

國立交通大學

電信工程學系碩士班 碩士論文

適用於一位元量化 GPS 接收機之 新型載波追蹤方法

A Novel Carrier Tracking Scheme for One-bit Quantized GPS Receiver

研究生：姜琇云

Student: Hsiu-Yun Chiang

指導教授：高銘盛 博士

Advisor: Dr. Ming-Seng Kao

中華民國九十七年六月

適用於一位元量化 GPS 接收機之

新型載波追蹤方法

A Novel Carrier Tracking Scheme for

One-bit Quantized GPS Receiver

研究生：姜琇云

Student: Hsiu-Yun Chiang

指導教授：高銘盛 博士

Advisor: Dr. Ming-Seng Kao

國立交通大學

電信工程學系碩士班

碩士論文

A Thesis

Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Communication Engineering

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

適用於一位元量化 GPS 接收機之 新型載波追蹤方法

學生：姜琇云

指導教授：高銘盛 博士

國立交通大學電信工程學系碩士班

摘要

在現代全球定位系統中，軟體接收機 (software receiver) 為一廣受矚目的技術。有別於一般追蹤程序 (tracking process)，本論文探討如何精確估算數控振盪器 (NCO) 的頻率，以期改善衛星訊號追蹤迴路之效能。針對地面運輸工具的使用環境，我們提出利用特定時間的衛星訊號數據來估算精確的頻率偏移之方法。利用調整 NCO 初始頻率和相位，從模擬結果顯示載波頻率的誤差可以有效地降低至 ± 10 Hz 之內。


在載波頻率精確度大幅提升的同時，必須考慮儲存 NCO 載波對照表的記憶體空間需求。有別於事先建立的載波對照表，我們提出一種僅使用簡單的數值運算，稱作“虛擬載波對照表”的演算法。利用所提的演算法，我們成功地只使用極少的記憶體空間產生所需的虛擬載波對照表。

A Novel Carrier Tracking Scheme for One-bit Quantized GPS Receiver

Student: Hsiu-Yun Chiang Advisor: Dr. Ming-Seng Kao

Department of Communication Engineering
National Chiao Tung University

Abstract



It is well known that software receiver is very attractive in modern GPS. In this thesis, we focus on the improvement of the tracking loop performance with the fine frequency estimation for numerical controlled oscillator (NCO). For land vehicle users, we propose a frequency offset measurement scheme that uses the first few data sets to estimate the frequency offset. By adjusting the initial phase and frequency of NCO, the simulation results show that the carrier frequency resolution is within ± 10 Hz.

Since the frequency resolution has been enhanced, the memory space to store the look-up table of NCO is of concern. Instead of pre-built carrier table, a simple algorithm using numerical operation called “virtual carrier table” generator is proposed. Using the proposed approach, we have successfully generated the virtual carrier table, which requires very small memory space.

Acknowledgement

I would like to express my deepest gratitude to my advisor, Dr. Ming-Seng Kao, for his enthusiastic guidance and great patience. I learn a lot from his positive attitude in many aspects. Heartfelt thanks are also offered to Dr. Jeff Chang in National Space Organization for his support on research. Besides, I deeply appreciate the constant encouragement from my college classmates in the Communication System Design and Signal Processing (CSDSP) Lab. Finally, I would like to show my sincere thanks to my parents for their invaluable love.



Contents

Chinese Abstract	I
English Abstract	II
Acknowledgement	III
Contents	IV
List of Figures	VI
List of Tables	VIII
Acronym Glossary	IX
Notations	X
Chapter 1 Introduction	1
Chapter 2 Global Positioning System	5
2.1 GPS receiver	5
2.2 Acquisition of GPS C/A Code signals	8
2.2.1 Pull-in process	12
2.3 Tracking GPS signals	13
2.4 Computer simulations	19



2.5	Summary	24
Chapter 3 Tracking System Improvement		25
3.1	Phase estimation.....	25
3.1.1	Optimization in the analog domain	28
3.1.2	Approximation with one-bit quantized software receiver.....	29
3.2	Carrier fine frequency resolution	30
3.3	Resolving ambiguity in fine frequency measurement.....	32
3.4	Computer simulations	34
3.5	Summary	42
Chapter 4 Virtual Carrier Table for One-Bit Quantized		
Software Receivers		43
4.1	Carrier generator	44
4.2	Virtual carrier table	45
4.2.1	Algorithm	46
4.2.2	System description	53
4.3	Computer simulations	55
4.4	Summary	59
Chapter 5 Conclusions		60
Bibliography		63

List of Figures

Figure 2.1: Traditional GPS receiver.....	6
Figure 2.2: GPS software receiver.	6
Figure 2.3: Software processor in GPS receiver.	7
Figure 2.4: Block diagram of acquisition in time domain.	9
Figure 2.5: Frequency bins needed for carrier frequency estimation.....	10
Figure 2.6: A block diagram of acquisition using FFT.....	12
Figure 2.7: Ambiguous range of carrier frequency.	13
Figure 2.8: Code and carrier tracking loops referred to [2].	14
Figure 2.9: Part of code tracking loop.....	15
Figure 2.10: Carrier tracking loop.....	16
Figure 2.11: Z domain.	17
Figure 2.12: Acquisition for satellite id #4.	19
Figure 2.13: Code phase search of satellite #4 for a given carrier frequency = 11.324 MHz.	20
Figure 2.14: Frequency search of 11 frequency components, in 1k Hz step for a given code phase.	20
Figure 2.15: Fine frequency search in 200Hz step in the pull-in process.....	21
Figure 2.16: Phase estimation of phase detector.	22
Figure 2.17: Adjusted phase output of NCO.....	22
Figure 2.18: In-phase and quadrature Costas loop components.	23

Figure 3.1: Conventional phase detection method.....	25
Figure 3.2: Comparison of Costas PLL discriminators.....	29
Figure 3.3: Phase angle from two consecutive data sets.....	31
Figure 3.4: Frequency offset estimation using carrier tracking loop.	31
Figure 3.5: Flowchart of carrier tracking in GPS.....	34
Figure 3.6: Frequency offset performance under different SNR.	35
Figure 3.7: Frequency offset performance under different SNR for $ \Delta f_l \leq 110\text{Hz}$	36
Figure 3.8: Frequency offset performance under different SNR for $ \Delta f_l \leq 120\text{Hz}$	36
Figure 3.9: Frequency offset estimation performance under different SNR with N=100.	37
Figure 3.10: Frequency offset estimation performance under analog process with N=100.	38
Figure 3.11: Frequency offset performance under different f_s	39
Figure 3.12: Δf versus frequency offset error under different f_s	39
Figure 3.13: Resulting error signals for zero frequency initialization and 6 msec estimation time (50 Hz initial offset and various phase shift.)	42
Figure 4.1: The general sin table with full range of 2π	44
Figure 4.2: Flowchart of VCT generator for $p \geq 2$	54
Figure 4.3: Flowchart of VCT generator for $p=1$	54
Figure 4.4: The errors between programming and VCT vs. discrete time index k	56
Figure 4.5: The accumulated number of errors vs. time index k for 4 simulation examples.	57
Figure 4.6: The accumulated number of errors vs. time index k for $f_0=11324200$ Hz, $f_u=1.13242$ Hz.	58

List of Tables

Table 4.1: Frequency and system factors.	55
Table 4.2: The analysis of the corresponding phase angel at the error point for $m=3$, $f_c=1000300\text{Hz}$, $f_s=4\text{ MHz}$	56
Table 4.3: Frequency and system factors for $f_0=11324200\text{ Hz}$, $f_u=1.13242\text{ Hz}$	58

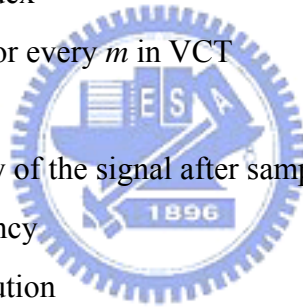


Acronym Glossary

AWGN	additive white Gaussian noise
ADC	analog-to-digital converter
BPSK	binary phase-shift keying
C/A	Coarse / Acquisition
cw	continuous wave
DLL	delay locked loop
DDS	direct digital frequency synthesizer
DSSS	direct sequence spread spectrum
FFT/IFFT	fast Fourier transform / inverse fast Fourier transform
FLL	frequency locked loop
gcd	greatest common divisor
GPS	Global Positioning System
IEEE	institute of electrical and electronics engineering
IF	intermediate frequency
NAVSTAR GPS	Navigation Signal Timing and Ranging Global Positioning System
NCO	numerical controlled oscillator
NSPO	National Space Organization Taiwan, ROC
TTF	time to first fix
P-code	Precision code
PLL	phase locked loop
PRN	Pseudo-random noise
SDR	software defined receiver
SNR	signal to noise ratio
VCO	voltage-controlled oscillator
VCT	virtual carrier table

Notations

*	complex conjugate
$a \tan(\cdot)$	arc-tangent-1 function
$a \tan 2(\cdot)$	arc-tangent-2 function
B_n	natural frequency
ε	phase estimate in PLL
c_{SV}	C/A code of satellite numbered SV
k	discrete time index
k_{mS}	the reset point for every m in VCT
r_b	bit rate
$IFnom$	center frequency of the signal after sampling
f_{aq}	acquired frequency
F_{inc}	frequency resolution
Δf	frequency offset
f_{offset}	desired frequency offset estimation for initial guess of NCO
f_s	sampling frequency
f_0	fundamental frequency
f_u	the minimum spacing between frequency bins
$r_{sv,i}$	the correlation of i th sampled data set and the PRN code of satellite SV
S	the maximum size of the particular time index set
SV	space vehicle number
t_s	sampling interval
T_{ob}	the observation time interval
T_s	sampling period
$\Delta\theta$	the variation of phase angle
$\hat{\phi}_k$	phase estimate for each sampled data point in PLL



$\hat{\phi}_i$	average phase estimate for i th data set in PLL
σ	frequency offset estimation error
θ_f	filtered phase offset for NCO
ω_n	noise (loop) bandwidths
ζ	damping factor



Chapter 1

Introduction

A. GPS overview

The Global Positioning System (GPS) is officially named Navigation Signal Timing and Ranging Global Positioning System (NAVSTAR GPS), which was developed by the United States Department of Defense. It is a kind of satellite-based navigation system with a constellation of at least 24 Medium Earth Orbit satellites that orbit the earth in 11 hours 58 minutes per period.

The current GPS consists of three major segments: the space segment, the control segment and the user segment. The user segment is namely the general application that the public investigate and use. To travel from a satellite to a receiver, a GPS signal uses direct sequence spread spectrum (DSSS) technique to modulate the carrier frequency with binary phase-shift keying (BPSK) modulation. The signal structure contains three part of information — carrier wave, a pseudo-random (PRN) code and navigation messages. The space vehicles transmit two microwave carrier signals utilizing two kinds of carrier frequency based on the fundamental frequency $f_0 = 10.23$ MHz ; one at 1575.42 MHz ($154 f_0$) called L1 and a second at 1227.60 MHz ($120 f_0$) called L2.

The PRN code is simply an identification code to specify which satellite information is received. It is also used to measure the virtual distance called the

pseudorange from user to the satellite. There are two ranging codes: the restricted Precision code (P-code), usually reserved for military applications and the other one is freely available to the public, called Coarse / Acquisition (C/A) code. The C/A code transmitted at 1.023 M bit/s, 1023 chips last for 1 millisecond (msec). In other words, the C/A code is 1 msec long. As we know in [2], there are 32 C/A code of satellites, but only 24 satellites are in orbit. The other five C/A codes are reserved for specific applications.

The navigation messages are broadcasted at 50 bit/s, including three major components. The satellite health information tells the GPS date and time, the satellite's status and an indication of its health. The second part is the ephemeris data which contains orbital information used to calculate the position of the satellite in orbit. Finally, the almanac data, being constantly transmitted by each satellite, has information about the current date and time, and the status of the satellite constellation (satellite health). Since the navigation data bit rate is 50 bit/s, a data bit is 20 msec long and contains 20 C/A codes.

The details of GPS concept can be obtained in [1]-[3]. The basic principle of GPS navigation system is to measure the distance between a known location of the satellite and the user's receiver and then obtain the specific receiver location by integrating these satellite data. Thus a typical GPS receiver needs at least four satellite signals to compute user's position in three dimensions (latitude, longitude and altitude) and the time offset in the receiver clock. Once the user's position has been obtained, other information can be calculated, like speed, distance to destination and so on.

B. Our work

While the GPS equipment is more and more popular, competition in the market will be increasingly severe. The major aspirations of the strategy include cost reduction,

functions increase and enhancement of performance. To reduce cost, the main practice is to enhance the system integration. The increase of function aims at enhancing the added value of the equipment, such as adding Bluetooth PND, DVB - T, and other functions.

However, we believe that there is still room to improve the performance. In the GPS, the critical part of the performance evaluation is the GPS receiving technique. The performance indicators for GPS equipment include time to first fix (TTFF), sensitivity, accuracy and so on. To target a moving user's position, we must acquire and track the satellite signal, time synchronization, navigation decoding, and then calculate the position. Besides, the software receiver is widely used in GPS such that the structure will be more flexible [7]. A major difficulty in receiving satellite signal is because of the poor transmission power (SNR is about $-19 \pm 3 \sim 5$ dB [2][3]). Thus we need to obtain the navigation data from the mixed satellite signals by using some acquisition and tracking process [2]-[5] that is time-consuming.

In order to improve the performance of GPS receivers, one of the key considerations is the baseband signal processing. To achieve faultless signal processing, the acquisition of satellite signals, tracking accuracy, and the Doppler effects of carrier all have to be carefully managed in GPS software receiver. In this thesis, we focus on the improvement of signal carrier tracking. We propose a novel carrier tracking scheme by accurately specifying the initial phase/frequency of NCO. Further, a universal virtual carrier generation method is designed, so as to significantly reduce the required memory space for carrier table.

This thesis is organized as follows. In Chapter 2, the description of the software acquisition method and the general tracking technique will be discussed respectively and the model of GPS software process will be built. In Chapter 3, we propose a fine frequency/phase pre-estimation to improve tracking performance. By using a few

incoming satellite data, the proposed scheme can reduce the lock time and enhance the accuracy of frequency resolution substantially. In Chapter 4, a one-bit quantized carrier generator using virtual carrier table (VCT) is developed. The main feature of the VCT is computing the phase angle variation to decide the output of carrier generator. The proposed VCT algorithm is simulated and the numerical analysis is illustrated. Finally, we conclude this thesis in Chapter 5.



Chapter 2

Global Positioning System

As we know that baseband signal processing can achieve high-speed signal processing by means of dedicated hardware and software. By using specially-designed algorithm, we can acquire and track the satellite signals of GPS, obtain 50 bps of the navigation information and provide the estimation of C / A code and carrier, and then get a virtual distance. A baseband signal processing part includes numerical controlled oscillator (NCO), correlators and digital signal processors.

In this chapter, we will offer an overview of GPS receiver first. Then, we focus on the discussion of the software receiver technique referred to [2]-[7]. Finally, we exploit the receiving algorithm for GPS software system to acquire and track satellite signals. The result of the acquired parameters and the tracking performance will be presented.

2.1 GPS receiver

The GPS receiver is shown in Figure 2.1. In traditional GPS receivers, acquisition and tracking of signal are realized with hardware approach. Since the baseband signal processing is digital, acquisition and tracking can naturally be a pure software implementation. In this way, the receiver structure can be much simplified.

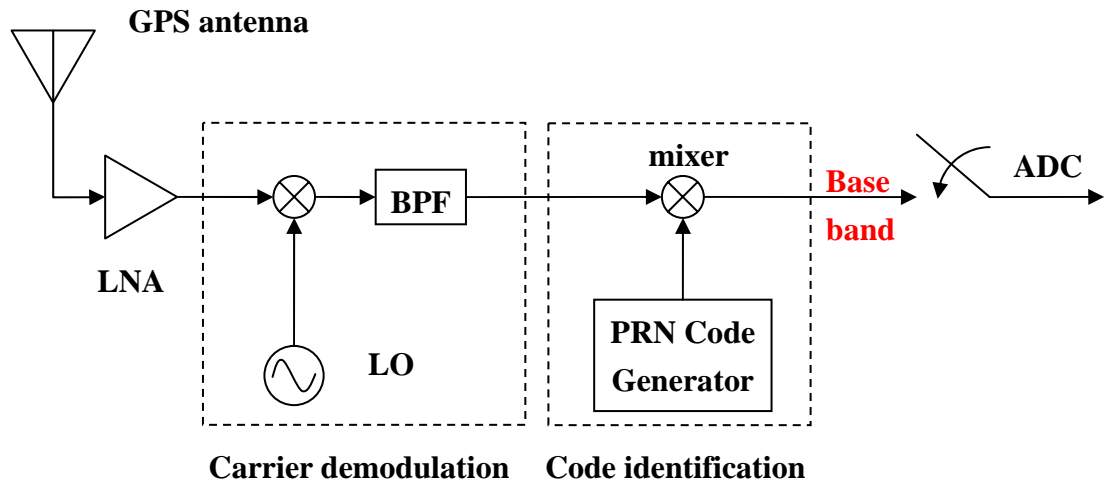


Figure 2.1: Traditional GPS receiver.

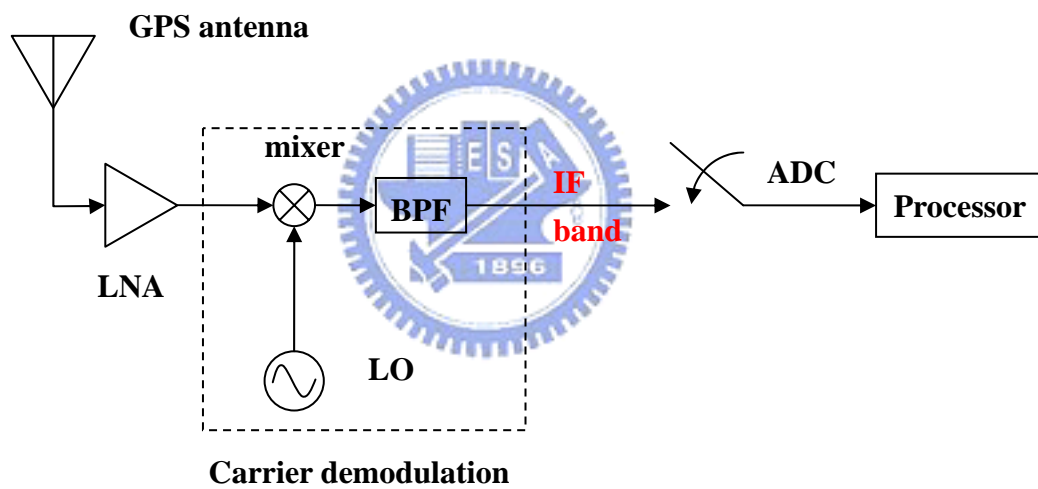


Figure 2.2: GPS software receiver.

The modern tendency of GPS receiver design is to use the concept of software defined receiver (SDR), which makes the position of analog-to-digital converter (ADC) as close to the antenna as possible. It can reduce the detrimental effects of temperature and aging in analog components. Moreover, apparent advantages including the flexibility of multiple navigation system and reprogrammable solution are afforded by software signal processing.

Figure 2.2 shows the block diagram of GPS software receiver. The purpose of signal processing in Figure 2.2 is to remove possible disturbing signals by filtering, amplify signal to an acceptable level, and down-sample signal to a selected intermediate frequency (IF). Since the IF of 15.42 MHz and the sampling frequency (f_s) of 4.096 MHz are chosen in our study, the center frequency of the signal is 11.324 MHz ($=IF-f_s$), denoted as IF_{nom} . (The detailed treatment of the antenna, the RF chain, and the digitizers can be found in Chapter 6 of Ref. [2].) As long as the signal is digitized and stored in memory, we can use the powerful DSP to replace some of the hardware features. The so-called software GPS is to use software to achieve the acquisition and tracking of satellite signals. In this thesis, we divide the software processor into four function blocks as Figure 2.3 and will discuss detailed treatment of each block in the following sections.

A software receiver can process data in batches. One simple example is the application of the FFT to a batch of data [4]. As will be shown later, such processing allows for more robust signal acquisition.

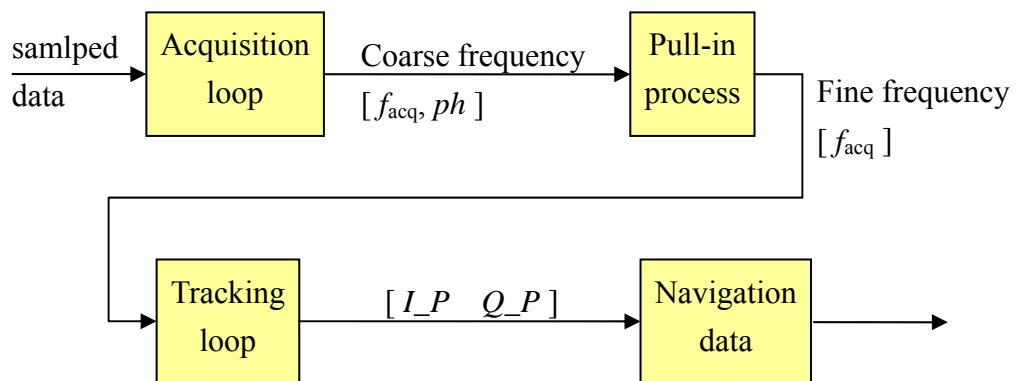


Figure 2.3: Software processor in GPS receiver.

GPS signals are all contained in a set of sampled data and we can search for different satellites by examining the sampled data. If the processor speed is quick enough to deal with different code phase matching, the relevant results may be faster than finding in time division by hardware. Under this condition, the processing speed and cost (size of the memory) determines the latency allowable for the signal processing software.

2.2 Acquisition of GPS C/A Code signals

The primary purpose of acquisition is to determine what visual satellite is. In the acquisition loop, two important parameters must be obtained and passed to the next function block. One is the coarse value of carrier frequency and the other is the phase shift of PRN code. In general, acquisition algorithms have the following two approaches: time domain correlation and frequency domain correlation. Time domain correlation algorithms can be divided into the linear search and data in a bank. Frequency domain algorithms can be divided into the full FFT treatment and half-FFT treatment [2].

The block diagram of conventional acquisition method is shown in Figure 2.4. The information needed for acquisition is the sampled data. First of all, we need to decide the operation data length. In practice, the maximum data length used for acquisition is limited to 10ms because [2]:

1. There could be a navigation data transition since the navigation data is 20 ms or 20 C/A code long.
2. The Doppler effect on the C/A code.

To simplify the experimentation, it's sensible to operate on 1ms block of data corresponding to the length of a complete C/A code. If the signal is digitized at 4.096 MHz, 1 ms data contains 4096 data points.

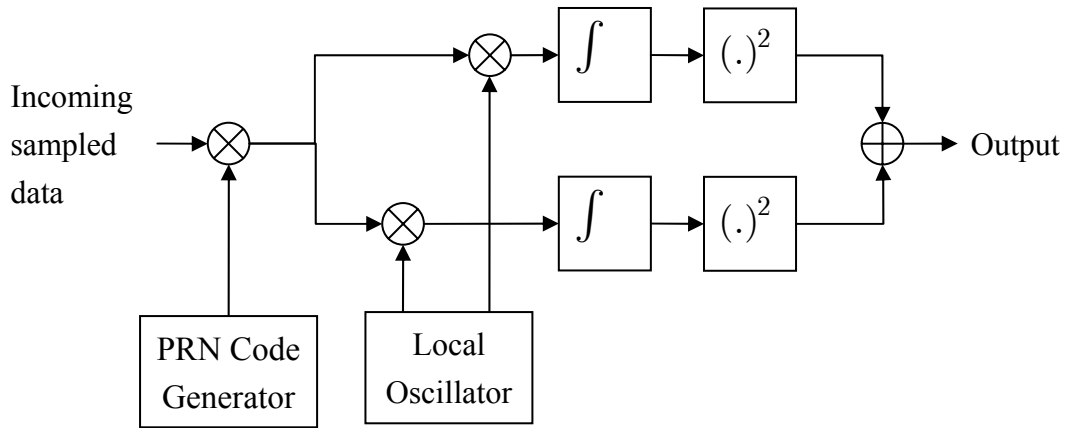


Figure 2.4: Block diagram of acquisition in time domain.

Assuming that the received signal $s(t)$ is composed of several visible satellite signals, we determine whether the signal is belonging to a particular satellite by comparing with 32 possible satellite PRN codes defined for satellite identification numbers. When we acquire space vehicle number (SV), $s(t)$ must multiply together with the digitized C/A code which is generated by local PRN code of satellite SV . To find the right PRN code offset, that is, the beginning of C/A code, the receiver slides a replica of the code in time until maximum correlation occurs. The local PRN code can be expressed as

$$l_{sv,i} = c_{sv} e^{j2\pi f_i t}, \quad (2.1)$$

where c_{sv} is the C/A code of satellite SV and f_i is the carrier frequency. Once the signal is de-spread to a continuous wave (cw) by a correct C/A code phase, the carrier frequency will be found. The 4096 real and imaginary values of the products by cw and local generated carrier are squared and added together, and the square root of this value represents the amplitude of the output frequency bin.

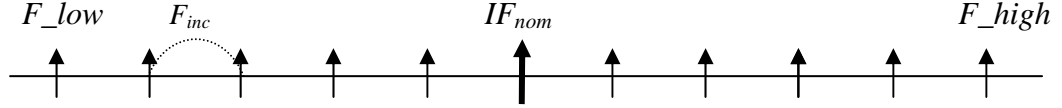


Figure 2.5: Frequency bins needed for carrier frequency estimation.

Because the assumption of 1 ms pre-detection integration time, the frequency resolution denoted as F_{inc} is 1 kHz, which is the inverse of the data length [2]. Besides, the acquisition method must search over a frequency range of ± 5 kHz to cover all of the possible Doppler shift for low-speed vehicles in contrast with high-speed aircraft [2]. Under these conditions, we have to adjust 11 frequency bins, 5 for each side of IF_{nom} , until maximum correlation is obtained.

As a result, the total acquisition time is proportional to the product of the C/A circular shift time and the frequency bin number, that is, 45056 (4096×11). This process could be time consuming.

In an effort to accelerate execution speed, D. J. R. Van Nee proposed a new fast GPS code acquisition technique using FFT [5]. It is convenient for us to implement the acquisition scheme of software systems based on FFT/IFFT. To begin with, we denote the N -points sampled data and the PRN code of satellite SV as $s(n)$ and $c(n)$, and then the circular correlation operation will be represented as

$$r_{sv,i}(m) = \sum_{n=0}^{N-1} s(n)c(n+m) . \quad (2.2)$$

To translate the above equation in the discrete time domain to frequency domain by FFT, it results in

$$\begin{aligned}
R_{sv,i}(k) &= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} s(n)c(n+m)e^{-j2\pi mk/N} \\
&= \sum_{n=0}^{N-1} s(n) \left[\sum_{m=0}^{N-1} c(n+m)e^{-j2\pi(n+m)k/N} \right] e^{-j2\pi(-n)k/N} \\
&= C(k) \sum_{n=0}^{N-1} s(n)e^{j2\pi nk/N} = C(k)S^{-1}(k) = C^{-1}(k)S(k).
\end{aligned} \tag{2.3}$$

where $S^{-1}(k)$ and $C^{-1}(k)$ are the inverse FFT (IFFT). If $s(n)$ is real, $s^*(n) = s(n)$ where $*$ is the complex conjugate. Hence the correlation will be obtained from IFFT as below [6]:

$$|r_{sv,i}(m)| = |F^{-1}(R_{sv,i}(k))| = |F^{-1}(C(k)S^*(k))| = |F^{-1}(C^*(k)S(k))|. \tag{2.4}$$

It is easy to derive the above results and the proofs of them are ignored. From above properties, it is obvious that the circular correlation and FFT/IFFT is necessary if the frequency domain acquisition is used. The block diagram of frequency domain acquisition is shown in Figure 2.6. Before multiplied by the locally generated C / A code, the signal must be multiplied by the local carrier frequency as depicted in Figure 2.5 to remove the Doppler shift.

For each frequency bin the following procedures are required:

1. Complex multiply 1ms input data $s(n)$ with the one-bit quantized carrier point by point and transform the result into frequency domain as $X_i(k)$.
2. Generate the local code of satellite $SV, c_{sv}(n)$ which is sampled at 4.096 MHz.
3. Perform FFT on $c_{sv}(n)$ and take the complex conjugate to obtain $C_{sv}(k)^*$.
4. Multiply $X_i(k)$ with $C_{sv}(k)^*$ point by point to result in $R_{sv,i}(k)$.

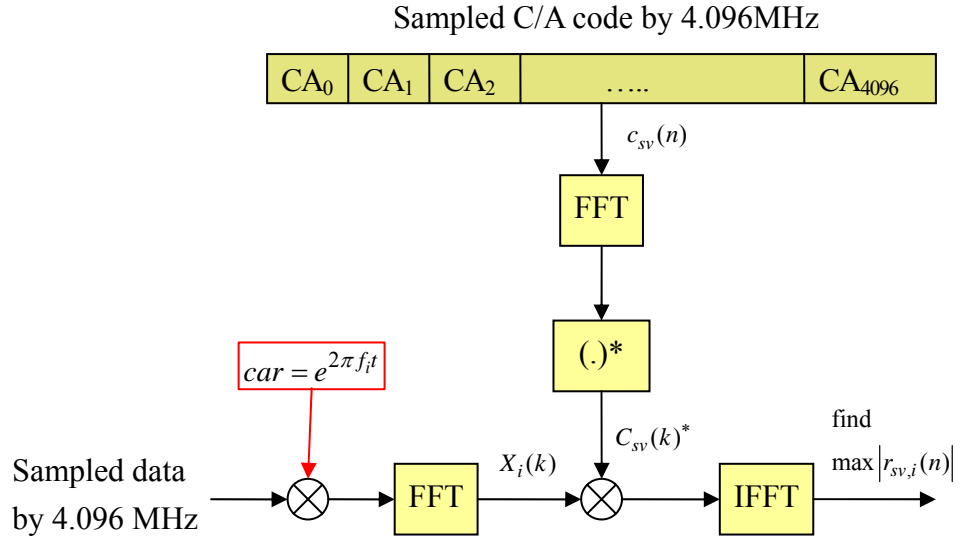


Figure 2.6: A block diagram of acquisition using FFT.

5. Take an inverse complex FFT of $R_{sv,i}(k)$ into $r_{sv,i}(n)$ and get a two dimension matrix $|r_{sv,i}(n)|$.
6. Finally, code phase is the n th location in circular shift of the local code that gives maximum correlation, $\max |r_{sv,i}(n)|$, and the carrier frequency is the corresponding i th frequency bin in 1 kHz resolution.

2.2.1 Pull-in process

From the above discussion, considering the frequency resolution of 1 kHz, the coarse frequency acquired from acquisition is not enough for accurate tracking. To solve the problem, a “pull-in” process is implemented to get fine value of carrier frequency. First of all, the main purpose here is to reduce the frequency resolution from 1 kHz to 200 Hz. Furthermore, Figure 2.7 shows that there will exist ambiguous range between the accurate carrier frequency and the acquired frequency, f_{acq} .

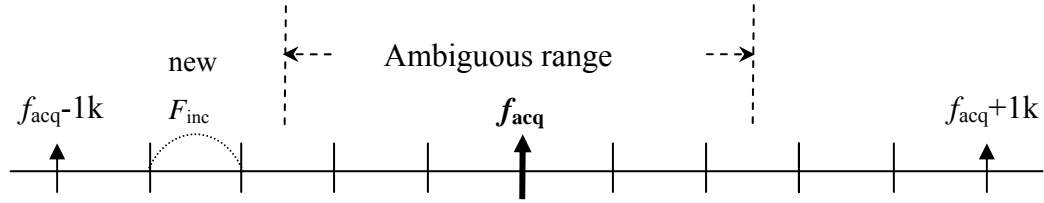


Figure 2.7: Ambiguous range of carrier frequency.

For that matter, we determine the fine carrier frequency by searching 7 candidate frequencies that ranges between $f_{acq} \pm 200i$ Hz, $i = 1, 2, 3$. As the C/A code has been found and only a few of candidate frequencies need to be examined, the pull-in function is performed by the time-domain correlation as shown in Figure 2.4.

2.3 Tracking GPS signals

In this section, conventional tracking technique is presented. The main objective of tracking is to enhance the precision for the coarse evaluation from acquisition, keep tracking the satellite signals varying over time. The accuracy of the PRN code phase shift will affect virtual distance evaluation. In order to strip off the C/A code and track the frequency/phase of input signal, it requires two tracking loops: the code loop and the carrier loop, as shown in Figure 2.8. We will introduce the tracking processing for these two loops.

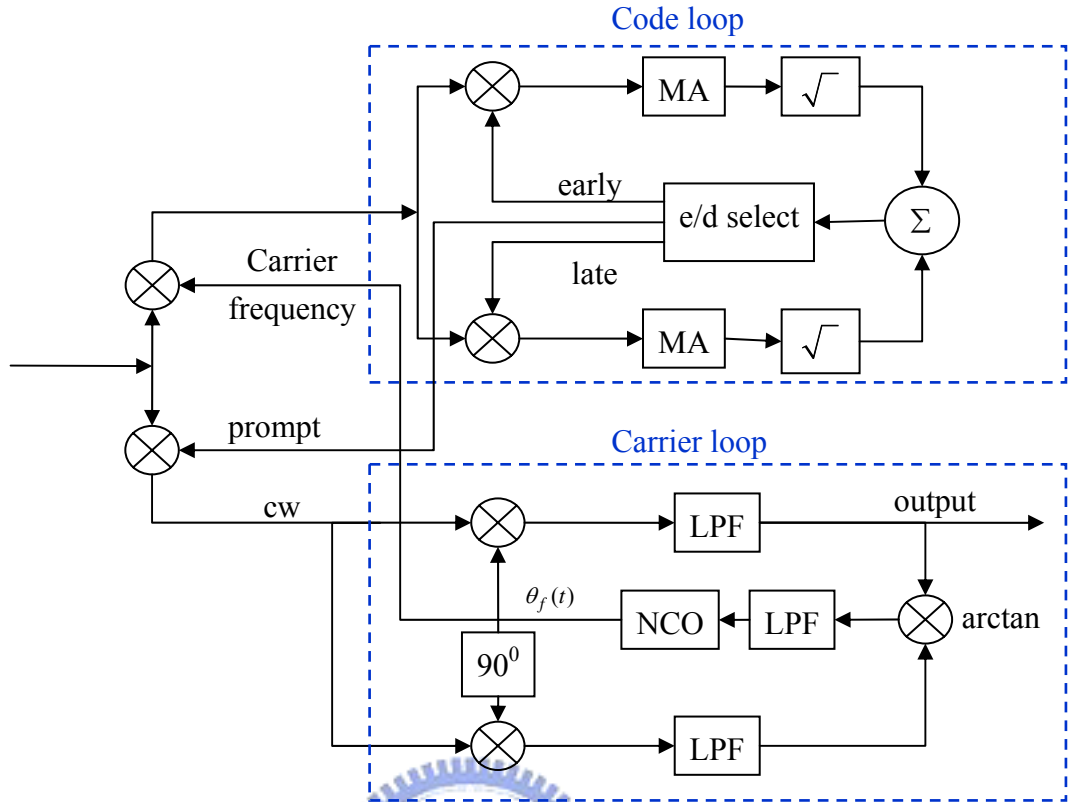


Figure 2.8: Code and carrier tracking loops referred to [2].

A. Code tracking

The code tracking enhances the accuracy of code phase obtained via acquisition by using the delay locked loop (DLL). The DLL generates three local PRN codes: a prompt code which is the locally generated C/A code with the beginning determined from acquisition process, an early code and a late code which are shifted early and late by approximately one-half chip, respectively.

Digital baseband I / Q signals, i.e. carrier wipe-off data in Figure 2.9, will be multiplied by these three codes individually. After passing through the correlation operation, a moving average filter, and finally the square function as we can see in Figure 2.9, it may result in three correlation value denoted by *Early_corval*, *Prompt_corval* and *Late_corval*.

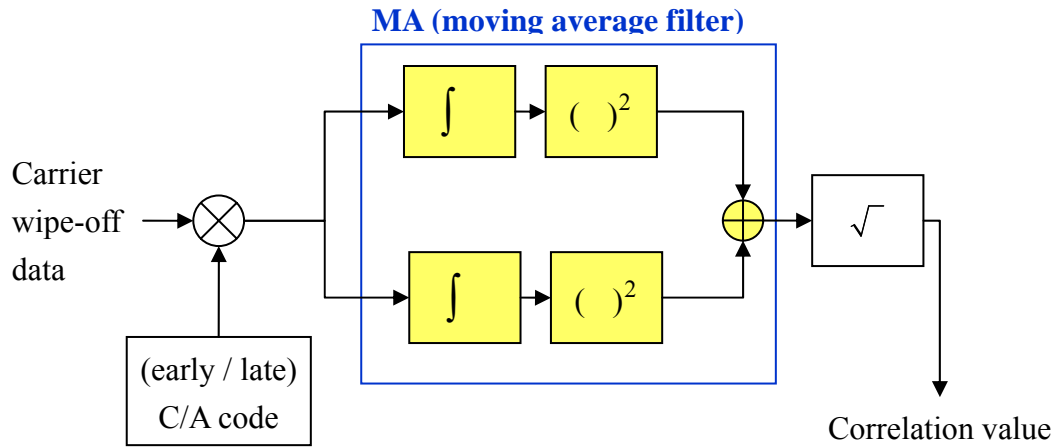


Figure 2.9: Part of code tracking loop.

According to the correlation result, we adjust code phase to generate new prompt code by a code loop discriminator, where the factor is denoted as EL which can be expressed as:

$$EL = \frac{\text{Early_corval}}{\text{Late_corval}} \quad (2.5)$$

We define the following operations for different early-late select value via statistics simulations [3]:

1. When resulting $EL < 0.8$, the new prompt code will be generated by shifting first point of original prompt code to the end.
2. If the discrimination EL is more than 1.5, the new prompt will be obtained by shifting end point of original prompt code to the head.

B. Carrier tracking

The carrier tracking follows the carrier frequency and phase changed after acquisition by using phase locked loop (PLL). A block diagram for the Costas carrier loop implementation is in Figure 2.10. PLL generates local carrier signal and measure the phase error between local carrier and the incoming digitized signal. Finally and most importantly, the PLL will adjust locally generated frequency to match the phase/frequency of the input signal.

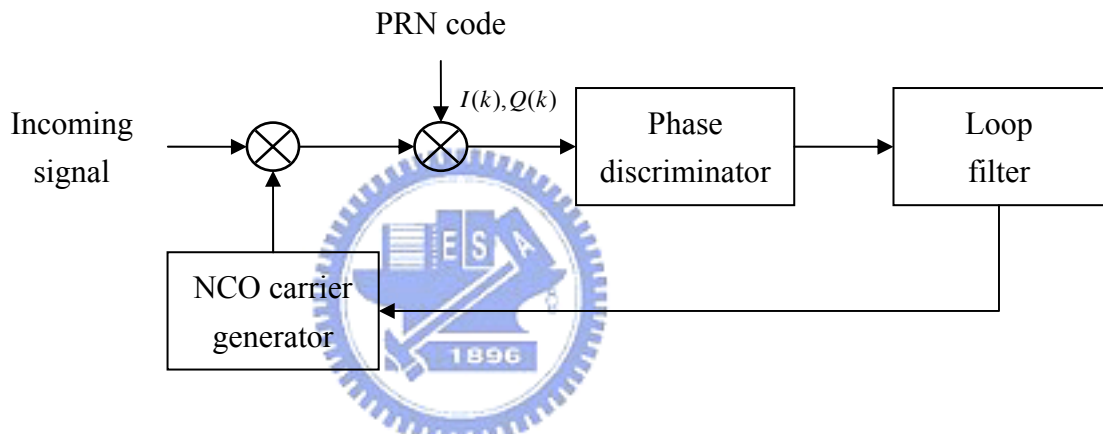


Figure 2.10: Carrier tracking loop.

To build software for digitized data, the PLL should transfer into the discrete z-domain through bilinear transform as

$$s = \frac{2}{t_s} \frac{1 - z^{-1}}{1 + z^{-1}}, \quad (2.6)$$

where t_s is the sampling interval given by 10^{-3} . Usually, the second-order PLL is widely used in GPS receiver because it has better performance in steady-state error [2].

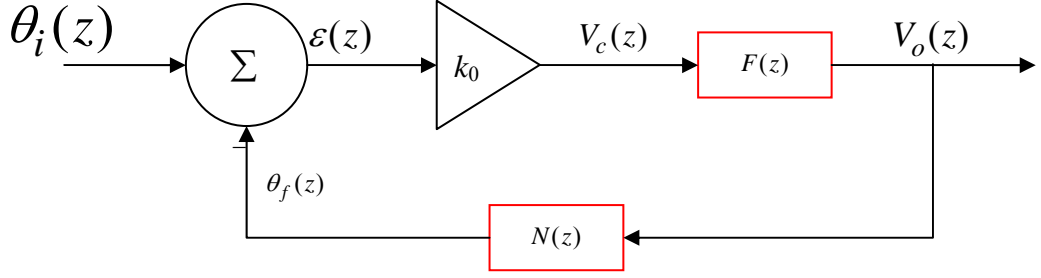


Figure 2.11: Z domain.

Therefore, the filter function shown in Figure 2.11 will be

$$F(z) = \frac{C_1 + C_2 - C_1 z^{-1}}{1 - z^{-1}} = \frac{V_0(z)}{V_c(z)} = \frac{V_0(z)}{k_0(\theta_i(z) - \theta_f(z))}, \quad (2.7)$$

where we denote $\theta_i(z) - \theta_f(z)$ as the phase estimate, $\varepsilon(z)$, and $\frac{V_0(z)}{k_0}$ as the filtered phase, while C_1 and C_2 are constants and will be calculated later.

On the other hand, direct digital frequency synthesizer (DDS), also named numerical controlled oscillator (NCO), will replace the voltage-controlled oscillator (VCO) in PLL and its transfer function can be defined as

$$N(z) = \frac{\theta_f(z)}{V_0(z)} = \frac{k_1 z^{-1}}{1 - z^{-1}}. \quad (2.8)$$

Consequently, we can obtain the filtered phase offset for NCO as follows:

$$\theta_f(z) = V_0(z) \frac{k_1 z^{-1}}{1 - z^{-1}} = \frac{V_0(z)}{k_0} \frac{k_0 k_1 z^{-1}}{1 - z^{-1}}. \quad (2.9)$$

Besides, two factors play an important role in the dynamic performance of the tracking loop and must be specified in order to implement the tracking loops. Based on the dynamics as discussed in [7], we denote the damping factor and noise (loop) bandwidths by ζ and ω_n while optimally flat response will be achieved by $\zeta = 1/\sqrt{2} \approx 0.7$ and $\omega_n = 50$ Hz, respectively. According to the noise bandwidth approximation for the tracking loop, the corresponding natural frequency will be presented as

$$B_n = \frac{2\omega_n}{\left(\zeta + \frac{1}{4\zeta}\right)} = 94.5945945945946. \quad (2.10)$$

Assuming the carrier loop gain k_0k_1 to be $4\pi \times 100$, then C_1 and C_2 in (2.7) can be found as

$$C_1 = \frac{1}{k_0k_1} \frac{8\zeta\omega_n t_s}{4 + 4\zeta\omega_n t_s + (\omega_n t_s)^2} = 9.8635 \times 10^{-5},$$

$$C_2 = \frac{1}{k_0k_1} \frac{4(\omega_n t_s)^2}{4 + 4\zeta\omega_n t_s + (\omega_n t_s)^2} = 6.6645 \times 10^{-6}. \quad (2.11)$$

Since the operation is performed every millisecond (msec), the local C/A will be regenerated and the phase of the local carrier frequency will be adjusted to match the incoming signals as closely as possible.

In this case, the acquisition algorithm could merely provide a rough estimate of the necessary parameters, and then the tracking loops could further update the information that would be utilized to keep following the input signals changed over time.

2.4 Computer simulations

In a software receiver, the acquisition algorithm described in Section 2.3 could offer a rough estimate of the necessary parameters. The results for a particular GPS PRN code, SV #4 is presented by a 3-dimensional plot shown in Figure 2.12. Specifically, the collected experimental data is the real satellite signal received by NSPO. In this case, 4096 possible code phases are evaluated and 11 different frequencies are looked over for each code phase from 11.319 MHz to 11.329 MHz, separated by 1k Hz.

The clear peak indicates PRN SV#4 is currently in use. The maximum correlation corresponds to the actual code phase of the 4079th sample and frequency offset of 11.324 MHz, as shown in Figure 2.13 and Figure 2.14, respectively.

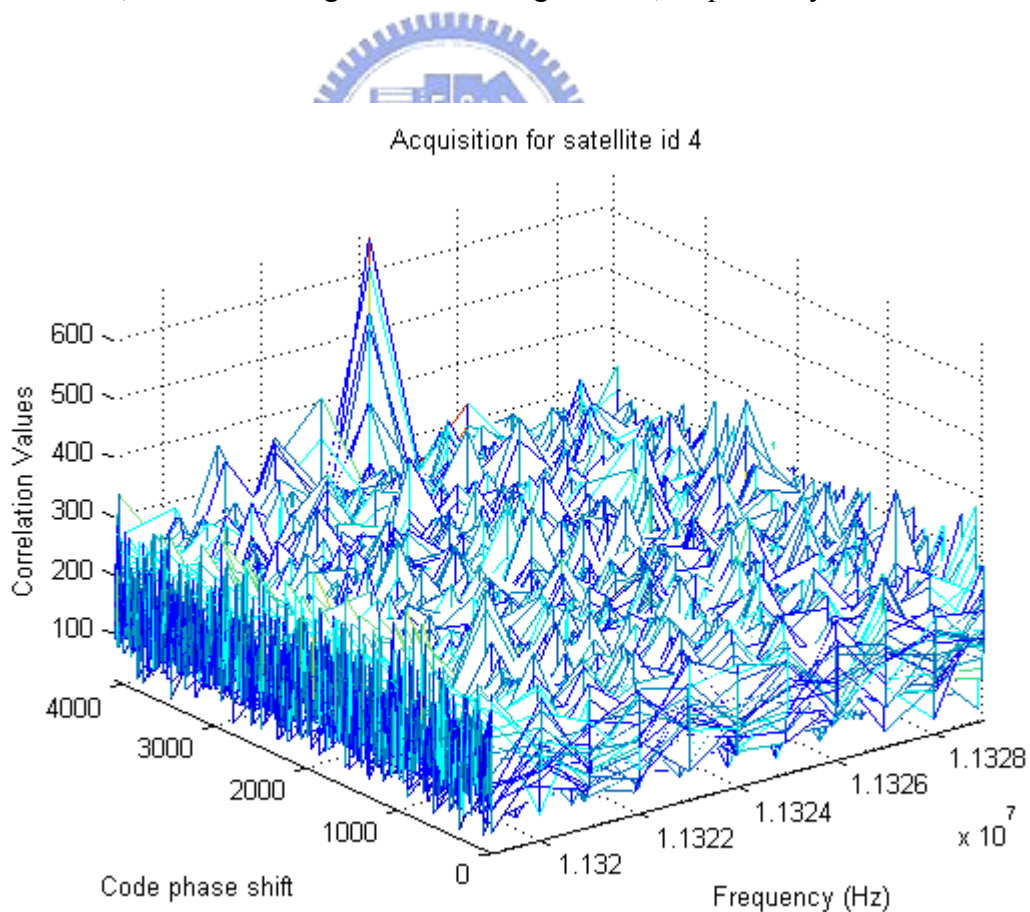


Figure 2.12: Acquisition for satellite id #4.

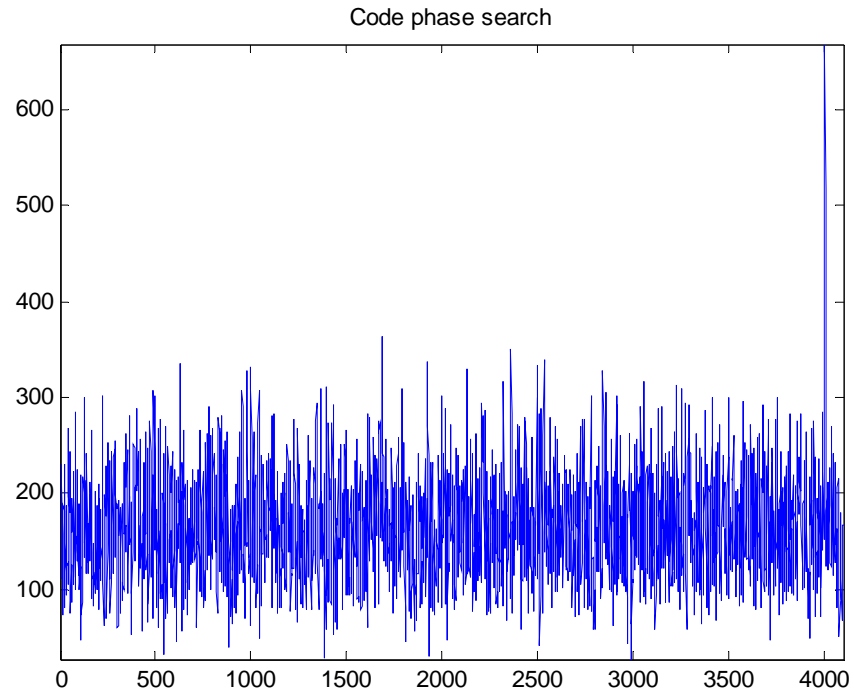


Figure 2.13: Code phase search of satellite #4 for a given carrier frequency = 11.324 MHz.

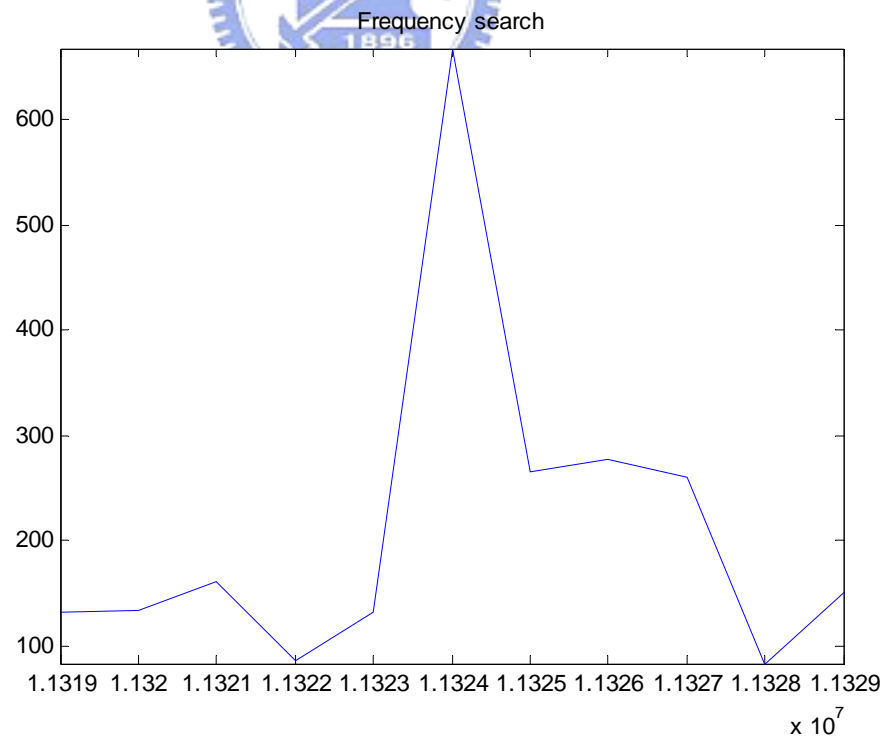


Figure 2.14: Frequency search of 11 frequency components, in 1k Hz step for a given code phase.

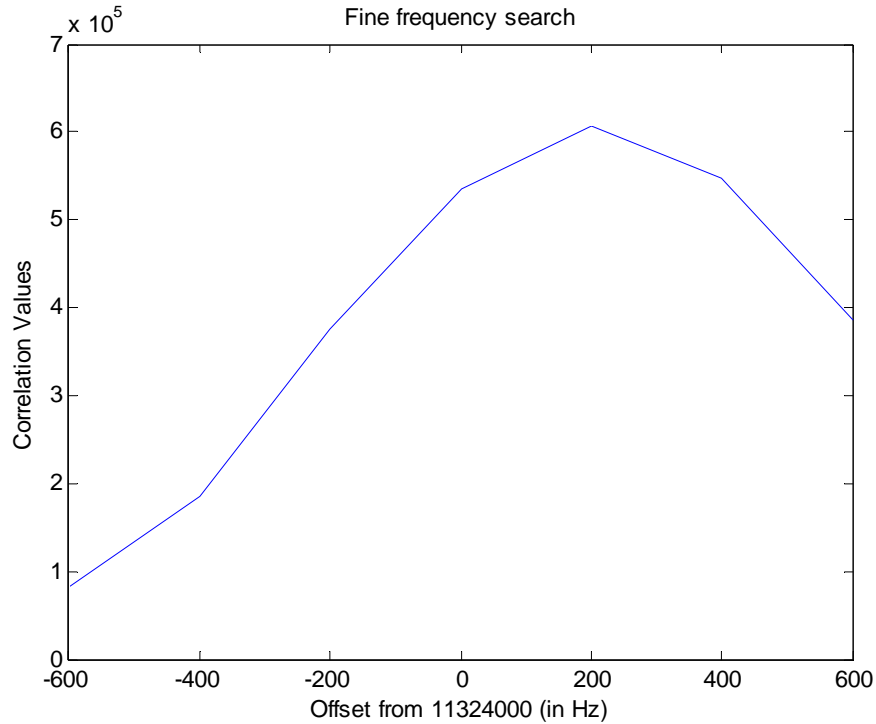


Figure 2.15: Fine frequency search in 200Hz step in the pull-in process.

The second stage in processing the GPS signal is to pull in the carrier frequency. This is demonstrated in Section 4.2.2 that the real frequency is within $f_{acq} \pm (1k/2)$, namely 11324000 ± 500 Hz. As shown in Figure 2.15, the maximum occurs at a frequency offset of 200Hz from the acquired frequency. In other words, we obtain a more accurate carrier frequency, 11.3242 MHz, after this processing.

In the following, the general tracking implementation as described in Section 2.4 is performed. Here we concentrate on the phase component of carrier tracking loop. The simulation result for the filtered phase of NCO is presented in Figure 2.17. Compared with precise detection output of the phase detector shown in Figure 2.16, it is obvious that the phase angle of NCO changes little by little, which is based on previous estimated phase and detected phase after filtering by some coefficient according to (2.9).

Also note that the adjusted output phase of NCO continuously increases as time goes on, indicating that there is a slight phase error caused by the frequency mismatch between NCO and incoming carrier.

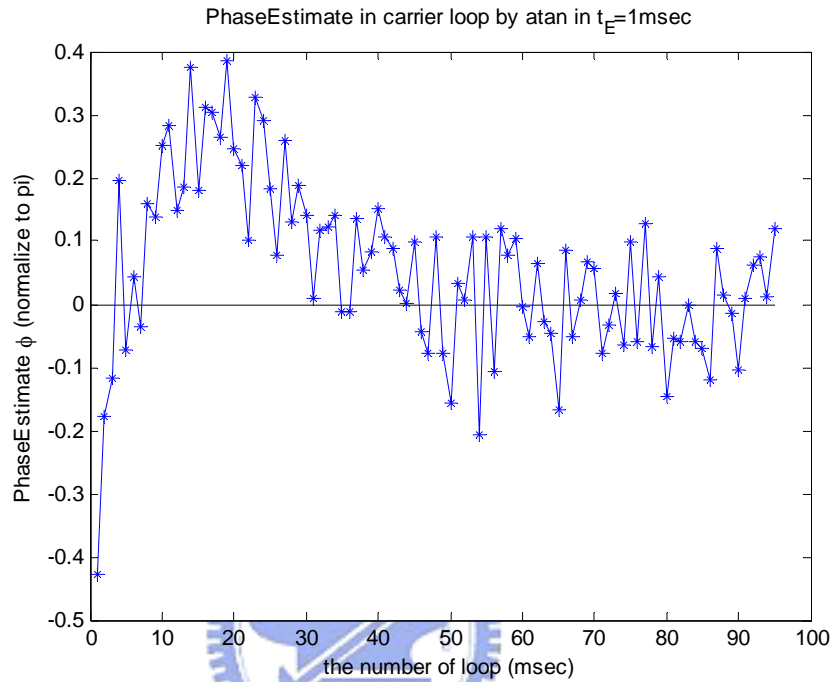


Figure 2.16: Phase estimation of phase detector.

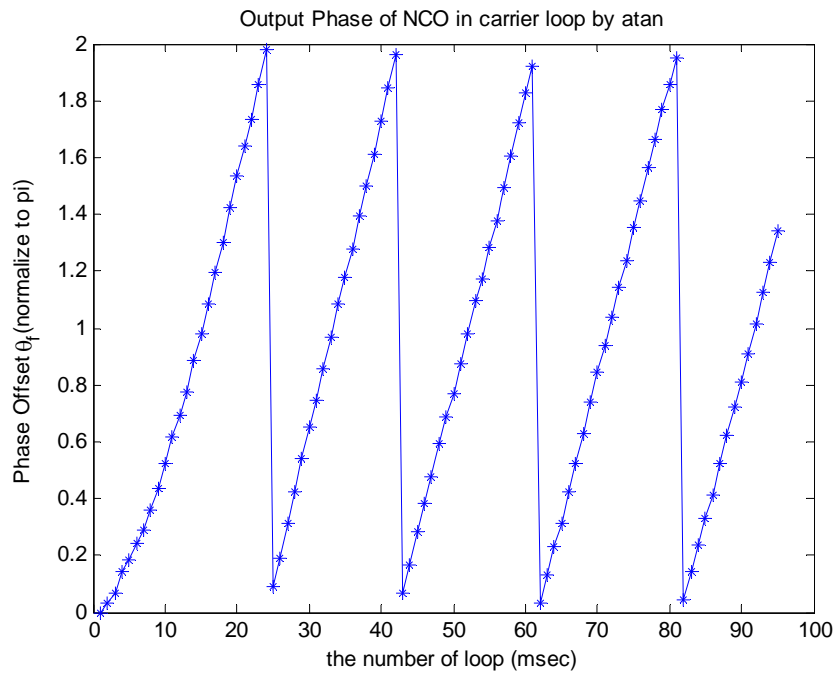
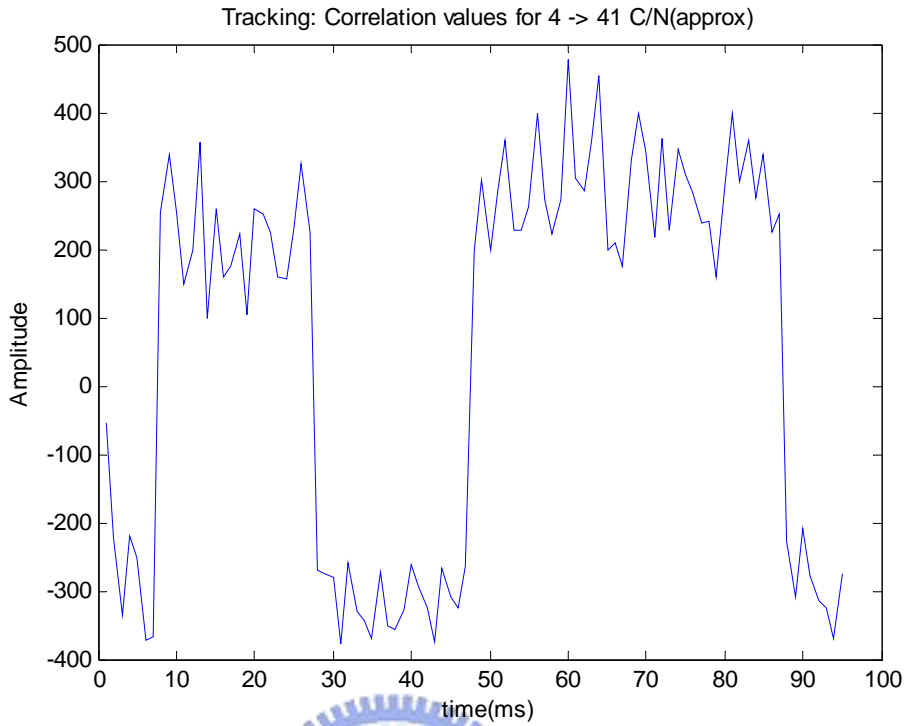
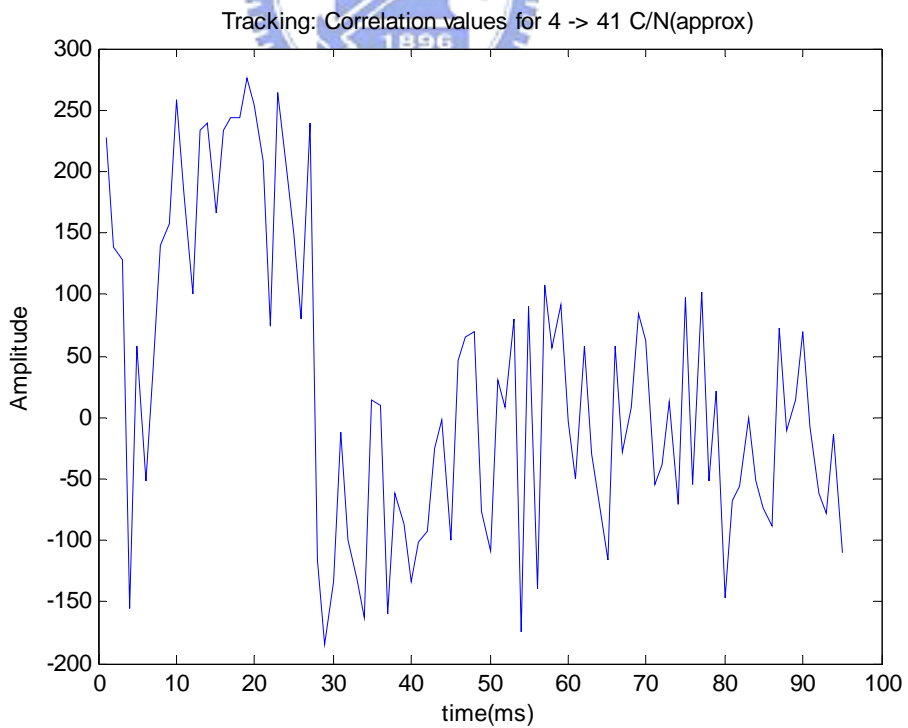


Figure 2.17: Adjusted phase output of NCO.



(a) In-phase



(b) Quadrature-phase

Figure 2.18: In-phase and quadrature Costas loop components.

Finally, if the detected phase is toward zero, that means the tracking loop is operating at the locked state, the in-phase component will contain the navigation data where 20 points express a data bit of 20 msec long as shown in Figure 2.18 (a). On the contrary, the quadrature-phase arm of the loop as Figure 2.18 (b) looks like noise.

2.5 Summary

In global positioning systems, the software receiver is preferred to offer tremendous flexibility. It should be noted that in an effort to reduce acquisition time and provide more robust signal acquisition, the parallel search technique using FFT based on frequency domain acquisition is preferred. Two important acquisition parameters, the beginning of the C/A code period as well as the carrier frequency, will be used in the subsequent tracking loop.

After pull-in process, we can obtain fine carrier frequency with resolution of 200 Hz. The tracking loop contains a code loop which enhances the code phase by DLL and the carrier loop which tracks the signal frequency/phase by PLL. However, from the simulation result, there is a transient period for tracking loops to lock the signal because of the mismatch between NCO and incoming signal. The result implies that we can have some improvement on the carrier tracking loop structure.

Chapter 3

Tracking System Improvement

To improve the tracking performance, here we design a frequency offset pre-estimation for the NCO in PLL. The idea is using the first few data to estimate frequency offset so as to reduce the required lock time. That is, we propose to improve the PLL performance by modifying the initial tracking loop structure.

3.1 Phase estimation

Figure 3.1 shows an approach widely-used in practice, which employs the arctangent method to detect signal phase.

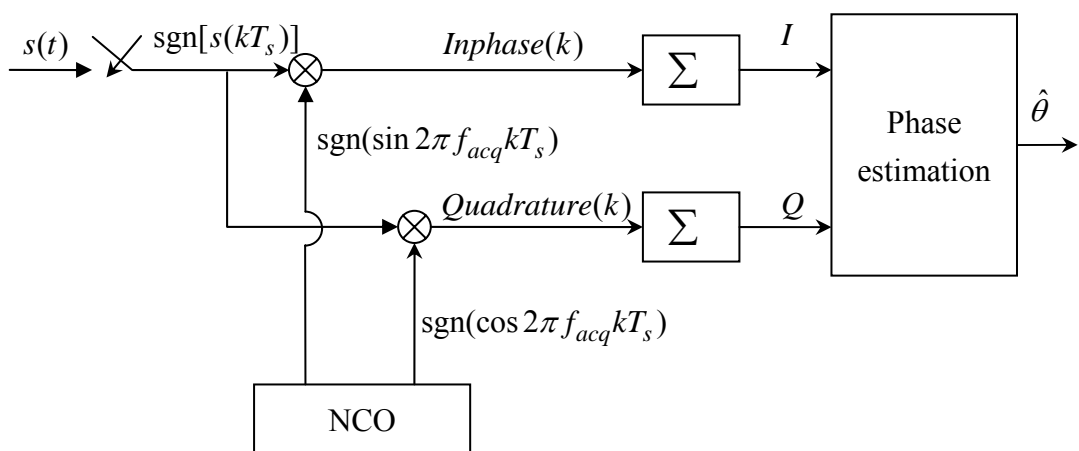


Figure 3.1: Conventional phase detection method.

Consider a received signal $s(t)$ given as

$$s(t) = A \sin(2\pi f_c t + \theta) + n(t), \quad (3.1)$$

where A is the signal amplitude, f_c is the down-converted carrier frequency, θ is the unknown phase and $n(t)$ is the added Gaussian noise.

In the figure, the signal $s(t)$ is one-bit quantized at $t = kT_s$, $k = 0, 1, 2, \dots$, where T_s is the sampling period. A NCO is used to detect the signal phase. The NCO generates two one-bit quantized local carriers:

$$I_{NCO} = \text{sgn}(\sin 2\pi f_{acq} k T_s), \quad (3.2)$$

$$Q_{NCO} = \text{sgn}(\cos 2\pi f_{acq} k T_s). \quad (3.3)$$

where f_{acq} is the carrier frequency obtained from the pull-in process, and $\text{sgn}(x)$ is the sign function which returns 1 if $x \geq 0$ and -1 if $x < 0$.

However, the acquired frequency is not the same as the carrier frequency since the frequency resolution of pull-in process is limited to 200 Hz. As the accuracy of carrier frequency is of concern, the frequency of NCO is assumed to be unequal to the incoming carrier frequency and we denote the frequency offset as Δf . Accordingly, the incoming signal in (3.1) can be written as

$$s(t) = \sin(2\pi(f_{acq} + \Delta f)t + \theta) + n(t). \quad (3.4)$$

Thus, the signal $Inphase(k)$ is given as

$$Inphase(k) = \text{sgn}[s(kT_s)] \cdot \text{sgn}[\sin(2\pi f_{acq}kT_s)]. \quad (3.5)$$

Nevertheless, as one-bit ADC is employed to detect the phase θ , only polarity of the sampled signal is available. Based on the definition of the polarity function, it is easy to have the following equality:

$$\text{sgn}(x) \cdot \text{sgn}(y) = \text{sgn}(x \cdot y), \quad (3.6)$$

where x and y are arbitrary real numbers. Using the above property, we can reformulate (3.5) as

$$\begin{aligned} Inphase(k) &= \text{sgn}[s(kT_s) \cdot \sin(2\pi f_{acq}kT_s)] \\ &= \text{sgn}\left\{\frac{A}{2}[\cos \phi_k - \cos(\omega'kT_s + \theta)] + n(kT_s) \cdot \sin(2\pi f_{acq}kT_s)\right\}, \end{aligned} \quad (3.7)$$

where $\phi_k = 2\pi\Delta f kT_s + \theta$ is the phase difference, $\omega' = 2\pi(2f_{acq} + \Delta f)$ is the high-frequency terms. Similarly, the signal $Quadrature(k)$ in Figure 3.1 can be expressed as

$$Quadrature(k) = \text{sgn}\left\{\frac{A}{2}[\sin \phi_k - \sin(\omega'kT_s + \theta)] + n(kT_s) \cdot \cos(2\pi f_{acq}kT_s)\right\}. \quad (3.8)$$

3.1.1 Optimization in the analog domain

Suppose T_{ob} is the given observation time interval which is 1msec because of the length of C/A code. Assuming $\Delta f = 0$ such that $\phi_k = \theta$. If all the signals are kept in the analog domain without quantization, then the I-channel and Q-channel outputs become

$$I' = \int_0^{T_{ob}} \frac{A}{2} [\cos \theta - \cos(\omega' k T_s + \theta)] \approx \frac{A}{2} \cos \theta, \quad (3.9)$$

$$Q' = \int_0^{T_{ob}} \frac{A}{2} [\sin \theta + \sin(\omega' k T_s + \theta)] \approx \frac{A}{2} \sin \theta. \quad (3.10)$$

where the contribution of high-frequency terms is neglected in the summation and noise is neglected. As I' and Q' are proportional to $\cos \theta$ and $\sin \theta$, respectively, the optimum detection of carrier phase angle θ can be estimated with the arc-tangent (atan) or arc-tangent-2 (atan2) function that is popular in practical phase estimation for software receivers [3]:

$$\hat{\theta} = \begin{cases} a \tan\left(\frac{Q'}{I'}\right), & -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \\ a \tan 2(Q', I'), & -\pi \leq \theta \leq \pi. \end{cases} \quad (3.11)$$

As shown in Figure 3.2, the results of atan function are limited to the interval

$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$ while atan2 function is limited to $-\pi \leq \theta \leq \pi$.

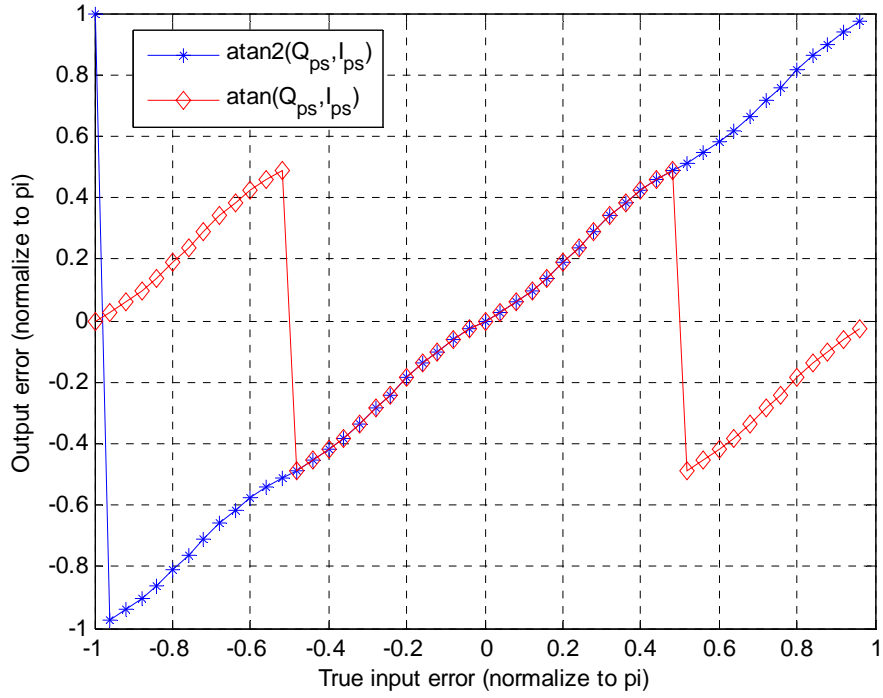


Figure 3.2: Comparison of Costas PLL discriminators.

3.1.2 Approximation with one-bit quantized software receiver

Since we employ one-bit ADC to detect the phase estimation in the software receiver, the summation process in digital domain is actually an approximation mimicking integration in the analog domain. Assuming $K = T_{ob} / T_s$ is the number of samples within T_{ob} after ADC, that is, 4096 points from 1 msec data per operation.

Hence the factors for phase detector, I and Q , in Figure 3.1 are obtained as

$$I = \sum_{k=0}^{K-1} Inphase(k) \approx \cos \theta, \quad (3.12)$$

$$Q = \sum_{k=0}^{K-1} Quadrature(k) \approx \sin \theta. \quad (3.13)$$

Finally, the estimated phase can also be calculated with the arc-tangent-2 function and the results will be used in next section. Importantly, it should be noted that the estimated phase in discrete time will be deviated from the real one due to significant quantization error resulting from the one-bit quantization.

3.2 Carrier fine frequency resolution

From the above discussion, we can obtain the phase estimation through the approximation while the phase of the samples within T_{ob} is constant.

But we understand the frequency offset Δf is a principal factor affecting lock time. In order to improve the tracking speed, an additional frequency estimation to provide an accurate initial guess is inserted into the PLL.

Considering slow-varying vehicle or user, the frequency variation within each sampled data set is little and the following algorithm can work. To analyze the carrier phase of each data set, we can figure out an average phase estimation $\hat{\phi}_i$ per 1 msec long data set and it corresponds to the phase angle of the middle data point in this data set, as shown in Figure 3.3. Since $\phi_k = 2\pi\Delta f k T_s + \theta$, the average phase of the i th data set can be approximated as

$$\hat{\phi}_i = 2\pi\Delta f \times (i - 0.5) \times T_p + \theta. \quad (3.14)$$

where T_p is the time between processing ,i.e. $T_p = 1$ msec .

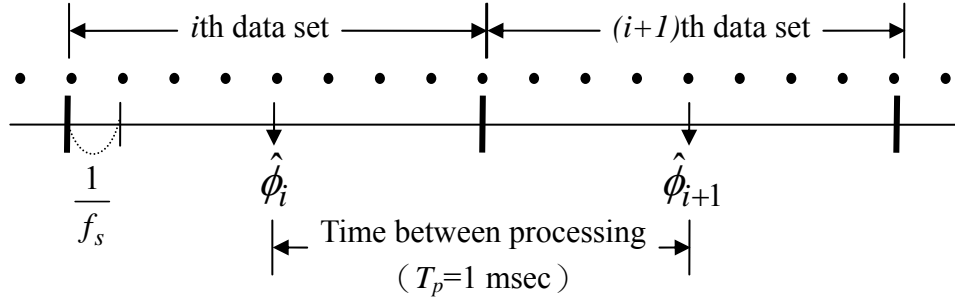


Figure 3.3: Phase angle from two consecutive data sets.

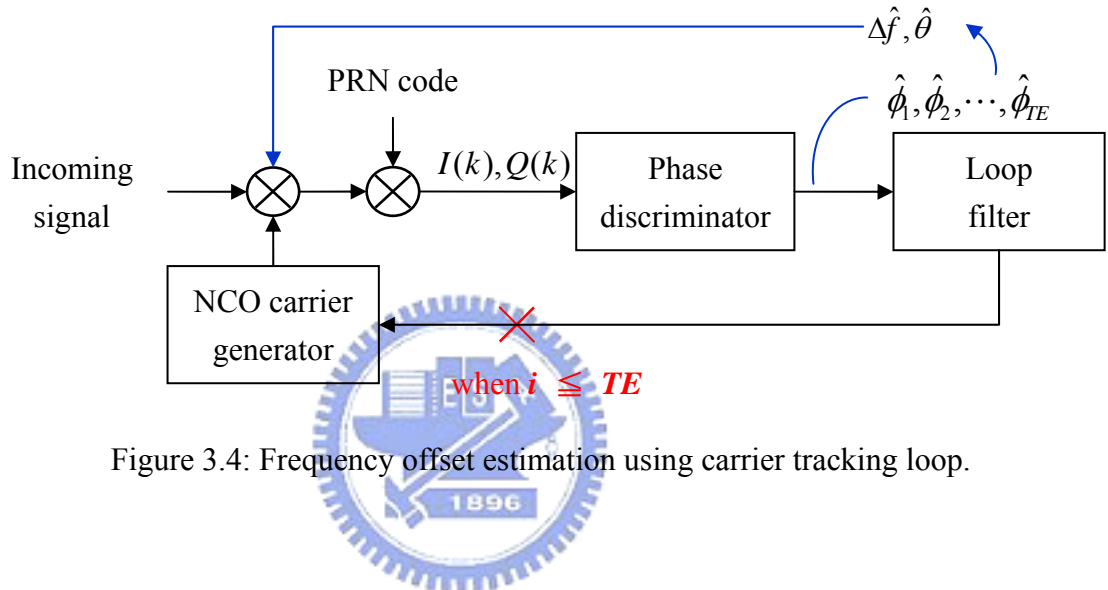


Figure 3.4: Frequency offset estimation using carrier tracking loop.

Assuming it takes TE time to get a frequency offset estimation. In other words, the tracking loop doesn't work before TE time. Figure 3.4 shows the block diagram of the modified tracking loop where $\{\hat{\phi}_1, \hat{\phi}_2, \dots, \hat{\phi}_{TE}\}$ are obtained initially. Once $\{\hat{\phi}_i\}$ from each data are obtained, the estimation of frequency offset Δf can be calculated easily via (3.14), which is determined by

$$\Delta f_i = \frac{\Delta \hat{\phi}_i}{2\pi \cdot T_p}, \quad (3.15)$$

where $\Delta \hat{\phi}_i = \hat{\phi}_{i+1} - \hat{\phi}_i$, $i = 1, \dots, TE - 1$.

3.3 Resolving ambiguity in fine frequency measurement

It is noted that each phase estimation $\hat{\phi}_i$ would be affected by noise. While the GPS has poor SNR, the noise will seriously affect the detection of phase. The calculation of frequency offset measurement has been presented in (3.15), here we introduce the limitation of (3.15) and design a reasonable frequency offset evaluation.

Whereas the pull-in frequency resolution obtained from 1ms data is 200Hz, the maximum absolute value of the frequency offset must be smaller than but not less than 100 Hz. Therefore, the phase difference $|\Delta\hat{\phi}|$ must be less than $2\pi|\Delta f|_{\max} T_p = 0.2\pi$.

Moreover, the possibility of a π phase shift between two data sets due to navigation data could happen at the same time. Let $\Delta\hat{f}_1, \dots, \Delta\hat{f}_L$ denote the L rational frequency offset estimations sieved out from $\Delta\hat{\phi}_1, \dots, \Delta\hat{\phi}_{TE-1}$ with the limitation. These frequency offsets will be used to calculate the desired fine frequency offset, denoted as f_{offset} . The desired frequency offset estimation is evaluated as below:

$$f_{offset} = \text{sign}(\Delta\hat{f}_{av}) \cdot |\Delta\hat{f}_{av}|. \quad (3.16)$$

The sign of f_{offset} is determined as

$$\text{sign}(\Delta\hat{f}_{av}) = \text{sign}\left[\sum_{l=1}^L \text{sign}(\Delta\hat{f}_l)\right], \quad (3.17)$$

and the value of f_{offset} will be obtained by

$$|\Delta\hat{f}_{av}| = \text{average of } |\Delta\hat{f}_{l,ok}|, \quad (3.18)$$

where $\Delta\hat{f}_{l,ok}$ are those with the same sign as $sign(\Delta\hat{f}_{av})$ in $\Delta\hat{f}_1, \dots, \Delta\hat{f}_L$.

While we have got the frequency offset, the corresponding phase angle θ referred to (3.14) will be evaluate by

$$\hat{\theta} = \hat{\phi}_{TE} - 2\pi f_{offset} \cdot (TE - 0.5) \cdot T_p. \quad (3.19)$$

After we obtain the estimated frequency offset and phase, the output of NCO will be adjusted to

$$I_{NCO} = \text{sgn}(\sin(\omega_{new}kT_s + \theta_f)), \quad (3.20)$$

$$Q_{NCO} = \text{sgn}(\cos(\omega_{new}kT_s + \theta_f)), \quad (3.21)$$

where ω_{new} is equal to $2\pi(f_{acq} + f_{offset})$ and θ_f is the resultant value from (3.19).

Figure 3.5 shows the frequency offset estimation approach on GPS in practice.

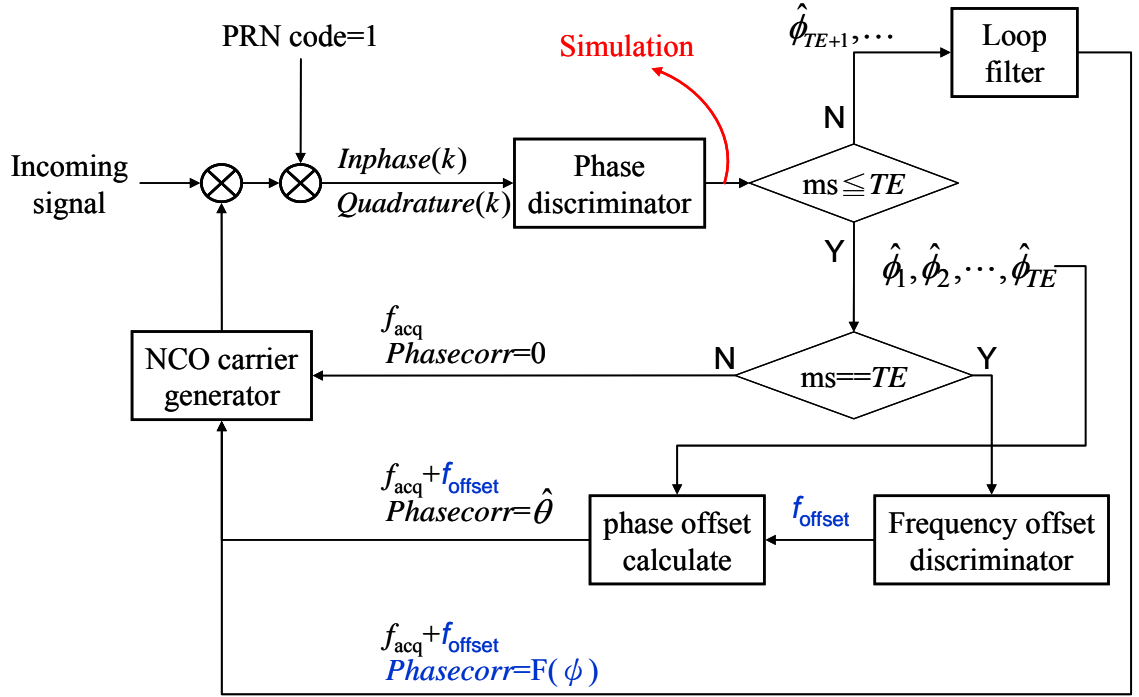


Figure 3.5: Flowchart of carrier tracking in GPS.

Finally, in order to evaluate the accuracy of the proposed frequency offset approach, we define the frequency offset error as:

$$\sigma = \sqrt{\frac{1}{N} \sum_{n=1}^N (f_{\text{offset}} - \Delta f)_n^2}, \quad (3.22)$$

where N is the total number of test times which is equal to 3000 without special indication.

3.4 Computer simulations

A. Estimation time (TE)

In this Chapter, we adjust the initial frequency offset at the expense of the first TE msec. Hence the most important issue is how long we need to get acceptable

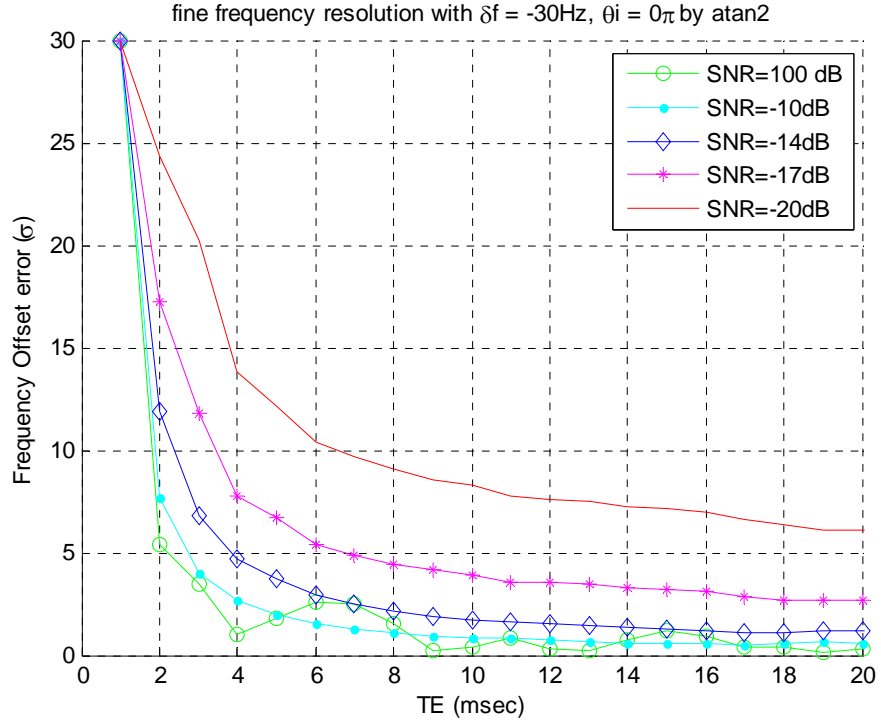


Figure 3.6: Frequency offset performance under different SNR.

f_{offset} . From the simulation result in Figure 3.6, we see that the error is lower than 10 Hz if TE is greater than 6 msec, thus $TE = 6$ msec will be our choice.

B. Limitation bound

In Figure 3.7 and Figure 3.8, we show different performances which employ different bounds for limitation such as $|\Delta\hat{\phi}| \leq 0.22\pi$ and $|\Delta\hat{\phi}| \leq 0.24\pi$ to find $\Delta\hat{f}_l$, $l = 1, \dots, L$ that are used to calculate f_{offset} . Comparing Figure 3.6 with Figure 3.7, it is shown that the bound for $|\Delta\hat{\phi}| \leq 0.24\pi$ ($=2\pi \times 120\text{Hz} \times 1\text{msec}$) has better performance especially in high frequency offset region.

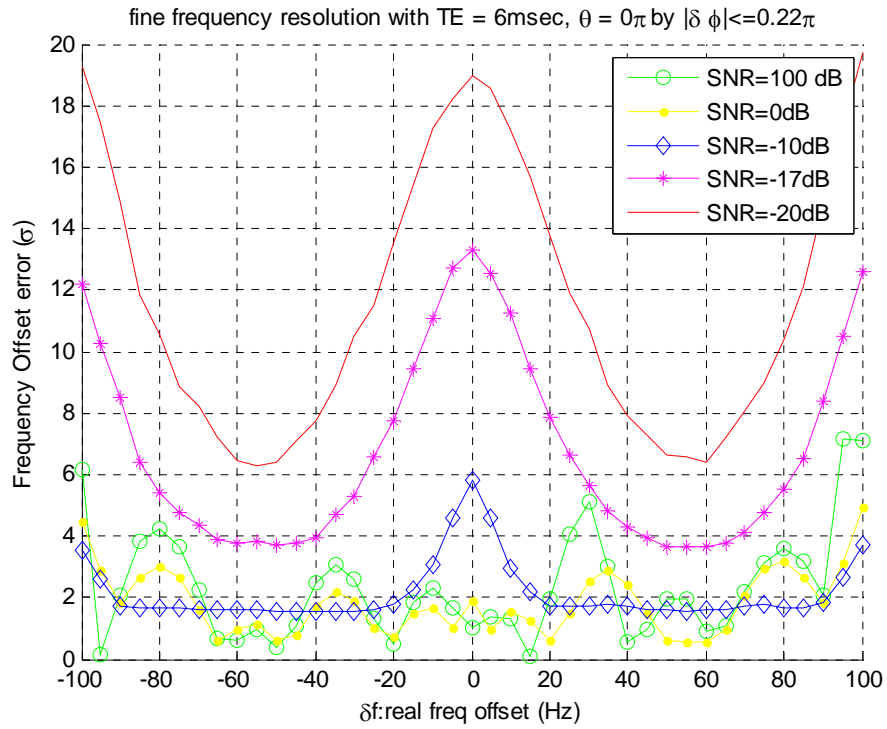


Figure 3.7: Frequency offset performance under different SNR for $|\Delta f_l| \leq 110\text{Hz}$.

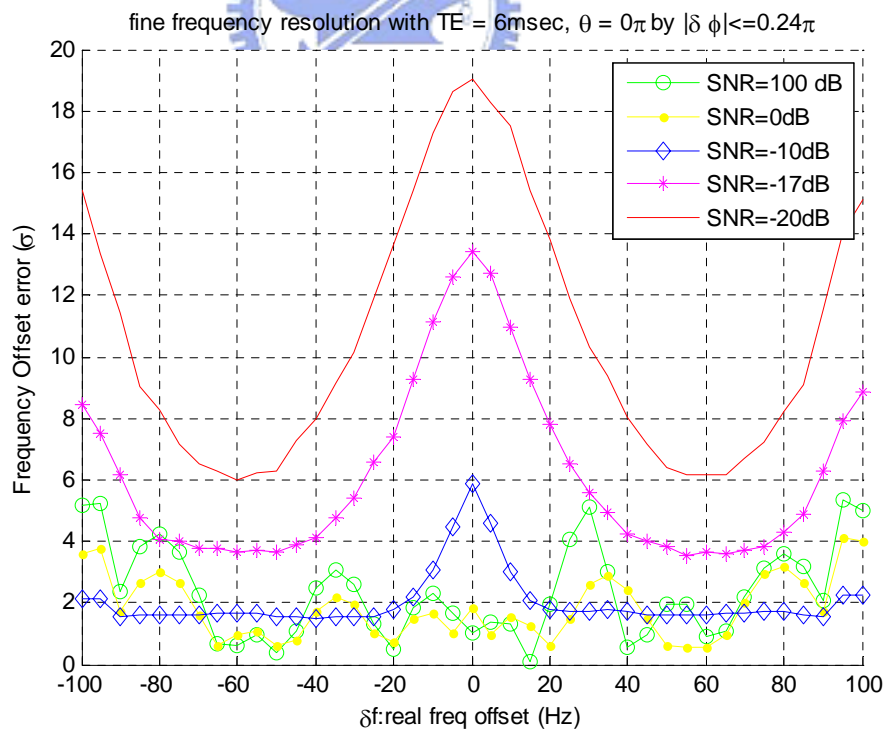


Figure 3.8: Frequency offset performance under different SNR for $|\Delta f_l| \leq 120\text{Hz}$.

C. Frequency offset error under different SNR

From above simulation results, we find a special result that the performance of frequency offset estimation under SNR=100 dB is worse than SNR=-10 dB in some initial frequency offset. This is shown in Figure 3.9, for example the error when SNR=100 dB could be larger than that of SNR=-10 dB at $\Delta f=70$ Hz. To verify the cause, we present the simulated frequency offset error at various SNR under analog processing, as shown in Figure 3.10. As we can see, the results coincide with our intuition, that high SNR implies less error. Compared with analog processing, we consider the special result in Figure 3.9 is similar to “Dither” effect which is routinely used in processing of digital audio and video data: an intentionally applied noise can randomize quantization error and have better performance on the contrary, as detailed in [8].

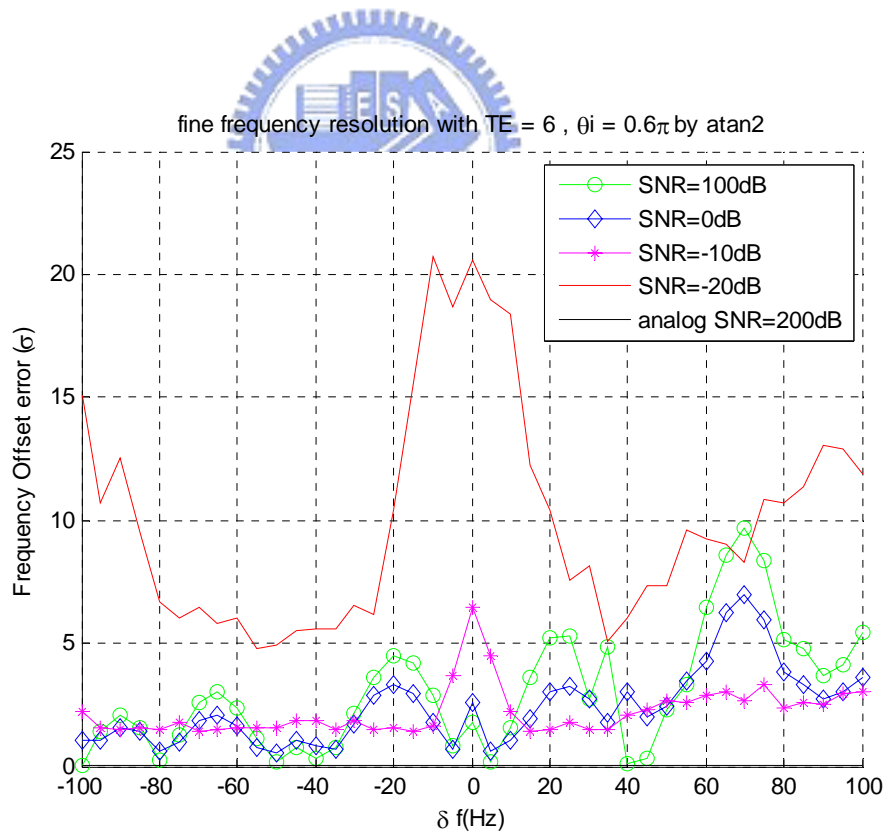


Figure 3.9: Frequency offset estimation performance under different SNR with N=100.

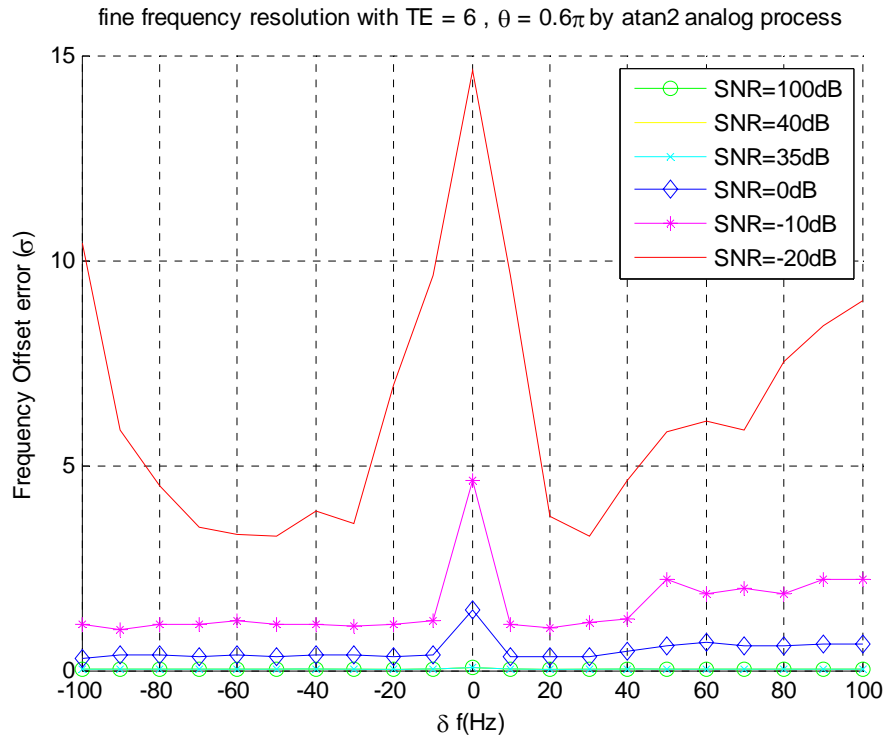


Figure 3.10: Frequency offset estimation performance under analog process with $N=100$.

D. Adjust sampling frequency to obtain better performance

As discussed in Section 2.1, we know that GPS has poor SNR from -14dB to -24dB, here we concentrate on the performance for SNR = -20 dB. Since the GPS is implemented with one-bit quantization, using higher sample rate can improve the frequency offset measurement. First, it is shown that the error will be reduced to half when sampling frequency is doubled in Figure 3.11. Figure 3.12 shows all possible Δf versus frequency offset error under different f_s . Consequently, if the system has four times sampling frequency, the estimation error will be restricted within 10Hz.

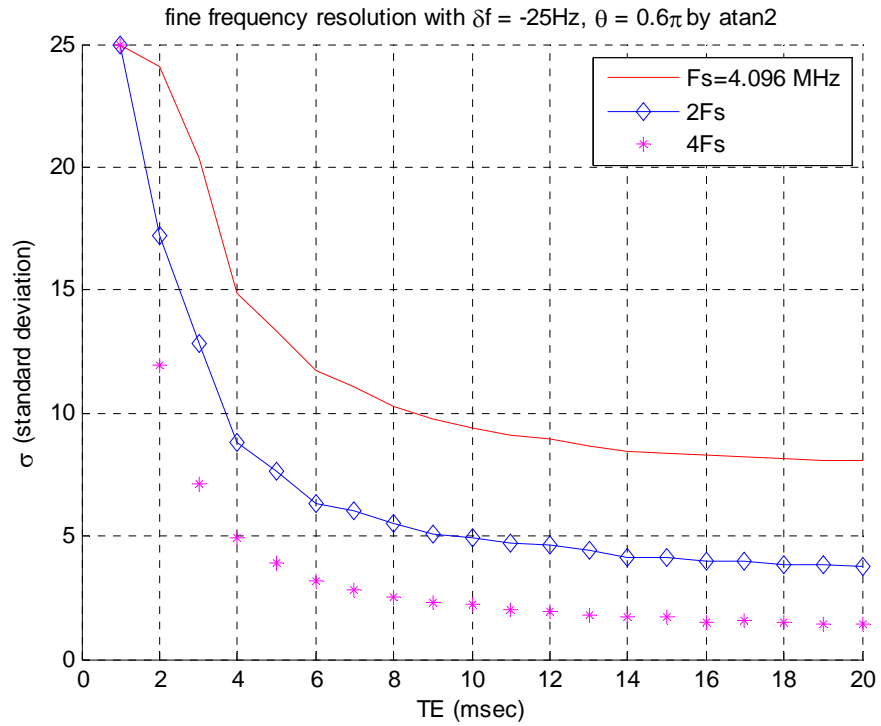


Figure 3.11: Frequency offset performance under different f_s .

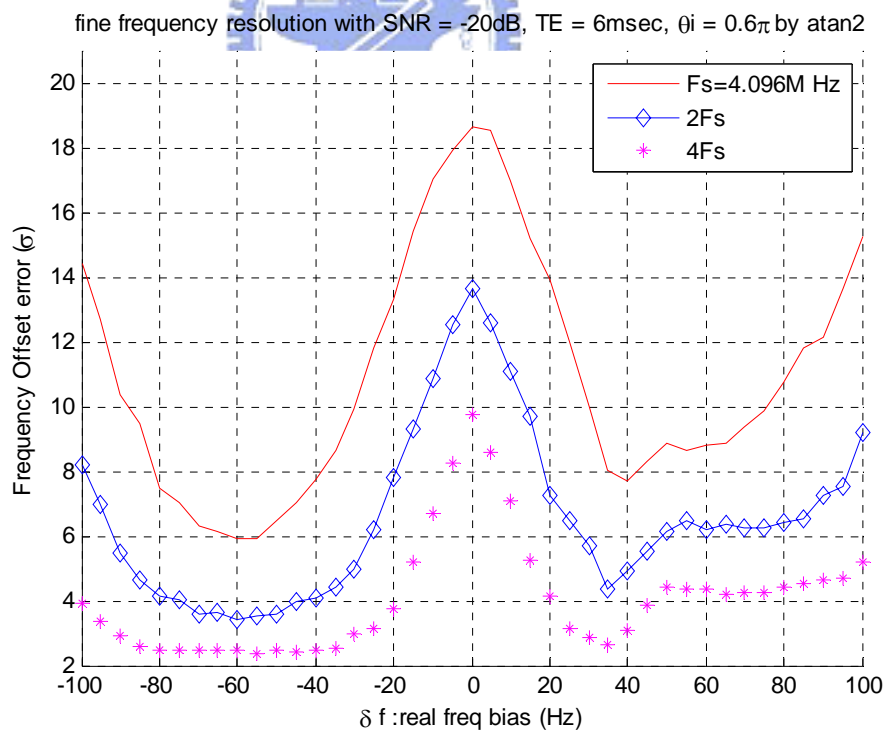
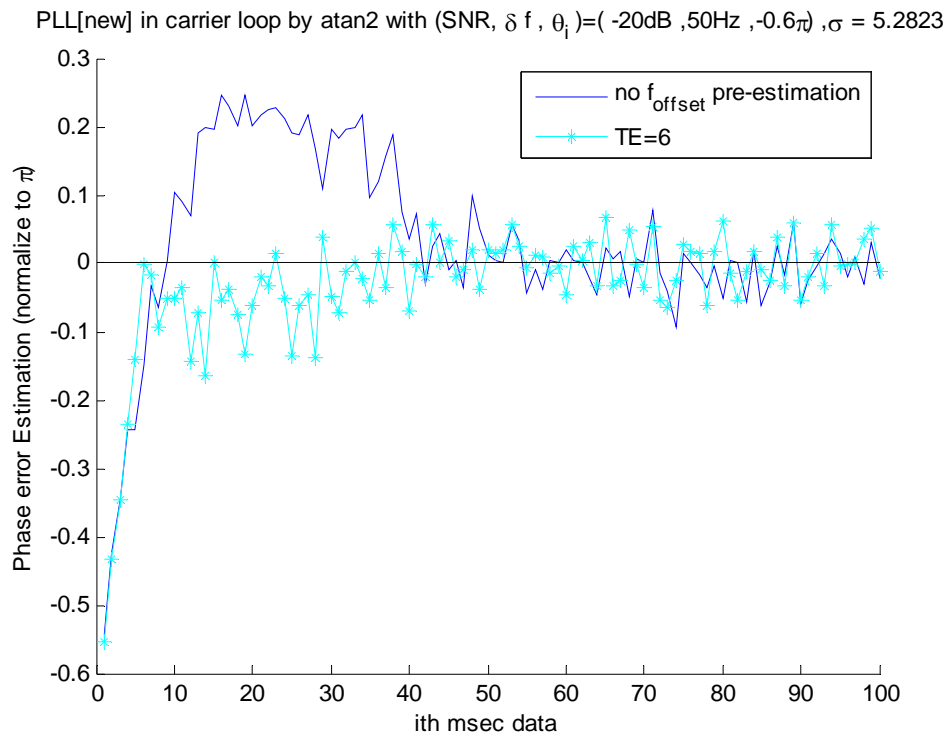


Figure 3.12: Δf versus frequency offset error under different f_s .

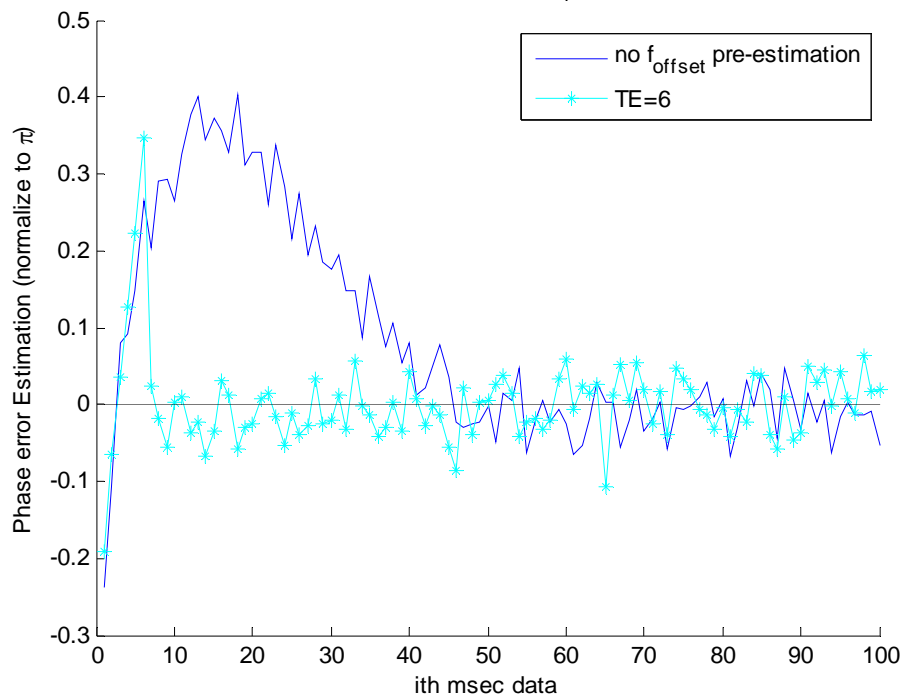
E. Carrier tracking loop implementation

From the above simulation results, we implement the proposed scheme on carrier tracking loop. The frequency tracking performance for 50 Hz initial frequency offset and various phase shift θ are shown in Figure 3.13 (a)-(d). It is clear that our proposed approach can significantly reduce the required lock-in time. Although it has to waste 6 msec, lock is achieved much more rapidly with precise frequency and phase evaluation.



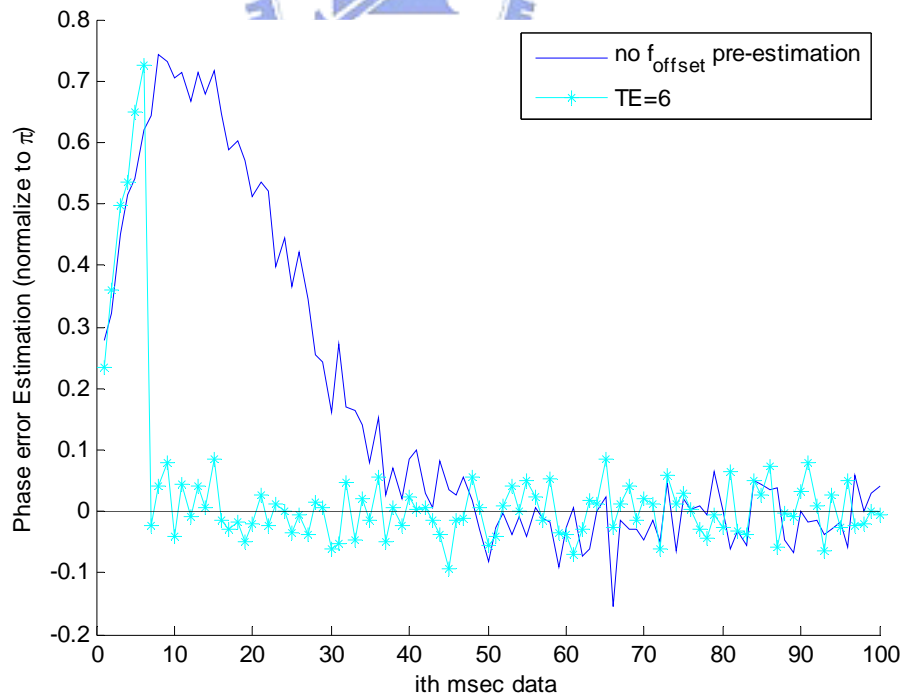
(a)

PLL[new] in carrier loop by atan2 with $(\text{SNR}, \delta f, \theta_i) = (-20\text{dB}, 50\text{Hz}, -0.2\pi)$, $\sigma = 3.7149$

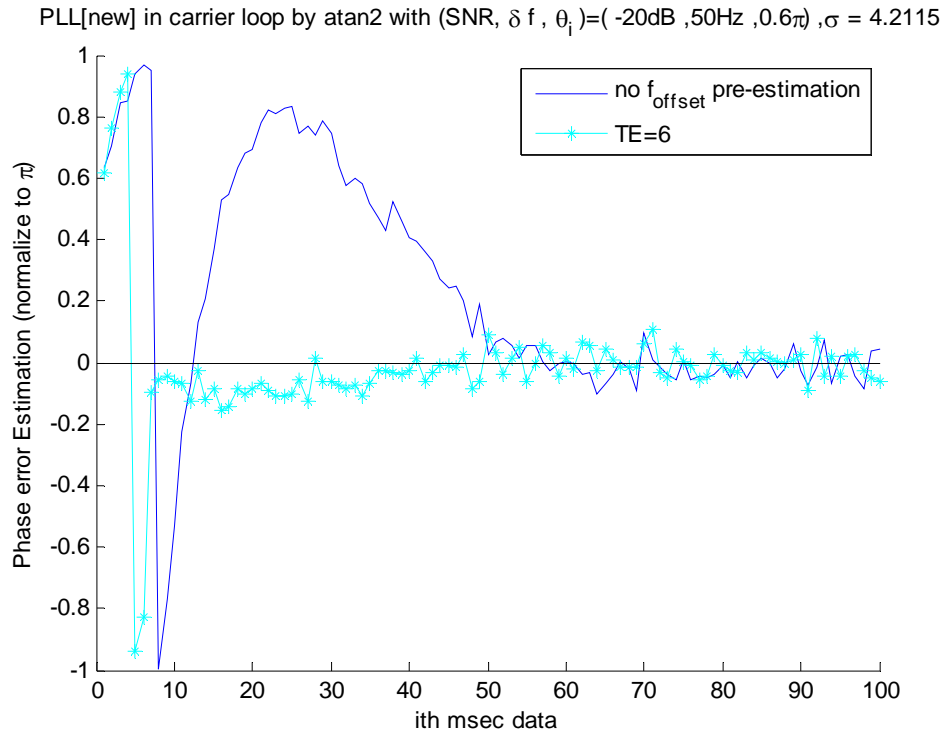


(b)

PLL[new] in carrier loop by atan2 with $(\text{SNR}, \delta f, \theta_i) = (-20\text{dB}, 50\text{Hz}, 0.2\pi)$, $\sigma = 1.0749$



(c)



(d)

Figure 3.13: Resulting error signals for zero frequency initialization and 6 msec estimation time (50 Hz initial offset and various phase shift.)

3.5 Summary

In this chapter, we propose a frequency offset measurement for improving tracking performance. Being applied to the slow-varying cases, we derive phase detection algorithm to execute offset pre-estimation and update the initial condition. By means of frequency offset and phase shift pre-estimation, we can improve the tracking performance even in the low SNR region. As shown in this chapter, such processing allows for more rapid signal tracking. To cooperate with a suitable increase of sampling frequency, the carrier frequency was within 10 Hz in error in above simulation results. Thus the software-based finer resolution algorithm has been successfully validated.

Chapter 4

Virtual Carrier Table for One-Bit Quantized Software Receivers

When we begin to implement our PLL, the NCO should be able to produce a sinusoid with variable frequency. There are many well-known methods for performing functional mapping from phase to sine amplitude such as ROM look-up, coarse/fine segmentation into multiple ROM's, the Nicholas method, the use of Taylor series, CORDIC algorithm. Also, the comparison of the total compression ratio, the size of the memory, the worst case spur level, and additional circuits and their effect on distortion and trade-offs are investigated in detail [9]-[11].

Because the possible number of generated frequencies is large and computing sine values would require a prohibitive amount of computation and program complexity, a pre-built look-up table is usually a better alternative in GPS software receiver.

In this chapter, an elementary description of table-lookup method for the conventional NCO will first be introduced. However, while we enhance the frequency resolution by the method in Chapter 3, it requires a large sine table. Unfortunately, large table means high volume memory and high cost. In place of table look-up method, we propose a different approach under one-bit quantization which uses an iterative computation algorithm.

4.1 Carrier generator

First, the common way to produce a table of discrete angle θ_i and $\sin \theta_i$ is introduced in [9]. Suppose a sine table stores N (usually a power of 2) samples from one cycle of the waveform, each sample can be expressed as

$$\sin \theta_i = \sin\left(\frac{2\pi}{N}i\right), \quad i = 0, \dots, N-1, \quad (4.1)$$

which is the mapped discrete amplitude of the sine waves with discrete frequencies $\omega = 2\pi/N$. Thus we can pre-build a phase to sin amplitude table shown as Figure 4.1 where we use n bits to represent the truncated phase address and the word-length of the ROM is m bits. Then the size of the ROM required to store this sin table is $2^n \times m$ bits.

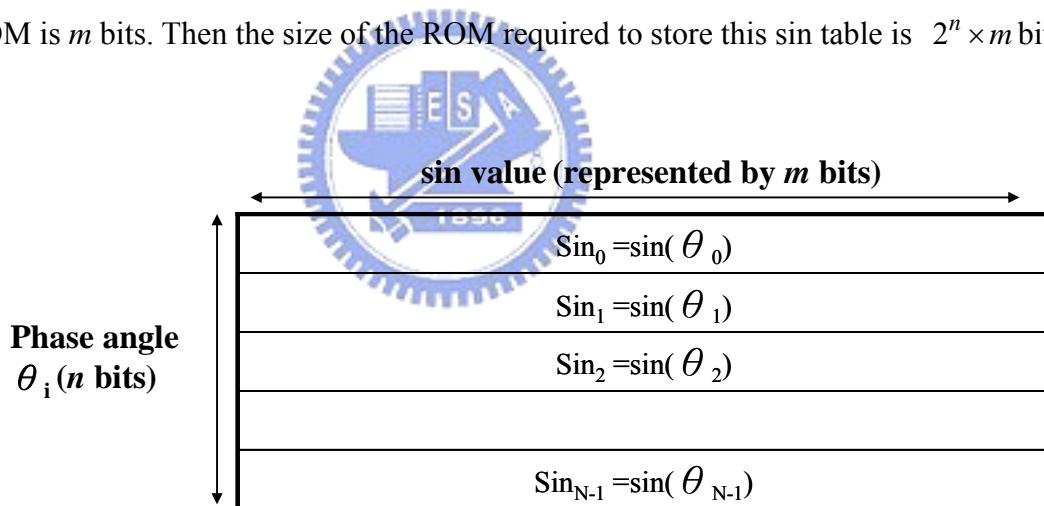


Figure 4.1: The general sin table with full range of 2π .

To return the sine value of an angle by looking up the table, we need to determine the discrete phase angle expressed in radian (rad.) first. To simplify the interpolation problem, we usually select the “nearest phase angle” interpolation to approximate the in-between sine samples. It is obvious that the resolution of the values stored in the sine table determine the spectral purity of the conventional NCO. Nevertheless, while we enhance the accuracy of our frequency resolution in NCO from 200Hz to 20 Hz with the

approach in Section 3.2, the sin table has to be pre-built with 10 times precision by brute force method. In other words, we will require $(2^n \times 10) \times m$ bits memory to store the full sin table. Even though there is a well-known technique of compression to generate the ROM samples for the full range of 2π by only $\pi/2$ rad. of sine information and another additional logic necessary, the pre-built sin table look-up method seems to be inefficient.

4.2 Virtual carrier table

From above description, the memory size is important for us to build the overall sin table that we need to evaluate the NCO output. However, since one-bit quantization is employed in the SDR, only the polarity of the carrier is of concern. In order to simplify the carrier generator, we propose a universal approach for one-bit quantized carrier generator. The key feature of the virtual carrier table (VCT) is that we determine the NCO output value being either positive or negative from the relationship between the previous result and the variation of phase angle.

Let the generated carrier table contain N carriers, in which the carrier frequencies are given as

$$f_m = f_0 + m \cdot f_u \quad , \quad m = 1, 2, \dots, N \quad , \quad (4.2)$$

where f_0 is a fixed frequency and f_u is the minimum spacing between frequency bins. Assume $\sin 2\pi f_m t$ is one of the carriers to be generated. In discrete system, $t = k / f_s$, $k=0,1,2,3, \dots$, are the sampling instants of the one-bit quantized SDR of concern while k is the discrete time index and f_s is the sampling frequency. Thus the sampled signal of $\sin 2\pi f_m t$ can be written in

$$\sin \theta_k = \sin 2\pi \frac{f_m}{f_s} k = \sin 2\pi \frac{f_0 + mf_u}{f_s} k. \quad (4.3)$$

Since the result of carrier generation is deeply affected by the variation of phase angle between two discrete time, now we attempt to find out the relationship between two phase angles, that is, θ_{k-1} and θ_k at discrete time $k-1$ and k , which is written as

$$\theta_{k-1} = 2\pi \frac{f_0 + mf_u}{f_s} (k-1), \quad (4.4)$$

$$\theta_k = 2\pi \frac{f_0 + mf_u}{f_s} k. \quad (4.5)$$

From Equations (4.4) and (4.5), the relationship between changes of phase angle and system coefficients is obtained; the variation, $\Delta\theta$, can be written in terms of the parameters (f_0, m, f_u, f_s) as follows:

$$\Delta\theta = 2\pi \frac{f_0 + mf_u}{f_s}. \quad (4.6)$$

4.2.1 Algorithm

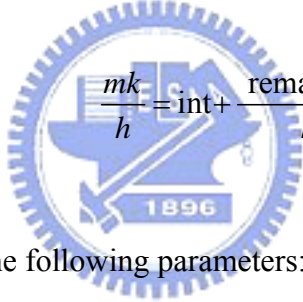
Assume $f_u = \frac{f_0}{h}$ and we choose the sampling frequency such that $f_s = q \cdot f_0$. In general, q is a relatively small number while h is a big one. For example, if $f_0 = 10^6 \text{ Hz}$, $q = 4$ and $h = 10^4$, then $f_u = 10^2 \text{ Hz}$ and $f_s = 4 \times 10^6 \text{ Hz}$. Under these assumptions, the phase angle of (4.3) can be expressed as

$$\theta_k = 2\pi \frac{f_0}{f_s} k + 2\pi \frac{mf_u}{f_s} k = \underbrace{\frac{2\pi}{q} \cdot k}_{\alpha_k} + \underbrace{\frac{2\pi}{q} \cdot \frac{mk}{h}}_{\beta_k}. \quad (4.7)$$

In the case, we can partition θ_k into two variables denoted as

1. $\alpha_k = \frac{2\pi}{q} \cdot k$ is a fast-varying phase angle changing at a step of $\frac{2\pi}{q}$.
2. $\beta_k = \frac{2\pi}{q} \cdot \frac{mk}{h}$ is a slow-varying one since h is large.

We further analyze how and when the slow-varying factor affects the carrier generator. The variation would be classified into two categories, an integer plus an irreducible fraction, as be shown below:



$$\frac{mk}{h} = \text{int} + \frac{\text{remainder}}{h}. \quad (4.8)$$

Furthermore, we define the following parameters:

$$b_k = \left\lfloor \frac{mk}{h} \right\rfloor \bmod q, \quad b_k \in (0, 1, \dots, q-1), \quad (4.9)$$

$$c_k = \frac{mk}{h} - \left\lfloor \frac{mk}{h} \right\rfloor, \quad (4.10)$$

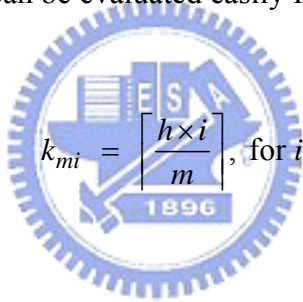
where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x . Based on the above definitions, Equation (4.7) can be written in

$$\theta_k = \alpha_k + \beta_k = \frac{2\pi}{q} \cdot k + \frac{2\pi}{q} \cdot \left(\left\lfloor \frac{mk}{h} \right\rfloor + c_k \right) = \frac{2\pi}{q} \cdot (k + b_k) + \frac{2\pi}{q} \cdot c_k. \quad (4.11)$$

Form the above equation, it is obvious that the term $k + b_k$ will be the critical factor affecting the change of phase angle. For a given m , corresponding to a specific carrier frequency $f_m = f_0 + mf_u$, we can calculate the value of k , denoted as k_{mi} , at which b_k changes its value from $b_k = i - 1$ to $b_k = i$ based on the definition of b_k . Assume S (it will be discussed later) is the maximum size of the particular time index set where k_{mS} is the reset point, the change of b_k can be detected by determining k as follows:

$$b_k = 0 \xrightarrow[k_{m1}]{} 1 \xrightarrow[k_{m2}]{} 2 \rightarrow \dots (i-1) \xrightarrow[k_{mi}]{} i \rightarrow \dots (q-1) \xrightarrow[k_{mq}]{} 0 \xrightarrow[k_{m(q+1)}]{} 1 \rightarrow \dots \xrightarrow[k_{mS}]{} j. \quad (4.12)$$

On the other hand, k_{mi} can be evaluated easily from



$$k_{mi} = \left\lceil \frac{h \times i}{m} \right\rceil, \text{ for } i = 1, 2, \dots, S, \quad (4.13)$$

where $\lceil x \rceil$ denotes the ceiling function, which returns the smallest integer not less than x , or, formally, $\lceil x \rceil = \min\{n \in \mathbb{Z} \mid x \leq n\}$.

In practice, we can calculate the set $(k_{m1}, k_{m2}, \dots, k_{mS})$ for every possible m in (4.2) by simply computing (4.13) and store them in memory. This can be done easily before carrier generation is actually performed.

When carrier generation for a particular frequency $f = f_0 + mf_u$ is required, the corresponding polarity of $\sin \theta_k$ is obtained as follows. In the beginning, let $sign[\sin \theta_k]$ be the polarity function of $\sin \theta_k$, defined as

$$\begin{aligned} \text{sign}[\sin \theta_k] &= 1, & 0 \leq \theta_k < \pi, \\ &= -1, & \pi \leq \theta_k < 2\pi. \end{aligned} \quad (4.14)$$

Next, referring to θ_k from Equation (4.11), a new integer denoted u_k is now given as

$$u_k = k + b_k, \quad (4.15)$$

Under some manipulations from (4.12) and (4.15), hence we can derive the value of u_k from u_{k-1} as

$$u_k = \begin{cases} u_{k-1} + 1, & \text{if } k \neq k_{mi} \\ u_{k-1} + 2, & \text{if } k = k_{mi}, i = 1, 2, \dots, S. \end{cases} \quad (4.16)$$

Therefore u_k is generally expressed as

$$u_k = u_{k-1} + x, \quad (4.17)$$

where $x = 1$ if $k \neq k_{mi}$ and $x = 2$ if $k = k_{mi}$.

There is an important assumption that q in (4.11) must be an even number, given $q = 2p$, where p is an integer. Then Equation (4.11) can be rewritten as

$$\theta_k = \frac{\pi}{p} \cdot u_k + \frac{\pi}{p} \cdot c_k. \quad (4.18)$$

In the following we shall consider the general case with $p \geq 2$ while the special case of $p = 1$ will be treated later. The simple expressions are exploited to determine

the variant phase angle in order to obtain one-bit quantized result of carrier generator. To consider $sign[\sin \theta_k]$ for $0 \leq k < k_{mS}$ in part A, B.

A. General case

In the beginning, for the general case of $p \geq 2$, we define

$$v_k = u_k \bmod p. \quad (4.19)$$

Moreover, from (4.17) and (4.19), we obtain the following relationship:

$$\begin{aligned} v_k &= u_k \bmod p \\ &= (u_{k-1} + x) \bmod p \\ &= [(u_{k-1} \bmod p) + x] \bmod p \\ &= [v_{k-1} + x] \bmod p. \end{aligned} \quad (4.20)$$

Combining Equations (4.18) and (4.20), we just determine $sign[\sin \theta_k]$ from $sign[\sin \theta_{k-1}]$ as

$$sign[\sin \theta_k] = \begin{cases} sign[\sin \theta_{k-1}], & \text{if } v_k = v_{k-1} + x < p, \\ -sign[\sin \theta_{k-1}], & \text{otherwise.} \end{cases} \quad (4.21)$$

Therefore the polarity of $sign[\sin \theta_k]$ can be easily determined if v_{k-1} and $sign[\sin \theta_{k-1}]$ are given. Consequently, if the initial parameters $v_0 = 0$ and $sign[\sin \theta_0] = 1$ are given, v_k and $sign[\sin \theta_k]$, $k = 1, 2, \dots, k_{mS} - 1$, all can be obtained via (4.20) and (4.21).

B. Special case

On the other hand, for the special case of $p=1$, Equation (4.17) still holds and (4.18) becomes

$$\theta_k = \pi \cdot u_k + \pi \cdot c_k. \quad (4.22)$$

If $\text{sign}[\sin \theta_{k-1}]$ is given, the polarity of $\text{sign}[\sin \theta_k]$ is simply given as

$$\text{sign}[\sin \theta_k] = \begin{cases} -\text{sign}[\sin \theta_{k-1}], & \text{if } k \neq k_{mi} \text{ (that is } u_k = u_{k-1} + 1), \\ \text{sign}[\sin \theta_{k-1}], & \text{if } k = k_{mi} \text{ (that is } u_k = u_{k-1} + 2). \end{cases} \quad (4.23)$$

Consequently, based on the proposed algorithm, we can generate all the virtual carrier tables for $q = 2p$, $p = 1, 2, 3, \dots$

C. Reset point

Finally, we consider $\text{sign}[\sin \theta_k]$ for $k \geq k_{mS}$. The particular reset time index k_{mS} can be resolved with careful observation from Equations (4.8) and (4.10). Obviously, when $k = h$, the parameter $c_k = 0$. Therefore the particular time index $k_{mS} = h$ should be the reset point while $b_k = m \bmod q$ and $S = m$ at the same time. In this way, the size of $(k_{m1}, k_{m2}, \dots, k_{mS})$ that we should calculate and store would be equal to m for every possible m .

However, for a given m and h decided from a particular frequency $f = f_o + mfu$, there may be a further simplification for the parameter size S . Assume $\text{gcd}(m, h)$ is the greatest common divisor of m and h , Equations (4.8) can be rewritten as

$$\frac{mk}{h} = \frac{\frac{m}{\gcd(m,h)} \times k}{\frac{h}{\gcd(m,h)}} = \text{int} + \frac{\text{remainder}}{\frac{h}{\gcd(m,h)}}. \quad (4.24)$$

And then the reset condition will be changed to

$$c_k = 0, \text{ when } k = \frac{h}{\gcd(m,h)} = k_{mS}. \quad (4.25)$$

At the same time,

$$b_k = \frac{m}{\gcd(m,h)} \bmod q. \quad (4.26)$$

Consequently, the particular time index $k_{mS} = \frac{h}{\gcd(m,h)}$ should be the reset point and the maximum size of S , would be set as $\frac{m}{\gcd(m,h)}$ for every m .

As (4.20) and (4.21) still holds for $k = k_{mS}$, we can obtain $v_{k_{mS}}$ and $\text{sign}[\sin \theta_{k_{mS}}]$ as before. Since $b_k = \frac{m}{\gcd(m,h)} \bmod q$ at $k = k_{mS}$, we can set $v_{k_{mS}}$ and $\text{sign}[\sin \theta_{k_{mS}}]$ as the initial condition for the new signal sequence, i.e. $k = k_{mS} \rightarrow k = 0$, $v_{k_{mS}} \rightarrow v_0$ and $\text{sign}[\sin \theta_{k_{mS}}] \rightarrow \text{sign}[\sin \theta_0]$, and apply the same algorithm to generate carrier table for $k \geq k_{mS}$. Accordingly, the carrier generation can be continuously generated without length constraint.

4.2.2 System description

The proposed algorithm to generate virtual carrier table are implemented as follows (see Figure 4.2 and Figure 4.3).

In the figure, $\sin VCT_k$ corresponds to $\text{sign}[\sin \theta_k]$ using VCT. The initial conditions are set as: $k = 0$, $v_0 = 0$ and $\text{sign}[\sin \theta_0] = \text{sign}[\sin 0] = 1$. The parameter set $(k_{m1}, k_{m2}, \dots, k_{mS})$ is pre-calculated and stored in the memory for every m . For example, if $q=4$, $h=10^4$ and $m=7$, then $(k_{m1}, k_{m2}, \dots, k_{m7}) = (1429, 2858, 4286, \dots, 10^4)$. Based on the algorithm, the polarity of $\sin \theta_k$, $k > 0$, can be obtained for any carrier frequency $f = f_o + mf_u$.



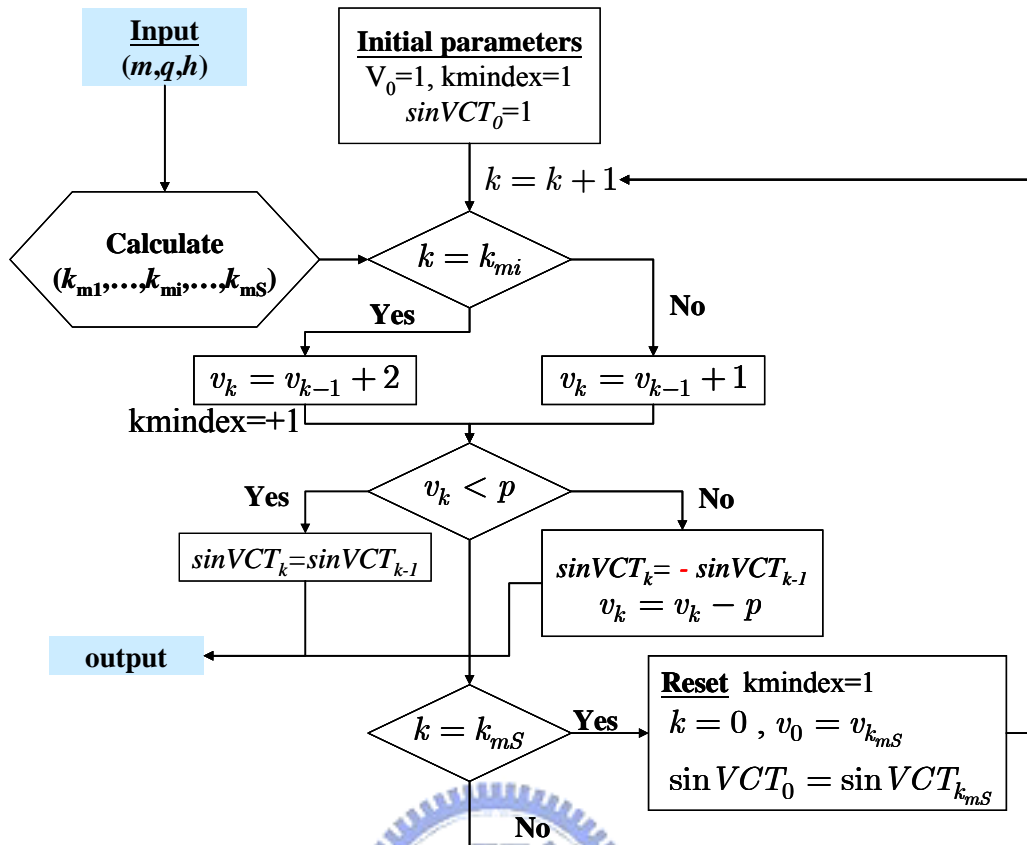


Figure 4.2: Flowchart of VCT generator for $p \geq 2$.

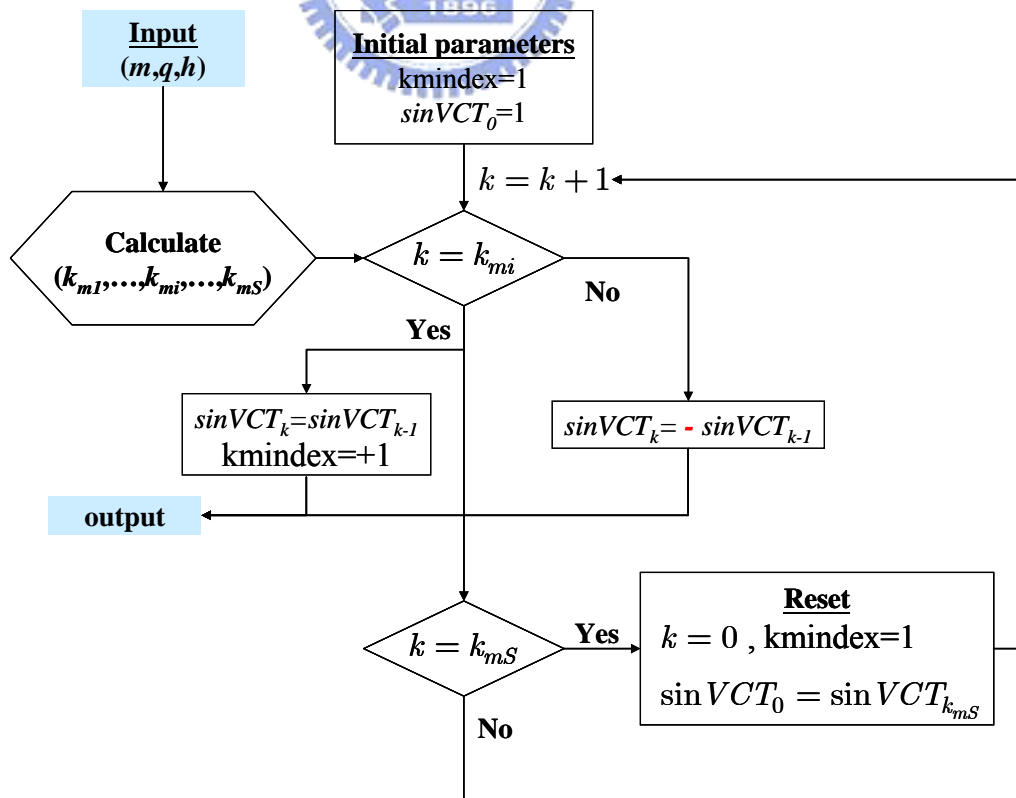


Figure 4.3: Flowchart of VCT generator for $p=1$.

4.3 Computer simulations

As the derivation in Section 4.2.1, it is unnecessary to pre-store all the particular decision parameters $(k_{m1}, k_{m2}, \dots, k_{mS})$ for every possibly m . Instead, the parameters will be evaluated immediately via numerical calculation and stored in memory when the frequency and system parameters are decided. To test and verify the accuracy of the proposed virtual carrier table approach, we take some examples of frequencies and system parameters for simulation, shown in Table 4.1. In these cases, the carrier frequency of NCO is $f_c = f_0 + mf_u$, where $f_0 = 10^6$ Hz, $f_u = \frac{f_0}{h} = 100$ Hz, $f_s = q \times f_0$ Hz.

m	q	h	S	$(k_{m1}, k_{m2}, \dots, k_{mS})$
3	4	10^4	3	(3334,6667,10000)
7	2	10^4	7	(1429,2858,4286,5715,7143,8572,10000)
7	4	10^4	7	(1429,2858,4286,5715,7143,8572,10000)
14	4	10^4	7	(715,1429,2143,2858,3572,4286,5000)

Table 4.1: Frequency and system factors.

First, Figure 4.1 shows the errors vs. discrete time index k of proposed approach that are compared with the $\text{sign}[\sin \theta_k]$ generated by Matlab programming, denoted $\text{sin}_p[k]$. It is evident that the resulting carrier using VCT is not completely matched with the one by programming. To understand the cause of the error points, we examine the corresponding time index k of the error point and analyze the phase angel accordingly, as resulted in Table 4.2.

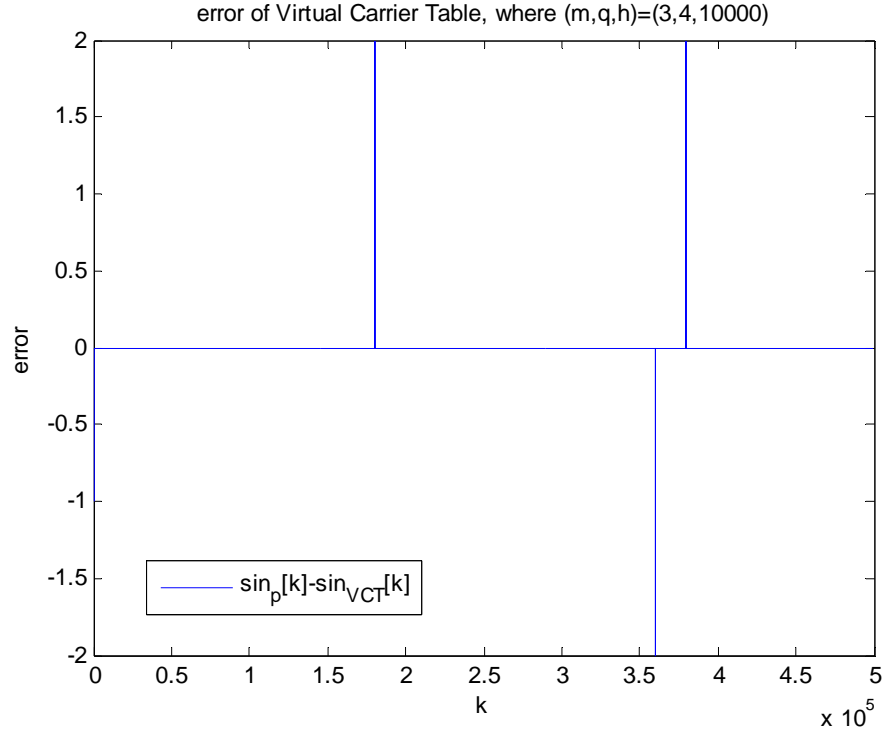


Figure 4.4: The errors between programming and VCT vs. discrete time index k .

k	$\theta_k = 2\pi f_c \frac{k}{f_s}$	$\sin \theta_k$ in programming
0	0π	0
180000	90027π	+1.1824e-011
360000	180054π	-2.3649e-011
380000	190057π	+8.3170e-011

Table 4.2: The analysis of the corresponding phase angel at the error point

for $m=3, f_c=1000300\text{Hz}, f_s=4\text{ MHz}$.

In Table 4.2, the first error point results from the different definition of initial condition ($\theta=0$) that we assume $sign(\sin 0)=1$ and $sign(\sin 0)=0$ in Matlab programming. Moreover, the second row ($k=180000, \theta_k=90027\pi$) should be $\sin \theta_k=0$, by definition. But a little inaccuracy leads to $\sin \theta_k=+1.1824e-011$ in programming. Similar results happen for $k=360000$ and $k=380000$.

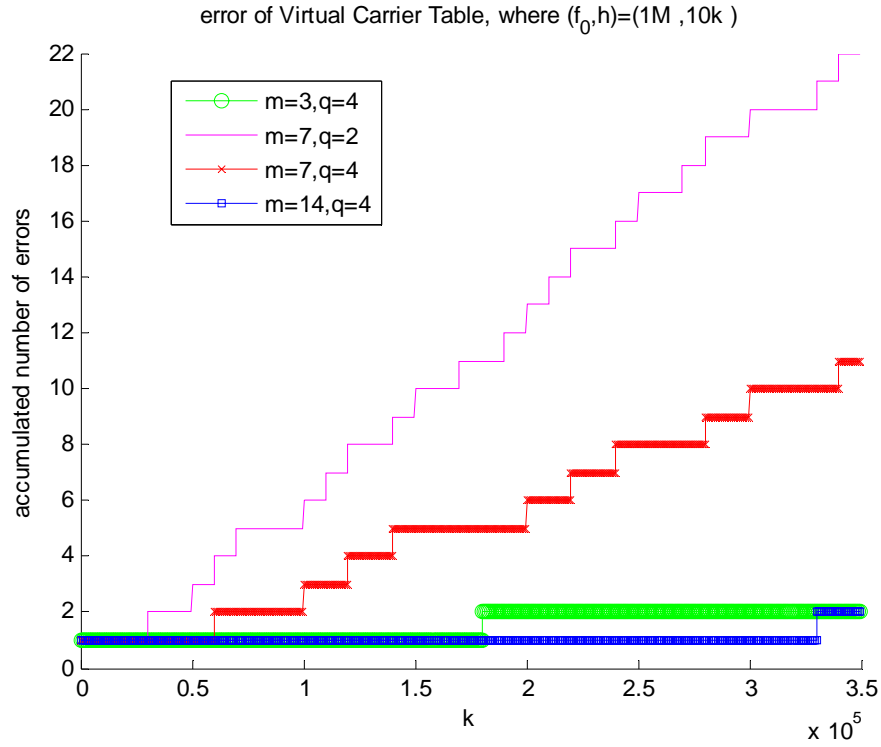


Figure 4.5: The accumulated number of errors vs. time index k for 4 simulation examples.

To illustrate the correctness of the proposed carrier generator, we present several simulation examples as shown in Figure 4.5. Within the total of 3.5×10^5 simulation points, the number of error points is less than 22 and all are due to numerical error in calculating multiple of π . Thus the validity of the proposed VCT is justified.

In Figure 4.6, we show another simulation example when it is used for specific center frequency denoted f_{acq} after fine search by pull-in function block and a tuned frequency offset denoted Δf from tracking in Chapter 3. Considering that the modified frequency estimation is under 10Hz and the minimum spacing between frequency bins, f_u , is dominated by some definition in Section 4.2.1. VCT could be implemented by assuming that the fixed frequency $f_0 = f_{acq}$ and Δf can be expressed as $m \times f_u$, where m is an integer and $f_u = f_0/h$. As an illustration, $f_0 = 11324200$ Hz,

$f_s = q \times f_0$ Hz, $h = 10^7$, so $\Delta f = m \times 1.13242$ Hz. The proof of virtual carrier table is shown in Figure 4.6. From the result, it can be proved that the simulation result is correct for all the time index k within $k \leq 10^5$.

m	q	h	S	$(k_{m1}, k_{m2}, \dots, k_{mS})$
4	2	10^7	1	(2500000)
7	2	10^7	7	(1428572, 2857143, 4285715, 5714286, 7142858, 8571429, 10000000)
7	4	10^7	7	(1428572, 2857143, 4285715, 5714286, 7142858, 8571429, 10000000)

Table 4.3: Frequency and system factors for $f_0=11324200$ Hz, $f_u=1.13242$ Hz.

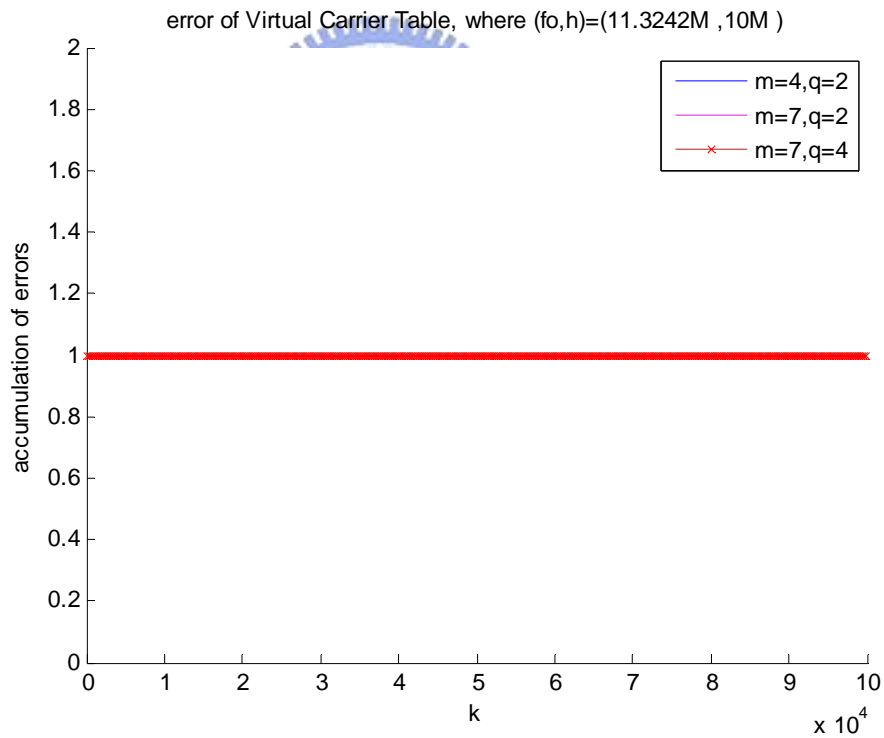


Figure 4.6: The accumulated number of errors vs. time index k for $f_0=11324200$ Hz, $f_u=1.13242$ Hz.

4.4 Summary

In Section 4.1, we introduce the general solution with respect to the digital carrier generation. Because the key feature of NCO under one-bit quantization is related to the preceding output and the variant of phase angle, we find a simpler approach called virtual carrier table to generate carrier in Section 4.2. Further, the variation of phase angle are derived and expressed in terms of a set of specific parameters. In particular, there is a reset time, $k=h/gcd(m,h)$ such that the VCT can be generated continuously. From the simulation results, the correctness of VCT is justified.

In the proposed VCT generation method, we merely have to compute N sets of parameters, i.e. $(k_{m1}, k_{m2}, \dots, k_{mS})$, $m = 0, 1, \dots, N - 1$, and store them in the memory. The total number of stored parameters is much less than the pre-established carrier table.



Chapter 5

Conclusions

In this thesis, we consider a software-based receiver to acquire and track the frequency and code phase of satellite signals and then obtain the navigation data. In order to enhance the frequency resolution, we propose to adopt frequency offset estimation for better tracking loop performance. To reduce memory space for fine frequency generation, we further design a simple one-bit quantized virtual carrier generation algorithm.

In Chapter 2, we introduce the popular software technique which can offer great flexibility. By the parallel acquisition technique using FFT, we can figure out which satellite signal is received and obtain the coarse carrier frequency and the beginning of C/A code (phase), where the frequency resolution is 1kHz. Using a simple time domain correlation in the pull-in process, we achieve a more precise frequency with resolution of 200 Hz. Then the signal can be tracked and demodulated to obtain navigation data by a general tracking loop which contains DLL as the code loop and PLL as the carrier loop. However, from the result, it must be noted that the mismatch between NCO and incoming signal causes a transient period for tracking loop to lock the signal.

To improve the performance of tracking, a frequency offset measurement scheme is proposed in Chapter 3. It is designed via average phase estimation under the slow-varying land vehicle assumption. The frequency offset between NCO and incoming signal can therefore be measured. Using the estimated frequency offset and phase shift to modify the initial guess of NCO, the tracking performance is much improved even in the low SNR region. The simulations show that the carrier frequency resolution is within ± 10 Hz cooperated with a suitable increase of sampling frequency.

In Chapter 4, increasing the accuracy of local carrier frequency leads to large memory space for the carrier table of the NCO. Under one-bit quantization, we propose a virtual carrier generation algorithm with simple numerical calculation which results in very small memory space. In particular, there is a reset time, $k_{mS} = h/gcd(m, h)$ such that the virtual carrier table can be generated continuously by applying the same algorithm for $k \geq k_{mS}$. It is verified that our VCT approach merely need to compute and store N sets of parameters, i.e. $(k_{m1}, k_{m2}, \dots, k_{mS})$, $m = 0, 1, \dots, N-1$ for N carrier frequencies. Therefore, the required memory space is much less than that of a pre-established look-up table. To implement the NCO using the virtual carrier table in practice, the carrier frequency and sampling frequency can be designed to satisfy the requirement easily.

The study presented in the thesis includes a new tracking scheme by modifying the initial tracking loop structure for slow-varying land vehicles or users. In particular, we derive the bound to select the reasonable estimated frequency and get averaged frequency offset to make the frequency mismatch between NCO and incoming carrier within ± 10 Hz. Considering fast-varying vehicles or users, another assisted scheme is needed to track the variation of carrier frequency, such as FLL (frequency locked loop). Additionally, the possibility to replace the pull-in process with frequency offset estimation is considered since the result in Chapter 3 is attractive. We then take the

frequency resolution of 1k Hz into account to derive the corresponding limitation.

Except the pull-in process, two considerations require further considerations. One is the complexity of dealing with the selection of frequency search set in acquisition and the limitation in the new tracking scheme. The other is that the processing time of the modified algorithm is less than the pull-in process. The derivation of total solution is thus a problem worthy of investigation in the future.



Bibliography

- [1] P. Rinder, N. Bertelsen, "Design of a single frequency GPS software receiver," Aalborg University, Denmark, Master thesis, 2004
- [2] J. B. Y. Tsui, "Fundamentals of Global Positioning System Receivers : A Software Approach," 2nd Edition, John Wiley & Sons Inc., 2005.
- [3] E. D. Kaplan, C. J. Hegarty, "Understanding GPS: Principles and Applications," 2nd edition, Artech House Incorporated, Norwood, Mass, USA, 2006.
- [4] B. Schipper, M. Braasch, J. Campbell, S. Nair, "A testbed for frequency domain GPS acquisition and receiver simulation," AIAA PLANS, pp.471-477, 2004.
- [5] D. J. R. Van Nee, A. J. R. M. Coenen, "New fast GPS code acquisition technique using FFT," *IEEE Electronic Letters*, Vol. 27, No. 2, 1991.
- [6] A. V. Oppenheim, R. W. Schaffer, "Digital signal processing," 2nd Edition, Prentice-Hall Inc., New Jersey, 1999.
- [7] D. M. Akos, "A software radio approach to global navigation satellite system receiver design," School of Electrical Engineering and Computer Science, Ohio University, Athens, Ohio, Ph.D. Dissertation, August 1997.
- [8] L. Schuchman, "Dither signals and their effect on quantization noises," *IEEE Trans. Communications*, vol. 12, no. 4, pp.162–165, 1964.
- [9] S. Mehrgardt, "Noise spectra of digital sine-generators using the table lookup method," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1037–1039, Aug. 1983.
- [10] J. Vankka, "Methods of mapping from phase to sine amplitude in direct digital synthesis," *IEEE Trans. ultrasonics, ferroelectrics, and frequency control*, vol. 44, no 2, pp.526-534, 1997.
- [11] V. Kantabutra, "On hardware for computing exponential and trigonometric

functions," *IEEE Trans. Computers*, vol. 45, no. 3, pp.328–339, 1996.

