



UCGE Reports
Number 20184

Department of Geomatics Engineering

IF GPS Signal Simulator Development and Verification

(URL: <http://www.geomatics.ucalgary.ca/links/GradTheses.html>)

by

Lei Dong

December 2003



UNIVERSITY OF
CALGARY

THE UNIVERSITY OF CALGARY

IF GPS Signal Simulator Development and Verification

by

Lei Dong

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

November, 2003

© Lei Dong 2003

ABSTRACT

A software-based intermediate frequency (IF) GPS signal simulator for the L1 C/A code signal is developed and verified in this thesis. The signal simulator is implemented in MATLAB with some key functions coded in C to accelerate speed of operation. A mathematical signal model is developed that expresses the digitized IF GPS signal as a function of various errors during propagation, such as satellite clock error, ionosphere error, and troposphere error; and user defined parameters, such as date and GPS time, user trajectory, expected C/N_0 , noise density, intermediate frequency, sampling rate, front-end bandwidth, and quantization bit. The simulator is verified by comparing the simulated signal with the hardware-collected IF GPS signal in the time, frequency, correlation and position domains.

Partially for the simulator verification purpose, a software GPS receiver was developed and verified, which was used to process the simulated signal to get the pseudorange and carrier phase measurements. The signal acquisition and tracking techniques originally studied and verified in this thesis were used for the first generation software receiver developed by the PLAN group of the Department of Geomatics Engineering. The receiver consists of the following modules: signal acquisition, signal tracking, bit synchronization, sub-frame synchronization and the navigation solution. It is verified by using a set of real GPS signal measurements collected by a commercial hardware front-end and comparing the tracking results and navigation solution with the corresponding results of a hardware GPS receiver.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor, Professor Gérard Lachapelle, for his continuous guidance and support during my Master's program. I will always remember his sincere spirit of cooperation, his positive attitude, and understanding.

I would like to acknowledge the following graduate students and research associates: Olivier Julien, Aaron Morton, Changlin Ma, and Mark Petovello for their support and kindness.

Thanks also to the professors, students, and staff of the Department of Geomatics Engineering who have made my time fruitful and enjoyable.

My deepest thanks go to my dear husband, Changlin Ma, who supported me tremendously throughout my study, and to our son, Xingyu, for bringing so much joy into my life. I am also indebted to my parents and parents-in-law for their untiring support.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF TABLES	ix
LIST OF FIGURES	x
NOTATION	xvi
0.1 Symbols	xvi
0.2 Abbreviations and Acronyms	xix

CHAPTER 1

INTRODUCTION	1
1.1 Background	1
1.2 GPS Signal Simulation Review	3
1.3 Objective	6
1.4 Thesis Outline	7

CHAPTER 2

GPS SIGNAL AND RECEIVER THEORY	9
2.1 GPS Signals	9
2.1.1 GPS Signal Structure	9
2.1.2 GPS Signal Power Levels	12
2.1.3 L1 C/A Code Signal in the Frequency Domain	14
2.1.4 C/A Code Auto-correlation and Cross-correlation	16
2.2 Conventional Hardware Front-end and GPS Receiver	17
2.2.1 Front-end.....	19

2.2.2 Signal Processing	21
2.3 Error Sources	25
2.3.1 Ephemeris Errors	26
2.3.2 Satellite Clock Error	26
2.3.3 Ionospheric Error and Tropospheric Error.....	27
2.3.4 Multipath.....	28
2.3.5 Receiver Noise Errors.....	29

CHAPTER 3

IF GPS SIGNAL SIMULATOR REALIZATION	30
3.1 IF GPS Signal Simulation Model Derivation	30
3.1.1 Transmitted GPS Signal.....	31
3.1.2 Propagation through the Atmosphere to the Antenna.....	34
3.1.3 Down-conversion.....	37
3.1.4 Intermediate Frequency Processing	39
3.2 Simulator Realization	39
3.2.1 Satellites in View	41
3.2.2 Amplitude Modeling.....	43
3.2.3 Navigation Data Simulation.....	44
3.2.4 C/A Code Modeling.....	46
3.2.5 Signal Propagation Time and Doppler.....	47
3.2.6 Satellite Clock Error Modeling.....	51
3.2.7 Ionospheric Error Modeling.....	52
3.2.8 Tropospheric Error Modeling	59
3.2.9 Noise	63
3.2.10 Filtering.....	65
3.2.11 Quantization.....	67
3.2.12 Program Flow Chart.....	71

CHAPTER 4

SOFTWARE GPS RECEIVER	76
4.1 Introduction.....	76
4.2 Signal Acquisition	77
4.2.1 DFT-based Acquisition.....	79
4.3 Signal Tracking	82
4.3.1 Basic Phase Lock Loop.....	83
4.3.2 Delay Lock Loop	86
4.3.3 Carrier Tracking Loop	90
4.4 DLL Lock Detectors and C/N_0 Estimation	94
4.5 Bit Synchronization	96
4.6 Sub-frame Synchronization	98
4.7 Raw Measurement Derivation	102
4.7.1 Pseudorange	102
4.7.2 Carrier Phase and Doppler	104
4.8 Navigation Solution.....	104

CHAPTER 5

VERIFICATION OF THE SOFTWARE GPS RECEIVER BY A REAL IF GPS SIGNAL	105
5.1 Software Receiver Verification Scheme.....	105
5.2 Acquisition Results Verification	108
5.3 Tracking Results of the Software GPS Receiver and Navigation Solution Verification	108

CHAPTER 6

SIMULATED IF GPS SIGNAL VERIFICATION RESULTS	115
6.1 Simulator Verification Scheme.....	115

6.2 Simulated Signal Description.....	116
6.3 Time Domain Verification.....	117
6.4 Verification in the Correlation Domain.....	118
6.5 Tracking Result Comparison.....	122
6.5.1 Data Bits	122
6.5.2 Carrier-to-noise Ratio	125
6.5.3 Delay Lock Loop Detector	128
6.5.4 Doppler	131
6.5.5 Pseudoranges	134
6.6 Verification in Position Domain	139
6.7 Some Other Results for Various Simulation Cases	141
CHAPTER 7	
CONCLUSIONS AND FUTURE WORK	151
7.1 Conclusions.....	151
7.1.1 Software IF GPS Signal Simulator	151
7.1.2 Software GPS Receiver	152
7.2 Future Work.....	153
7.2.1 Future Work for the Software IF GPS Signal Simulator	153
7.2.2 Future Work for the Software GPS Receiver	154
REFERENCES.....	156

LIST OF TABLES

Table 2.1: Received GPS signal power levels	12
Table 2.2: GPS ranging errors	26
Table 3.1: Simulator parameters	75
Table 4.1: PLL discriminators	93
Table 5.1: Doppler and C/N_0 of the observed satellites.....	108
Table 5.2: Acquired Doppler values of the Signal Tap collected signal	108
Table 5.3: Doppler and C/N_0 values obtained by the OEM4 receiver and software GPS receiver.....	112
Table 6.1: Signal simulation parameters.....	116
Table 6.2: Simulated C/N_0 and ionospheric and tropospheric errors.....	117
Table 6.3: Correlation peak values of all acquired satellites	121
Table 6.4 Pseudorange differences of the simulated signals and simulated ionospheric errors	147

LIST OF FIGURES

Figure 2.1: Relationship between GPS signal components	12
Figure 2.2: GPS signal description in the frequency domain	15
Figure 2.3: Autocorrelation and cross-correlation of GPS gold codes	17
Figure 2.4: Generic GPS receiver functional block diagram	18
Figure 3.1: GPS signal propagation and processing in the front-end	31
Figure 3.2: Digitized IF GPS signal simulator structure.....	40
Figure 3.3: Simulated receiver trajectory modes	41
Figure 3.4: GPS navigation data format	44
Figure 3.5: TLM and HOW word.....	46
Figure 3.6: C/A code generator.....	47
Figure 3.7: Effect of Earth rotation on signal propagation	48
Figure 3.8: Satellite and receiver position in the ECEF frame at the transmission time and reception time.....	50
Figure 3.9: Description of the single-layer ionosphere model.....	54
Figure 3.10: Compute TEC in a grid network	58
Figure 3.11: Noise characteristics in the time and frequency domains	64
Figure 3.12: Band-pass filter magnitude and phase response.....	65
Figure 3.13: IF GPS signal before and after filtering	66
Figure 3.14: IF 1-bit quantization GPS signal in the time and frequency domains.....	67
Figure 3.15: Two-bit quantization strategy.....	68
Figure 3.16: A/D conversion gain in Gaussian noise	70

Figure 3.17: IF 2-bit quantization GPS signal in the time and frequency domains.....	71
Figure 3.18: Signal simulator functions.....	73
Figure 3.19: Program flow chart.....	74
Figure 4.1: Structure of the software GPS receiver	77
Figure 4.2: Signal acquisition	78
Figure 4.3: DFT-Based acquisition.....	80
Figure 4.4: Coupled code and carrier tracking loops.....	83
Figure 4.5: Basic phase lock loop.....	84
Figure 4.6: Delay lock loop	87
Figure 4.7: Code mismatch vs early, punctual and late components.....	88
Figure 4.8: DLL discriminators	88
Figure 4.9: Frequency lock loop.....	91
Figure 4.10: Phase lock loop.....	92
Figure 4.11: Statistics of normalized power NP for M=20.....	95
Figure 4.12: A successful bit synchronization histogram.....	96
Figure 4.13: Navigation message.....	99
Figure 4.14: TLM word and HOW word.....	100
Figure 4.15: Pseudorange construction.....	103
Figure 5.1: Hardware front-end “GPS Signal Tap”	106
Figure 5.2: OEM4 receiver tracking status	107
Figure 5.3 (a): Average C/N_0 and Doppler for SV1	109
Figure 5.3 (b): Average C/N_0 and Doppler for SV2	109
Figure 5.3 (c): Average C/N_0 and Doppler for SV4	110

Figure 5.3 (d): Average C/N_0 and Doppler for SV13	110
Figure 5.3 (e): Average C/N_0 and Doppler for SV16	110
Figure 5.3 (f): Average C/N_0 and Doppler for SV20.....	111
Figure 5.3 (g): Average C/N_0 and Doppler for SV25	111
Figure 5.4: Navigation solution of the real GPS signal	113
Figure 5.5: Navigation solution computed from the OEM4 receiver generated raw measurements.....	114
Figure 6.1: Power spectrum of the real and simulated IF GPS signal	117
Figure 6.2 (a): Acquisition results of the real signal and simulated signal for SV1	119
Figure 6.2 (b): Acquisition results of the real signal and simulated signal for SV2.....	119
Figure 6.2 (c): Acquisition results of the real signal and simulated signal for SV4.....	119
Figure 6.2 (d): Acquisition results of the real signal and simulated signal for SV13.....	120
Figure 6.2 (e): Acquisition results of the real signal and simulated signal for SV16.....	120
Figure 6.2 (f): Acquisition results of the real signal and simulated signal for SV20	120
Figure 6.2 (g): Acquisition results of the real signal and simulated signal for SV25.....	121
Figure 6.3 (a): Data bits of the real signal and simulated signal for SV1	122
Figure 6.3 (b): Data bits of the real signal and simulated signal for SV2.....	123
Figure 6.3 (c): Data bits of the real signal and simulated signal for SV4.....	123
Figure 6.3 (d): Data bits of the real signal and simulated signal for SV13.....	123
Figure 6.3 (e): Data bits of the real signal and simulated signal for SV16.....	124
Figure 6.3 (f): Data bits of the real signal and simulated signal for SV20	124
Figure 6.3 (g): Data bits of the real signal and simulated signal for SV25.....	124
Figure 6.4 (a): C/N_0 of the real signal and simulated signal for SV1	125

Figure 6.4 (b): C/N_0 of the real signal and simulated signal for SV2.....	126
Figure 6.4 (c): C/N_0 of the real signal and simulated signal for SV4.....	126
Figure 6.4 (d): C/N_0 of the real signal and simulated signal for SV13.....	126
Figure 6.4 (e): C/N_0 of the real signal and simulated signal for SV16.....	127
Figure 6.4 (f): C/N_0 of the real signal and simulated signal for SV20.....	127
Figure 6.4 (g): C/N_0 of the real signal and simulated signal for SV25.....	127
Figure 6.5 (a): DLL lock detector of the real signal and simulated signal for SV1.....	129
Figure 6.5 (b): DLL lock detector of the real signal and simulated signal for SV2.....	129
Figure 6.5 (c): DLL lock detector of the real signal and simulated signal for SV4.....	129
Figure 6.5 (d): DLL lock detector of the real signal and simulated signal for SV13.....	130
Figure 6.5 (e): DLL lock detector of the real signal and simulated signal for SV16.....	130
Figure 6.5 (f): DLL lock detector of the real signal and simulated signal for SV20.....	130
Figure 6.5 (g): DLL lock detector of the real signal and simulated signal for SV25.....	131
Figure 6.6 (a): Doppler of the real signal and simulated signal for SV1.....	132
Figure 6.6 (b): Doppler of the real signal and simulated signal for SV2.....	132
Figure 6.6 (c): Doppler of the real signal and simulated signal for SV4.....	132
Figure 6.6 (d): Doppler of the real signal and simulated signal for SV13.....	133
Figure 6.6 (e): Doppler of the real signal and simulated signal for SV16.....	133
Figure 6.6 (f): Doppler of the real signal and simulated signal for SV20.....	133
Figure 6.6 (g): Doppler of the real signal and simulated signal for SV25.....	134
Figure 6.7 (a): Doppler of the real signal and simulated signal for SV1.....	135
Figure 6.7 (b): Doppler of the real signal and simulated signal for SV2.....	135
Figure 6.7 (c): Doppler of the real signal and simulated signal for SV4.....	135

Figure 6.7 (d): Doppler of the real signal and simulated signal for SV13	136
Figure 6.7 (e): Doppler of the real signal and simulated signal for SV16	136
Figure 6.7 (f): Doppler of the real signal and simulated signal for SV20	136
Figure 6.7 (g): Doppler of the real signal and simulated signal for SV25	137
Figure 6.8 Pseudorange differences of the real signal and simulated signal	138
Figure 6.9: Navigation solution of the real GPS signal	139
Figure 6.10: Navigation solution of the simulated GPS signal.....	139
Figure 6.11: Carrier smoothed navigation solution of the simulated GPS signal.....	141
Figure 6.12: Navigation solution of the simulated GPS signal with 1-bit quantization .	143
Figure 6.13: Pseudorange and Doppler differences of the simulated GPS signal with 1-bit quantization.....	143
Figure 6.14: Navigation solution of the simulated GPS signal with noise, filtering and 1- bit quantization.....	144
Figure 6.15: Pseudorange and Doppler differences of the simulated GPS signal with noise, filtering and 1-bit quantization	144
Figure 6.16: Navigation solution of the simulated GPS signal with noise, filtering and 2- bit quantization.....	145
Figure 6.17: Pseudorange and Doppler differences of the simulated GPS signal with noise, filtering and 2-bit quantization	145
Figure 6.18: Navigation solution of the simulated GPS signal with noise, filtering, 1-bit quantization, and ionospheric error.....	146
Figure 6.19: Pseudorange and Doppler differences of the simulated GPS signal with noise, filtering, 1-bit quantization, and ionospheric error.....	146

Figure 6.20: Navigation solution of the simulated GPS signal with tropospheric error model applied in C ³ NAVIG ² ™	148
Figure 6.21: Navigation solution of the simulated GPS signal (receiver trajectory mode 1)	149
Figure 6.22: Pseudorange and Doppler differences of the simulated GPS signal (receiver trajectory mode 1)	149
Figure 6.23: Navigation solution of the simulated GPS signal (receiver trajectory mode 2)	150
Figure 6.24: Pseudorange and Doppler differences of the simulated GPS signal (receiver trajectory mode 2)	150

NOTATION

0.1 Symbols

A	Signal Amplitude
\sqrt{A}	Square Root of the Semi-major Axis
B_n	Bandwidth of the Filter in the Receiver to Remove the Out of Band Noise
C	Pseudorandom Noise Spreading Sequence (C/A code)
c	Speed of Light
D	Navigation Data Bits
e	Partial Pressure of the Water Vapour
E_k	Eccentric Anomaly of Satellite Orbit
f_d	Doppler Frequency
f_{L1}	Centre Frequency of L1
f_{L2}	Centre Frequency of L2
f_{data}	Frequency of Navigation Data
F_e	$-4.442807633 \times 10^{-10} s / m^{1/2}$
G_d	Spectral Density of the Navigation Data
$G_{C/A}$	Spectral Density of the C/A Code
$m(z)$	Single-layer Mapping Function

N	The Number of the Satellites in View
N_0	Noise Density
N_{d0}	Dry Reflectivity at the Surface of the Earth
N_{w0}	Wet Reflectivity at the Surface of the Earth
P_c	Effective Carrier Power
P_d	Dry Air Pressure
P_t	Transmitted Signal Power
P_r	Received Signal Power
\tilde{P}_{nm}	Normalized Associated Legendre Functions of Degree n and Order m
R	Auto Correlation Function
R_h	Relative Humidity
RH_0	Average Relative Humidity
RH_v	Daily Change of the Relative Humidity
s	Sun-fixed Longitude of the Ionospheric Pierce Point
S_{L1}	GPS L1 Signal
SF	Scale Factor
t	GPS Time
t_{oc}	Reference Time for Clock Correction
$t_{T \max}$	The Epoch with the Highest Temperature
$t_{RH \max}$	The Epoch with the Maximum Relative Humidity
Δt_R	Relativistic Correction

T_0	Average Temperature over a Given Day
T_c	Temperature
T_p	Signal Propagation Time from Satellite to Receiver Antenna
T_v	Daily Change of the Relative Humidity
\bar{v}	Velocity of the Satellite
\bar{v}_u	Velocity of the User
Y	Encrypted P Code Spreading Sequence
z	Geocentric Zenith Distance of a Satellite at the Height of GPS Receiver
z'	Geocentric Zenith Distance of the Single Layer of the Ionosphere
$\bar{\mathbf{l}}$	Line of Sight Unit Vector from User to Satellite
ω_e	Frequency Error
ω_{IF}	Intermediate Frequency
ω_{L1}	L1 Carrier Frequency
ω_{Lo}	Frequency of Local Oscillator
$\Delta\omega$	Frequency Offset
ϕ_0	Nominal Carrier Phase
φ_0	Initial Phase of the GPS Signal
φ_e	Fixed Phase Error Offset
δt_{iono}	Ionospheric Error
δt_r	Receiver Clock Error
δt_{sv}	Satellite Clock Error

δt_{tropo}	Tropospheric Error
δt_{MP}	Multipath Signal Path Delay
α	Signal Power Attenuation Factor
β	Geocentric Latitude of the Ionospheric Pierce Point
θ	Elevation Angle
Λ_{nm}	Normalization Factor
Δ_s	Troposphere Slant Delay

0.2 Abbreviations and Acronyms

A/D	Analog to Digital
AGC	Automatic Gain Control
BPSK	Binary Phase Shift Keying
C/A	Coarse/Acquisition
CDDIS	Crustal Dynamics Data Information System Institute
C/N ₀	Carrier to Noise Density Ratio
C ³ NAVIG ² ™	Combined Code and Carrier for Navigation with GPS and GLONASS
DLL	Delay Lock Loop
ECEF	Earth-Centred, Earth-Fixed
ECI	Earth-Centred Inertial
FIR	Finite Impulse Response
FLL	Frequency Lock Loop
GPS	Global Positioning System

GLONASS	Global Orbiting Navigation Satellite System
I	In-Phase Carrier Signal
IF	Intermediate Frequency
IGS	International GPS Service
L1	Link 1
L2	Link 2
LNA	Low Noise Amplifier
LO	Local Oscillator
NCO	Numerically Controlled Oscillator
PLL	Phase Lock Loop
PPS	Precise Positioning Service
PRN	Pseudorandom Noise
P(Y)	Precise
Q	Quadra-Phase Carrier Signal
RMS	Root Mean Square
RF	Radio Frequency
SA	Selective Availability
SNR	Signal to Noise Ratio
SPS	Standard Positioning Service
Std Dev	Standard Deviation
TEC	Total Electron Content
UTC	Coordinated Universal Time
WAAS	Wide Area Augmentation System

WGS-84 World Geodetic System - 1984

CHAPTER 1

INTRODUCTION

1.1 Background

For positioning purposes, a GPS receiver needs to fulfill several tasks to derive the raw measurements from the GPS radio frequency (RF) signals transmitted by the satellites. First, a well-designed antenna is necessary to receive the RF signals which are down-converted to a low frequency band and digitized into intermediate frequency (IF) discrete signals by an RF section, referred to as the front-end. At this point, the digitized IF signal is ready to be processed by the receiver channels. Then, the signals are fed into different channels allocated to each satellite. Each receiver channel conducts operations of acquisition, tracking, and navigation message demodulation. Acquisition is used to detect a signal and derive its initial code shift and Doppler estimates. Tracking, bit synchronization, and sub-frame synchronization are used to keep track of the carrier and code as well as to obtain pseudorange, carrier phase measurements, and the navigation message. Finally, the navigation solution is computed. In a conventional receiver, the front-end, and part of the digital signal processing (such as correlation and accumulation), are implemented in hardware while the other part of the digital signal processing is implemented in software.

Unlike conventional GPS receivers, a software GPS receiver is mainly realized by software except for the front-end part, which offers various advantages. First, a software

approach removes the nonlinear, temperature-dependent and age-dependent components of the hardware receiver. Second, a software receiver can provide researchers and developers with more evaluation and testing flexibility. It is easy to prototype the latest theoretical signal processing developments. The performance of different algorithms can be compared without any hardware development. Third, it provides an effective simulation environment. The processing techniques of new signals, such as L5 signals in the GPS modernization program and Galileo signals, can be studied without any hardware development.

All research on software GPS receivers start from the basis of the digitized IF signal. Only a digitized IF signal with proper base band intermediate frequencies can be fed to and processed by a software GPS receiver. Various hardware front-ends and A/D (Analog-to-Digital) converters can be used to implement frequency down-conversion, signal sampling, and quantization to provide a suitable IF GPS signal; however, using a hardware front-end has several shortcomings. Firstly, a hardware front-end affords little flexibility because it normally has its own specific and multiple, generally component-driven, parameters that cannot be easily changed. Secondly, the use of a hardware front-end requires a GPS simulator to generate actual radio frequency signals based on predetermined parameters to simulate various GPS environment conditions. However, in the typical development stage of a project, the cost of a GPS simulator is prohibitively expensive, typically in the hundreds of thousands of dollars. Furthermore, some new GPS signals may not be available even in the latest hardware simulators.

A software simulator is an efficient way to generate the digitized IF GPS signal as the input to a software receiver. First, a software simulator allows simulation of various signal conditions. The user can configure the GPS signal propagation and processing chain, allowing the introduction of specialized situations to create digitized IF GPS signals with specific properties as required. This makes it convenient to test and evaluate various receiver acquisition and tracking algorithms and to investigate the software receiver performance. Second, it is possible to add new signal features in a software simulator; for example, operation with signals that are not yet available (such as L5 and Galileo signals) can be simulated to develop and evaluate new receiver algorithms. The flexibility and lower cost of software-based simulation provides a viable method of generating the expected signal.

1.2 GPS Signal Simulation Review

Several software-based GPS signal simulators are currently under development, as documented in the literature. Although the specific implementation varies between the available simulators, they can simulate either digitized IF signals or analytical accumulated In-phase (I) and Quadra-phase (Q) components. In some simulators, the real GPS signals - including codes, navigation messages, and error sources - are faithfully simulated while, in others, mathematical models are used to directly generate analytical I and Q components with error sources without simulating the actual carrier and code waveforms. Although the effects of errors are important, not all error sources may be simulated in any one of the simulators due to their different research purposes.

Three commercial simulators were found in the literature. The MATLAB GPS signal simulation toolbox developed by the NAVSYS Company (Brown et al., 2000) provides a complete set of GPS signal simulations, test, and analysis tools. It can generate the simulated RF GPS signals as a function of the signal strength, pseudorange, Doppler and Doppler rate. The effects of interfering signal sources, antenna and front-end on the received GPS signal are also modeled. This simulation approach is very computationally expensive because of the simulation of RF GPS signals.

The second product found is the software-based GPS signal simulator released by the Centre for Remote Sensing (CRS) (CRS, 2002). It is part of a suite of products developed by CRS for the development of advanced GPS systems. It allows users to introduce various specialized situations such as scintillation, interference and jamming, namely narrow-band, wide-band and directional, high-dynamics; specified orbits, wave front simulation at multiple antenna locations, plasma re-entry bodies, obscuration, multipath effects, and ocean scattering.

The third product, available commercially through Navtech Inc., is the GPS correlator simulator developed by Accord Software and Systems. According to Navtech, Accord's product is a "very powerful software package for understanding the characteristics of the GPS satellite signal and its processing in a typical GPS receiver" (Navtech, 2002). The simulator is composed of two parts: an input signal simulator and a signal processor. The input signal simulator can simulate noise effects, receiver circuit parameters and the base band converter. However, its true environment is not driven by satellite position and

receiver trajectory, which is a requirement for simulating ultra-tightly coupled navigation systems.

Burns et al. (2002) presented a software GPS signal simulation method for the purpose of investigating GPS software radio performance in combating narrow band interference. An L1 signal was simulated including the C/A code, the navigation data, atmospheric noise, and radio frequency interference (RFI); the performance of the antenna, filtering, gain, down-conversion and thermal noise inherent in the analog hardware were taken into account. Seitz and Braasch (1998) implemented a high fidelity GPS receiver simulation that is structured to include a GPS signal generator, a propagation path that allows for the addition of signal anomalies, and a GPS receiver model. The functions of the first two items were similar to the simulation made by Burns et al. (2002).

Corbell (2000) developed a GPS signal simulator based on the true satellite and user trajectories. Only the receiver errors that affect the receiver tracking loop performance were taken into account. The errors, such as clock error, amplitude of signal and noise, and quantization effects were modeled at the A/D output point by derivation of analytical error expressions.

A distinctive analytical representation of a GPS receiver was also realized by Miller and Corbell (2000) and Corbell and Miller (2001). The GPS signal simulator part does not actually generate the carrier or code waveforms, but keeps track of their phases and relationships to generate the analytical I and Q samples by means of mathematical

models. The computational burden is greatly reduced using this simulation method, but it is less accurate at high dynamics and over long periods.

1.3 Objective

The objective of this thesis is to develop and verify a digitized IF GPS signal simulator. This objective is part of a larger objective within the PLAN group of the Department of Geomatics Engineering to develop a series of generic GNSS software receiver modules to study existing and new GPS and Galileo signal performance, as well as study the effect of various types of interference and external aiding on system performance.

The main purpose of the signal simulator developed herein is to provide a basic user-friendly simulation tool to generate the digitized IF GPS signal for evaluation of software receiver performance. The research is thus complementary to advanced research into various acquisition, tracking and navigation algorithms for software GPS receivers underway by other researchers within the PLAN group. Because of time constraints, a basic version of the simulator, based on the L1 C/A code, is developed and verified in this thesis.

The simulation is based on receiving time, satellites' trajectories, user trajectory, and a selection of GPS errors. The generated signal is saved in a data file that can be directly read by a software receiver. The simulator endeavors to create an accurate representation of the signals as a function of propagation channel parameters, front-end parameters, and environmental noise. However, only those error sources that significantly affect the

measurements will be simulated in the first stage. To reduce the potentially large computational burden, the error simulation is based only on the mathematical models at the IF sample point with some key functions written in C code. Another consideration is that the simulator must be easily reconfigured so that users can combine new components or functions to generate the new signals and change parameters for specific signal conditions.

Verification of the simulator's operation is critical to the success of the project. It can be conducted directly by feeding the simulator-generated IF GPS signal to a software GPS receiver and observing if the correct tracking results and navigation solution are obtained. To this end, the first step involves the development of a software GPS receiver. However, the self-developed receiver also needs verification. Therefore, real IF GPS signals collected by a hardware front-end at a known position is fed to the receiver to verify the correctness of the software GPS receiver. The simulator simulates this group of real IF GPS signals, and then feeds the results to the same software receiver. The verification is conducted by comparing the simulated and real signals in the frequency, acquisition and position domains and the tracking results of both signals.

1.4 Thesis Outline

The remaining parts of this thesis consist of the following chapters: Chapter 2 describes the structure of the GPS signal, conventional front-end and receiver design, as well as various error sources. Chapter 3 first develops an intermediate frequency GPS signal simulation model, which includes the RF GPS signal propagation in the atmosphere and

processing in the front-end till the IF signal point after A/D; then the realization of the digitized IF GPS simulator is described. Chapter 4 describes the design of a software GPS receiver. Chapter 5 presents the software GPS receiver verification results using the real IF GPS data collected by a hardware front-end called “Signal Tap”. Chapter 6 presents the simulator verification by comparing the simulated and real signals in the frequency, acquisition and position domains and the tracking results of both signals. Chapter 7 describes the conclusions and recommendations for further research.

CHAPTER 2

GPS SIGNAL AND RECEIVER THEORY

2.1 GPS Signals

Currently, each GPS satellite broadcasts signals on two specific centre frequencies, L1 (1575.42 MHz) and L2 (1227.60 MHz). The transmitted signal is a modulation of the navigation message on Pseudorandom Noise (PRN) codes and carriers. The Standard Positioning Service (SPS) signal (C/A code) and the Precise Positioning Service (PPS) signal (P(Y) code) are both broadcast on the L1 frequency. The C/A code is the in-phase component, and the P(Y) code is quadrature component. Each satellite has the capability of broadcasting either the SPS or the PPS signal on L2 but, in practice, only the PPS signal is broadcast. According to the GPS Modernization Program, new signals such as L5 and the M code will be implemented in the satellites of Block IIF (McDonald and Hegarty, 2000).

2.1.1 GPS Signal Structure

Each signal consists of three components:

- Carrier: RF sinusoidal signal with frequency f_{L1} or f_{L2} .
- Ranging code: The sequences, called PRN sequences or PRN codes have special properties which allow all satellites to transmit at the same frequency without interfering with each other. SPS uses a family of Gold codes for its PRN

sequences (C/A codes), at a chipping rate of 1.023 MHz. The PPS uses a unique segment of an extremely long ($\sim 10^{14}$ chips) PRN sequence called the P(Y) code, which has a chipping rate of 10.23 MHz, ten times that for a C/A code. The Gold codes are 1023 chips long, and therefore repeat every millisecond. The P(Y) code segment repeats each week, giving it a period of 604800 seconds and a unique sequence of 6.1871×10^{12} chips.

- Navigation data: this is a binary coded message consisting of data on the satellite health status, ephemeris, clock bias parameters, and an almanac giving reduced precision ephemeris data on all satellites in the constellation. Each GPS satellite's BPSK (Binary Phase Shift Keying) unit modulates all signals with a 50 Hz data message 37500 bits long, which continuously repeats and periodically changes. An entire data message from one satellite is received only after 12.5 minutes of continuous reception.

These three components of a signal are derived coherently from one of the atomic standards aboard the satellite, which is 10.23 MHz. The relationship with the chipping rate is obvious. The relationships with L1, L2 and the navigation data rate are as follows:

$$\begin{aligned}
 f_{L1} &= 1575.42 \text{ MHz} = 154 \times 10.23 \text{ MHz} \\
 f_{L2} &= 1227.60 \text{ MHz} = 120 \times 10.23 \text{ MHz} \\
 f_{data} &= 50 \text{ Hz} = 1/204600 \times 10.23 \text{ MHz}
 \end{aligned}
 \tag{2.1}$$

Each code is combined with the binary navigation data using modulo-2 addition: if the code chip and the data bit are the same (both are 0s or both are 1s), the result is 0; and if

they are different, the result is 1. The composite binary signal is then modulated on the carrier by BPSK modulation.

The L1 signal leaving the antenna of the i^{th} satellite can be represented analytically as:

$$S_{L1i} = \sqrt{2P_t} D_i(t) C_i(t) \cos(\omega_{L1}t) + \sqrt{2P_p} D_i(t) P_i(t) \sin(\omega_{L1}t) \quad (2.2)$$

where

- S_{L1i} is the GPS L1 signal transmitted from SV_i
- P_t is the transmitted C/A code signal power
- P_p is the transmitted P(Y) code signal power
- $D_i(t)$ is the navigation data bits
- $C_i(t)$ is the pseudorandom noise spreading sequence (C/A code)
- ω_{L1} is the L1 carrier frequency
- t is the GPS time
- $P_i(t)$ is the P code or encrypted P code (Y code) spreading sequence

In order to express the BPSK signals in Equation (2.2) the binary values of the navigation data need to be switched to ± 1 . A bit 0 maps into 1 and a bit 1 maps into -1. Figure 2.1 illustrates the relationships of the C/A code signal components specified in Equation (2.2). This plot represents an idealized depiction of the signals for the purpose of illustrating this relationship. In practice, phase transitions at these frequencies are never as instantaneous as depicted in this figure.

2.1.2 GPS Signal Power Levels

GPS signals received on the Earth are extremely weak. They are well below the background RF noise level sensed by an antenna. It is the knowledge of the signal structure (i.e. PRN code) that allows a receiver to extract the signal buried in the background noise and to make precise measurements. The specifications of the received power level for users on the Earth are given in Table 2.1.

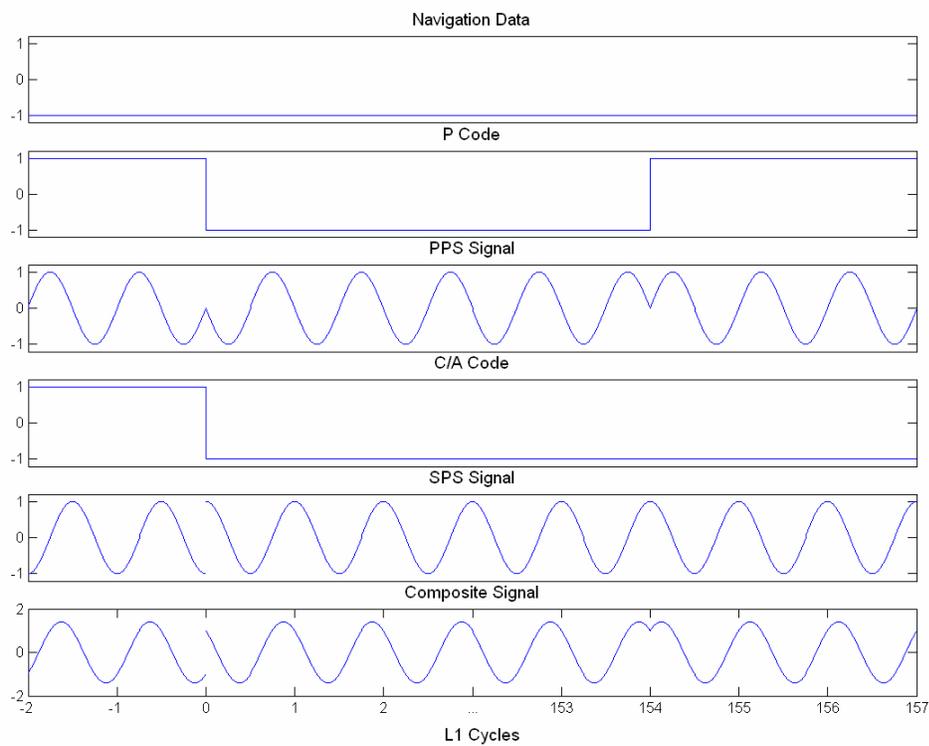


Figure 2.1: Relationship between GPS signal components

Table 2.1: Received GPS signal power levels

	L1 C/A Code	L1 P Code	L2 P Code
Minimum received power	-160.0 dBW	-163.0 dBW	-166.0 dBW

Maximum expected power	-152.0 dBW	-155.5 dBW	-158.0 dBW
------------------------	------------	------------	------------

These measurements are referenced to a power level of one Watt. However, absolute signal power is not necessary meaningful, because it does not consider the power level of the background noise. Receiver performance is more dependent on a signal-to-noise ratio (SNR) than the absolute signal power. SNR is a ratio of signal power to noise power as shown in Equation (2.3). The noise power depends on the processing bandwidth of the GPS receiver.

$$SNR = \frac{S}{N_0 B_n} \quad (2.3)$$

where

B_n is the band width of the filter in the receiver to remove the out of band noise

S is the signal power in bandwidth B_n

N is the noise power in bandwidth B_n

N_0 is the noise density

In practice, the ratio of total carrier power to the noise density C/N_0 in dB-Hz is the most generic representation of signal power as it is independent of the implementation of the receiver front-end bandwidth. The relationship of SNR and C/N_0 can be represented as:

$$SNR (dB) = \frac{C}{N_0} (dB - Hz) - B_n (dB) \quad (2.4)$$

where

C is the total received signal power

It can be seen from the above equations that the signal power and noise power can be calculated if the noise density, receiver processing bandwidth, and C/N_0 are specified.

2.1.3 L1 C/A Code Signal in the Frequency Domain

The GPS signal can be readily described in the frequency domain. As soon as the GPS signal is generated, it travels through a propagation channel which contains noise and interference. The received GPS signal is buried in the environmental noise; nothing else but noise behaviour can be observed in the time domain for the signal before acquisition and tracking. A description of the GPS signal in the frequency domain is definitely a powerful tool for identifying signal properties. Since the simulator developed in this thesis is for the L1 C/A code, the P code and L2 will not be discussed any further.

The GPS signal is generated through spectrum spreading, which is a method of taking the data signal of bandwidth B_d that is modulated on a sinusoidal carrier to form $d(t)$, and then spreading its bandwidth to a much larger value B_s . Bandwidth spreading can be accomplished by multiplying the data modulated carrier by a wide bandwidth spreading code. The GPS signal spectrum spreading process is shown in Figure 2.2. Since the purpose of this figure is principally to show the GPS signal power spectra density properties in the frequency domain, no scale is given on the y axis.

The GPS navigation data bit stream with values ± 1 , and clock rate f_d (50 Hz) is first modulated on a carrier of power P_d to form a narrow bandwidth signal as shown in Figure 2.2 (a). This narrow bandwidth signal of bandwidth B_d is then spread in bandwidth by a binary PRN sequence C/A code which has a clock rate f_c (1.023 MHz) that greatly exceeds the data rate. The navigation data and C/A code have the following power spectral densities, respectively:

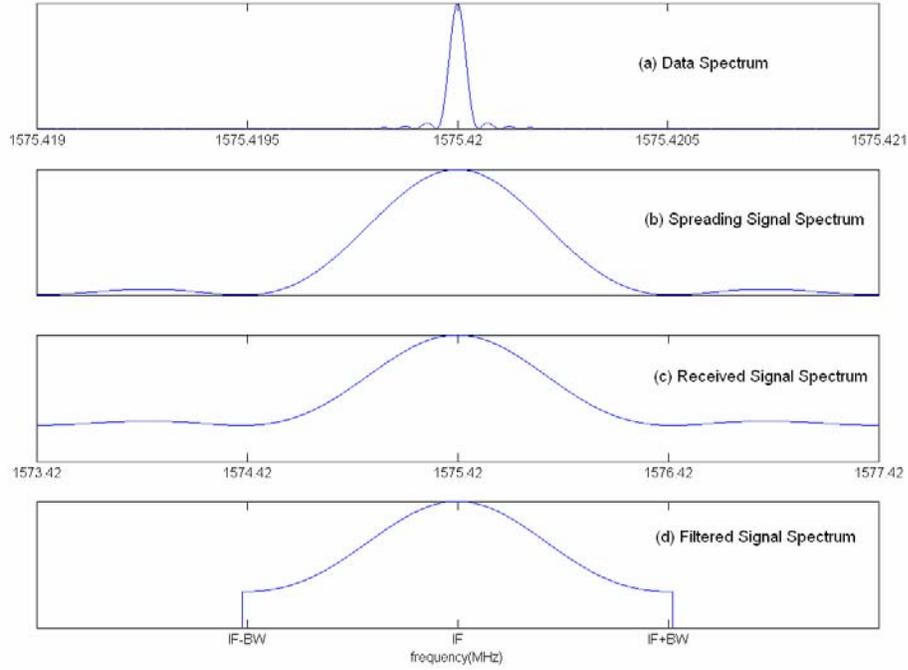


Figure 2.2: GPS signal description in the frequency domain

$$G_d(f) = \frac{1}{f_d} \left[\frac{\sin(\pi f / f_d)}{\pi f / f_d} \right]^2 \quad (2.5)$$

$$G_{C/A}(f) = \frac{1}{f_c} \left[\frac{\sin(\pi f / f_c)}{\pi f / f_c} \right]^2 \quad (2.6)$$

Because the timing of the data and clock transitions are synchronous, the spread spectrum product of the navigation data and the C/A code has exactly the same spectrum as that of the C/A code alone. Finally, the spectrum of the spreading signal is moved to the position of f_{LI} as the centre frequency as shown in Figure 2.2 (b). The GPS signal next passes through a propagation channel (assume zero channel delay for the sake of simplicity and without loss of generality) with additive white noise, which has power spectral density N_0 , to form the received signal as shown in Figure 2.2 (c). The received signal is then

down-converted and filtered by processing in the front-end. The resulting signal is a filtered intermediate frequency signal as shown in Figure 2.2 (d).

2.1.4 C/A Code Auto-correlation and Cross-correlation

The definition of the autocorrelation for function $f(t)$ is:

$$R(\tau) = \int_{-\infty}^{+\infty} f(t)f(t + \tau)dt \quad (2.5)$$

The autocorrelation function of white noise is zero everywhere except at a time difference of $\tau = 0$. That is, a vector of samples of a white noise process is uncorrelated with any shifted version of itself. The PRN code sequence has a similar autocorrelation as that of white noise. But its autocorrelation plot is periodic and the autocorrelation values are not zeros at any time shift greater than one code chip width because the PRN sequence is finite in length. The autocorrelation peaks occur at $\tau = 0$ and $\tau = \pm nT_c$, where T_c is the chip width in seconds and n is an integer value. Its second peak values are +63 or -65 when the time shift is greater than one code chip.

Two independent, discrete, finite length white noise vectors are uncorrelated with each other, making their circular cross-correlation values ideally zero (Maybeck, 1994). Cross-correlations between Gold codes in the same family are not completely zero, but are reasonably low compared to either code's autocorrelation peak. Any two Gold codes within the same family have the same cross-correlation values, which becomes lower relative to a given code's autocorrelation peak as a code gets longer (Peterson et al.,

1995). In the case of the C/A codes, the maximum power of any cross-correlation value is at least -23.9 dB lower than the autocorrelation peak (only with zero Doppler offset), (Kaplan, 1996).

The autocorrelation of PRN 1 and the cross-correlation of two GPS Gold codes (PRN 1 and PRN 2) are shown in Figure 2.3. The auto and cross-correlation plots for all other PRNs have similar values.

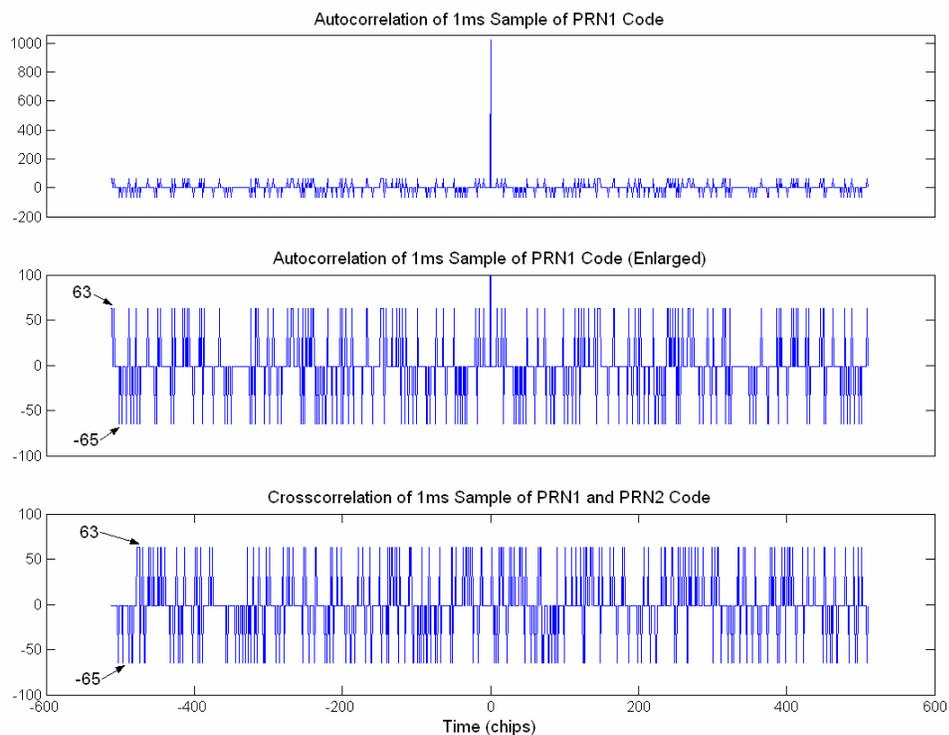


Figure 2.3: Autocorrelation and cross-correlation of GPS gold codes

2.2 Conventional Hardware Front-end and GPS Receiver

There are various GPS receivers specifically designed for various range of applications. It is very difficult to define a typical type of conventional hardware GPS receiver.

However, a characterization of a generic GPS receiver can still be made based on the receiver design requirements. A generic GPS receiver functional block diagram is shown in Figure 2.4. A GPS receiver can be broken down into various components. First, the antenna, possibly followed by a pre-amp, receives the RF GPS signals. After the antenna, comes the front-end that performs down-conversion, filtering, and digitization to convert the RF signal to an intermediate frequency signal. The next section, signal processing, includes acquisition, tracking, and navigation message demodulation to produce the pseudorange, and carrier phase measurements, as well as ephemeris. The final component performs the navigation processing to obtain the user position and velocity solution.

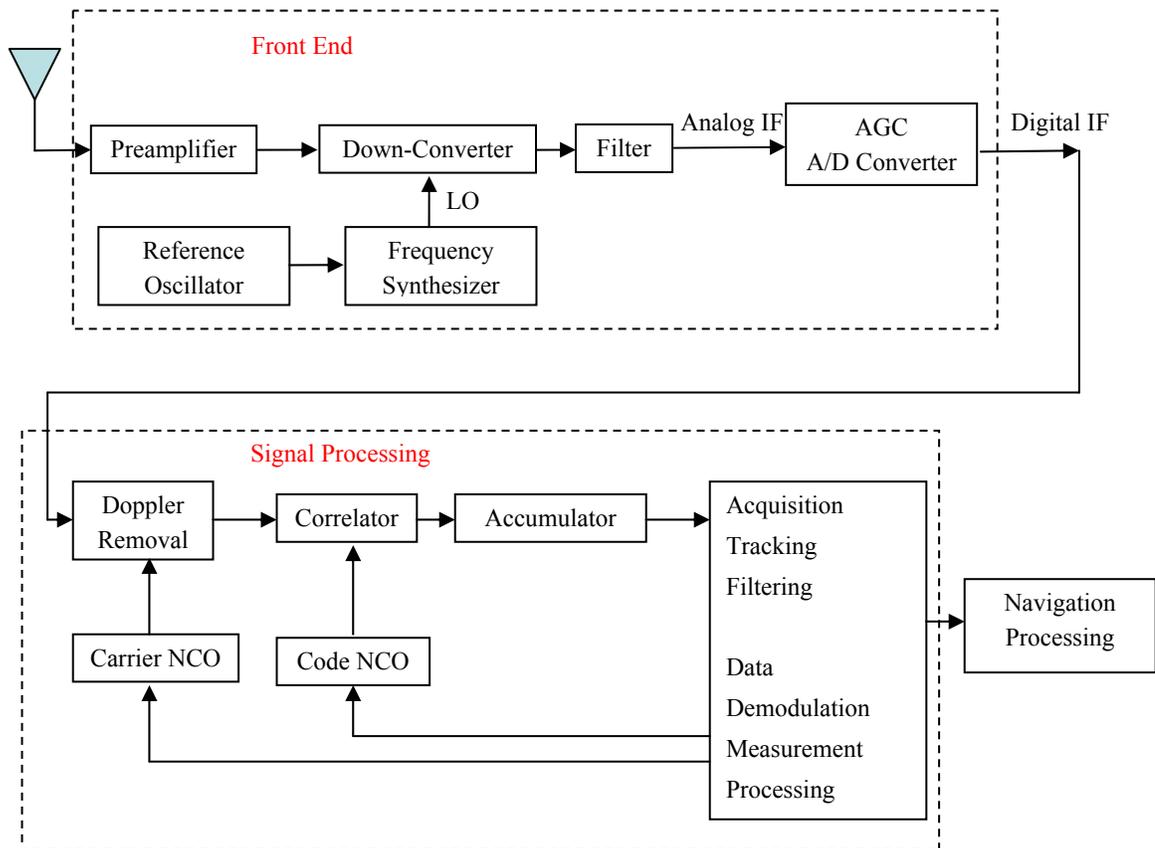


Figure 2.4: Generic GPS receiver functional block diagram

2.2.1 Front-end

High carrier frequencies are necessary to propagate signals from space through the atmosphere. However, these high frequencies are not easy to filter and demodulate. Therefore, down-conversion, by mixing the incoming signal and noise with a Local Oscillator (LO), is applied to reduce the received carrier frequency to a more manageable value.

Usually, the receiver front-end consists of filtering and limiting, a Lower Noise Amplifier (LNA), a frequency synthesizer, and down-conversion. Initial filtering, limiting and LNA can be housed within the antenna to comprise an integrated antenna electronics (Van Dierendonck, 1996). As a result, the signal after the antenna becomes an RF signal plus narrow band noise. This signal will be mixed with the LO in the following step. Two-stage down-conversion to IF is typical, but one stage down-conversion and even direct L-band digital sampling have also been used (Akos, 1997). Only one stage is represented here as an example. The processing of the GPS signal in the front-end can be depicted analytically as Equation 2.6 (a) (b) (c) (Van Dierendonck, 1996). The errors induced by the mixing process are included.

$$S_{IF}(t) + n_{IF}(t) = (S(t) + n(t)) \cdot 2 \cos(\omega_{Lo}t) \quad (2.6 \text{ a})$$

+ Harmonics + LO feedthrough + image noise

$$S_{IF}(t) = AC(t)D(t)\{\cos((\omega_{L1} + \omega_{Lo} + \Delta\omega)t + \phi_0) + \cos((\omega_{L1} - \omega_{Lo} + \Delta\omega)t + \phi_0)\} \quad (2.6 \text{ b})$$

$$n_{IF}(t) = r(t) \cos(\omega_{IF}t + \varphi(t)) \quad (2.6 \text{ c})$$

where

- A is the signal amplitude
- $S_{IF}(t)$ is the intermediate frequency signal
- $n_{IF}(t)$ is the noise after down conversion
- $\Delta\omega$ is the frequency offset (Doppler, etc.)
- ϕ_0 is the nominal (but ambiguous) carrier phase
- ω_{Lo} is the frequency of local oscillator
- ω_{IF} is the intermediate frequency
- $\varphi(t)$ is the noise phase

When down-converting the desired signal, the frequency difference term specifies the generated IF frequency; any other frequencies are removed from the signal through filtering. Harmonic frequencies, generated by the mixer because it is a non-linear device, are generally much higher than the IF frequency and are easily filtered out. Local oscillator feed-through is a small local oscillator frequency component that “leaks through” the demodulation process. Image noise is caused by noise energy at the image frequencies of the mixing process, located at $\omega_{L1} - 2\omega_{LO}$ and $\omega_{L1} - 2\omega_{IF}$. The frequency plan of a receiver specifies all of the previously mentioned frequencies (LO, IF, carrier, image, and harmonic frequencies) so that only the desired signal is present and the other frequencies are rejected by filtering. The frequency plan design is key to maintaining signal quality at the final IF stage. The performance of the filters used in the down-conversion process is also critical to removing out-of-band interference, thereby increasing the signal-to-noise ratio at the final IF.

Once the signal has been down-converted and filtered, it is ready to be sampled. Sampling performs analog-to-digital conversion of the GPS signal. The analog signal is typically quantized using 1 to 5-bit resolution. Five-bit resolution results in only 0.5 dB of loss and, counter-intuitively, a greater number of bits do not yield appreciable improvement (Van Dierendonck, 1996). An Automatic Gain Control (AGC) circuit is used to maximize sampling resolution. The AGC scales the received signal spectrum so that the A/D converter uses the full resolution of the A/D converter as the received signal and noise power fluctuate. Most commercial receivers use 1-bit quantization. This implementation does not require AGC, provided that the front-end has the required dynamic range to accommodate gain variations and expected interference. The sampling frequency can be selected to convert the IF frequency directly to base band or to some residual frequency offset. The sampling frequency can also alias the signal to some residual frequency offset by sampling the final IF frequency at less than the Nyquist rate. This type of sampling has the effect of causing a frequency and phase reversal, due to the high-side mixing process that occurs.

2.2.2 Signal Processing

After down-conversion and sampling, the digitized signal contains the GPS spreading and data modulation at a reduced IF frequency. Signal acquisition must be done prior to tracking.

GPS signal acquisition is a two-dimensional search process in which a replica code and carrier are aligned with the digitized IF signal. The correct alignment can be identified by

measurement of the output power of the correlators. In other words, when both the code phase and carrier Doppler match the incident signal, the signal is despread. Acquisition is a coarse synchronization process giving estimates of the code phase and carrier phase. This information is then used to initialize the tracking loops. After acquisition, control is then handed over to the tracking loop, the function of which is to track the variation in code phase or carrier Doppler.

The first step is signal demodulation as shown in Figure 2.4. Demodulation is the process of removing any residual carrier and the spreading modulation via code correlation. First the digital IF is treated to carrier removal by the replica carrier (plus carrier Doppler) to produce in-phase (I) and quadra-phase (Q) sampled data. Note that the replica carrier signal is being mixed with all of the GPS Space Vehicle (SV) signals and noise at the digital IF stage. The I and Q signals at the outputs of the mixers have desired phase relationships with respect to the desired SV. However, at this point, the code stripping processes that collapse these signals to base band have not yet been applied. Therefore, the I signal at the output of the in-phase mixers would consist mostly of thermal noise multiplied by the replica digital sine wave (to match the SV carrier at IF) and the Q signal at the output of the quadra-phase mixers would be mostly thermal noise multiplied by the replica digital cosine wave (to match the SV carrier at IF). The desired SV signal remains buried in noise until the I and Q signals are collapsed to base band by the code stripping process that follows. The I and Q signals are then correlated with early, prompt, and late replica codes (plus code Doppler). The prompt replica code phase is aligned with the incoming SV code phase, producing maximum correlation if it is tracking the incoming

SV code phase. The replica carrier (including carrier Doppler) signals are synthesized by the carrier Numerical Controlled Oscillator (NCO) and discrete sine and cosine mapping functions and the replica codes are synthesized by the code generator, a 3-bit shift register and the code NCO. To this point, the IF signal has been converted to base band after carrier removal and code correlation.

Next, the base band I and Q signals are accumulated and “dumped” (latched) at a fixed time intervals. The three in-phase components are integrated and dumped to produce I_E, I_P, I_L , while the three quadra-phase components similarly produce Q_E, Q_P, Q_L . The integration and dump accumulators provide filtering and re-sampling at a lower base band rate which depend on the desired pre-detection bandwidth. The integration time is a compromise in the design. It must be as long as possible to operate under weak signal conditions and it must be as short as possible to operate under high dynamic stress signal conditions.

The integrated and dumped signals are then relayed to the discriminators which provide the noisy estimates of the errors between the input signals and locally generated replicas due to low SNR. The classification of the discriminators is defined by the type of tracking loop used: Delay Lock Loop (DLL), Phase Lock Loop (PLL) or Frequency Lock Loop (FLL). A DLL discriminator outputs the code phase error of the input signal and the replica code. A PLL discriminator produces the phase error and an FLL discriminator provides the frequency error.

The loop filter performs filtering on the perceived signal demodulation errors after the discriminator as the discriminator outputs are very noisy. The objective of the loop filter is to reduce noise in order to produce an accurate estimate of the original signal at its output. The filtered error estimations are then input to the carrier NCO or code NCO to adjust the locally generated carrier and code phase to match the input GPS signals.

The spreading code and the residual carrier must be simultaneously tracked in order to de-modulate the received signal. Code tracking is the process of using the code autocorrelation function to maintain alignment between the code modulation in the signal and a locally generated replica. Typically, a Delay Lock Loop is used. Code tracking loops can be either coherent or non-coherent, which is to say that they operate with or without carrier phase information (Van Dierendonck, 1996). Misalignments between the received signal and the locally generated code are sensed by comparing the autocorrelations of the parallel correlators.

To track the received carrier frequency and phase, carrier tracking should be implemented. The carrier tracking loop can consist of either a FLL or a PLL. Frequency tracking is more robust than phase tracking, but is not as precise as phase tracking (Kaplan, 1996). The prompt I and Q values from the code correlation are used as the inputs for the frequency or phase discriminator which provides a noisy frequency or phase error measurement between the received signal and the local NCO generated signal.

If lock is lost on either the code or the carrier, the receiver will lose lock on the signal and must re-acquire it. The complete operation from signal acquisition to tracking, maintain tracking, lose tracking, and re-acquisition is a closed loop dynamic process. The correct code phase and carrier phase measurements can be extracted when both DLL lock and PLL lock are achieved. Of the two signal components, the code is easier to track in dynamic environments, due to the 293 metres equivalent range of one chip length. Carrier frequency lock is more precise, but also more difficult to maintain than code lock, and carrier phase lock is the most difficult to achieve due to a carrier wavelength for L1 of only 19 cm. A phase error of 90 degrees between the received signal and the NCO generated frequency equals to a ranging misalignment of only 4.7 cm. As a result, the DLL is easy to lock but not precise, whereas the PLL is precise but hard to lock.

2.3 Error Sources

There are six classes of ranging errors (Parkinson, 1996) as shown in Table 2.2. A distinction must be made between the errors that affect position accuracy and those that affect GPS signal tracking. For example, errors that are slowly changing over time do not adversely affect the tracking performance of a GPS receiver, but they do affect the accuracy of the calculated position. In the thesis, not all of the listed errors are simulated due to time restrictions. Only the errors that greatly affect software GPS receiver tracking performance or ranging errors are simulated.

Table 2.2: GPS ranging errors

Ephemeris data	Errors in the transmitted location of the satellite
Satellite clock	Errors in the transmitted clock
Ionosphere	Errors in the corrections of pseudorange caused by ionospheric effects
Troposphere	Errors in the pseudorange corrections caused by tropospheric effects
Multipath	Errors caused by reflected signals entering the receiver's antenna
Receiver	Errors in the receiver's measurement of range caused by thermal noise, software accuracy, and interchannel biases

2.3.1 Ephemeris Errors

Uncertainties in the broadcast ephemeris produce orbital errors. These uncertainties are due to the accuracy limitations associated with the predicted nature of the broadcast ephemeris. Tests have shown that the orbital error is generally a few metres (Kaplan, 1996). A very effective way to handle orbital errors is to use post-processed precise orbits. Precise orbits are derived from an extensive reference network and the accuracy can be as high as a few centimetres. However, the ephemeris errors do not have a significant impact on the pseudorange measurements and receiver tracking performance relative to other errors. Thus, they are not simulated in the simulator developed herein.

2.3.2 Satellite Clock Error

Each GPS satellite carries an atomic clock with which to maintain accurate frequency and time. These clocks have stabilities of about 1 part in 10^{13} over a day and they are monitored closely and maintained by the control segment to maintain this precision

(Spilker, 1996b). A major source of apparent clock error is Selective Availability (SA). Since SA has been turned off in May, 2000, the remaining satellite clock errors are due to slowly changing signals of the clock and have no significant effect on receiver tracking. But they do result in an offset of pseudorange. For short periods of time, the clock bias drift can be obtained from the broadcast ephemeris. The satellite clock effect is taken into account in the simulator.

2.3.3 Ionospheric Error and Tropospheric Error

The ionospheric error is caused by the presence of free electrons as GPS signals pass through the upper layer of the atmosphere. The ionosphere affects the propagation speed of the GPS signal, resulting in a time-of-arrival error at the receiver. The effects of an ionospheric error on the signal code and carrier are different: it causes a delay on the code, but an advance on the carrier. But the magnitudes of the delay and the advance are equal. The effect on range may vary from 150 metres (at midday, during periods of maximum sunspot activity, with the satellite near the horizon of the observer) to less than five metres, (at night, during periods of minimum sunspot activity, with the satellite at the zenith) (Luo, 2001).

The tropospheric error is caused by the refraction of the GPS signal in the troposphere. Similar to the ionosphere in terms of effect, the troposphere also affects the propagation speed of the GPS signal. Since the troposphere is non-dispersive at GPS frequencies, it has the same effect on the L1 and L2 code and carrier, a net delay. The delay contains

two parts, the larger of which is caused by the dry atmosphere component, which is stable and predictable. The delay resulting from water vapour is smaller, but varies greatly.

Ionospheric and tropospheric errors exhibit some spatial correlation between received signals, depending on whether the received signals traveled through similar atmosphere on their way to the receiver. Signals that propagate through the same elevation angles typically share similar ionospheric and tropospheric errors. These errors can be characterized as noisy, slowly changing biases. A model to simulate ionospheric and tropospheric errors, developed by Luo (2001) is used in this thesis.

2.3.4 Multipath

Multipath occurs when reflected signals, in addition to the direct signal, reach the antenna. It depends greatly on the properties of the reflector, the antenna gain pattern, and the type of correlator used in the GPS receiver. Multipath interferes with the correlator's ability to precisely determine the time instant of signal reception. It affects both code and carrier phase measurements. Code multipath is much larger than carrier phase multipath. It can reach up to one-half of a chip length of the PRN code, assuming an environment in which the multipath signal strength never exceeds that of the direct signal (Goldhirsh and Vogel, 1989). By contrast, carrier phase multipath is always less than one-quarter of the carrier wavelength (Georgiadou and Kleusberg, 1988). Typically, for static observations, multipath is non-Gaussian in nature and shows sinusoidal oscillations with periods of a few minutes due to the change of satellite geometry. In kinematic applications, multipath behaves more randomly because the movement of the vehicle changes the reflecting

geometry in a relatively random way. Multipath errors are unique to each receiver, and uncorrelated between signals. They are not simulated in the project presented in this thesis due to time restrictions.

2.3.5 Receiver Noise Errors

Receiver noise errors can be divided into two categories. The first type includes those errors related to the chosen implementation. These effects include the AGC finite response time effects, sampling and quantization losses, losses in the code tracking design, and hardware resolution limitations. The second type originates from the practical limitations of component accuracy. These errors include receiver oscillator errors, filter-induced distortions on the signal, and inherent system noise. Typically, these errors are not completely minimized due to complexity and cost limitations. They are, rather, more related to the hardware implementation; some errors, such as receiver clock errors caused by the instability of the oscillator, greatly affect the receiver performance. The noise component is incorporated in this thesis as Gaussian white noise in the implementation of the simulator. The noise density is selected as a typical value -203 dBW/Hz. Filtering and quantization (1-bit and 2-bit) are performed in the simulation. The simulation of the receiver clock error is not conducted in the current version of the simulator.

CHAPTER 3

IF GPS SIGNAL SIMULATOR REALIZATION

3.1 IF GPS Signal Simulation Model Derivation

An IF GPS signal simulation model is developed before any discussion of implementation. The objective of this thesis requires the development of a software simulator to generate the digitized IF GPS signal immediately following the front-end and before input to the signal processing stage. The simulation model is based on this selected simulation point and takes into account all errors introduced during the signal propagation in the atmosphere and in the front-end.

A key consideration in developing the simulator is the simulation of various errors. The effects of these errors have been studied in detail by the GPS community. All error sources - specifically, errors due to the satellite clock, orbit, ionosphere, troposphere, etc. - behave in essentially the same manner. They all produce errors on the pseudorange and carrier phase measurements, namely, a code delay, or a carrier phase delay or advance error. The effect on the carrier phase is also indicated by the Doppler shift. For this reason, the simulation model is structured so as to handle any form of error by specifying the above parameters.

The derivation of the simulation model is along the real GPS signal propagation path. It starts with GPS signal generation in the satellites which is followed by signal propagation through the ionosphere and troposphere, signal reception at the receiver antenna and, finally, signal processing in the front-end as shown in Figure 3.1. The functional block of the front-end consists of signal reception by the receiver antenna, down-conversion, and IF processing which further includes filtering, sampling and digitization. All necessary functional components are analytically simulated in the following sections.

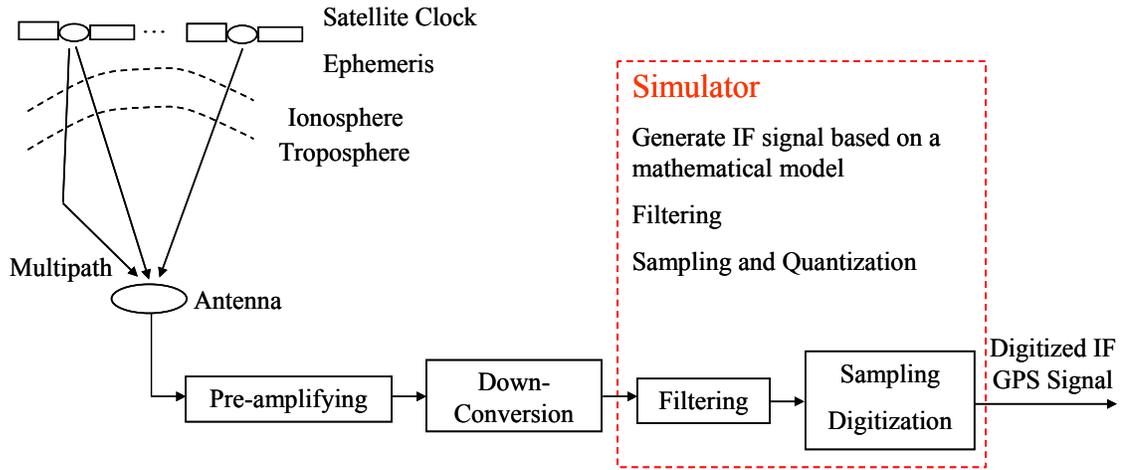


Figure 3.1: GPS signal propagation and processing in the front-end

3.1.1 Transmitted GPS Signal

For the sake of clarity, the analytical derivation starts with a single ideal GPS L1 signal as presented in Chapter 2. The L1 GPS signal transmitted by the satellite i can be represented as (Spilker, 1996a):

$$\begin{aligned}
 S_{Li} = & \sqrt{2P_t} D_i(t_r) C_i(t_r) \cos(\omega_{L1}t_r + \varphi_0) \\
 & + \sqrt{2P_t} D_i(t_r) Y_i(t_r) \cos(\omega_{L1}t_r + \pi/2 + \varphi_0)
 \end{aligned}
 \tag{3.1}$$

where

- S_{L1i} is the GPS L1 signal transmitted from SV_i
- P_i is the transmitted signal power
- ω_{L1} is the L1 carrier frequency
- t_{tr} is the signal transmit time
- φ_0 is the initial phase of the GPS signal

With minor amplitude modulation neglected (Spilker, 1996b), a satellite clock generator produces a waveform that, in general, can be represented as:

$$A \sin \left[(\omega_0 t + \varphi_0) + (\varphi_e + \omega_e t + 2\pi d \frac{t^2}{2}) + x(t) \right] \quad (3.2)$$

where

- $\omega_0 t + \varphi_0$ is the desired clock phase
- $\varphi_e + \omega_e t + 2\pi d t^2 / 2$ is the causal phase error effects
- φ_e is the fixed phase error offset
- ω_e is the frequency error
- $2\pi d t^2 / 2$ is the frequency drift or aging effect
- $x(t)$ is the random phase error

From Equation (3.2), it can be seen that satellite clock errors include a fixed phase bias, a frequency error, a frequency drift, and a random phase error. To obtain high accuracy position and time keeping, the satellite clock error is usually corrected by adjusting the signal transmission time. The satellite clock correction term is approximated by a polynomial $\delta t_{SV} = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + \Delta t_R$, where a_{f0} , a_{f1} , and a_{f2} are the polynomial correction coefficients corresponding to the phase error, frequency error, and frequency drift, respectively; the relativistic correction is Δt_R ; and t_{oc} is the reference

time for the clock correction. All of these parameters are estimated by the Control Segment, and are contained in the ephemeris data that is transmitted to users.

With the presence of the satellite clock error, the transmitted L1 C/A code signal can be expressed as:

$$S_{L1i} = \sqrt{2P_t} D_i(t_{tr} + \delta t_{SV}) C_i(t_{tr} + \delta t_{SV}) \cos[\omega_{L1}(t_{tr} + \delta t_{SV}) + \varphi_0] \quad (3.3)$$

where

δt_{SV} is the satellite clock error

Because this thesis focuses on L1 C/A code signal simulation, the P code component in Equation (3.1) is eliminated without significant effect for purpose of the simulation produced herein. The justification for removing this signal from the derivation is as follows. The power of the P code signal is half of that of the C/A code signal, which is about 20 dB below the noise floor. Also, the noise spreading properties apply to both P code and C/A code in the same manner. Additionally, if the signal simulator filters the incoming signal at the C/A code bandwidth, the band-pass filtering of the signal would filter out 90% of the P code signal power. Given these facts, it is reasonable to eliminate the P code component from the received signal for sake of simplicity of derivation. The parameter φ_0 in Equation (3.2) is assumed zero for simplicity, without loss of generality. It can be seen from the above equation that the satellite clock error causes a frequency offset and phase offset.

3.1.2 Propagation through the Atmosphere to the Antenna

The signal transmitted by a satellite travels through the atmosphere before reaching the receiver antenna. The value of the propagation delay is a major factor in the signal model. The ionosphere and troposphere affect the GPS signal and cause errors on both code phase and carrier phase. The GPS signal transmitted by satellite i can be represented by the following equation with the propagation delay, ionospheric and tropospheric errors taken into account.

$$S_{L1i} = \sqrt{2P_t} D_i(t - T_p - \delta t_{iono} - \delta t_{tropo} + \delta t_{SV}) C_i(t - T_p - \delta t_{iono} - \delta t_{tropo} + \delta t_{SV}) \cos[\omega_{L1}(t - T_p + \delta t_{iono} - \delta t_{tropo} + \delta t_{SV}) + \varphi_0] \quad (3.4)$$

where

- t is the GPS signal receiving time
- δt_{iono} is the ionospheric error
- δt_{tropo} is the tropospheric error
- T_p is the signal propagation time from satellite to receiver antenna

Noise, interference and multipath are also added prior to the signal's arrival at the antenna. The signal received by the antenna can be expressed as:

$$R_{L1i} = \sqrt{2P_r} D_i(t - T_p - \delta t_{iono} - \delta t_{tropo} + \delta t_{SV}) C_i(t - T_p - \delta t_{iono} - \delta t_{tropo} + \delta t_{SV}) \cos[\omega_{L1}(t - T_p + \delta t_{iono} - \delta t_{tropo} + \delta t_{SV}) + \varphi_0]_i + MP + noise + interference \quad (3.5)$$

where

- P_r is the received signal power

The multipath signal is composed of the portion of the GPS signal that is reflected. The multipath error relates to the signal path delay caused by the extra path length and the reflection coefficient of the reflector. The effect of multipath on the direct signal can also be shown in the code delay and carrier phase offset. In addition, it is attenuated and thus generally has a lower power level than the direct signal. The multipath signal is represented as:

$$MP = \alpha \sqrt{2P_r} D_i(t - T_p - \delta t_{iono} - \delta t_{tropo} + \delta t_{SV} - \delta t_{MP}) C_i(t - T_p - \delta t_{iono} - \delta t_{tropo} + \delta t_{SV} - \delta t_{MP}) \cos[\omega_{L1}(t - T_p + \delta t_{iono} - \delta t_{tropo} + \delta t_{SV} - \delta t_{MP}) + \varphi_0]_i \quad (3.6)$$

where

α is the signal power attenuation factor

δt_{MP} is the multipath signal path delay

Equation (3.5) is an expression of the received GPS signal at the receiver antenna with all observable errors included. As mentioned above, not all of these errors are simulated herein. With the exclusion of multipath errors and interference, the resulting signal model for the RF GPS signals that arrive at the receiver antenna is:

$$R_{L1i} = \sqrt{2P_r} D_i(t - T_d - \delta t_{iono}) C_i(t - T_d - \delta t_{iono}) \cos[\omega_{L1}(t - T_d + \delta t_{iono}) + \varphi_0]_i + noise \quad (3.7)$$

where

$$T_d = T_p + \delta t_{tropo} - \delta t_{SV}$$

Although it may not be obvious, the Doppler term is embedded in the product of $T_d - \delta t_{iono}$ and ω_{L1} in Equation (3.7). The following derivation can prove this. Let us start from the definition of the Doppler as shown in Equation (3.8) (Axelrad and Brown,

1996). It is equivalent to Equation (3.9) after multiplying by a time t to both sides of the equation and then multiplying by a factor $\frac{T_d - \delta t_{iono}}{T_d - \delta t_{iono}}$ on the right side. It can be seen that $(\bar{v}_i - \bar{v}_u) \cdot \bar{l}_i$ is the relative velocity in the line of sight from the user to the satellite, and thus $(\bar{v}_i - \bar{v}_u) \cdot \bar{l}_i * t$ is the distance change between the user and the satellite; $T_d - \delta t_{iono}$ is the time delay of the signal propagation between the satellite and the user, and thus $C * (T_d - \delta t_{iono})$ is also the distance change between them. As a result, $\frac{(\bar{v}_i - \bar{v}_u) \cdot \bar{l}_i * t}{C * (T_d - \delta t_{iono})}$ equals one, and thus Equation (3.10) stands. Therefore, an equivalent expression of Equation (3.7) is Equation (3.11) with an obvious Doppler component.

$$f_d = -\left(\frac{\bar{v}_i - \bar{v}_u}{c} \cdot \bar{l}_i\right) * f_{L1} \quad (3.8)$$

where

- f_d is the Doppler frequency
- \bar{v}_i is the velocity of the satellite i
- \bar{v}_u is the velocity of the user
- \bar{l}_i is the line of sight unit vector from the user to the satellite
- c is the speed of light
- f_{L1} is the L1 frequency

$$f_d * t = -\frac{(\bar{v}_i - \bar{v}_u) \cdot \bar{l}_i * t}{C * (T_d - \delta t_{iono})} * f_{L1} * (T_d - \delta t_{iono}) \quad (3.9)$$

$$f_d * t = -f_{L1} * (T_d - \delta t_{iono}) \quad (3.10)$$

$$R_{L1i} = \sqrt{2P_r} D_i(t - T_d - \delta t_{iono}) C_i(t - T_d - \delta t_{iono}) \cos[(\omega_{L1} + \omega_d)t + \varphi_0]_i + noise \quad (3.11)$$

where

$$T_d = T_p + \delta t_{tropo}$$

$$\omega_d = 2\pi f_d$$

The received signal at the antenna can be expressed as:

$$R_{L1} = \sum_{i=1}^N \left\{ \sqrt{2P_r} D_i(t - T_d - \delta t_{iono}) C_i(t - T_d - \delta t_{iono}) \cos_i[\omega_{L1}(t - T_d + \delta t_{iono}) + \varphi_0] \right\} + noise \quad (3.12)$$

where

N is the number of satellites in view

3.1.3 Down-conversion

So far, the modeled GPS signal is still at radio frequency. After the signal propagates to the front-end and is down-converted to the IF frequency, the model will be a signal representation at intermediate frequency. The simulation produced in this thesis is not meant to simulate an actual front-end processing stage. The following derivation will focus on the signal analytical expression. No matter how many stages of the down-conversion are used in a front-end, the final output is the desired IF signal after mixing and band-pass filtering. Assume that the local oscillator-generated signal is as shown in Equation (3.13), and that the result of the cosine mixing is shown in Equation (3.14).

$$LO_1(t) = 2 \cos(\omega_{L1}t) \quad (3.13)$$

$$\begin{aligned}
S_{IF} &= R_{Lli} * LO_1 \\
&= \sqrt{2P_r} D_i(t - T_d - \delta t_{iono}) C_i(t - T_d - \delta t_{iono}) \\
&\quad \left\{ \cos[\omega_{L1}(t - T_d + \delta t_{iono}) - \omega_{L1}(t + \delta t_r) + \varphi_0] + \cos[\omega_{L1}(t - T_d + \delta t_{iono}) + \omega_{L1}(t + \delta t_r) + \varphi_0] \right\}
\end{aligned} \tag{3.14}$$

Filtering out the high frequency component, we obtain:

$$\begin{aligned}
S_{IF} &= R_{Lli} * LO_1 \\
&= \sqrt{2P_r} D_i(t - T_d - \delta t_{iono}) C_i(t - T_d - \delta t_{iono}) \cos[\omega_{L1}(t - T_d + \delta t_{iono}) - \omega_{LO1}(t + \delta t_r) + \varphi_0] \\
&= \sqrt{2P_r} D_i(t - T_d - \delta t_{iono}) C_i(t - T_d - \delta t_{iono}) \cos[(\omega_{L1} - \omega_{LO1})t - \omega_{L1}(T_d - \delta t_{iono}) - \omega_{LO1}\delta t_r + \varphi_0] \\
&= \sqrt{2P_r} D_i(t - T_d - \delta t_{iono}) C_i(t - T_d - \delta t_{iono}) \cos[\omega_{IF}t - \omega_{L1}(T_d - \delta t_{iono}) - \omega_{LO1}\delta t_r + \varphi_0]
\end{aligned} \tag{3.15}$$

where

- ω_{LO1} is the frequency of the local oscillator
- ω_{IF} is the expected intermediate frequency
- δt_r is the receiver clock error

The receiver clock error, which is caused by the instability of the oscillator, is a significant issue in receiver operation. The oscillator error can produce a frequency shift in the received signal, and therefore a significant frequency shift also occurs in the down-converted signal. A portion of the signal power would be filtered out when filtering is applied because of this frequency shift. From the tracking point of view, the Doppler is also changed. As well, the code DLL may lose track because of the misalignment of the carrier. In light of all of these facts, the receiver clock error commands our attention when studying receiver performance. However, due to time restrictions, this error will not be discussed in this thesis; instead, it is assumed to be zero herein and a full treatment is left for future work.

Equation (3.15) represents the IF signal for the satellite i . Finally, taking noise into account, the composite IF signal model including all satellites in view is:

$$S_{IF} = \sum_{i=1}^N \left\{ \sqrt{2P_r} D_i(t - T_d - \delta t_{iono}) C_i(t - T_d - \delta t_{iono}) \cos[\omega_{IF}t - \omega_{L1}(T_d - \delta t_{iono}) + \varphi_0] \right\}_i + noise \quad (3.16)$$

3.1.4 Intermediate Frequency Processing

The IF GPS signal model has been established as shown in Equation (3.16). This model forms the basis of the simulation discussed in this work. The composite signal then propagates to a band-pass filter and A/D converter to be filtered and digitized and, finally, the digitized IF GPS signal is generated. A band-pass filter is simulated, the bandwidth of which can be selected as required. One-bit and two-bit digitization are realized in this thesis, the details of which are described in Section 3.2.11.

3.2 Simulator Realization

The IF GPS signal simulator was developed in MATLAB, with some functions written in C++ and called by MATLAB to accelerate speed of operation. The receiver trajectory, satellite ephemeris, required carrier-to-noise ratio and other environment parameters are specified to create a valid IF GPS signal. The signals of visible satellites for the receiver as well as noise are added together to generate the combined IF GPS signal. The functional structure of the digitized IF GPS signal simulator is shown in Figure 3.2.

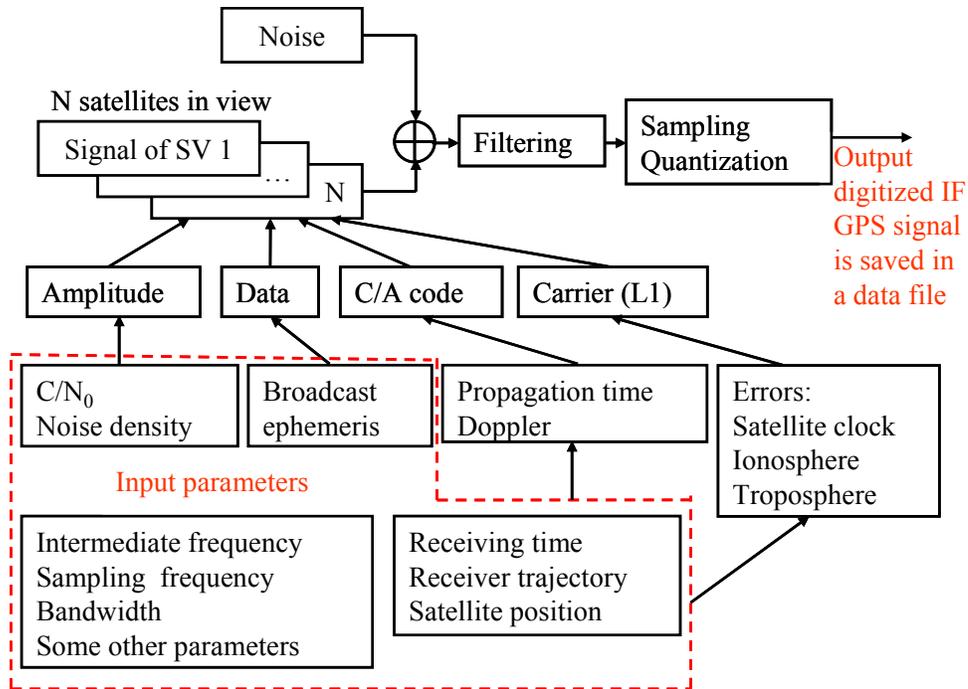


Figure 3.2: Digitized IF GPS signal simulator structure

This structure is completely based on the signal simulation model as represented in Equation (3.16). To start a simulation, the simulation time period has to be assigned which is referred to the signal receiving time. The receiver trajectory is simulated and the satellite's positions are computed based upon the simulation time and the satellite ephemeris. Then, the signal propagation time, Doppler, and errors are calculated to generate the required components, such as signal amplitude, navigation data, C/A code, carrier, etc. In this thesis, satellite clock error, ionospheric and tropospheric errors are modeled to express the signal propagation environment. The IF GPS signal for a single satellite is the product of these components. Then, the composite IF signal is computed by the summation of noise and the signals of all satellites in view. After filtering and quantization, the digitized IF GPS signal samples are obtained and saved in a data file.

3.2.1 Satellites in View

Normally, four to seven satellites are visible for a given GPS user. Only the signals transmitted by these satellites can reach the receiver antenna. The signal simulated in this thesis is the combination of these signals, together with the environment noise. If the receiver positions and satellite positions are known, the visible satellites' PRN numbers can be calculated.

The receiver and satellite trajectories during the simulation period are fundamental components required to conduct the simulation. The receiver trajectory can be simulated for different scenarios. Three trajectory modes are designed, namely mode 0, mode 1, and mode 2. The first one is a static mode. Mode 1 is that the receiver starts at a reference point, moving northward along a straight line with a constant velocity. Mode 2 is that the receiver starts at a reference point, moving along a circle with a constant angular velocity. The latter two are shown in Figure 3.3.

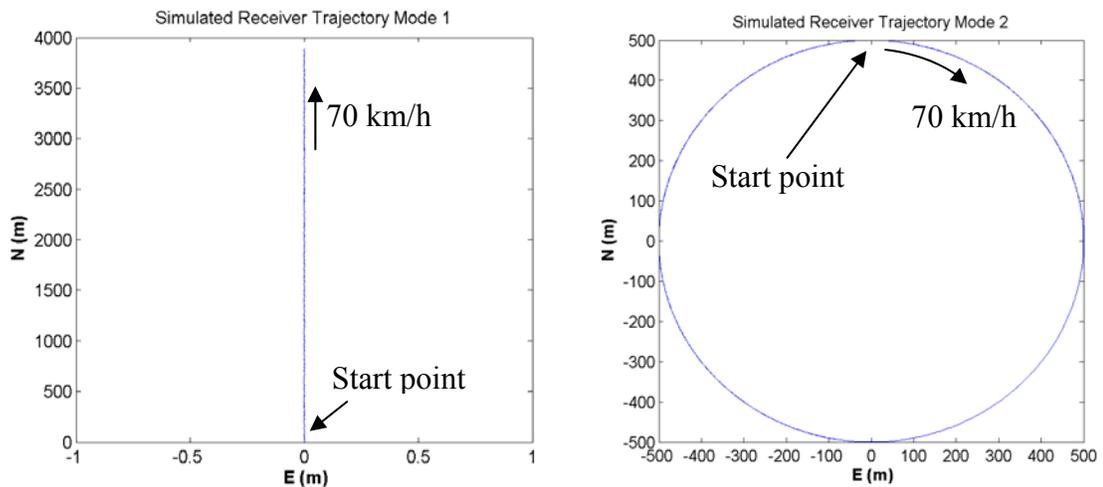


Figure 3.3: Simulated receiver trajectory modes

To compute the satellite trajectory, the satellite ephemeris must be known. In this simulation, the broadcast ephemeris is used. It can be obtained from the website of the Crustal Dynamics Data Information System Institute (CDDIS), one of the analysis centres of the International GPS Service (IGS). The ephemeris file, in global compressed RINEX navigation format, is available for any specified day. It includes the ephemeris for all available global satellites with a 2-hour time interval.

Given the GPS time, a satellite's position can be calculated from the satellite ephemeris data. The satellites in view can be determined with knowledge of the simulation time, receiver position and satellite position. Considering the fact that satellites with very low elevation angles suffer from increased power attenuation and atmospheric errors, an elevation angle cut-off of 10° was considered in the simulation tests. However, this value can be easily changed to the required value by the users.

Both the satellite position and the receiver position should be in the same coordinate system. The simulated receiver position is in the local frame, while the satellite position computed from the ephemeris is in the Earth-centred, Earth-fixed (ECEF) frame. So, the satellite position is converted to the local frame to be consistent with the receiver position. Assuming that the satellite's position in this local frame is (x_s, y_s, z_s) and the receiver's position is $(0,0,0)$, the elevation angle is calculated as:

$$\theta = \arctan\left(\frac{z_s}{\sqrt{x_s^2 + y_s^2}}\right) \quad (3.17)$$

3.2.2 Amplitude Modeling

The signal amplitude is related to the signal power as:

$$P = \frac{A^2}{2} \quad (3.18)$$

where

P is the signal power

In the simulated IF GPS signal model, the signal power is the effective carrier power of the received signal after amplification, transmission through the cables, and filtering. The received RF signal is transmitted from the satellite by shaped pattern antennas, and attenuated as a result of traveling through the atmosphere as well as free space loss. In a conventional receiver, the minimum expected carrier power at L1 for the C/A code is -160 dBW. The effective carrier power is the combination of this power level with the effects of antenna gain, loss from the transmission line, the filtering bandwidth, and the amplifier gain (Spilker, 1996a).

However, receiver performance is more dependent on the signal to noise ratio rather than on the absolute power. The carrier to noise ratio (C/N_0), which is the ratio of total carrier power to the noise power density, is frequently used in receiver design and performance estimation; thus, it is an important value used in the simulator design presented herein. Thus, the expected C/N_0 and the noise density are defined as input parameters. The signal power is calculated as:

$$P = P_c = 10^{\frac{C/N_0 + N_0}{10}} \quad (3.19)$$

where

P_c is the effective carrier power

N_0 is the noise density

3.2.3 Navigation Data Simulation

In an actual GPS system, the navigation data is a 50 Hz bi-phase coded. So, each data bit is 20 ms long. Since the period of the C/A code is 1 ms, there are 20 C/A codes in one data bit. Thus, all 20 C/A codes have the same phase in one data bit. Thirty data bits make a navigation word; 10 words make a sub-frame; 5 sub-frames make a page; and 25 pages make a super frame. Figure 3.4 shows these relationships.

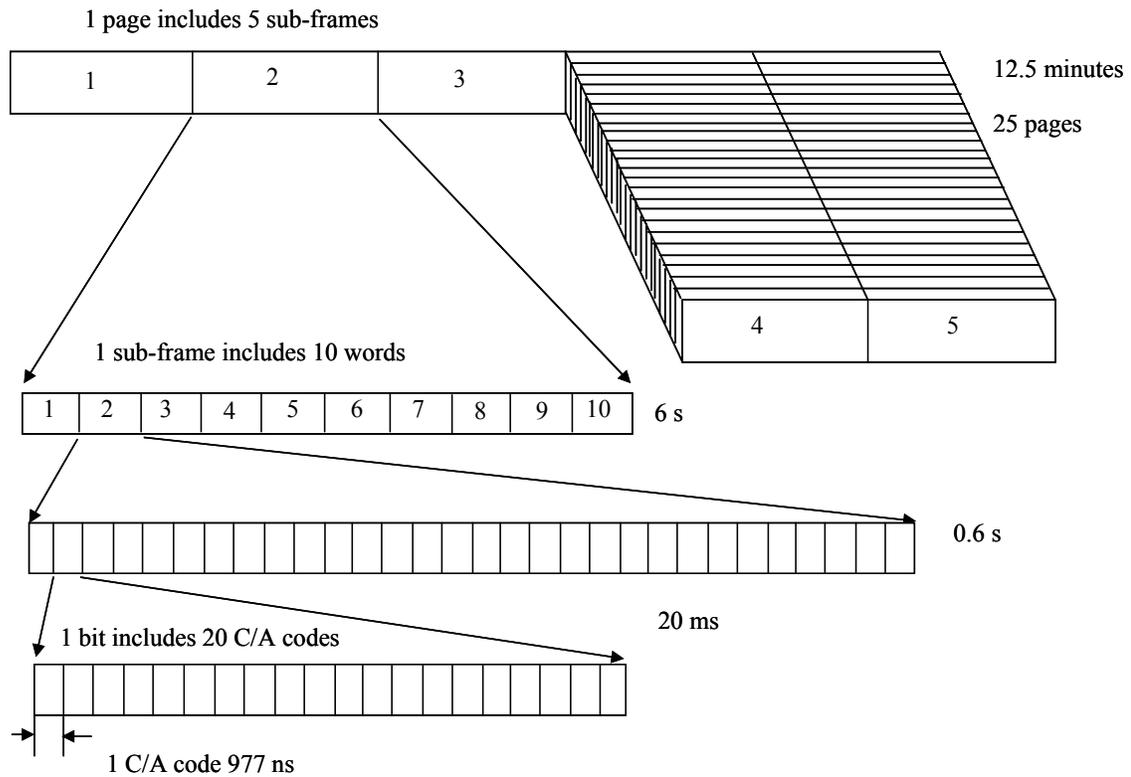


Figure 3.4: GPS navigation data format

The GPS time is given by the number of seconds in one week and this value is reset every week at the start of a week. At this time, the cyclic paging to sub-frames 1 through 5 will restart with sub-frame 1 regardless of which sub-frame was last transmitted prior to the start of the week. The cycling of the 25 pages will restart with page 1 of each of the sub-frames, regardless of which page was the last to be transmitted prior to the end of the week. All upload and page cutovers will occur on frame boundaries. Accordingly, new data in sub-frames 4 and 5 may start to be transmitted with any of 25 pages of these sub-frames.

The first three sub-frames of the navigation data contain all of the ephemeris parameters. It is possible to calculate the satellite position with all of these parameters. Sub-frames 4 and 5 include almanac data which provides a truncated, reduced precision set of ephemeris parameters. The almanac data is mainly used to aid acquisition and to give the approximate Doppler and delay information. Each of the 25 pages of almanac data is different. The simulation of sub-frames 4 and 5 is very complicated and not very meaningful for this research. Therefore they are simulated only by formatting the first two words - the TLM word and the handover (HOW) word which include the preamble, time of the week count message, sub-frame ID and parity check bits. The formats of the TLM and HOW words are shown in Figure 3.5. They are the same for each sub-frame. To simulate the data of other three sub-frames, a proper ephemeris file for the user-defined time is downloaded from the website of CDDIS, as mentioned in Section 3.2.1. Then these parameters, combined with the proper scale factors, (Spilker, 1996b) are

converted to binary bits which fit the navigation data format. The first three sub-frames are simulated with 1 and -1 sequence in the simulation.

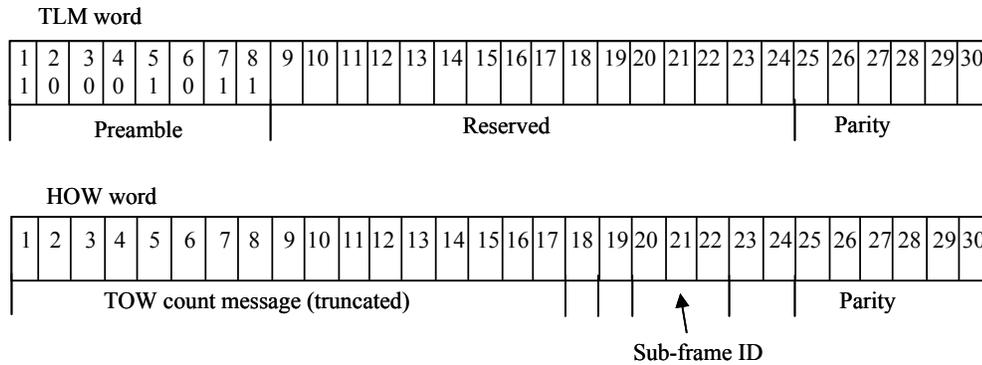


Figure 3.5: TLM and HOW word

3.2.4 C/A Code Modeling

The C/A code signals are generated from the product of two 1023-bit PRN sequences, G1 and G2. Both G1 and G2 are generated by 10-stage linear shift registers which are driven by a 1.023 MHz clock. The C/A code generator is shown in Figure 3.6 (Tsui, 2000). Both G1 and G2 shift registers have 10 stages; thus, the generated sequence length is 1023 ($10^{10} - 1$). The C/A code sequence is the result of modulo-2 adding of the outputs of G1 and G2. The output of G1 is from the last bit of the shift register, while G2's output is generated from two bits which are referred to as code phase selection through another modulo-2 adder.

A function for generating the C/A code was coded in MATLAB, according to Figure 3.6 and the above operation rule. It produces C/A code sequences, +1 and -1, for each satellite with the PRN number as the input parameter.

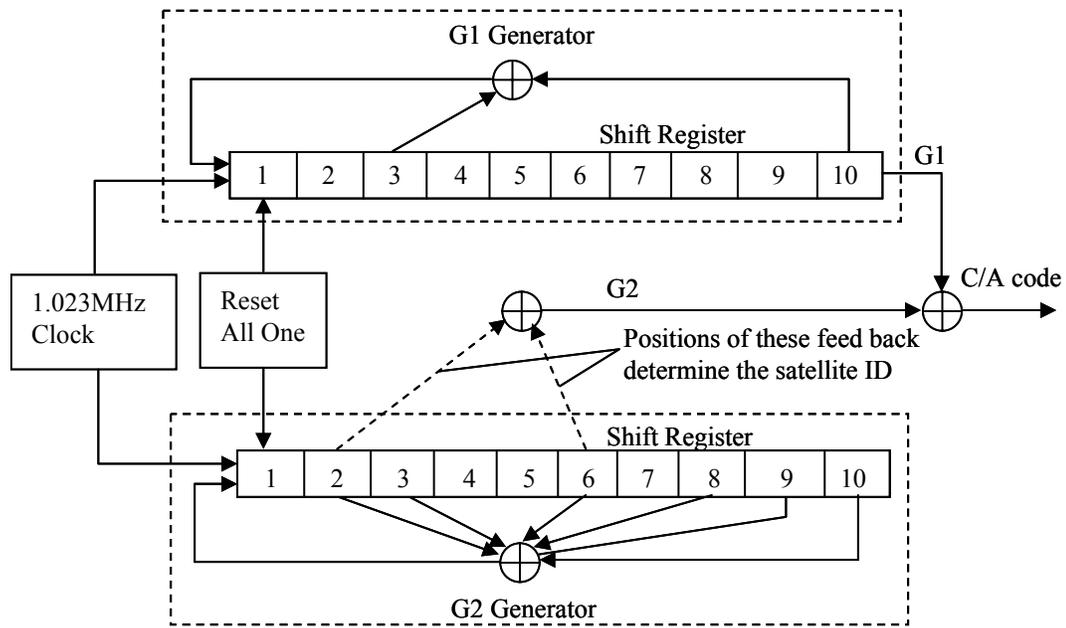


Figure 3.6: C/A code generator

3.2.5 Signal Propagation Time and Doppler

Intuitively, the signal propagation time is simply the distance between the transmitter and receiver divided by the speed of light. However, the position of the receiver, which is on the Earth, changes during signal propagation due to the Earth rotation. The effect of Earth rotation is not a trivial issue that can be ignored by the GPS system. The following discussion takes this into account.

For the sake of simplicity, consider the computation of the propagation time in the Earth-Centred Inertial (ECI) frame first. Referring to Figure 3.7, there is a satellite at position P_t at the instant a signal is transmitted and a receiver is fixed at the position R_t on the Earth's surface. Assume the satellite-transmitted signal propagates with speed c in a straight line in the ECI frame. At the time of reception, the receiver has moved to the

position R_r in the ECI frame due to the Earth rotation. Suppose that the signal propagation time from the satellite to the receiver at position R_r is T_p ; the transmission time is t_t ; the reception time is t_r ; and the receiver velocity is \bar{v} , which is the same as that of the Earth rotation. Equation (3.20) presents the above case from the point of view of a non-rotating ECI frame.

$$CT_p = \left| \bar{r}_{P_t R_t} + \bar{v} T_p \right| \quad (3.20)$$

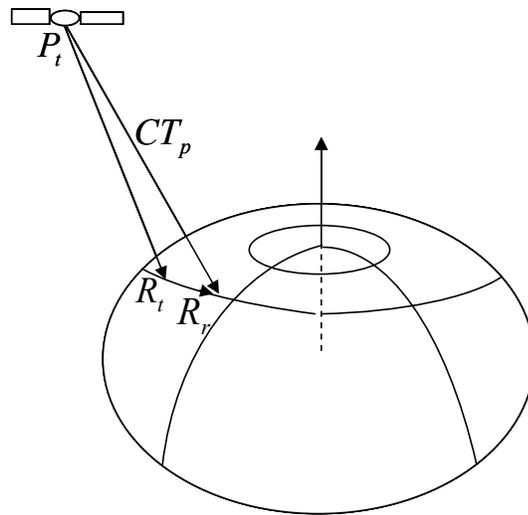
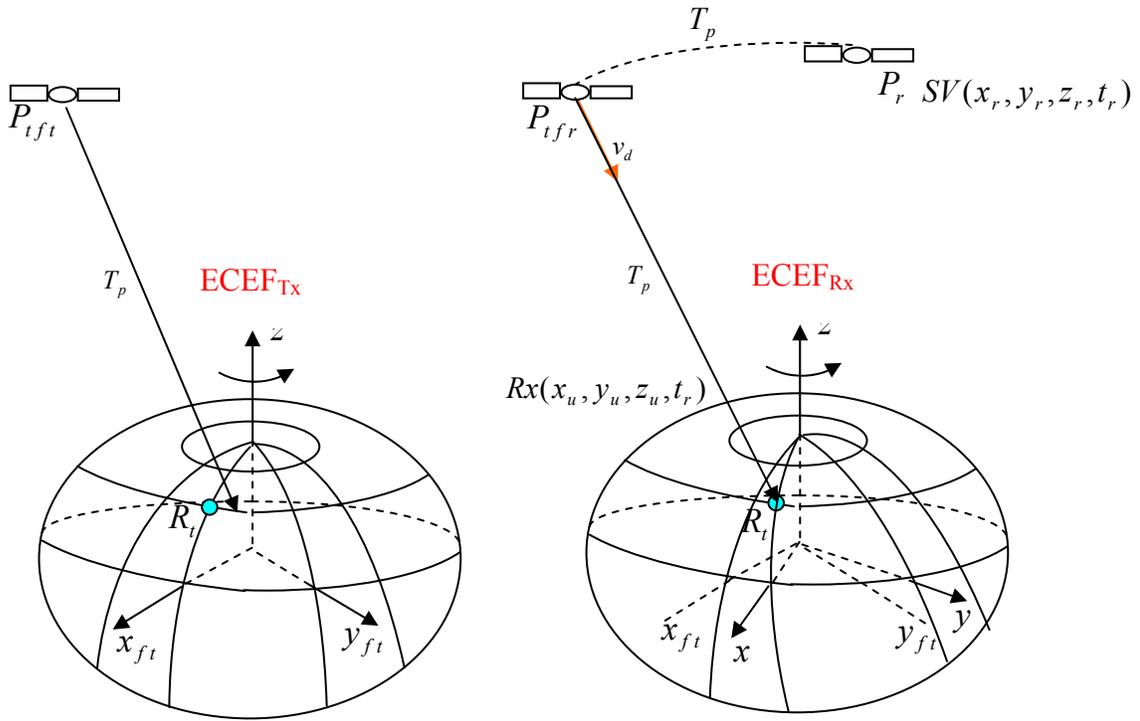


Figure 3.7: Effect of Earth rotation on signal propagation

However, the GPS application is usually in an ECEF reference frame, in which the Earth rotates about a fixed axis with a defined rotation rate, $\omega_e = 15$ "/s. This reference frame is designated by WGS-84 (Kaplan, 1996). During the time that the signal propagates from the satellite to the receiver, the reference coordinate frame changes because of the Earth rotation. As the receiver is tied to the Earth, its coordinates do not change in ECEF although its position in the ECI frame changes. The satellite position also changes between signal transmission time and reception time.

The distance between the satellite position at transmission time and the receiver position at reception time is required to compute the signal propagation time. But these two positions straddle two time instants and, thus, two ECEF frames, ECEF frame at transmission time (ECEF_{Tx}) and ECEF frame at reception time (ECEF_{Rx}) as shown in Figure 3.8. Computation of signal propagation time, therefore, requires an iterative process, using satellite coordinates from these two frames. First, let us discuss the computation in ECEF_{Rx} referring to Figure 3.8 (b). Some assumptions have to be made: (1) all computations and position coordinates are in ECEF_{Rx}; and (2) the satellite position $P_{t_{fr}}$ at transmission time t_t moves to P_r at reception time t_r with a constant velocity (v_x, v_y, v_z) . Signal propagation time is represented by T_p . Only the reception time and receiver position at this instant are known in the simulation. The satellite position at t_r can be calculated from the time t_r and the ephemeris data, and thus its position $P_{t_{fr}}$ at t_t can be expressed by P_r , (v_x, v_y, v_z) and T_p as shown in Equation (3.21) which consists of four equations and four unknowns. Solving the system of equations yields an estimate of the propagation time (and thus the transmission time) and the approximate coordinates of the satellite at that time. However, the resulting satellite coordinates represent the satellite position in ECEF_{Rx}. To confirm their correctness, the computation in ECEF_{Tx} is discussed in the following paragraph.



(a) ECEF frame at transmission time (b) ECEF frame at reception time

Figure 3.8: Satellite and receiver position in the ECEF frame at the transmission time and reception time

$$\begin{cases}
 x_{tfr} = x_r - v_x T_p \\
 y_{tfr} = y_r - v_y T_p \\
 z_{tfr} = z_r - v_z T_p \\
 \sqrt{(x_{tfr} - x_u)^2 + (y_{tfr} - y_u)^2 + (z_{tfr} - z_u)^2} = CT_p
 \end{cases} \quad (3.21)$$

The satellite position P_{tft} in $ECEF_{Tx}$ can be calculated from the ephemeris after obtaining the transmission time $t_t = t_r - T_p$. If this position is transformed to $ECEF_{Rx}$, theoretically, it should be the same as it is computed directly in this frame since P_{tfr} and T_p should also keep the same value. Therefore, the coordinates of the satellite position P_{tft} are

transformed to ECEF_{Rx} with Equation (3.22). A new signal propagation time T'_p is calculated in the same frame. Then, compare T'_p and T_p ; if they have a significant difference, a new satellite position in ECEF_{Tx} will be recomputed based on T'_p and the ephemeris data. Then, rotate it to ECEF_{Rx} to compute T'_p again; the old value of T'_p is saved to T_p . The above steps will be iterated until T'_p matches T_p within a desired range.

$$\begin{bmatrix} x_{tfr} \\ y_{tfr} \\ z_{tfr} \end{bmatrix} = \begin{bmatrix} \cos(\omega_e * T_p) & \sin(\omega_e * T_p) & 0 \\ -\sin(\omega_e * T_p) & \cos(\omega_e * T_p) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{tft} \\ y_{tft} \\ z_{tft} \end{bmatrix} \quad (3.22)$$

Once the signal propagation time is obtained, the Doppler can be calculated as:

$$\begin{aligned} v_d &= \frac{x_{tfr} - x_u}{r} \cdot (v_x - v_{xu}) + \frac{y_{tfr} - y_u}{r} \cdot (v_y - v_{yu}) + \frac{z_{tfr} - z_u}{r} \cdot (v_z - v_{zu}) \\ r &= \sqrt{(x_{tfr} - x_u)^2 + (y_{tfr} - y_u)^2 + (z_{tfr} - z_u)^2} \\ f_d &= -f_{L1} \cdot \frac{v_d}{c} \end{aligned} \quad (3.23)$$

3.2.6 Satellite Clock Error Modeling

The satellite clock correction δt_{SV} can be obtained using coefficients broadcast from the satellite after being uploaded by the GPS Control Segment. The data sets are then transmitted in the downlink data stream to users together with ephemeris in a 2 hour interval. Sub-frame 1 (words 8, 9 and 10) contains the data needed by users to perform corrections of the satellite clocks. These corrections represent a second order polynomial

in time. Specifically, four clock correction parameters, t_{oc} , a_{f2} , a_{f1} , a_{f0} , are provided in the ephemeris, and the correction term is approximated by a polynomial

$$\delta t_{sv} = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + \Delta t_R \quad (3.24)$$

where

- a_{f0} corresponding to the phase error
- a_{f1} corresponding to the frequency error
- a_{f2} corresponding to rate of change of the frequency error
- Δt_R is the relativistic correction
- t_{oc} is the reference time for clock correction

The relativistic correction can be computed by Equation (3.25) (Spilker, 1996b). \sqrt{A} can be obtained from the ephemeris sub-frame 2, and E_k can be calculated from the ephemeris parameters.

$$\Delta t_R = F_e \sqrt{A} \sin E_k \quad (3.25)$$

where

- F_e $-4.442807633 \times 10^{-10} \text{ s} / \text{m}^{1/2}$
- \sqrt{A} is the square root of the semimajor axis
- E_k is the eccentric anomaly of satellite orbit

3.2.7 Ionospheric Error Modeling

There are many methods for ionosphere delay modeling. In this research, the objective is to develop an ionospheric model which is valid for the simulation of the time delay along the observation line of sight. A combined Spherical Harmonics (SPHA) and grid model is

used in this research, which was originally developed by Luo (2001). It is described as follows.

The parameter of the ionosphere that produces most of the effects on GPS signals is the Total Electron Content (TEC). Once the TEC is available, the ionospheric group delay can be calculated as:

$$\delta t_{iono} = \frac{40.3 \times TEC}{c f^2} \quad (3.26)$$

where

c is the speed of light

f is the frequency of carrier

The computation of the vertical TEC involves three steps:

- Generating a global profile of the TEC distribution.
- Generating a global grid-network with additional random TEC.
- Computing the vertical TEC at any pierce point.

Among the global ionospheric models in existence, the spherical harmonics (SPHA) model is selected to simulate the ionospheric error. The SPHA model is based on the assumption that the ionosphere is a thin single layer. This means that all free electrons are distributed in a spherical shell of infinitesimal thickness. Figure 3.9 shows the concept of this model.

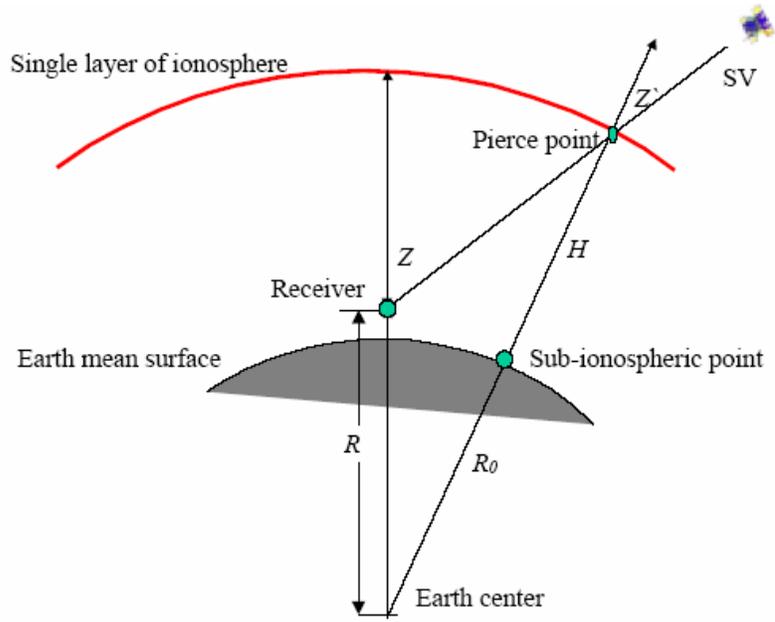


Figure 3.9: Description of the single-layer ionosphere model

In this model, the TEC is developed into a series of spherical harmonics in a Sun-fixed reference frame (Luo, 2001):

$$E(\beta, s) = SF \sum_{n=0}^{n_{\max}} \sum_{m=0}^n \tilde{P}_{nm}(\sin \beta) (a_{nm} \cos ms + b_{nm} \sin ms) \quad (3.27)$$

where

E is the vertical TEC value

β is the geocentric latitude of the pierce point of the ionosphere

s is the Sun-fixed longitude of the ionospheric pierce point or sub-ionospheric point, see Figure 3.9. It is also the difference between the Earth-fixed longitude of the pierce point and the longitude of the Sun

n_{\max} is the maximum degree of the spherical harmonics expansion

$\tilde{P}_{nm} = \Lambda_{nm} P_{nm}$ are the normalized associated Legendre functions of degree n and

order m based on the normalization factor Λ_{nm} and the classical Legendre functions P_{nm}

$a_{nm}b_{nm}$ are the coefficients of the spherical harmonics

SF is the scale factor to control the level of TEC value in simulation

The conversion from a vertical TEC value, E , into the slant TEC value, E_s , is realized by multiplying E with a mapping function:

$$E_s = m(z)E = \frac{1}{\cos z'} E \quad \text{with} \quad \sin z' = \frac{R}{R_0 + H} \sin z \quad (3.28)$$

where

$m(z) = \frac{1}{\cos z'}$ is the single-layer mapping function

z, z' are the geocentric zenith distances of a satellite at the height of the GPS receiver and the single layer of the ionosphere, respectively

R is the radius with respect to the receiver considered

R_0 is the mean radius of the Earth, approximated as 6371 km

H is the height of the single-layer of ionosphere above the Earth mean surface, which is set to 450 km.

To use the SPHA model, the coefficients $a_{nm}b_{nm}$ must be known. In this model, Global Ionosphere Maps (GIMs) files are used. These files contain the coefficients of spherical harmonics and other ionospheric parameters (Schaer, 1997). These GIMs can be obtained from the Centre for Orbit Determination in Europe (CODE), one of the Analysis Centres of the International GPS Service (IGS). Since January 1, 1996, GIMs have been routinely

presented as an additional product at CODE. Every day, a set of TEC coefficients is determined. These coefficients give an approximation of the distribution of the vertical TEC on a global scale by analyzing the so-called geometry-free linear combination of GPS carrier phase data collected at 84 globally distributed stations.

By the time of the model was developed, the GIMs could give a spherical harmonics expansion with a maximum degree of twelve and a maximum order of eight. To conduct the simulation in a small area, the resolution of the TEC distribution of the ionosphere should be greatly increased. An approach to achieve this objective is to add a high-resolution TEC increment grid-network to the profile of the global TEC distribution. The TEC increment can be treated as an unmodeled part of the ionospheric delay by the SPHA model. The following describes the procedures to generate a global TEC distribution with an enhanced resolution.

- Partition the ionosphere single-layer into a network with averagely distributed $n \times m$ grids in the Sun-fixed frame. The vertical TEC at a certain grid point (β_i, s_i) , $VTEC_{SPHA}(\beta_i, s_i)$, can be calculated using Equation (3.28). It should be noticed that, although the density of the grids in the network can be very high by increasing m and n , the actual resolution of the TEC distribution has not yet been increased, since the maximum degree and order of the model are not changed.
- Generate the TEC increment $v_{i,j}$ for each grid point. $v_{i,j}$ is a random variable representing the un-modeled part of the ionospheric delay. The statistical

characteristics of $v_{i,j}$ can be described by the following distribution (Christie et al., 1999):

$$f(x) = 1/(\sqrt{2}\sigma) \exp(-\sqrt{2}|x - m|/\sigma) \quad (3.29)$$

- Build the high-resolution grid network by adding the profile of the TEC distribution and the TEC increment. The vertical TEC at a grid point can be computed as follows:

$$VTEC(\beta_i, s_i) = VTEC_{SPHA}(\beta_i, s_i) + v_{i,j} \quad (3.30)$$

By introducing the TEC increment, the grid-network can represent the TEC distribution of the ionosphere with an enhanced resolution of $360^\circ / m \times 180^\circ / n$.

The resolution of the grid network is $1.5^\circ \times 1.5^\circ$.

After building the grid network of the ionosphere with enhanced spatial resolution, the vertical TEC value at the pierce point can be computed in two ways. First, the SPHA model can be expanded to a higher degree and order.

In the first two steps, a high-density grid network of TEC distribution is established. The TEC value at any point within the network can be computed using interpolation. Figure 3.10 gives the concept of the four-point grid-based algorithm to estimate the vertical TEC at the pierce point where P is the pierce point and G_i is the surrounding grid point.

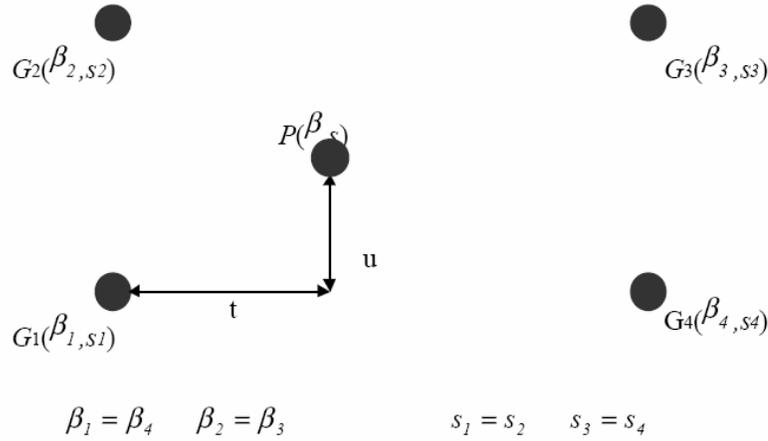


Figure 3.10: Compute TEC in a grid network

The vertical TEC at P can be computed using the following equation:

$$VTEC_P = \sum_{i=1}^4 w_i VTEC_i \quad (3.31)$$

where

$$\begin{aligned}
 w_1 &= w(t, u) & w_2 &= w(t, 1-u) \\
 w_{31} &= w(1-t, 1-u) & w_4 &= w(1-t, u) \\
 w(t, u) &= (1-t)(1-u) \\
 u &= (\beta - \beta_1) / (\beta_2 - \beta_1) & t &= (s - s_1) / (s_4 - s_1)
 \end{aligned}$$

To obtain the geomagnetic latitude, β , and longitude, s , the geographical latitude and longitude of the pierce point and the Sun must first be calculated; then they can be converted to the geomagnetic frame using a rotational transformation. The rotation angles are relevant to the position of the north geomagnetic pole, which is set to (79.45° N, 71.71° W). The choice of the spatial weighting function $w(\bullet)$ is arbitrary. The algorithms may be based on inverse distance weighted averaging, bilinear interpolation, or multi-

quadratic weighting functions. In the chosen model, the bilinear interpolation weighting function is adopted according to the specifications for users of the Wide Area Augmentation System (WAAS) (FAA, 1997; RTCA, 1998).

From the above modeling procedures, it can be seen that the characteristics of both the spherical harmonic model and the grid model are combined.

3.2.8 Tropospheric Error Modeling

There are two major delay effects due to the troposphere. The first and larger effect is the dry atmosphere excess delay, typically on the order of 2.3 m. The dry effect varies slowly with local temperature and atmospheric pressure in a reasonably predictable manner. The second effect caused by water vapour is generally smaller, 1-80 cm at zenith. Although approximately one-tenth the size of the dry effect, the wet delay varies markedly; 10-20% in a few hours, and is less predictable even with surface humidity measurements (Spilker, 1996c). A model based on the modified Hopfield model, which was originally developed by Luo (2001), is described in the following.

Hopfield (1969) has developed a two-quartic zenith model of the refraction index, with different quartics for the dry and wet atmospheric profiles. Black and Eisner (1984) have extended this zenith model to add the elevation angle mapping function.

The tropospheric delay is caused by the larger refractive index, $n(n > 1)$ of atmospheric gases than that of free space, $n = 1$, which causes the speed of light (group velocity) in

the medium to decrease below its free space value c . The increase of propagation time caused by the troposphere can be expressed as

$$\delta t_{tropo} = \int_{TransmissionPath} [n(s) - 1] ds = \Delta t_d + \Delta t_w \quad (3.32)$$

where n represents both the wet and dry terms. The refractivity is defined as $N=10^6(n-1)$.

The basic two-quartic model for the refractivity versus altitude h can be expressed as:

$$\begin{aligned} N_d &= N_{d0} (1 - h / h_d)^4 & h \leq h_d = 43 \text{ km} \\ N_w &= N_{w0} (1 - h / h_w)^4 & h \leq h_w = 12 \text{ km} \end{aligned} \quad (3.33)$$

where

N_{d0} is the dry refractivities at the surface of the Earth

N_{w0} is the wet refractivities at the surface of the Earth

h_d is the height of the top shell of the “dry gas”

h_w is the maximum height of the wet atmosphere

The total zenith delay is then the sum of the integrated dry and wet delays along the vertical path:

$$\Delta = 10^{-6} \int_{h_{usr}}^{h_d} N_{d0} \left(1 - \frac{h}{h_d}\right)^4 dh + 10^{-6} \int_{h_{usr}}^{h_w} N_{w0} \left(1 - \frac{h}{h_w}\right)^4 dh \quad (3.34)$$

where

$$\Delta = \Delta_d + \Delta_w$$

$$\Delta_d = \begin{cases} \frac{10^{-6}}{5} N_{d0} (h_d - h_{usr}) & h_{usr} \leq h_d \\ 0 & h_{usr} > h_d \end{cases}$$

$$\Delta_w = \begin{cases} \frac{10^{-6}}{5} N_{w0} (h_w - h_{usr}) & h_{usr} \leq h_w \\ 0 & h_{usr} > h_w \end{cases}$$

where

h_{usr} is the altitude of the user antenna

N_{d0} and N_{w0} can be expressed by the surface meteorologic data, namely,

temperature, air pressure and humidity (Spilker, 1996c)

$$\begin{aligned} N_{d0} &= 77.604(P_d / T)Z_d^{-1} \\ N_{w0} &= (e / TZ_w)(64.79 + 377600 / T) \end{aligned} \quad (3.35)$$

where

T_c is the temperature in Kelvin

P_d is the dry air pressure in millibars

e is the partial pressure of the water vapour in millibars

R_h is the relative humidity, which can vary from 0.0 to 1.0

Z_d , Z_w and e can be computed as follows:

$$Z_d^{-1} = 1 + P_d [57.97 \times 10^{-8} (1 + 0.52 / T) - 9.4611 \times 10^{-4} T_c / T^2] \quad (3.36)$$

$$Z_w^{-1} = 1 + 1650(e / T^3) [1 - 0.01317T_c + 1.75 \times 10^{-4} T_c^2 + 1.44 \times 10^{-6} T_c^3]$$

$$e = 6.108 \cdot \exp\left(\frac{17.15T - 4684}{T - 38.45}\right)$$

Black and Eisner (1984) built a mapping function which can describe the slant tropospheric delay as a function of both the local elevation angle of a satellite and the surface temperature:

$$M(E, T) = 1 / \sqrt{1 - [\cos E / (1 + X_{dw} h_d / R_e)]^2} \quad (3.37)$$

For elevation angles in the range $7^\circ < E < 90^\circ$ and surface temperatures in the region $-30^\circ C < T < 40^\circ C$, the value of $X_{dw} h_d / R_e$ is in the range $0.00088 < X_{dw} h_d / R_e < 0.01$.

Because the temperature dependence is small enough to be ignored, the above mapping function can be simplified as follows:

$$M(E, T) \approx M(E) = 1.001 / \sqrt{(0.001)^2 + 0.002 + \sin^2 E} \quad (3.38)$$

Finally, the total slant delay of the troposphere can be denoted as:

$$\Delta_s = \Delta \cdot M(E) \quad (3.39)$$

The temporal variation of the tropospheric delay is simulated in the model developed herein. This variation is realized by adjusting the meteorological data with time. The diurnal variations of the temperature and relative humidity are simulated with the functions shown below:

$$T(t) = T_0 + \frac{T_v}{2} \cos\left(\frac{\pi}{12}(t - t_{T_{\max}})\right) \quad (3.40)$$

$$RH(t) = \begin{cases} RH_0 + \frac{RH_v}{2} \cos\left(\frac{\pi}{12}(t - t_{RH_{\max}})\right) & 0 \leq RH(t) \leq 1 \\ 0 & RH(t) < 0 \\ 1 & RH(t) > 1 \end{cases} \quad (3.41)$$

where

T_0 is the average temperature over a given day

T_v is the daily change of temperature

$t_{T_{\max}}$ is the epoch with the highest temperature

RH_0 is the average relative humidity

RH_v is the daily change of the relative humidity

$t_{RH \max}$ is the epoch with the maximum relative humidity

The time used in Equations (3.40) and (3.41) is local time with units of hours. These two functions are designed according to experimental results (Lachapelle, 1997).

3.2.9 Noise

The received noise is usually represented as white noise with constant spectral density. Its bandwidth is considered infinite at the receiving point, but is narrowed down by several filters during down-conversion.

In the simulation, the discrete noise vector at the output of the final band-pass filter is directly simulated, since it is impossible to generate, via MATLAB, a continuous white noise with an infinite bandwidth. This noise vector is band-limited white noise with the input sampling rate. A bandwidth of 2 MHz is used in the simulation. The reasons of the bandwidth is selected as 2 MHz are ninety percent C/A code signal power are in the 2 MHz main lobe and the bandwidth of the hardware front-end used to collect the real IF GPS signal in this thesis is also 2 MHz. The noise power is the product of the filter bandwidth and the noise density, another input parameter. Thus, the noise model can be described as follows:

$$noise = \sqrt{N_0 BW} n(t) \quad (3.42)$$

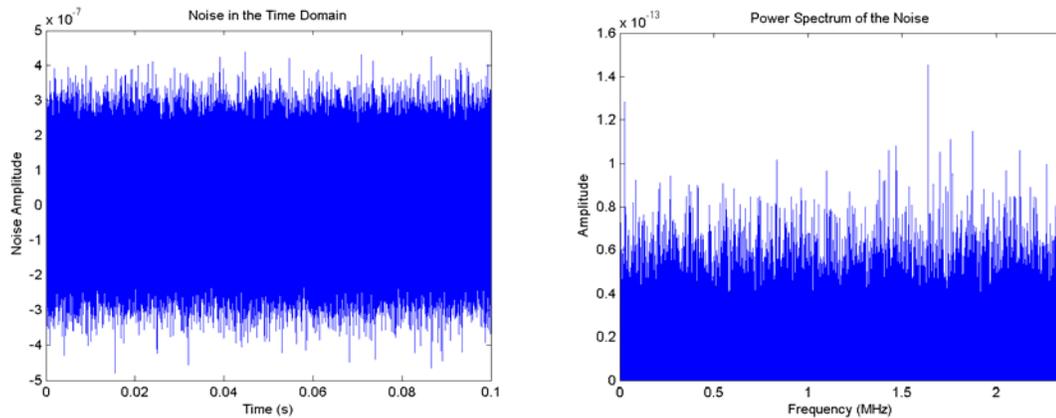
The power spectrum is the key quantity in the frequency domain signal analysis. Applied to GPS signals, it can demonstrate signal power level and signal bandwidth. Applied to

noise, it can make clear the background noise level, interference power level, and the interference type. Because of this, the concepts of frequency domain and power spectrum will be introduced first, and these concepts will be used throughout this thesis to show the signal properties in the frequency domain.

From stochastic process theory, the power spectrum of a signal and its autocorrelation function form a Fourier transform pair. Thus, the power spectrum can be determined from the following equation (Kaplan, 1996):

$$S(\omega) = \int_{-\infty}^{\infty} R(\tau) \exp(-j\omega\tau) d\tau = \int_{-\infty}^{\infty} R(\tau) \cos \omega\tau d\tau \quad (3.43)$$

In the above equation, the second relationship holds as well because the autocorrelation function is a real even function and the power spectrum is a real function. In MATLAB, there exist several functions that can perform spectral estimation based on a finite set of data.



(a) Noise in the time domain

(b) Noise in the frequency domain

Figure 3.11: Noise characteristics in the time and frequency domains

The simulated noise in the time domain and its power spectrum are shown in Figure 3.11. In the simulation, the noise density is -203 dBW/Hz, the filter bandwidth is 2 MHz, the sampling frequency is 4.75 MHz, and the IF is 15.42 MHz. The resultant base band frequency is 1.17 MHz.

3.2.10 Filtering

Signal filtering is completed with a MATLAB function, 'FIR1', which realizes a classical windowed linear-phase Finite Impulse Response (FIR) digital filter. This filter is actually a standard band-pass filter defined by the filter type, filter order, and pass band bandwidth. The magnitude and phase response of the band-pass filter used in the thesis are shown in the Figure 3.12.

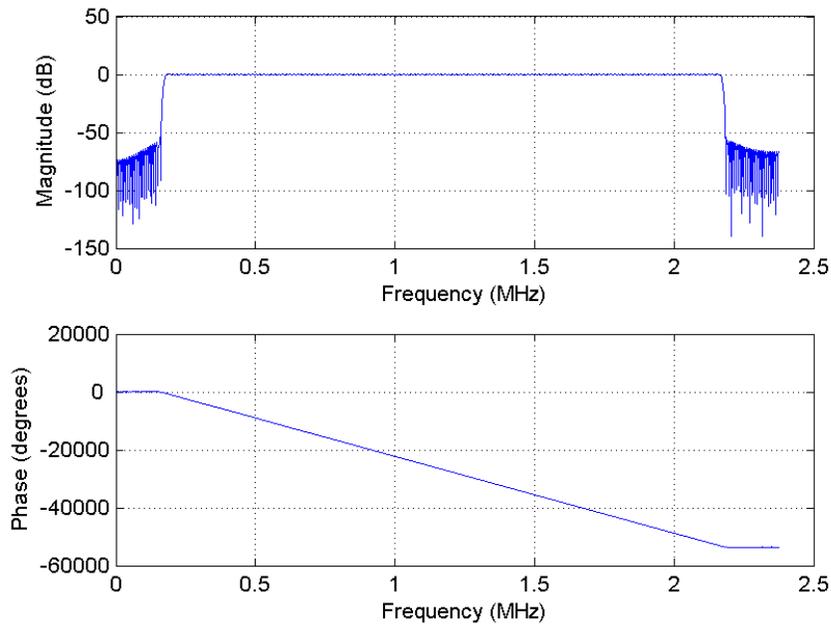


Figure 3.12: Band-pass filter magnitude and phase response

The filter order greatly affects the performance of the filter. The higher the filter order, the more accurate the filter, but the computation load is also higher. So, a tradeoff must be made. An empirically-derived value of 700 is used in the simulation. It produces a satisfactory response in the filter, as well as a tolerable computation time of about 1.6 seconds in the computer with a P4 CPU at a speed of 2.6 GHz.

The signals before filtering and after filtering in the time and frequency domains are shown in Figure 3.13. The two plots on the left-hand side show the results before filtering whereas the right-hand figures give the results after filtering. It is evident that the side lobe out of pass band is removed after the signal sequence passes the filter.

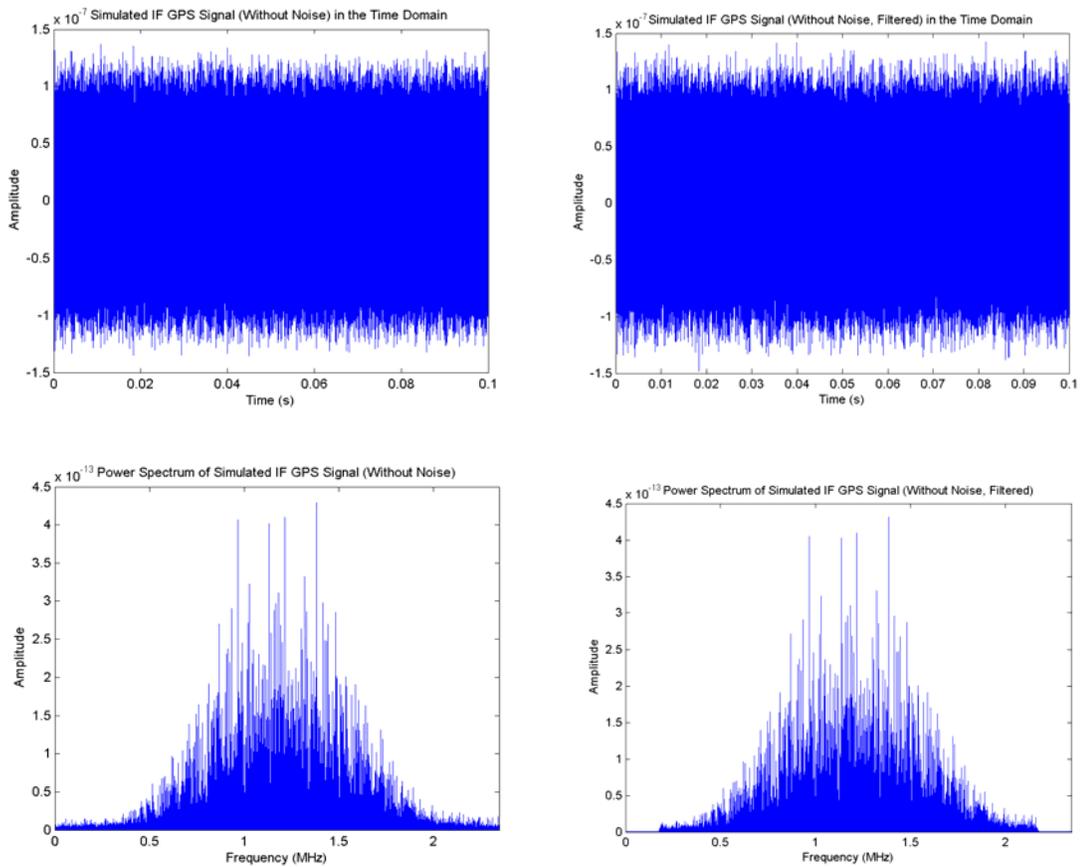


Figure 3.13: IF GPS signal before and after filtering

3.2.11 Quantization

The fewer the number of quantization bits, the easier the digital process becomes. However, the fewer the number of quantization bits, the greater the signal degradation caused by quantization. For most systems it is desirable to quantize as coarsely as the performance will permit (Krumvieda et al., 2001). One-bit and 2-bit quantization are used in many commercial GPS receivers. A rule of thumb states that 1-bit quantization degrades the signal-to-noise ratio by 1.96 dB and 2-bit quantization degrades the signal-to-noise ratio by 0.55 dB (Van Dierendonck, 1996). The implementation of 1-bit and 2-bit quantization is presented in the following paragraphs.

One-bit quantization is realized in the simulation as it is typically used in most commercial receivers and does not need an AGC. The implementation threshold is zero for 1-bit quantization. The resulting data value is 1 if the signal is greater than 0; and -1 if the signal is less than 0. The 1-bit quantized signals in the time and frequency domains are shown in the Figure 3.14.

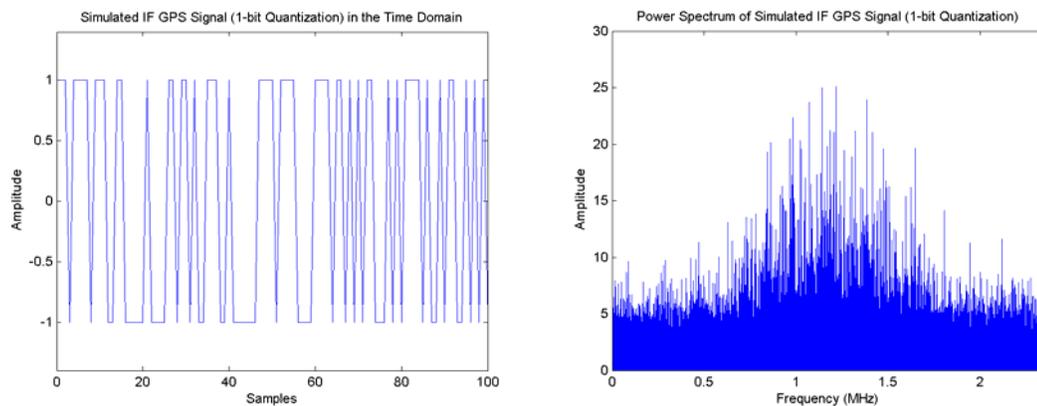


Figure 3.14: IF 1-bit quantization GPS signal in the time and frequency domains

Two-bit quantization usually results in four level magnitudes data samples by three thresholds. Any time multi-bit quantization is used, an A/D threshold control is required (Van Dierendonck, 1996). A feedback loop is necessary to perform AGC, which is a dynamic process. However, it is difficult to simulate this function without a dynamic simulation tool such as ‘Simulink’. Therefore, another method, developed by Amoroso (1983), is used to conduct 2-bit quantization for the software implementation. In this method, the high magnitude samples are given much more weight in the data detection process than the small magnitude samples to achieve good performance in terms of Gaussian noise. The general quantization strategy is illustrated in Figure 3.15. The sign threshold is indicated on the input axis as T_0 , and the positive and negative magnitude thresholds are indicated as $T_0 + \Delta$ and $T_0 - \Delta$, respectively. Samples falling below the magnitude thresholds are given unity weight, while samples exceeding the magnitude thresholds are given weight R .

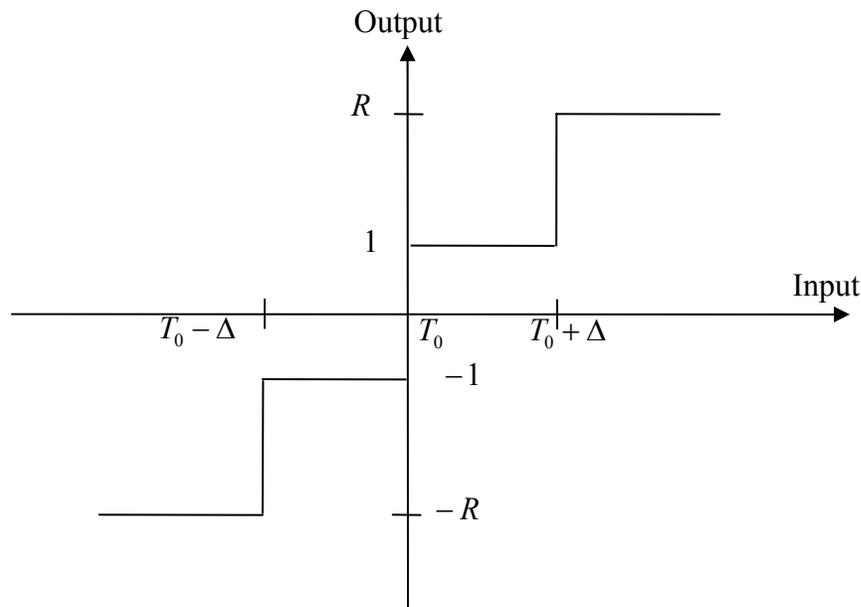


Figure 3.15: Two-bit quantization strategy

In the next step, one computational criterion is made for setting the Δ and R values. The received GPS signal is present amidst a background of Gaussian noise. The signal-to-noise ratio of the A/D conversion input signal is extremely low. A high A/D conversion gain is expected because the channel bit voltage will be the sum of a large number of quantized chip voltages and will therefore tend to be Gaussian distributed with high processing gain according to Amoroso (1983). Hence, the criterion is to set Δ and R to maximize the A/D conversion gain.

Assuming that the input signal is $\pm V_i$ and the Gaussian noise with zero mean has a standard deviation of σ_i , the A/D conversion gain, G , can be presented by the following equation:

$$\begin{aligned} \lim_{\frac{V_i}{\sigma_i} \rightarrow 0} G\left(R, \frac{\Delta}{\sigma_i}\right) &= \frac{\left[\sqrt{\frac{2}{\pi}} + 2\Phi\left(\frac{\Delta}{\sigma_i}\right)(R-1)\right]^2}{R^2\left[1 - \alpha\left(\frac{\Delta}{\sigma_i}\right)\right] + \alpha\left(\frac{\Delta}{\sigma_i}\right)} \\ &= G_0\left(R, \frac{\Delta}{\sigma_i}\right) \end{aligned} \quad (3.44)$$

where

$$\alpha(u) = \int_{-u}^u \Phi(y) dy, \quad \Phi(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}$$

The value of V_i/σ_i is typically very small in a system with high processing gain. G_0 is independent of the input signal level V_i as long as the level is small. Therefore, the A/D converter acts almost as a linear device which degrades the SNR by a ratio depending on Δ/σ_i and R . The conversion gain is plotted in Figure 3.16 with Δ/σ_i reinterpreted as the

percentage of input samples falling at a high magnitude, reflecting the adaptive threshold strategy. This percentage is essentially independent of the input signal level as long as

that level is low. It is defined as $100 \times \left[1 - \int_{\frac{\Delta}{\sigma_i}}^{\frac{\Delta}{\sigma_i}} e^{-y^2/2} dy \right]$.

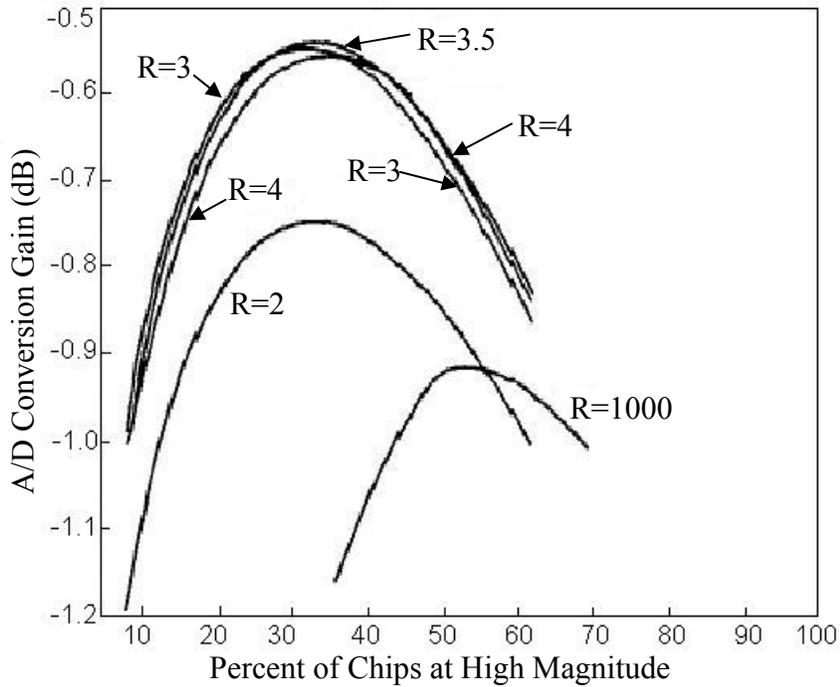


Figure 3.16: A/D conversion gain in Gaussian noise

Figure 3.16 shows that the joint optimum condition is substantially reached at $R = 3.5$ and 35 percent of chips having high magnitude. The nearly optimal state occurs when R is 3 or 4.

For expression simplicity, the 2-bit quantization in this thesis takes the values $R = 3$ and 30 percent of chips at high magnitude to determine the threshold. The sign threshold T_0 is set to 0 since the Gaussian noise is zero mean. The MATLAB quantization routine

iteratively determines the threshold values to meet 30 percent high magnitude sample distribution goals. The resultant values for the digitized signal are +3 when the signal is greater than the positive threshold; +1 when the signal is between zero and the positive threshold; -1 when the signal is between the negative threshold and zero; and -3 when the signal is less than the negative threshold. The resultant digitized IF GPS signal is shown in Figure 3.17.

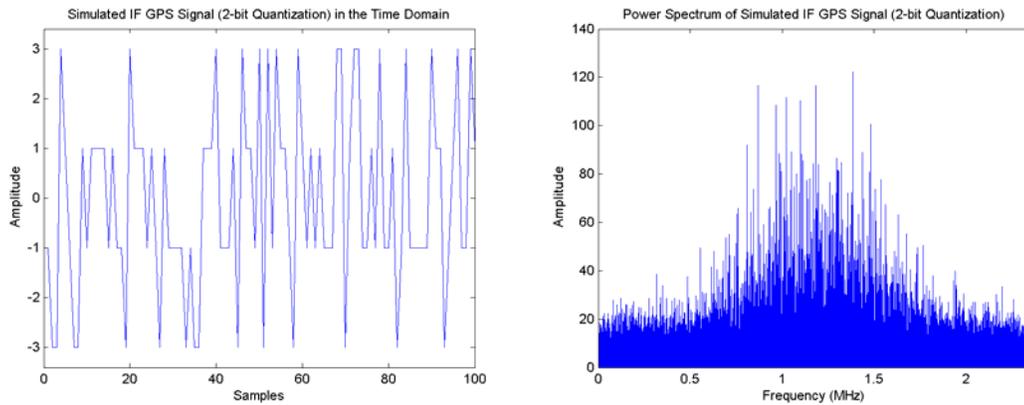


Figure 3.17: IF 2-bit quantization GPS signal in the time and frequency domains

3.2.12 Program Flow Chart

All of the above functions and models are realized by MATLAB functions. A block diagram of the code of the signal simulator, structured with respect to the hierarchy of functions, is shown in Figure 3.18. The order in which the functions are called is predominantly from left to right. A more detailed program flow chart is given in Figure 3.19.

In summary, the simulator has the capabilities of simulating the L1 C/A code and data modulation, ionospheric and tropospheric errors, noise, filtering and quantization. Any of

these functions can be turned on or off simply by a switch variable. The switch variables as well as most of the system parameters are considered in the simulation configuration function 'parameters'. To simulate a specific scenario, a user simply reconfigures the 'parameters' function as required before simulation. The parameters included in the 'parameters' function are described in Table 3.1.

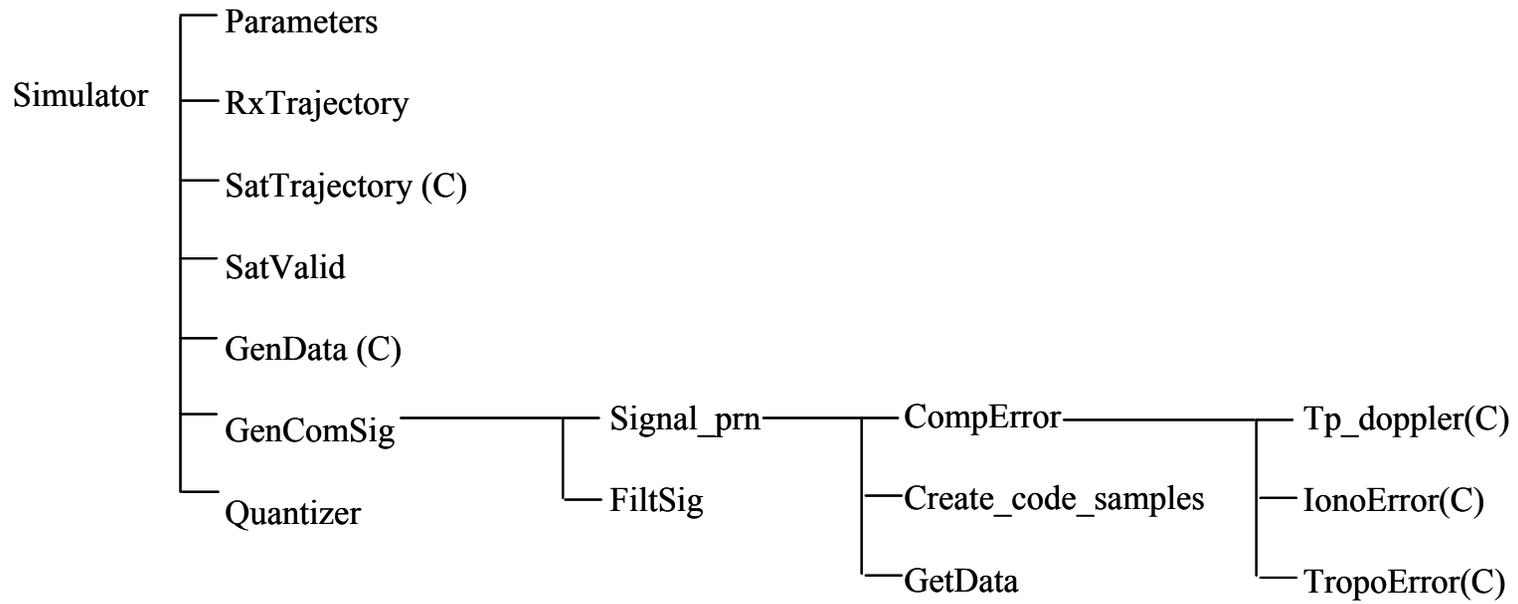


Figure 3.18: Signal simulator functions

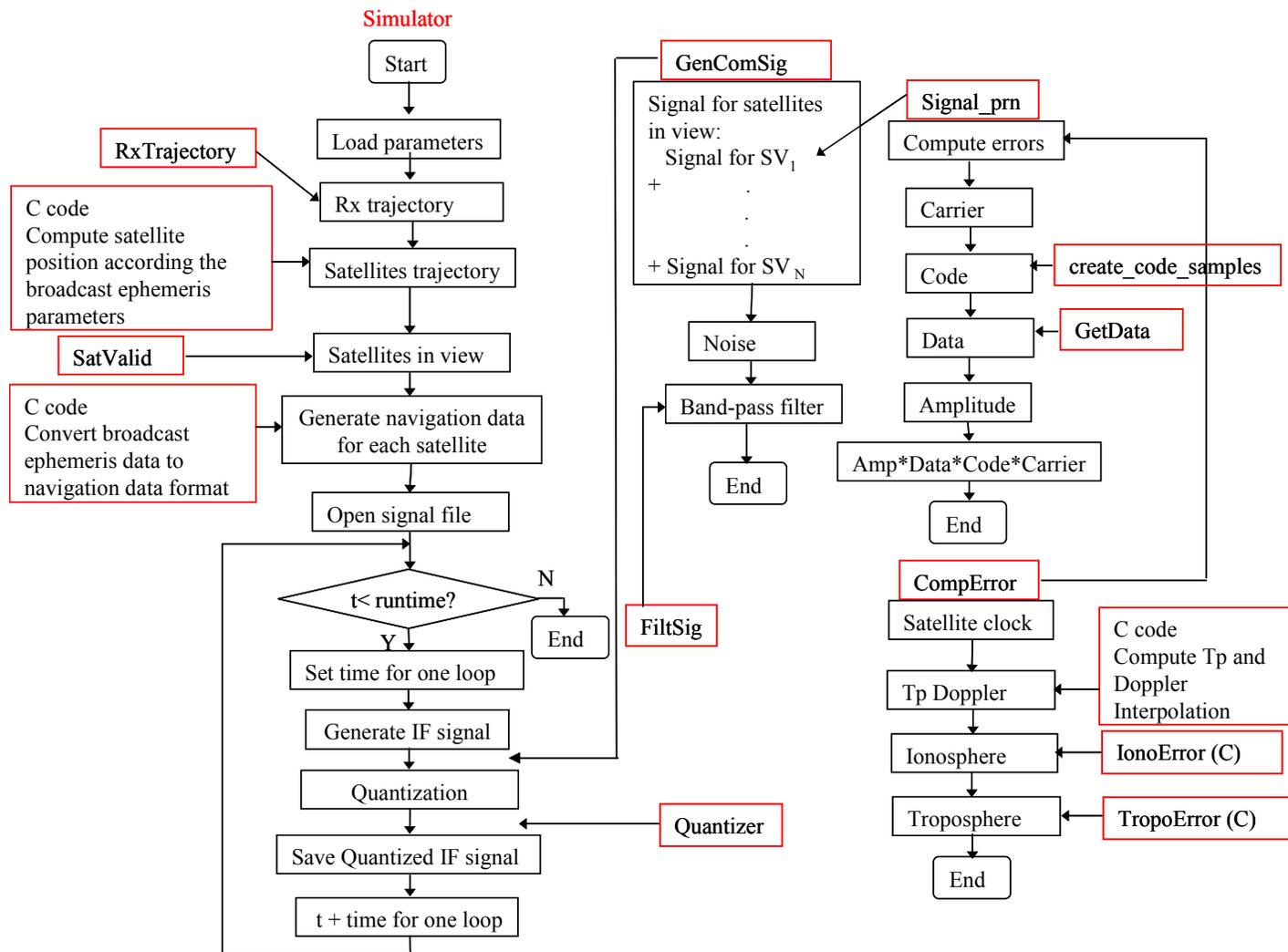


Figure 3.19: Program flow chart

Table 3.1: Simulator parameters

Parameter Name	Description
GPS System Constants:	
Frequency L1	GPS L1 frequency 1575.42 MHz
ChipRate	C/A code chipping rate 1.023 MHz
LightSpeed	Speed of light 299792458 m/s
Simulation Control Parameters:	
AddAmp	Simulate signal amplitude
AddData	Simulate navigation data
AddCode	Simulate C/A code
AddCarrier	Simulate Carrier
AddNoise	Simulate noise
ApplyDopplerTp	Simulate Doppler and propagation time
ApplyIono	Simulate ionosphere error
ApplyTropo	Simulate troposphere error
ApplyBandFilt	Conduct signal filtering
QuantSig	Conduct quantization
Simulation System Parameters:	
StartTime	Simulation start time in GPS time (s)
EndTime	Simulation end time in GPS time (s)
Loop_time	Simulation time of each loop
RxMode	Receiver trajectory mode (0, 1, 2)
TrajTD	Receiver trajectory time interval
ElevCutoff	The cutoff elevation angle
NoiseDensity	Expected front-end noise density
C/N ₀	Expected C/N ₀
FE_BW	Band-pass filter bandwidth
FE_filter_order	Filter order
QuantBit	Quantization bit 1 or 2

CHAPTER 4

SOFTWARE GPS RECEIVER

4.1 Introduction

A software GPS receiver becomes the advisable option if the validity of the signal simulator needs to be verified. Partially to this end, a post-mission multiple channel GPS receiver was developed with the functional diagram shown in Figure 4.1. Just like a usual GPS receiver, this software receiver is composed of signal acquisition, signal tracking, sub-frame synchronization, and navigation processing components. To enable successful operation, a few assisting modules are also developed; for example, a measurement derivation module to obtain pseudorange, carrier phase and Doppler from each channel; an inner clock generation module to maintain an inner GPS timer; and a receiver monitoring module to manage receiver channels.

The simulated signal samples are saved in an input data file. After reading these samples into memory, the acquisition stage is engaged first to provide the initial code offset and Doppler estimates. With these initial values, a tracking loop is able to keep track of the simulated signals and to provide pseudorange and carrier phase estimates. Sub-frame synchronization is used to demodulate the navigation message, to obtain satellite

ephemeris, and to remove the transmit time ambiguity. Finally, the user position is computed and the inner clock bias is removed to improve the quality of the pseudorange measurements in the navigation solution stage.

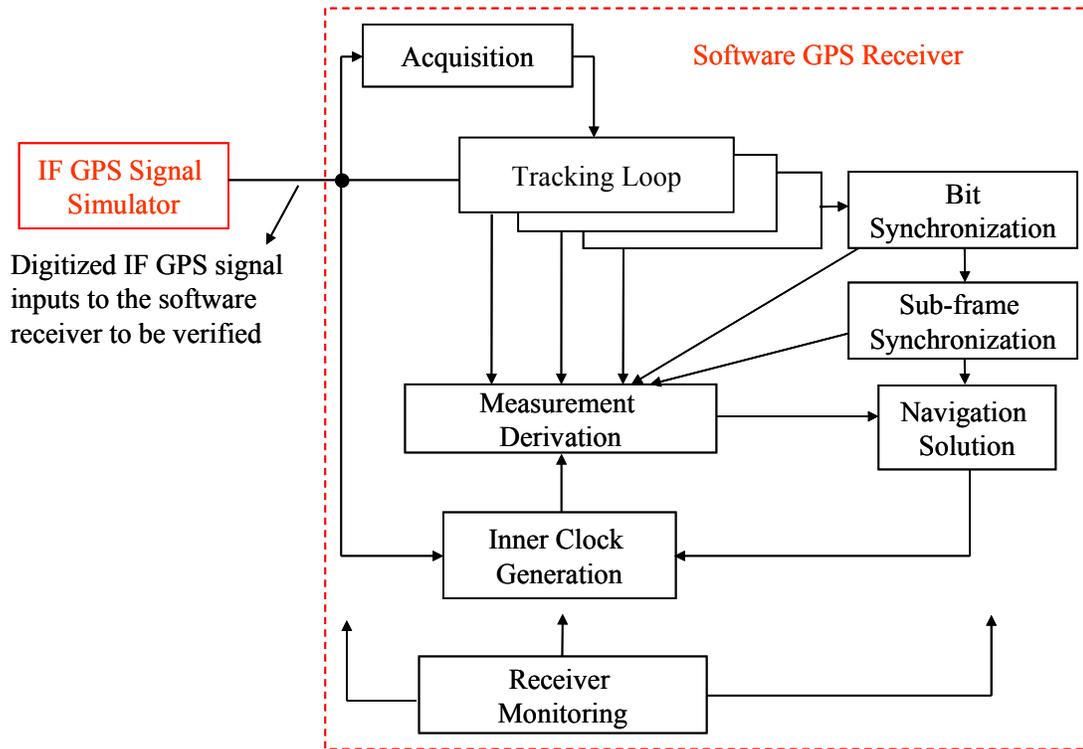


Figure 4.1: Structure of the software GPS receiver

4.2 Signal Acquisition

Acquisition is a coarse synchronization process giving rough estimates of the PRN code offset and the carrier Doppler which initialize tracking loops for signal tracking and navigation data demodulation (Lin and Tsui, 2000). As illustrated in Figure 4.2, GPS signal acquisition is basically a two-dimensional search process in which a replica code and a replica carrier are aligned with the received signal. The correct alignment is identified by measurement of the output power of the correlators. In other words, when both the code and the carrier Doppler match the incident signal or simulated signal, the

signal can be despread and the data can be demodulated. The results of the two-dimensional search consist of an estimate of the code offset to within half a chip and a Doppler estimate to within the lock range of the tracking loops.

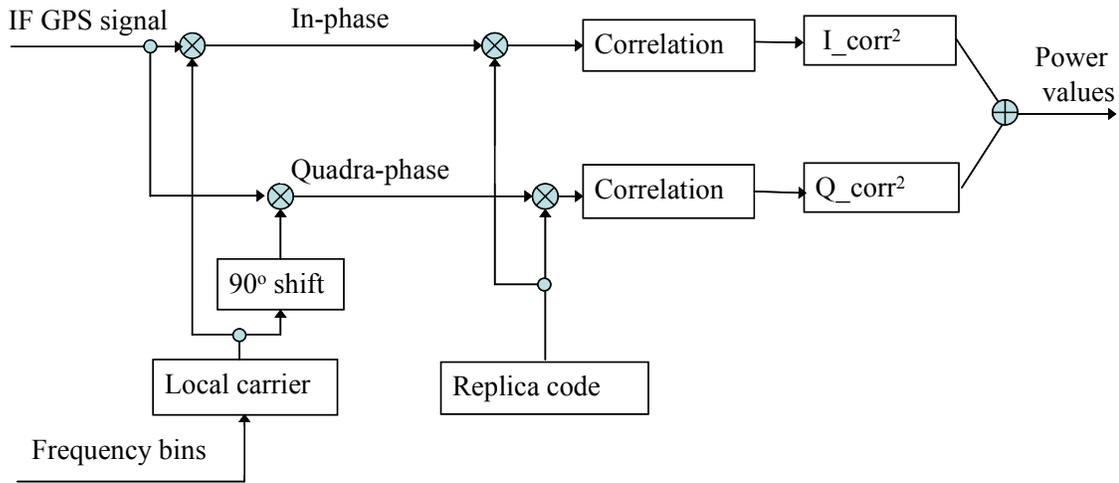


Figure 4.2: Signal acquisition

The two-dimensional search space covering the full range of the uncertainty of code and Doppler needs to be defined before acquisition can be performed. It is difficult to predetermine the code offset because this offset is a function of the starting point and changes at the chipping rate with time. Thus, the code search space typically includes all possible code offset values. The resolution of the code search needs to be smaller than half a chip; usually, the sampling frequency is used to specify this resolution. By comparison, the change of Doppler is a function of user dynamics and the stability of the receiver oscillator which is normally a low frequency quantity and easily estimated. Therefore, the frequency searching space can be reduced if an initial estimate of Doppler is known in advance. Without such an initial estimate, the Doppler searching space is usually from -5 kHz to 5 kHz. The frequency resolution is determined by the coherent

integration time (or dwell time). The relationship is roughly $\Delta f = (2/3)T$ (Raquet, 2001) where Δf is the frequency bin width in Hz and T is the predetection integration time in seconds. There is a tradeoff between the predetection integration time and acquisition speed. Longer integration can provide better frequency resolution and higher sensitivity, but it needs to search a greater number of bins and requires more time.

Various acquisition methods with search and detect strategies have been proposed in the literature (Krumvieda et al., 2001; Kaplan, 1996). In a conventional receiver, acquisition is performed using a carrier and a C/A-code replica. It resides physically on each bin for a predetermined dwell time and, thus, the acquisition time is the product of the dwell time and the number of search bins. Consequently, the acquisition time is very long. In a post-mission software GPS receiver, the samples of the incident signal are first read into a large buffer, making block data processing possible. A popular block signal acquisition technique is the Discrete Fourier Transform (DFT) based technique which is able to search all possible code offsets in one DFT-based computation, dramatically reducing the computational burden (Tsui, 2000).

4.2.1 DFT-based Acquisition

Figure 4.3 demonstrates the concept of the DFT-based acquisition method. The local data is a combination of the carrier and the C/A-code. The circular convolution gives the acquisition results of all possible code offsets at a specific carrier frequency.

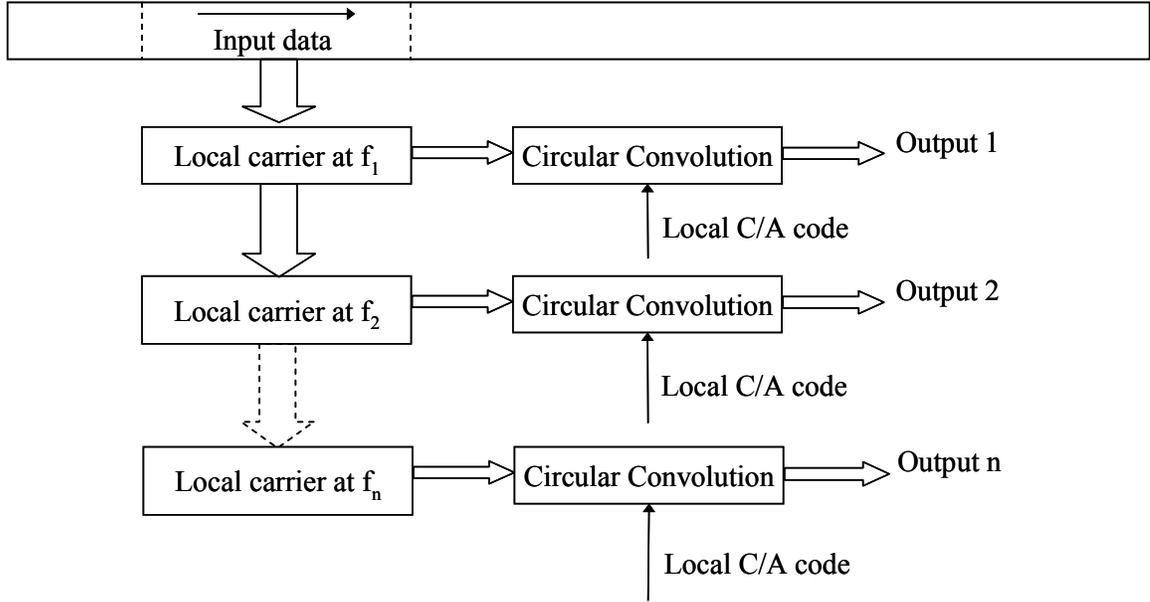


Figure 4.3: DFT-Based acquisition

Let the input signal be expressed as

$$s = AC \cos(2\pi ft), \quad (4.1)$$

and the local carrier at f_i be expressed as

$$l_i = \exp(j2\pi f_i t) \quad (4.2)$$

Thus, their product is

$$\begin{aligned} p_i &= \frac{AC}{2} \exp(j2\pi(f + f_i)t) + \frac{AC}{2} \exp(j2\pi(f_i - f)t) \\ &= \frac{AC}{2} \exp(j2\pi(f + f_i)t) + \frac{AC}{2} \exp(j2\pi\Delta f t) \\ &= p_i^H + p_i^L \end{aligned} \quad (4.3)$$

The correlation result z_i between p_i and the local C/A-code, C_{CA} , gives the final acquisition result. Since p_i contains both high and low frequency components, z_i can be expressed as the summation of the two parts

$$z_i = z_i^H + z_i^L \quad (4.4)$$

where

z_i^H is the correlation between p_i^H and C_{CA} , and

z_i^L is the correlation between p_i^L and C_{CA}

Written in component form, z_i^L is equal to

$$\begin{aligned} z_i^L &= \sum_{n=0}^{N-1} AC(n) \exp(j2\pi\Delta f T n) C_{CA}(n+m) \\ &= \begin{cases} A \cdot \left(\frac{1 - \exp(j2\pi\Delta f T N)}{1 - \exp(j2\pi\Delta f T)} \right) & m = 0 \\ \text{sufficiently small} & m \neq 0 \end{cases} \\ &= \begin{cases} A \cdot N \cdot \exp(j\pi\Delta f T (N-1)) \left(\frac{Sa(\pi\Delta f T N)}{Sa(\pi\Delta f T)} \right) & m = 0 \\ \text{sufficiently small} & m \neq 0 \end{cases} \end{aligned} \quad (4.5)$$

Obviously, z_i^L is significantly large only when the incident code is synchronized with the local code and when the local carrier frequency is close to the incident carrier. z_i^H is always small because it is the integration of a higher frequency component.

Such correlation can be computed by circular convolution because of the following property (Van Vee and Coenen, 1991): the correlation between two periodic sequences $x(n)$ and $h(n)$,

$$z(n) = \sum_{m=0}^N x(m)h(n+m) \quad (4.6)$$

equals

$$z(n) = IFFT\left(DFT(x(n)) \cdot \overline{DFT(h(n))}\right), \quad (4.7)$$

where $\overline{(\cdot)}$ denotes a complex conjugate. This method can give the correlation value at all possible code offsets in one computation. Fast acquisition speed can be achieved if the FFT technique is applied, a routine that greatly reduces the computational burden. For this reason, a DFT-based acquisition method is used in this software GPS receiver.

4.3 Signal Tracking

Acquisition produces a coarse estimate of the carrier Doppler and the code offset of the incoming signals. Control is then handed over to the tracking loops, the function of which is to track variations in the carrier Doppler and code offset due to line-of-sight dynamics between satellites and the receiver. Together with bit and sub-frame synchronization, precise pseudorange and carrier phase measurements can be derived. Another important function of the tracking loop is to demodulate the navigation data to obtain the ephemeris from which satellites positions can be calculated.

To track an incoming GPS signal, both the carrier phase and code offset need to be matched by the locally generated carrier and code. Thus, the lock status of both the carrier lock loop (FLL or PLL) and delay lock loop are required for signal tracking; they must be coupled together as shown in Figure 4.4 (Tsui, 2000). Once one of them loses lock, the other one will subsequently lose lock as well. Before discussing the delay lock loop and the carrier lock loop implemented in the developed software GPS receiver, the basic concept of a phase lock loop is first described in the following section.

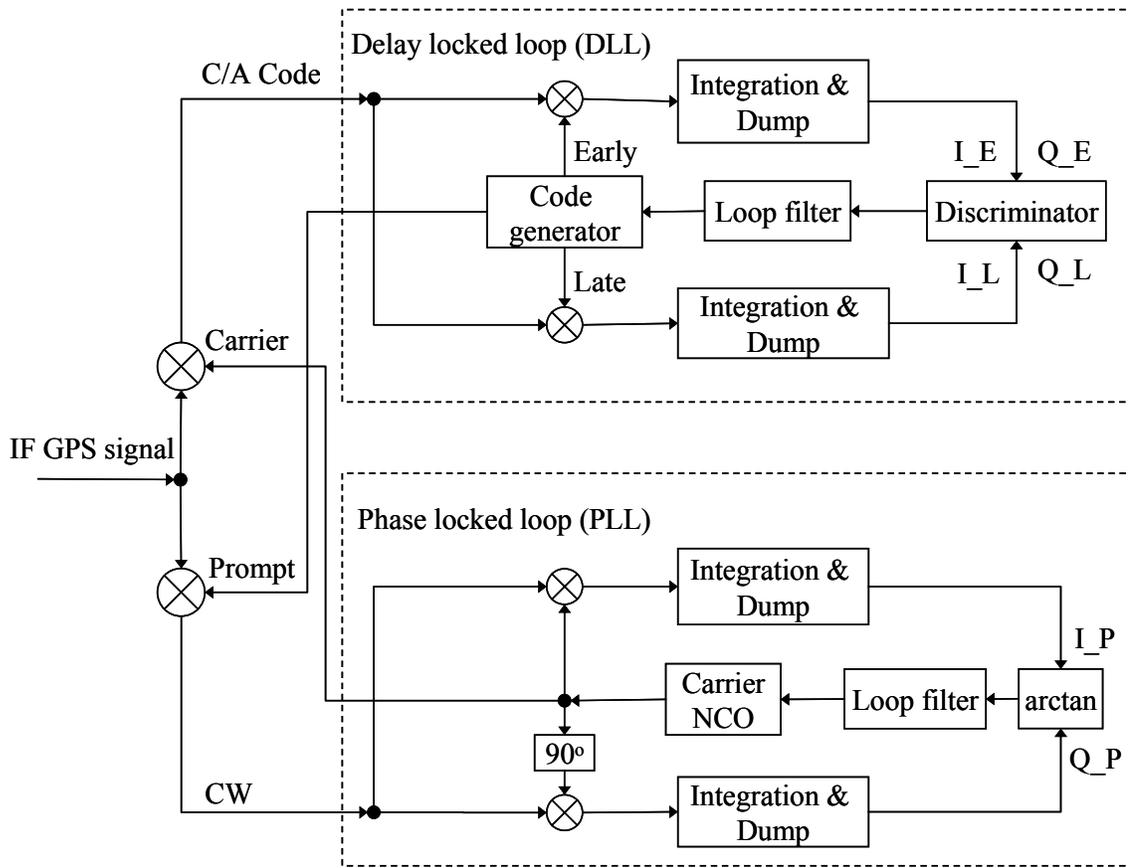
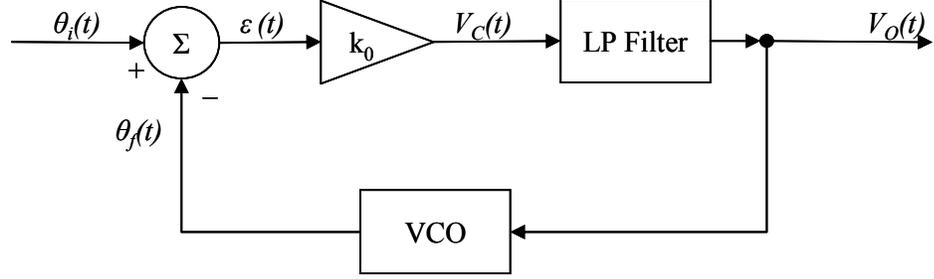


Figure 4.4: Coupled code and carrier tracking loops

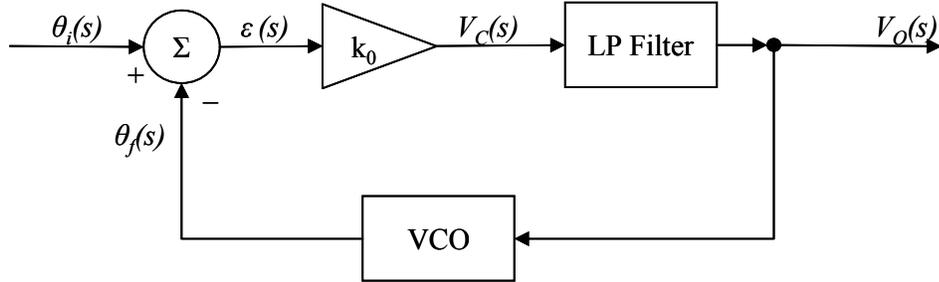
4.3.1 Basic Phase Lock Loop

In this section, the basic concept of the phase lock loop is described, which includes the transfer function, the error transfer function, and the noise bandwidth.

The main purpose of a PLL is to adjust the frequency or phase of a local oscillator to match the frequency or phase of an input signal, referred to as the reference signal. The basic phase lock loop is shown in Figure 4.5.



(a) Time domain



(b) S domain

Figure 4.5: Basic phase lock loop

Figure 4.5(a) shows the time domain configuration and Figure 4.5(b) shows the S domain configuration, which are related by the Laplace transform. The input signal is $\theta_i(t)$ and the output from the voltage controlled oscillator (VCO) is $\theta_f(t)$. The phase comparator measures the phase difference between these two signals. The amplifier, k_0 , represents the gain of the phase comparator and the low-pass filter limits the noise in the loop. The input voltage V_0 to the VCO controls its output frequency, which can be expressed as

$$\omega_2(t) = \omega_0 + k_1 V_0(t) \quad (4.8)$$

where ω_0 is the centre angular frequency of the VCO and k_1 is the gain of the VCO. The phase angle of the VCO can be obtained by integrating Equation (4.8) as

$$\int_0^t \omega_2(t) dt = \omega_0 t + \int_0^t k_1 V_0(t) dt \quad (4.9)$$

where

$$\theta_f(t) = \int_0^t k_1 V_0(t) dt \quad (4.10)$$

The Laplace transform of $\theta_f(t)$ is

$$\theta_f(s) = V_0(s) \frac{k_1}{s} \quad (4.11)$$

From Figure 4.5b, the following equations can be written.

$$V_c(s) = k_0 \varepsilon(s) = k_0 [\theta_i(s) - \theta_f(s)] \quad (4.12)$$

$$V_o(s) = V_c(s) F(s) \quad (4.13)$$

From these three equations, one can obtain

$$\varepsilon(s) = \theta_i(s) - \theta_f(s) = \frac{V_c(s)}{k_0} = \frac{V_o(s)}{k_0 F(s)} = \frac{s \theta_f(s)}{k_0 k_1 F(s)} \quad (4.14)$$

or

$$\theta_i(s) = \theta_f(s) \left(1 + \frac{s}{k_0 k_1 F(s)} \right) \quad (4.15)$$

where $\varepsilon(s)$ is the error function

The transfer function $H(s)$ of the loop is defined as

$$H(s) = \frac{\theta_f(s)}{\theta_i(s)} = \frac{k_0 k_1 F(s)}{s + k_0 k_1 F(s)}. \quad (4.16)$$

The error transfer function is defined as

$$H_e(s) = \frac{\varepsilon(s)}{\theta_i(s)} = \frac{\theta_i(s) - \theta_f(s)}{\theta_i(s)} = 1 - H(s) = \frac{s}{s + k_0 k_1 F(s)} \quad (4.17)$$

The equivalent noise bandwidth is defined as

$$B_n = \int_0^{\infty} |H(j\omega)|^2 df \quad (4.18)$$

where ω is the angular frequency and is related to the frequency f by

$$\omega = 2\pi f \quad (4.19)$$

It is evident from the above discussion that the loop filter, $F(s)$, is the key component of a PLL. It determines the loop order, natural frequency, damping coefficient, and noise bandwidth, and thus greatly affects the overall performance. In the software receiver discussed below, a three order phase lock loop and a two order DLL are implemented by careful loop filter selection and parameter adjustment.

4.3.2 Delay Lock Loop

Inside a GPS receiver, a DLL is used to track the C/A code of incident signals. As in a regular phase lock loop, it consists of a code phase discriminator, a loop filter, and the C/A code NCO. As illustrated in Figure 4.6, the DLL realized in the software receiver is a second order phase lock loop which tolerates constant acceleration.

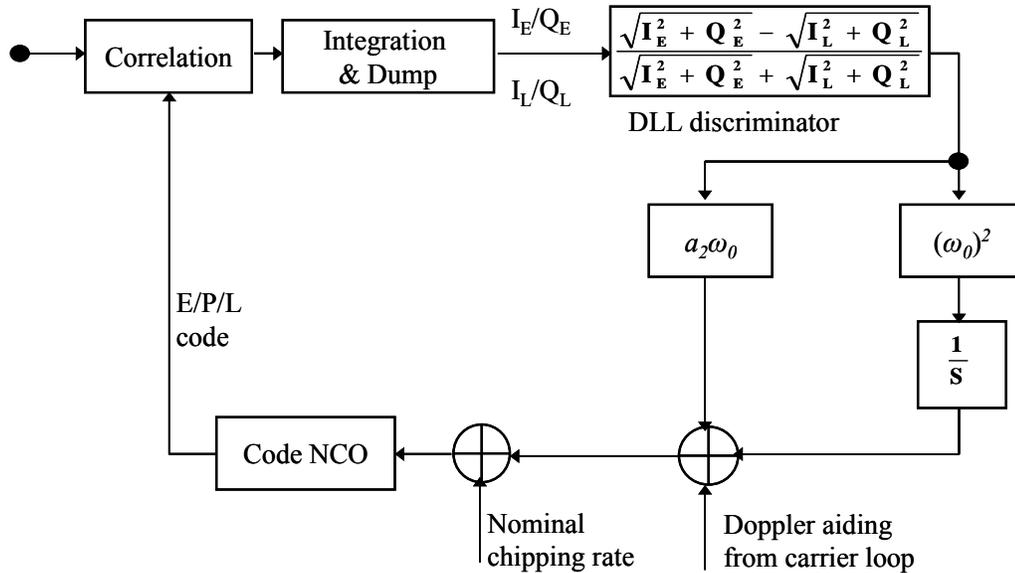


Figure 4.6: Delay lock loop

As mentioned before, the carrier and code need to be matched simultaneously by the locally generated signal in order to keep track of the incoming signal, thus after Doppler removal, the input of a DLL is nearly a pure C/A code (together with the navigation bits). The correlation process mixes this input with the local early (E), punctual (P), and late (L) code replicas in both the In-phase and Quadrature-phase arms to get the I_E , Q_E , I_P , Q_P , I_L , and Q_L . As shown in Figure 4.7, the correlation power or the correlation envelope of the early component and the late component contains the code mismatch information $\Delta\tau$; thus, the I_E , Q_E , I_L , and Q_L are usually used in the DLL discriminator to obtain the estimate of $\Delta\tau$. Several DLL discriminators have been proposed, and their input-output relationships are illustrated in Figure 4.8. A normalized early minus late envelope DLL discriminator is used in the software receiver because its tracking range is (-1.5 chips, 1.5 chips), which is much wider than that of other types of discriminators; its input-output

relationship is linear among $(-0.5 \text{ chips}, 0.5 \text{ chips})$, so no extra approximation will be introduced in the estimation of $\Delta\tau$.

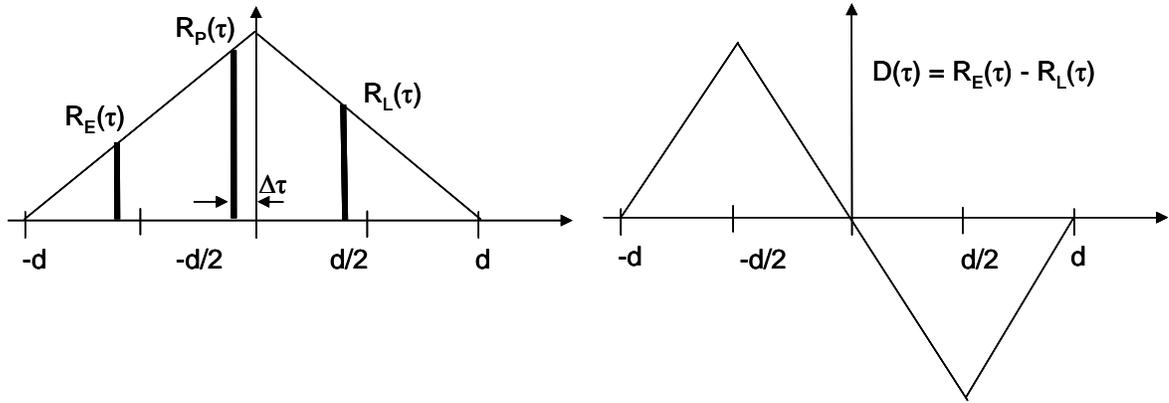


Figure 4.7: Code mismatch vs early, punctual and late components

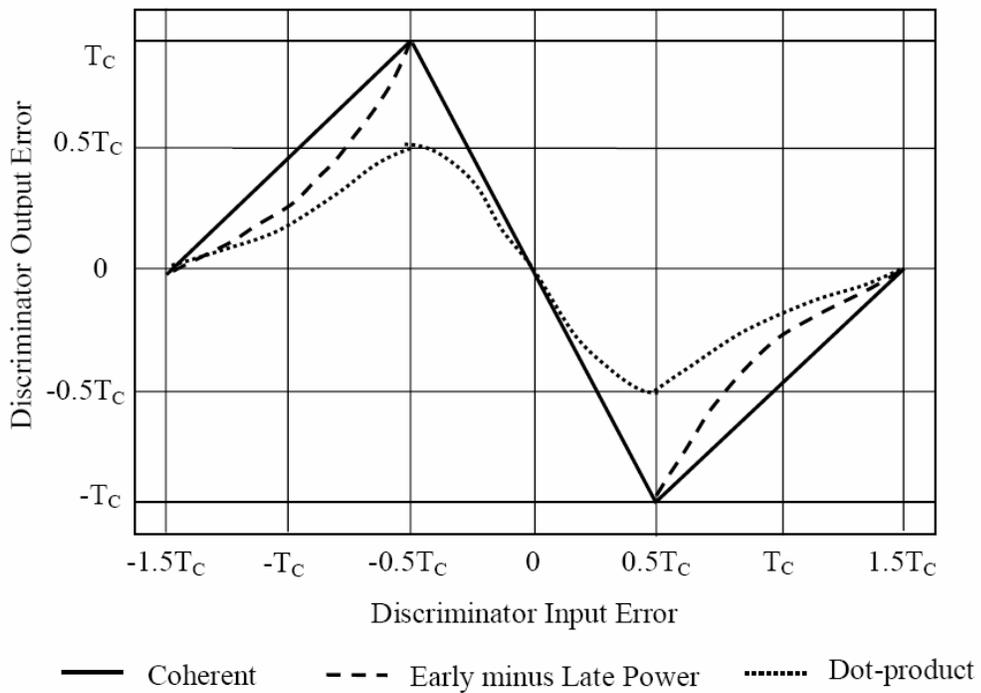


Figure 4.8: DLL discriminators

External aiding from the carrier loop is normally applied to improve DLL accuracy as shown in Figure 4.6. In this case, a DLL needs only to correct some errors, such as initial

tracking errors, the rate of change of the ionosphere or differences in code, and carrier multipath. These residual variations are ordinarily quite small and change very slowly with time; thus, the DLL loop bandwidth can be significantly reduced – approaching a magnitude on the order of 0.05 ~ 1 Hz, depending on the application. With external aiding from the carrier loop, the pre-detection time can be extended dramatically based on the Doppler aiding accuracy. Based upon the above analysis, a PLL aided DLL is realized and a 2 Hz bandwidth and a 10 ms coherent integration time are selected in the software receiver. It can achieve the satisfactory DLL tracking accuracy for the normal user dynamic situation with a speed of tens of kilometers per hour.

A first-order loop filter is used here to construct a second-order DLL loop which can track constant acceleration in an unbiased manner. The transfer function of the loop filter is

$$F(s) = a_2\omega_0 + \frac{\omega_0^2}{s} \quad (4.20)$$

The DLL transfer function can be derived as follows after substituting $F(s)$ into Equation (4.16) and assuming $k_0 = 1$ and $k_1 = 1$,

$$H(s) = \frac{a_2\omega_0s + \omega_0^2}{s^2 + a_2\omega_0s + \omega_0^2} \quad (4.21)$$

Obviously, the natural frequency of the loop is ω_0 , and the damping factor is $0.5a_2$. The DLL noise bandwidth, B_n , the damping factor $0.5a_2$, and the natural frequency are related via the following equation for a second-order loop

$$B_n = \frac{\omega_0(1 + a_2^2)}{4a_2} \quad (4.22)$$

To make the DLL a critically damped loop, a_2 needs to be equal to 1.414. Therefore, $B_n = 0.53\omega_0$ can be derived and all of the parameters related to designing a DLL are thus determined. The output of the loop filter is a variation of the code chipping rate. Together with the Doppler aiding from the carrier loop and the nominal chip rate, it can drive the code NCO properly to keep track of the code of the incoming signal and to provide an accurate pseudorange measurement.

4.3.3 Carrier Tracking Loop

There are two types of carrier tracking loops, FLL and PLL. An FLL keeps track of the frequency of the incoming signal by measuring the frequency mismatch between the local carrier and the incoming carrier, while a PLL keeps track of the phase of the incoming signal by measuring the phase mismatch between the local carrier and the incoming carrier. In general, a PLL is more accurate than an FLL, but it is more sensitive to dynamic stress. As a result, an FLL is normally used in high dynamic situations or in transition stages from signal acquisition to phase tracking where it is difficult to acquire the phase of the carrier because of large frequency uncertainties. An FLL is also less sensitive to data bit transition than a PLL (Van Dierendonck 1996). In the software receiver, a FLL is designed to keep working until fine phase tracking can be done and bit synchronization is achieved. After that, a PLL will take over the carrier tracking in order to provide accurate carrier phase and Doppler measurements.

The second-order FLL used in the software GPS receiver is illustrated by Figure 4.9.

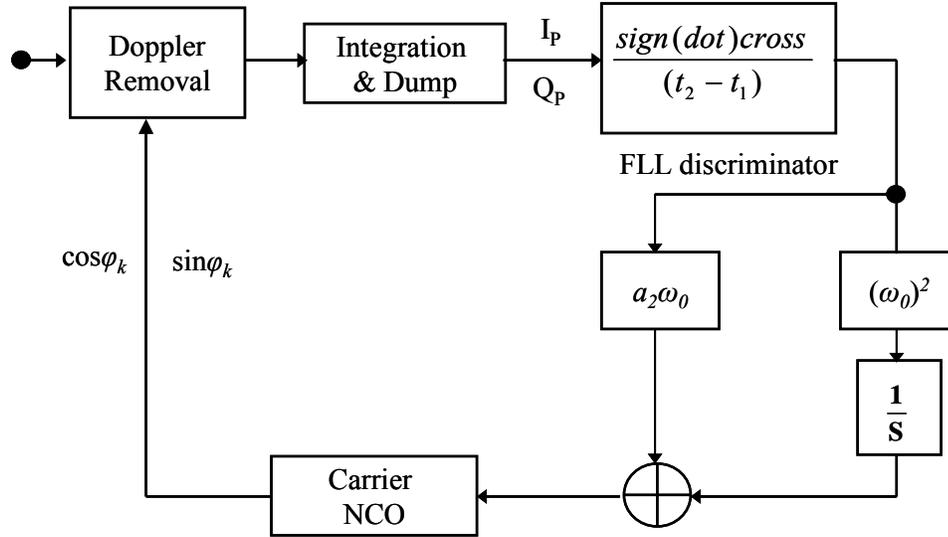


Figure 4.9: Frequency lock loop

The key characteristic of an FLL is its discriminator which detects frequency mismatch. In GPS receivers, frequency mismatch is usually estimated by investigating the phase rotation between two consequential (I_p, Q_p) pairs. There are several FLL discriminators (Kaplan, 1996), such as the cross-product discriminator,

$$\delta f_i = I_{i-1}Q_i - I_iQ_{i-1} \quad (4.23)$$

the decision-directed cross-product discriminator,

$$\delta f_i = (I_{i-1}Q_i - I_iQ_{i-1})\text{sign}(I_{i-1}I_i + Q_iQ_{i-1}) \quad (4.24)$$

and the differential arctangent discriminator,

$$\delta f_i = \arctan\left(\frac{Q_i}{I_i}\right) - \arctan\left(\frac{Q_{i-1}}{I_{i-1}}\right) \quad (4.25)$$

In the software GPS receiver, the decision-directed cross-product discriminator is used because it is insensitive to data bit transitions.

A PLL is much more accurate in carrier tracking than an FLL because it tracks the phase instead of the frequency; thus, receiver channels usually transform from an FLL state to a PLL state after the carrier mismatch converges into the lock range of the PLL used. The PLL realized in the software GPS receiver is a third-order PLL which is illustrated in Figure 4.10.

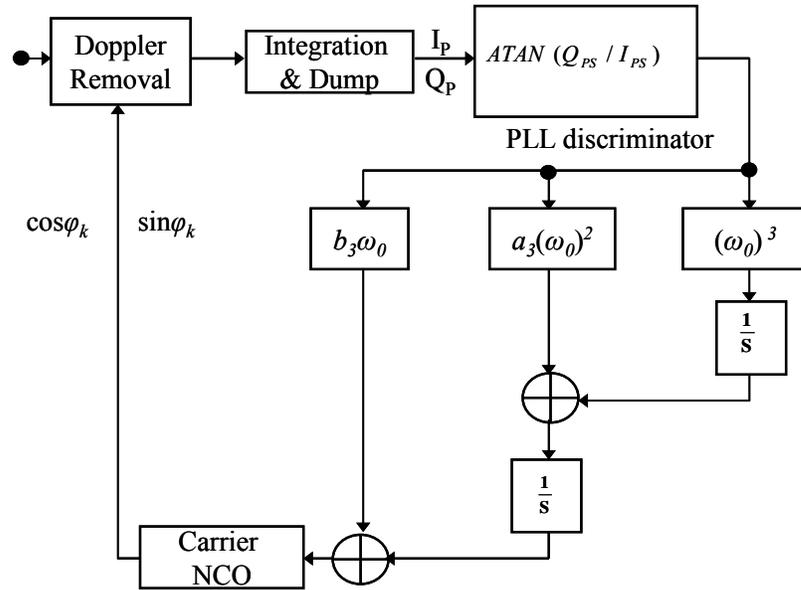


Figure 4.10: Phase lock loop

As mentioned before, the fundamental difference between a PLL and an FLL is the nature of the discriminator. The discriminator of an FLL detects frequency mismatch while the discriminator of a PLL detects phase mismatch. As listed in Table 4.1 (Raquet, 2001), there exist quite a few PLL discriminators which investigate the phase angle of the punctual in-phase component and quadrature-phase component pair (I_P, Q_P) .

Table 4.1: PLL discriminators

Discriminator Algorithm	Costas Loop	Output phase error	Notes
Sign(I)·Q	Y	$\sin\phi$	Near optimal at high SNR. Slope proportional to signal amplitude.
I·Q	Y	$\sin 2\phi$	Near optimal at low SNR. Slope proportional to signal amplitude squared.
Q/I	Y	$\tan\phi$	Suboptimal, but good at high and low SNR. Slope is not signal amplitude dependent.
atan2(Q,I)	N	ϕ	Optimal (maximum likelihood estimator) at high and low SNR. Slope is not signal amplitude dependent.
atan(Q/I)	Y	ϕ	Optimal (maximum likelihood estimator) at high and low SNR. Slope is not signal amplitude dependent.

In the software receiver, the $\text{atan}(Q/I)$ discriminator is used because of its high performance. The loop filter transforms phase mismatch into a carrier Doppler change which drives the carrier NCO to reduce the phase error. Without external aiding, such as aiding from an Inertial Measurement Unit (IMU), a PLL needs to keep track of the total dynamic stress, such as user dynamics, inner oscillator error, atmospheric effects, and multipath effects. This makes the carrier loop a weak loop which may lose lock easily. As a result, a third-order PLL is realized here because it can tolerate higher dynamics. A high order DLL is not necessary because a DLL is normally assisted by the carrier loop which can remove user dynamics and variation of the receiver oscillator, the major

sources of loop dynamic stress. A DLL simply needs to track the difference between the code and carrier loops which is quite small.

4.4 DLL Lock Detectors and C/N₀ Estimation

Knowledge of code lock is obviously the same as the knowledge of received signal power (Van Dierendonck 1996). Code lock detection is very similar to estimating C/N₀. A good C/N₀ would be required only if the receiver is operating on or near the correlation peak, say in a situation where code lock is achieved.

Consider measurements of total power in 1/T (wide band power) and 1/MT (narrow-band power) noise bandwidths of the following form:

$$WBP_k = \left(\sum_{i=1}^M (I_i^2 + Q_i^2) \right)_k \quad (4.26)$$

$$NBP_k = \left(\sum_{i=1}^M I_i \right)_k^2 + \left(\sum_{i=1}^M Q_i \right)_k^2 \quad (4.27)$$

The exact relationship between these measurements and signal-plus-noise power is not known. However, a normalized power defined as Equation 4.28 gives statistics that provide a monotonic function of C/N₀ which is shown in Figure 4.11 (Van Dierendonck, 1996).

$$NP_k = \frac{NBP_k}{WBP_k} \quad (4.28)$$

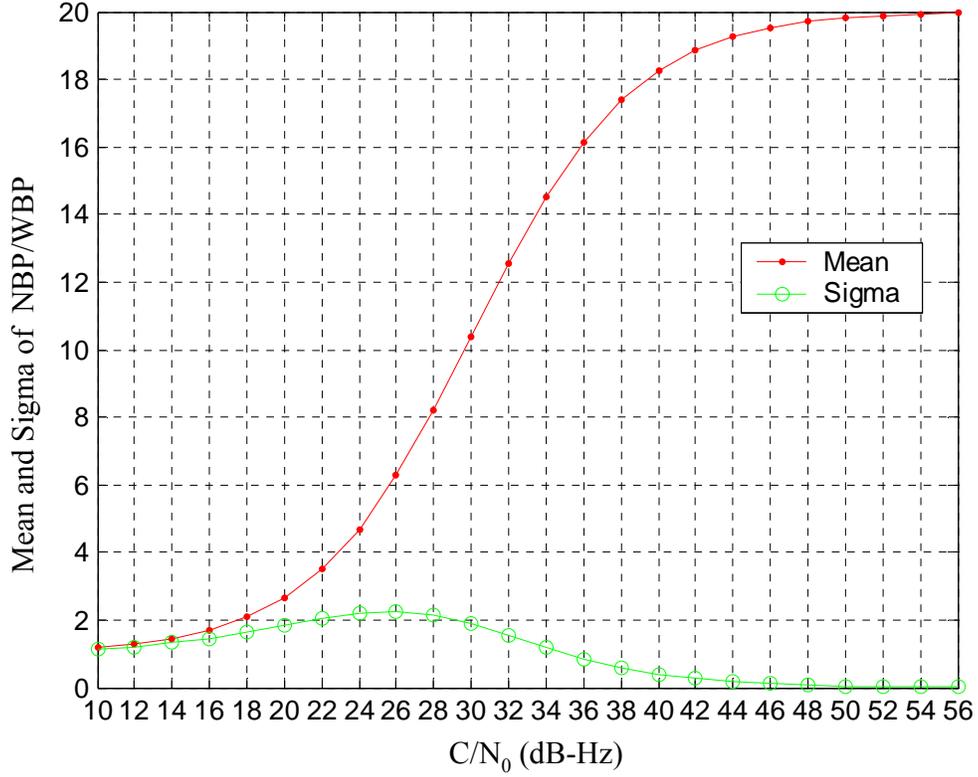


Figure 4.11: Statistics of normalized power NP for M=20

The lock detector measurement is:

$$\hat{\mu}_{NP} = \frac{1}{K} \sum_{k=1}^K NP_k \quad (4.29)$$

A threshold can be set at a desired minimum C/N_0 . If 26 dB is the minimum expected C/N_0 , the average of NP_k should be greater than 6. When this average value is less than 6, it indicates a loss of lock of the DLL.

The same measurement is used to estimate C/N_0 as follows (Raquet 2001):

$$\frac{\hat{C}}{N_0} = 10 \log_{10} \left(\frac{1}{T} \frac{\hat{\mu}_{NP} - 1}{M - \hat{\mu}_{NP}} \right) \quad (4.30)$$

Equations (4.29) and (4.30) are used in the software receiver to compute a DLL detector value and to estimate one second average C/N_0 with $M=20$ and $K=50$.

4.5 Bit Synchronization

The histogram approach is used to perform bit synchronization in this software GPS receiver. This approach breaks a data bit period (20 ms) into 20 C/A-code 1 ms epoch periods and senses sign changes between successive epochs. For each sensed sign change, a corresponding histogram cell count is incremented until a count in one specific cell exceeds the other 19 bins by a pre-specified amount. An example of one such histogram for successful bit synchronization along with count thresholds is shown in Figure 4.12. The procedure can be described as follows (Krumvieda et al., 2001):

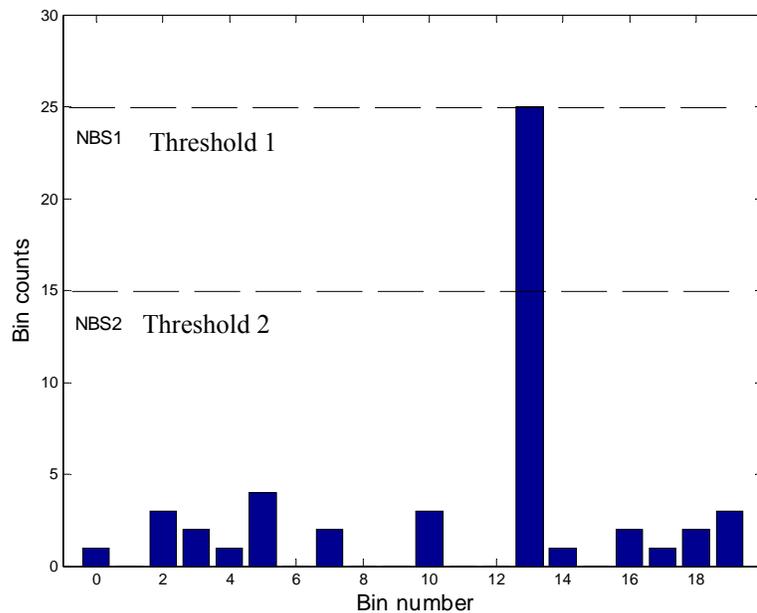


Figure 4.12: A successful bit synchronization histogram

- 1) A cell counter K_{cell} is arbitrarily set and runs from 0 to 19.
- 2) each sensed sign change is recorded by adding 1 to the histogram cell corresponding to K_{cell} .
- 3) The process continues until one of the following occurs:
 - (a) Two cell counts exceed threshold NBS_2
 - (b) Loss of lock
 - (c) One cell count exceeds threshold NBS_1
- 4) If (a) occurs, the bit synchronization fails and is reinitialized. If (b) occurs, one tries to reestablish lock. If (c) occurs, bit synchronization is successful, and the C/A code epoch count is reset to the correct value.

The thresholds NBS_1 and NBS_2 are determined as follows:

- 1) The probability of making an error in determining a sign change at a desired S/N_0 is

$$P_{esc} = 2P_e(1 - P_e) \quad (4.32)$$

where

$$P_e = \text{erfc}'\left[\sqrt{2(S/N_0)T}\right] \text{ if a PLL is used}$$

$$\text{erfc}'(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2} dy$$

- 2) The number of entries, N_{bs} , in a cell has a binominal distribution. Over T_{bs} seconds, the average number of sign changes (bit transitions) is $25T_{bs}$, so that

$$NBS_1 = 25T_{bs} \quad (4.33)$$

in the correct cell, and

$$\bar{N}_{bs} = 50 T_{bs} P_{esc} \quad (4.34)$$

in the other cells. The standard deviation of N_{bs} in any cell is

$$\sigma_{N_{bs}} = \sqrt{50 T_{bs} P_{esc} (1 - P_{esc})} \quad (4.35)$$

3) The thresholds, as well as the time interval T_{bs} , are selected to provide a good spread (say 3σ) between them at a desired SNR. That is, given NBS_1 , select NBS_2 and T_{bs} for a desired SNR so that

$$25T_{bs} - 3\sqrt{50 T_{bs} P_{esc} (1 - P_{esc})} \geq NBS_2 \geq 50 T_{bs} P_{esc} \quad (4.36)$$

4.6 Sub-frame Synchronization

Frame synchronization is required in order to process the GPS data. When timing uncertainties are large, parity decoding is not possible since the boundaries of the words are unknown. The following section details the procedure necessary to synchronize frames.

The GPS navigation message frame structure is shown in Figure 4.13. Each frame is partitioned into five sub-frames. Each sub-frame is subdivided into ten 30-bit words with the two leading words being the telemetry (TLM) word and the handover (HOW) word.

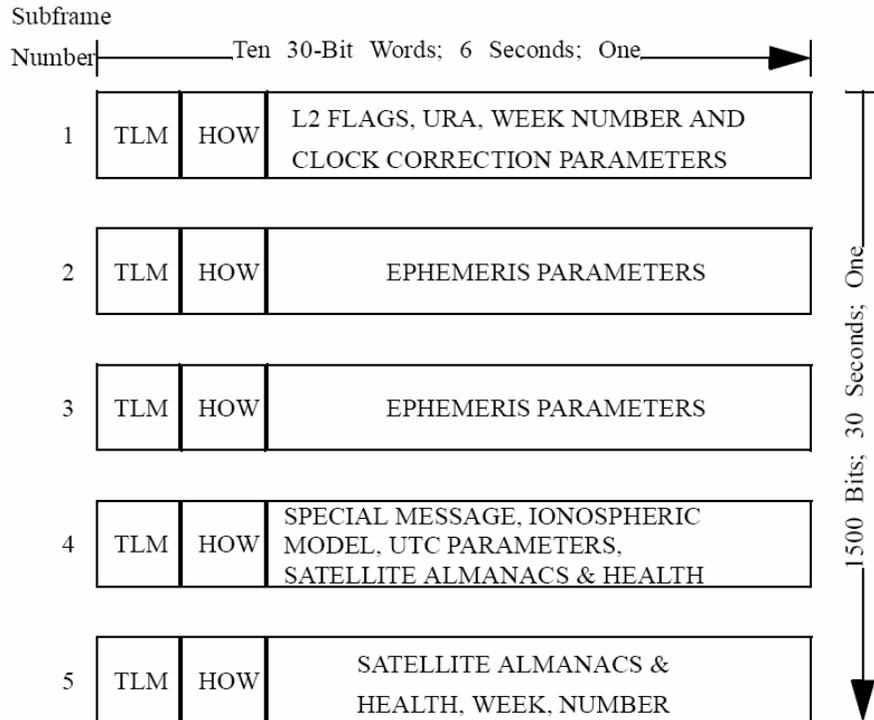


Figure 4.13: Navigation message

The structure for the TLM and HOW words is given in Figure 4.14. The TLM word is 30 bits long, occurs every six seconds in the data frame, and is the first word in each sub-frame. Each TLM word begins with a preamble, followed by the TLM message, two reserved bits and six parity bits.

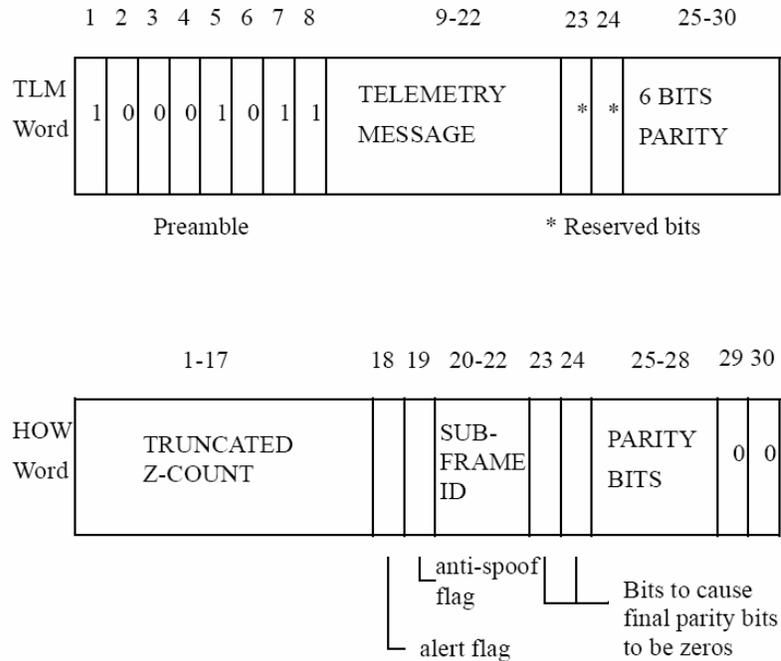


Figure 4.14: TLM word and HOW word

The HOW word is 30 bits long, occurs every six seconds in the data frame and is the second word in the sub-frame, immediately preceded by the TLM word. The convention is that the most significant bits (MSB) are transmitted first. The HOW word begins with seventeen MSBs of the time-of-week (TOW) count. The full TOW count consists of the 19 least significant bits (LSB) of the 29-bit Z-count. Bit 18 is the momentum flag (for SV configuration 000) or “alert” flag (for SV configuration 001) and bit 19 is the synchronization flag (for SV configuration 000) or antispoof flag for (SV configuration 001) (Van Dierendonck, 1996).

The algorithm for positively finding the correct preamble and performing frame synchronization is as follows:

[1] Search for either an upright or inverted preamble.

[2] When one is found, which could be a legitimate pattern somewhere else in the data stream, a check is required to verify it is the beginning of a 30-bit word. This is accomplished by collecting the following 22 bits and checking parity. If parity does not pass, the candidate preamble is discarded.

[3] If parity passes, it confirms that the preamble existed at the beginning of a word. The parity algorithm will also resolve the sign ambiguity. However, there are also other legitimate patterns at the beginning of other words, so additional checks are required. If it is the correct TLM word, the following word must be a HOW word that contains a truncated Z-count. The first eight bits of this truncated Z-count can also resemble a preamble.

[4] Parity should pass on the HOW word. If not, the frame synchronization procedure should be restarted. Two checks can be made to verify a legitimate HOW word that the Z-count is reasonable, and it agrees with the sub-frame count. Of course, there is a small probability that these conditions could also occur elsewhere in the message. Thus, further checking is required.

[5] If the HOW seems to be legitimate, provisional demodulation of the other words can commence, and they can be stored in memory. A final check on the next preamble and

the next Z-count solidifies the frame synchronization. That is, the preamble is where it is supposed to be, and the Z-count increments by one.

4.7 Raw Measurement Derivation

4.7.1 Pseudorange

A whole pseudorange measurement requires more than the DLL. The DLL provides the high resolution or fine measure of the overall pseudorange, but the whole pseudorange also requires some gross information. To get such gross information, certain counters become necessary.

A pseudorange measurement can be calculated by the following equation

$$\rho(t) = c[t_u(t) - t^{(s)}(t - \tau)] \quad (4.37)$$

where

$$\begin{aligned} t^{(s)}(t - \tau) = & \text{Z - count} \\ & + \text{number of navigation bits} \\ & + \text{number of C/A codes} \\ & + \text{number of whole C/A code chips} \\ & + \text{fraction of C/A code chip} \end{aligned} \quad (4.38)$$

This situation is shown in Figure 4.15 (Misra and Enge, 2001). As shown, the arrival time $t_u(t)$ kept by an inner clock is defined by a transition of the receiver clock. In general, these transitions occur sometime in the middle of a C/A code chip, and so the larger task

is to establish the transmission time, according to the satellite, of the received code feature identified by the receiver clock transition. Satellite time is kept by the Z-count which is also included in the navigation message. More specifically, the Z-count increments every 1.5 seconds, and the navigation message starts a new sub-frame every 6 seconds. These events are used to broadcast a coarse notion of satellite time to all users, and they are needed to compute pseudoranges.

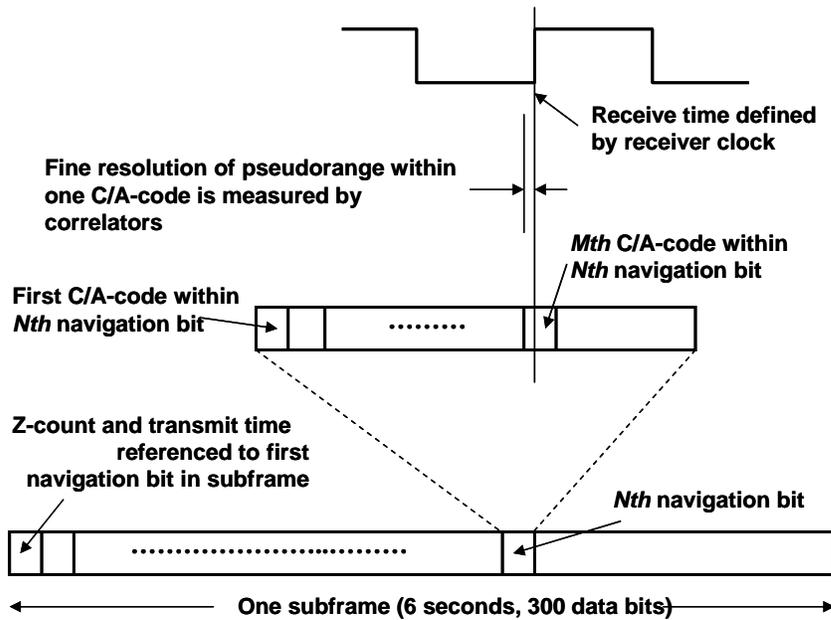


Figure 4.15: Pseudorange construction

Since the Z-count establishes satellite time at the beginning of each sub-frame, the transmission time is the Z-count plus the whole number of C/A-code chips since the beginning of the sub-frame. The elapsed time can be measured using the following components: the whole number of navigation bits; plus the whole number of C/A-codes since the beginning of the current navigation bits; plus the number of whole C/A-code chips since the beginning of the current code; plus the fraction of the current chip. The last two are measured by the DLL and the rest are measured by counters in the bit

synchronization and sub-frame synchronization module in the software GPS receiver developed herein.

4.7.2 Carrier Phase and Doppler

It is much easier to obtain Doppler and carrier phase measurements than to obtain pseudorange measurements because only the fractional part is desired. The Doppler can be read directly from the carrier NCO while the carrier phase needs to be assisted by a carrier counter which is used to count the integer cycles the incoming carrier has changed. The fractional part is recorded with the carrier NCO and the summation of the integer part and the fractional part give the carrier phase measurement since the loop is locked.

4.8 Navigation Solution

The following work focuses on how to obtain the navigation solution after the pseudorange and carrier phase raw measurements are derived by the software receiver. To produce the navigation solution, C³NAV²™, a software package that has been proven successful by the PLAN group in processing GPS and GLONASS pseudorange data in both static and kinematic modes (Petovello, 2000) is used. It can also detect cycle slips and can smooth pseudorange measurements with the carrier phase. C³NAV²™ takes an observation file and ephemeris file as input and outputs four files containing estimated position coordinates, estimated velocities, measure of satellite geometry and a processing results summary.

CHAPTER 5

VERIFICATION OF THE SOFTWARE GPS RECEIVER BY A REAL IF GPS SIGNAL

5.1 Software Receiver Verification Scheme

The procedure for conducting the software GPS receiver verification begins with collecting real IF GPS data at a known position; then feeding the data to the software receiver; acquiring the pseudorange and Doppler measurements via the receiver tracking loops; computing the receiver position; and, finally, comparing the computed positions with the known position information. According Parkinson (1996), the world wide civilian horizontal positioning error for GPS is potentially about 10 metres. Therefore, if the position error of the software receiver is better than 10 metres, it is considered to be performing properly. In addition, the RF GPS signal can be received and processed by a hardware GPS receiver simultaneously. Therefore, an objective comparison can be made between the results of the hardware GPS receiver and the results of the software GPS receiver.

A hardware front-end, 'GPS Signal Tap' (Figure 5.1), made by the Accord company is used to collect the real IF GPS signal. It is an L1 GPS receiver front-end and consists of a

two-stage RF down-converter. The RF signal is down-converted to the intermediate frequency, 15.42 MHz, at the second stage. This IF signal can be sampled and stored in binary format on a PC for analysis by the user at a programmable frequency.



Figure 5.1: Hardware front-end “GPS Signal Tap”

Eighty seconds of real IF GPS data were collected by the GPS Signal Tap at Mountain Standard Time 10:55 AM on October 20, 2003. The antenna was fixed atop the roof of the Calgary Centre for Innovative Technology building, University of Calgary, having WGS84 coordinates X: -1641944.887 m, Y: -3664803.795 m, Z: 4940009.362 m. The IF bandwidth of the Signal Tap is 2 MHz. A signal with 15.42 MHz intermediate frequency was sampled at a sampling rate of 4.75 MHz, resulting in a base-band signal centred at 1.17 MHz. One-bit quantization was conducted. The collected data consists of 1 and 0 sequence which is stored in a binary file. For processing convenience, it was then converted to a 1 and -1 sequence with binary format and sent to the software GPS receiver described in Chapter 4.

At the same time, a Novatel OEM4 hardware GPS receiver was used to receive the RF GPS signal; some real time processing results at a certain epoch during the signal collecting period are shown in Figure 5.2.

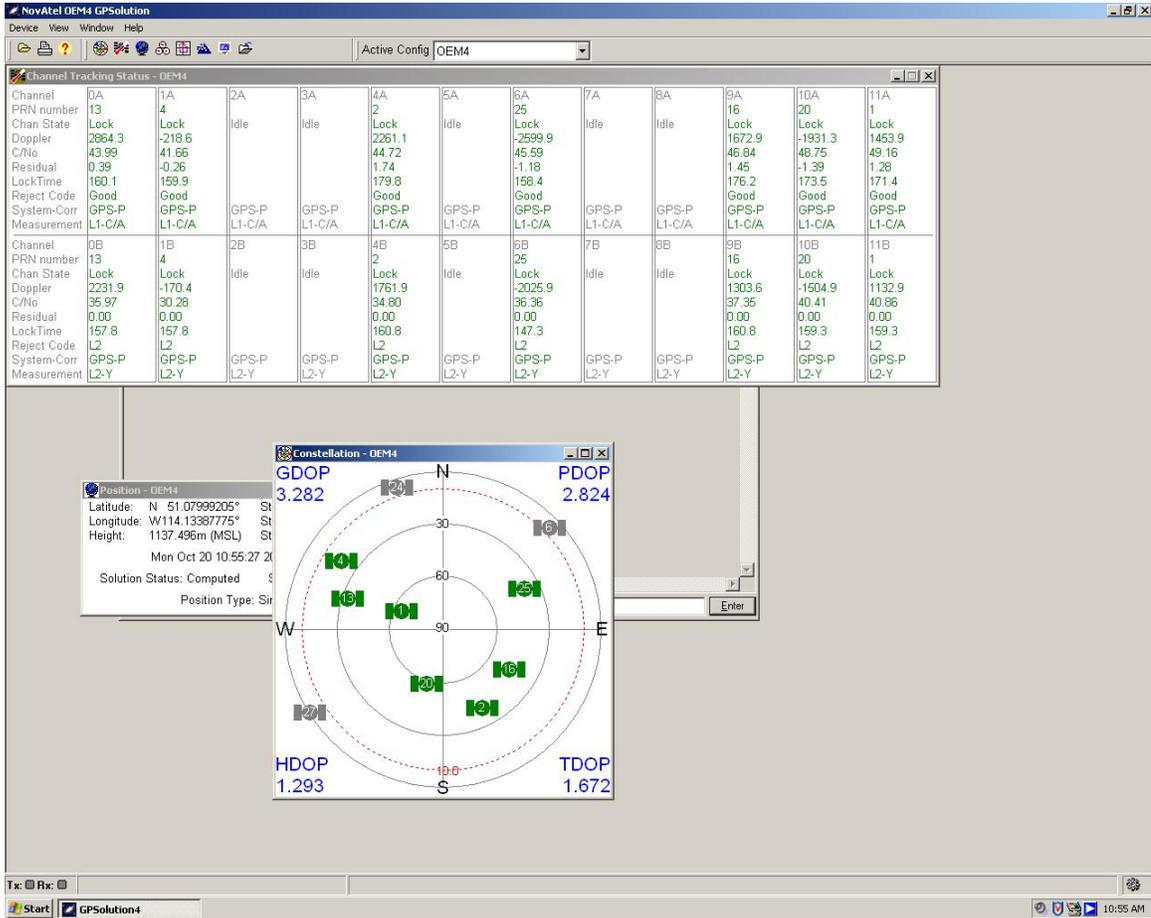


Figure 5.2: OEM4 receiver tracking status

From the above figure, it can be seen that seven satellites were available. The Doppler and C/N_0 of each satellite are shown in Table 5.1.

Table 5.1: Doppler and C/N_0 of the observed satellites

PRN number	Doppler (Hz)	C/N_0 (dB-Hz)
1	1453.9	49.2
2	2261.1	44.7
4	- 218.6	41.7
13	2864.3	44.0
16	1672.9	46.8
20	-1931.3	48.8
25	-2599.9	45.6

5.2 Acquisition Results Verification

The collected real IF GPS data was fed to the software GPS receiver. Acquisition was conducted first. All 32 satellites were searched and seven satellites (1, 2, 4, 13, 16, 20, 25) were acquired. This is exactly the same as the result of the OEM4 GPS receiver. The acquired Doppler values are shown in Table 5.2. They are similar to the Doppler values tracked by the OEM4 receiver.

Table 5.2: Acquired Doppler values of the Signal Tap collected signal

PRN number	1	2	4	13	16	20	25
Doppler (Hz)	1466.7	2666.7	- 200	2866.7	1666.7	-1933.3	-2600

5.3 Tracking Results of the Software GPS Receiver and Navigation Solution

Verification

Data bits, Doppler, signal time delay (pseudorange), DLL lock detector, and C/N_0 values can be obtained from the tracking loops. However, only the average C/N_0 and Doppler derived from the software receiver are shown in the following figures and are compared

with those derived from the OEM4 GPS receiver, because not all of the other intermediate processing results of the OEM4 receiver are available for comparison purposes. To avoid repetition, the tracking results of the software receiver for all seven satellites except for C/N_0 and Doppler will be shown in Chapter 6 where simulator verification is discussed.

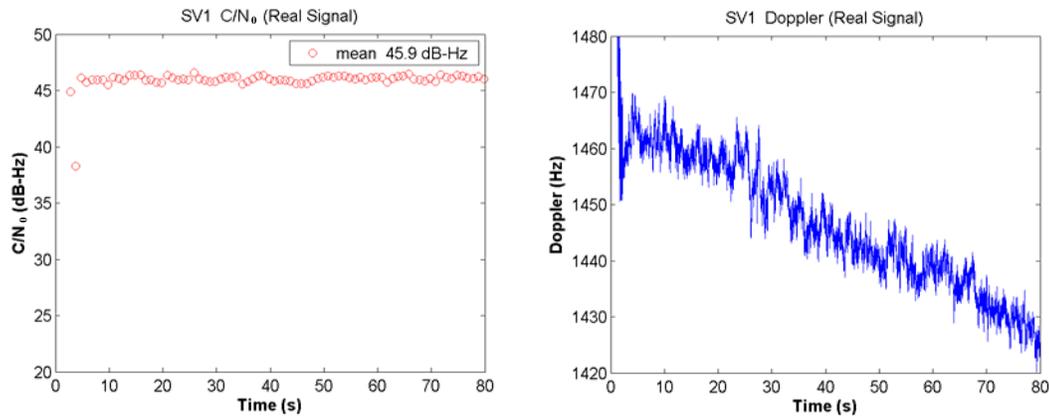


Figure 5.3 (a): Average C/N_0 and Doppler for SV1

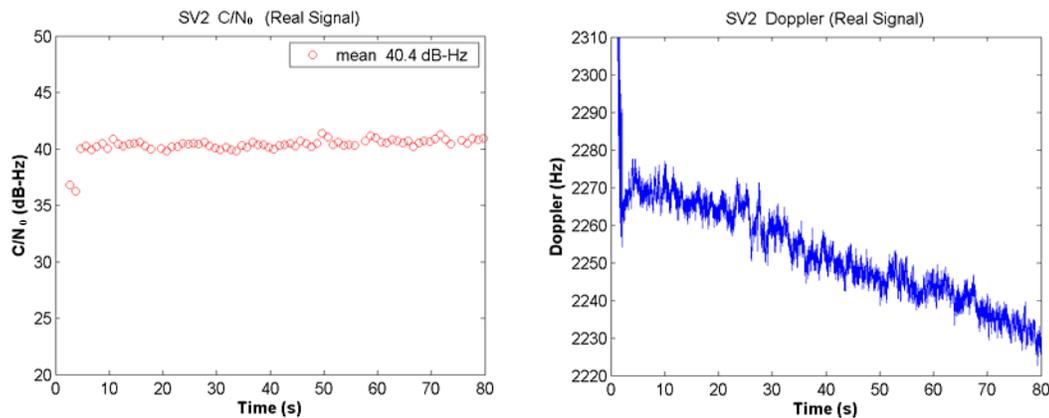


Figure 5.3 (b): Average C/N_0 and Doppler for SV2

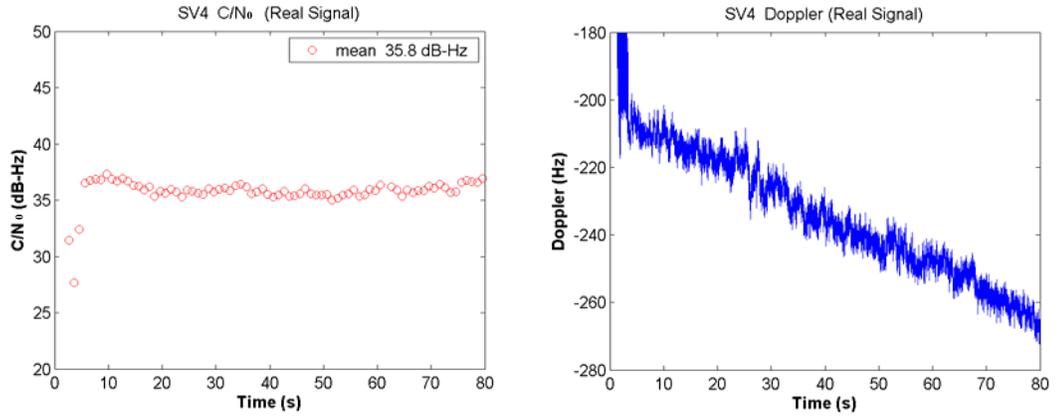


Figure 5.3 (c): Average C/N₀ and Doppler for SV4

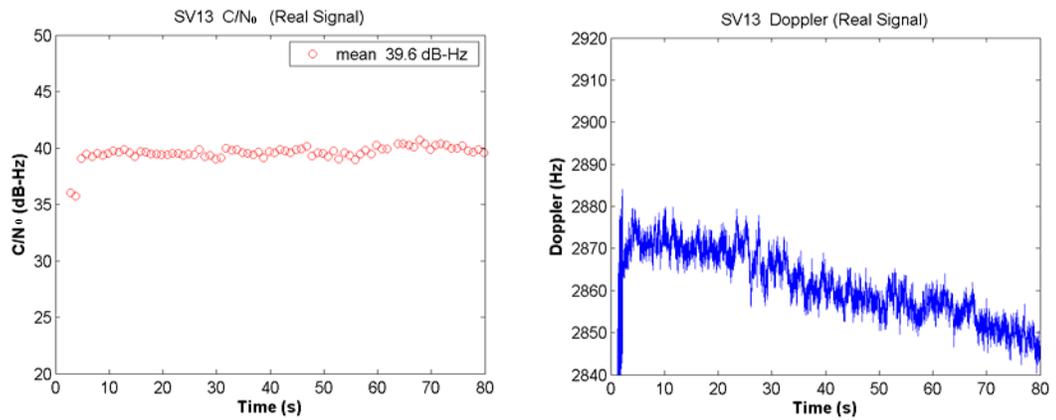


Figure 5.3 (d): Average C/N₀ and Doppler for SV13

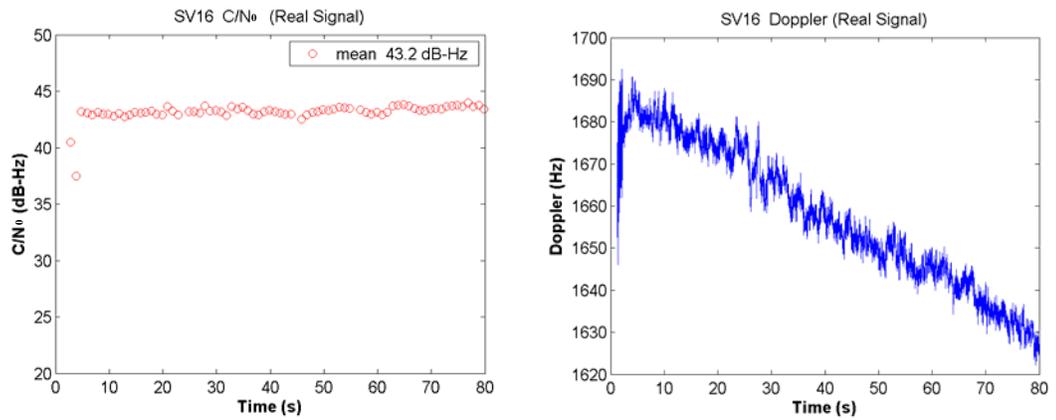


Figure 5.3 (e): Average C/N₀ and Doppler for SV16

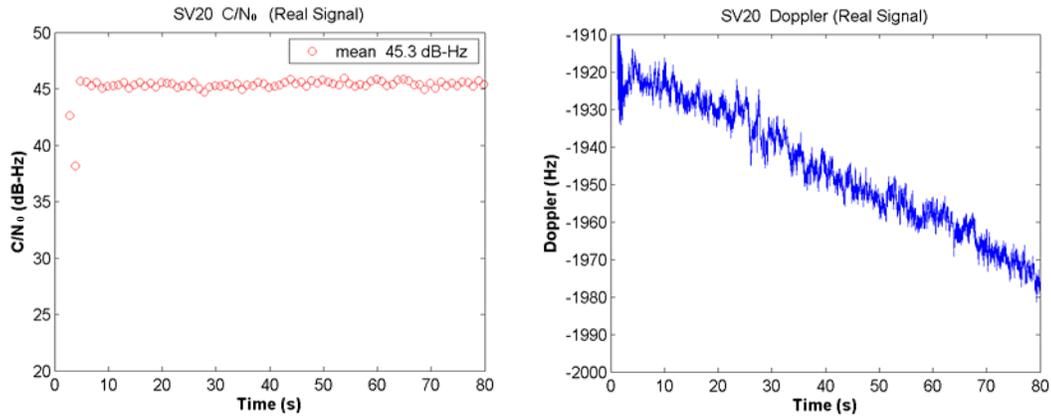


Figure 5.3 (f): Average C/N₀ and Doppler for SV20

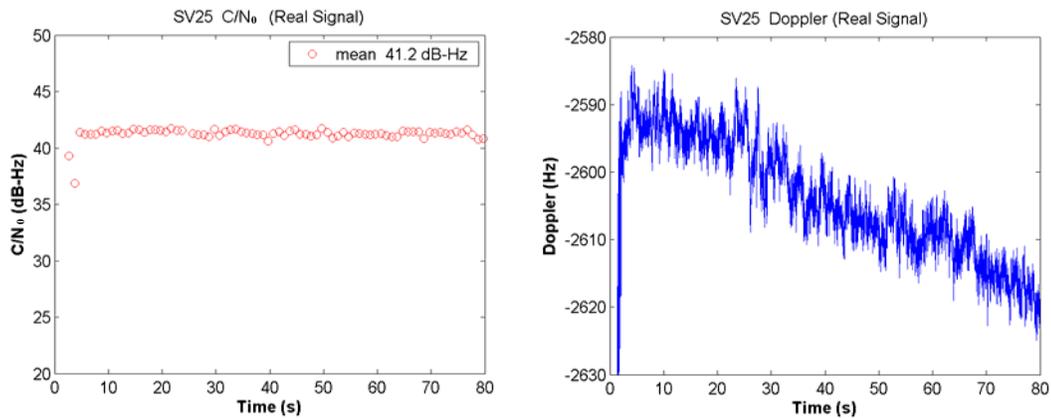


Figure 5.3 (g): Average C/N₀ and Doppler for SV25

For clarity of comparison, the results of the average C/N₀ and Doppler from the software receiver for each satellite and those from the OEM4 receiver results are listed in Table 5.3. The Doppler values from the software receiver were collected at GPS time 147327 s, the same as the time of the OEM4 receiver results shown in Figure 5.1 which was October, 20 10:55:27, Mountain Standard Time.

Table 5.3: Doppler and C/N₀ values obtained by the OEM4 receiver and software GPS receiver

PRN	Doppler (Hz)		C/N ₀ (dB-Hz)	
	OEM4 receiver	Software receiver	OEM4 receiver	Software receiver
1	1453.9	1456.6	49.2	45.9
20	-1931.3	-1928.9	48.8	45.3
16	1672.9	1674.3	46.8	43.2
25	-2599.9	-2598.1	45.6	41.2
2	2261.1	2260.8	44.7	40.4
13	2864.3	2869.1	44.0	39.6
4	- 218.6	- 215.6	41.7	35.8

From Table 5.3, it can be seen that the Doppler and the average C/N₀ of the software receiver and the OEM4 receiver are very close. There is about a 3 Hz difference for the Doppler. This may be caused by a different receiver implementation and different oscillator frequency errors occurring between the OEM4 receiver and the Signal Tap. The C/N₀ figures estimated by the software receiver are 3.5 to 6 dB lower than that estimated by the OEM4 receiver, but the relative value of C/N₀ is identical, which is listed in descending order in Table 5.3. This may have resulted from different C/N₀ estimation algorithms and better tracking performance of the OEM4 receiver. The comparison of Doppler and C/N₀ indicates that the software receiver can acquire and track the real IF GPS signal.

The receiver positions are computed by C³NAV^G²™ after the pseudorange and the carrier phase measurements are extracted from the tracking loops as described in Chapter 4. The navigation solution of the real IF GPS signal is shown in Figure 5.4.

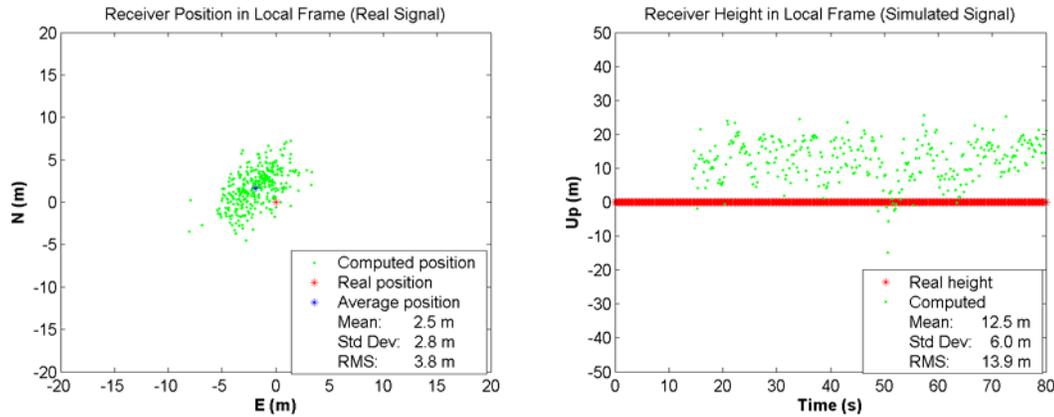


Figure 5.4: Navigation solution of the real GPS signal

It can be seen that the navigation solution matches the known real receiver position with an RMS of 3.8 m. A 2.5 m bias can be observed although all position points centred at the real position. This is because the pseudorange error caused by the ionospheric error, tropospheric error, and satellite geometry. To confirm this, it is better to compare the navigation solution in Figure 5.4 to the solution of an OEM4 receiver. Unfortunately, the required GPS data is not logged by OEM4 receiver simultaneously. An indirect comparing of the results of the software receiver and OEM4 receiver is conducted as the following description. A group of GPS data is logged by an OEM4 receiver during the test period at another point (X: -1641889.852 m, Y: -3664879.310 m, Z: 4939969.467 m) on the roof of the Engineering Building, University of Calgary. That point is about 100 metres away from the test point. Therefore, the navigation solution of the OEM4 receiver for that point should include almost same effects of the ionospheric error, tropospheric error, and satellite geometry. Figure 5.5 gives the local position of that point computed from the raw measurements generated by OEM4 receiver.

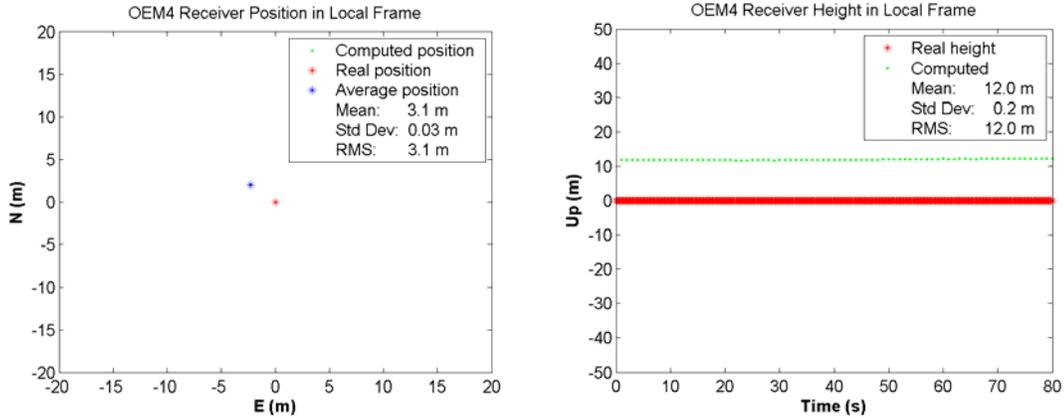


Figure 5.5: Navigation solution computed from the OEM4 receiver generated raw measurements

Comparing Figure 5.4 and 5.5, it can be seen that the receiver positions derived by the software receiver and OEM4 receiver have similar mean and RMS values and both of them have a position bias in the northwest direction. This confirms that the position bias in Figure 5.4 is caused by the ionospheric error, tropospheric error, and satellite geometry instead of other unexpected errors resulted by the implementation of the software receiver and the software receiver has similar navigation solution as OEM4 receiver. The software receiver is therefore proven having ability to conduct the GPS signal processing and positioning with a positioning accuracy less than 5 metres. In addition, it is noticed that the standard deviation of the position derived by the software receiver is not as small as that of the OEM4 receiver. This is because some more advanced techniques, such as narrow correlator, are used in OEM4 receiver.

The above comparison and analysis confirms the software GPS receiver works very well. It will be used to process the simulated IF GPS signal to check its validity in the next chapter.

CHAPTER 6

SIMULATED IF GPS SIGNAL VERIFICATION RESULTS

6.1 Simulator Verification Scheme

Verification of the simulator operation is conducted by analyzing the simulated signal and comparing it to a real GPS signal in the time, frequency, correlation, measurement and position domains. The real IF GPS signal collected by the Signal Tap used in Chapter 5 is simulated and stored in a data file. A Fast Fourier Transform is applied to both the simulated and real IF GPS signals to obtain signal power spectrums so that they can be compared in the frequency domain. Since the software GPS receiver has been verified as described in Chapter 5, it is used to process both the Signal Tap collected data and the simulated signals. The simulated signal is fed to the receiver, which performs acquisition and tracking and, finally, the receiver position is computed from the raw measurements obtained from the tracking loops. All of these results are compared with those produced from the real IF GPS signal. The position result derived by the simulated signal is also compared with the known receiver position. In addition, the effects of the simulation components, such as noise, filtering, quantization and errors, on the navigation solution are analyzed for different simulation cases.

6.2 Simulated Signal Description

The parameters used in signal simulation are shown in Table 6.1, which exactly copies the parameters of the Signal Tap. In the simulation, the receiver noise, satellite clock error, and ionospheric and tropospheric errors are simulated and filtering and quantization are performed. The C/N_0 for the signal of each satellite is simulated as a constant value according that of the OEM4 receiver as shown in Table 5.3. Ionospheric and tropospheric errors change a little (only tens of centimeters) during 80 seconds simulation and different for each satellite. The simulated C/N_0 and average ionospheric and tropospheric errors are listed in Table 6.2. For the selected simulation time, the simulated signal includes SV1, SV2, SV4, SV13, SV16, SV20, and SV25. Hereafter, the simulated signal means it is generated with noise, filtering, 1-bit quantization, and ionospheric and tropospheric errors added unless stated otherwise.

Table 6.1: Signal simulation parameters

Simulation time	Start time: GPS time 147310, Oct.20, 2003 80 seconds
Receiver trajectory	Mode 0: Static
Initial position	Latitude: 51.07997674° Longitude: - 114.13384815° Height: 1118.596 m
C/N_0	Constant C/N_0
Noise density	-203 dBW/Hz
Intermediate frequency	15.42 MHz
Sampling frequency	4.75 MHz
Base band frequency	1.17 MHz
Front-end Bandwidth	2 MHz

Table 6.2: Simulated C/N_0 and ionospheric and tropospheric errors

PRN	C/N_0 (dB-Hz)	Ionospheric error (m)	Tropospheric error (m)
1	49.2	8.8	2.6
20	48.8	11.6	3.7
16	46.8	8.1	6.8
25	45.6	8.8	4.3
2	44.7	10.8	3.3
13	44.0	10.0	2.8
4	41.7	10.1	3.8

6.3 Time Domain Verification

Because the signal was digitized, it is reduced to a 1 and -1 sequence and shows noise behaviour in the time domain. However, some properties can be shown in the frequency domain. The signal power spectrums of the real GPS signal and simulated signal are shown in Figure 6.1.

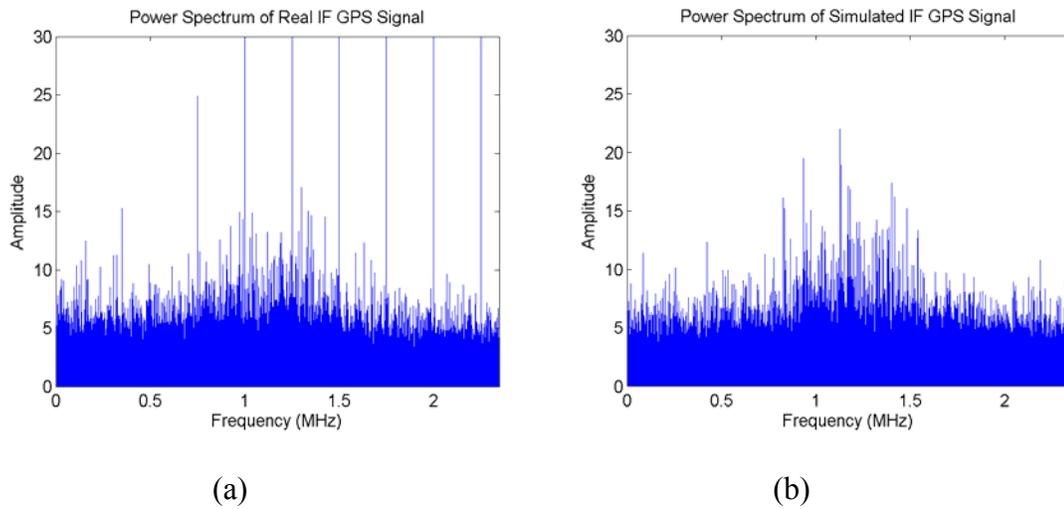


Figure 6.1: Power spectrum of the real and simulated IF GPS signal

The left-hand image in Figure 6.1 shows the power spectrum of the real IF GPS signal collected by the Signal Tap, while the right-hand image shows that of the simulated signal. It can be seen that the power spectrum of the real and simulated signals are of a similar shape. However, the power spectrum amplitude of the simulated signal is a little higher than that of the real signal. This is because of a range of external effects - such as noise, interference, Signal Tap hardware implementation errors, as well as factors that are present in the real GPS signal but are not simulated by the simulator. This comparison indicates that the simulated signal has essentially the same properties, but with a higher signal-to-noise ratio which results in better acquisition and tracking performance as demonstrated later.

6.4 Verification in the Correlation Domain

The alignment of the incident signal and the replica code and carrier is based on the correlation value of the two signals. The correct alignment is identified by the measurement of the output of the correlators. The acquisition results of both the simulated signal and the Signal Tap collected real signal are shown in the following figures.

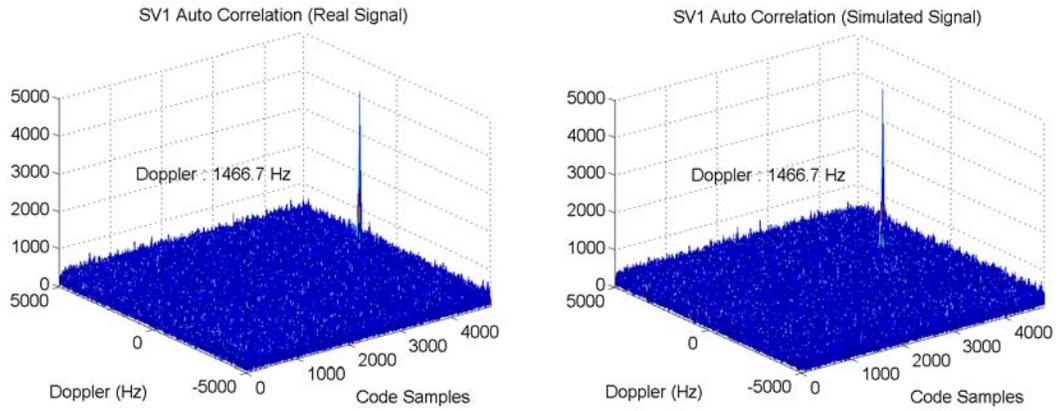


Figure 6.2 (a): Acquisition results of the real signal and simulated signal for SV1

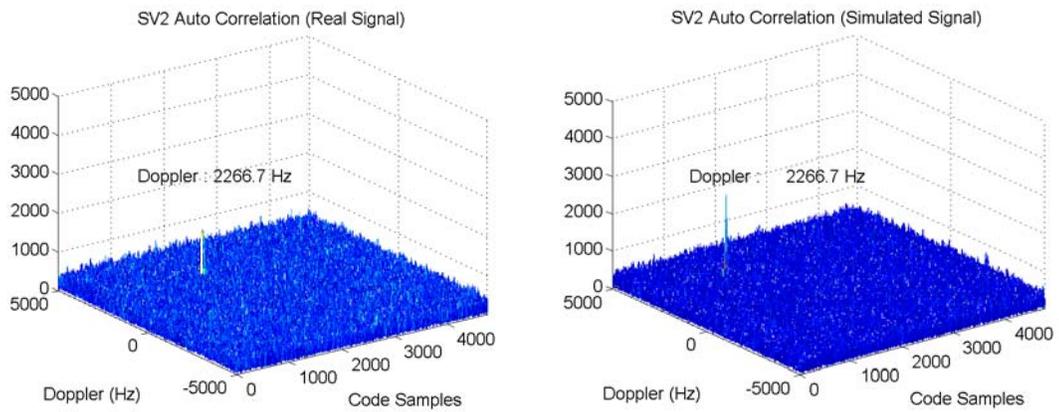


Figure 6.2 (b): Acquisition results of the real signal and simulated signal for SV2

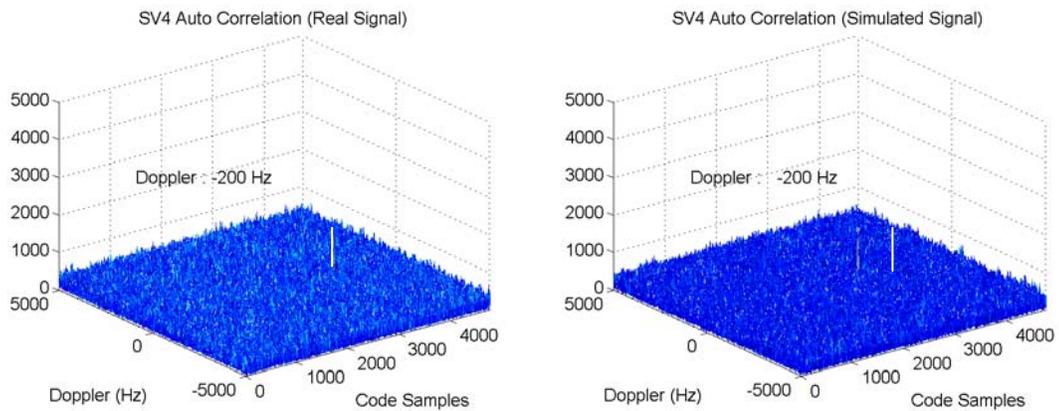


Figure 6.2 (c): Acquisition results of the real signal and simulated signal for SV4

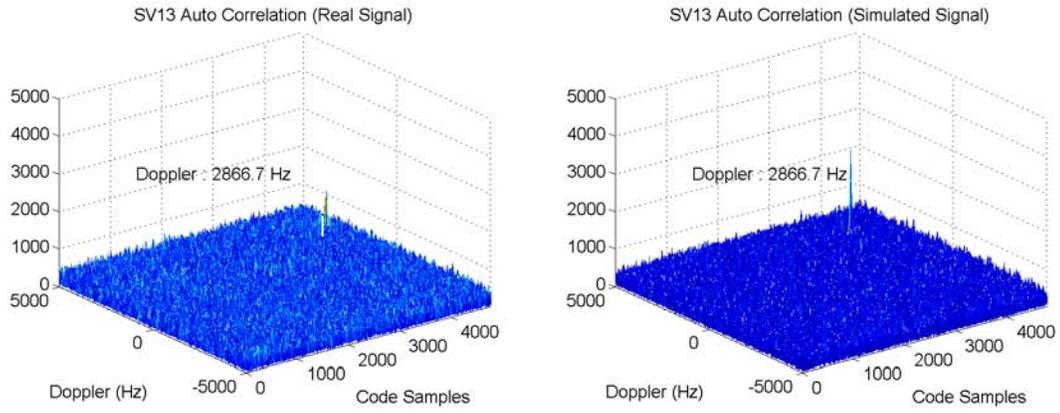


Figure 6.2 (d): Acquisition results of the real signal and simulated signal for SV13

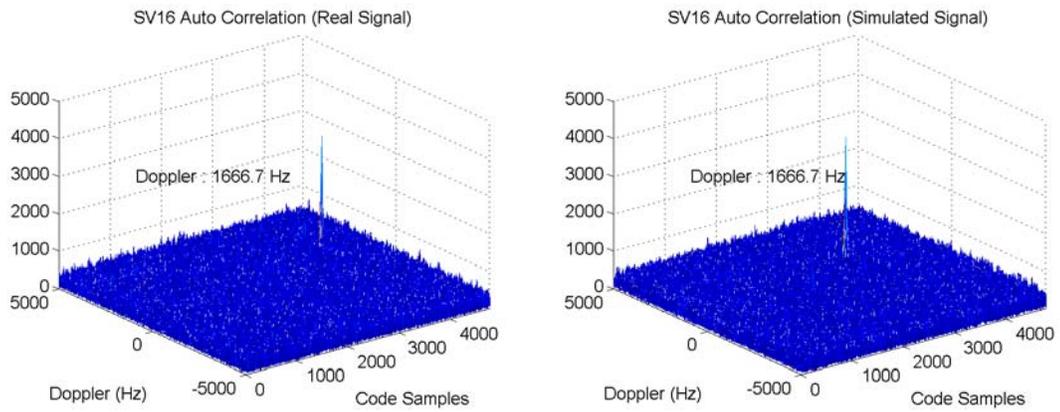


Figure 6.2 (e): Acquisition results of the real signal and simulated signal for SV16

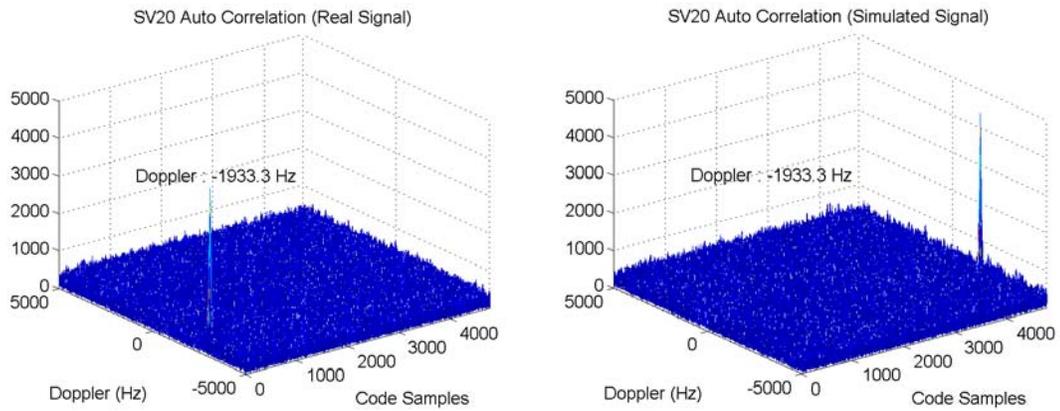


Figure 6.2 (f): Acquisition results of the real signal and simulated signal for SV20

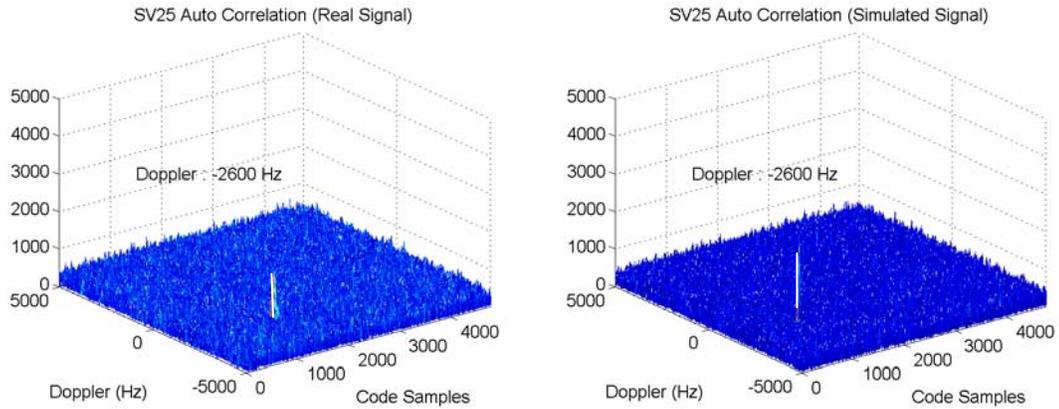


Figure 6.2 (g): Acquisition results of the real signal and simulated signal for SV25

The above acquisition results demonstrate that the simulated signal has the same acquired Doppler as the Signal Tap collected signal. The noise floors of both signals are at the same level, but the correlation values of the simulated signal are obviously larger than the real signal for all satellites. The approximate correlation peak values are listed in Table 6.3.

Table 6.3: Correlation peak values of all acquired satellites

PRN	Auto correlation values	
	Real signal	Simulated signal
1	4299	4638
20	4097	4542
16	3377	3579
25	1619	2820
2	1561	2690
13	1412	2689
4	1412	1914

In the above table, the correlation values of all acquired satellites are listed in descending order, which corresponds to the order of the C/N_0 values in Table 5.3. This demonstrates,

once again, that the simulated signal has a higher signal-to-noise ratio than the real signal, but the relative signal power is identical to the real signal. Therefore, the simulated signal has similar performance as the real signal in the correlation domain.

6.5 Tracking Result Comparison

6.5.1 Data Bits

Data bit demodulation is performed by a PLL. The I_{PS} samples are simply accumulated for the coherent integration time, with the sign of the result determining the data bits. The change of the sign of the I_{PS} integration value means that the phase of the data bit is reversed by 180° . There are usually a number of integration values in one data bit interval depending on the coherent integration time, which is 1 ms for the software receiver in this thesis. Figure 6.3 shows the resulting data bits of the Signal Tap collected and simulated signal for each satellite.

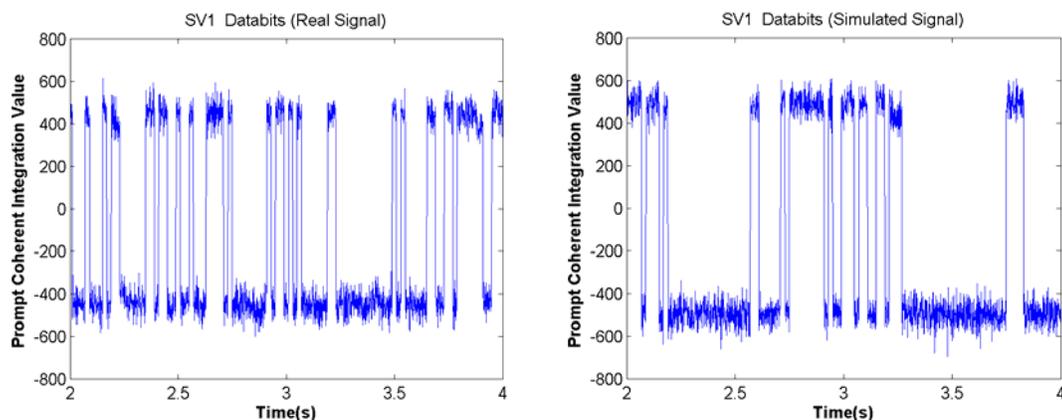


Figure 6.3 (a): Data bits of the real signal and simulated signal for SV1

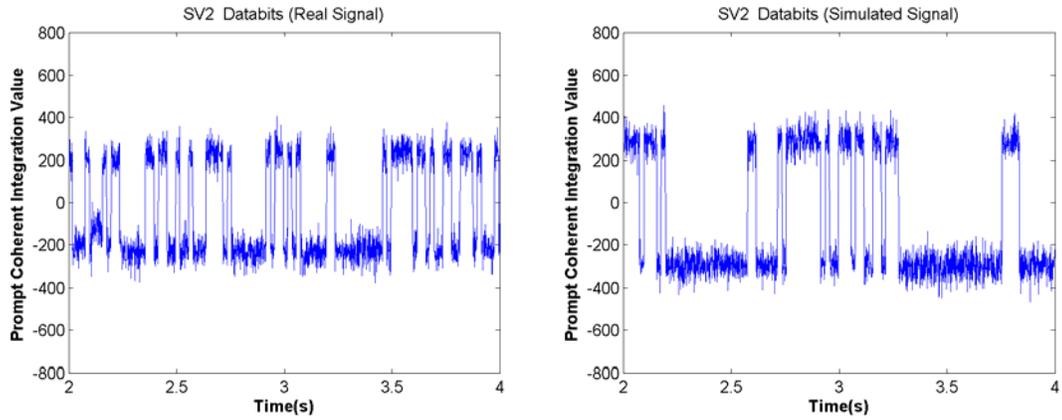


Figure 6.3 (b): Data bits of the real signal and simulated signal for SV2

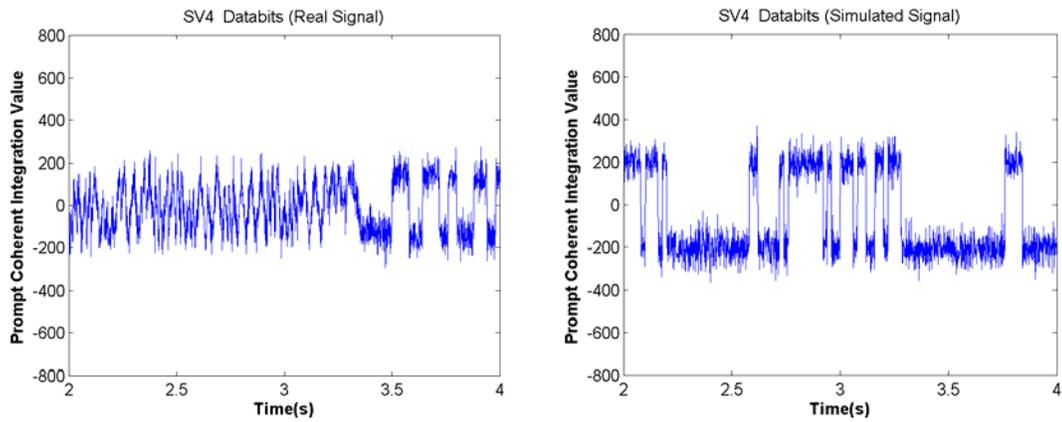


Figure 6.3 (c): Data bits of the real signal and simulated signal for SV4

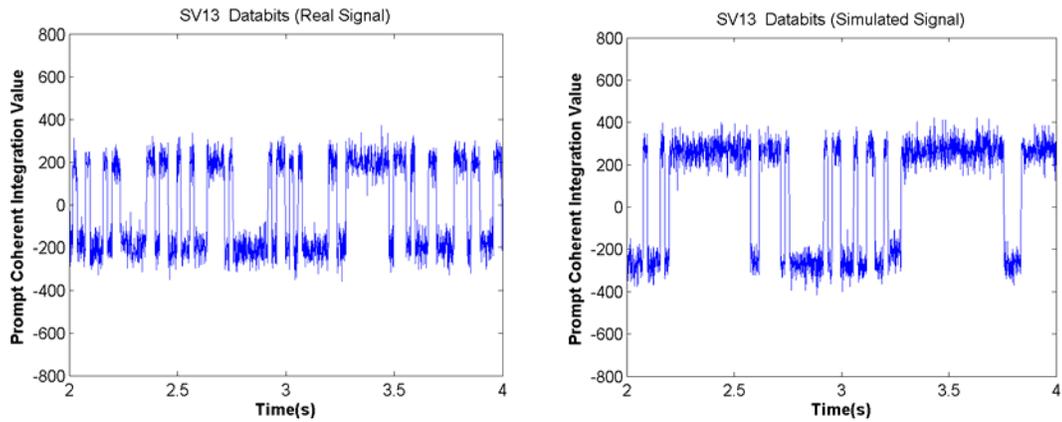


Figure 6.3 (d): Data bits of the real signal and simulated signal for SV13

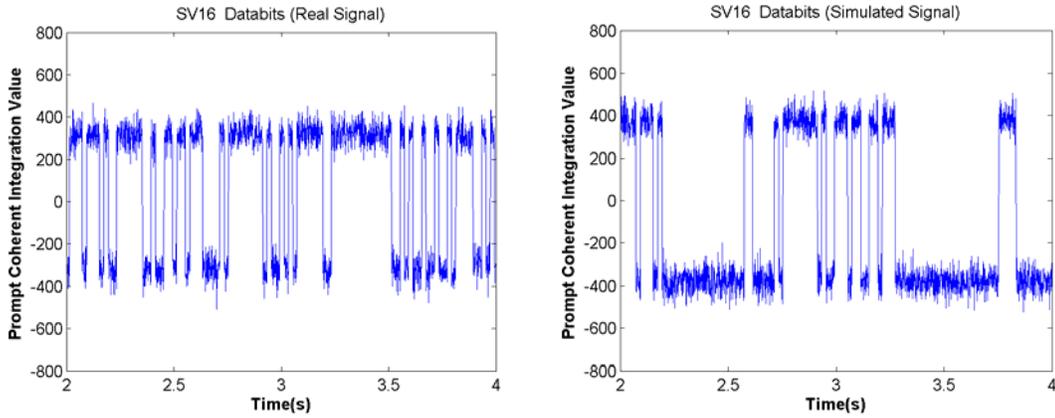


Figure 6.3 (e): Data bits of the real signal and simulated signal for SV16

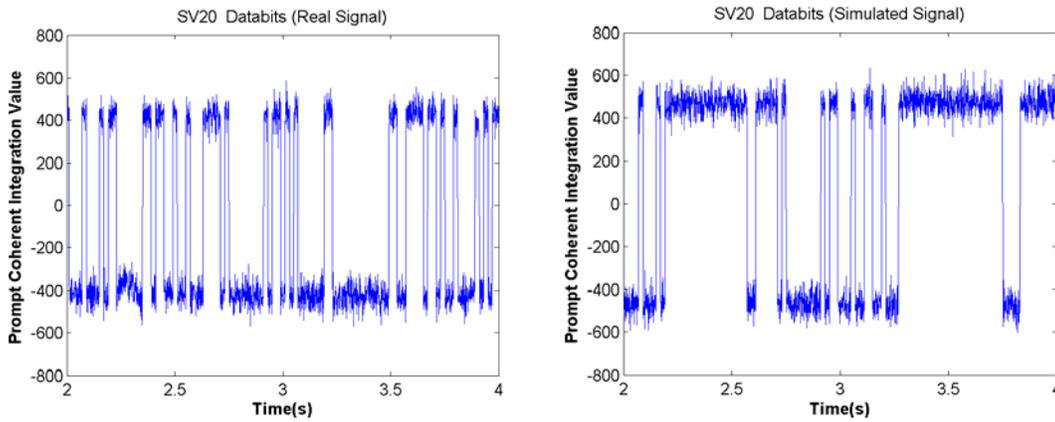


Figure 6.3 (f): Data bits of the real signal and simulated signal for SV20

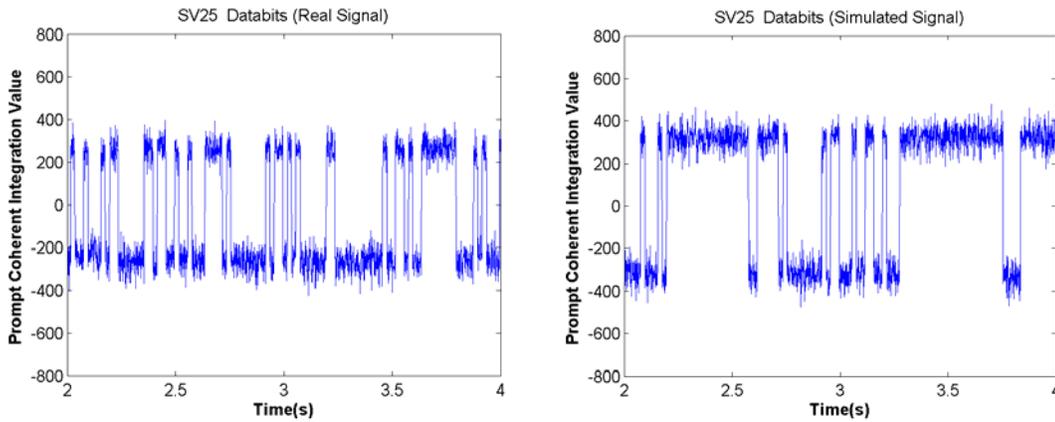


Figure 6.3 (g): Data bits of the real signal and simulated signal for SV25

From above figures, it can be seen that, the amplitude of the data bits of the simulated signal is higher than that of the real signal. It is to say the prompt coherent integration value I_{ps} of the simulated signal is higher. This is also because of the simulated signal has a higher signal-to-noise ratio than that of the real signal. Another thing that can be noticed is that the simulated signal has different data bits from the real signal; however, the simulated data bits for all satellites are the same. This is because not all ephemeris data are added in this batch of simulated navigation data although the simulator has the ability to do so. Only the data bits such as the preamble, sub-frame ID, Z count, and parity check, which greatly affect the tracking and pseudorange derivation, are available in the above figures. Therefore, the data bits in the left-hand images are different from those in the right-hand images.

6.5.2 Carrier- to-noise Ratio

The C/N_0 estimated by the software receiver are shown in Figure 6.4.

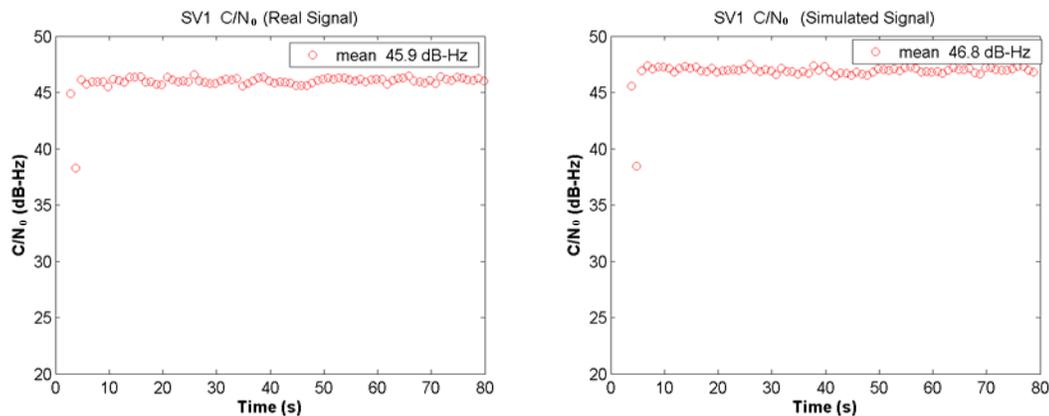


Figure 6.4 (a): C/N_0 of the real signal and simulated signal for SV1

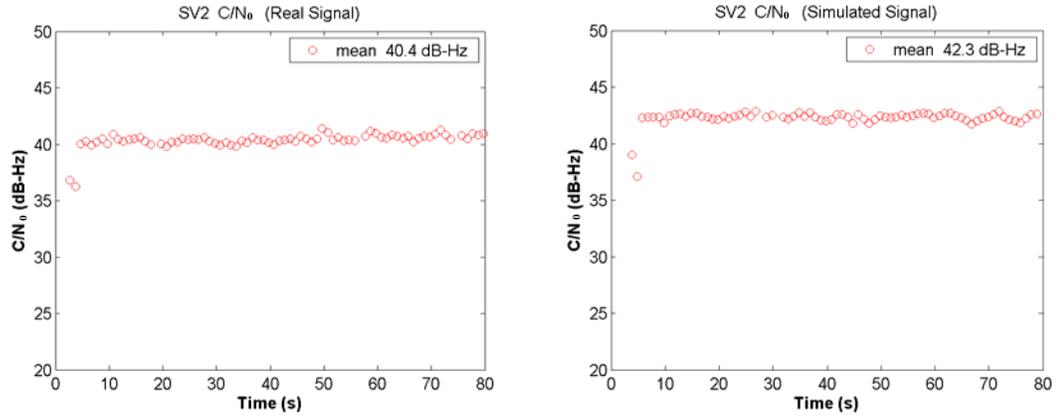


Figure 6.4 (b): C/N_0 of the real signal and simulated signal for SV2

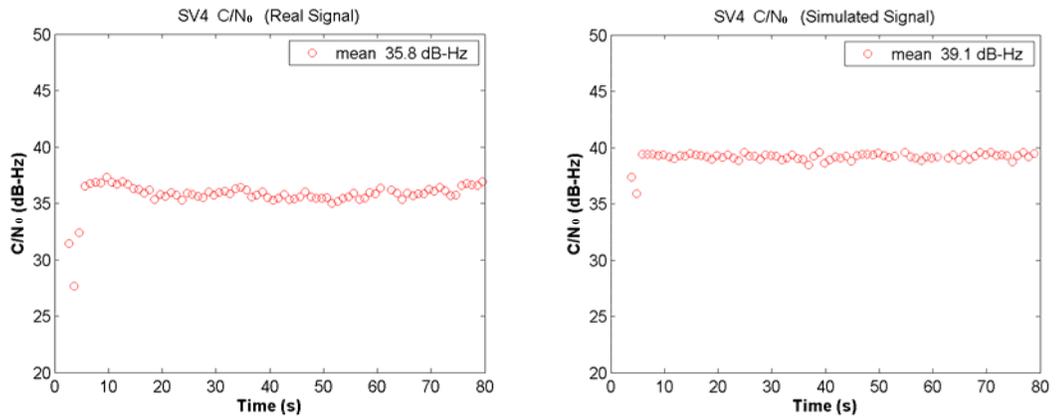


Figure 6.4 (c): C/N_0 of the real signal and simulated signal for SV4

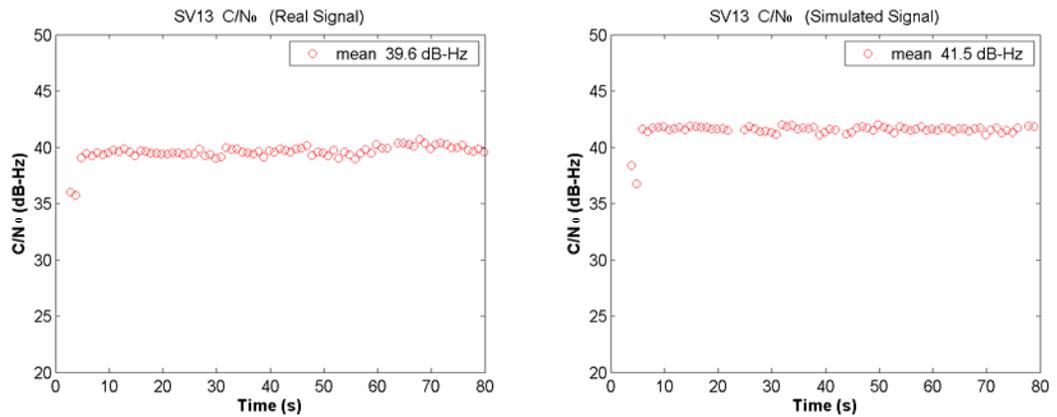


Figure 6.4 (d): C/N_0 of the real signal and simulated signal for SV13

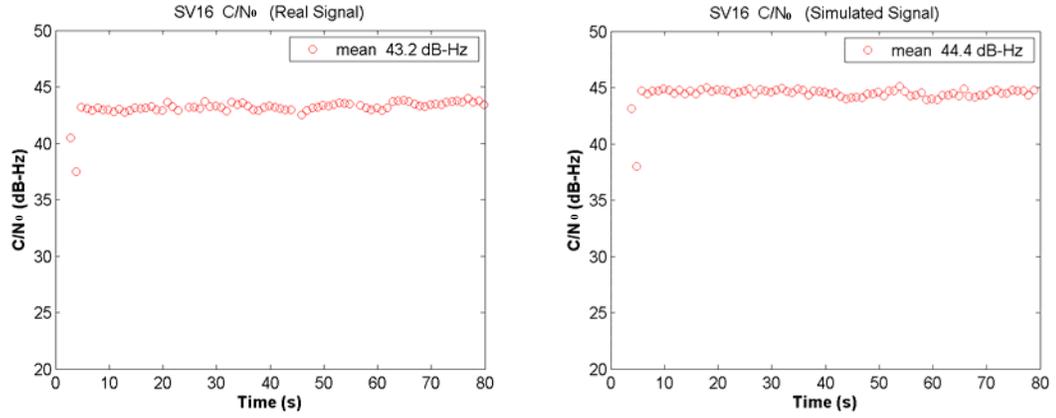


Figure 6.4 (e): C/N_0 of the real signal and simulated signal for SV16

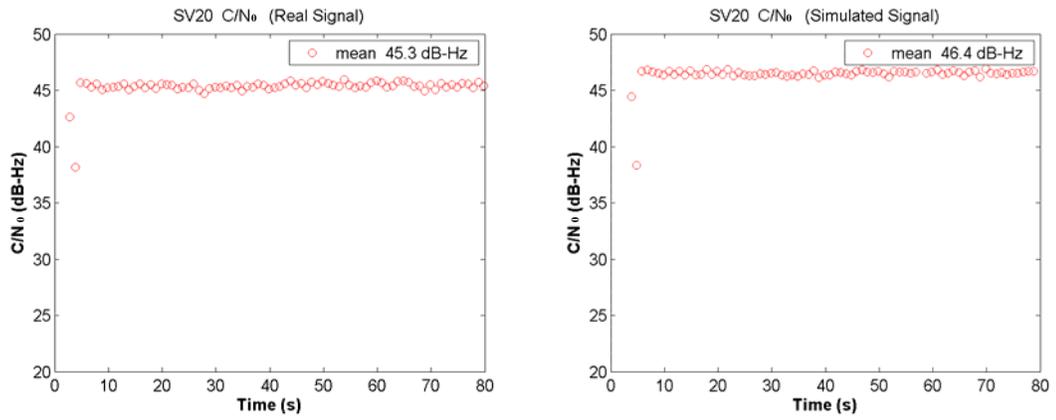


Figure 6.4 (f): C/N_0 of the real signal and simulated signal for SV20

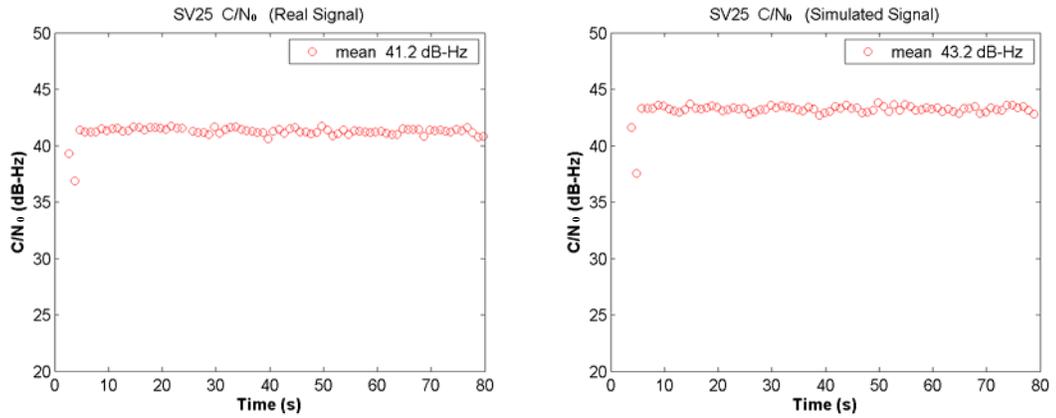


Figure 6.4 (g): C/N_0 of the real signal and simulated signal for SV25

It can be seen that the estimated C/N_0 of the simulated signal is 1 to 2 dB higher than that of the real signal. This confirms the results shown in Figure 6.1, 6.2, and 6.3. As described in Chapter 3, the signal power is computed by Equation 3.19, which is the function of the input parameters C/N_0 and noise density. In the simulation, the C/N_0 is set to a constant value according the OEM4 receiver estimated value for each satellite and the noise density is arbitrarily selected as -203 dBW/Hz. However, both the signal power and noise density for the real signal are not known. The simulated signal power may be close to the real value, may not exactly copy it. Therefore, the input parameters C/N_0 and noise density are main reasons of the simulated signal has a higher signal-to-noise ratio and better performance than the real signal. This will also be shown in the following results.

6.5.3 Delay Lock Loop Detector

The lock detector is a parameter that indicates the tracking status of the tracking loop. The receiver can make some status changes according its value. If the lock detector value is lower than the threshold, it means that the tracking loop has lost lock. The receiver will, consequently, change the loop bandwidth, or even return to acquisition mode. This function has not been developed in the software receiver as of yet, and is an excellent candidate for development as a future recommendation. The DLL detectors for all seven satellites are shown in the Figure 6.5.

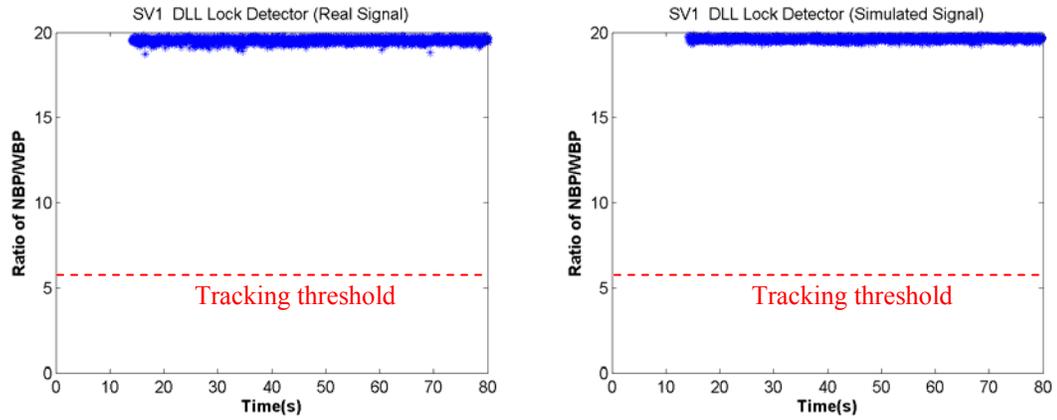


Figure 6.5 (a): DLL lock detector of the real signal and simulated signal for SV1

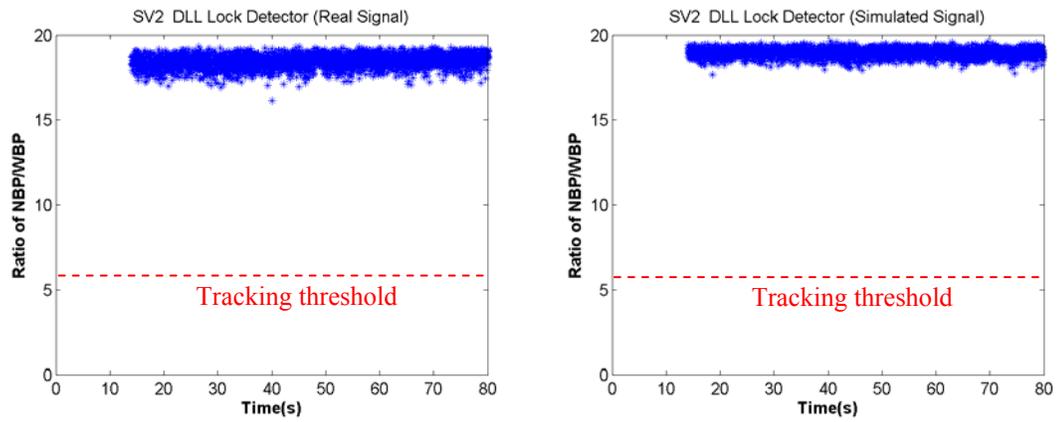


Figure 6.5 (b): DLL lock detector of the real signal and simulated signal for SV2

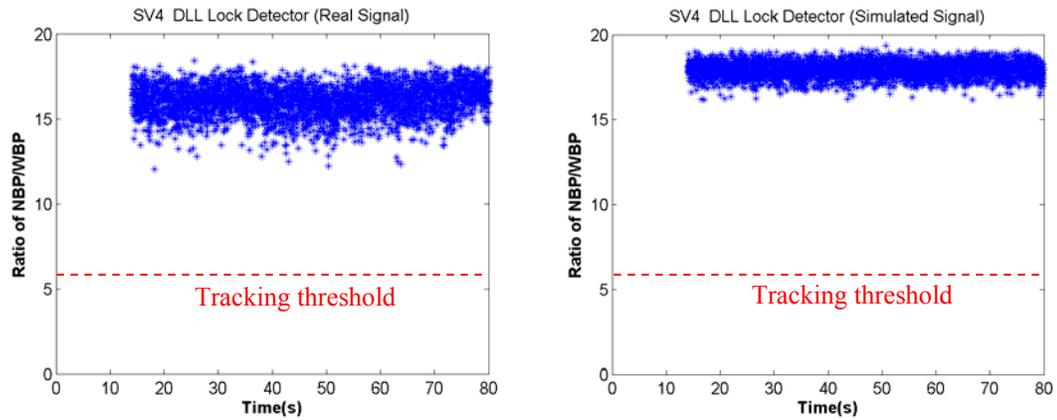


Figure 6.5 (c): DLL lock detector of the real signal and simulated signal for SV4

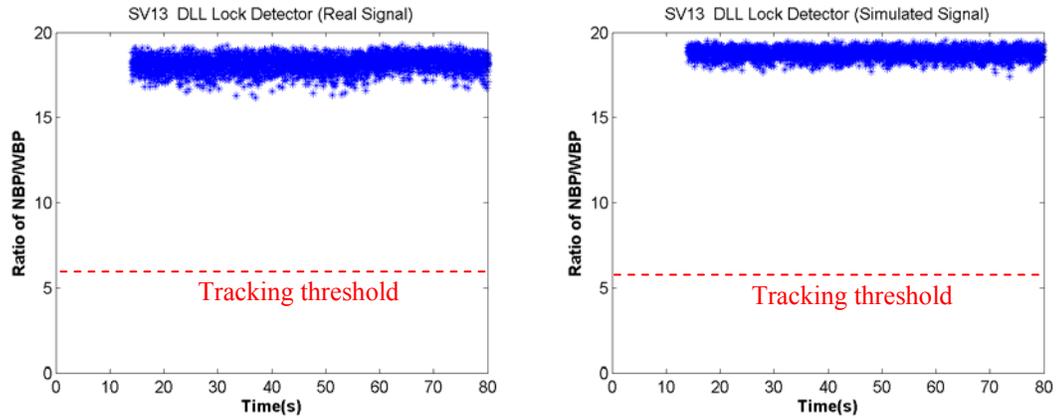


Figure 6.5 (d): DLL lock detector of the real signal and simulated signal for SV13

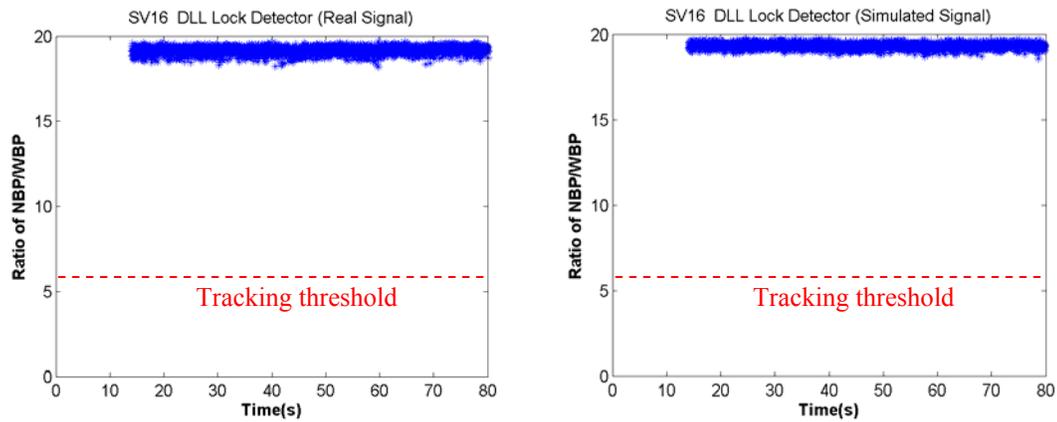


Figure 6.5 (e): DLL lock detector of the real signal and simulated signal for SV16

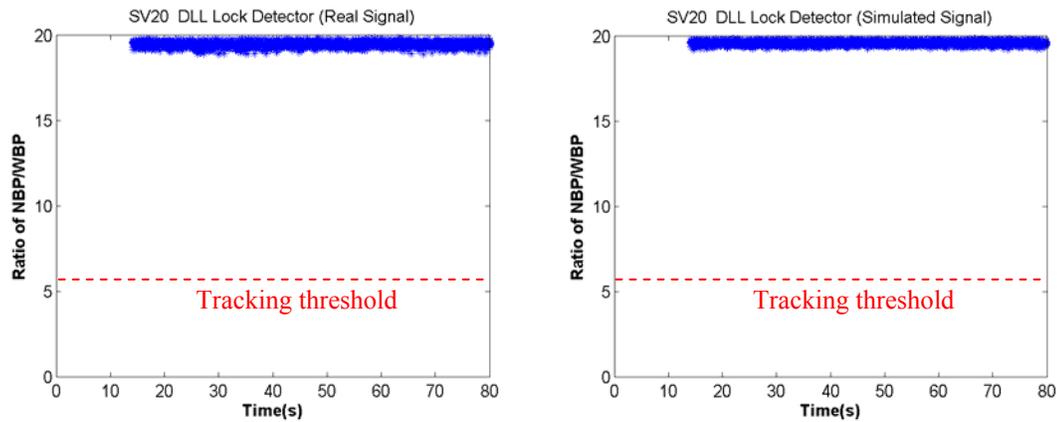


Figure 6.5 (f): DLL lock detector of the real signal and simulated signal for SV20

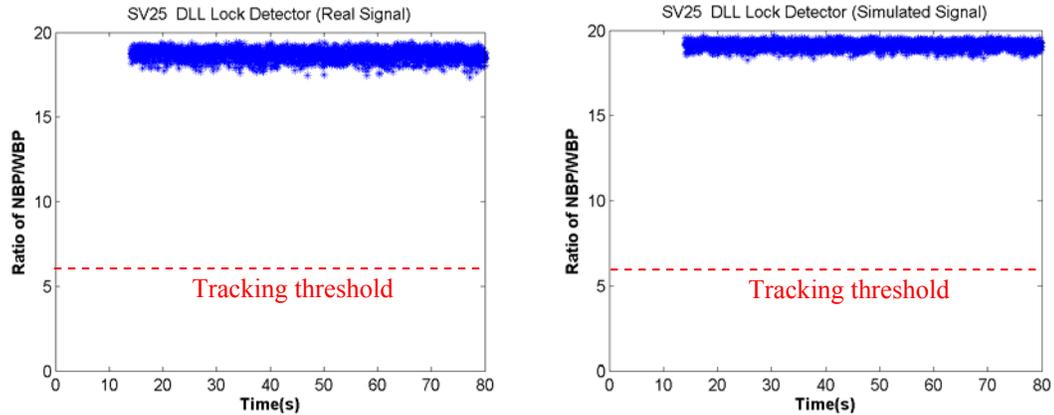


Figure 6.5 (g): DLL lock detector of the real signal and simulated signal for SV25

If the DLL detector threshold is set at a desired minimum C/N_0 , say 26 dB-Hz, the threshold value is 6. If the lock detector is lower than 6, the loop has lost lock. Figure 6.5 indicates that the delay lock loop of both the real and simulated signals maintain tracking. But the loop detector value of the real signal is slightly lower and undergoes more variation than that of the simulated signal. This is because the simulated signal has higher C/N_0 than that of the real signal as shown in Figure 6.4. The delay lock loop detector is tightly related to the estimation of C/N_0 . The same measurement is used to estimate the delay lock loop detector and C/N_0 as shown in Equation 4.29 and 4.30. If the C/N_0 dramatically decreased, DLL will lose lock.

6.5.4 Doppler

The Doppler is a very important feature of the signal which demonstrates the dynamic condition of the user platform. Specifically, it is the carrier phase change rate that gives us this information. The Doppler of all satellites are given in Figure 6.6.

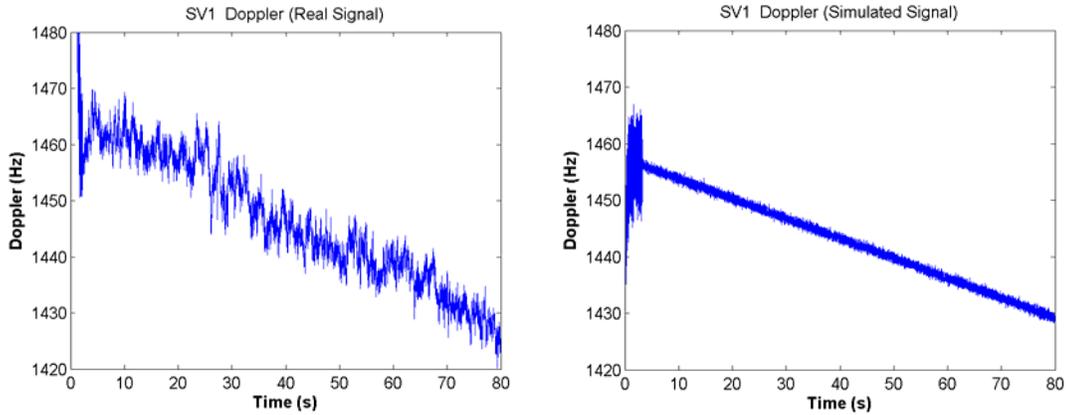


Figure 6.6 (a): Doppler of the real signal and simulated signal for SV1

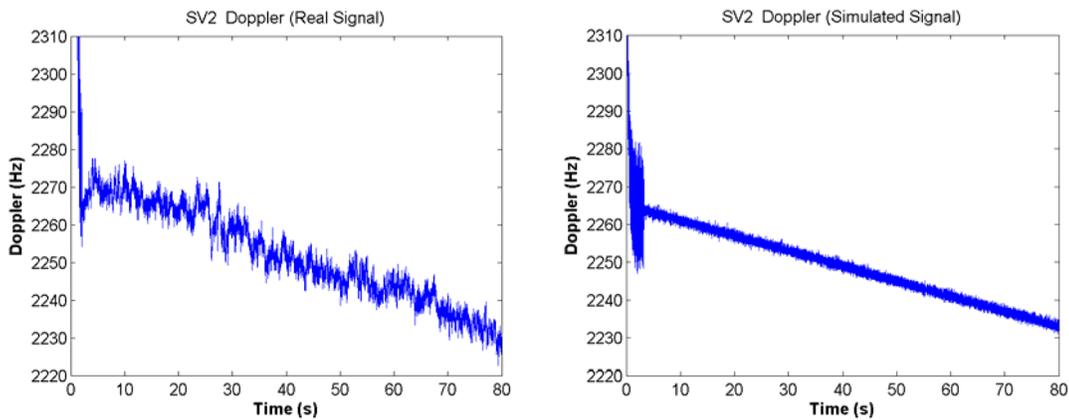


Figure 6.6 (b): Doppler of the real signal and simulated signal for SV2

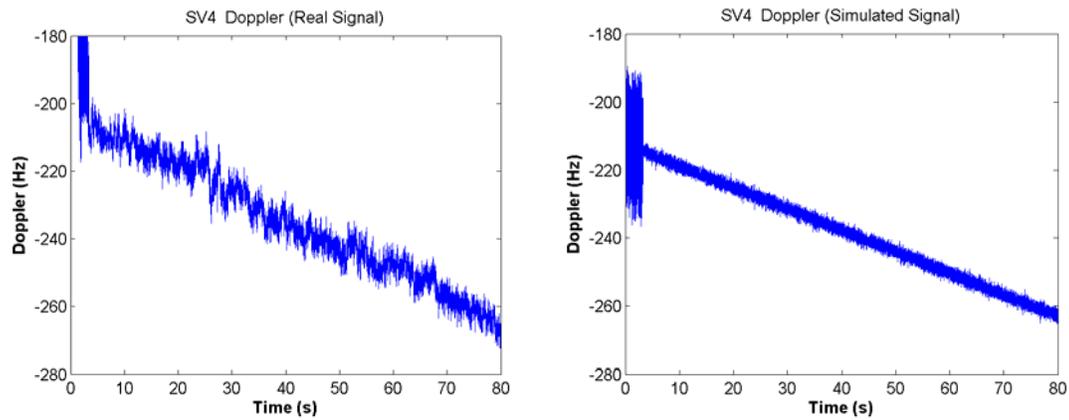


Figure 6.6 (c): Doppler of the real signal and simulated signal for SV4

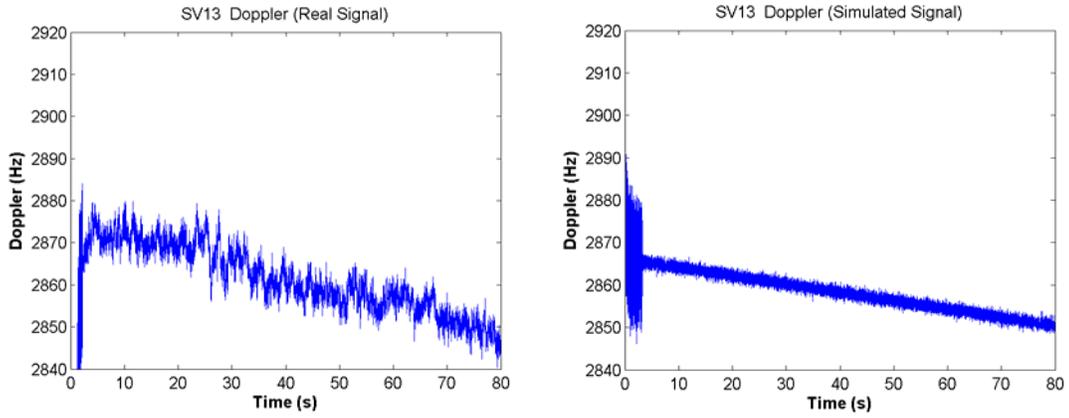


Figure 6.6 (d): Doppler of the real signal and simulated signal for SV13

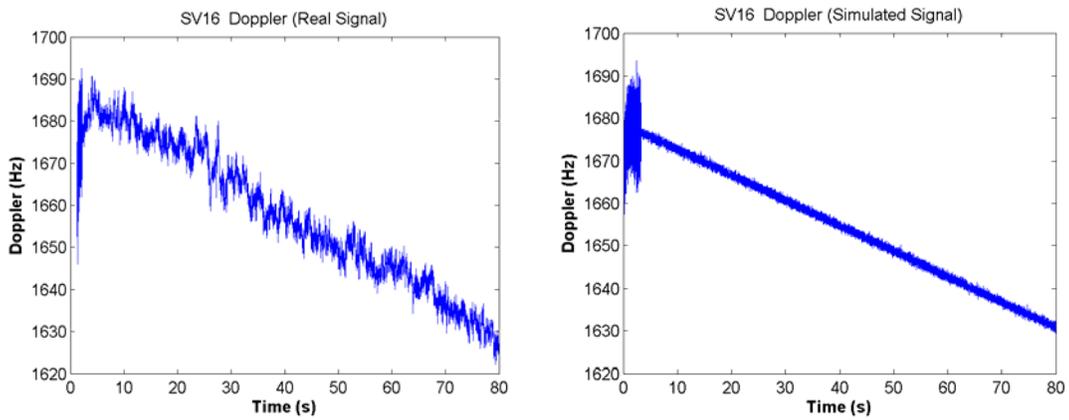


Figure 6.6 (e): Doppler of the real signal and simulated signal for SV16

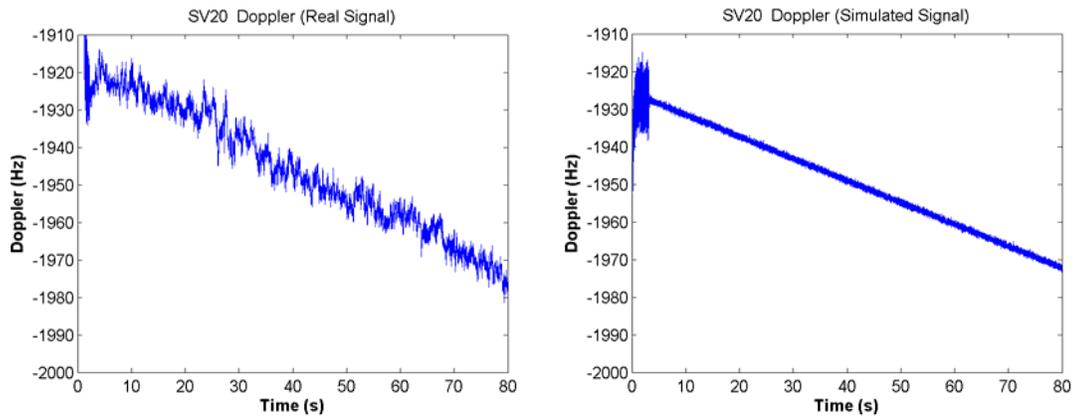


Figure 6.6 (f): Doppler of the real signal and simulated signal for SV20

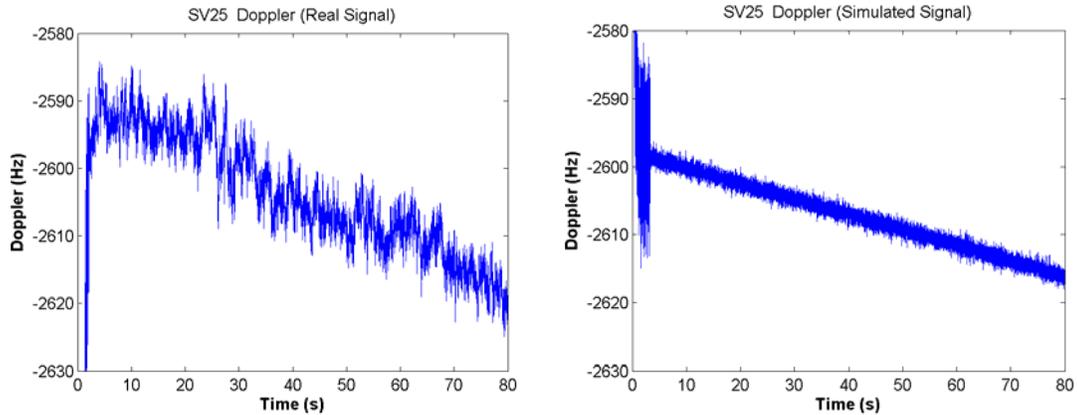


Figure 6.6 (g): Doppler of the real signal and simulated signal for SV25

It can be seen clearly that the resultant Doppler of the real signal and the simulated signal show similar values and same trends over time; however, the former shows more variability. The significant aspect of these comparisons is that the Doppler for the real signal, from the beginning to around 23 s, has a slightly different changing trend than later on. This occurs for each satellite studied, and may be the result of frequency drift of the oscillator in the Signal Tap.

6.5.5 Pseudoranges

The pseudorange is the most important measurement for obtaining navigation solution. Errors in the pseudorange will definitely result in the errors of the computed receiver positions. The following figures show the pseudoranges of all satellites, as derived by the software receiver, on a millisecond scale.

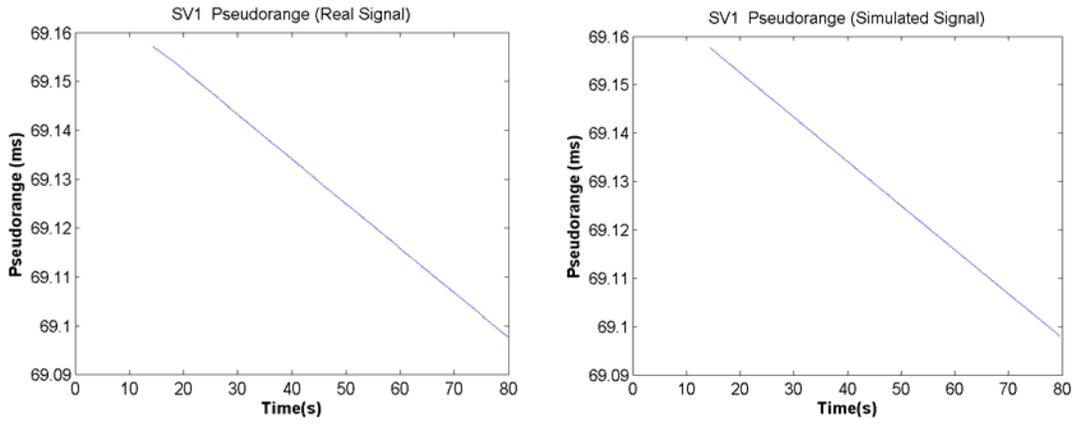


Figure 6.7 (a): Doppler of the real signal and simulated signal for SV1

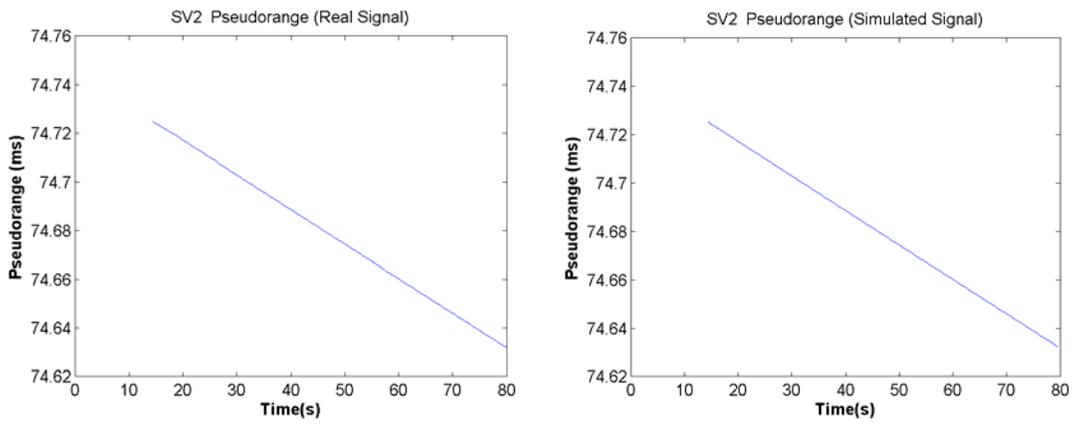


Figure 6.7 (b): Doppler of the real signal and simulated signal for SV2

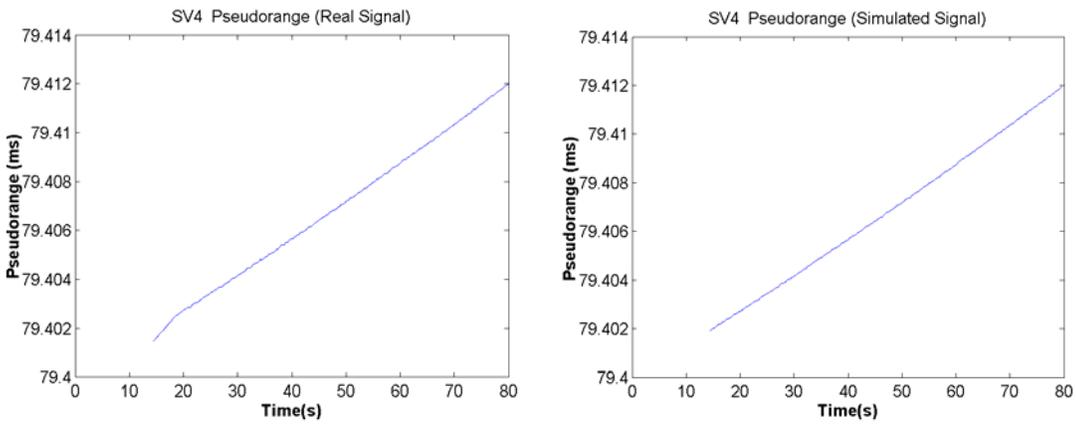


Figure 6.7 (c): Doppler of the real signal and simulated signal for SV4

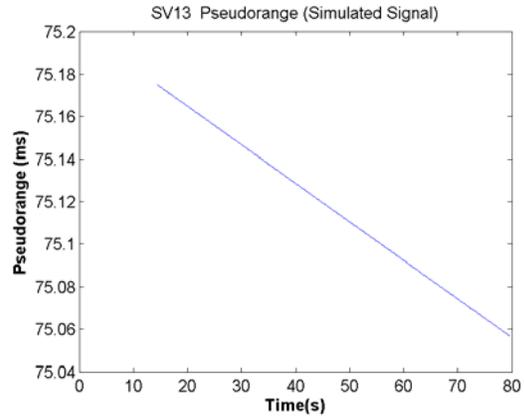
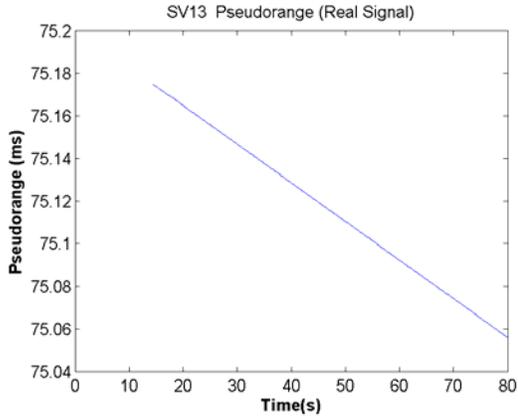


Figure 6.7 (d): Doppler of the real signal and simulated signal for SV13

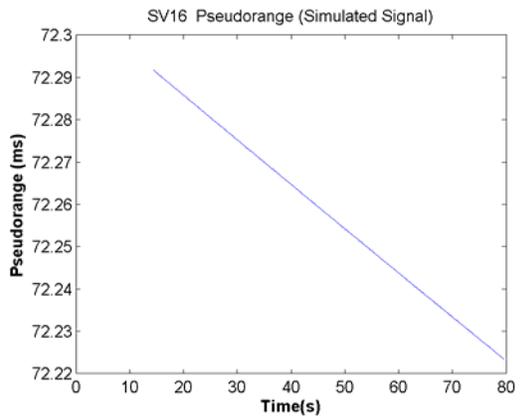
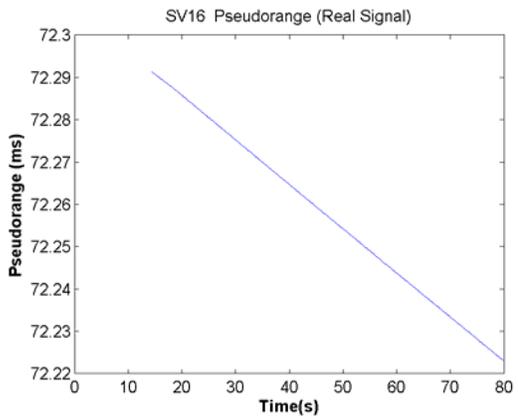


Figure 6.7 (e): Doppler of the real signal and simulated signal for SV16

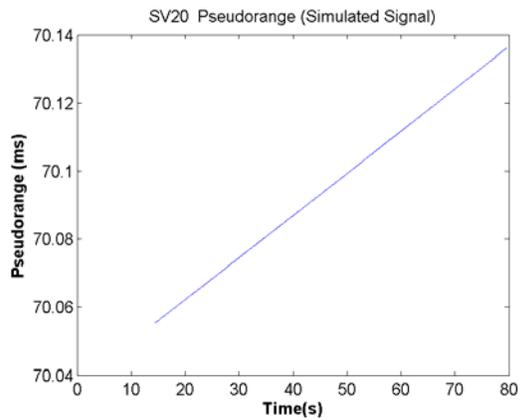
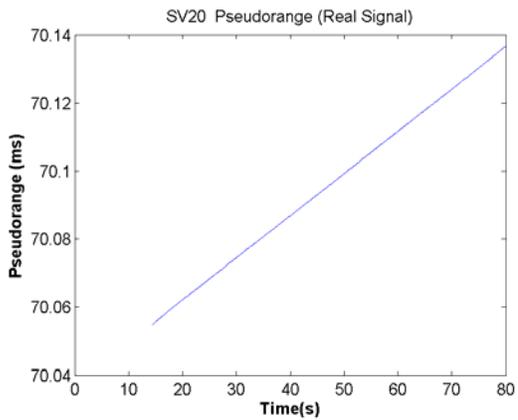


Figure 6.7 (f): Doppler of the real signal and simulated signal for SV20

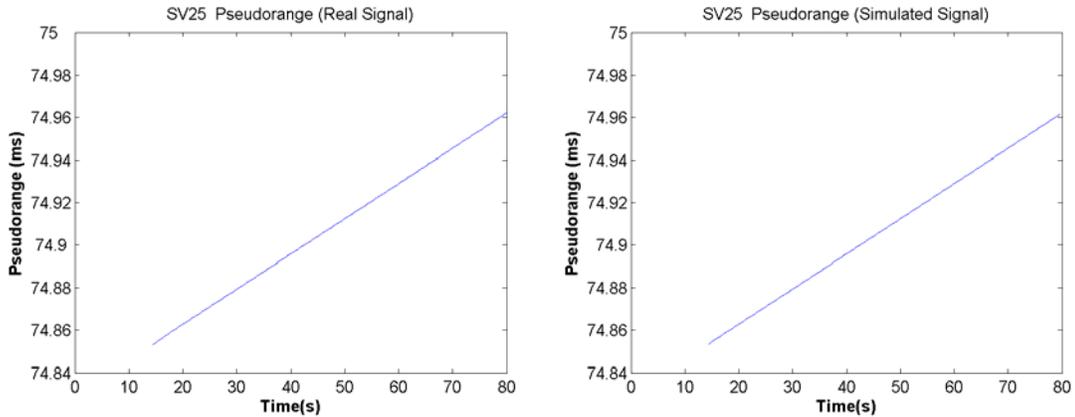


Figure 6.7 (g): Doppler of the real signal and simulated signal for SV25

No obvious differences can be observed between the pseudoranges of the simulated and real signals due to the comparatively coarser scale of the y-axis. But the pseudorange changing trend indicates that the pseudorange is decreasing or increasing with satellite motion. The changing trend is identical to the sign of the satellite Doppler. If the satellite is moving in a direction toward the receiver, the Doppler is positive and the pseudorange is decreasing, and vice versa.

To see clearly the difference between the pseudoranges of the simulated signal and those of the Signal Tap collected signal, one method is to compare them both to the true range between the satellite and the receiver. However, since it is not possible to know this quantity, a simulated theoretical pseudorange, which consists of only the theoretical range between the satellite and receiver and the satellite clock error computed by the satellite clock correction parameters which are provided in the down-loaded ephemeris, is taken as a reference. The resultant pseudoranges of both the real and simulated signals are then compared with it. Figure 6.8 shows such pseudorange differences for all

satellites which are obtained by subtracting the simulated theoretical pseudoranges from the software receiver derived pseudoranges.

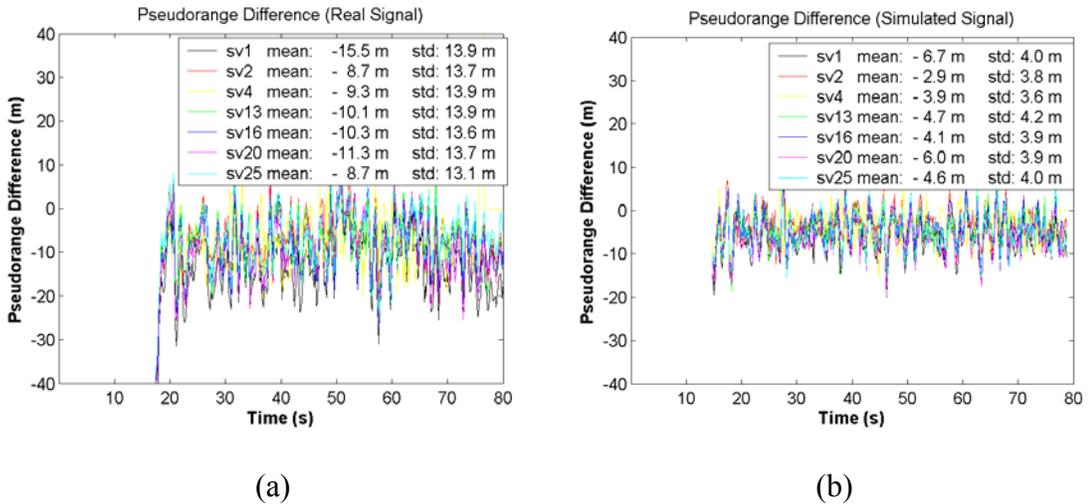


Figure 6.8 Pseudorange differences of the real signal and simulated signal

The values shown in the above figure demonstrate that the real and simulated signal have different pseudorange differences. There are mainly four factors which contribute to this result. Firstly, the simulated ionospheric and tropospheric errors are not exact simulations of the real ones. Secondly, the multipath signals present in the real signal are not simulated. Thirdly, the receiver clock offsets estimated by the software receiver as one of the navigation solution parameters are different. The software receiver makes adjustment according to this receiver clock offset to the output pseudoranges for each computation epoch. This definitely results in the different pseudoranges of the real and simulated signals. Finally, not all errors present in the real signal are included in the simulated signal. The figure 6.8 also shows the standard deviation of the pseudorange differences of the simulated signal is much smaller than that of the real signal. This is because the simulated signal has higher signal-to-noise ratio and therefore less noisy. In addition, the

pseudorange difference for each satellite is also different. This is because the errors, such as ionospheric and tropospheric errors, are different for each satellite and it will result in positioning errors as shown later.

6.6 Verification in Position Domain

The navigation solutions for both the real and simulated GPS signals processed by C³NAV²™ are shown in Figures 6.9 and 6.10, respectively.

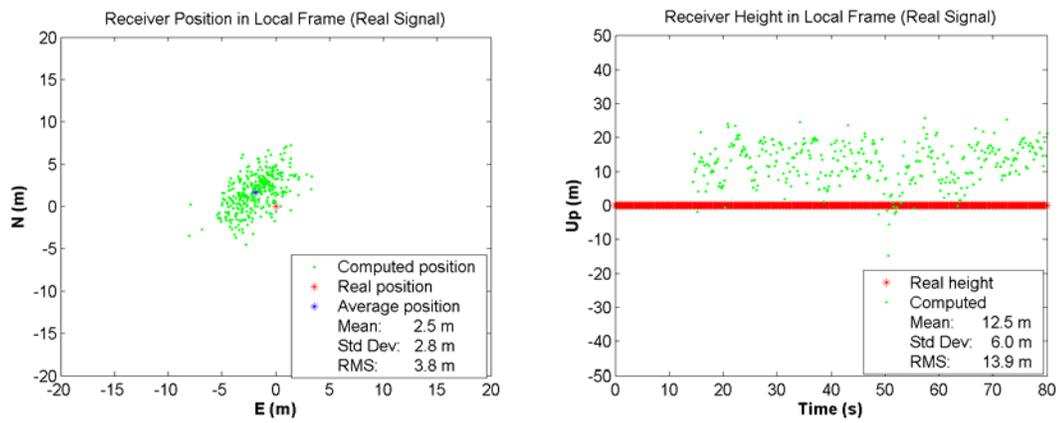


Figure 6.9: Navigation solution of the real GPS signal

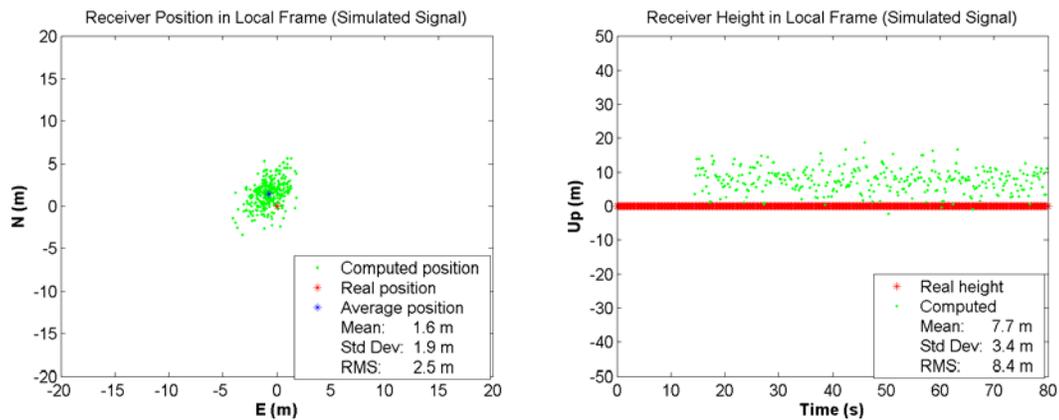


Figure 6.10: Navigation solution of the simulated GPS signal

Comparing the above two figures, the navigation solutions of both the real and simulated signals have a horizontal positioning accuracy better than 5 metres. The result of the simulated signal is even better. The average receiver position derived by the simulated signal is closer than that derived by the real signal and the standard deviation and RMS of the receiver position in Figure 6.10 for the simulated signal are also smaller than the results in Figure 6.9 for the real signal. This indirectly illustrates the impacts of the simulated errors and signal-to-noise ratios. A small position bias can be observed for the solutions of both the real and simulated signals, which results from different pseudorange errors, such as ionospheric and tropospheric errors, for different satellites. And the bias is different for the real and simulated signals. This is because the simulated ionospheric and tropospheric errors are different from the real ones and some interference factors and multipath signal which are present in the real signal are not included in the simulated signal. The higher signal-to-noise ratio of the simulated signal makes the smaller standard deviation for the derived pseudorange and position.

The result for the simulated signal shown in Figure 6.10 is computed with only pseudorange measurements. To check the validity of the carrier phase of the simulated signal, the carrier smooth method is used in C³NAV²™ to calculate the navigation solution. The carrier smoothed result is shown in Figure 6.11.

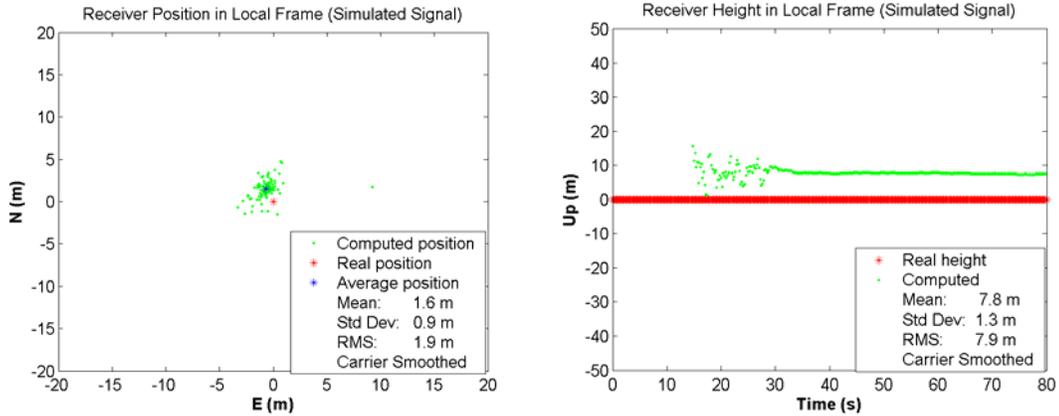


Figure 6.11: Carrier smoothed navigation solution of the simulated GPS signal

Result shown in figure 6.11 shows a significant positioning accuracy improvement. The standard deviation is greatly decreased but the mean value maintains the same as before, which is the typical carrier smoothing behaviour. This verifies the correctness of the simulated carrier phase from another point of the view.

Based on the above analysis, the simulated signal is confirmed to be able to produce correct navigation solution and to have similar properties to the real signal in the position domain.

6.7 Some Other Results for Various Simulation Cases

As described before, the simulated signal is generated by adding noise, and ionospheric and tropospheric errors to the original signal and performing filtering, quantization. All these components can be chosen separately in the simulation. In this section, some results for seven different simulation cases are given to show the effects of the simulation components. The simulation cases include:

- (a) Simulated signal without noise with 1-bit quantization, receiver trajectory mode 0;
- (b) Simulated signal with noise, filtering, and 1-bit quantization, receiver trajectory mode 0;
- (c) Simulated signal with noise, filtering, and 2-bit quantization, receiver trajectory mode 0;
- (d) Simulated signal with noise, ionospheric error, filtering, and 1-bit quantization, receiver trajectory mode 0;
- (e) Simulated signal with noise, ionospheric error, tropospheric error, filtering, and 1-bit quantization, receiver trajectory mode 0;
- (f) Simulated signal with noise, filtering, and 1-bit quantization, receiver trajectory mode 1;
- (g) Simulated signal with noise, filtering, and 1-bit quantization, receiver trajectory mode 2.

For the simulated signal, the theoretical pseudorange, Doppler, and real position are known, which will be used as the references to compare with the results derived from different simulation cases by subtracting these theoretical values from the derived ones. The navigation solution, pseudorange difference, and Doppler difference for all simulation cases are given below.

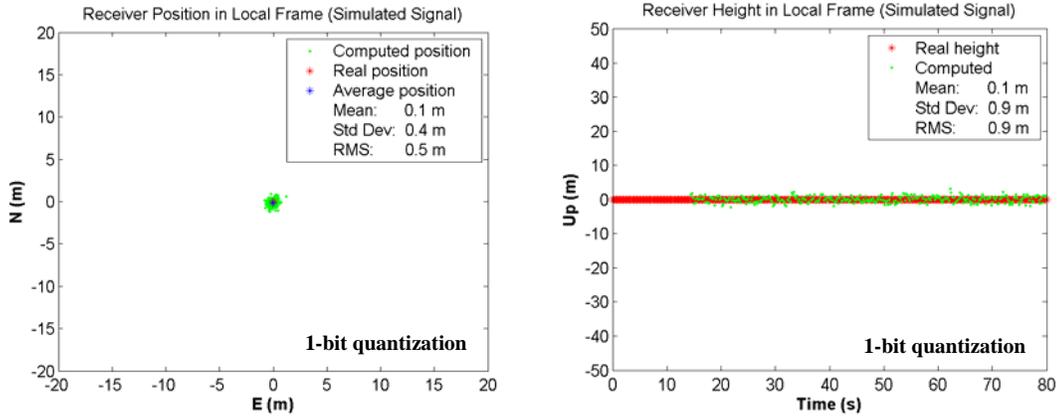


Figure 6.12: Navigation solution of the simulated GPS signal with 1-bit quantization

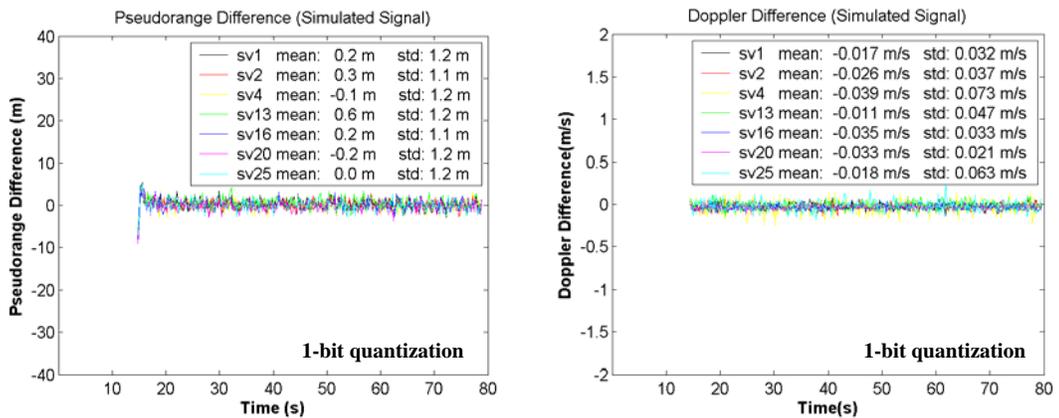


Figure 6.13: Pseudorange and Doppler differences of the simulated GPS signal with 1-bit quantization

Case (a) is a simplest simulation case with only 1-bit quantization applied and no simulated noise. The position RMS 0.5 m can be observed in Figure 6.12. The pseudorange and Doppler difference results are shown in Figure 6.13. The pseudorange difference for each satellite is tens of centimetres and the Doppler difference is several centimetres per second (0.01 m/s corresponding to 0.05 Hz). These errors actually include the effects of both 1-bit quantization and the tracking error of the software receiver.

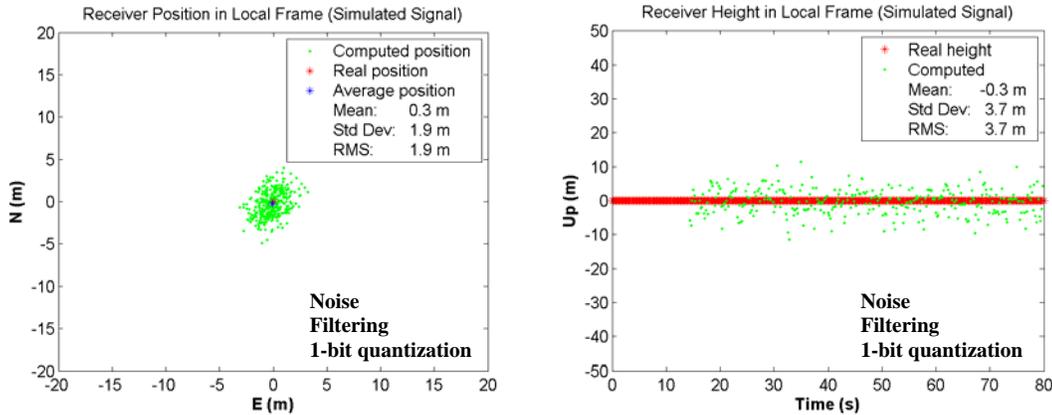


Figure 6.14: Navigation solution of the simulated GPS signal with noise, filtering and 1-bit quantization

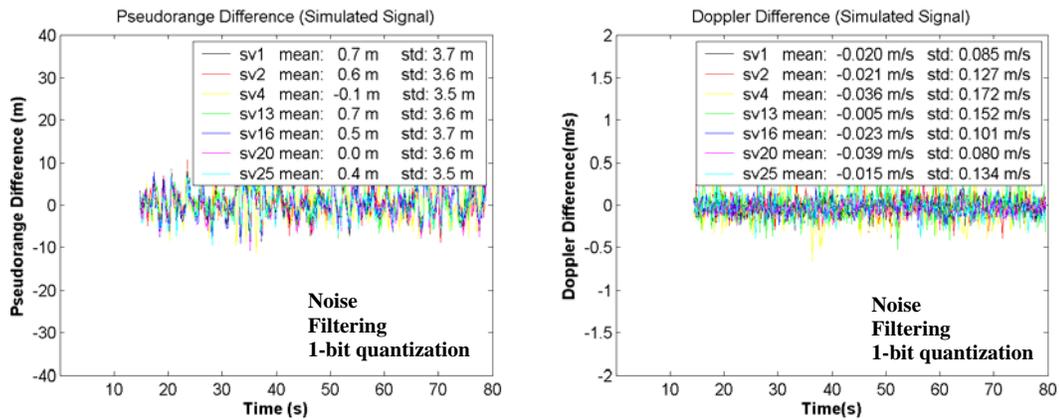


Figure 6.15: Pseudorange and Doppler differences of the simulated GPS signal with noise, filtering and 1-bit quantization

Figure 6.14 and 6.15 show the results for the case (b). The noise is added and the filtering is performed comparing to the case (a). It can be seen that the RMS of the horizontal position is greatly increased from 0.5 m to 1.9 m and standard deviation from 0.4 m to 1.9 m. The standard deviation of the pseudorange difference and Doppler difference are increased as well. These changes demonstrate the effects of the noise and filtering. The presence of the noise definitely reduces the signal-to-noise ratio relating to the case (a) and causes larger tracking error and tracking error variation. Filtering limits out of band

noise in one hand, but filters out a part of signal power of the side lobes in the other hand. However, ninety percent C/A code signal power is in the 2 MHz main lobe (Raquet, 2001). Therefore the application of 2 MHz filtering in this simulation limits the noise but has no significant effect on the power of the C/A code signal.

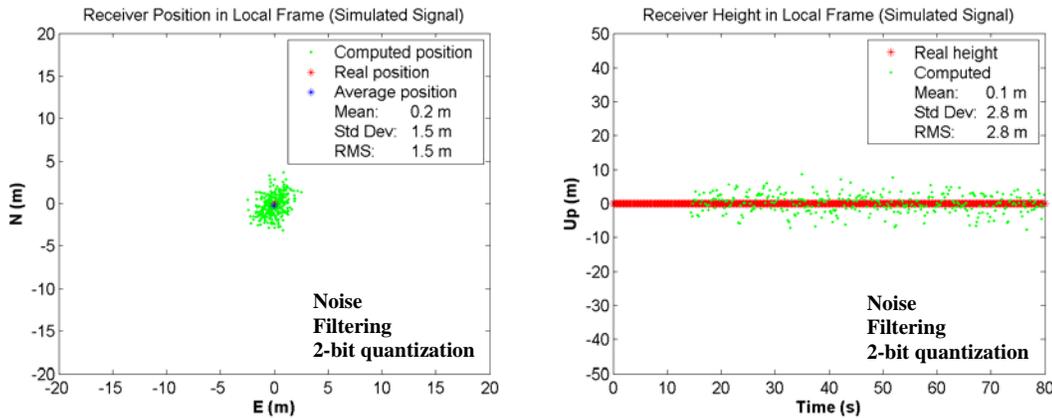


Figure 6.16: Navigation solution of the simulated GPS signal with noise, filtering and 2-bit quantization

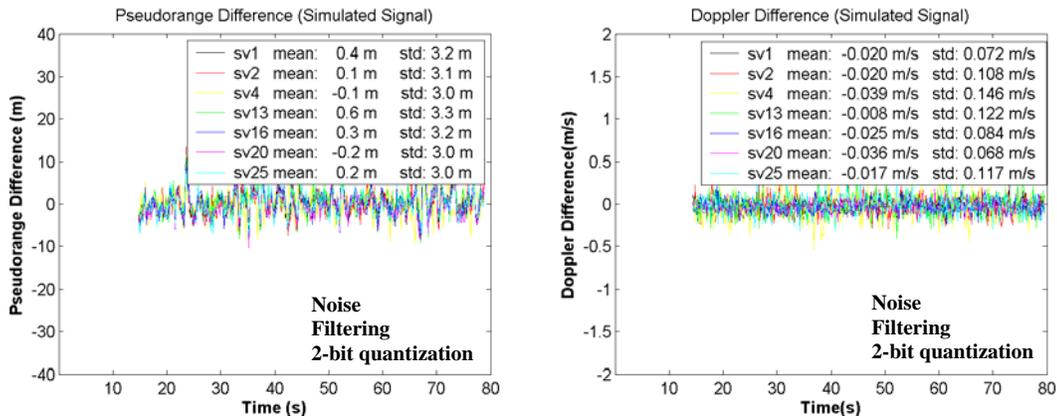


Figure 6.17: Pseudorange and Doppler differences of the simulated GPS signal with noise, filtering and 2-bit quantization

To check the effect of 2-bit quantization, a simulated signal with noise, filtering, and 2-bit quantization is generated in case (c), the results of which are shown in Figure 6.16 and

6.17. Comparing with case (b), the standard deviation of position, pseudorange difference and Doppler difference are all reduced. This demonstrates the 2-bit quantization causes less quantization loss.

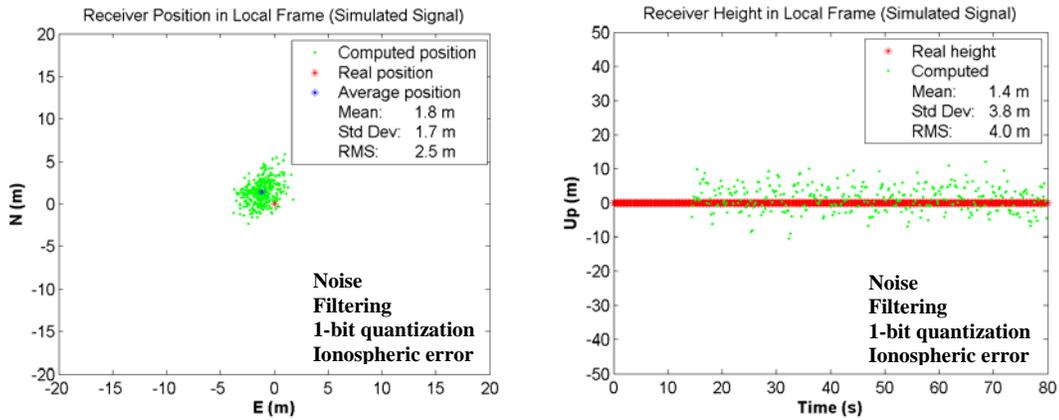


Figure 6.18: Navigation solution of the simulated GPS signal with noise, filtering, 1-bit quantization, and ionospheric error

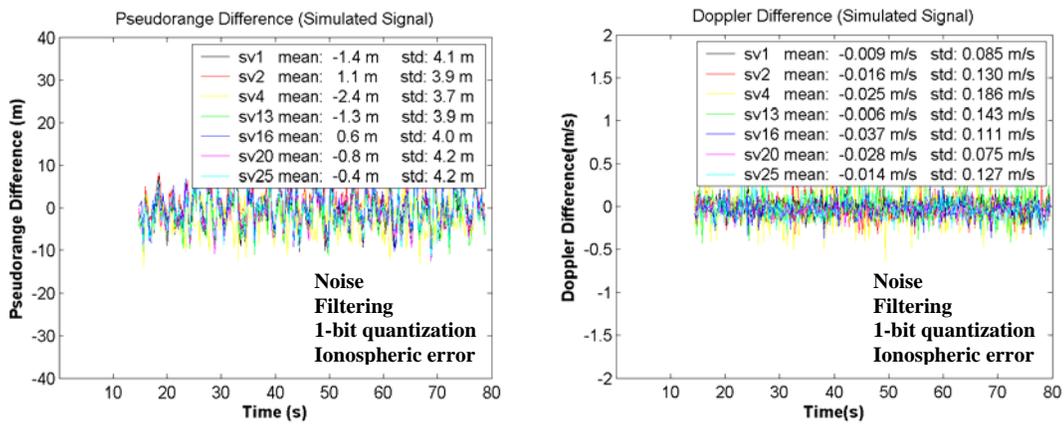


Figure 6.19: Pseudorange and Doppler differences of the simulated GPS signal with noise, filtering, 1-bit quantization, and ionospheric error

For the case (d), the ionospheric error is added in addition to noise, filtering, and 1-bit quantization. A position bias can be clearly seen in Figure 6.18. This is because the different pseudorange variations of the different satellites which are resulted by different

simulated ionospheric errors are applied for the different satellites. Theoretically, the pseudorange differences as shown in Figure 6.19 should be the summation of the values in Figure 6.15 and the simulated ionospheric errors. But it is not true. This is because the software receiver conducts adjustment to the output pseudorange according the computed common receiver clock offset of all satellites for each computation epoch. Therefore, an offset value should be taken into account to check the validity of the pseudorange differences. So, the pseudorange differences for the simulated signal with ionospheric errors as shown in Figure 6.19 should be the summation of the values for the simulated signal without ionospheric errors as shown in Figure 6.15, the simulated ionospheric errors, and a common offset value. But the common offset value is not known. In Table 6.4, it is computed for each satellite according other three known values. It can be seen that the common offset value is almost the same for all satellites. This demonstrates the relationship described above stands and the results shown in case (d) are correct.

Table 6.4 Pseudorange differences of the simulated signals and simulated ionospheric errors

PRN	Pseudorange difference		Simulated ionospheric error (m)	Common offset value (m)
	Simulated signal without ionospheric error (m)	Simulated signal with ionospheric error (m)		
1	0.7	- 1.4	8.8	10.9
2	0.6	1.1	11.6	11.1
4	- 0.1	- 2.4	8.1	10.4
13	0.7	- 1.3	8.8	10.8
16	0.5	0.6	10.8	10.7
20	0.0	- 0.8	10.0	10.8
25	0.4	- 0.4	10.1	10.9

The results of case (e) have already shown in Figure 6.10. Comparing to case (d), the only difference of the two cases is that the tropospheric error is simulated in the case (e). However, the height biases for the two cases differ by 6.3 m. This difference is resulted by the tropospheric error as no tropospheric model is applied when using C³NAV²™ to derive the result of case (e). When the tropospheric error model is applied, this bias is gone as shown in Figure 6.20.

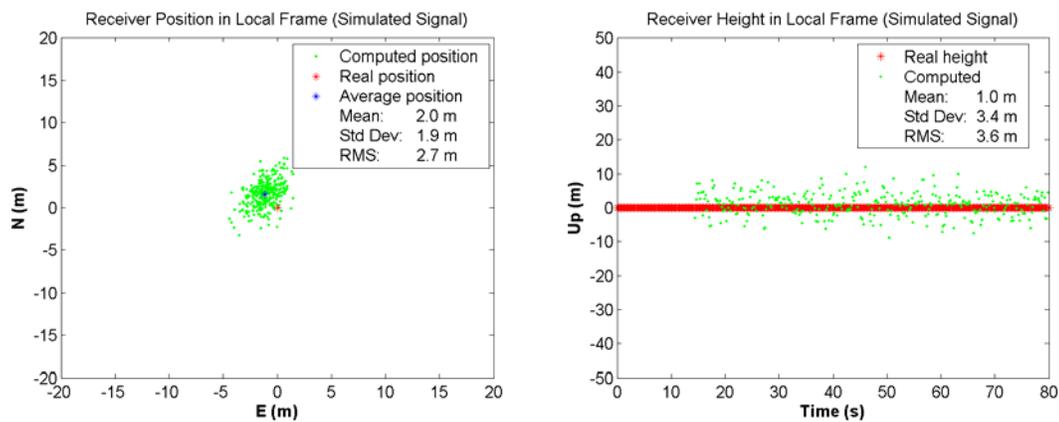


Figure 6.20: Navigation solution of the simulated GPS signal with tropospheric error model applied in C³NAV²™

So far, all results and analysis are for the static receiver trajectory mode 0. The following figures show the navigation solutions, pseudorange differences, and Doppler differences for the receiver trajectory mode 1 and 2. Figure 6.21 and 6.22 are the results of the simulated signal with noise, filtering, 1-bit quantization, ionospheric and tropospheric errors for the receiver trajectory mode 1 and Figure 6.23 and 6.24 are for the receiver trajectory mode 2.

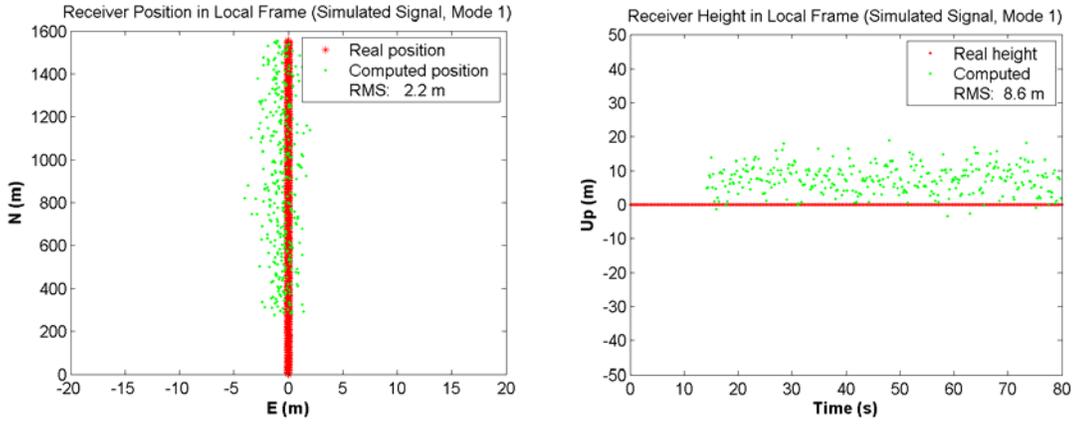


Figure 6.21: Navigation solution of the simulated GPS signal (receiver trajectory mode 1)

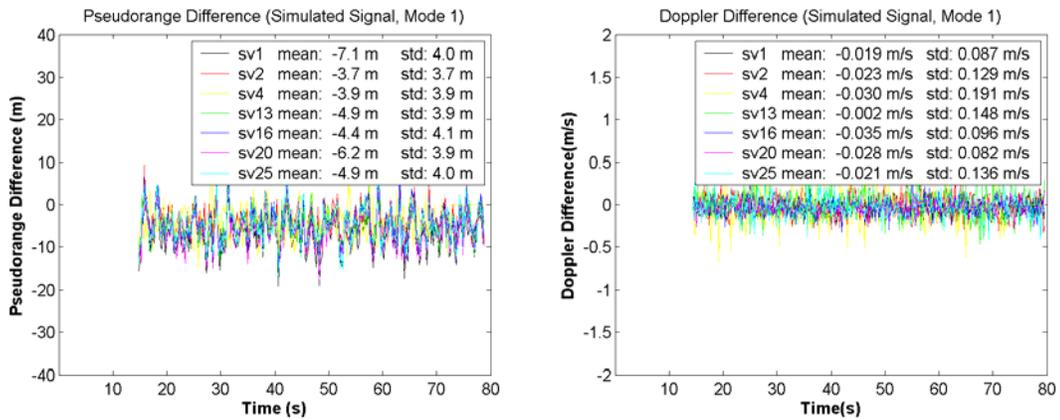


Figure 6.22: Pseudorange and Doppler differences of the simulated GPS signal (receiver trajectory mode 1)

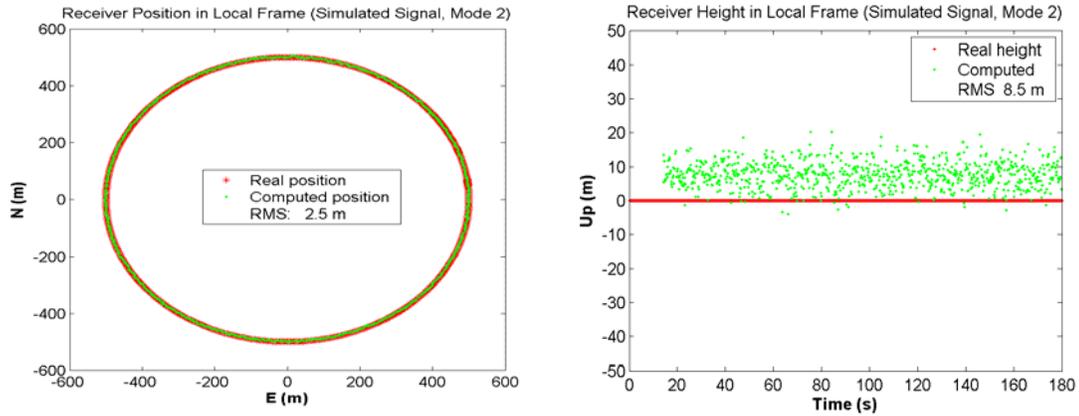


Figure 6.23: Navigation solution of the simulated GPS signal (receiver trajectory mode 2)

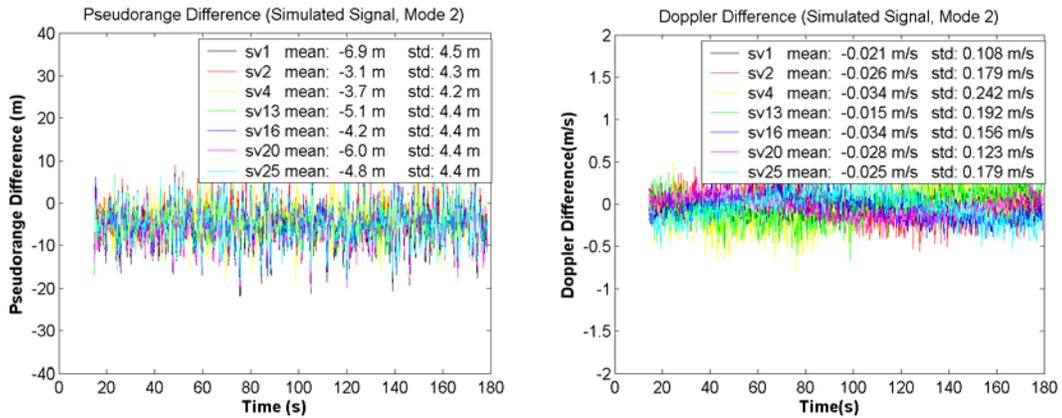


Figure 6.24: Pseudorange and Doppler differences of the simulated GPS signal (receiver trajectory mode 2)

It can be seen that the navigation solutions for both receiver trajectory mode 1 and mode 2 follow the simulated trajectory. The RMS of the positions, mean values, and standard deviations are in the same level as the results for receiver trajectory mode 0 shown in Figure 6.8 (b) and 6.10. This demonstrates the validity of the simulation for trajectory mode 1 and mode 2.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

7.1.1 Software IF GPS Signal Simulator

A GPS signal simulator has been developed and verified in this thesis. The simulator is based on a mathematical model at an intermediate frequency. The desired simulation time, receiver trajectory, and other parameters such as the selected intermediate frequency, sampling frequency, filtering bandwidth, C/N_0 , and noise density are the required inputs that drive the simulator. Both filtering and quantization processing steps were implemented. The satellite clock error, ionospheric and tropospheric errors were simulated, with the simulator outputting a quantized bit stream. All of the above components can be easily altered or conveniently selected or rejected in a parameter definition file. The size of the computation burden was also optimized, by developing simulator on MATLAB, with some heavy computational load functions developed in C code to accelerate the simulator running speed. The simulator currently requires 100 seconds to generate 1 second of IF GPS signal.

The simulator has been verified by comparing the simulated signal and the Signal Tap observed signal in the time, frequency, measurement, and position domains. Acquisition and tracking results were also compared to prove the correctness of the simulator. It was evident from this comparison that the simulated IF GPS signal has the same properties as those of the hardware front-end collected real IF GPS signal in all respects. Therefore, this confirmed that the simulator-generated signal can accurately duplicate the true signal. Further, it demonstrates that the simulator itself can act as an effective and versatile tool for GPS receiver studies since it is easily adapted to generate not only the GPS signals that have already been implemented but also the signals that will be implemented in the future, such as GPS L5 and Galileo signals.

7.1.2 Software GPS Receiver

A software GPS receiver was also developed and verified in this thesis. It is a multi-channel receiver that operates in post-mission. Like a conventional receiver, it includes several functional blocks: acquisition, code and carrier phase tracking, bit synchronization, sub-frame synchronization, and navigation solution computation. This receiver is driven by the digitized IF GPS signal. An inner clock based on the incoming IF GPS signal samples is generated as the time reference because no physical clock is available in the software receiver as in existing hardware receivers. The navigation data bits, from which the satellite ephemeris can be acquired, are demodulated in the phase lock loop. The pseudorange and carrier phase measurements can be derived from the tracking loop with the help of bit synchronization and sub-frame synchronization. Finally, receiver positions in the WGS84 system are output from the software receiver.

The software receiver has been verified by testing against real GPS signals. The real GPS signal collected via the Signal Tap was fed to the receiver. Various tracking results, carrier phase and pseudorange measurements, and position solution were logged to verify the receiver correctness. For example, the Doppler derived from signal acquisition and signal tracking indicates that it can track the carrier correctly; the resulting navigation solutions demonstrate that the receiver can obtain satisfactory pseudorange measurements and provide an accurate receiver position solution. From these test results, it can be safely concluded that the software receiver operates properly and can be used for the simulated GPS signal verification.

7.2 Future Work

7.2.1 Future Work for the Software IF GPS Signal Simulator

In addition to the design principles, implementation factors, and verification processes that have been described in this thesis, the following items are recommended for further study:

[1] A receiver clock error model should be simulated. It is one of the major error sources in a conventional receiver, caused by the instability of the oscillator. It has a significant impact on maintaining coherent signal tracking.

[2] Multipath should be simulated. In an actual positioning environment, the received signal is the combination of direct and multipath signals. Strong multipath signals will deform the autocorrelation function and mislead the delay lock loop to track the wrong autocorrelation, potentially resulting in a tracking error in the order of tens of metres.

[3] The simulator in this thesis is based on the L1 C/A code signal. It thus constitutes only a basic application of signal simulation. New signals, such as the L5 signal and even the Galileo signal should be simulated to study their behavior.

[4] Simulation of interference would be a valuable addition to this research. A software method such as this one allows a great degree of flexibility in interference spectrum analysis.

[5] Realizing the simulator using C++ would further enhance its efficiency, due to C++'s capability to accelerate simulation speed. Developing a graphical user interface to make the simulator easier to operate would also be valuable.

7.2.2 Future Work for the Software GPS Receiver

The software GPS receiver used in the simulated signal verification is only a basic version because it fulfilled only the core part of a conventional receiver, such as signal acquisition, signal tracking, raw measurement derivation, and positioning fixing. To further improve receiver performance, the following tasks should be conducted in the future:

[1] Improve the tracking accuracy. The parameters of the FLL, PLL, and DLL should be adjusted based on automatic control and phase lock loop theory to decrease the tracking loop errors. The parameters should be changed adaptively with respect to the receiver status, enhancing its capability to handle either low or high dynamics, and to accommodate light or heavy signal attenuation.

[2] Apply external aiding to the receiver to remove large dynamics or other major error sources to improve receiver performance.

[3] Apply interference mitigation methods to improve receiver performance in hostile environments.

[4] Upgrade the receiver to a full GNSS receiver which can process not only the GPS C/A signal but also GPS L2, GPS L5, and Galileo signals.

REFERENCES

Accord, (December 8, 2002), <http://www.accord-products.com/prodgsim.htm>

Akos, D. M. (1997), *A Software Radio Approach To Global Navigation Satellite System Receiver Design*, Dissertation, Ohio University.

Amoroso, F. (1983), *Adaptive A/D Converter to suppress CW Interference in DSPN Spread-Spectrum Communications*, IEEE 1983.

Axelrad, P., R. G. Brown (1996), *Global Positioning System: Theory and Applications*, Volume I, *Chapter 9: GPS Navigation Algorithms*, University of Colorado 80309, Iowa State University, Ames, Iowa 50010. Inc.: 409-433.

Brown, A., N. Gerein, and K. Taylor (2000), *Modeling and simulation of GPS Using Software Signal Generation and Digital Signal Reconstruction*, Proceedings of ION NTM, Anaheim, CA, pp.646-652.

Burns, J. R., C. Cutright, and M. Braasch (2002) *Investigation of GPS Software Radio Performance in Combating Narrow Band Interference*, ION 58th Annual Meeting/CIGTF 21st Guidance Test Symposium, Albuquerque, NM, pp.523-530.

Black, H. D., A. Eisner (1984), *Correcting Satellite Doppler Data for Tropospheric Effects*, Journal of Geophysical Research, Vol. 89, No. D2: 2616-2626.

Corbell, P. M., M. M. Miller (2000) *Design and Analysis of a Matlab Based Digitized IF GPS Signal Simulator and A Simulink Based Configurable GPS Receiver*, Proceedings of ION GPS, Salt Lake City, UT, pp. 1906-1915.

Corbell, P. M. (2000) *Design and Validation of an Accurate GPS Signal And Receiver Truth Model For Comparing Advanced Receiver Processing Techniques*, Thesis for master degree, Department of Air Force, Air University.

Corbell, P. M., M. M. Miller (2001) *A Configurable GPS Accumulated I and Q Signal Component Simulator in Matlab*, Proceedings of ION NTM, Long Beach, CA, pp. 205-215.

Christie, J. R., P. Ko, A. Hansen, D. Dai, S. Pullen, B. S. Pervan, B. W. Parkinson (1999), *The Effects of Local Ionospheric Decorrelation on LAAS: Theory and Experimental Results*, Proceedings of ION National Technical Meeting, the Institute of Navigation, Alexandria, VA: 769-777.

CRS, (November 12, 2002), <http://www.centerforremotesensing.com/press/gps.pdf>

FAA (1997), Specification for the Wide Area Augmentation System, FAA-E-2892C.

Georgiadou, Y., A. Kleusberg (1988), *On Carrier Signal Multipath Effects in Relative GPS Positioning*, Manuscripta Geodaetica, Vol. 13, No. 3: 172-179.

Goldhirsh, J., W. J. Vogel (1989), *Mobile Satellite System Fade Statistics for Shadowing and Multipath from Roadside Trace at UHF and L-Band*, IEEE Transactions on Antennas and Propagation, Vol. 37, No. 4: 489-498.

Hopfield, H. S. (1969), *Two-Quartic Tropospheric Refractivity Profile for Correcting Satellite Data*, Journal of Geophysical Research, Vol. 74, No. 18: 4487-4499.

Kaplan, E. D. (1996), *Understanding GPS, Principles and Applications*, Boston: Artech House, Inc.

Krumvieda K., P. Madhani et al (2001), *A Complete IF Software GPS Receiver: A Tutorial about the Details*, ION GPS 2001, 11-14 September, 2001, Salt Lake City, UT.

Lachapelle, G. (1997), *ENGO 625 GPS Theory and Applications*. Department of Geomatics Engineering, the University of Calgary.

Ledvina, B.M., M. L. Psiaki et al (2003), *A 12-Channel Real-Time GPS L1 Software Receiver*, ION NTM.

Lin, D.M., Tsui, J. (2000), *Comparison of acquisition methods for software GPS receiver*, Proceeding of ION GPS 2000, 19-22 September, 2000, Salt Lake City, UT, pp. 2385-2389.

Luo, N. (2001), *Precise Relative Positioning of Multiple Moving Platforms Using GPS Carrier Phase Observables*, Ph.D. thesis, UCGE reports 20147, the University of Calgary.

Miller, M. M., P. M. Corbell (2000) *Design and Validation of Digitized Intermediate Frequency GPS Signal and Receiver Software Models for Developing and Comparing Advanced GPS Receiver Technologies*, Proceedings of ION GPS, Salt Lake City, UT, pp. 2362-2371.

- Maybeck, P. S. (1994), *Stochastic Models, Estimation and Control*, I. New York: Academic Press, Inc. Republished, Arlington, VA: Navtech.
- McDonald, K. D. and C. Hegarty (2000), *Post-Modernization GPS Performance Capabilities*, ION GPS 2000 Meeting, Salt Lake City UT, 19-22 Sep 2000
- Misra, P. and P. Enge (2001) *Global Positioning System - Signals, Measurements, and Performance*, Ganga-Jamuna Press, Massachusetts, 2001.
- NAVSYS, (November 12, 2002), <http://www.navsys.com/Products/toolbox.htm>
- NAVTECH,(December8, 2002), http://www.navtechgps.com/supply/correlator_simulator
- Parkinson, B.W. (1996), *Global Positioning System: Theory and Applications*, Volume I, *Chapter 11: GPS Error Analysis*, Stanford University, Stanford, California 94305. Inc.: 469-483.
- Petovello, M. (2000), *C³NAVIG²TM Operating Manual*, Department of Geomatics Engineering, The University of Calgary.
- Peterson, R.L., R. E. Ziemer, D. E. Borth (1995), *Introduction to Spread Spectrum Communications*, Englewood Cliffs, NJ: Prentice Hall.
- Raquet, J. F. (2001), *ENGO 699.64 Lecture Note: GPS Receiver Design*, Department of Geomatics Engineering, The University of Calgary.
- Ray, J. K. (2000), *Mitigation of GPS Code and Carrier Phase Multipath Effects using a Multi-Antenna System*, Ph.D. thesis, UCGE Report No. 20136, February 2000.

RTCA (1998), *Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment*, RTCA/DO-229A, SC-159. Washington, DC.

Schaer, S. (1997), *How to Use CODE's Global Ionosphere Maps*, Astronomical Institute, University of Berne.

Seitz, A. C., and M. S. Braasch (1998), *High Fidelity GPS Receiver Simulation*, ION Proceedings of the 54th Annual Meeting, The Institute of Navigation, Colorado, pp. 573-581.

Spilker Jr., J.J. (1996a), *Global Positioning System: Theory and Applications, Volume I, Chapter 3: GPS Signal Structure and Theoretical Performance*, Stanford Telecom, Sunnyvale, California 94089. Inc.: 57-119.

Spilker Jr., J.J. (1996b), *Global Positioning System: Theory and Applications, Volume I, Chapter 4: GPS Navigation DATA*, Stanford Telecom, Sunnyvale, California 94089. Inc.: 121-176.

Spilker Jr., J.J. (1996c), *Global Positioning System: Theory and Applications, Volume I, Chapter 13: Tropospheric Effects on GPS*, American Institute of Aeronautics and Astronautics. Inc.: 517-546.

Tsui, James B-Y. (2000), *Fundamentals of Global Positioning System Receivers: A Software Approach*, John Wiley & Sons Inc.

Van Dierendonck, A.J. (1996), *Global Positioning System: Theory and Applications*, Volume I, *Chapter 8: GPS Receivers*, AJ Systems, Los Altos, CA 94024. Inc.: 329-407.

Van Vee, R.D.J., Coenen. A.J.R.M. (1991), *New Fast GPS Code-Acquisition Technique Using FFT*, *Electronic Letters*, Vol. 27, pp.158-160, Jan. 1991.