavIC Architecture

2.2.2. Ground segment

Ground segment takes care of operation and maintenance of the constellation. It consists of

1. ISRO Navigation Centre
2. IRNSS Spacecraft Control Facility
3. IRNSS Range and Integrity Monitoring Stations
4. IRNSS Network Timing Centre
5. IRNSS CDMA Ranging Stations
6. Laser Ranging Stations
7. Data Communication Network

2.2.3. User segment

User segment consists of

1. A single frequency receiver having capability to receive SPS signal at either L1, L5 or S band frequency

2. A multi-frequency receiver having capability to receive SPS signal at combination of L1, L5 and S band frequencies

3. A multi-constellation receiver compatible with NavIC and other GNSS signals.

The Figure2.3 above specifies the radio frequency interface between space and user segments.

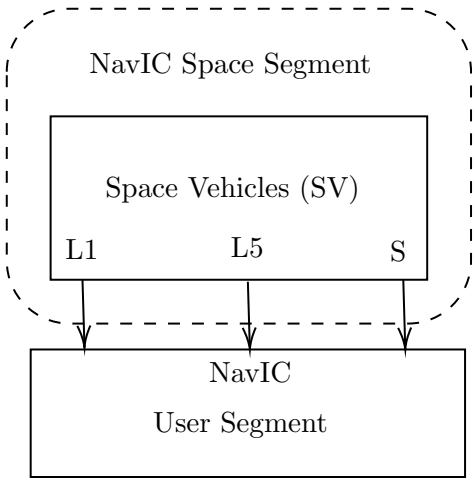


Figure 2.3: the NavIC bands segment blocks

2.3. NavIC Services

The NavIC provides basically two types of services:

1. Standard Positioning Service (SPS)
2. Restricted Service (RS)

Both SPS and RS signals contain ranging codes that allow receivers to compute their travelling time from satellite to receiver, along with navigation data, in order to know the satellite's position at any time.

2.3.1. Standard Positioning Service (SPS)

It is available to all civilian users free of charge and provides positioning, navigation, and timing information with a moderate level of accuracy. The SPS signals in NavIC primarily operate in the L5 and S frequency bands.

2.3.2. Restricted Service (RS)

The RS is intended for authorized users and offers enhanced accuracy, integrity, and availability compared to the SPS signals. The RS signals in NavIC operate in both the L5 and S bands and broadcast through a phased array antenna to keep required coverage and signal strength.

Chapter 3

Navigation Data

Navigation data in satellite communication refers to the crucial information transmitted between satellites and ground-based receivers to facilitate accurate positioning and navigation. It includes data related to satellite orbits, precise timing, and other parameters necessary for determining the satellite's position relative to the Earth's surface.

3.1. Frame structure

The NavIC L1 Master Frame is of 1800 symbols long made of 3 subframes. Subframe 1 consists of 52 symbols, Subframe 2 is composed of 1200 symbols, and Subframe3 is comprised of 548 symbols,The master frame structure is shown in figure 3.1

The Time of Interval (TOI)is transmitted in Subframe 1 is shown in figure 3.2. Subframe 2 as shown in figure 3.3, transmitting the primary navigation parameters, and Subframe 3 is shown in figure 3.4responsible for transmitting secondary navigation parameters.The secondary navigation parameters are transmitted in message format.It identifies the message types

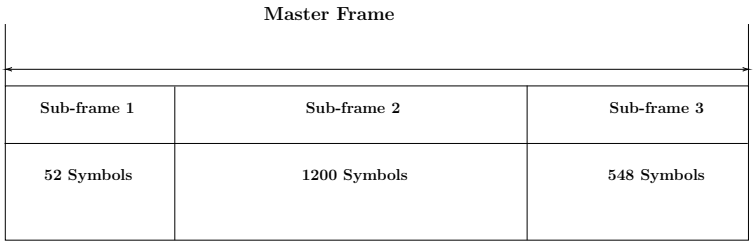


Figure 3.1: Master Frame Structure

that the NavIC satellites will transmit. Provision exists to define new messages for future requirements in NavIC. Each message is identified by a unique message identifier.

Figure 3.2: Sub-frame 1 Layout

3.1.1. L1 SPS DATA STRUCTURE

The NavIC Signal-In-Space transmits navigation message data through the SPS service, in the L1 band. The sub-frame structure is shown in Figure 3.5

The 600 bits of subframe 2 and 274 bits of subframe 3 are Low Density Parity Check (LDPC) Forward Error Correction (FEC) encoded and interleaved

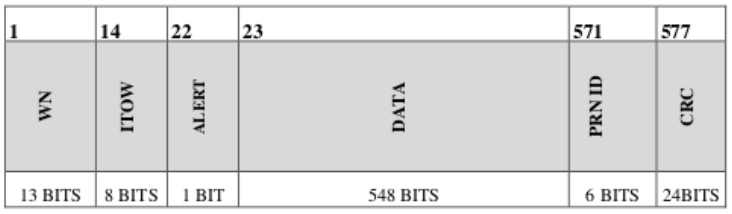


Figure 3.3: Sub-frame 2 Layout

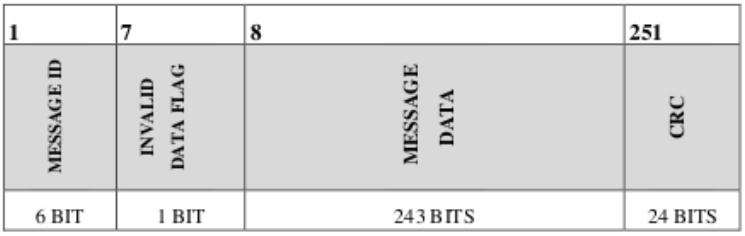


Figure 3.4: Sub-frame 3 Layout

as shown in Figure 3.5. Subframe 2 and subframe 3 are separately encoded using rate $\frac{1}{2}$ Quasi Cyclic LDPC codes. Subframe 1 consists of 9 bits that are Bose, Chaudhuri, and Hocquenghem (BCH) encoded. Subframe 2 has a total of 600 bits, comprising 576 bits for primary navigation parameters and 24 bits for CRC. Subframe 3 contains a total of 274 bits, with 250 bits for secondary navigation parameters and 24 bits for CRC. sAs a result of rate $\frac{1}{2}$ Quasi Cyclic LDPC encoding, there are 1200 symbols (coded bits) for subframe 2 and 548 symbols for subframe 3, as described in Figure 3.5.

3.2. Cyclic Redundancy Check(CRC)

The parity coding of data signal follows 24Q polynomial for each subframe. 24 bits of CRC parity will provide protection against burst as well as random

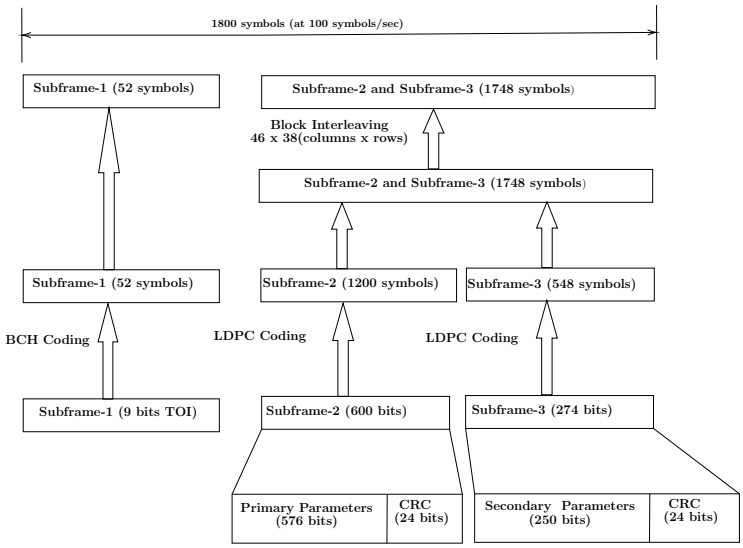


Figure 3.5: NavIC L1 SPS Subframe Structure

errors with undetected error probability of 2^{-24} for all channel bit error probabilities 0.5.

$$g(X) = \sum_{i=0}^{24} g_i X^i \tag{3.1}$$

$g_i = 1$; for $i = 0, 1, 3, 4, 5, 6, 7, 10, 11, 14, 17, 18, 23, 24$

$= 0$ otherwise

Chapter 4

Simulation Approach

The NavIC simulator approach is as shown in figure 4.1.

Figure 4.1: Transmitter Block diagram

Navigation data is randomly generated and subframes and master frames are created as per the frame structure described earlier. The transmitter module creates the required baseband signal as per the modulation scheme, with relevant channel encosing schemes. Channel modelling module adds various modelling parameteers and AWGN noise to the baseband signals for different satellites, forming a composite signal. The receiver module receives the composite signal, processes it to extract the navigation data, that was originally sent.

Chapter 5

Transmitter

The NavIC transmitter is simulated to send baseband signal to the channel as shown in Fig 5.1.

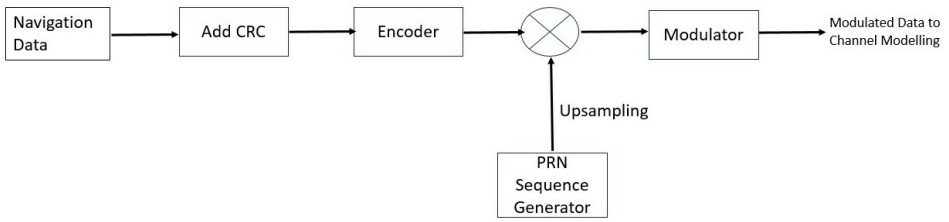


Figure 5.1: Transmitter Block diagram

5.1. Encoding

5.1.1. BCH Coding

To transmit nine bits of Time of Interval (TOI) data, we employ BCH $(52, 9)$ coding. The generator polynomial used in this encoding process is 1767 in octal notation. This coding scheme is illustrated conceptually in Figure 5.2 below, utilizing a 9-stage linear shift register generator.

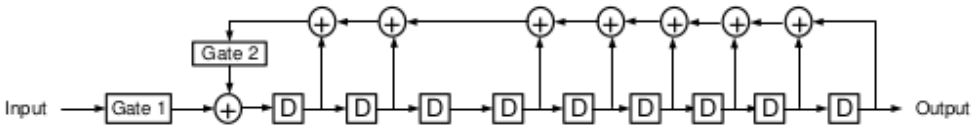


Figure 5.2: Diagram of the BCH Encoder Circuit

Encoding Process

Load TOI data bits 1 to 9 into the generator, starting with the Most Significant Bit (MSB).Shift the loaded data 52 times through the generator to generate 52 encoded symbols.The BCH (n, k) encoders are realized using k-stage registers as illustrated in Figure 5.2. During encoding, Gate 1 is closed for the initial k clock periods and then disconnected. Likewise, Gate 2 is disconnected during the first k periods and then closed.

Table 5.1: Generator Polynomials of BCH Encoders

BCH Code	Encoding Characteristics			Generator Polynomials (g(x))
	n	k	$d_{\min}$	
(52, 9)	52	9	20	$x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + x + 1$

5.1.2. LDPC

The LDPC (Low-Density Parity-Check) encoder structure is based on a parity-check matrix $H(m, n)$ consisting of m rows and n columns. Specifically, for subframe-2, $m = 600$ and $n = 1200$, and for subframe-3, $m = 274$ and $n = 548$ are selected.

The LDPC matrix H is assumed to be in an approximate lower triangular form with a dual diagonal structure. Matrix $H(m, n)$ is further decomposed

into six submatrices: A, B, T, C, D, and E, as illustrated in Figure 5.3 below.

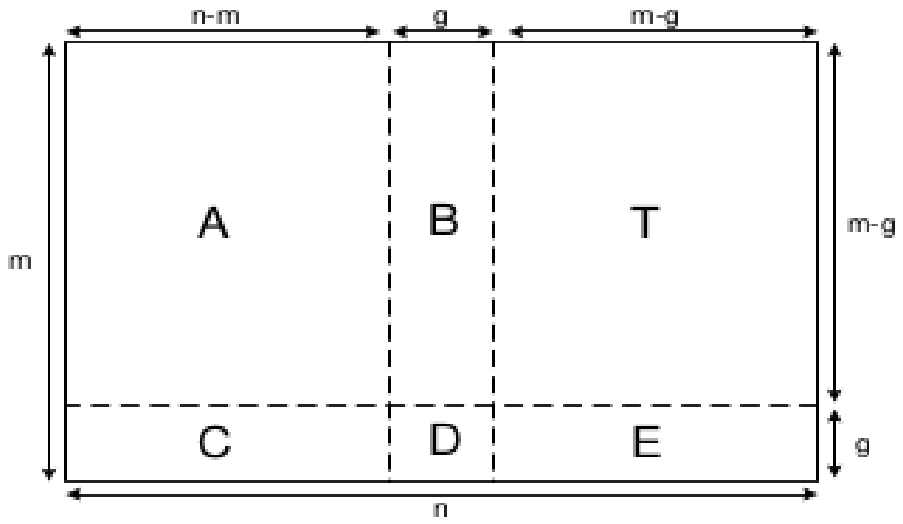


Figure 5.3: LDPC Matrix Structure

Each element of matrix $H(m, n)$ takes on either the value "0" or "1".The inverse of matrix T, denoted as T^{-1}, is not included in this document. However, it is worth noting that since T is a lower triangular matrix, its inverse can be readily identified.

For a rate 1/2 LDPC encoder, the encoding process utilizes the matrices A, B, T, C, D, and E to generate the encoded symbols based on the following algorithm:

$$p_1^t = -\varphi^{-1}(-E \cdot T^{-1} \cdot A + C) \cdot s^t \tag{5.1}$$

$$p_2^t = -T^{-1}(A \cdot s^t + B \cdot p_1^t) \tag{5.2}$$

Where:

$$\varphi = -E \cdot T^{-1} \cdot B + D$$

$s =$ subframe 2 and subframe 3 data

x_t indicates transpose

The elements of matrices p_1 and p_2 are modulo 2 numbers.

The encoded symbols for broadcast are composed of (s, p_1, p_2), where s represents the systematic portion of the codeword, and $\{p_1, p_2\}$ constitute the combined parity bits.

5.1.3. Interleaving

Any burst errors during the data transmission can be corrected by interleaving. In matrix interleaving, input symbols are filled into a matrix column-wise and read at the output row-wise. This will spread the burst error, if any, during the transmission.The 1748 symbols of LDPC encoded navigation data of subframe-2 and subframe-3 are interleaved using a block interleaver with n columns and k rows. Data is written in columns and then read in rows. The Table 5.2 below indicates the interleaving mechanism.

Parameter	**Arrangement**
Block Interleaver size	1748
Block Interleaver Dimensions (n columns x k rows)	46 x 38

Table 5.2: Interleaving Parameters

5.1.4. PRN codes for SPS

The NavIC L1 signal utilizes a family of Interleaved Z4 – Linear (IZ4) PRN spreading codes implemented using coupled shift registers. The PRN code has a length of 10230 chips with a code period of 10 ms in both data and pilot channels. Furthermore, the pilot channel incorporates a secondary overlay code with a length of 1800 and a period of 18 s. Importantly, the pilot and data signals are designed to be orthogonal. The IZ4 family of spreading codes has been found to deliver superior or equivalent performance compared to the PRN code families employed by GPS and BeiDou in the L1 band. Moreover, the resources required for implementing the code generator are of the same order as Weil codes.

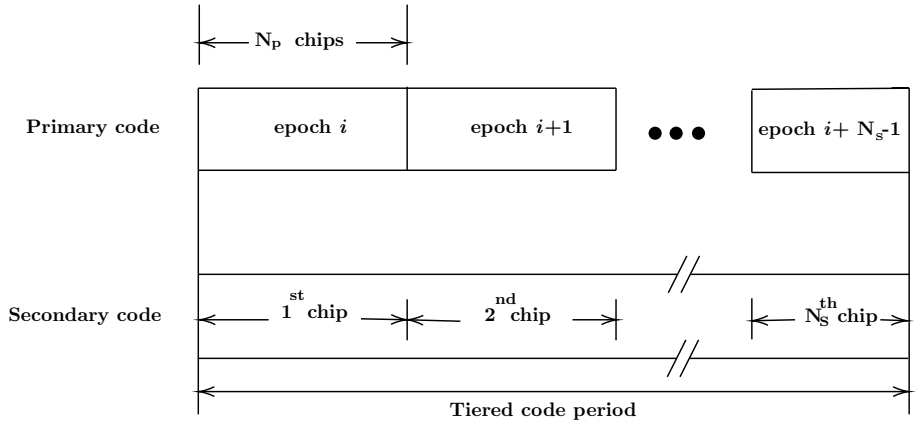


Figure 5.4: Tiered code structure and timing relationship between primary and secondary codes

Component	Primary Code Type	Primary Code Length	Primary Code Period (msec)	Secondary or Overlay Code Type	Secondary or Overlay Code Length	Secondary or Overlay Code Period (msec)
L1 Data (L1D i (t))	Z4–linear (IZ4)	10230	10	-	-	-
L1 Pilot (L1P i (t))	Interleaved Z4–linear (IZ4)	10230	10	Truncated Z4-linear sequences	1800	18000

Table 5.3: Characteristics of the L1 ranging codes

5.1.4.1. Code Generator Architecture for primary L1 Pilot and L1 Data PRN Code

The IZ4 ranging code generator comprises the following principal components:

1. Shift Registers R0 and R1: These are two fifty-five tap binary shift registers.

2. Shift Register C: This is a single, five-tap, binary, pure-cycling shift register.

Code Generation

The IZ4 code is generated as the chip-by-chip modulo-2 sum of the synchronized output of Shift Register C and Register R1. Specifically, it is computed using the following equation:

$$IZ4(t) = C(t,0) \oplus R1(t,0) \tag{5.3}$$

In this equation $C(t, 0)$ represents the first tap of Register C at time t, abbreviated as $C(0)$. $R1(t, 0)$ represents the contents of the first tap of Register R1 at time t, abbreviated as $R1(0)$. It's important to note that all three registers are synchronized with respect to each other, ensuring proper code generation. It's important to note that all three registers are synchronized with respect to each other, ensuring proper code generation.

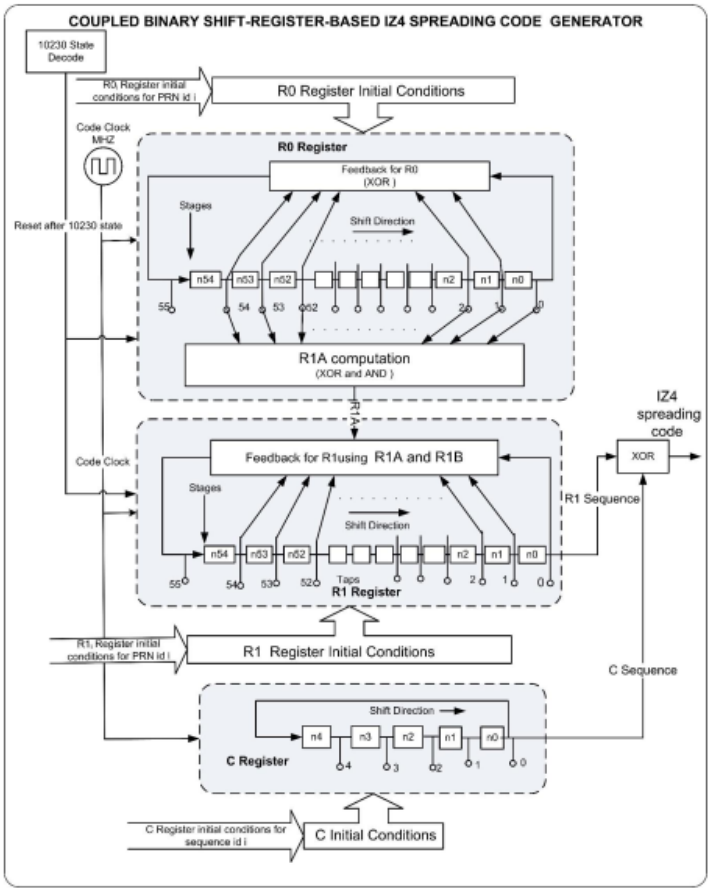


Figure 5.5: Functional Description of IZ4 Sequence generation using Binary shift registers.

Functional description of each register involved in the IZ4 code generation process:

5.1.4.2. Shift Register R0

A fifty-five tap long $R0$ shift register is the first component.This register generates binary codes with a period of 10230, shifting its contents at each clock cycle. The register's initial state is determined by stored initial conditions. It produces a binary code sequence with a 10230-chip period, resetting after 10230 cycles. Feedback operations are governed by a feedback polynomial, with the output fed back to the 55^{th} tap. The first tap,$R0(0)$, provides the component code output. In this process, seven out of 55 taps are employed, following the equation:

$$R0(54) = R0(50) \oplus R0(45) \oplus R0(40) \oplus R0(20) \oplus R0(10) \oplus R0(5) \oplus R0(0) \quad (5.4)$$

The feedback logic, illustrating inputs representing tap contents at time (t) and output corresponding to the 55th tap at $(t + 1)$, is depicted in Figure 5.6.

5.1.4.3. Shift Register R1

The shift register $R1$, consisting of fifty-five taps, serves as the source for generating the second component of the IZ4 ranging code. Similar to $R0$, it operates by producing a binary code with a period of 10230 through content shifts at each clock cycle. The initial state of $R1$ is determined by

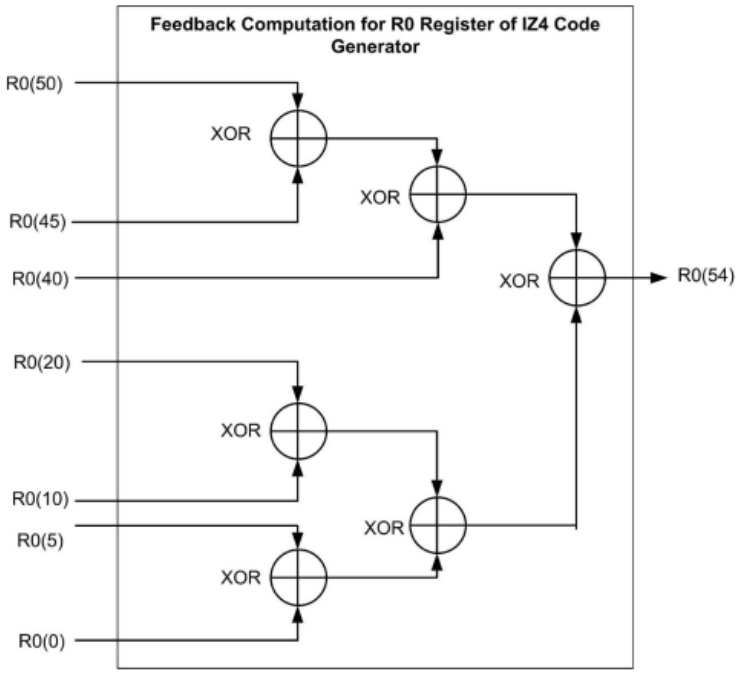


Figure 5.6: Feedback logic used to generate the linear feedback to Register R0 for IZ4 codes

stored initial conditions, and it resets after 10230 clock cycles. The output originates from the first tap.

Feedback to $R1$ encompasses both $R1A$ and $R1B$ components, computed as functions of the taps from both $R0$ and $R1$. The feedback to $R1$ is the modulo-2 sum of $R1A$ and $R1B$.

Specifically, three sub-components, $\sigma_2 A$, $\sigma_2 B$, and $\sigma_2 C$, which depend on the tap contents of $R0$, contribute to the computation of $R1A$ as per the following equations:

$$\sigma_2 A = [R0(50) \oplus R0(45) \oplus R0(40)] \text{ AND } [R0(20) \oplus R0(10) \oplus R0(5) \oplus R0(0)] \tag{5.5}$$

$$\sigma_2 B = ([R0(50) \oplus R0(45)] \text{ AND } R0(40)) \oplus ([R0(20) \oplus R0(10)] \text{ AND } [R0(5) \oplus R0(0)]) \tag{5.6}$$

$$\sigma_2 C = [R0(50) \text{ AND } R0(45)] \oplus ([R0(20) \text{ AND } R0(10)] \oplus [R0(5) \text{ AND } R0(0)]) \tag{5.7}$$

$$\sigma_2 = \sigma_2 A \oplus \sigma_2 B \oplus \sigma_2 C \tag{5.8}$$

$$R1A = \sigma_2 \oplus [R0(40) \oplus R0(35) \oplus R0(30) \oplus R0(25) \oplus R0(15) \oplus R0(0)] \tag{5.9}$$

$$R1B = R1(50) \oplus R1(45) \oplus R1(40) \oplus R1(20) \oplus R1(10) \oplus R1(5) \oplus R1(0) \tag{5.10}$$

$$R1(54) = R1A \oplus R1B \tag{5.11}$$

In equations 5.5 through 5.10, all quantities represent contents or functions of register contents at time t. Equation 5.11 computes $R1(54)$ at time $t+1$ based on the quantities on the right side at time t. All registers initialize with initial conditions at $t = 0$. Implementation of the feedback computation for Shift Register $R1$, where $R1A$ and $R1B$ are computed at time t, and $R1(54)$ refers to the contents $R1(t+1, 54)$ at time $t+1$ of the 55th tap of Register $R1$.

5.1.4.4. Initial conditions for Shift Registers R0, R1 and C

Each R0 and R1 component code generator uses 55 bits of unique initial conditions stored in memory.Shift Register C is a 5-tap pure-cycling register, where the first tap's output, denoted as C(0), is looped back as input to the fifth tap. C(0) is XORed with the output of the R1 shift register on a chip-by-chip basis to produce the IZ4 code. Initial five-bit conditions for Register C are provided in Tables 7 and 8, corresponding to each PRN code.

5.1.5. Secondary Overlay Code Generator Architecture

The secondary or overlay codes linked to each L1 pilot primary code have a length of 1800 chips. These overlay codes are independently synchronized in time and have a duration of 18 seconds, operating at a rate of 100 bps. The 1800-chip overlay codes are produced by cyclically cycling the Z4-linear codes, which have a period of 2046. The generation process for overlay codes resembles that of primary IZ4 codes. To generate overlay codes, two ten-tap shift registers, denoted as R0 and R1, are utilized with specific feedback polynomials. Unlike primary code generation, the C register is not necessary for the generation of overlay codes.

$$R0(9) = R0(5) \oplus R0(2) \oplus R0(1) \oplus R0(0) \tag{5.12}$$

$$\sigma_{2A} = [R0(5) \oplus R0(2)] \text{ AND } [R0(1) \oplus R0(0)] \tag{5.13}$$

$$\sigma_{2B} = [R0(5) \text{ AND } R0(2)] \oplus [R0(1) \text{ AND } R0(0)] \tag{5.14}$$

$$R1A = \sigma_2 \oplus R0(6) \oplus R0(3) \oplus R0(2) \oplus R0(0) \tag{5.15}$$

$$R1B = R1(5) \oplus R1(2) \oplus R1(1) \oplus R1(0) \tag{5.16}$$

$$R1(9) = R1A \oplus R1B \tag{5.17}$$

In Figures 5.12,5.13, and 5.14 below, the feedback computation process for R_0 and R_1 shift registers for overlay codes is explained. Figure 5.12 demonstrates how the feedback for Register R_0 is performed, along with the computation of certain components of the feedback for Register R_1. The output of Register R_1 corresponds to the overlay code. Quantities shown on the left, such as $R_0(5)$ in the topmost sub-figure of Figure 5.12, represent the contents of the respective registers at time t, while the quantity $R_0(9)$ on the opposite side represents the contents of R_0 at time $t + 1$ for position 9.

5.2. Modulation

5.2.1. Standard Positioning Service

The SPS signal is modulated using Synthesized Binary Offset Carrier (SBOC) in L1 band and BPSK in L5 and S bands.

5.2.2. Baseband Modulation

SBOC modulation contains BOC(1,1) and BOC(6,1) components in both data signal and pilot signals. In this scheme, the data channel's BOC(1,1), the pilot channel's BOC(1,1), and the pilot channel's BOC(6,1) components are interplexed to create the data channel's BOC(6,1) component. Data and pilot signals are quadrature multiplexed, with 41.82% power to data and 58.18% to pilot, ensuring constant envelope modulation for MBOC.

5.2.2.1. Mathematical Equations

The mathematical representation of baseband navigation signals is as follows:

Pilot Signal:

$$S_{p,a}(t) = \sum_{i=-\infty}^{\infty} C_p(|i|L_p) \cdot \mathrm{rect}_{T_{c,p}}\left(t - iT_{c,p}\right) \cdot s_{c,p,a}(t,0) \tag{5.18}$$

$$S_{p,b}(t) = \sum_{i=-\infty}^{\infty} C_p(|i|L_p) \cdot \mathrm{rect}_{T_{c,p}}\left(t - iT_{c,p}\right) \cdot sc_{p,b}(t,0) \tag{5.19}$$

The sub-carrier is defined as:

$$sc_{p,x}(t,\varphi) = \mathrm{sgn}[\sin(2\pi f_{sc,x^t} + \varphi)] \tag{5.20}$$

Ranging code C_p, defined in equations (5.18) and (5.19), includes primary code and secondary overlay code.

Data Signal:

$$S_{d,a}(t) = \sum_{i=-\infty}^{\infty} C_d(|i|_{L_d}) \cdot d_d([i]_{CD_d}) \cdot \mathrm{rect}_{T_{c,d}}\left(t - iT_{c,d}\right) \cdot sc_{d,a}(t,0) \tag{5.21}$$

The sub-carrier is defined as:

$$sc_{d,x}(t,\varphi) = \mathrm{sgn}[\sin(2\pi f_{sc,x^t} + \varphi)] \tag{5.22}$$

The subcarrier signals are sinBOC. Hence, the subcarrier phase $\varphi = 0$.

The interplexed component ($S_{d,b}(t)$ is given by:

$$S_{d,b}(t) = \sum_{i=-\infty}^{\infty} C_d(|i|L_d) \cdot d_d([i]C_{D_d}) \cdot \mathrm{rect}_{T_{c,d}}\left(t - iT_{c,d}\right) \cdot sc_{d,b}(t,0) \tag{5.23}$$

The sub-carrier is defined as:

$$sc_{d,x}(t,\varphi) = \mathrm{sgn}[\sin(2\pi f_{sc,x^t} + \varphi)] \tag{5.24}$$

The subcarrier signals are sinBOC. Hence, the subcarrier phase $\phi = 0$.
The composite SBOC modulated signal ($S(t)$ is generated by quadrature multiplexing of data and pilot signals, as given below:

$$S(t) = [\alpha S_{p,a}(t) - \beta S_{p,b}(t)] + j[\gamma S_{d,a}(t) + \eta S_{d,b}(t)] \tag{5.25}$$

The baseband composite SBOC modulated signal ($S(t)$ can also be denoted as:

$$S(t) = S_I(t) + jS_Q(t) \tag{5.26}$$

Based on Equation 5.26, the band-pass representation of the SBOC modulated navigation signal ($S_{RF}(t)$) at L1 band is defined as follows:

$$S_{RF}(t) = S_I(t) \cdot \cos(2\pi f_{L1} t) - S_Q(t) \cdot \sin(2\pi f_{L1} t) \tag{5.27}$$

where f_{L1} is the frequency of 1575.42 MHz.

Symbol	Description
$C_d(i)$	i'th chip of spreading code of data channel
$C_p(i)$	i'th chip of spreading code of pilot channel
$d_d(i)$	i'th bit of navigation message of data channel
$sc_{p,x}(t)$	Binary NRZ subcarrier for pilot channel
$sc_{d,x}(t)$	Binary NRZ subcarrier for data channel
$\|i\|_X$	i modulo X
$[i]_X$	Integer part of (i/X)
CD_x	No. of chips per navigation data bit
L_x	Length of spreading code in chips
$rect_x(t)$	Rectangle pulse function with duration x
$T_{c,x}$	Spreading code chip duration
$f_{sc,x}$	Subcarrier frequency
φ	Subcarrier phase

Table 5.5: Symbol Description

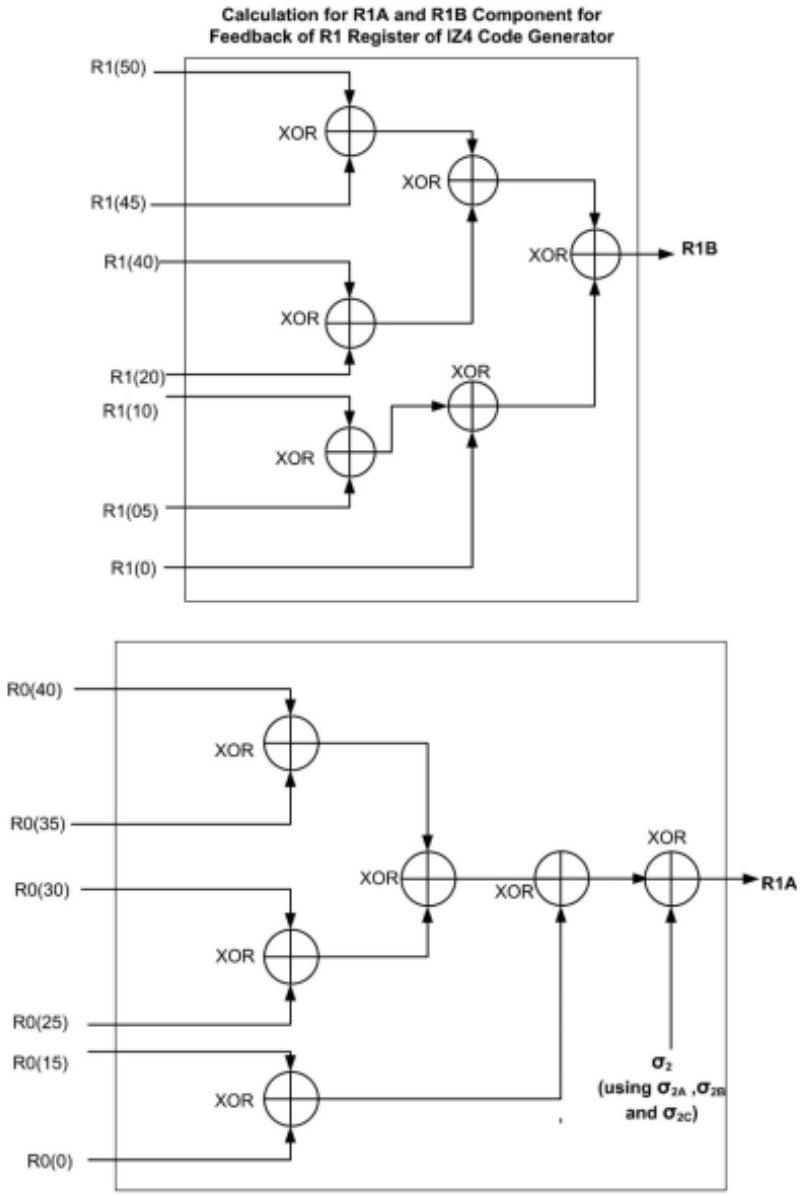
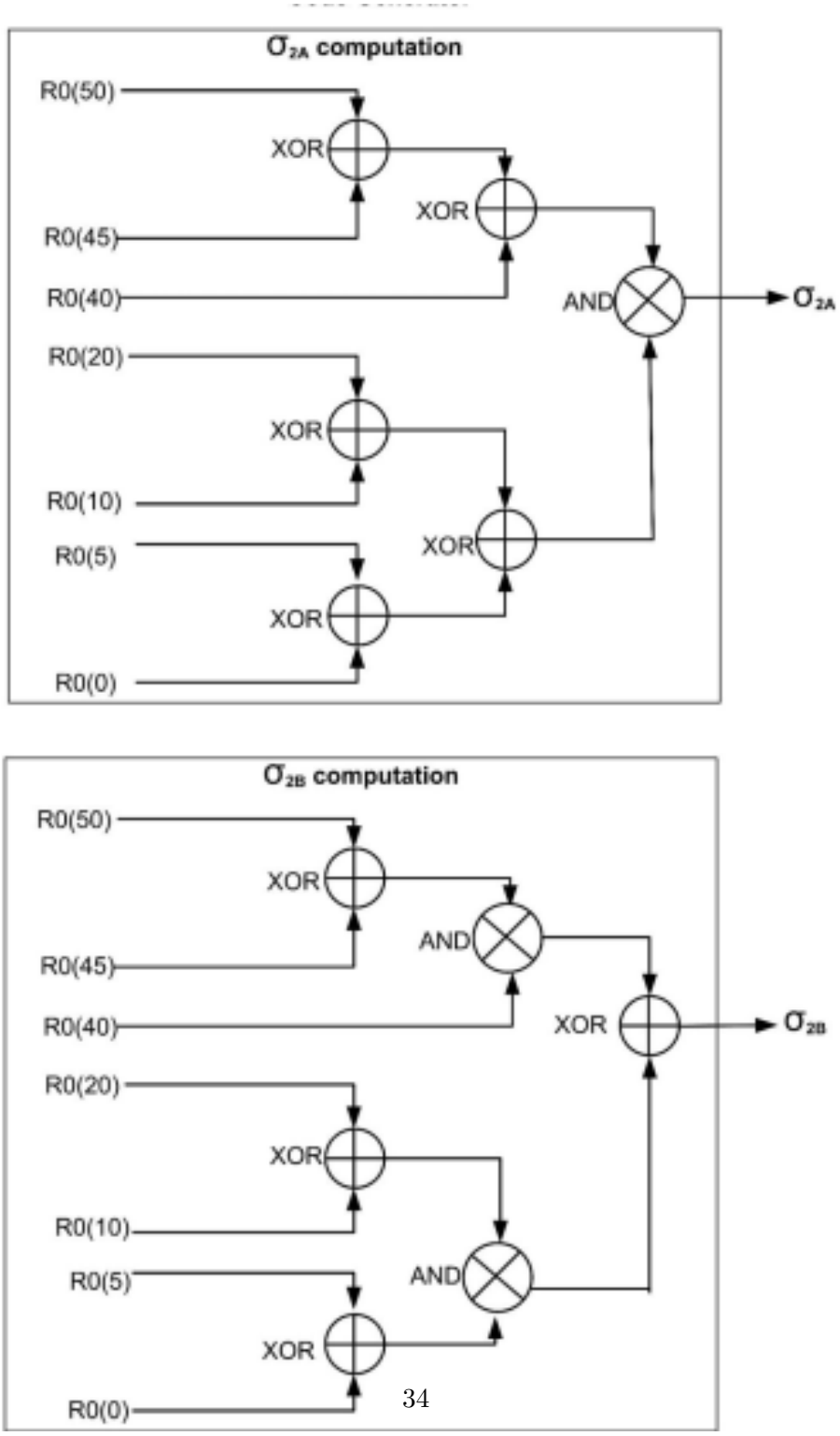


Figure 5.7: Feedback Computation for R0 and R1A Determination for Overlay Codes


σ2A computation
R0(50)
XOR
R0(45)
XOR
R0(40)
AND
σ2A
R0(20)
XOR
R0(10)
XOR
R0(5)
XOR
R0(0)
σ2B computation
R0(50)
XOR
R0(45)
AND
R0(40)
XOR
σ2B
R0(20)
XOR
R0(10)
AND
R0(5)
XOR
R0(0)

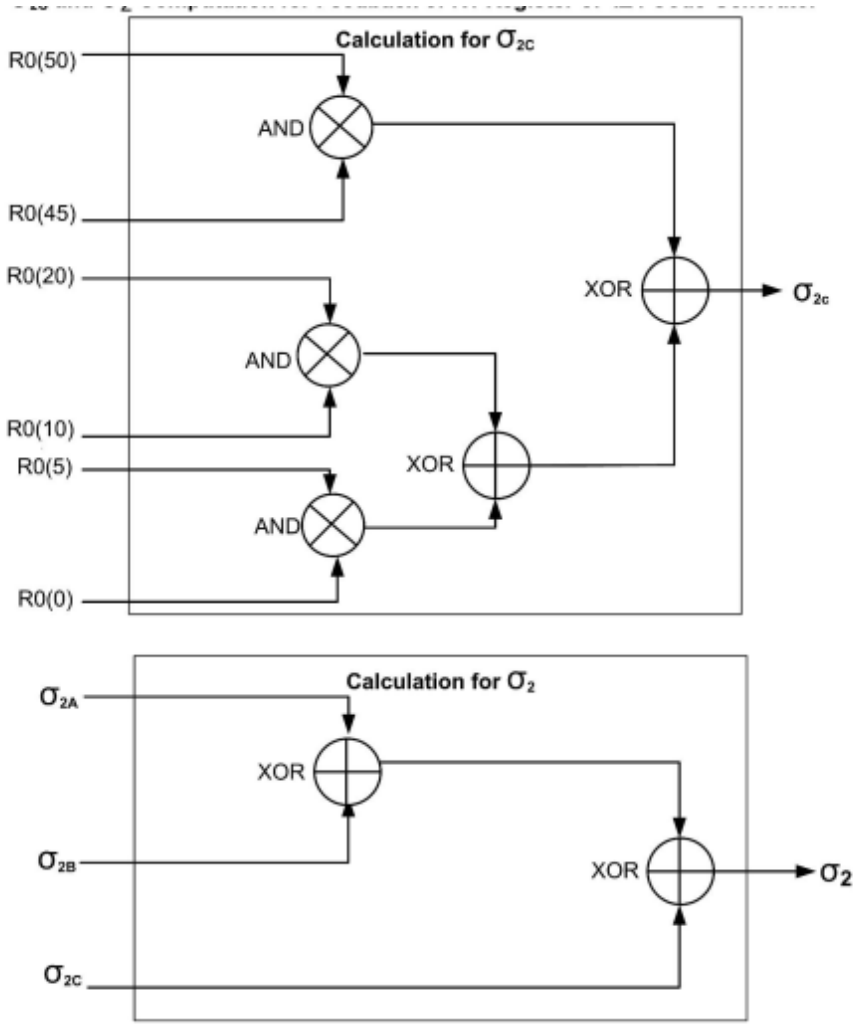


Figure 5.9: *sigma*2*C*and *sigma*2 Computation for R1A Determination for Primary IZ4 Codes

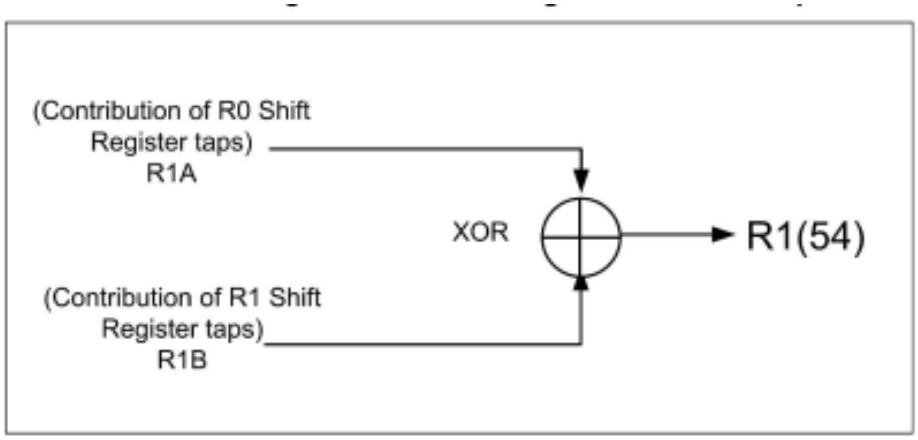


Figure 5.10: R1 Feedback Computation Using R1 and R0 Shift Register Taps for Primary IZ4 Codes

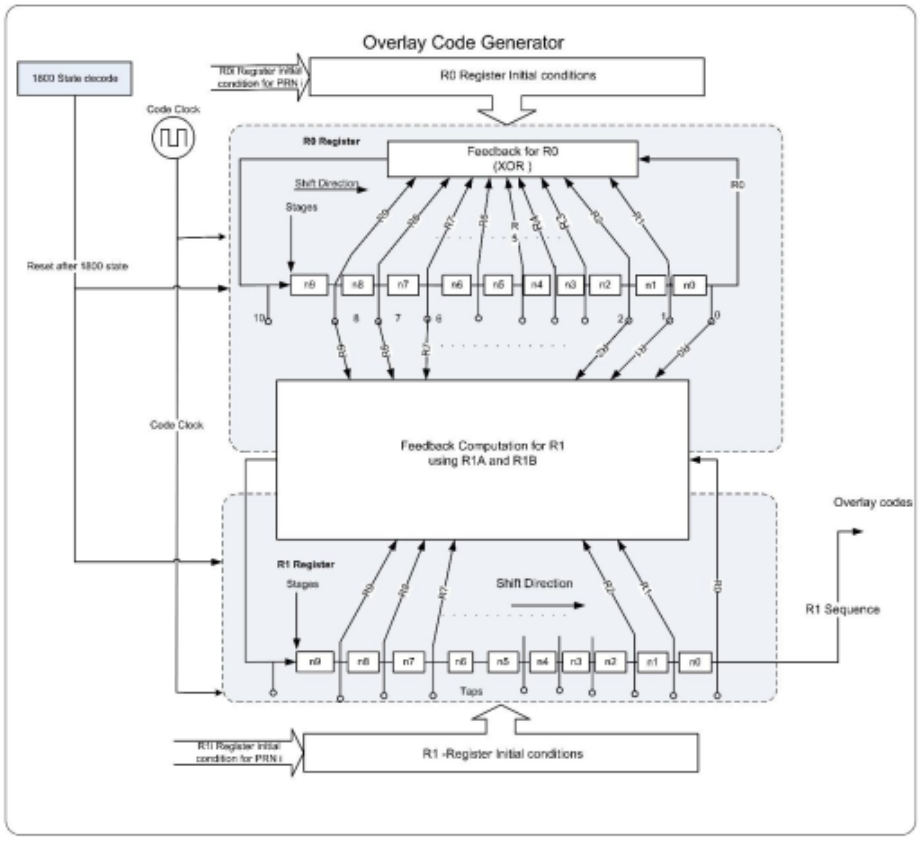


Figure 5.11: Block Diagram of overlay sequence generator

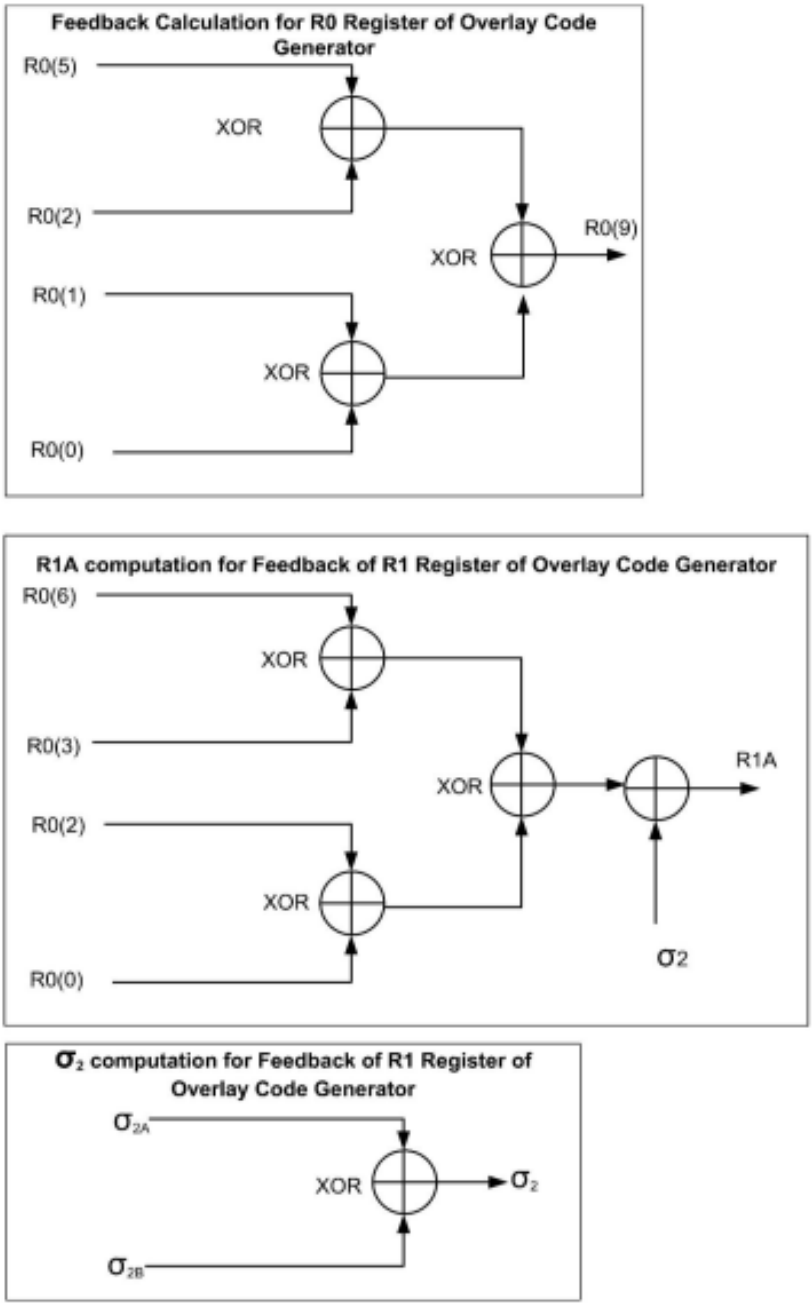


Figure 5.12: Feedback Computation for R0 and R1A Determination for Overlay Codes

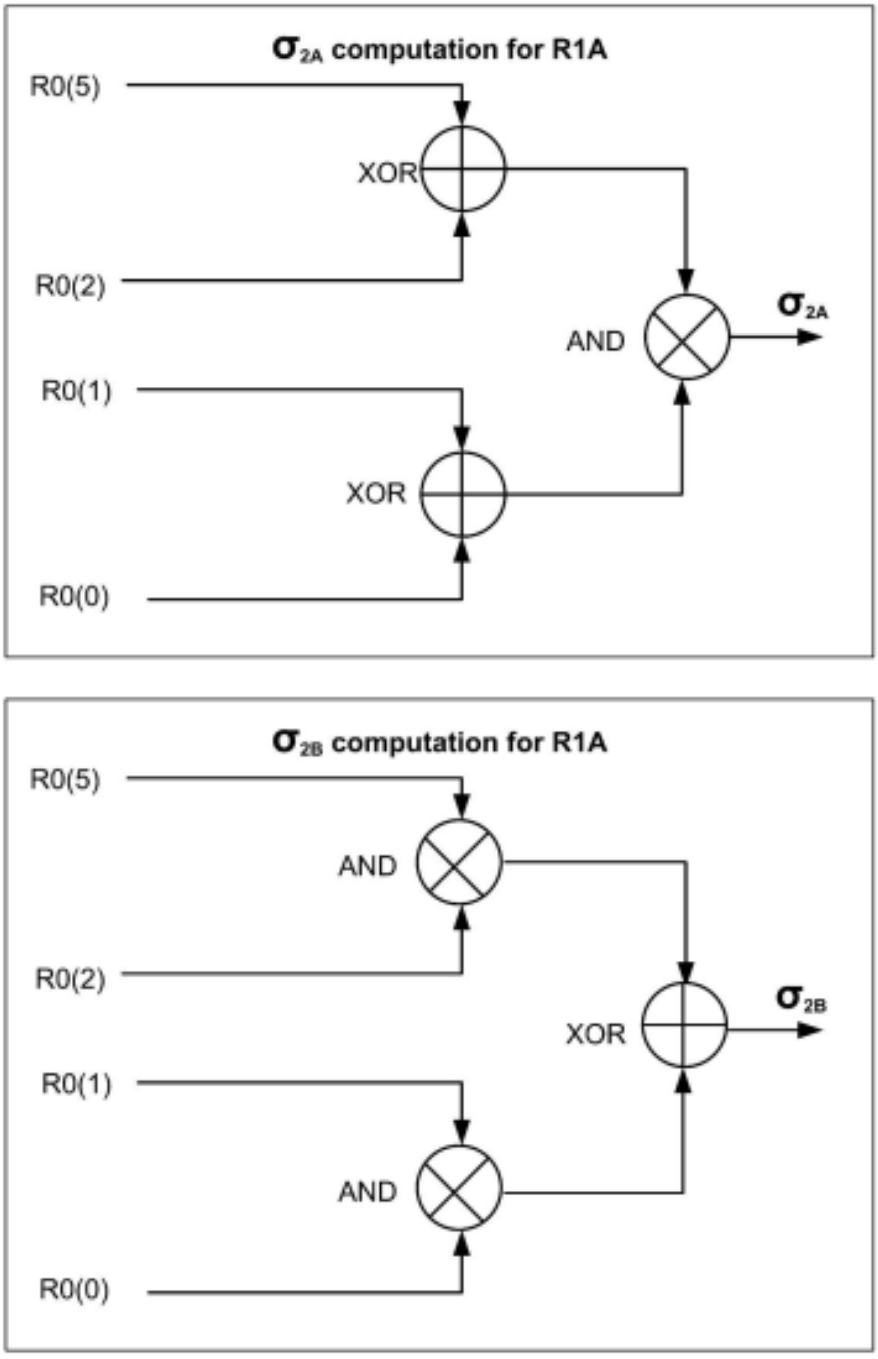


Figure 5.13: $\sigma 2A$ and $\sigma 2B$ Computation for Overlay Codes

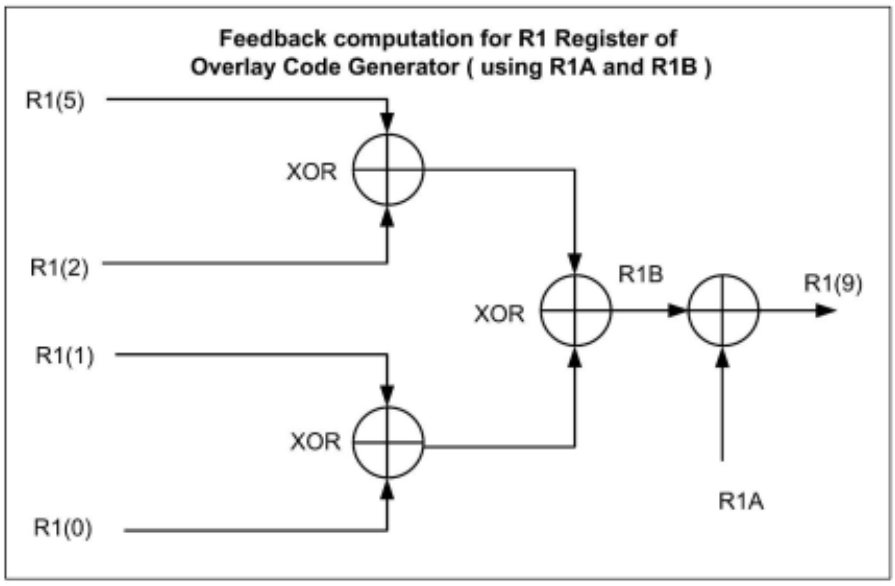


Figure 5.14: R1 Register Feedback Computation for Overlay Codes

Chapter 6

Channel Modelling

The phenomena modelled in the satellite communication channel are

1. Doppler shift
2. Delay
3. Power scaling and
4. Thermal noise at the receiver

6.1. Doppler shift

Due to relative motion between the satellites and the receiver, the transmitted signals undergo a frequency shift before arriving at the receiver. This shift in frequency is called Doppler shift and can be computed as

$$f_{shift} = f_d - f_c = \left(\frac{V_{rel}}{c - V_{S,dir}} \right) f_c \tag{6.1}$$

where,

f_{shift} = Frequency shift due to Doppler effect

f_d = Frequency observed at receiver

f_c = Carrier frequency at transmitter

V_{rel} = Relative velocity of transmitter and receiver

$V_{S,dir}$ = Velocity of satellite along radial direction

c = Speed of light

V_{rel} is given by

$$V_{rel} = V_{S,dir} - V_{R,dir} \tag{6.2}$$

where,

$V_{R,dir}$ = Velocity of receiver along radial direction

$V_{R,dir}$ and $V_{S,dir}$ are given by

$$V_{R,dir} = \mathbf{V}_R \cdot \hat{\mathbf{dir}} \tag{6.3}$$

$$V_{D,dir} = \mathbf{V}_S \cdot \hat{\mathbf{dir}} \tag{6.4}$$

where,

$\hat{\mathbf{dir}}$ = Unit vector from satellite to receiver i.e. radial direction

$\mathbf{V_S}$ = Velocity of satellite

$\mathbf{V_R}$ = Velocity of receiver

$\hat{\mathbf{dir}}$ is given by

$$\hat{\mathbf{dir}} = \frac{\mathbf{P_S} - \mathbf{P_R}}{\|\mathbf{P_S} - \mathbf{P_R}\|} \tag{6.5}$$

where,

$\mathbf{P_S}$ = Position of satellite

$\mathbf{P_R}$ = Position of receiver

The Doppler shift is introduced by muliplying the satellite signal with a complex exponential,

$$x_{Shift}[n] = x[n]\, e^{-2\pi j\left(f_c + f_{Shift}\right)nt_s} \tag{6.6}$$

where,

$x_{Shift}[n]$ = Doppler shifted signal

$x[n]$ = Satellite signal

t_s = Sampling period

6.2. Delay

Since there is a finite distance between the satellite and the receiver, the signal at the reciever is a delayed version of the transmitted signal. This delay is given by

$$D_s = \frac{d}{c} f_s \tag{6.7}$$

where,

D_s = Total delay in samples

d = Distance between satellite and receiver

c = Speed of light

f_s = Sampling rate

The total delay on the satellite signal is modeled in two steps. First, a

static delay is modeled which does not change with time and it is always an integer number of samples. Then, a variable delay is modeled which can be a rational number of samples. While modelling the static delay, the entire delay is not introduced so that variable delay modelling handles the remaining delay.

To introduce the static delay, the samples are read from a queue whose size is the desired static delay length. When samples are read from the queue, an equal number of new samples are updated in the queue. To introduce the variable delay, the signal is passed through an all-pass FIR filter with an almost constant phase response. Its coefficients are calculated using the delay value required.

6.3. Power Scaling

When a transmitting antenna transmits radio waves to a receiving antenna, the radio wave power received is given by,

$$P_r = P_t D_t D_r \left(\frac{1}{4\pi \left(f_c + f_{Shift} \right) D} \right)^2 \tag{6.8}$$

where,

P_r = Received power

P_t = Transmitted power

D_t = Directivity of transmitting antenna

D_r = Directivity of receiving antenna

D = Total delay in seconds

To scale the received signal as per the received power calculated,

$$x_{Scaled}[n] = \frac{\sqrt{P_r}}{\text{rms}(x[n])} x[n] \tag{6.9}$$

6.4. Thermal noise

The thermal noise power at the receiver is given by,

$$N_r = kTB \tag{6.10}$$

where,

N_r = Noise power in watts

k = Boltzmann's constant

T = Temperature in Kelvin

B = Bandwidth in Hz

AWGN (Additive White Guassian Noise) samples with zero mean and variance N_r are generated and added to the satellite signal to model thermal noise at receiver.

Chapter 7

Receiver

The signal processing chain at the receiver are divided into four steps:

1. Signal acquisition
2. Signal tracking
   (a) Carrier Tracking
   (b) Code Tracking
3. Signal demodulation
4. Channel decoding

The signal processing part for NavIC signals at receiver are as shown in figure 7.1.

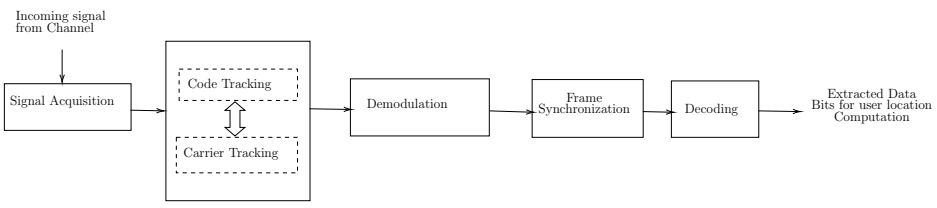


Figure 7.1: The Block Level Architecture for Receiver

1. **Signal acquisition:** The receiver searches for and acquires the NavIC signal for a given satellite(s) by correlating the received signal with a locally generated replica of the spreading code used by the satellite(s). This process helps in identifying the presence of the NavIC signal and estimating coarse value of both doppler frequency shift and code delay.

2. **Carrier tracking:** Once the signal is acquired, the receiver performs carrier tracking to estimate and track the carrier frequency and phase of the received signal. This is crucial for demodulation as it ensures accurate demodulation of the navigation message and ranging signal.

3. **Code delay tracking:** The receiver performs code delay tracking to estimate and track the spreading code used by the satellites. This helps in maintaining synchronization with the transmitted signal and extracting the navigation data and ranging information.

4. **Signal demodulation:**After the aquisition and tracking has been performed, the received data is mapped back using BPSK demodulation, mapping -1 to binary 1 and $+1$ to binary 0.

5. **Frame Synchronization:** The pilot PRN for every satellite is locally generated to determine the start of frame.

6. **Signal decoding:** Once the signal has been demodulated, the decoding is performed removing all the extra bits that were added to navigation data during the encoding process.

7.1. Signal Acquistion

The role of the aqusition block is to examine the presence/absence of signals coming from a given satellite. In the case of signal being present, it should provide coarse estimations of the Code delay and the Carrier Doppler shift, yet accurate enough to initialize the frequency and code tracking loops.

A generic IRNSS signal defined by its complex baseband equivalent, $S_T(t)$, the digital signal at the input of an Acquisition block can be written as:

$$x_{IN}[k] = A(t)\hat{s}_T(t-\tau(t))e^{j(2\pi f_D(t)t+\Phi(t))}\Big|_{t=kT_s} + n(t)\Big|_{t=kT_s} \tag{7.1}$$

Symbol	Definition
$x_{IN}[k]$	Complex vector I, Q samples of received signal
A(t)	Signal Amplitude
$\hat{s}_T(t)$	filtered version of $s_T(t)$
$f_D(t)$	Time varying doppler shift
$\Phi(t)$	Time varying carrier phase shift
$\tau(t)$	Time varying code delay
n(t)	Time varying random noise
T_s	Sampling period

Table 7.2: Parameters Table in Signal Acquisition

7.1.1. Implementation of CA PCPS Acquisition

The Parallel Code Phase Search (PCPS) algorithm is used in Acquisition block and is depicted in figure 7.2 and described as follows:

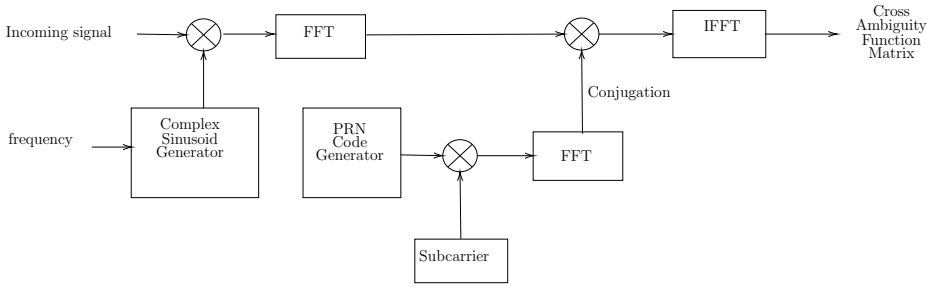


Figure 7.2: PCPS algorithm flow

Given:

1. Input signal buffer x_{IN} of K complex samples, provided by the Signal Conditioner

2. On-memory FFT of the local replica

$$D[k] = FFT_K\{d[k]\} \tag{7.2}$$

3. Acquisition threshold γ

4. Frequency span : $[f_{min}, f_{max}]$

5. Frequency step : f_{step}

Expected:

1. Find out if signal is acquired or not for a given satellite(s)

2. If signal is acquired, for each given satellite, calculate coarse estimation of Doppler shift $\hat{f}_{D_{acq}}$ and Code delay $\hat{\tau}_{acq}$

Algorithm:

1. Calculate input signal power estimation $\hat{P}_{in} = \frac{1}{K}\sum_{k=0}^{K-1}\left|x_{IN}[k]\right|^2$

2. for $\check{f}_D = [f_{min} to f_{max}]$ in f_{steps}

   (a) Calculate carrier wipe off $x[k] = x_{IN}[k]e^{-(j2\pi\check{f}_D kT_s)}$,for $k = 0, ..., K-1$

   (b) Calculate $X[k] = FFT_K\{x[k]\}$

   (c) Calculate $Y[k] = X[k].D[k]$, for $k = 0, ...K-1$

   (d) Calculate corresponding column in the Cross ambiguity function matrix - $R_{xd}(\check{f}_D, \tau) = \frac{1}{K^2}IFFT_K\{Y[k]\}$

3. Search maximum and its indices in the search grid:

$$\{S_{max}, f_i, \tau_j\} = max_{f,\tau}\left|R_{xd}(f,\tau)\right|^2 \tag{7.3}$$

4. Calculate the Generalized Likelihood Ratio Test (GLRT) function with normalized variance:

$$\Gamma_{GLRT} = \frac{2KS_{max}}{\hat{P}_{in}} \tag{7.4}$$

5. if $\Gamma_{GLRT} > \gamma$

   Declare positive acquisition and provides coarse estimation of code de-

lay $\hat{\tau}_{acq} = \tau_j$ and Doppler shift $\hat{f}_{D_{acq}} = f_i$,

other wise declare negative acquisition.

7.2. Tracking

The role of tracking block is to follow signal synchronization parameters: code phase, Doppler shift and carrier phase and extract the baseband signal. It performs the following 3 function to decipher the baseband signal from the incoming signal as shown in figure 7.3.

1. Carrier and code wipeoff
2. Pre-detection integration
3. Baseband signal processing

7.2.1. Carrier and code wipeoff

Carrier wipeoff: Referring to the figure 7.3, first the incoming signal is stripped off the carrier (plus carrier Doppler) by the replica carrier (plus carrier Doppler) signals. The replica carrier (including carrier Doppler) signals are synthesized by the carrier numerically controlled oscillator (NCO). In closed loop operation, the carrier NCO is controlled by the carrier tracking loop in the receiver processor.

Code wipeoff: The received signal is then correlated with very early(VE),

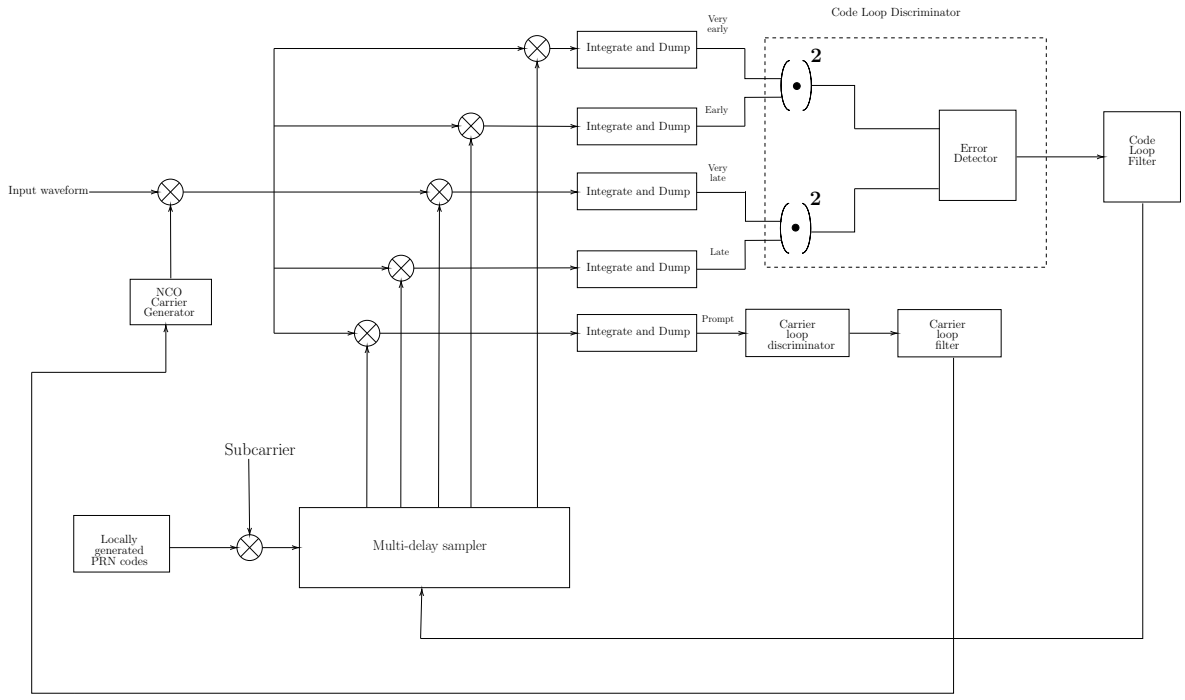


Figure 7.3: Tracking block diagram

early(E), prompt(P), late(L) and very late(VL) replica codes (plus code Doppler) synthesized by a multi-delay sampler. In the closed loop operation, the code NCO is controlled by the code tracking loop in the receiver processor. E and L are typically separated in phase by 0.3 chips and P is in the middle. VE and VL are separated by 1.2 chips. The prompt replica code phase is aligned with the incoming satellite code phase producing maximum correlation if it is tracking the incoming satellite code phase. Under this circumstance, the early phase is aligned a fraction of a chip period early, and the late phase is aligned the same fraction of the chip period late with respect to the incoming code phase, and these correlators produce about half the maximum correlation. Any misalignment in the replica code phase

with respect to the incoming code phase produces a difference in the vector magnitudes of the early and late correlated outputs so that the amount and direction of the phase change can be detected and corrected by the code tracking loop.

7.2.2. Pre-detection and integration

Extensive digital predetection integration and dump processes occur after the carrier and code wiping off processes. Figure 7.3 shows five complex correlators required to produce five components, which are integrated and dumped to produce very early, early, prompt, late and very late vesions of the signal . The carrier wipeoff and code wipeoff processes must be performed at the digital IF sample rate, while the integrate and dump accumulators provide filtering and resampling at the processor baseband input rate, which can be at 1,000 Hz during search modes or as low as 50 Hz during track modes, depending on the desired dwell time during search or the desired predetection integration time during track.

7.2.3. Baseband signal processing

This entails Carrier tracking and Code tracking using Phase locked loop (PLL), Frequency locked loop (FLL) and Delay locked loop (DLL).

7.2.3.1. Carrier tracking loop

Phase locked loop(PLL)

The carrier loop discriminator defines the type of tracking loop as a PLL, a Costas PLL (which is a PLL-type discriminator that tolerates the presence of data modulation on the baseband signal), or a frequency lock loop (FLL). Carrier tracking loop tracks the frequency and phase of the received signal by detecting the phase error between replicated signal and incoming signal and accordingly replicated signal produced by numerically controlled oscillator (NCO) is adjusted to synchronize with incoming signal in both frequency and phase. For zero phase error detected, navigation data is accurately extracted.

$$\text{Phase error}_{pilot} = ATAN2(I_P, Q_P) = \tan^{-1}\left(\frac{I_P}{Q_P}\right) \tag{7.5}$$

$$\text{Phase error}_{data} = ATAN2(Q_P, I_P) = \tan^{-1}\left(\frac{Q_P}{I_P}\right) \tag{7.6}$$

The ATAN2 discriminator is the only one that remains linear over the full input error range of $\pm 180^\circ$. However, in the presence of noise, both of the discriminator outputs are linear only near the 0° region. These PLL discriminators will achieve the 6-dB improvement in signal tracking threshold (by comparison with the Costas discriminators) for the dataless carrier because they track the full four quadrant range of the input signal.

Frequency locked loop

PLLs replicate the exact phase and frequency of the incoming SV (converted to IF) to perform the carrier wipeoff function. FLLs perform the carrier wipeoff process by replicating the approximate frequency, and they typi-

cally permit the phase to rotate with respect to the incoming carrier signal. The algorithm used in FLL discriminator is $\frac{\text{ATAN2}(cross,dot)}{t_2-t_1}$. The frequency error is given by

$$\text{Frequency error} = \frac{\phi_2 - \phi_1}{t_2 - t_1} \tag{7.7}$$

The phase change $\phi_2 - \phi_1$ between two adjacent samples of I_{PS} and Q_{PS} at times t_2 and t_1 is computed. This phase change in a fixed interval of time is proportinal to frequenct error in the carrier tracking loop. The error is fed to carier NCO to adjust the frequency to lock to the right frequency.

7.2.3.2. Code tracking loop

Delay locked loop: Post the carrier signal synchronization, received CA code samples are synchronized by aligning with replicated CA code samples by shifting right or left. To determine the direction of shift, the I and Q outputs are multiplied with prompt code (PRN code which is phase aligned), early code (prompt PRN code shifted by some samples to the right) and late code (prompt PRN code shifted by some samples to the left) resulting in corresponding to I and Q channel respectively. Following algorithm is used to lock the code phase.

$$E_K = \sqrt{VE^2 + E^2} \tag{7.8}$$

$$L_K = \sqrt{VL^2 + L^2} \tag{7.9}$$

$$\text{DLL Discriminator}(\epsilon) = \frac{1}{2}\frac{E_K - L_K}{E_K + L_K} \tag{7.10}$$

If the replica code is aligned, then the early and late envelopes are equal in amplitude and no error is generated by the discriminator. If the replica code is misaligned, then the early and late envelopes are unequal by an amount that is proportional to the amount of code phase error between the replica and the incoming signal (within the limits of the correlation interval). The code discriminator senses the amount of error in the replica code and the direction (early or late) from the difference in the amplitudes of the early and late envelopes. This error is filtered and then applied to the code loop NCO, where the output code shift is increased or decreased as necessary to correct the replica code generator phase with respect to the incoming SV signal code phase.

7.2.3.3. Loop filter characteristics

The values for the second-order coefficient a_2 and third-order coefficients a_3 and b_3 can be determined from Table 3. These coefficients are the same for FLL, PLL, or DLL applications if the loop order and the noise bandwidth,B_n , are the same.Note that the FLL coefficient insertion point into the filter is one integrator back from the PLL and DLL insertion points.This is because the FLL error is in units of hertz (change in range per unit of time).

Loop Order	**Noise Bandwidth B_n (Hz)**	**Typical Filter Values**
First	$\frac{\omega_o}{4}$	ω_o $B_n = 0.25\omega_o$
Second	$\frac{\omega(1+a_2^2)}{4a_2}$	ω_o^2 $a_2\omega_o = 1.414\omega_o$ $B_n = 0.53\omega_o$
Third	$\frac{\omega(a_3b_3^2+a_3^2-b_3)}{4(a_3b_3-1)}$	ω_o^3 $a_3\omega_o^2 = 1.1\omega_o^2$ $b_3\omega_o = 2.4\omega_o$ $B_n = 0.7845\omega_o$

Table 7.4: Loop order filters

7.2.3.4. Implemented tracking algorithm.

Complex sample stream, x_{IN} ; estimations of code phase $\hat{\tau}_{acq}$ and Doppler shift $\hat{f}_{d_{acq}}$; buffer size for power estimation, U; carrier lock detector threshold, τ ; $CN0_{min}$; maximum value for the lock fail counter, ϑ; cor- relators spacing ϵ and ϵ' ; loop filters bandwidth BW_{DLL} and BW_{PLL} ; integration time T_{int}. Using $\hat{\tau}_{\text{acq}}$ and a sample counter N, skip samples until x_{IN} is aligned with local PRN replica. Set $\upsilon = 0, k = 0, \hat{f}_{d0} = \hat{f}_{\text{dacq}}, \hat{\phi}_0 = 0, \psi_1 = 0, N_1 = \text{round}(T_{\text{int}} f_{\text{IN}})$.

1. Increase the integration period counter: $k = k + 1$.

2. Generate local code references: for $n = 1...N_k$, $s[n] = d_{\text{E1B/E1C}}\,(p[\text{round}(\delta_k \cdot n + \psi_k)])$, where

$$\delta_k = \frac{1}{T_c, E_1B.f_{IN}}(1 + \frac{\hat{f}_{k-1}}{f_c^{(GalE1)}})$$

and the Very Early, Early Late,and Very Late Versions with ϵ and ϵ'.

3. Generate local carrier: for $n = 1...N_k$,

$$c[n] = e^{-j(2\pi \hat{f}_{d_{k-1}} \frac{n}{\mathrm{f}_{IN}} + \mathrm{mod}(\hat{\phi}_{k-1}, 2\pi))}$$

4. Perform carrier wipe-off and compute the complex samples VE_k, E_k, P_k, P_k, and VL_k.

   Example:

$$P_k = \frac{1}{N_k} \sum_{n=0}^{N_k-1} x_{\mathrm{IN}}[n] s[n] c[n]$$

5. Compute PLL Discriminator: $\Delta\hat{\phi}_k = \mathrm{atan2}(\frac{P_{Q_k}}{P_{I_k}})$

6. Filter $\Delta\hat{\phi}_k$ with a bandwidth BWPLL:$h_{PLL}(\Delta\hat{\phi}_k)$

7. Update Carrier frequency estimation (in Hz):

$$\hat{f}_{d_k} = \hat{f}_{d_{acq}} + \frac{1}{2\phi T_{int}} h_{PLL}(\Delta\hat{\phi}_k)$$

   .

8. Update carrier phase estimation (in rad):

$$\hat{\phi}_k = \hat{\phi}_{k-1} + 2\pi \hat{f_{d_k}} T_{int} + h_{PLL}(\Delta\hat{\phi}).$$

9. Compute DLL discriminator: $\Delta\hat{\tau}_k = \frac{E_k - L_k}{E_k + L_k}$,Where :

$$E_K = \sqrt{VE_{I_K}^2 + VE_{Q_K}^2 + E_{I_K}^2 + E_{Q_K}^2},$$

$$L_K = \sqrt{VL_{I_K}^2 + VL_{Q_K}^2 + L_{I_K}^2 + L_{Q_K}^2},$$

10. Filter $\Delta\hat{\tau}_k$ with a bandwidth BWPLL:$h_{DLL}(\Delta\hat{\tau}_k)$

11. Update code phase estimation (in samples): N_{K+1} =round(S) and $\psi_{k+1} = S - N_{K+1}$,where

$$S = \frac{T_{int} f_{IN}}{(1 + \frac{\hat{f}d_k}{f_c^{(GalE1)}})} + \psi_k + h_{DLL}(\Delta\hat{\tau}_k) f_{IN}$$

12. Code lock indicator: $\hat{CN0} = 10.log_{10}(\hat{\rho}+10.log_{10}(\frac{f_{IN}}{2})-10.log_{10}(L_{PRN})$,Where :

$$\hat{\rho} = \frac{\hat{P}_s}{\hat{P}_n} = \frac{\hat{P}}{\hat{P}_{tot} - \hat{P}_s},$$

$$\hat{P}_s = (\frac{1}{u}\sum_{i=0}^{u-1}|P_{I_{k-i}}|)^2, \hat{P_{tot}} = (\frac{1}{u}\sum_{i=0}^{u-1}|P_{k-i}|)^2$$

13. Phase lock indicator :

$$T_{carrier} = \frac{(\sum_{i=0}^{u-1} P_{I_{k-i}})^2 - (\sum_{i=0}^{u-1} P_{Q_{k-i}})^2}{(\sum_{i=0}^{u-1} P_{I_{k-i}})^2 + (\sum_{i=0}^{u-1} P_{Q_{k-i}})^2}$$

14. **If** $T_{\text{carrier}} < \tau$ or $\text{CN0} < \text{CN0}_{\text{min}}$**then**:

    - Increase lock fail counter $\nu \leftarrow \nu + 1$.

15. **else**

    - Decrease lock fail counter $\nu \leftarrow \max(\nu - 1, 0)$.

16. **endif**

17. **If** $\nu > \vartheta$ **then**:

    - Notify the loss of lock to the control plane through the message queue.

18. **end if**

19. Output: P_k, accumulated carrier phase error $\hat{\phi}_k$ code phase $\hat{N} \leftarrow \hat{N} + N_k + \psi_k$, $C\hat{N}0$.

7.3. Demodulation

Demodulation is the process of extracting the original information or base-band signal from a modulated carrier signal. The purpose of demodulation is to retrieve the modulating signal, which could be analog or digital data, audio, video, or other forms of information. Demodulation is essential in various communication systems such as radio, television, cellular networks, and wireless data transmission.

After the aquisition and tracking has been performed, the received data is mapped back using BPSK demodulation, mapping -1 to binary 1 and $+1$ to binary 0.

7.4. Frame Synchronization

The pilot PRN of every satellite is locally generated to determine the starting point of the recieving frame. This is done by correlating the locally generated PRN with the recieved pilot PRN. The pilot PRN is correlated with the received pilot PRN from the latter's forst bit. The correlation output is stored. The pilot PRN then slides to the right by one bit so that it starts at the second bit of the received PRN. Correlation is carried out and the output is stored. This process is repeated for the entire length of the received frame. The index where the first occurence of maximum correlation takes place is taken as the start of the frame.

7.5. Decoding

Demodulated data is first separated into subframes. Subframes 2 and 3 are deinterleaved and decoded using belief propagation. The deinterleaving process involves reversing the interleaving algorithm used during transmission. By applying the inverse operation, the interleaved data are rearranged back into their original order. Then belief propagation algorithm is used to calculate the decoded sequence. After decoding, CRC is calculated to verify if there are any errors. Subframe 1 is decoded using Maximum-likelihood method.

7.5.1. Process

The high-level description of the channel decoding process in NavIC is shown in figure 7.4

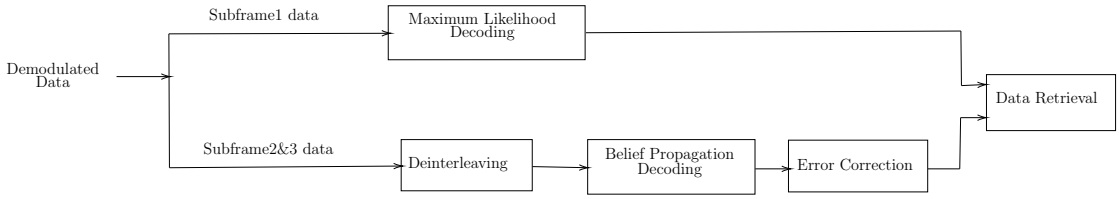


Figure 7.4: The Block Level Architecture for Channel decoding

7.5.2. Maximum Likelihood Decoding

The first 52 bits of the frame are a part of subframe 1 and are BCH encoded. To decode these bits, we use a modified version of ML decoding. Before encoding, subframe 1 has 9 bits. These 9 bits give us the TOI (Time of Interval) data. The TOI for NavIC L1 ranges from 1 (000000001) to 400 (110010000). After encoding, SF1 will have 52 symbols. So there will be 400 possible combinations of SF1 before and after encoding.

So, on the receiving side, we generate these 400 possible codewords and keep them in a lookup table. When a frame is received, the first 52 symbols are separated. The received codeword is compared with every other codeword in the loookup table and the hamming distance is calculated. The codeword in the lookup table which has the least hamming distance with the received codeword is the corrected codeword (NOTE: Works only if error ¡ 20 bits. This method fails for very noisy channels). The corresponding bit

sequence (decoded) can be mapped from the chosen codeword.

7.5.3. Deinterleaving

To reduce the burst error while transimission, the LDPC encoded symbols are interleaved into a 38x46 matrix. Data is written in columns and then, read in rows. So even if there is any burst error, it is spread out. At the receiving side, the matrix is then converted to a regular row of symbols.

7.5.4. Belief Propagation

The basic concept of the algorithm involves message passing. Messages are passed iteratively between adjacent nodes until convergence dictated maximal number of iterations. The nature of the messages involves probabilities or log-likelihoods of probabilities. As nodes receive messages from their nearest neighbors, they improve their estimate regarding received bits. For sufficiently long blocks, with sufficiently long cycles, the algorithm has been shown empirically to perform well.

Variable nodes pass messages that involve the log-likelihood of some bit being 0 or 1 by processing information obtained from other check nodes and the channel whereas check nodes pass messages which involve the log-likelihood of a parity equation is satisfied by processing information obtained from other variable nodes.

For each node of the factor graph, there are two directions of soft information in the form of LLR. The LLR propagated from right to left is

denoted as $L_{i,j}$, while that from left to right is denoted as $R_{i,j}$. The decoder employs the so-called processing element (PE), which performs the following LLR estimations with the Min-Sum approximation, so that LLRs are traversed through the factor graph.

BPD is performed on the factor graph by operating the PE from left to right, and from right to left to update L and R, respectively. The final decision is estimated after the completion of all the iterations. After decoding, CRC is calculated to verify the decoded data.

Chapter 8

Results

8.1. Acquisition

```
Acquisition results for PRN ID 2
 Status:True Doppler:680 Delay/Code-Delay:18655/4771.01625
Acquisition results for PRN ID 3
 Status:True Doppler:1480 Delay/Code-Delay:18942/4844.4165
Acquisition results for PRN ID 4
 Status:True Doppler:3960 Delay/Code-Delay:18843/4819.09725
Acquisition results for PRN ID 6
 Status:True Doppler:5000 Delay/Code-Delay:18886/4830.0945
```

8.2. Tracking and Decoding

Result

Tracking and decoding output for PRN ID:2

Subframe1 Before Encoding (9 bits): [1 0 0 0 1 1 1 0 1]

Subframe1 After Encoding (52 Symbols): [0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1
0 0 1 1 0 1 0 0 1 1 1 1 0 1 1 1 0 0 0 0
1 1 1 1 1 1 1 0 0 0 1 1 1 0 1]

Subframe1 After Receiving (52 Symbols): [1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 1.
1. 0. 1. 1. 0. 1. 1. 0. 0. 1. 0. 1.
1. 0. 0. 1. 1. 0. 1. 1. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 1.
1. 1. 0. 1.]

Subframe1 After decoding (9 bits): (1, 0, 0, 0, 1, 1, 1, 0, 1)

Subframe2 Before Encoding (600 bits): [1 0 1 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0
1 1 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 0 1 1 1 1 1
0 1 0 1 1 0 0 0 1 1 1 0 1 1 0 1 1 0 .....1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 1
1 1 1 0 1 1 1 0 1 0 1 0 1 0 0 1 1 0 0
1 0 1 1 0 1 0 1]

Subframe2 After Encoding (1200 Symbols): [1 1 1 ... 0 1 0]

Subframe2 After Receiving (1200 Symbols): [1. 1. 1. ... 0. 1. 0.]

Subframe2 After decoding (600 bits): [1 0 1 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 1
1 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 0 1 1 1 1 1
0 1 0 1 1 0 0 0 .....1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0

```
       1 0 1 0 0 1 1 0 0
 1 0 1 1 0 1 0 1]
Subframe 2 is same before encoding and after decoding: True
                0 <<< remainder
Subframe3 Before Encoding (274 bits): [1 1 1 1 1 0 0 1 0 1 0 1 0 0 0 0 1 0
    0 1 0 0 1 0 0 1 0 0 0 1 1 0 0 1 0 1 0
 1 1 1 1 1 1 0 0 0 0 0 1 1 0 1 1 0 1 1 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 1 0 0
..... 0 1 0 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 1 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1
     1
 1 0 1 1 1 0 1 0 0 1 0 1 1 1 1]
Subframe3 After Encoding (548 Symbols): [0 0 1 1 0 0 1 0 0 1 0 0 1 0 1 1
    0 1 0 1 1 0 0 0 1 1 1 1 0 0 1 0 0 0 1 1 1
 1 1 0 0 1 0 1 1 1 1 1 0 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1
 1 0 1 1 1 0 1 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 1 1 1 0 0 1 1 0 0 0 1
 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 1
```

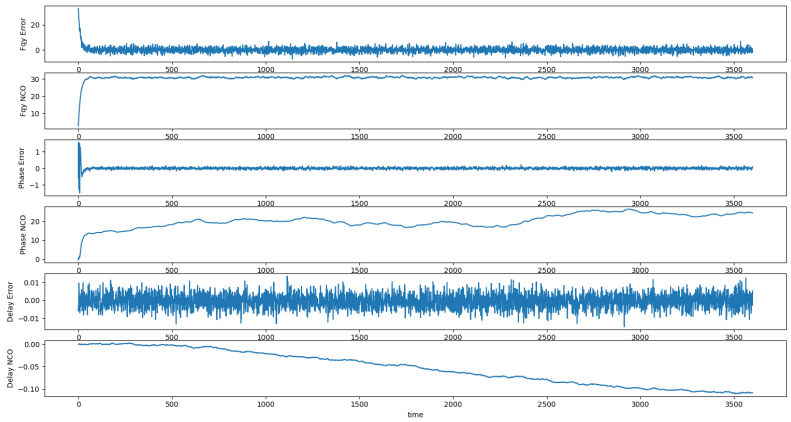


Figure 8.1: Tracking result plot

Appendix A

References

1. https://yair-mz.medium.com/decoding-ldpc-codes-with-belief-propagation-43c859f4276d

2. https://www.mdpi.com/2073-8994/14/12/2633

3. https://www.isro.gov.in/media_isro/pdf/Publications/Vispdf/Pdf2017/1a_messgingicd_receiver_incois_approved_ver_1.2.pdf

4. https://gnss-sdr.org/docs/sp-blocks/acquisition/

5. https://gnss-sdr.org/docs/sp-blocks/tracking/

6. Elliott D. Kaplan and Christopher J. Hegarty, Understanding GPS Principles and Applications, 3^{rd} edition