UCGE Reports

Number 20208

Department of Geomatics Engineering

Improving Tracking Performance of PLL in High Dynamic Applications

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

by

Ping Lian

November 2004

THE UNIVERSITY OF CALGARY

Improving Tracking Performance of PLL in High Dynamic Applications

by

Ping Lian

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, 2004

ABSTRACT

The Phase-locked loop (PLL) is used in GPS receivers to track an incoming signal and to provide accurate carrier phase measurements. However, the PLL tracking performance and measurement accuracy are affected by a number of factors, such as signal-to-noise power ratio, Doppler frequency shift, the GPS receiver's jitter caused by vibration, and the Allan deviation. Among these factors, the thermal noise and Doppler shift are the most predominant and have a large influence on the design of the PLL. In high dynamic situations, the conflict between improving PLL tracking performance and the ability to track the signal necessitates some compromises in PLL design. This thesis investigates the strategies to resolve this conflict.

Three methods are investigated to improve PLL tracking performance in high dynamic applications: a Kalman filter-based tracking algorithm, application of a wavelet de-noising technique in PLL, and an adaptive bandwidth algorithm. The Kalman filter-based tracking algorithm makes use of a carrier phase dynamic model and a measurement from the output of the discriminator to estimate the phase difference between the incoming signal and the Numerical Controlled Oscillator (NCO) output, Doppler frequency and the change rate of Doppler frequency. The wavelet de-noising technique effectively decreases the noise level and allows broadening of the PLL bandwidth to track high dynamics signals. The adaptive bandwidth PLL algorithm adapts the bandwidth of the PLL according to the estimation of the incoming signal dynamics and noise level.

The performance is evaluated in terms of signal-to-noise ratios and dynamic variations using simulating signals. The first two methods are found to produce better improvements when the signal-to-noise ratio is low and the signal dynamic is high. The third method works well under high signal-to-noise ratios and less random dynamic variations.

The results show that these methods outperform the ordinary PLL under high dynamic conditions and the resulting carrier phase measurement is more accurate.

ACKNOWLEDGEMENTS

I would like thank my supervisor Dr. Gérard Lachapelle for his support and guidance throughout my graduate studies. His continuous inspiration and advice were greatly appreciated. The lessons I learned from his conscientious attitude and commitment to graduate students will benefit me in future work.

I would also like to thank many graduate students and research associates. Particular thanks go to Dr.Changlin Ma and Olivier Julien for their invaluable help. Many thanks must go to Dr. Mark G. Petovello, Sameet Deshpande, Syed Salman, Zhi Jiang, Dharshaka Karunanayake, Tao Hu, MinMin Lin and Haitao Zhang for the many discussions and advice.

Finally thanks are given to my family, to my husband for his support and understanding, to my son who gives me the motivation to study and live in a new country and my parents for their encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GPS	Global Positioning System
DLL	Delay Lock Loop
PLL	Phase-Locked Loop
PLAN	Position, Location and Navigation
S/N	Signal-to-Noise Ratio
FLL	Frequency-Locked Loop
NCO	Numerical-Controlled Oscillator
LPLL	Linear Phase-Locked Loop
DPLL	Digital Phase-Locked Loop
ADPLL	All-digital Phase-Locked Loop
SPLL	Software Phase-Locked Loop
PD	Phase Detector
LF	Loop Filter
VCO	Voltage-Controlled Oscillator
DCO	Digital-Controlled Oscillator
SNR	Signal-to-Noise Ratio
LNA	Low Noise Amplifier
FPGA	Field Programmable Gate Arrays
C/A	Coarse Acquisition
BPSK	Binary Phase Shift Keying
TTFF	Time to First Fix
RMS	Root Mean Square
I	Inphase
Q	Quadrature
DSP	Digital Signal Processing
DWT	Discrete Wavelet Transform
STFT	Short-Time Fourier Transform
CWT	Continuous Wavelet Transform
IF	Intermediate Frequency
C/N_O	Carrier-to-Noise Ration
dB	deciBel
dBm	deciBel one Milliwatt

CHAPTER ONE: INTRODUCTION

Research and development continue to expand the capabilities and to increase the robustness of GPS receivers. Due to their flexibility, software receivers are quite valuable and convenient in evaluating these potential improvements. For GPS receivers, the technique used to compute the user's position is based on the pseudorange and carrier phase measurements. The common method of pseudorange and carrier phase measurement is to use a Delay-Locked Loop (DLL) for code phase measurements and a Phase-Locked loop (PLL) for carrier phase measurements. Unfortunately, a PLL can not meet precision requirements all the time especially under high dynamic situations and weak signals. Many methods have been developed to address this issue. Compared to existing hardware, software receivers have more flexibility and advantages in improving the tracking performance under high dynamic situations. This thesis investigates three methods of PLL design in the software receiver to improve PLL tracking performance for high dynamic applications.

1.1 Background

In practice, a GPS receiver must create the PRN code and carrier frequency plus Doppler frequency using a DLL and PLL to track the incoming signals by synchronizing its local carrier and code with the incoming signals. The accuracy of the frequency and phase synchronization depends on the parameters of the DLL/PLL, the characteristics of the incoming signals (such as signal-to-noise ratio and Doppler frequency), and the receiver clock quality. In a high dynamics situation, the Doppler frequency changes rapidly with time. This gives rise to a dilemma in GPS receiver design on the pre-integration and PLL bandwidth. To tolerate dynamic stress, the most effective way is to broaden the PLL bandwidth and reduce the pre-integration time. However, in order to decrease the thermal noise and improve the tracking performance, a narrow PLL bandwidth and longer pre-integration time are required. In reality, some compromise must be made to resolve this conflict, especially under a high dynamic stress situation.

A software GPS receiver provides maximum flexibility in the design. It allows for the design scheme to be easily simulated and implemented. It also meets the cost-effective requirement for upgrading the system easily with the development of the new technology – re-configurability. This research takes the advantage of software receiver technology and tests the proposed algorithms in a GPS software receiver developed by the "PLAN" group in our department, namely GNSS_SoftRx™ (Ma et al 2004)

The common method for designing a PLL tracking loop is to choose the loop bandwidth which is mainly determined by the loop filter considering the worst case of S/N and the highest Doppler frequency caused by the dynamics. Usually these designs are robust but not optimal. During a period of low dynamics, the loop bandwidth is not optimal for reducing the tracking errors, as the signal-to-noise ratio is inversely proportional to the loop bandwidth.

FLL (Frequency-Locked Loop) assisted PLL is another widely used method due to its easy implementation (Jovancevic et al. 2003, Krumvieda et al. 2001). Under a low dynamic situation, it can provide better performance due to the narrow bandwidth of PLL. However, under high dynamic situations, it has to switch to FLL in order to offer robustness. In this case the measurement accuracy deteriorates.

Open-Loop assisted Close-Loop PLL is also used in some applications (Yang 2003). This method simply returns to the acquisition process when it operates under a high dynamics. It can guarantee a reliable signal search in a wide spectral range. But the measurements are very noisy therefore it is hard to obtain an accurate GPS carrier phase measurement when it turns to the Open-Loop state.

Macabiau & Legrand (2000) tried to use “an extended Arctan discriminator” to increase the PLL lose-lock threshold under high dynamics. Compared to an Arctan discriminator, the extended Arctan discriminator has nearly a $-\pi$ to $+ \pi$ range. They proposed use of the extended Arctan discriminator in the L5 receiver’s pilot channel to benefit from the absence of data on this channel. For L1/L2 receivers, it is restricted by the data bit synchronization. If synchronization is not achieved, it cannot remove the effect of the data bit transition.

A Doppler aiding carrier tracking loop provides the benefit of mitigating the phase error. The basic concept of the Doppler aiding is to use internal or external Doppler information to adjust the NCO frequency and therefore reduce, or cancel the effect of dynamic stress (Demoz et al 2003). This allows narrowing of the bandwidth of the PLL even in high dynamics. For closed carrier loop operation, the bandwidth of the loop is usually so narrow that the aiding must be very precise with little or no latency.

The internal Doppler aiding is based on the estimation of the Doppler frequency inside the GPS receiver. The most common example is to use the Doppler frequency derived from PLL to aid the DLL through use of a scaling factor (Jovancevic et al. 2003). External Doppler aiding is available from another sensor, i.e. an inertial navigation system (Gold & Brown 2004). However, due to the errors in the sensor, the external Doppler estimate is not always accurate. The disadvantage of this method is that the quality of the sensor largely affects the Doppler estimate accuracy.

Adapting the PLL bandwidth based on the incoming signal dynamics is a better strategy for improving the tracking performance of the PLL (Legrand & Macabiau 2000, Legrand & Macabiau 2001). It was reported to have generated very good results, but no high dynamics test results are presented. The work described herein implements the adaptive bandwidth algorithm in a different way with special attention given to high dynamic applications.

Although emphasis has been given to improving the tracking performance of PLL, the reported accuracy is still not satisfactory specifically for high dynamic applications. Accurate carrier phase measurement is one of the key issues for cm-level positioning which has a broad range of applications. Therefore a thorough investigation and analysis of improving PLL tracking performance are needed.

1.2 Objectives and Contributions

Given the deficiencies of previous PLL designs and the lack of research towards satisfactory improvement of PLL tracking performance in high dynamic situations, this thesis has the following objectives:

1. To develop algorithms to improve the tracking performance of PLL under high signal dynamics;
2. To implement the proposed algorithms in the software receiver developed by the “PLAN” group; and
3. To test and evaluate the performance of the proposed algorithms using a simulator (STR6550) and SignalTap (front end of GPS receiver) produced by the Accord company.

Based on the work described above, three algorithms are proposed that are implemented in software. As mentioned earlier, the thermal noise and dynamic stress are the most common errors and have a large influence on the design of PLL. This thesis focuses on these two main error sources. Previous work did not achieve robustness and optimality at the same time. The proposed algorithms try to improve the PLL tracking performance with robustness and optimality in mind.

The proposed algorithms do not require external aiding and can be easily implemented in a software receiver and therefore are cost-effective from this point of view. In addition, they work in the real-time mode and can be used in many practical applications.

Another purpose of this research is to develop and test algorithms under a variety of high dynamics situations and different signal-to-noise ratios. Previous research assumed low or medium dynamics. This thesis complements the lack of results in this area and presents many results based on simulated high dynamics signals. Comparisons between the traditional design and the proposed designs are given.

1.3 Outline

In Chapter 2 of this thesis, there is a review of PLL theory followed by a detailed explanation of the four main tracking errors and their effects on PLL tracking performance. Following this is an introduction to the COSTAS loop which is widely used in GPS receivers.

The existing GPS software receiver used for the investigations, namely GNSS_SoftRx™, is introduced in Chapter 3, including its structure and a software PLL algorithm.

The proposed algorithms are presented in three Chapters. The Kalman filter-based tracking algorithm is presented in Chapter 4. The Kalman filter algorithm is reviewed first. A carrier phase dynamic model is developed based on the Kalman filter algorithm. This is followed by a description on how to apply the model to the PLL.

Chapter 5 presents the method which applies the wavelet de-noising technique to PLL. The basic wavelet theory is reviewed first. Details of the wavelet de-noising technique are then presented with attention given to the method used in this thesis. The design scheme combining the wavelet de-noising technique with PLL is introduced.

In Chapter 6, the Adaptive Bandwidth PLL Algorithm is presented. This involves the algorithm methodology and the detailed implementation process.

The test results for the proposed algorithms presented in Chapter 4, 5, 6 are shown in Chapter 7. These include a description of the testing configuration, the equipment used and implementations tested.

Conclusions and recommendations are presented in Chapter 8.

CHAPTER TWO: PHASE -LOCKED LOOP REVIEW

GPS receivers track the Doppler frequency and phase to obtain very accurate carrier phase measurements which are one of the important issues for many precise positioning applications. Carrier phase tracking is accomplished using a PLL. This chapter reviews PLL theory. Attention is given to PLL tracking capability with different types of exciting signals and the related noise performance. Then the four main factors which affect PLL tracking performance are explained in detail. Finally the Costas loop, which is widely utilized in GPS receivers, is introduced.

2.1 Phase-Locked Loop

2.1.1 Basic Principle of Phase-Locked Loop

A PLL is a control loop which synchronizes its output signal (generated by a voltage or numerical controlled oscillator) with a reference or input signal in frequency as well as in phase. In the synchronized –often called locked – state the output frequency of a PLL is exactly same as the input signal and the phase error between the oscillator's output signal and the reference signal is zero, or remains constant. In the unlocked state, the PLL generates a control signal which is related to the phase error. This signal acts on the oscillator in such a way that the phase error is again reduced to a minimum. In such a control mechanism, the PLL always adjusts the phase of the output signal to lock to the phase of the reference signal (Best 1999).

There are four different types of PLLs: LPLL (linear PLL), DPLL (“classical” digital PLL), ADPLL (all-digital PLL) and SPLL (software PLL). The LPLL and SPLL are relevant to this thesis.

A typical PLL block diagram is shown in Figure 2.1. It consists of three basic functional components: a discriminator or a phase detector (PD), a loop filter (LF) and a voltage – controlled oscillator (VCO).

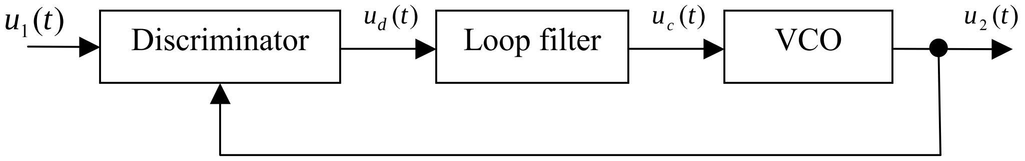

Figure 2.1: A Typical PLL Block Diagram

Usually the discriminator is a multiplier and the loop filter is a low-pass filter. The output of VCO maybe is a sine wave or a square wave. We assume here that the input signal is a sine wave and the output signal is a square wave and it can be written as a Walsh function (Best 1999) as follows

$$u_1(t) = U_1 \sin[\omega_1 t + \theta_1(t)] \tag{2.1}$$

$$u_2(t) = U_2 W[\omega_2 t + \theta_2(t)] \tag{2.2}$$

where $u_1(t)$ and $u_2(t)$ are input and output signals, U_1 and U_2 are input and output signals amplitudes, ω_1 and ω_2 are input and output signals frequencies, $\theta_1(t)$, $\theta_2(t)$ are input and output signals phases.

The Walsh function can be replaced by its Fourier series as follows

$$u_2(t) = U_2\left[\frac{4}{\pi}\cos(\omega_2 t + \theta_2(t)) + \frac{4}{3\pi}\cos(3\omega_2 t + \theta_2(t))\ldots\right] \tag{2.3}$$

For simplicity, we assume here that $\theta_1(t)$ is constant over time. The output of the discriminator is given by

$$u_d(t) = u_1(t)u_2(t) = U_1 U_2 \sin(\omega_1 t + \theta_1)\times\left[\frac{4}{\pi}\cos(\omega_2 t + \theta_2) + \frac{4}{\pi}\cos(3\omega_2 t + \theta_2) + \ldots\right] \tag{2.4}$$

When the LPLL is locked, we can obtain

$$\omega_2 = \omega_1 \tag{2.5}$$

$$u_d(t) = U_1 U_2\left[\frac{2}{\pi}\sin\theta_e + \ldots\right] \tag{2.6}$$

where $\theta_e = \theta_1 - \theta_2$ is the phase difference or phase error and is constant in the locked state. The other terms in $u_d(t)$ have high frequencies and will be filtered out by the loop filter. Neglecting the high frequency terms we can obtain

$$u_d(t) \approx K_d \sin\theta_e \tag{2.7}$$

where $K_d = \dfrac{2U_1U_2}{\pi}$ is called the discriminator gain. When the phase error is small, $u_d(t)$ can be expressed by a linear form as

$$u_d(t) \approx K_d\theta_e \tag{2.8}$$

When $u_d(t)$ passes through the loop filter, the high frequencies are eliminated and the direct current term and low frequency components will pass. For a VCO, its instantaneous frequency $\omega_o(t)$ is given by (Best 1999) as

$$w_o(t) = \omega_2 + K_o u_f(t) \tag{2.9}$$

where K_o is called the VCO gain. The instantaneous phase $\theta_o(t)$ of VCO is the integral of its instantaneous frequency as

$$\theta_o(t) = \int_0^t \omega_o(t)dt = \omega_2 t + K_o \int_0^t u_f(t)dt \tag{2.10}$$

By comparing Equation 2.2 to Equation.2.10, we obtain

$$\theta_2(t) = K_o \int_0^t u_f(t)dt \tag{2.11}$$

The control voltage from the loop filter adjusts the frequency and phase of the VCO to synchronize with the input signal's frequency and phase. In the locked state, the frequency difference and the phase difference between the input signal and the output signal from the VCO is zero. This means that the PLL replicates a signal whose frequency and phase are the exact same as those of the input signal.

2.1.2 Loop Filter

The loop filter is a low-pass filter which passes through the control signals and filters out most noise. The control signals are used to adjust the VCO frequency and phase to provide accurate synchronization with the incoming signal. The loop filter order and noise bandwidth determine the response of the loop to the input signal dynamics. The orders of the commonly used loop filters are one or two. Figures 2.2 and 2.3 show the block diagrams of the first and the second loop filters that are widely used in GPS receivers (Kaplan 1996).

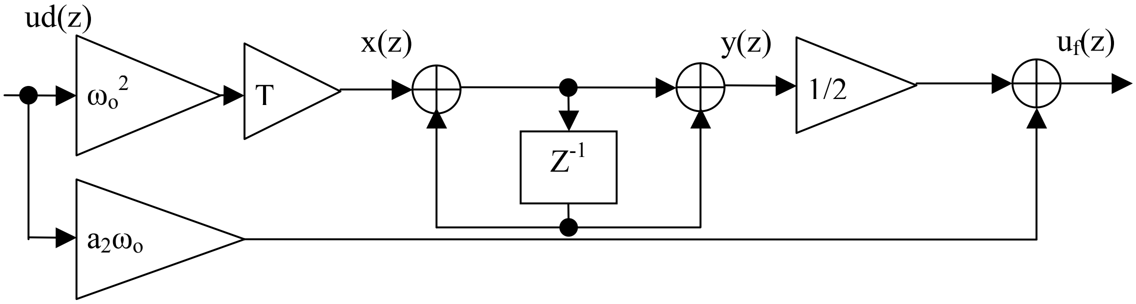

Figure 2.2: Block diagram of a first order loop filter (after Kaplan 1996)

Taking Figure 2.2 into account, the relationship between the input and output of the loop filter is derived as follows

$$x(z) = \omega_o^2 T u_d(z) \tag{2.12}$$

The transfer function from $x(z)$ to $y(z)$ is defined as (Hu 2001)

$$H_{xy}(z) = \frac{y(z)}{x(z)} = \frac{1+z^{-1}}{1-z^{-1}} \tag{2.13}$$

Inserting Equation 2.12 into Equation 2.13 yields

$$y(z) = \frac{1+z^{-1}}{1-z^{-1}} x(z) = \frac{1+z^{-1}}{1-z^{-1}} \omega_o^2 T u_d(z) \tag{2.14}$$

The output of the first order loop filter is described as

$$u_f(z) = \frac{1}{2} y(z) + a_2 \omega_o u_d(z) \tag{2.15}$$

Inserting Equation 2.14 into Equation 2.15 yields

$$u_f(z) = \frac{1}{2} \frac{1+z^{-1}}{1-z^{-1}} \omega_o^2 T u_d(z) + a_2 \omega_o u_d(z) \tag{2.16}$$

Rearranging Equation 2.16 we obtain

$$(1-z^{-1})u_f(z) = (\frac{1}{2}\omega_o^2 T + a_2\omega_o)u_d(z) + (\frac{1}{2}\omega_o^2 T - a_2\omega_o)u_d(z)z^{-1} \tag{2.17}$$

Performing the inverse Z transform to Equation 2.17 yields

$$u_f(n) = u_f(n-1) + (\frac{1}{2}\omega_o^2 T + a_2\omega_o)u_d(n) + (\frac{1}{2}\omega_o^2 T - a_2\omega_o)u_d(n-1) \tag{2.18}$$

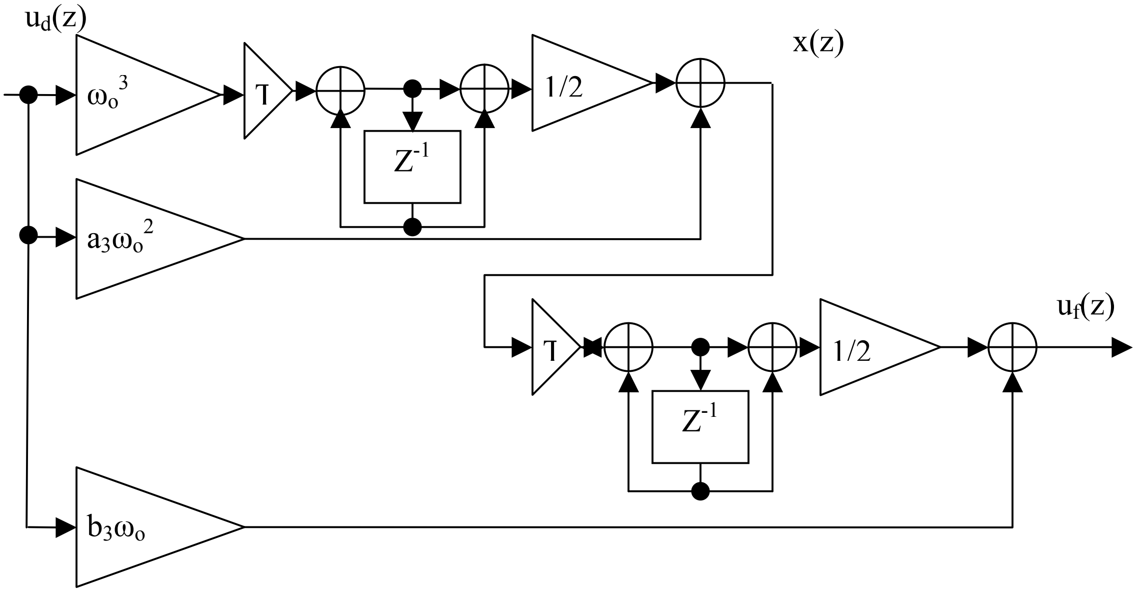

Figure 2.3: Block diagram of a second order loop filter (after Kaplan 1996)

Based on the derivation for the first order loop filter, the relationship between the input and output of the second order loop filter is easier to derive. From Equation 2.16 and Figure 2.3, $x(z)$ can be written as

$$x(z) = \frac{1}{2}\frac{1+z^{-1}}{1-z^{-1}}\omega_o^3 T u_d(z) + a_3\omega_o^2 u_d(z) \tag{2.19}$$

The output of the second order loop filter can be written as

$$u_f(z) = \frac{1}{2}\frac{1+z^{-1}}{1-z^{-1}}Tx(z) + b_3\omega_o u_d(z) \tag{2.20}$$

Inserting Equation 2.19 into 2.20

$$u_f(z) = \frac{1}{2}\frac{1+z^{-1}}{1-z^{-1}}T\left[\frac{1}{2}\frac{1+z^{-1}}{1-z^{-1}}\omega_o^3 T u_d(z) + a_3\omega_o^2 u_d(z)\right] + b_3\omega_o u_d(z) \tag{2.21}$$

Rearranging Equation 2.21 yields

$$\begin{aligned}(1-2z^{-1}+z^{-2})u_f(z) = {}& (\frac{T^2\omega_o^3}{4} + \frac{a_3\omega_o^2 T}{2} + b_3\omega_o)u_d(z) + (\frac{T^2\omega_o^3}{2} - 2b_3\omega_o)z^{-1}u_d(z) \\ & + (\frac{T^2\omega_o^3}{4} - \frac{a_3\omega_o^2 T}{2} + b_3\omega_o)z^{-2}u_d(z)\end{aligned} \tag{2.22}$$

Performing the inverse Z transform of Equation 2.22, the output of the second order loop filter is described as

$$\begin{aligned}u_f(n) = 2u_f(n-1) - u_f(n-2) + {}& (\frac{T^2\omega_o^3}{4} + \frac{a_3\omega_o^2 T}{2} + b_3\omega_o)u_d(n) + (\frac{T^2\omega_o^3}{2} - 2b_3\omega_o)u_d(n-1) \\ & + (\frac{T^2\omega_o^3}{4} - \frac{a_3\omega_o^2 T}{2} + b_3\omega_o)u_d(n-2)\end{aligned} \tag{2.23}$$

In general, the loop filter is a linear time-invariant system and its discrete form can be described by the following difference equation (Hu 2001)

$$u_f(n) = -\sum_{k=1}^{N-1} a_k u_f(n-k) + \sum_{r=0}^{N-1} b_r u_d(n-r) \tag{2.24}$$

where N is the order of the loop.

2.1.3 Voltage-Controlled Oscillator and Numerical or Digital Controlled Oscillator

In a LPLL, one of the three components is a Voltage-Controlled Oscillator (VCO). One important characteristics of a VCO is that it is an oscillator. What makes it different from an ordinary oscillator is that its oscillating frequency is controlled by an external control signal-voltage or sometime current. In an ideal situation, the oscillating frequency of the

VCO changes proportionally with the control voltage and the relationship was described by Equation 2.9.

In a software PLL, the VCO is in a digital form and usually called a NCO (Numerical-Controlled Oscillator) or DCO (Digital-Controlled Oscillator). The block diagram of a NCO is shown in Figure 2.4.

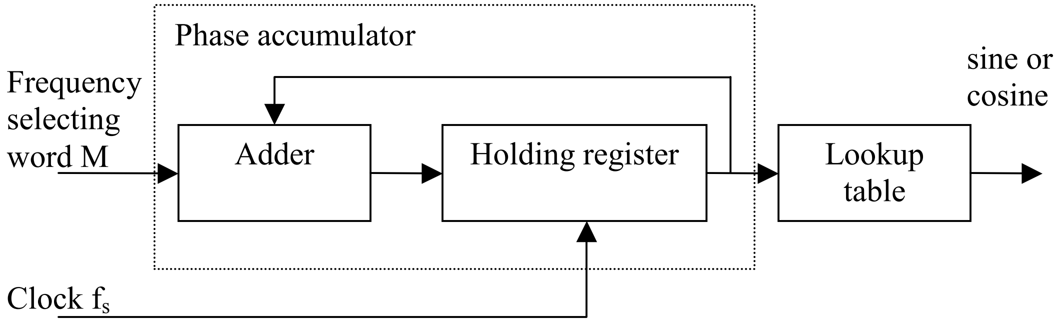

Figure 2.4: Block Diagram of a NCO (Analog devices 2004)

A NCO can be implemented from a phase accumulator which consists of an adder, a holding register and a Lookup table which is stored in a Read-Only-Memory (ROM). In this case, the digital amplitude information that corresponds to a completed cycle of a sine-wave or cosine-wave is stored in the ROM. Every amplitude value corresponds to a certain phase value. As the sine or cosine wave is periodic, the whole signal wave over a certain period time can be formed by reading out from the Look-up table periodically. Every time the clock outputs a pulse, the phase value in the phase accumulator adds a phase step once. This phase value is then used to address the look-up table to look for the corresponding amplitude. The output frequency of a NCO is determined by the phase step, sometime called a frequency selecting word. A higher frequency corresponds to a larger frequency selecting word. The period character of the sine or cosine wave is implemented by means of the carry function of the Phase accumulator which functions as a “phase wheel” (Analog devices 2004) (see Figure 2.5). Each designated point on the phase wheel corresponds to the equivalent point on a cycle of the sine-wave. As the phase accumulator operates the vector rotates around the wheel, the corresponding output sine-wave is read out from the look-up table. One revolution of the vector around the phase wheel at a constant speed, results in one complete cycle of the output sine-wave. The number of the discrete phase points contained in the wheel is determined by the resolution of the phase accumulator. For a K bit length phase accumulator, this number is 2^K.

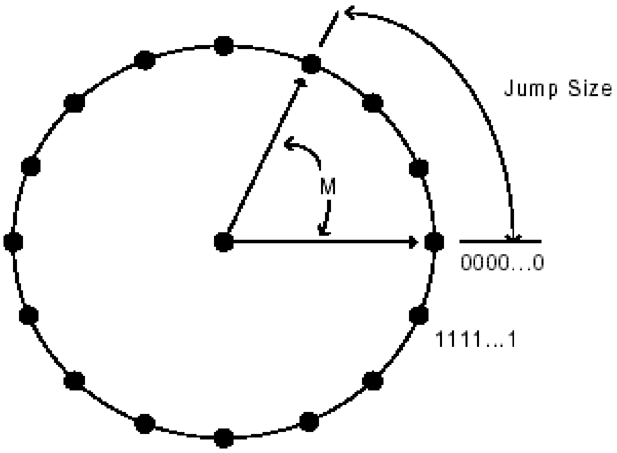

Figure 2.5: Carry Function of the Phase Accumulator (from Analog devices 2004)

A phase-to-amplitude lookup table is used to convert the instantaneous output value (sometimes a truncated value) of the phase accumulator into the sine-wave amplitude information that is the output of the NCO. The amplitude information of the lowest frequency $\frac{f_s}{2^K}$ is stored in ROM. The output frequency of the NCO is determined by the jump size M which corresponds to the phase increment for each clock period. The higher the output frequency is, the larger the jump size and the smaller the number of discrete points during one cycle. According to the sampling theory, the highest output frequency of the NCO is determined by the clock frequency. It is equal to half of the clock frequency. The lowest output frequency equals to $\frac{f_s}{2^K}$. In order to output the lowest frequency, all amplitude information stored in ROM is read out cyclically. By contrast, with the lowest frequency, the highest frequency only reads out twice every cycle (Analog devices 2004).

Many NCOs exploit the symmetrical nature of a sine-wave or cosine-wave and utilizes mapping logic to synthesize a complete sine-wave cycle to save the space on ROM. In this case, only ¼ of the cycle data is required to be stored in the look-up table. The phase-to-amplitude lookup table generates all the necessary data by reading forward, then back through the lookup table.

The output frequency of the NCO is given by (Analog devices 2004)

$$f_o = \frac{Mf_s}{2^K} \tag{2.25}$$

where M is called the frequency selecting word, K is the length of the phase accumulator, f_s is the clock frequency and f_o is the output frequency of the NCO. The resolution of the output frequency is $\frac{f_s}{2^K}$.

2.1.4 Phase Model and Key Parameters of a PLL

During the tracking period, the phase difference is zero or small. The PLL can be regarded as a linear system. In order to analyze the tracking performance of the PLL, a linear phase model is required. For the discriminator, it is linear when the phase difference is small and is described by Equation 2.8. Besides, all loop filters are linear time-invariant systems and their transfer function in the Z domain is described as (Legrand & Macabiau 2001)

$$F(z) = \frac{\sum_{n=0}^{N-1} b_n z^{-n}}{(1-z^{-1})^{N-1}} \tag{2.26}$$

The output phase of the NCO in the discrete domain can be derived from Equation 2.11

$$\theta_2(n) = K_o \sum_{k=0}^{n-1} u_f(k) = \theta_2(n-1) + K_o u_f(n-1) \tag{2.27}$$

Performing the Z transform for Equation 2.27, we can derive the model of NCO in the Z domain as

$$H_N(z) = \frac{\theta_2(z)}{U_f(z)} = \frac{K_o z^{-1}}{1-z^{-1}} \tag{2.28}$$

The simplified linear phase model of a digital PLL is shown in Figure 2.6

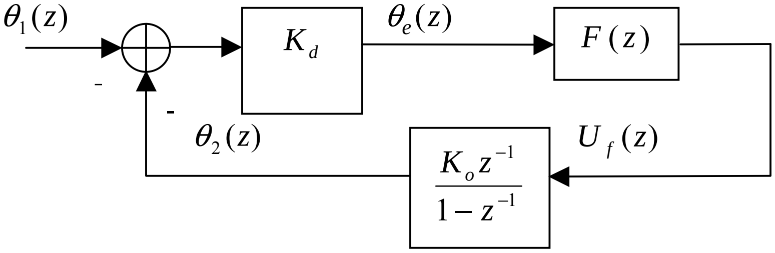

Figure 2.6: Linear phase model of a digital PLL

The transfer function of the loop is defined by (Best 1999) as follows

$$H(z) = \frac{\theta_2(z)}{\theta_1(z)} = \frac{KF(z)z^{-1}}{(1-z^{-1}) + KF(z)z^{-1}} \tag{2.29}$$

Inserting Equation 2.26 into Equation 2.29 we obtain

$$H(z) = \frac{K\sum_{n=0}^{N-1} b_n z^{-n-1}}{(1-z^{-1})^N + K\sum_{n=0}^{N-1} b_n z^{-n-1}} \tag{2.30}$$

where $K = K_o K_d$ is called as the loop gain

The number of the poles in the transfer function is defined as the order of the loop.

The error transfer function of the loop is defined by (Best 1999) as follows

$$H_e(z) = \frac{\theta_e(z)}{\theta_1(z)} = \frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K\sum_{n=0}^{N-1} b_n z^{-n-1}} \tag{2.31}$$

A set of key parameters which govern the dynamic performance of the PLL (Best 1999) are as follows:

- The lock range Δw_L. This is the frequency range within which a PLL locks within one single-beat note between the reference frequency and the output frequency. Normally the operating-frequency range of a PLL is restricted to the lock range.
- The pull-out range Δw_{PO}. This is the dynamic limit for stable operation of a PLL. If tracking is lost within this range, a PLL normally will lock again, but this process can be slow if it is a pull-in process.

- The pull-in range Δw_P. This is the range within which a PLL will always become locked, but the process can be rather slow.
- The hold range Δw_H. This is the frequency in which a PLL can statically maintain phase tracking. A PLL is conditionally stable only within this range.

In many designs, the relationship between these parameters is set by the following inequality:

$$\Delta w_L < \Delta w_{PO} < \Delta w_P < \Delta w_H \,. \tag{2.32}$$

2.1.5 PLL Responses to Different Excitation Signals

Let us assume that the PLL is locked at the initial moment. The tracking performance of the PLL responds to three important excitation signals which are calculated and analyzed with attention given to the steady-state response. The excitation signals are phase step, frequency step and frequency ramp.

2.1.5.1 A Phase Step Excitation Signal

If there is a phase step in the input signal at time $t = 0$, the digitized input phase $\theta_1(n)$ is a step function and can be given by

$$\theta_1(n) = u(n)\Delta\phi \tag{2.33}$$

where $u(n)$ is a unit step function and $\Delta\phi$ is the size of the phase step. The Z transform of $\theta_1(n)$ is given by

$$\theta_1(z) = \frac{z}{z-1}\Delta\phi$$

(2.34)

From Equation 2.31, the phase difference is given by

$$\theta_e(z) = H_e(z)\theta_1(z) = \frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K_o K_d \sum_{n=0}^{N-1} b_n z^{-n-1}}\theta_1(z) \tag{2.35}$$

Inserting Equation 2.34 into Equation 2.35 and using the Final Value Theorem, the steady-state phase difference is given by

$$\theta_{e\infty} = \lim_{z\to 1}(z-1)\theta_e(z) = \lim_{z\to 1}(z-1)\frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K_o K_d \sum_{n=0}^{N-1} b_n z^{-n-1}}\frac{z}{z-1}\Delta\phi \tag{2.36}$$

Analyzing Equation 2.36, it is found that

$$\theta_{e\infty} = 0 \qquad for\ N \geq 1 \tag{2.37}$$

Equation 2.37 leads to the conclusion that any loop can track a phase step excitation signal with no phase difference.

2.1.5.2 A Frequency Step Excitation Signal

If there is a frequency step $\Delta\omega$ in the input signal, the digitized input phase $\theta_1(n)$ is given by

$$\theta_1(n) = \Delta\omega n u(n) \tag{2.38}$$

Performing a Z transform to Equation 2.38, we obtain

$$\theta_1(z) = \Delta\omega\frac{z}{(z-1)^2} \tag{2.39}$$

Similarly, using the Final Value Theorem the steady-state phase difference is given by

$$\theta_{e\infty} = \lim_{z\to 1}(z-1)\theta_e(z) = \lim_{z\to 1}(z-1)\frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K_o K_d \sum_{n=0}^{N-1} b_n z^{-n-1}}\Delta\omega\frac{z}{(z-1)^2} \qquad (2.40)$$

Analyzing Equation 2.40 (derivation is in Appendix A), the following conclusion is obtained

$$\theta_{e\infty} = \begin{cases} \dfrac{\Delta\omega}{K_o b_0} & N = 1 \\ 0 & N \geq 2 \end{cases} \qquad (2.41)$$

Equation 2.41 means that the first order loop (no loop filter) can track a frequency step excitation signal but with a constant phase difference and other loops whose orders are larger than one can track a frequency step excitation signal with no phase difference.

2.1.5.3 A Frequency Ramp Excitation Signal

If there is a frequency ramp $\Delta\omega t$ in the input signal, the digitized input phase $\theta_1(n)$ is given by

$$\theta_1(n) = \frac{1}{2}\Delta\omega n^2 u(n) \qquad (2.42)$$

Performing the Z transform on Equation 2.42 we obtain

$$\theta_1(z) = \frac{\Delta\omega}{2}\frac{z(z+1)}{(z-1)^3} \qquad (2.43)$$

Again, using the Final Value Theorem, the steady-state phase difference is given by

$$\theta_{e\infty} = \lim_{z\to 1}(z-1)\theta_e(z) = \lim_{z\to 1}(z-1)\frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K_o K_d \sum_{n=0}^{N-1} b_n z^{-n-1}}\frac{\Delta\omega}{2}\frac{z(z+1)}{(z-1)^3} \qquad (2.44)$$

Analyzing Equation 2.44 (derivation is in Appendix A), the following conclusion is obtained

$$\theta_{e\infty} = \begin{cases} \infty & N = 1 \\ \dfrac{\Delta\omega}{K_o(b_0 + b_1)} & N = 2 \\ 0 & N > 2 \end{cases} \tag{2.45}$$

Equation 2.45 means that the first order loop (no loop filter) cannot track a frequency ramp excitation signal; the second order loop can track but with a constant phase difference and other loops whose orders are larger than two can track a frequency ramp excitation signal with no phase difference.

Summarizing the above discussion of three excitation signals, the conclusion is that the Nth order loop can track the Nth order variation in the input phase with a constant phase difference and can track less than the Nth order variation in the input phase with no phase difference but can not track higher than the Nth order variation in the input phase.

It is obvious from the above derivation that the higher order variations is in the input signal, the higher order of the loop is needed to track it.

In addition, the following three conditions are necessary for a PLL system to maintain phase tracking (Best 1999)

- The frequency variation in the input signal must be within the hold range.
- The maximum tolerable frequency step must be smaller than the pull-out range.

- The rate of change of the reference frequency is limited by the loop natural frequency.

Note that the discussion in this section is based on the assumption that the initial state of the PLL is locked. If the PLL is unlocked initially, the above analysis is not suitable due to the uncertain phase error. In most cases, the phase error is so large that it is unreasonable to use the linear model. Instead of the tracking process, an acquisition process with a nonlinear model can be used to analyze the PLL response.

2.1.6 Noise in the PLL

It is hard to derive the exact solutions for noise performance due to its randomness. However, some simulated and experimental results can be used to discuss how the noise affects the PLL. We assume that all noise signals discussed here are white.

Supposing that white noise with power P_n is superimposed on the input signal of the PLL with power P_s, the bandwidth of the noise spectrum is limited to B_i by a pre-filter. If the input signal is a sine wave, with the effect of noise, the zero crossings of the resulting signal vary back or forward depending on the instantaneous polarity of the noise signal. This so-called phase jitter or phase noise is designated by $\theta_{n1}(t)$ (Best 1999). The signal-to-noise ratio at the input of the PLL is defined as

$$(SNR)_i = \frac{P_S}{P_n} \tag{2.46}$$

The square of the RMS phase noise is given by (Best 1999)

$$\overline{\theta_{n1}^2} = \frac{P_n}{2P_S} = \frac{1}{2(SNR)_i} \tag{2.47}$$

That is, the square of the RMS value of the phase jitter is inversely proportional to the SNR at the input of the PLL. The square of the spectral density of the phase jitter $\overline{\theta_{n1}(jw)}$ can be expressed as (Best 1999)

$$\overline{\theta_{n1}^2(jw)} = \frac{\overline{\theta_{n1}^2}}{B_i/2} \tag{2.48}$$

where B_i is the bandwidth of the pre-filter

For a linear system, the spectral density of the output signal is equal to the product of the spectral density of the input signal with the transfer function of the system. Based on this property, the spectral density of the phase noise at the output can be calculated using

$$\sqrt{\overline{\theta_{n2}^2(j\omega)}} = \left|H(j\omega)\right|\sqrt{\overline{\theta_{n1}^2(j\omega)}} \tag{2.49}$$

Integrating Equation 2.48 in the frequency domain, the RMS value of the phase noise at the output of the PLL is obtained as (Best 1999)

$$\overline{\theta_{n2}^2} = \int_0^\infty \overline{\theta_{n2}^2(j2\pi f)}df = \frac{\overline{\theta_{n1}^2}}{B_i/2}\int_0^\infty \left|H(j2\pi f)\right|^2 df \tag{2.50}$$

The integral $\int_0^\infty \left|H(j2\pi f)\right|^2 df$ is defined as the noise bandwidth B$_L$ (Best 1999)

$$B_L = \int_0^\infty \left|H(j2\pi f)\right|^2 df \tag{2.51}$$

For a second order loop, the solution of this integral is (Best 1999)

$$B_L = \frac{\omega_n}{2}(\zeta + \frac{1}{4\zeta}) \tag{2.52}$$

where ς is the damping factor and w_n is the natural frequency.

Considering the transient response and noise performance, ζ = 0.7, $B_L = 0.53\omega_n$ are chosen for most applications.

Substituting Equation 2.47 and 2.51 into 2.50, the output phase noise $\overline{\theta_{n2}^2}$ can be rewritten as (Best 1999)

$$\overline{\theta_{n2}^2} = \frac{\overline{\theta_{n1}^2}}{B_i/2} \times B_L = \frac{P_n}{P_s} \times \frac{B_L}{B_i} \tag{2.53}$$

By analogy to Equation 2.47, we can define a signal-to-noise ratio $(SNR)_L$ at the output of the loop (Best 1999)

$$\overline{\theta_{n2}^2} = \frac{1}{2(SNR)_L} \tag{2.54}$$

Comparing Equation 2.53 and Equation 2.54, we can write (Best 1999)

$$(SNR)_L = \frac{P_s}{P_n} \times \frac{B_i}{2B_L} = (SNR)_i \frac{B_i}{2B_L} \tag{2.55}$$

For most applications, B_i is much narrower than B_L. Equation 2.55 shows that the PLL has the advantage of improving the SNR of the input signal. The narrower the noise bandwidth, the greater the improvement the PLL can provide.

One issue in the PLL design is how large $(SNR)_i$ must be to enable the safe acquisition of the PLL. Experimental results (Best 1999) with second order loops have demonstrated the following:

1. For $(SNR)_L = 1\ (0dB)$, a lock-in process will not occur because the output phase noise is excessive;
2. At $(SNR)_L = 2\ (3dB)$, lock-in is eventually possible; and
3. For $(SNR)_L = 4\ (6dB)$, stable operation is generally possible.

For example, in the case of a second–order loop with $B_i = 2MHz$ and $B_L = 18Hz$, when $(SNR)_i > -41.4dB$, a stable operation is generally possible.

Another issue in the PLL design is how often on the average a PLL will temporarily unlock. Defining T_{av} as the average time interval between two lockouts, for second-order loops, T_{av} has been found to be a function of $(SNR)_L$ according to experimental results and it gets longer as $(SNR)_L$ increases (Best 1999).

2.2 PLL Tracking Loop Measurement Errors

In precise positioning applications, carrier phase measurements are used to estimate navigation related solutions. Therefore accurate Doppler frequency and phase measurements play an important role in determining the positioning precision. The PLL tracking loop measurement errors have a significant effect on the tracking performance. This section discusses four dominant error sources that affect PLL phase errors: thermal noise, dynamic stress, vibration and Allan deviation. The last two errors are related to the clock used in the receiver.

2.2.1 Thermal Noise

Thermal noise is the most common present error source in a PLL. The PLL phase jitter caused by thermal noise is computed as follows (Kaplan 1996):

$$\sigma_t = \frac{360}{2\pi}\sqrt{\frac{B_n}{C/N_0}\left(1+\frac{1}{2TC/N_0}\right)} \tag{2.56}$$

where σ_t is the 1-sigma thermal noise, B_n is the carrier loop noise bandwidth (Hz), C/N_O is the carrier-to-noise ratio and T is the pre-integration time (s).

In order to mitigate the thermal noise error, a narrow bandwidth, high carrier over noise ratios and longer pre-integration time are required.

2.2.2 Dynamic Stress

The dynamic stress error is closely related to the order of the loop. For a second order loop, the dynamic stress error is defined as (Kaplan 1996)

$$\theta_{e2} = 0.2809\frac{\frac{dR^2}{dt^2}}{B_n^2} \tag{2.57}$$

where θ_{e2} is the 1-sigma dynamic stress error (deg), and $\frac{dR^2}{dt^2}$ is the acceleration along the line of sight (deg/sec^2)

For a third order loop, the dynamic stress error is defined as (Kaplan 1996)

$$\theta_{e3} = 0.4828 \frac{\frac{dR^3}{dt^3}}{B_n^3} \tag{2.58}$$

where $\frac{dR^3}{dt^3}$ is the jerk along the line of sight (deg/sec^3)

The dynamic stress is strictly dependent on signal dynamics and the bandwidth of the loop for a given loop order. High dynamics produce high dynamic stress. However, increasing the bandwidth will effectively decrease the dynamic stress error especially for the higher order loop.

2.2.3 Vibration

Vibration induces clock phase noise which can be computed as follows (Kaplan 1996)

$$\sigma_v = \frac{360 f_L}{2\pi} \sqrt{\int_{f_{\min}}^{f_{\max}} S_v^2(f_m) \frac{P(f_m)}{f_m^2} df_m} \tag{2.59}$$

where σ_v is the 1-sigma vibration induced clock phase noise, f_L is the L-band input frequency (Hz), $S_v^2(f_m)$ is the oscillator vibration sensitivity of $\Delta f / f_L$ per G as a function of f_m, f_m is the random vibration modulation frequency (Hz) and $P(f_m)$ is the power curve of the random vibration as a function of f_m (G^2 / Hz).

Vibration induced clock phase noise has no relationship with the loop order and the bandwidth. It is mainly determined by the vibration environment.

2.2.4 Allan Deviation

Allan deviation also induces clock phase noise. The equation for computing the Allan deviation-induced phase noise for a second order PLL is (Kaplan 1996)

$$\theta_{A2} = 144\frac{\sigma_A(\tau) f_L}{B_n} \tag{2.60}$$

where θ_{A2} is the Allan deviation-induced jitter (deg), τ is the short-term stability gate time for the Allan variance measurement (s) and $\sigma_A(\tau)$ is the Allan deviation.

The equation for computing short-term Allan deviation induced phase noise for a third order PLL is (Kaplan 1996)

$$\theta_{A3} = 160\frac{\sigma_A(\tau) f_L}{B_n} \tag{2.61}$$

Under the situation of a narrow bandwidth and poor clock quality, the Allan deviation effect dominates the PLL tracking error. Clock quality is the key element in decreasing this error (Kaplan 1996).

2.2.5 Total PLL Tracking Loop Measurements Errors and Thresholds

The 3-sigma value of the total PLL tracking loop measurement errors can be written as (Kaplan, 1996)

$$3\sigma_{PLL} = 3\sqrt{\sigma_t^2 + \sigma_v^2 + \theta_A^2} + \theta_e \tag{2.62}$$

where $3\sigma_{PLL}$ is the 3-sigma total tracking error.

Equation 2.62 indicates that the dynamic stress error is a 3-sigma effect and is additive to the phase jitter. According to the rule-of-thumb, the PLL tracking threshold is computed as (Kaplan 1996)

$$\sqrt{\sigma_t^2 + \sigma_v^2 + \theta_A^2} + \frac{\theta_e}{3} \le 15^\circ \tag{2.63}$$

Equation 2.63 means that the 1-sigma value of the total errors must be less than 15 degrees. Otherwise the PLL lock state can not be guaranteed.

2.3 COSTAS Loop

One prominent use of the PLL is its carrier recovery or extraction from phase-coded or phase shift keying modulated signals. Among these signals, the BPSK (binary phase shift keying) modulated signal is the typical one which is acquired by changing the carrier phase according to the modulating data bits. Data bit "0" corresponds to "180°" or "0°" phase change in the carrier signal and "1" to "0°" or "180°" degree change. Recovering the carrier frequency from this modulated signal is difficult because a randomly modulated BPSK signal has no discrete energy line at the carrier frequency. In order to extract the carrier frequency from the BPSK modulated signal, some special strategy is needed to eliminate the effect of the data bit modulation. One well-known method is to use a COSTAS loop.

Figure 2.7 shows a carrier-recovery loop for BPSK signals, called a COSTAS loop (Egan 1998). Both the output of the VCO and its 90° phase shift signal multiplies with the input signal. The rationale of COSTAS is explained as follows.

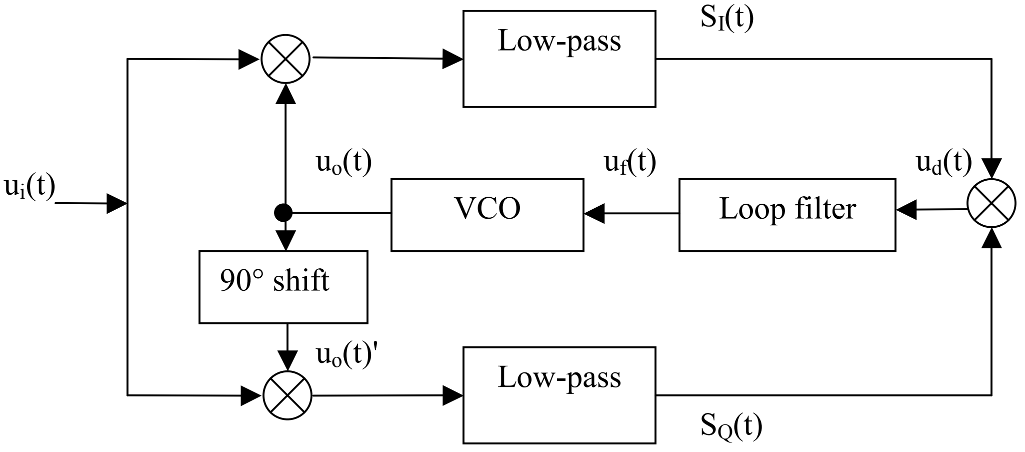

Figure 2.7: Block Diagram of COSTAS Loop

Assuming the input BPSK signal is given by

$$u_i(t) = U_i \sin(\omega_1 t + \varphi + \theta_1(t)) \tag{2.64}$$

where U_i is the amplitude of the input signal and w_1 and $\theta_1(t)$ are input carrier frequency and phase and

$$\varphi = \begin{cases} 180^0 & when \text{ mod} ulating \ signal = 0 \\ 0^0 & when \text{ mod} ulating \ signal = 1 \end{cases}$$

Assuming the output signal is given by

$$u_o(t) = U_o \sin(\omega_2 t + \theta_2(t)) \tag{2.65}$$

$$u_o(t)' = U_o \cos(\omega_2 t + \theta_2(t))$$

where U_o is the amplitude of the NCO output signal and w_2 and $\theta_2(t)$ are frequency and phase of the NCO output signal. (2.66)

The output of the first multiplier (neglecting the high frequency component) is

$$S_I(t) = u_i(t)u_o(t) = \frac{U_iU_o}{2}\cos\left(\left(\omega_1 - \omega_2\right)t + \varphi + \theta_1(t) - \theta_2(t)\right) \tag{2.67}$$

$$S_Q(t) = u_i(t)u_o(t)' = \frac{U_iU_o}{2}\sin\left(\left(\omega_1 - \omega_2\right)t + \varphi + \theta_1(t) - \theta_2(t)\right) \tag{2.68}$$

At the output of the discriminator we obtain

$$u_d(t) = S_I(t)S_Q(t) = \frac{U_i^2U_o^2}{8}\sin\left(2\left(\omega_1 - \omega_2\right)t + 2\varphi + 2(\theta_1(t) - \theta_2(t))\right) \tag{2.69}$$

For BPSK signals, 2φ = 0 or 2π. Because the sine function's period is 2π, the above Equation can be rewritten as

$$u_d(t) = S_I(t)S_Q(t) = \frac{U_i^2U_o^2}{8}\sin\left(2\left(\omega_1 - \omega_2\right)t + 2(\theta_1(t) - \theta_2(t))\right) \tag{2.70}$$

From the above derivation, it is obvious that after the discriminator, the effect of the modulation has been removed. That is why a COSTAS loop is insensitive to a 180° phase change in the input signals. Because of this advantage, a COSTAS loop is widely used to recover the carrier frequency from the PSK signals.

Another advantage of the COSTAS loop is that it provides data detection simultaneously with carrier recovery from the output of the low-pass filter. Under a locked state,

$\omega_1 = \omega_2$,

$\theta_1(t) - \theta_2(t) = 0$,

we can obtain

$$\begin{aligned} S_I(t) &= u_i(t)u_o(t) \\ &= \frac{U_iU_o}{2}\cos\varphi \end{aligned} \tag{2.71}$$

Assuming $U_I = \frac{U_i U_o}{2}$

$$S_I(t) = U_I \cos\varphi = \begin{cases} U_I & \text{mod}\textit{ulating data is } 1 \\ -U_I & \text{mod}\textit{ulating data is } 0 \end{cases} \tag{2.72}$$

From Equation 2.72, the modulated data can be detected by judging the output of the low-pass filter $S_I(t)$.

CHAPTER THREE: GPS SOFTWARE RECEIVER REVIEW

The tremendous increase of microprocessor computational power is leading to the proliferation of software radio techniques in which signals are digitized and hence processed entirely in the digital domain. The advantages offered by a software receiver such as flexibility, reconfigurability and high fidelity make its application highly suitable in the GPS field. In addition, GPS software receivers enable the verification of many signal processing algorithms and a variety of advanced designs, thereby providing a very convenient platform for a wide range of research, development and testing. In this chapter, the GPS software receiver structure is presented first; then, the software PLL algorithm used in this thesis is introduced.

3.1 GPS Software Receiver Structure

A typical GPS software receiver consists of the hardware front end and a signal processing unit in software. The front end contains a LNA (low noise amplifier), two stages of down convertors as well as a signal buffer. The digitized intermediate frequency is fed into the signal processing unit which usually is implemented totally in software, even though sometimes FPGA (field programmable gate arrays) is deployed to accomplish the function of the correlators to speed up the calculation. A typical GPS software receiver block diagram is shown in Figure 3.1.

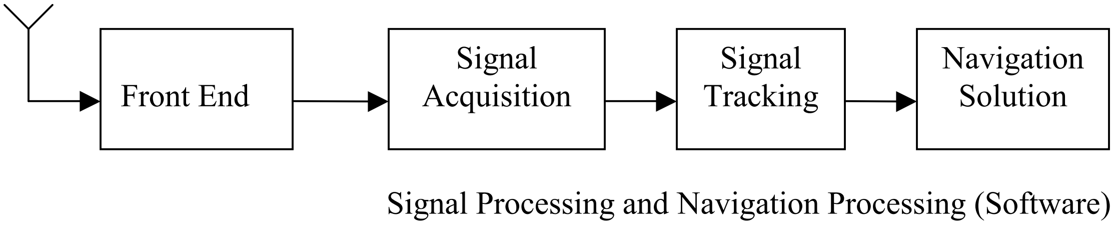

Figure 3.1: Generic GPS software receiver block diagram

3.1.1 Signal Acquisition

Signal acquisition is the initial process, prior to signal processing. It consists of a two-dimensional search in code phase and frequency. The receiver replicates the C/A code and the rough Doppler frequency for each satellite. After Doppler frequency removal, (and, strictly speaking, the carrier frequency thereof) a correlation process is implemented that attempts to align the replica code with the one in the incoming signal. At the conclusion of the signal acquisition part, the rough code phase and Doppler frequency for each satellite should be known (Accord 2004).

Signal acquisition should be achieved in the shortest possible time for all visible satellites. The almanac data, approximate time and the local position can aid the acquisition process and significantly increase its speed. Fast acquisition reduces the time to first fix (TTFF) and therefore improves the overall quality of the receiver.

3.1.2 Signal Tracking

The signal tracking process involves code wipe off and carrier wipe off. The code wipe off utilizes a DLL (delay lock loop) to synthesize the replica code phase with the one of the incoming signal. The carrier wipe off is accomplished by a carrier tracking loop to synthesize the replica carrier frequency and phase with the one of the incoming signals (Accord 2004).

The carrier tracking loop estimates the signal frequency and phase using an error feedback mechanism. The common carrier tracking loop is the FLL which can only track the signal frequency, or the PLL-Phase-locked loop which cannot only track the signal frequency but also the signal phase and therefore provide an accurate carrier phase measurement. A typical tracking loop block diagram is shown in Figure 3.2 (Ray 2003).

First, the digital IF is stripped off the carrier in the in-phase and quadra-phase mixers by the replica carrier from the carrier loop NCO and generates the I and Q signals. The amplitude of the signal can be determined from the vector sum of the I and Q components and the phase angle can be determined from the arctangent of Q/I. The objective of the carrier tracking loop is to track the phase of the incoming signal with no error. Otherwise the carrier tracking loop will detect the phase difference between the incoming signal and the replica signal and adjust the replica frequency and phase to reach alignment.

The carrier phase measurement is derived from the Doppler frequency and phase of the NCO (Accord 2004).

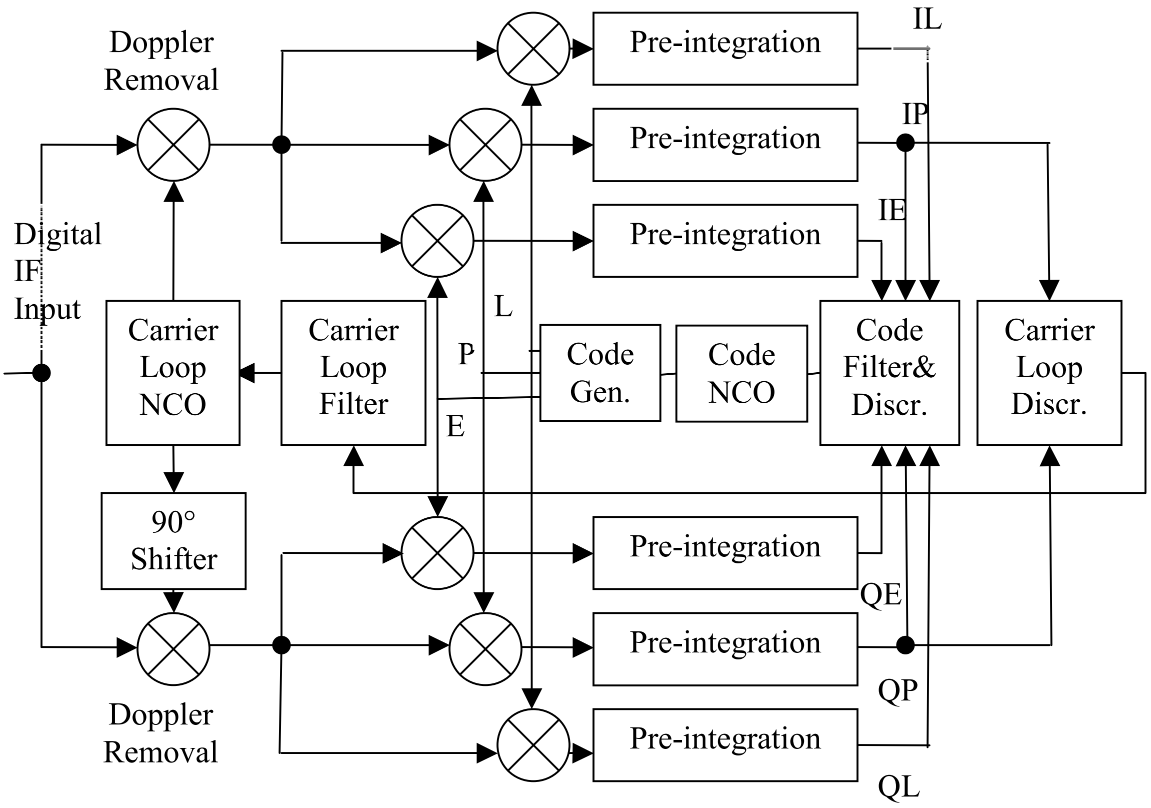

Figure 3.2: Signal tracking block diagram (Accord 2004)

The I and Q signals are then correlated with early, prompt, and late replica codes from the code generator. The phase shift between the early, prompt and late replica codes are usually 1/2 chips apart (In order to mitigate multipath effects, a smaller phase shift is used). The prompt replica code tries to align with the incoming code. The maximum correlation is obtained if alignment is achieved. In this case, the early and late correlators will half the maximum correlation. Any misalignment will produce a difference in the magnitudes of the early and late correlators which can be detected and corrected by the code tracking loop (DLL). The phase and adjusted chips at the alignment state are employed to calculate the pseudorange measurement.

Following the correlators are pre-integration units that increase the signal power. However, the pre-integration time is determined based on the signal power and user dynamics, etc. When the signal is weak, a longer pre-integration period is required to accumulate high signal power. Nevertheless, high signal dynamics applications present an opposite requirement. A compromised design is necessary in order to meet these two requirements simultaneously.

3.1.3 Navigation Solution

A navigation algorithm combines raw measurements from the signal processing with the GPS satellite orbit data to estimate position related parameters. Fundamentally, a navigation solution is an estimate of the user's position and any other required parameters. A GPS estimator is used to estimate the required parameters. The typical estimated states of a GPS receiver are three position components, the receiver clock offset and the clock drift. The velocity is often added in dynamic applications.

3.2 Software PLL (SPLL)

With advancements in microprocessors and digital signal processing, innovative software radio technology emerges. Nowadays, a lot of hardware may be replaced by software methods, providing increased flexibility and ease of reconfiguration that make software more competitive. Along with this trend, implementation of the PLL in software is used in many systems. As long as the speed of the microprocessors or the DSP (digital signal processing) is fast enough, the PLL algorithms can be realized in real time. Even though all types of PLL can be implemented in software, this thesis will focus on an LPLL-like software implementation due to its straightforward realization. Based on the knowledge of the underlying PLL components in digital forms, the PLL algorithm is developed in software as shown in Figure 3.3. The software receiver used to test the LPLL implementations studied herein is GNSS_SoftRx™ developed by the PLAN group (Ma et al 2004).

In the algorithms implemented herein, the first step is to initialize the NCO. The next step is the implementation of the discriminator and loop filter functions. The control signal from the output of the loop filter changes the frequency of the NCO. According to the frequency change, the phase step is adjusted and consequently the phase index is updated to output the signal with a new frequency. The last step is to save the current values of the outputs of the discriminator and the loop filter which will be used for the next iteration.

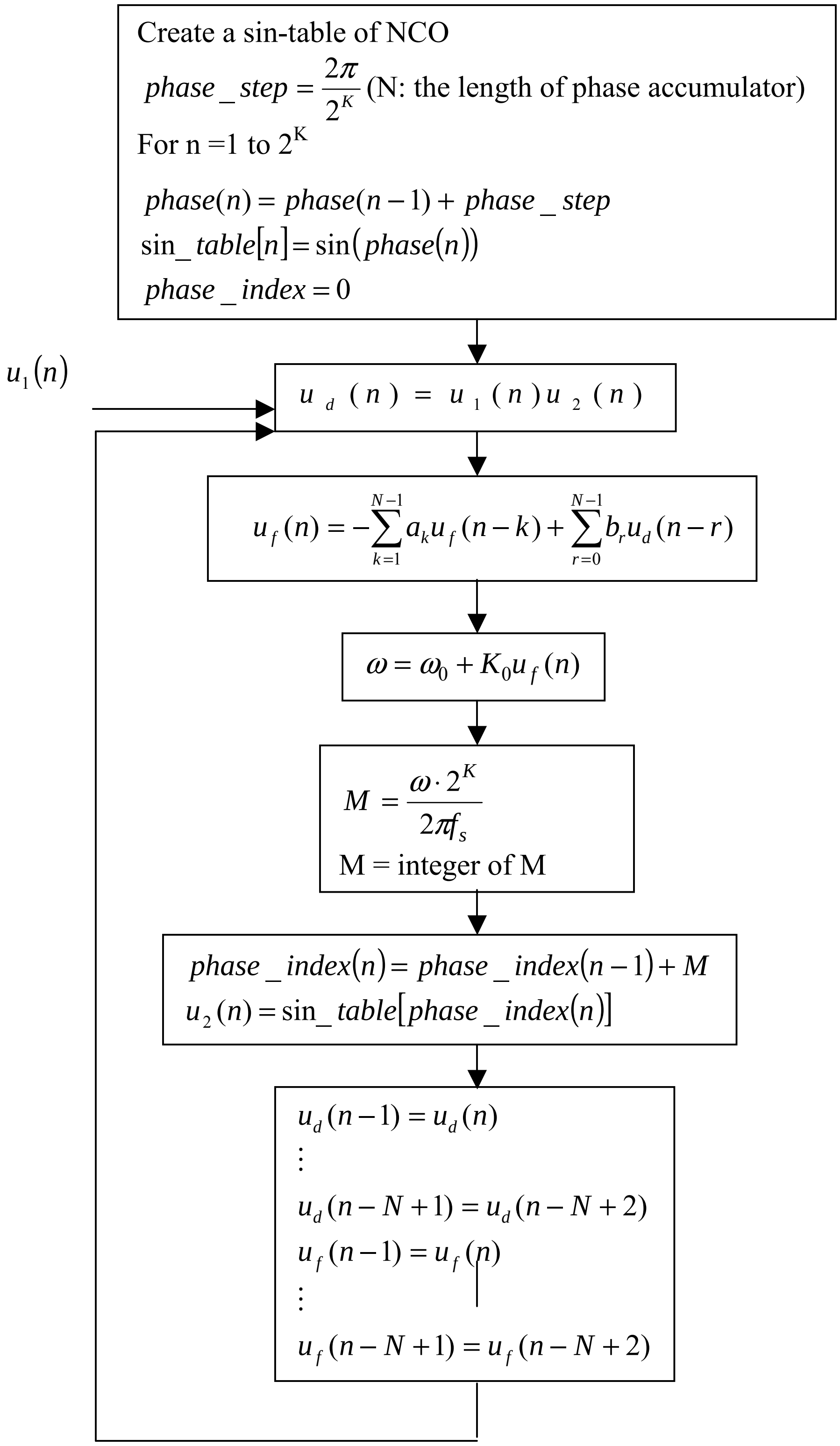
Create a sin-table of NCO
$phase\_step = \frac{2\pi}{2^K}$ (N: the length of phase accumulator)
For n =1 to 2^K
$phase(n) = phase(n-1) + phase\_step$
$\sin\_table[n] = \sin(phase(n))$
$phase\_index = 0$
$u_1(n)$
$u_d(n) = u_1(n)u_2(n)$
$u_f(n) = -\sum_{k=1}^{N-1} a_k u_f(n-k) + \sum_{r=0}^{N-1} b_r u_d(n-r)$
$\omega = \omega_0 + K_0 u_f(n)$
$M = \frac{\omega \cdot 2^K}{2\pi f_s}$
M = integer of M
$phase\_index(n) = phase\_index(n-1) + M$
$u_2(n) = \sin\_table[phase\_index(n)]$
$u_d(n-1) = u_d(n)$
$\vdots$
$u_d(n-N+1) = u_d(n-N+2)$
$u_f(n-1) = u_f(n)$
$\vdots$
$u_f(n-N+1) = u_f(n-N+2)$

Figure 3.3: PLL Algorithm Implementation in Software

CHAPTER FOUR: KALMAN FILTER BASED TRACKING ALGORITHM

This chapter first briefly reviews the well-known Kalman filter algorithm. It includes two adaptive Kalman filter algorithms. Then the design scheme using the Kalman filter in the tracking loop is presented. This is followed by a detailed discussion about how to perform a Kalman filter-based tracking algorithm in the PLL.

4.1 Kalman Filter Review

4.1.1 Kalman Filter Algorithm

The Kalman filter is essentially a recursive algorithm that implements a predictor-corrector type estimator. The predictor is based on a system model and the corrector is based on the measurement model. The Kalman filter is optimal in the sense that it minimizes the estimated error covariance. Let us consider any linear system which can be described using the following state model

$$\dot{x}(t) = Fx(t) + Bu(t) \tag{4.1}$$

where x is the state vector, F is the dynamic matrix, B is input matrix and u is the vector of input white noise. The estimation is based on the measurements which can be modeled as

$$z(t) = Hx(t) + v(t) \tag{4.2}$$

where z is the measurement vector, H is the measurement matrix and v is the vector of the measurement noise.

For a discrete Kalman filter, the system model and measurement model can be expressed as (Salychev 1998)

$$x_k = \Phi_{k,k-1} x_{k-1} + G_{k,k-1} w_{k-1} \tag{4.3}$$

$$z_k = H_k x_k + v_k \tag{4.4}$$

where $\Phi_{k,k-1}$ is called the transition matrix and w_k and v_k are white noise sequences with the following properties:

$$E(w_k) = E(v_k) = 0 \tag{4.5}$$

$$E(w_k w_l^T) = \begin{cases} Q & if\ k = l \\ 0 & other \end{cases} \tag{4.6}$$

$$E(v_k v_l^T) = \begin{cases} R & if\ k = l \\ 0 & other \end{cases} \tag{4.7}$$

The estimation of x is derived using the system model and the measurement model from the condition (Salychev 1998):

$$trace\ E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = trace\ E(\tilde{x}_k \tilde{x}_k^T) = \text{minimum} \tag{4.8}$$

where $\hat{x}_k$ is the estimate of x_k and $\tilde{x}_k$ is the estimation error. The solution of the above Equation is described by the algorithm of the discrete Kalman filter which is well known. Here, only the final results are given (Salychev 1998):

$$\hat{x}_k = \Phi_{k,k-1} \hat{x}_{k-1} + K_k (z_k - H_k \Phi_{k,k-1} \hat{x}_{k-1}) \tag{4.9}$$

where K_k is the Kalman filter gain which is calculated by (Salychev 1998)

$$K_k = P_{k,k-1} H_k{}^T [H_k P_{k,k-1} H_k{}^T + R]^{-1} \tag{4.10}$$

where $P_{k,k-1}$ is a priori covariance matrix of the estimation error and P_k is the covariance matrix of the estimation error and defined as $P_k = E(\tilde{x}_k \tilde{x}_k^T)$. P_k is caused by the modeling errors and it provides an indication of the accuracy of the estimate. They can be expressed as (Salychev 1998)

$$P_{k,k-1} = \Phi_{k,k-1} P_{k-1} \Phi_{k,k-1}^T + G_{k,k-1} Q G_{k,k-1}^T \tag{4.11}$$

$$P_k = (I - K_k H_k) P_{k,k-1} \tag{4.12}$$

Equation 4.9 consists of two terms. The first one is the solution of the deterministic system model. It estimates x using the system model neglecting the noise effects. The second is the correction to the estimation based on the measurements. The difference between the current measurement and a priori estimate is called the innovation sequence ν_k and is given by (Salychev 1998)

$$\begin{aligned} \mathrm{v}_k &= z_k - H_k \Phi_{k,k-1} \hat{x}_{k-1} \\ &= z_k - H_k \hat{x}_{k,k-1} \\ &= H_k x_k + v_k - H_k \hat{x}_{k,k-1} \\ &= H_k \tilde{x}_{k,k-1} + v_k \end{aligned} \tag{4.13}$$

where $\hat{x}_{k,k-1}$ is a priori estimation and $\tilde{x}_{k,k-1}$ is a priori estimation error.

It is obvious from the above equation that the innovation sequence includes the information of the prior estimation error which is derived from the measurements but is corrupted by the measurement noise. In order to reduce the difference between the estimate and the measurement, the estimated error is added to the estimated vector weighted by the Kalman filter gain. If the measurement is noisy, the Kalman filter gain will be low, and not too much correction will be done to the estimate. In contrast, an estimated error derived from the accurate measurement is weighted largely to correct the estimation.

In practice, the system model can never be exactly correct. Therefore the Kalman filter cannot estimate the state vector with absolute accuracy. The innovation sequence plays an important role in the Kalman filter algorithm. Especially in the adaptive algorithm, it is chosen as an indicator of the estimation error.

4.1.2 Adaptive Kalman Filter Algorithm

From the above discussion, it is clear that the Kalman filter algorithm estimation accuracy depends on the knowledge of the system model and noise statistics. An inaccurate system model and noise statistics deteriorate the performance of the Kalman filter. In reality, sometimes it is hard to obtain accurate noise characteristics. In these cases, an adaptive Kalman filter algorithm provides a better solution.

Here we deal with two typical cases: R is unknown and R and Q are unknown. Let’s deal with the first case where R is unknown. From Equation 4.13, we can derive

$$C_k = E(\mathrm{v}_k \mathrm{v}_k^T) = H_k P_{k,k-1} {H_k}^T + R_k \tag{4.14}$$

If C_k is known, R can be derived from Equation 4.14 as

$$R_k = C_k - H_k P_{k,k-1} H_k^T \tag{4.15}$$

For a stationary system, C_k can be estimated using (Salychev 1998)

$$\hat{C}_k = \frac{1}{k} \sum_{i=1}^{k} \mathrm{v}_i \mathrm{v}_i^T \tag{4.16}$$

or

$$\hat{C}_k = \frac{k-1}{k} \hat{C}_{k-1} + \frac{1}{k} \mathrm{v}_k \mathrm{v}_k^T \tag{4.17}$$

For a non-stationary system, C_k can be estimated using (Salychev 1998)

$$\hat{C}_k = \mathrm{v}_k \mathrm{v}_k^T \tag{4.18}$$

Summarizing the above derivation, the scheme for adaptive Kalman filter algorithm when Q is unknown is shown in Figure 4.1.

If Q and R are unknown, we need to derive Q and R. From Equations 4.3, 4.9 and 4.13, we can write

$$\hat{x}_k - \Phi_{k,k-1} \hat{x}_{k-1} = G_{k,k-1} \hat{W}_{k-1} \tag{4.19}$$

$$\hat{x}_k - \Phi_{k,k-1} \hat{x}_{k-1} = K_k \mathrm{v}_k \tag{4.20}$$

Comparing Equation 4.19 with Equation 4.20, we obtain

$$G_{k,k-1} \hat{W}_{k-1} = K_k \mathrm{v}_k \tag{4.21}$$

Calculating the covariance of the two sides of the above equation, the result is

$$G_{k,k-1} \hat{Q}_{k-1} G_{k,k-1}^T = K_k E(\mathrm{v}_k \mathrm{v}_k^T) K_k^T \tag{4.22}$$

Under the condition $K_k \approx K_{k-1}$, Equation 4.22 can be written as

$$G_{k,k-1}\hat{Q}_{k-1}G^{T}_{k,k-1} = K_{k-1}\hat{C}_{k}K^{T}_{k-1} \tag{4.23}$$

The scheme for an adaptive Kalman filter algorithm when Q and R are unknown is shown in Figure 4.2 (Salychev 1998).

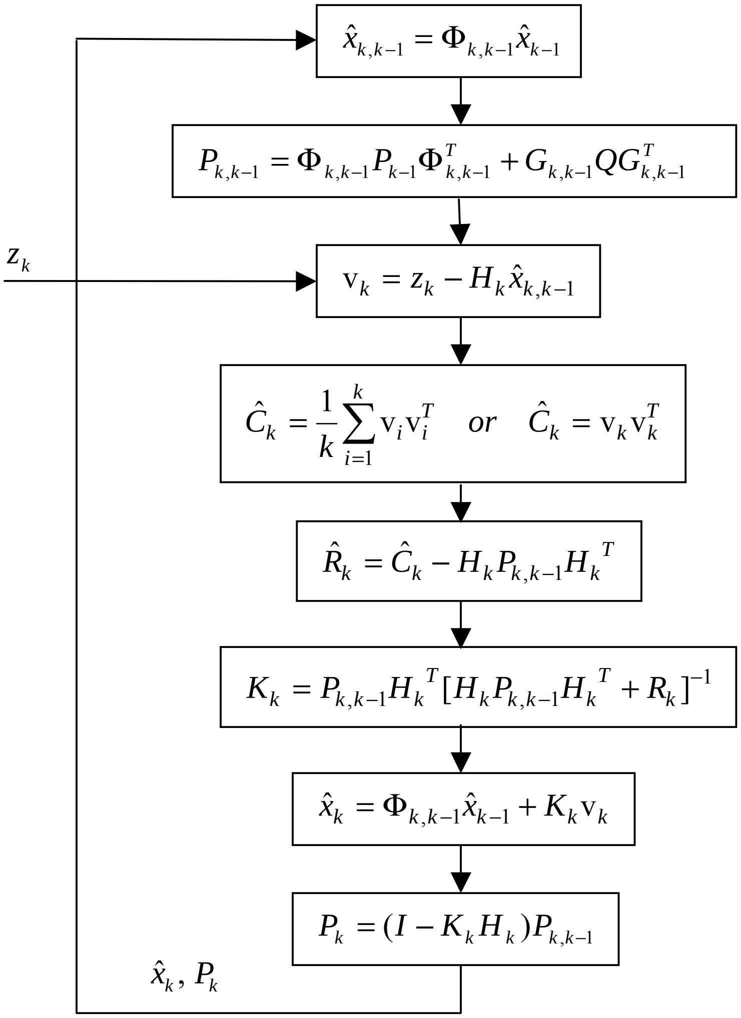
$\hat{x}_{k,k-1} = \Phi_{k,k-1}\hat{x}_{k-1}$
$P_{k,k-1} = \Phi_{k,k-1}P_{k-1}\Phi^{T}_{k,k-1} + G_{k,k-1}QG^{T}_{k,k-1}$
z_k
$\mathrm{v}_k = z_k - H_k\hat{x}_{k,k-1}$
$\hat{C}_k = \frac{1}{k}\sum_{i=1}^{k}\mathrm{v}_i\mathrm{v}_i^T \quad or \quad \hat{C}_k = \mathrm{v}_k\mathrm{v}_k^T$
$\hat{R}_k = \hat{C}_k - H_kP_{k,k-1}{H_k}^T$
$K_k = P_{k,k-1}{H_k}^T[H_kP_{k,k-1}{H_k}^T + R_k]^{-1}$
$\hat{x}_k = \Phi_{k,k-1}\hat{x}_{k-1} + K_k\mathrm{v}_k$
$P_k = (I - K_kH_k)P_{k,k-1}$
$\hat{x}_k,\ P_k$

Figure 4.1: Adaptive Kalman Filter Algorithm (R unknown)

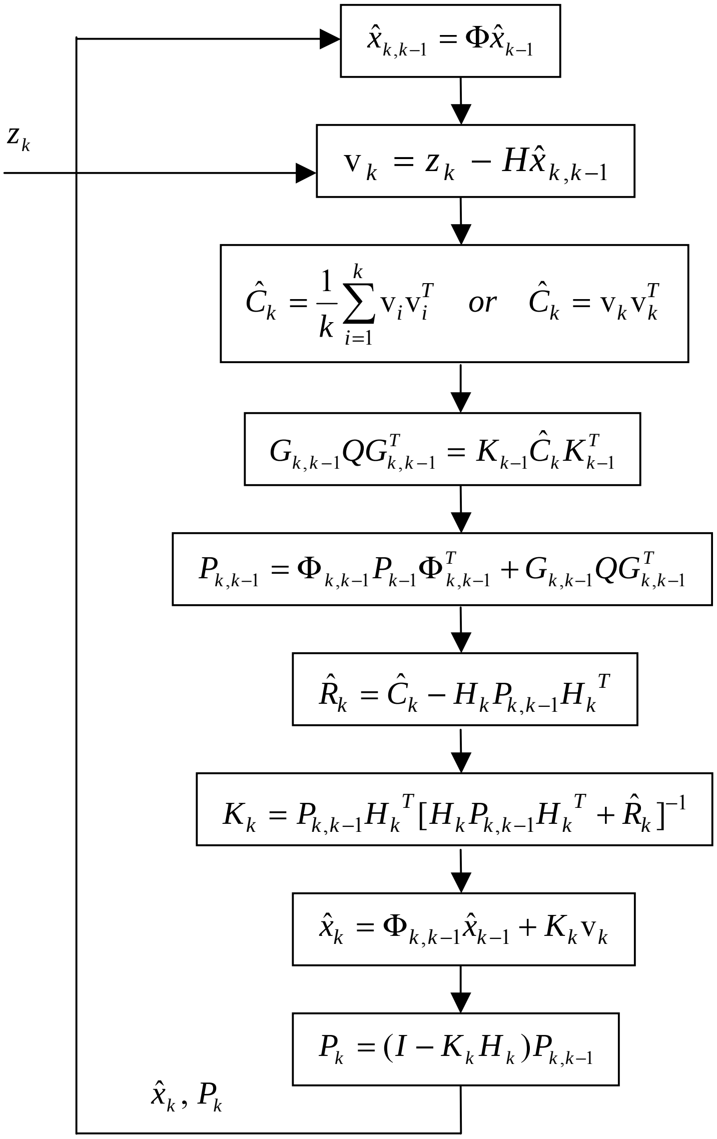
$\hat{x}_{k,k-1} = \Phi\hat{x}_{k-1}$
z_k
$\mathrm{v}_k = z_k - H\hat{x}_{k,k-1}$
$\hat{C}_k = \frac{1}{k}\sum_{i=1}^{k}\mathrm{v}_i\mathrm{v}_i^T \quad or \quad \hat{C}_k = \mathrm{v}_k\mathrm{v}_k^T$
$G_{k,k-1}QG_{k,k-1}^T = K_{k-1}\hat{C}_kK_{k-1}^T$
$P_{k,k-1} = \Phi_{k,k-1}P_{k-1}\Phi_{k,k-1}^T + G_{k,k-1}QG_{k,k-1}^T$
$\hat{R}_k = \hat{C}_k - H_kP_{k,k-1}H_k{}^T$
$K_k = P_{k,k-1}H_k{}^T[H_kP_{k,k-1}H_k{}^T + \hat{R}_k]^{-1}$
$\hat{x}_k = \Phi_{k,k-1}\hat{x}_{k-1} + K_k\mathrm{v}_k$
$P_k = (I - K_kH_k)P_{k,k-1}$
$\hat{x}_k, P_k$

Figure 4.2: Adaptive Kalman Filter Algorithm (Q and R unknown)

4.2 Design Scheme

The design scheme for using a Kalman filter in carrier tracking loop is shown in Figure 4.3

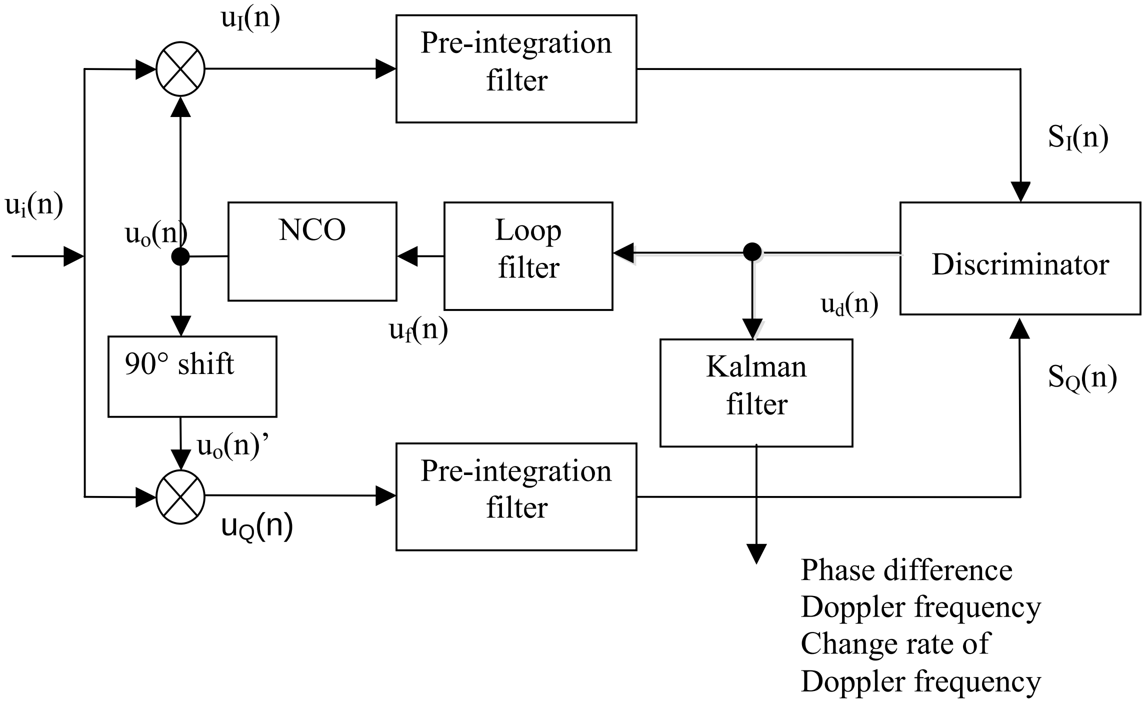

Figure 4.3: Design scheme for using a Kalman filter in a PLL

If the Kalman filter is removed, Figure 4.3 is a common COSTAS loop with two pre-integration filters. The pre-integration filter performs integration and dump functions. The integration time is determined by the incoming signal dynamics and signal-to-noise ratio. In order to increase the signal power, longer integration time is needed. However, in high dynamics situations, Doppler frequency changes faster with time and a longer pre-integration time cannot provide an accurate Doppler frequency during the pre-integration interval. Consequently, the pre-integration effect deteriorates and in this case a short pre-integration time is preferred. In this thesis, 1 ms is used.

The output from the discriminator is very noisy and fed into the loop filter. Noise that is beyond the bandwidth of the loop is filtered out by the loop filter. Noise which is within the bandwidth of the loop will pass through the loop filter, go into the NCO and will subsequently affect the tracking performance. Especially for high dynamic applications which require wide bandwidths, more noise will pass through the loop filter and cause the carrier phase measurement to be very noisy. In this thesis, the PLL bandwidth is configured at a high value of 18 Hz in order to track the high signal dynamics. Under this situation, the measurement from the PLL is very noisy and therefore the carrier measurement is not satisfactory. To address this problem, this design uses a Kalman filter to obtain less noisy measurements. Making use of this characteristic, the Kalman filter can change the weight on the measurement according to the noise level of the measurement. A much less noisy measurement can be obtained from a Kalman filter-based tracking algorithm.

The Kalman filter-based tracking algorithm takes the discriminator output as a measurement and deploys a carrier phase dynamic model to estimate the phase difference, Doppler frequency and the change rate of the Doppler frequency. However, the estimated phase difference and Doppler frequency are not used for Doppler removal. They are only used as measurements. If the Kalman filter-based tracking algorithm is used instead of the total PLL, the results often diverge. In order to keep reliable tracking capability and at the same time provide more accurate measurements, this research combines the PLL with the Kalman filter-based tracking algorithm. The PLL guarantees a basic tracking ability and the Kalman filter-based tracking algorithm refines the tracking performance with the help of the PLL tracking information (phase difference).

4.3 Kalman Filter-Based Tracking Algorithm

The estimated parameters of the Kalman Filter are phase difference, Doppler frequency and change rate of Doppler frequency. The measurement is the phase difference from the output of the discriminator.

4.3.1 System Model

Assuming that the acceleration along the line-of-sight is constant, the Doppler frequency in the incoming signal can be expressed as

$$f_d(t) = f_v + f_a t \tag{4.24}$$

where f_d is the overall Doppler frequency in the incoming signal, f_v is the frequency shift caused by the relative velocity and f_a is the change rate of the frequency shift caused by the acceleration along the line-of-sight between the satellite and the receiver.

From Equation 4.24, the Doppler frequency over a period of Δt is obtained as

$$f_d(t+\Delta t) = f_d(t) + f_a \Delta t \tag{4.25}$$

The carrier phase variation caused by the Doppler frequency in the incoming signal over a period of Δt is

$$\begin{aligned} \Delta\theta_1 &= \int_0^{\Delta t} f_d(t+\tau)d\tau \\ &= \int_0^{\Delta t}[f_d(t)+f_a\tau]d\tau \\ &= f_d\Delta t + \frac{1}{2}\Delta t^2 \end{aligned} \tag{4.26}$$

Now we can create the carrier phase dynamics model as (Psiaki 2000, Psiaki 2001, Psiaki & Jung 2002, Jung & Psiaki 2003, Ziedan & Garrison 2003)

$$\begin{bmatrix} \theta_e \\ f_d \\ f_a \end{bmatrix}_k = \begin{bmatrix} 1 & \Delta t & \dfrac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_e \\ f_d \\ f_a \end{bmatrix}_{k-1} - \begin{bmatrix} \Delta t \\ 0 \\ 0 \end{bmatrix} f_{nco\_d} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} W_n \tag{4.27}$$

where θ_e is the carrier phase error between the incoming signal and the output of the NCO, and f_{nco_d} is the Doppler frequency replicated by the NCO at the moment k-1. Δt is the update period and equals to the pre-integration time. In this thesis, $\Delta t = 1$ ms is chosen. $W_n = [W_\theta \; W_d \; W_a]^T$ is a noise vector which consists of discrete Gaussian white noise sequences .

The next task is to calculate the covariance of W_n . Assuming that W_θ W_d W_a are independent and the spectral intensity of the continuous-time white noise process of W_θ, W_d and W_a are Q_θ, Q_d, Q_a respectively, the corresponding spectral intensity matrix Q_c of continuous-time white noise process is given by

$$Q_c = \begin{bmatrix} Q_\theta & 0 & 0 \\ 0 & Q_d & 0 \\ 0 & 0 & Q_a \end{bmatrix}$$

(4.28)

The covariance Q which corresponds to the discrete noise process can be calculated by (Ziedan & Garrison 2003)

$$Q = \int_0^{\Delta t} \Phi_{k,k-1} Q_c \Phi_{k,k-1}^T dt \tag{4.29}$$

From Equation 4.27, we obtain

$$\Phi_{k,k-1} = \begin{bmatrix} 1 & \Delta t & \dfrac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{4.30}$$

Substituting Equation.4.30 and 4.28 into Equation.4.29, we obtain

$$Q = Q_\theta \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + Q_d \begin{bmatrix} \dfrac{\Delta t^3}{3} & \dfrac{\Delta t^2}{2} & 0 \\ \dfrac{\Delta t^2}{2} & \Delta t & 0 \\ 0 & 0 & 0 \end{bmatrix} + Q_a \begin{bmatrix} \dfrac{\Delta t^5}{20} & \dfrac{\Delta t^4}{8} & \dfrac{\Delta t^3}{6} \\ \dfrac{\Delta t^4}{8} & \dfrac{\Delta t^3}{3} & \dfrac{\Delta t^2}{2} \\ \dfrac{\Delta t^3}{6} & \dfrac{\Delta t^2}{2} & \Delta t \end{bmatrix} \tag{4.31}$$

If we assume that Q_θ is caused by the receiver clock phase bias, Q_d is caused by the receiver clock frequency drift and Q_a is caused by the acceleration between the satellite and the receiver along the line-of-sight, Q_θ and Q_d can be determined depending on the receiver clock quality in the application and Q_a can be determined depending on the dynamics (Psiaki 2000, Psiaki 2001, Psiaki & Jung 2002, Jung & Psiaki 2003). Q_θ and

Q_d can be calculated using the key Allan variance parameters h_0, h_{-1} and h_{-2} according to the following formulas (Brown & Hwang 1992):

$$Q_\theta = S_f = 2h_0 \tag{4.31}$$

$$Q_d = S_g = 8\pi^2 h_{-2} \tag{4.32}$$

where S_f and S_g are spectral amplitudes.

The typical Allan variance parameters for various clocks are listed in Table 4.1 (Brown & Hwang 1992). An ovenized crystal clock is used in this thesis,

Table 4.1: Allan variance parameters for various clocks

Clock types	h_0	h_{-1}	h_{-2}
Crystal	2×10^{-19}	7×10^{-21}	2×10^{-20}
Ovenized Crystal	8×10^{-20}	2×10^{-21}	4×10^{-23}
Rubidium	2×10^{-20}	7×10^{-24}	4×10^{-29}

4.3.2 Measurement Model

In this thesis, the output of the discriminator is chosen as the measurement. There are two reasons for this selection. The first reason is just as mentioned before to simplify the algorithm. If we can select a suitable discriminator which provides a linear relationship with the carrier phase error, we can simply use an ordinary Kalman filter rather than more complicated Extended Kalman filter. The second reason is to eliminate the data bits effect. As discussed in Section 2.3, at the output of the discriminator the navigation data is removed (the code is removed after correlation). Otherwise we have to consider its effect in the Doppler frequency tracking. Here an Arctan discriminator is selected which has a very good linear property within $\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$. The relationship between the inputs and outputs of the discriminator is given by

$$\theta_{e\_mea} = \arctan(\frac{S_Q}{S_I}) \tag{4.33}$$

where θ_{e_mea} is the phase error between the incoming signal and the NCO. S_Q and S_I are outputs of the pre-integration filters.

The measurement can be modeled as θ_{e_ave}, which is the average of the phase error over the period Δt. θ_{e_ave} can be derived as follows:

$$\begin{aligned}\theta_{e\_ave}(\Delta t) &= \frac{1}{\Delta t}\int_0^{\Delta t}[\theta_e(k-1) + f_d(k-1)\tau + \frac{1}{2}f_a(k-1)\tau^2]d\tau - \frac{1}{\Delta t}\int_0^{\Delta t} f_{nco\_d}\tau d\tau \\ &= \theta_e(k-1) + \frac{1}{2}f_d(k-1)\Delta t + \frac{1}{6}f_a(k-1)\Delta t^2 - \frac{1}{2}f_{nco\_d}\Delta t\end{aligned} \tag{4.34}$$

The measurement model is given by (Psiaki 2000, Psiaki 2001, Psiaki & Jung 2002, Jung & Psiaki 2003).

$$\theta_{e\_mea} = \begin{bmatrix} 1 & \frac{\Delta t}{2} & \frac{\Delta t^2}{6} \end{bmatrix} \begin{bmatrix} \theta_e \\ f_d \\ f_a \end{bmatrix}_{k-1} - \frac{\Delta t}{2} f_{nco\_d} + V_k \tag{4.35}$$

where V_k is measurement noise which is a Gaussian white noise sequence and its covariance matrix is R. In this thesis, an adaptive R algorithm described in section 4.12 is utilized.

CHAPTER FIVE: APPLYING WAVELET DE-NOISING TECHNIQUE IN PLL

This chapter starts with a review of some basic concepts of wavelet. The focus is on an explanation of wavelet de-noising technique. This is followed by a detailed introduction of how to implement the wavelet de-noising technique in a PLL.

5.1 Wavelet De-noising Review

5.1.1 Introduction

The Wavelet transform was introduced at the beginning of the 1980s. Since then, various types of Wavelet transforms have been developed, along with many applications. The continuous-time Wavelet transform has been found to be an effective tool for data analysis. Compared to the continuous-time Wavelet transform, the discrete Wavelet transform (DWT) is much more efficient and is suitable for much wider applications. The DWT has an excellent compaction signal format and has therefore has the advantage of being computationally very efficient. The DWT has been applied to many fields including image compression, de-noising, numerical integration and pattern recognition (Mertins 1999).

The most well known tool for signal analysts is Fourier analysis, which breaks down a signal into constituent sinusoids of different frequencies. This means that Fourier analysis transforms signals from the time domain to the frequency domain. However, Fourier analysis has a serious drawback. In transforming to the frequency domain, time information is lost. In the frequency domain, it is impossible to tell which frequencies the signal contains at a given time. In order to overcome this deficiency, a “window” technique was created which analyzes only a small section of the signal at a time. The corresponding transform is called the Short-Time Fourier Transform (STFT). The STFT maps a signal into a two-dimensional function of time and frequency. It provides the signal’s frequency information at different periods of time. The drawback is that the window size is fixed and is not flexible to analyze different frequencies. Many applications require a more flexible approach; for example, for high frequencies it is more efficient to deploy a narrow window and, for low frequencies, a wide window (Mertins 1999).

Wavelet transform not only has the property of STFT but also has a distinctive one – a variable-size window which uses a narrower time window to analyze high frequency signals and a wider time window to analyze low frequency signals. Different from the Fourier transform, Wavelet analysis uses a time-scale domain rather than time-frequency.

5.1.2 Wavelet Transform

A wavelet is a waveform of effectively limited duration that has an average value of zero (The Mathworks, Inc. 2004). It is defined as

$$\psi_{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right), \qquad a \in R, a \neq 0; b \in R \tag{5.1}$$

where ψ (t) is called the mother wavelet or the basic wavelet which is the wavelet prototype, a is the scale factor, b is a shift factor and R is the real domain. From Equation 5.1, it is obvious that the wavelet is acquired by scaling and shifting the basic wavelet. Scaling a wavelet simply means dilating (or compressing) it. The larger (smaller) the scale factor, the more “stretching” (“compressing”) is the wavelet. The scale factor is related to the frequency of the signal. Shifting a wavelet simply means translating it in the time domain (The Mathworks, Inc. 2004).

5.1.2.1 Continuous Wavelet Transform

Similar to the Fourier transform, the continuous wavelet transform (CWT) is defined as the integration of the signal $f(t)$ multiplied by the scaled, shifted versions of the wavelet function ψ over time (The Mathworks, Inc. 2004) as

$$C(a,b) = \int_{-\infty}^{+\infty} f(t)\psi_{a,b}(t)dt = \int_{-\infty}^{+\infty} f(t)|a|^{-1/2}\psi^{*}\left(\frac{t-b}{a}\right)dt \tag{5.2}$$

where C is called a wavelet coefficient. This transform produces many wavelet coefficients that are a function of scale and translation. C represents how much a specific section of the signal correlates to a given wavelet. Higher C values correspond to high similarity. In practice, C is calculated according to the following (The Mathworks, Inc. 2004): first choose a wavelet and compare it with the initial section of the signal to be analyzed, then compute C using Equation 5.2, after that shift the wavelet from the initial section to the end section of the signal and compute the corresponding C; and, lastly, change the scale of the wavelet and repeat steps 2 and 3. This process is illustrated in Figure 5.1.

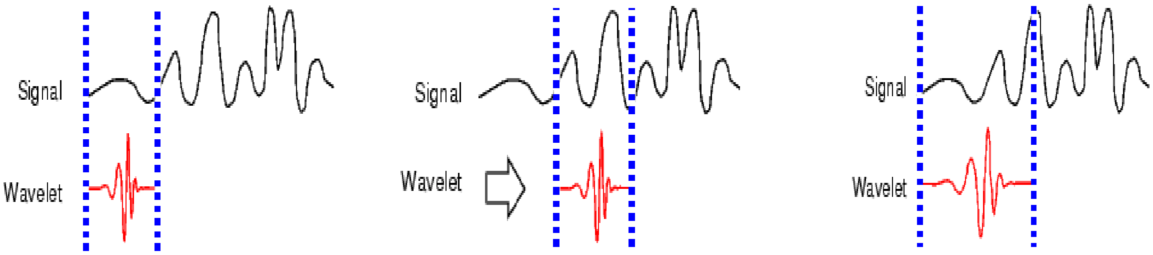

Figure 5.1: Illustration of wavelet shift and scale concepts (from The Mathwork, Inc. 2004)

The wavelet transform produces a time-scale aspects of a signal. The scale and frequency are related. A low scale corresponds to a compressed wavelet which shows the rapid change feature of the signal. Consequently we obtain the high frequency character of the signal. By contrast, a high scale corresponds to a stretched wavelet which shows the slow change feature of the signal. Therefore we obtain the low frequency characteristic of the signal.

5.1.2.2 Discrete Wavelet Transform (DWT)

The continuous wavelet coefficients have some redundancy. In some applications such as signal coding or compression, time-scale factors are sampled so that the signal is represented by wavelet coefficients in an economical manner. A typical choice is

$$a = 2^j, \qquad j \in z \tag{5.3}$$

$$b = k2^{j}, \qquad k \in z \tag{5.4}$$

where z is the integer domain.

First considering the case that the time is continuous but the scale and shift are sampled. The wavelet in this case is written as

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi(2^{-jt} - k) \tag{5.5}$$

The corresponding wavelet transform is given by (The Mathworks, Inc. 2004)

$$C_{i,j}(t) = 2^{-\frac{j}{2}} \int_{-\infty}^{+\infty} f(t)\psi^{*}(2^{-jt} - k)dt \tag{5.6}$$

This produces a so-called "dyadic" grid in a time-scale domain as shown in figure 5.2 (The Mathworks, Inc. 2004).

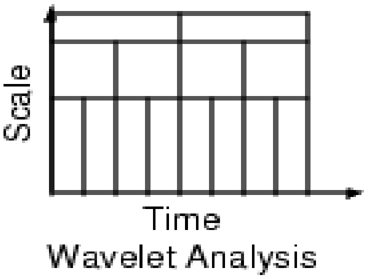

Figure 5.2: Discrete wavelet analysis (from The Mathwork, Inc. 2004)

5.1.3 Signal Decomposition

An efficient way to calculate the wavelet coefficients was developed by Mallat (1989) using filters. Based on this idea, a fast iterative algorithm was developed to compute the coefficients efficiently. In this algorithm, a special wavelet prototype was constructed for which the perfect reconstruction of the signal is possible. The synthesis takes the form

$$f(t)=\sum_{j\in Z}\sum_{k\in Z}C(j,k)\psi_{j,k}(t)\,. \tag{5.7}$$

Equation 5.5 is an orthonormal basis of $L^2(R)$. Since a basis consists of linearly independent functions, $L^2(R)$ may be understood as the direct sum of subspaces (Mertins 1999) as

$$L^2(R)=\cdots\oplus W_{-1}\oplus W_0\oplus W_1\oplus\cdots \tag{5.8}$$

with

$$W_j=span\left\{2^{-\frac{j}{2}}\psi(2^{-jt}-k),k\in Z\right\},\qquad j\in Z \tag{5.9}$$

where W_j represents a certain subspace which corresponds to a certain frequency band. For a given j and $k\in Z$, $C(j,k)$ gives the frequency information of the signal in the jth sub-band. The signal component corresponds to this sub-band can be expressed as

$$cD_j(t)=\sum_{k\in Z}C(j,k)\psi_{j,k}(t),\qquad cD_j(t)\in W_j \tag{5.10}$$

where $cD_j(t)$ is called the detail coefficient of the signal at level j. The signal can be represented as

$$f(t)=\sum_{j\in Z}cD_j(t) \tag{5.11}$$

Now we define the subspaces V_j, $j\in Z$ as the direct sum of V_{j+1} and W_{j+1} (Mertins 1999)

$$V_j=V_{j+1}\oplus W_{j+1} \tag{5.12}$$

and the subspace V_j has the property

$$\cdots\subset V_{j+1}\subset V_j\subset V_{j-1}\subset\cdots$$

For W_{j+1}, the lower j corresponds to the higher frequency components. Consequently V_{j+1} contains the low-pass components ofV_j. On the other hand, the subspace V_j should have the scaling property. For example $f(2t)$, which is obtained by scaling the signal $f(t)$ by the factor of two is in the subspace V_{j-1} and vice versa (Mertins 1999)

$$f(t) \in V_j \qquad \Leftrightarrow \qquad f(2t) \in V_{j-1} \tag{5.13}$$

In addition, if we form a sequence of functions $cA_j(t)$ by projection of $f(t) \in L^2(R)$ onto the subspaces V_j, based on Equation 5.12, we can derive the following expression (Mertins 1999)

$$cA_{j-1}(t) = cA_j(t) + cD_j(t), \qquad cA_{j-1}(t) \in V_{j-1} \tag{5.14}$$

where $cA_j(t)$ is defined as the approximation of the signal at level j. Combining Equation 5.11 and Equation 5.14 the signal f can be expressed as

$$f = cA_j + \sum_{J<J} cD_J \tag{5.15}$$

This means that the signal can be decomposed into several levels, and at each level, the signal is decomposed into the approximation and the detail coefficients. The approximation coefficients are the high-scale, low-frequency components of the signal. The detail coefficients are the low-scale, high-frequency components.

The detail and approximation coefficients can be acquired by means of a set of low-pass filters h and their complementary high-pass filters g. The decomposition algorithm starts with signal f to calculate the approximation and detail coefficients at the first level, and then those of the second level, and so on. This process is shown in Figure 5.3 (Mallat 1998, Zhang et al. 2001).

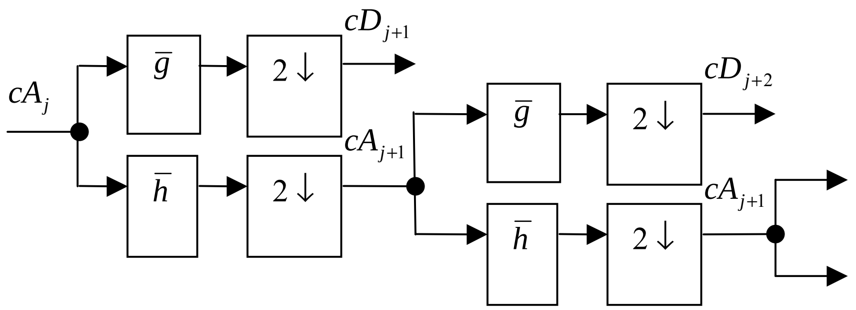

Figure 5.3: Signal Decomposition

5.1.4 Signal Reconstruction

Signal reconstruction is an inverse process of the signal decomposition. It begins at the highest level and uses the detail and approximation coefficients to build the next lower level approximation coefficient. The reconstruction process is illustrated in Figure 5.4 (Mallat 1998, Zhang et al 2001).

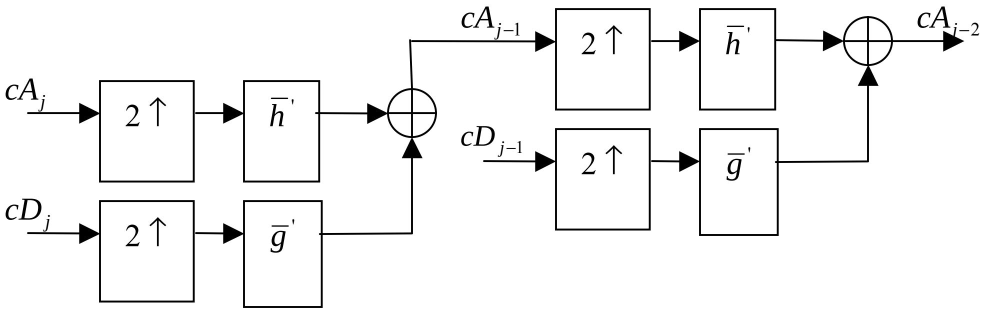

Figure 5.4: Signal Reconstruction

5.2 Wavelet De-noising by Soft-thresholding

The objective of de-noising is to suppress the noise part of a noisy signal in order to recover the real signal.

Wavelet de-noising is one of the most significant applications of wavelets. The soft thresholding-based de-noising algorithm has proven to have the advantages of optimizing MSE (mean-squared error) and to keep the smoothness of the de-noised signal (Donoho 1995). Because of these advantages, the wavelet de-noising technique has been used in this thesis.

The underlying model for the noisy signal is basically of the following form:

$$s(n) = f(n) + \sigma e(n) \tag{5.16}$$

where $f(n)$ is signal and $e(n)$ is the noise with level σ. In the simplest model we suppose that $e(n)$ is a Gaussian white noise. The general de-noising procedure involves three steps (Donoho 1995, Taswell 2000, Zhang et al 2001, Mertins 1999): first decompose the signal using an octave-band filter bank implementing a discrete wavelet transform, then threshold the wavelet coefficients to remove noise, and lastly reconstruct the signal using the modified wavelet coefficients.

5.2.1 Decomposition

An effective decomposition algorithm is discussed in Section 5.1.3. Here what is needed is to apply this algorithm to the noisy signal $s(n)$. First choose a wavelet, choose a decomposition level N and then compute the wavelet decomposition (approximate coefficient and detail coefficient) of the signal at each level. This process is illustrated in Figure 5.5.

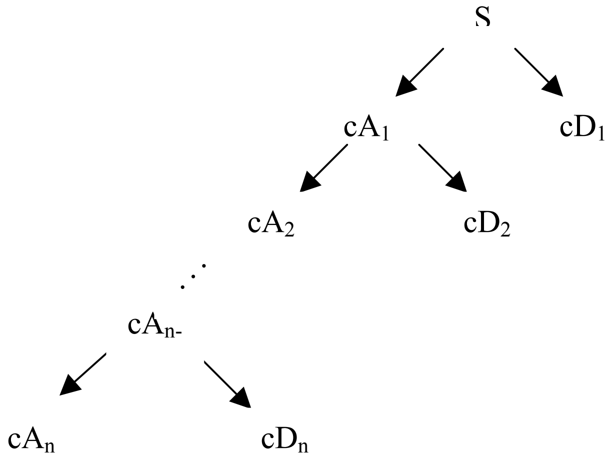

Figure 5.5: Decomposition of the signal

5.2.2 Threshold Detail Coefficients

In this step, for each level from 1 to N, a threshold is computed first and then a soft thresholding is applied to the detail coefficients. The threshold is chosen based on (Donoho 1995)

$$thr = r_1 \sigma \sqrt{\frac{2\log(n)}{n}} \qquad (5.17)$$

where r_1 is a constant and can be set to 1 while choosing the orthogonal wavelet, and n is the number of the data samples. Equation 5.17 shows that the threshold is proportional to the noise level σ and also related to the length n of the noisy signal. Considering a simple Gaussian white noise case, the detail coefficients at any level from the standard DWT are Gaussian white noise with the same standard deviation as the noise level σ.

Next the threshold is applied to the detail coefficients at each level. Two approaches were proposed for this purpose: hard and soft thresholding. Both methods are illustrated in Figure 5.6. The hard thresholding uses the following non-linearity (Mertins 1999):

$$mcD_j = \begin{cases} cD_j, & \left|cD_j\right| > thr \\ 0 & \left|cD_j\right| \le thr \end{cases} \tag{5.18}$$

where mcD_j is called the modified detail coefficient.

The soft thresholding uses the following non-linearity (Donoho 1995):

$$mcD_j = \operatorname{sgn}(cD_j)(\left|cD_j\right| - thr)_+ \tag{5.19}$$

where the sign '+' indicates a nonlinear process that only keeps positive values. It has the following property

$$(\left|cD_j\right| - thr)_+ = \begin{cases} \left|cD_j\right| - thr & \left|cD_j\right| > thr \\ 0 & \left|cD_j\right| \le thr \end{cases}$$

The function sgn has the following property:

$$\operatorname{sgn}(cD_j) = \begin{cases} 1 & if\ cD_j \ge 0 \\ -1 & if\ cD_j < 0 \end{cases} \tag{5.20}$$

Compared to hard thresholding, soft thresholding has several beneficial properties. If $\left|cD_J\right| \le thr$, both the hard and soft thresholding do the same thing – set the detail coefficients to zero. This process means that these detail coefficients are caused by noise and should be cut off. If $|cD_J| > thr$, the hard thresholding assumes that they are signals and do nothing about them. But soft thresholding assumes that noise is superimposed on the signals and therefore it removes the noise from the signals, which acts like shrinking the nonzero coefficients towards 0 by the amount of *thr* .Therefore, the soft-thresholding provides a more effective de-noising result.

If the threshold is chosen properly, the noise level in the reconstructed signal is much lower than the one in the original. If the threshold is too small, the improvement is not very much. On the other hand, a threshold that is too large may cause signal distortion. Therefore the de-noising performance depends on the proper selection of the threshold.

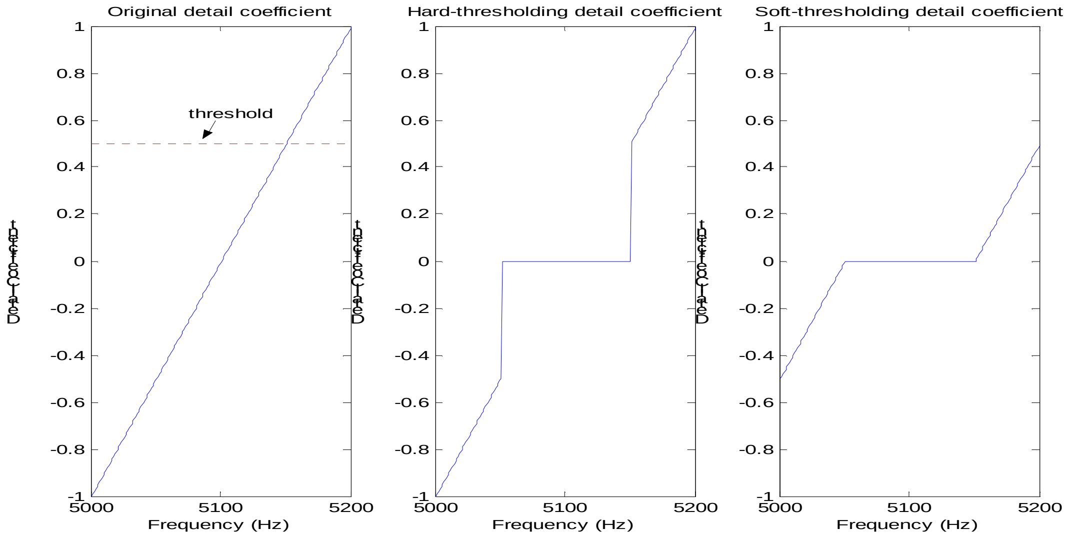

Figure 5.6: Hard and Soft Thresholding (from The Mathworks, Inc. 2004).

5.2.3 Reconstruction

Reconstruction is the inverse process of decomposition. Also the direction is inversed. From the highest level N, reconstruction is performed using the original approximation coefficients and the modified detail coefficients at level N. This process is shown in Figure 5.7.

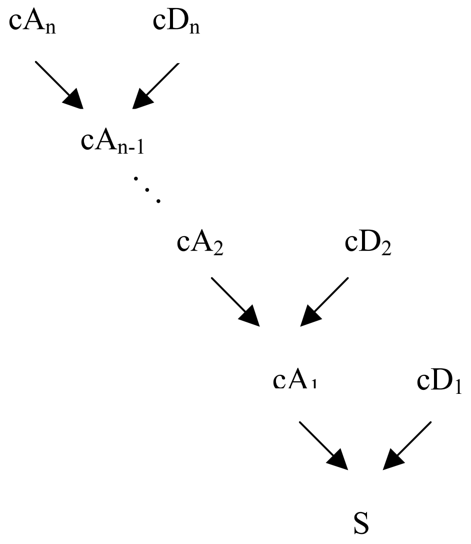

Figure 5.7: Reconstruction of the signal

From the above discussion, it is clear that the wavelet de-noising technique is different from filtering. The former method decreases the noise part of the signal at any place of the frequency bandwidth and at the same time keeps the frequency components of the signal, whereas the latter method eliminates (ideally) the noise beyond the frequency band of the signal but does nothing with the noise within the frequency bandwidth of the signal.

5.3 Applying Wavelet De-noising Technique in PLL

As discussed previously, there is a contradiction in the PLL design. Reducing noise and improving the tracking performance require a narrow PLL bandwidth but tracking high dynamic signals requires a wide one. Wavelet de-noising provides a better design scheme to deal with this problem. The rationale is quite simple: First decrease the noise level before the loop filter using a wavelet de-noising technique (the wavelet de-noising technique can also be placed at the output of the loop filter). Because the noise level is reduced, this allows one to increase the bandwidth of the loop to track high dynamic signals. This design scheme is shown in Figure 5.8.

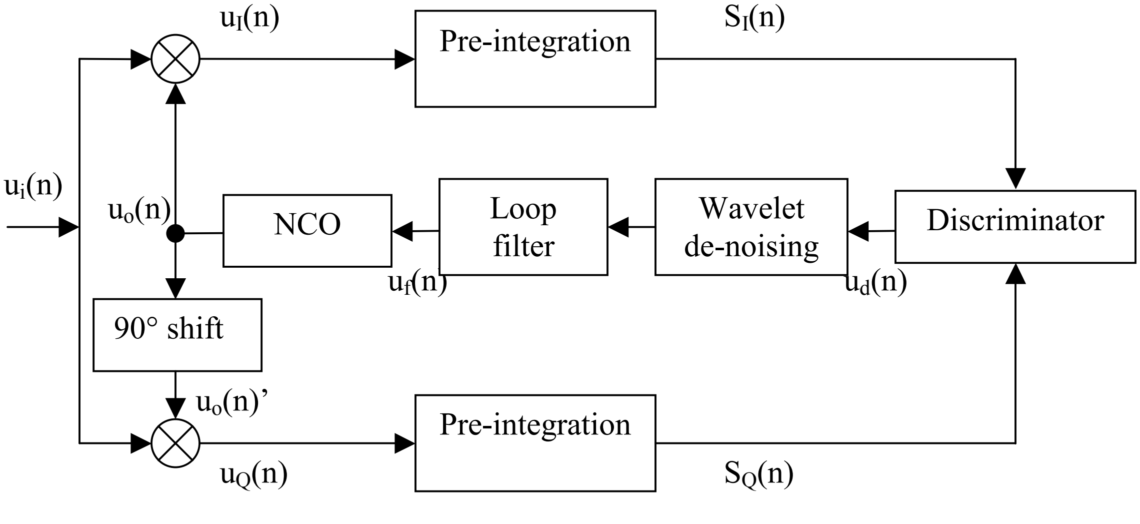

Figure 5.8: Applying Wavelet De-noising Technique in a PLL

The main function of the wavelet de-noising is to reduce the noise level within the bandwidth of the loop filter. It is known that the noise beyond the bandwidth of the loop filter will be filtered out by the loop filter, therefore only the noise within the bandwidth of the loop filter will pass through and affect the NCO tracking performance. It should be noted that the wavelet de-noising technique can only reduce the noise level rather than eliminate the noise totally. The remaining noise still can affect the NCO tracking performance but at a lower level.

The disadvantage of the wavelet de-noising technique is that it may cut off some useful signals which are smaller than the threshold. This will make the PLL spend more time to lock due to the loss of some control signals. However the decrease of the noise will produce smaller phase error which will help the PLL to maintain locked. The results shown in Chapter 7 indicate that the overall performance is improved.

CHAPTER SIX: ADAPTIVE BANDWIDTH ALGORITHM

The common solution of mitigating thermal noise and dynamic stress error is to choose the noise bandwidth for the highest dynamics. This results in a sub-optimal use of the loop during periods of low dynamics. The adaptive bandwidth algorithm is designed to adapt the noise bandwidth of the PLL in real-time in order to achieve a better tracking performance by estimating the signal dynamics. Aided by knowledge of the carrier-over-noise power ratio from the software receiver, an optimal bandwidth is calculated. This chapter first presents the design scheme. Then the method to estimate the incoming signal dynamics is introduced. After that, the strategy for selecting the optimal bandwidth is presented.

6.1 Design Scheme and PLL Linear Model

The design scheme of the adaptive bandwidth algorithm is shown in Figure 6.1

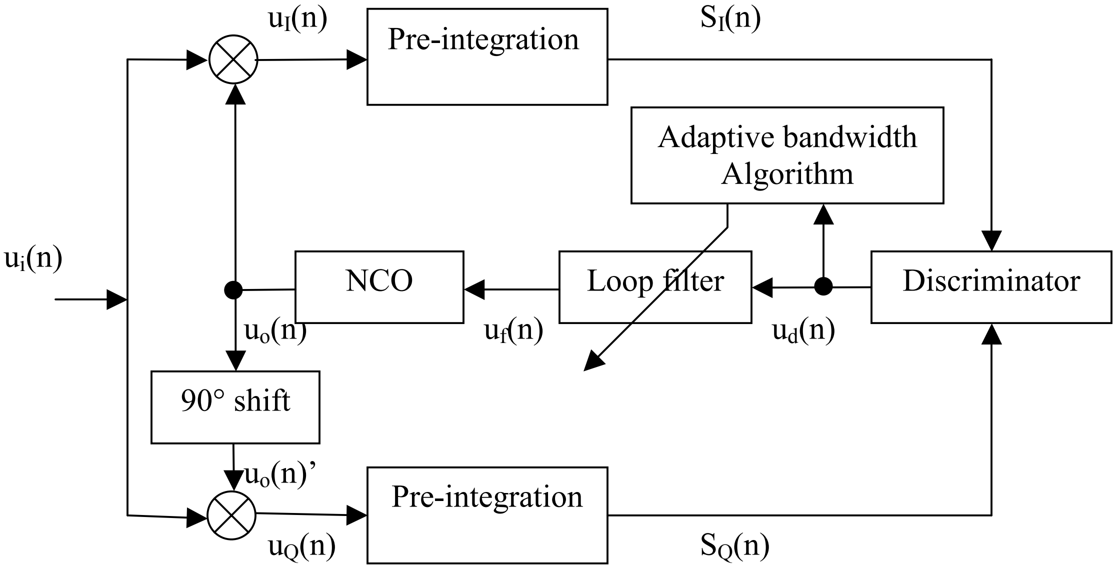

Figure 6.1: Adaptive Bandwidth Algorithm Design Scheme

As shown in Figure 6.1, this algorithm utilizes the phase difference from the output of the discriminator to estimate the dynamics of the incoming signal and then adjusts the parameters of the loop filter to change the PLL bandwidth. In order to obtain the dynamics estimation, a digital PLL phase model is required.

Unlike LPLL, the analysis of the digital loop is based on Z the transform. The transfer function of the loop filter in the Z domain is given by Equation 2.26 and the transfer function of the NCO in the Z domain is given by Equation 2.28. The simplified linear phase model of a digital PLL is shown in Figure 6.2

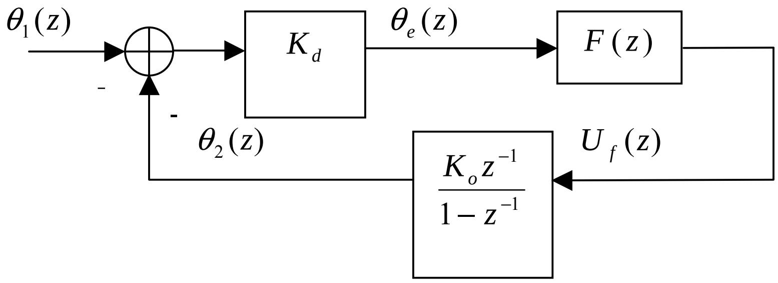

Figure 6.2: Linear phase model of a digital PLL

From this model, we can calculate the transfer function of the phase difference (Best 1999)

$$H_e(z) = \frac{\theta_e(z)}{\theta_1(z)} = \frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K\sum_{n=0}^{N-1} b_n z^{-n-1}} \tag{6.1}$$

where $K = K_o K_d$.

When the PLL is determined, $H_e(z)$ is known. $\theta_1(z)$ is the input phase caused by the Doppler frequency in the incoming signal. Therefore it is related to the incoming signal dynamics. In order to estimate the signal dynamics, the expression of $\theta_1(z)$ should be derived. Through the use of Equation 6.1, the relationship between the phase difference and signal dynamics can be obtained. If the phase difference measurement is available, the estimation of the signal dynamics is possible.

6.2 Estimation of Signal Dynamics

The distance between the satellite and the GPS receiver along the line of sight can be modeled as (Legrand & Macabiau 2000, Legrand & Macabiau 2001, Legrand & Macabiau 2001)

$$R(t) = \sum_{m=0}^{\infty} R_0^{(m)} \frac{t^m}{m!} \tag{6.2}$$

where $R_0^{(m)}$ is the coefficient for each term. $R_0^{(1)}$ is the velocity of the second term, $R_0^{(2)}$ is the acceleration of the third term and $R_0^{(3)}$ is the jerk of the fourth term. The relative velocity along the line-of-sight is the derivative of the distance

$$v(t) = \frac{d}{dt} R(t) = \sum_{m=1}^{\infty} R_0^{(m)} \frac{t^{m-1}}{(m-1)!} \tag{6.3}$$

The relative velocity generates the Doppler frequency. The relationship between them is given by

$$f_d(t) = \frac{v(t)}{c} f = \frac{v(t)}{\lambda} \tag{6.4}$$

where c is the velocity of light, f is the signal frequency (1575.42 MHz for L1 frequency) and the λ is the wavelength of the signal. Replacing Equation 6.3 in Equation 6.4, Equation 6.4 can be rewritten as

$$f_d(t) = \frac{1}{\lambda} \sum_{m=1}^{\infty} R_0^{(m)} \frac{t^{m-1}}{(m-1)!} \tag{6.5}$$

It is known that phase is the integration of the frequency. Therefore the phase caused by the Doppler frequency is written as

$$\theta_1(t) = 2\pi\int_0^t f_d(t)dt = \frac{2\pi}{\lambda}\sum_{m=1}^{\infty} R_0^{(m)} \frac{t^m}{m!} \tag{6.6}$$

Sampling $\theta_1(t)$ using a sample frequency f_s (sample period is T_S), one can write

$$t = nT_s = \frac{n}{f_s} \tag{6.7}$$

In a discrete form, Equation 6.6 is written as

$$\theta_1(n) = \frac{2\pi}{\lambda}\sum_{m=1}^{\infty} \frac{R_0^{(m)}}{m!}\frac{n^m}{f_s{}^m}\,. \tag{6.8}$$

The sample number during each integration period is designated as N_I which is equal to the integration period multiplied by the sample frequency. Under a linear condition, the synthesized input phase at the end of each integration period is the sum of the phase during the integration period (Legrand & Macabiau 2000). Therefore we can obtain

$$\theta_1(k) = \frac{2\pi}{\lambda}\sum_{m=1}^{\infty} \frac{R_0^{(m)}}{m!\,f_s{}^m}(\sum_{n=(k-1)N_I}^{kN_I-1} n^m) \tag{6.9}$$

because (Legrand & Macabiau 2000)

$$\frac{\sum_{n=(k-1)N_I}^{kN_I-1} n^m}{f_s{}^m} = \frac{k^m N_I^m}{f_s^m} = \frac{k^m}{B_L^m} \tag{6.10}$$

where $B_L = \frac{f_s}{N_I}$ is the bandwidth of the pre-integration filter. In addition,

$$\theta_1(k) = \frac{2\pi}{\lambda}\sum_{m=1}^{\infty} \frac{R_0^{(m)} k^m}{m!\,B_L^m} = \sum_{m=1}^{\infty}\theta_0^{(m)} \cdot \frac{k^m}{m!} \tag{6.11}$$

where $\theta_0^{(m)} = \frac{2\pi R_0^{(m)}}{\lambda B_L^m}$ (6.12)

Next we calculate the steady state of the phase difference that describes how the loop tracks the input phase variation. The input phase in the Z domain is

$$\theta_1(z) = \sum_{m=1}^{\infty} \frac{\theta_0^{(m)}}{m!} Z[k^m] \qquad (6.13)$$

Based on Equation 6.1 the phase difference in the Z domain is obtained by

$$\theta_e(z) = H_e(z)\theta_1(z) = \frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K\sum_{n=0}^{N-1} b_n z^{-n-1}} \sum_{m=1}^{\infty} \frac{\theta_0^{(m)}}{m!} Z[k^m]. \qquad (6.14)$$

Using the Final Value Theorem of the Z transform, the steady-state phase difference can be calculated by

$$\theta_e(\infty) = \lim_{z\to1}\left[(z-1)\theta_e(z)\right] = \lim_{z\to1}\left[(z-1)\frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K\sum_{n=0}^{N-1} b_n z^{-n-1}} \sum_{m=1}^{\infty} \frac{\theta_0^{(m)}}{m!} Z[k^m]\right]. \qquad (6.15)$$

Through analysis (the detail analysis is in Appendix A), one concludes that for the terms in the incoming signal whose orders are smaller than N , $\theta_e(\infty)$ equals zero, for the terms whose orders are larger than N, $\theta_e(\infty)$ equals to infinite, and for the term whose order equals to N, $\theta_e(\infty)$ is constant. This means that a Nth order loop can track the Nth order variation in the input phase but with a constant steady-state phase difference. However, it can track variations in the input phase whose order is less than N without error but cannot track the variation whose order is larger than N. The conclusion is summarized in Equation 6.16.

$$\theta_e(\infty) = \begin{cases} 0 & m < N \\ \dfrac{\theta_0^N}{K_O \sum_{n=0}^{N-1} b_n} & m = N \\ \infty & m > N \end{cases} \tag{6.16}$$

In this thesis, $K_O = 1$.

Based on Equation 6.16, it can shown that the steady state phase difference for a second order loop is

$$\theta_e(\infty) = \frac{\theta_0^{(2)}}{b_0 + b_1} \tag{6.17}$$

where $\theta_0^{(2)} = \dfrac{2\pi R_0^{(2)}}{\lambda B_L^2}$. (6.18)

Similarly, the steady state phase difference for a third order loop is

$$\theta_e(\infty) = \frac{\theta_0^{(3)}}{b_0 + b_1 + b_2} \tag{6.19}$$

where $\theta_0^{(3)} = \dfrac{2\pi R_0^{(3)}}{\lambda B_L^3}$. (6.20)

Equation 6.17 and 6.19 indicate that as long as the steady-state phase difference is available, the incoming signal acceleration can be obtained for a second order loop and the incoming signal jerk can be obtained for a third order loop.

For an ARCTAN discriminator, theoretically its output is the phase difference if the PLL is in a locked status. However, in reality the output of the discriminator consists of phase differences and noise. The noise is so large that only noise can be seen if one looks at the signal waveform from the discriminator. The next step is to extract the steady-state phase difference from the noise.

In this research, two steps are taken to extract less noisy information. The first step is to add the output of the discriminator together at every epoch and then multiply this sum with the pre-integration time. Using Equation 6.17 and 6.18, for a second order loop, this process is written as

$$\sum_{n=1}^{k} \theta_e(\infty)_n T = \frac{2\pi}{(b_0 + b_1)\lambda B_L^2} \sum_{n=1}^{k} R_0^{(2)}(n)T = \frac{2\pi}{(b_0 + b_1)\lambda B_L^2} Vel(k) \tag{6.21}$$

where k is the number of the pre-integration and $Vel(k)$ is the signal velocity.

Similarly, using Equation 6.19 and 6.20, for a third order loop, this process is written as

$$\sum_{n=1}^{k} \theta_e(\infty)_n T = \frac{2\pi}{(b_0 + b_1 + b_3)\lambda B_L^3} \sum_{n=1}^{k} R_0^{(3)}(n)T = \frac{2\pi}{(b_0 + b_1 + b_3)\lambda B_L^3} Acce(k) \tag{6.22}$$

where $Acce(k)$ is the signal acceleration.

The addition works as averaging and therefore the noise is reduced and the less noisy measurements $Vel(k)$ and $Acce(k)$ are obtained respectively for the second and third order loops.

The second step is to calculate the approximate derivatives of $Vel(k)$ or $Acce(k)$. Through this step, the signal acceleration is obtained for a second order loop and the signal jerk is obtained for a third order loop. For the second order loop, this process is written as

$$Acce(k) \approx \frac{Vel(k) - Vel(k-i)}{iT} \tag{6.23}$$

where i is a positive integer.

Similarly, for the third order loop this process is written as

$$Jerk(k) = \frac{Acce(k) - Acce(k-i)}{iT} \tag{6.24}$$

where $Jerk(k)$ is the signal jerk.

Actually i determines the time interval in computing the approximated derivatives. It should be noted that considering the noise influence, the time interval should not be too small.

After the estimation of the signal dynamics, next step is to determine the proper bandwidth.

6.3 Optimal bandwidth

As discussed before, the bandwidth selected is often a compromise design in order to reduce different factors effects on the tracking performance. Here, we focus on the thermal noise and dynamic stress influence and try to find an optimal bandwidth which can decrease both errors at the same time. Considering the Equation 2.56, 2.57, 2.58 and 2.63 together, the following equation should be satisfied for the second order PLL:

$$\frac{360}{2\pi}\sqrt{\frac{B_n}{C/N_0}\left(1+\frac{1}{2TC/N_0}\right)}+0.2809\frac{\frac{dR^2}{dt^2}}{3B_n^2}\leq 15^\circ$$

(6.25)

The corresponding equation for the third order PLL is

$$\frac{360}{2\pi}\sqrt{\frac{B_n}{C/N_0}\left(1+\frac{1}{2TC/N_0}\right)}+0.4828\frac{\frac{dR^3}{dt^3}}{3B_n^3}\leq 15^\circ \qquad (6.26)$$

Considering the effects of vibration induced phase noise and Allan deviation induced phase noise, the threshold should be smaller than 15°.

In the following, Equation 6.26 is taken as an example for the explanation on how to determine the optimal bandwidth using an iteration method. Here a 10° threshold is assumed.

Step1: assume that the error is caused only by dynamic stress and calculate the required bandwidth using the estimated jerk from Equation 6.24. The formula is written as

$$B_n=\sqrt[3]{0.4828\frac{\frac{dR^3}{dt^3}}{30}} \qquad (6.27)$$

Step 2: use the calculated bandwidth and carrier over noise power ratio acquired from the software receiver (otherwise it has to be calculated) to calculate the error σ_t caused by thermal noise using Equation 2.56

Step 3: use the thermal noise error calculated in step 2 and threshold to calculate the dynamic stress error θ_e using the following equation

$$\theta_e = 3 \times (10^\circ - \sigma_t) \tag{6.31}$$

Step 4: use the dynamic stress error calculated in step 3 to calculate the required bandwidth using

$$B_n = \sqrt[3]{0.4828 \frac{\frac{dR^3}{dt^3}}{\theta_e}} \tag{6.32}$$

Step 5: iterate step 2, 3 and 4 until the bandwidth variation between two iterations is less that a certain small value, 0.1Hz is chosen in this thesis.

CHAPTER SEVEN: TEST RESULTS AND ANALYSIS

The algorithms discussed in this thesis were tested using signals simulated with a GPS hardware receiver, a GPS receiver's front end which outputs a digitalized intermediate frequency (IF) and the GNSS_SoftRx™ software receiver developed by the PLAN group. The first section introduces the configuration of the test including hardware and software. In order to test the proposed algorithms performance, two tests are designed to demonstrate the tracking performance under high dynamics applications – a horizontal motion and a three-dimensional motion. The second and third section show test results for real-time and post-processing scenarios. The statistical results from an ordinary PLL and the proposed algorithms are presented in the last section.

7.1 Test Configuration

7.1.1 Test Scheme

The GSS 6560 GPS hardware simulator which can provide L1 C/A code signals under various scenarios is used to generate the signals. An aircraft flight is chosen to generate high dynamics motion. An Agilent Signal Generator (E 4431B) is used to generate white noise which is input into a signal combiner GSS 4766 together with the output of the GPS simulator. Therefore the output of the combiner provides simulated GPS signals with noise. The generated white noise is utilized to change the Signal-to-Noise Ratio. A LNA (low noise amplifier) is used to play the same role as the LNA in a real GPS antenna which can provide a 30 dB gain. Connected with the LNA is a GPS front end made by ACCORD Company, which can output digitized GPS IF signals with variable sample frequencies ranging from 2 MHz to 20 MHz. At last, the software receiver does the signal processing after IF and the computation of the navigation solution. The test configuration is shown in Figure 7.1.

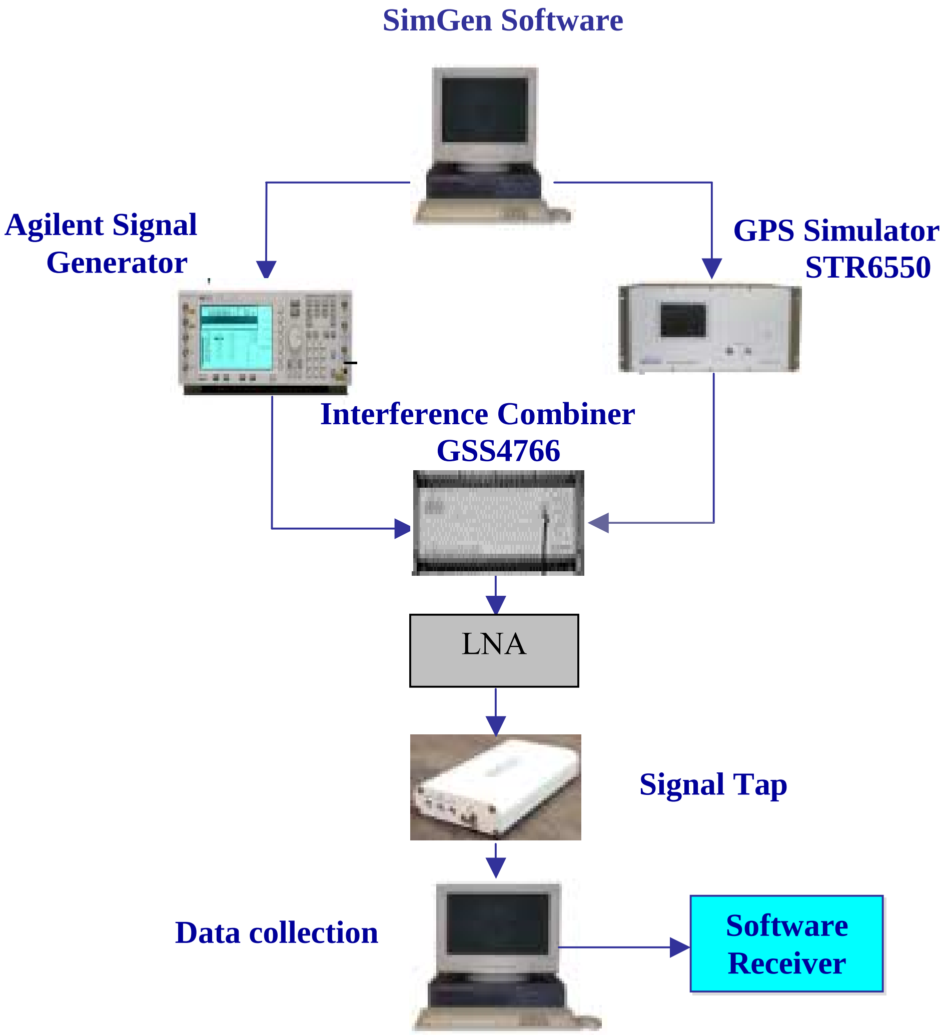

Figure 7.1: Test Configuration

Two kinds of high dynamics are simulated. One is a horizontal motion and the other one is a three dimensional motion. For the first two algorithms, three scenarios are tested based on different noise levels and loop bandwidths. The signal power is configured at -130 dBm to simulate an outdoor environment. The added white noise characteristics are shown in table 7.1

Table 7. 1: Noise Characteristics

Central Frequency	1575.42 MHz
Bandwidth	2 MHz
Power1	-120 dBm
Power2	-90 dBm

The simulated data is 100 seconds long (limited by the Signal Tap). At the beginning, the vehicle stays still for 30 seconds (which makes the acquisition easier) and then takes a low acceleration for 10 seconds. Following that, a high dynamics motion is simulated.

7.1.2 GPS Receiver Configuration

The front end is the GPS Signal Tap, whose block diagram is shown in Figure 7.2. The input GPS signal is amplified, filtered, down-converted, sampled and stored in a flash memory. The flash memory can store 100 seconds of data. The main configuration of the front end (Signal Tap) configuration is given in Table 7.2.

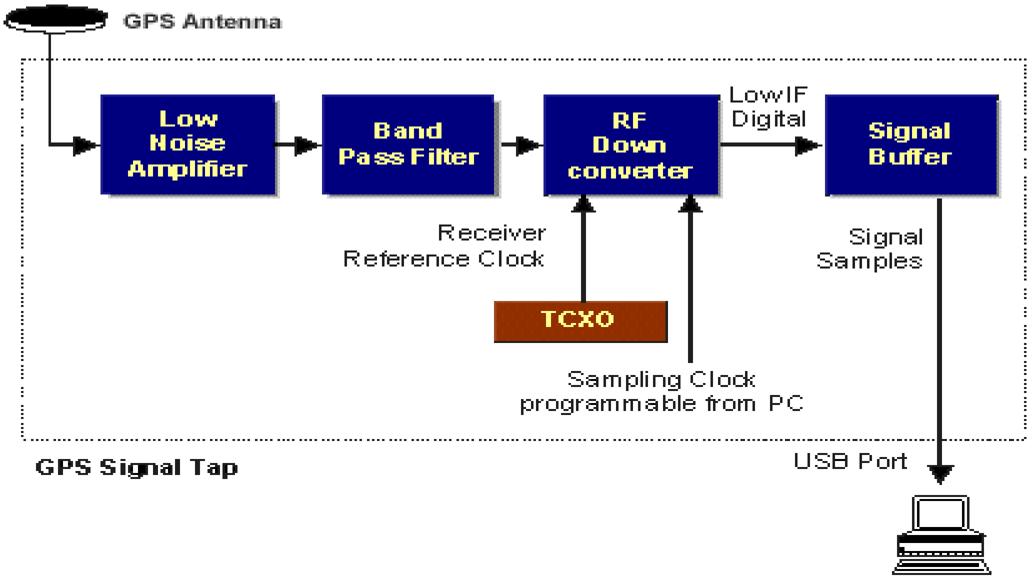

Figure 7.2: The Block Diagram of GPS Front End

Table 7. 2: Signal Tap Configuration

Frequency	GPS L1 at 1575.42 MHz
GPS code	C/A
Sampling Rate	4.75 MHz
IF bandwidth	2 MHz
Signal IF frequency	1.17 MHz
Quantization level	1 bit

The signal processing, which involves the signal acquisition, signal tracking and the computation of the navigation solution is carried out in the software receiver. As mentioned before, this research focuses on the PLL tracking performance. The PLL design scheme and parameters are shown in table 7.3.

Table7. 3: PLL Configuration

Order of the PLL	Three
Discriminator	ARCTAN
PLL Bandwidth	18 Hz, 30 Hz
Pre-integration time	1 ms

7.2 Horizontal Motion Testing, Results and Analysis

The first test is a horizontal motion with 180° turns and lateral acceleration from 3.6 g to 4.6 g. The simulated vehicle trajectory produced by the simulator is shown in Figure 7.3**.**

The simulator does not provide the Doppler frequency directly. However, it supplies the pseudorange rate. From this information, it is very easy to derive the Doppler frequency using Equation 6.4. The derived Doppler frequency for satellite 9 is shown in Figure 7.4 as an example.

In addition, the acceleration and jerk can be derived by making use of the pseudorange rate according to the following relations:

$$a = \frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} \tag{7.1}$$

$$\alpha = \frac{da}{dt} \approx \frac{\Delta a}{\Delta t} \qquad \text{3.6g} \tag{7.2}$$

where a is the acceleration along the line of sight between the satellite and the user, α is the jerk acceleration along the line of sight between the satellite and the user, Δt is the time interval.

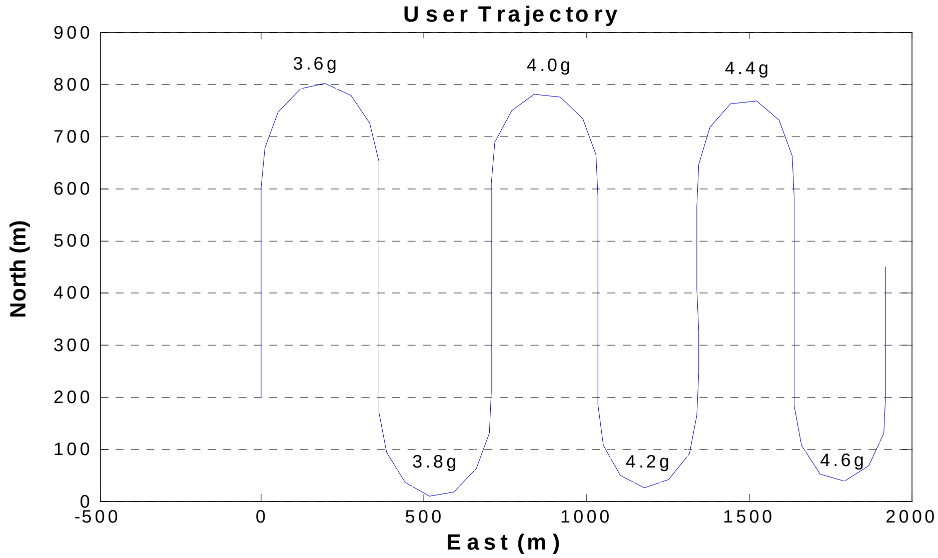

Figure 7.3 Simulated Horizontal Vehicle Trajectory

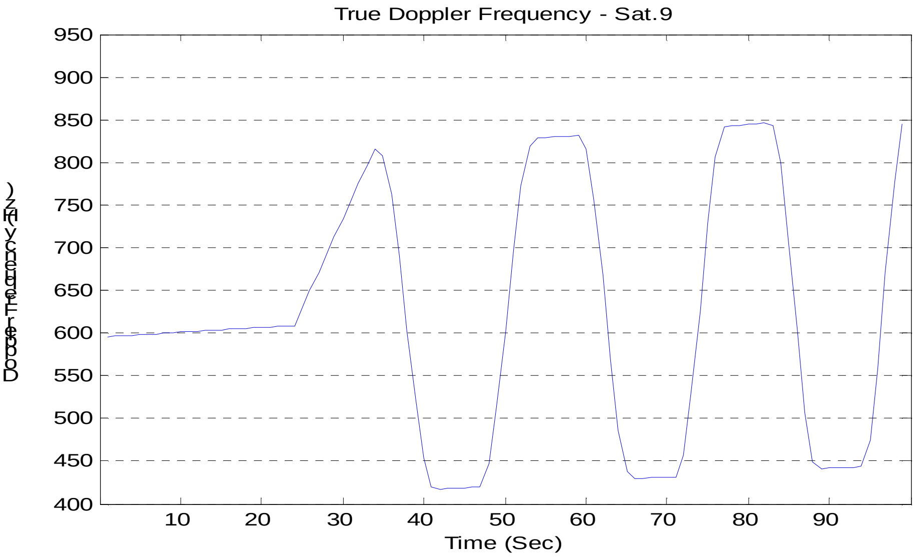

Figure 7.4 Doppler Frequency of Satellite 9

It is obvious that a smaller Δt produces better accuracy. However, the data interval that the simulator gives is 10 milliseconds which is not as small as we require. In the test, the pre-integration time is configured to 1 ms therefore we can obtain the Doppler frequencies output at 1-ms intervals. The Doppler frequency, acceleration and jerk from the simulator output shown herein are not as accurate as desired for a 1 ms interval. However they provide a reference to compare the tracking performance obtained from the software receiver. Figure 7.5, 7.6 and 7.7 show the Doppler frequencies from an ordinary PLL for satellite 9. For the first scenario, the bandwidth of the PLL is 18 Hz and the C/N$_O$ is 45 dB-Hz. For the second scenario, the bandwidth is 18 Hz and the C/N$_O$, 39 dB-Hz. For the third scenario, the bandwidth is 30 Hz and C/N$_O$, 39 dB-Hz.

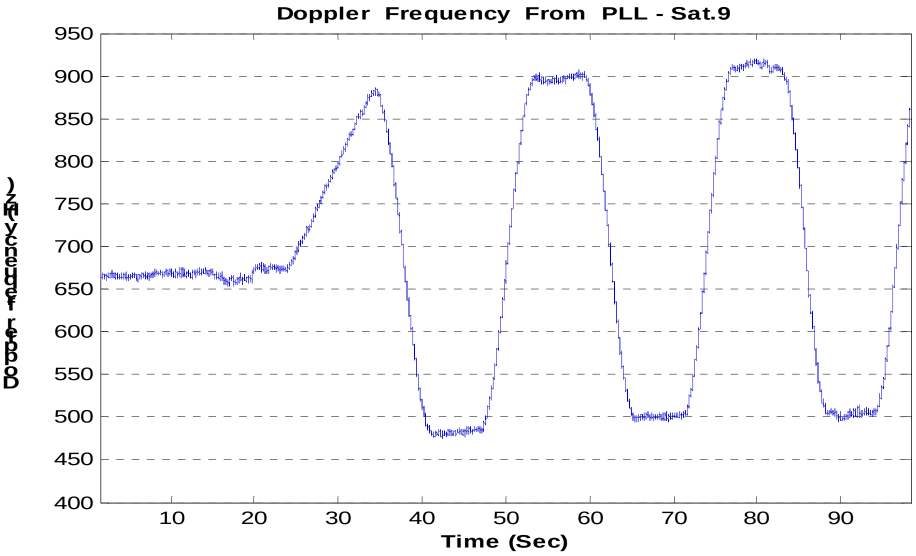

Figure 7.5: Doppler frequency from an ordinary PLL for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

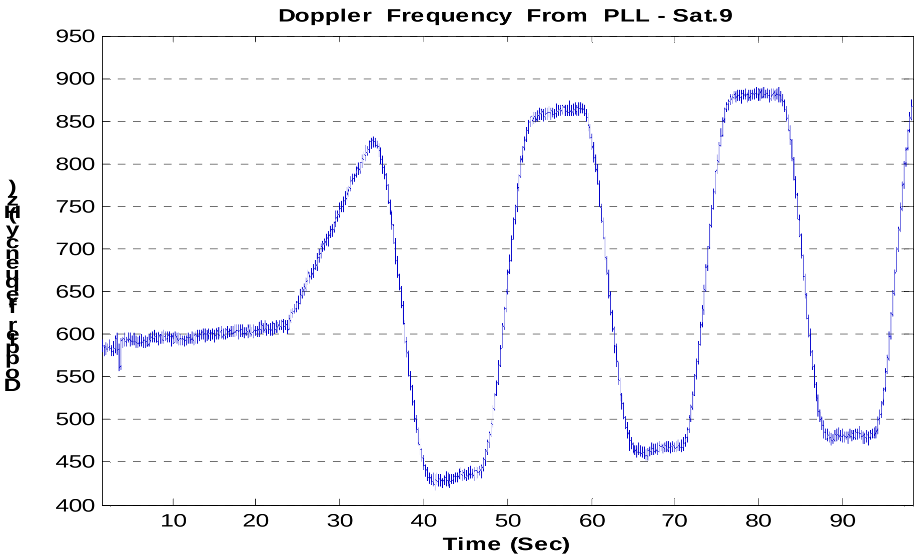

Figure 7.6: Doppler frequency from an ordinary PLL for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

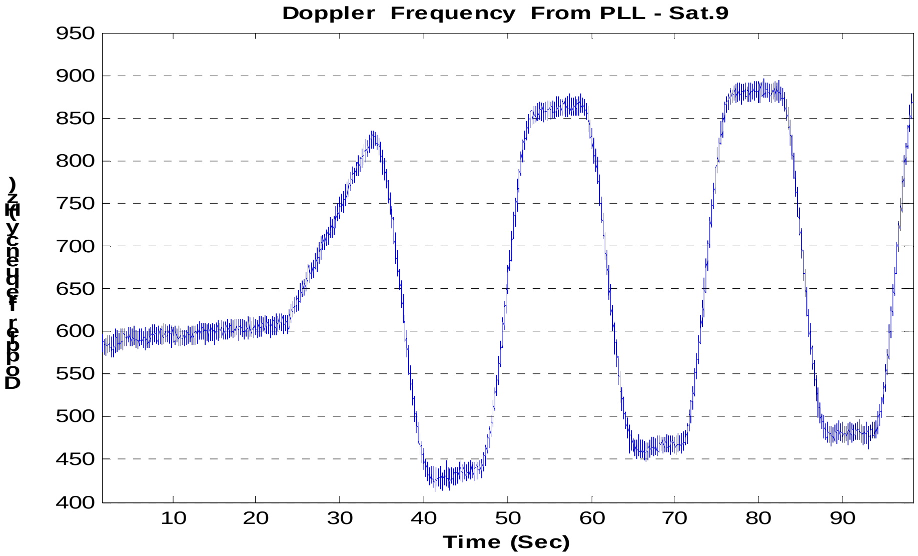

Figure 7.7: Doppler frequency from an ordinary PLL for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

Figure 7.5, 7.6 and 7.7 show that Doppler frequencies from the ordinary PLL get noisier with the increase of the noise power and the bandwidth of the PLL. Further studying

Figure 7.5 and Figure 7.6 and 7.7, it is found that there are Doppler frequency offsets between them. This is caused by the receiver clock's frequency drift. The data used for Figure 7.6 and 7.7 is the same but the data used for Figure 7.5 was collected at a different time. Due to the random frequency drift of the receiver clock, the resulting Doppler frequencies offsets randomly.

7.2.1 Real Time Processing

Real time processing herein means that the results were obtained using previous and present data.

7.2.1.1 Results of Kalman Filter-Based Tracking Algorithm

Three scenarios are used in this test. The Doppler frequency obtained from the Kalman filter-based tracking algorithm for three scenarios are shown in Figure 7.8, 7.9, 7.10. Examining these figures and comparing them with Figure 7.5, 7.6 and 7.7, it is obvious that the results from the Kalman filter-based tracking algorithm are less noisy than the ordinary PLL results. A further study of these figures also leads to the conclusion that, the lower signal-to-noise ratio, the better the tracking performance with the Kalman filter-based algorithm. The reason is that when the measurements are noisy, the Kalman filter-based algorithm puts more weight on its state model than on the measurement. As a result, the estimated Doppler frequency provides better performance than the one from the NCO.

However, if the Kalman filter-based tracking algorithm is used entirely instead of the ordinary PLL for high dynamics applications, especially when the measurements are noisy, the Kalman filter-based tracking algorithm will often diverge, especially for a fast changing trajectory. The problem is caused by the fast and random change in the phase difference, Doppler frequency and change rate of the Doppler frequency during the updating interval which cannot be modelled correctly. Consequently the estimations of these parameters are inaccurate. In this case, an inaccurate Doppler frequency will be used for Doppler frequency removal. Therefore the phase difference measurement from the output of the discriminator can not provide useful information. As a result the divergence occurs. To deal with this problem, this research combined the ordinary PLL with the Kalman filter-based tracking algorithm to provide more reliable tracking. The PLL can guarantee the reliable phase lock even in high dynamic situations if an appropriate bandwidth is chosen. Even though the estimations of Kalman filter are not accurate due to the inaccurate system model, the Kalman filter can correct its estimations using the accurate phase difference measurement. Therefore a reliable tracking is guaranteed.

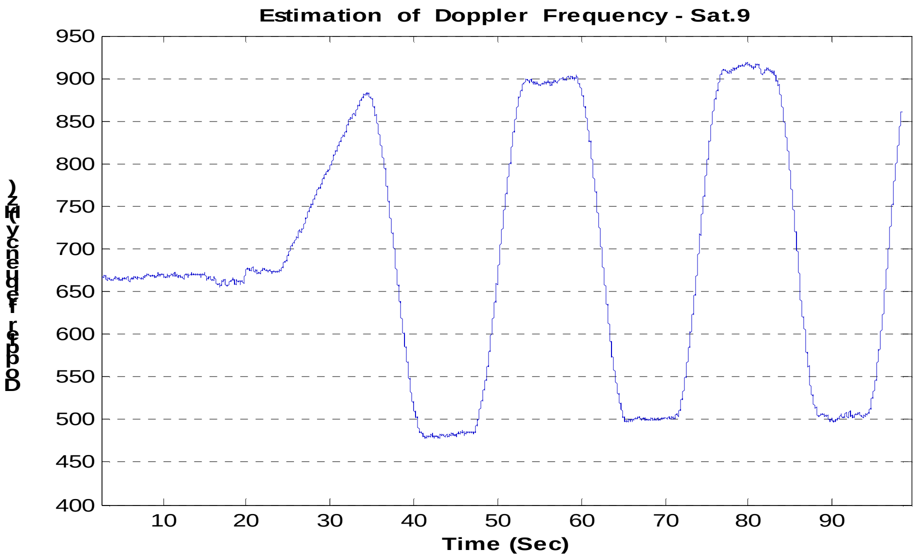

Figure 7.8: Doppler frequency from Kalman filter-based tracking algorithm for satellite 9 (PLL Bandwidth = 18 Hz, C/N_O = 45 dB-Hz)

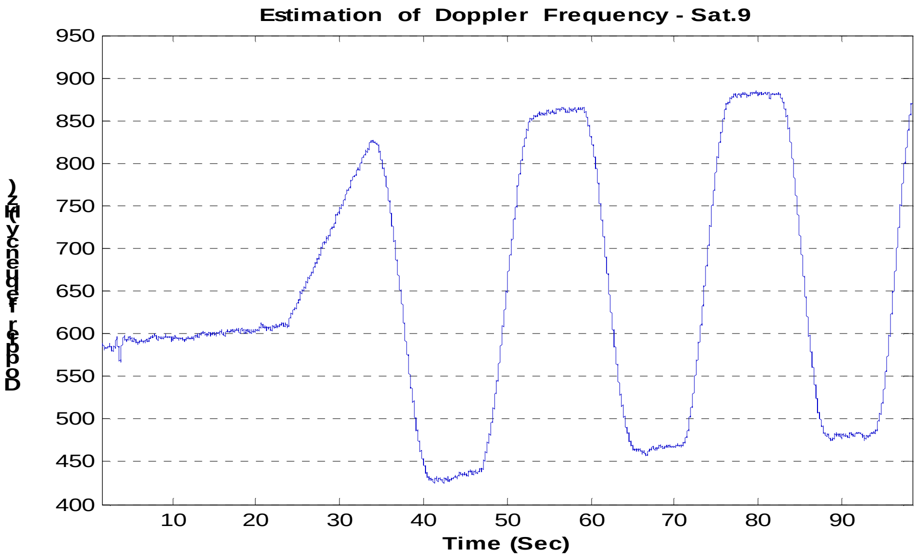

Figure 7.9: Doppler frequency from Kalman filter-based tracking algorithm for satellite 9 (PLL Bandwidth = 18 Hz, C/N_O = 39 dB-Hz)

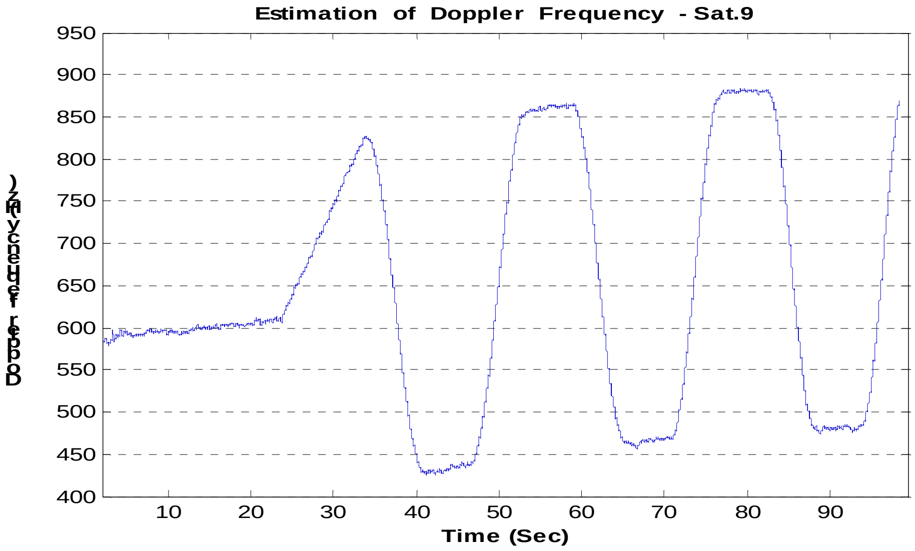

Figure 7.10: Doppler frequency from Kalman filter-based tracking algorithm for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

Figure 7.11, 7.12, 7.13 indicate the PLL Lock status. The Y axis of these figures represents the cosine values of the phase differences. When the PLL is locked, the phase difference should be zero therefore its cosine value should be 1. However due to the noise effect, it is not possible to obtain exactly 1. It is however safe to assume that, if the cosine value is greater than 0.8, the PLL is locked. When carefully examining the output from the PLL Lock Indicator, it is found that sometimes the PLL lost lock. A further investigation reveals that the phase status becomes worse as the noise power increases. However the phase status improves with an increase in bandwidth. The former result is easier to understand. For a given bandwidth, the PLL tracking error increases with the increase of the noise. When the tracking error is larger than the lock threshold, the PLL loses lock. The latter result can be explained in this way: under the high dynamic situations, Doppler frequency changes fast with time that requires PLL adjusts its frequency quickly to catch this change trend. As long as the frequency difference between the incoming signal and the NCO is within the Lock range, the PLL can still lock within one single-beat note between the incoming signal and the NCO frequencies. The lock time is inversely proportional to the PLL bandwidth. With the help of the wider bandwidth, more control signals pass through the loop filter and affect the NCO. Consequently, the NCO adjusts itself much faster to tracking the changes in the incoming signal.

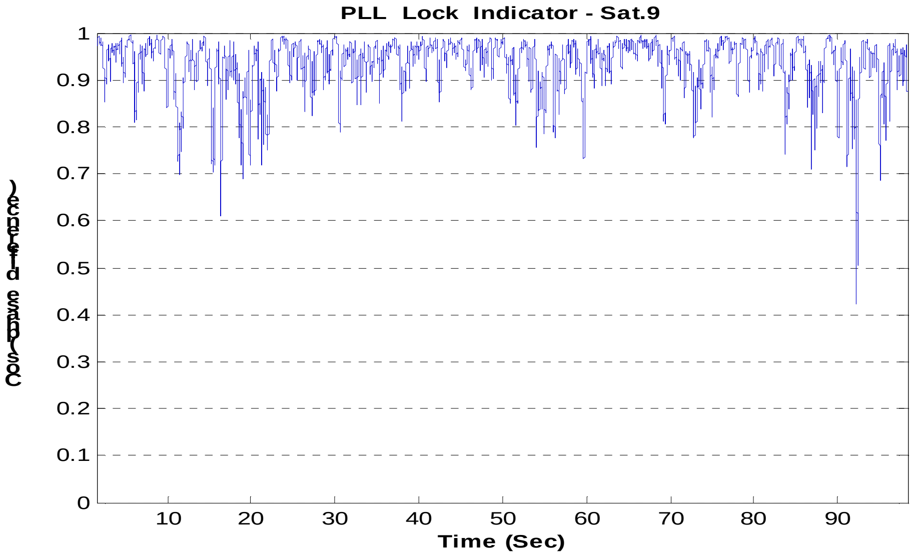

Figure 7.11: PLL lock indicator for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

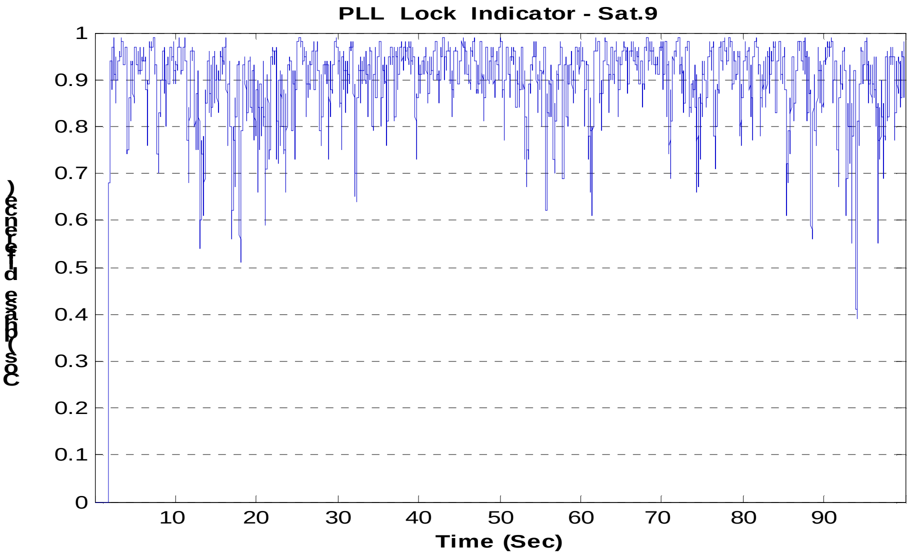

Figure 7.12: The PLL lock indicator for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

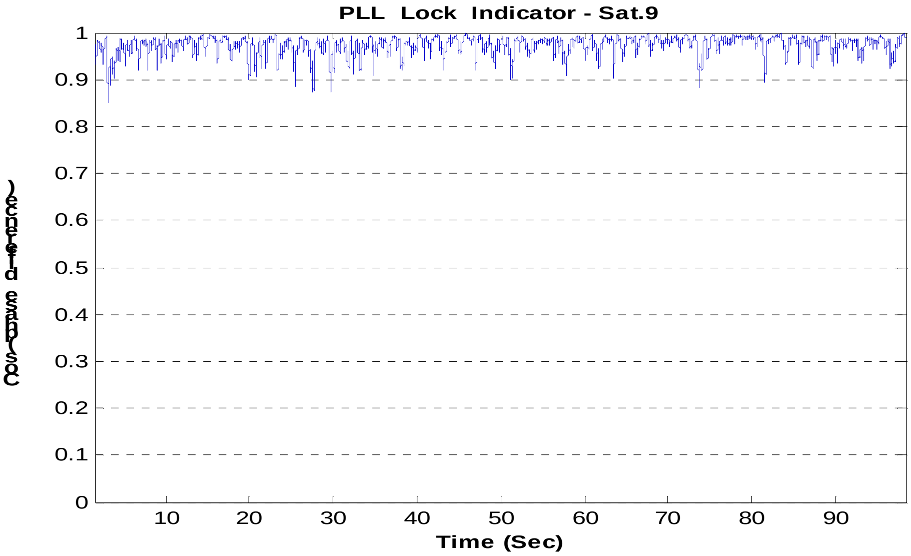

Figure 7.13: The PLL lock indicator for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

7.2.1.2 Results Using the Wavelet De-noising Technique

The wavelet db16 (Daubechies wavelet) and 10 levels decomposition are chosen herein. Figures 7.14, 7.15 and 7.16 illustrate the results from the wavelet de-noising technique. These figures demonstrate improvements compared with Figures 7.5, 7.6 and 7.7. Even though the improvements are significant with the increase of the noise power and bandwidth, the results are poorer when compared with those from the Kalman filter-based tracking algorithm (Figures 7.8, 7.9 and 7.10). It should be noted that the wavelet de-noising technique can only mitigate the noise level but cannot eliminate the noise totally. The remaining noise which is within the bandwidth can still pass through the loop filter and affect the NCO tracking performance.

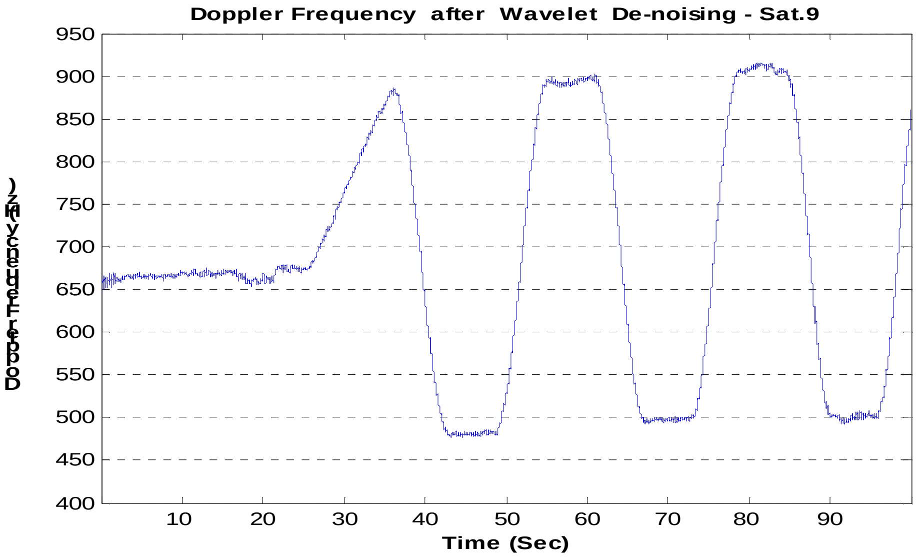

Figure 7.14: Doppler frequency after applying wavelet de-noising technique in the PLL for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

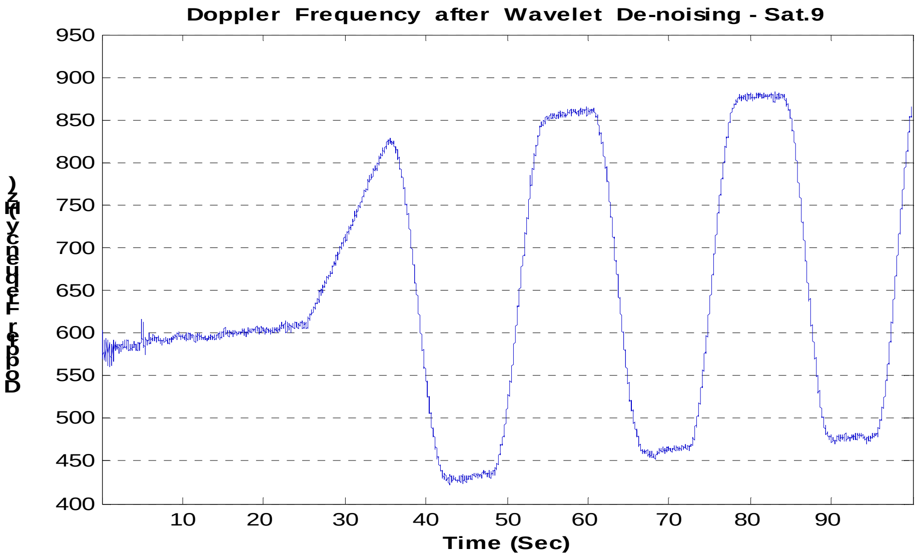

Figure 7.15: Doppler frequency after applying wavelet de-noising technique in the PLL for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

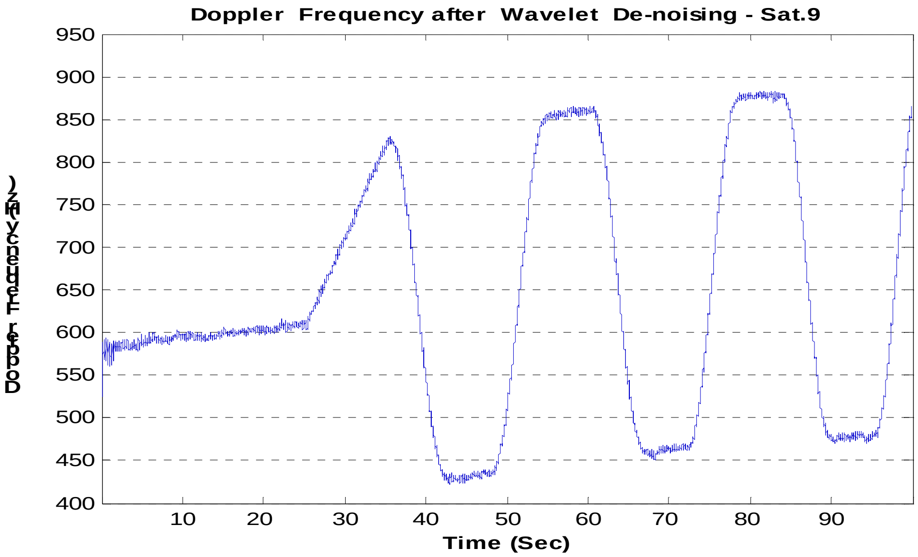

Figure 7.16: Doppler frequency after applying wavelet de-noising technique in the PLL for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

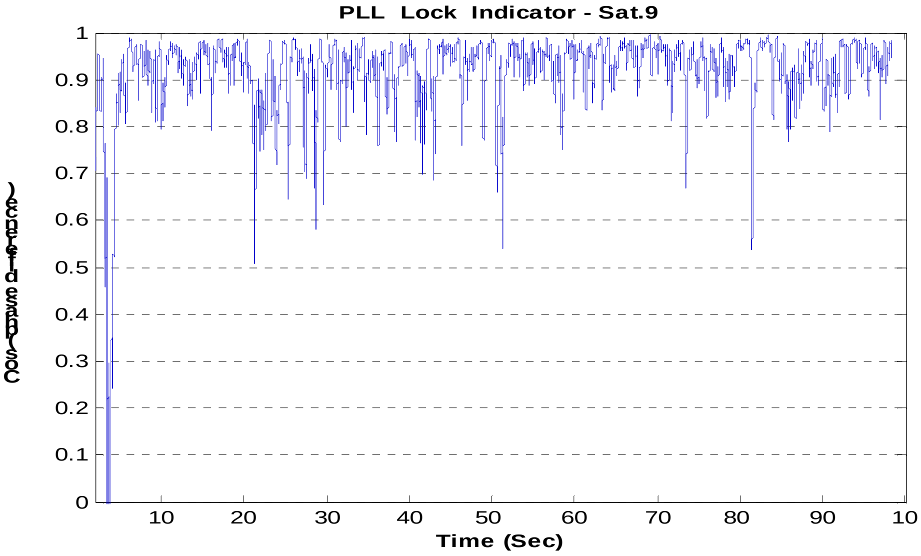

Figure 7.17: The PLL lock indicator for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

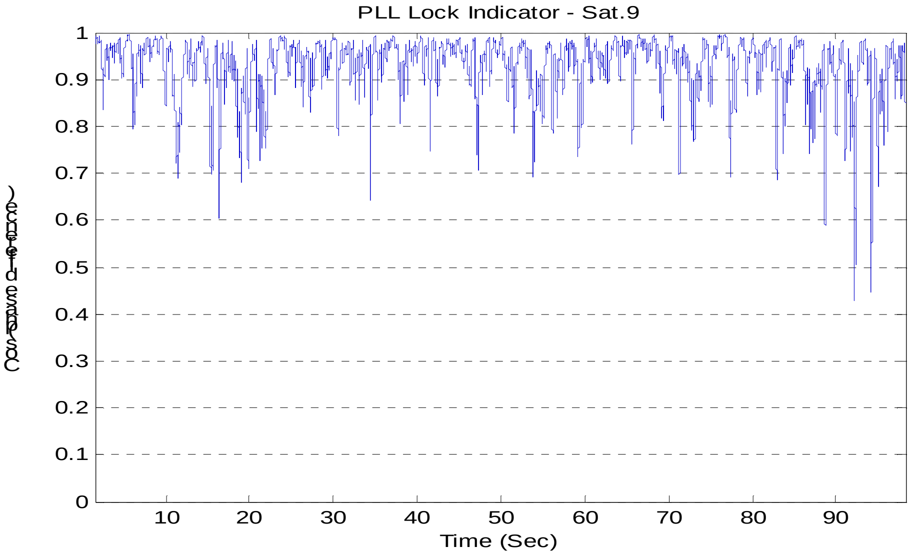

Figure 7.18: The PLL lock indicator for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

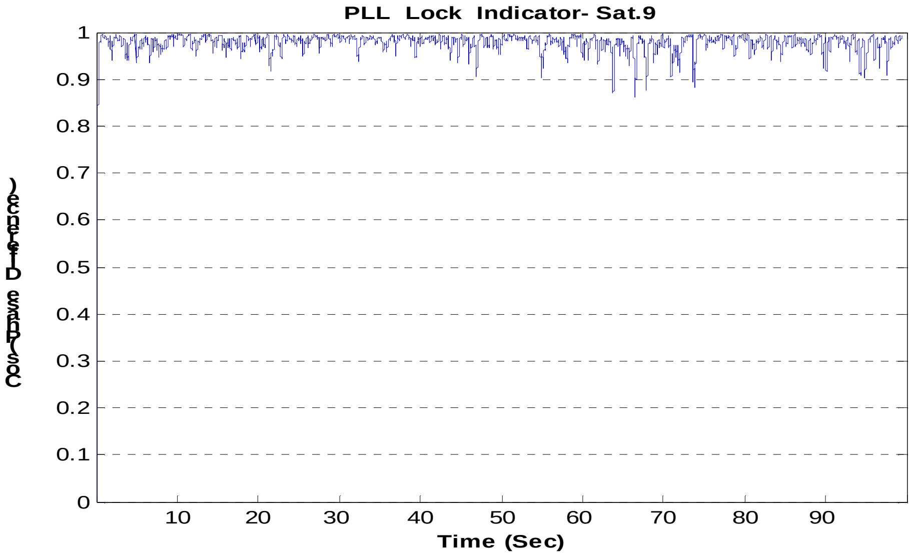

Figure 7.19: The PLL lock indicator for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

Similar to the Kalman filter-based tracking algorithm results, Figure 7.17, 7.18 and 7.19 indicate that the PLL loses lock more frequently with the decrease of the signal-to-noise ratio. The wider bandwidth keeps the PLL locked.

The de-noising performance is related to the selection of the wavelet and decomposition levels. In theory, the decomposition can proceed until the individual detail consists of a single sample. In practice, you will select a suitable number of levels based on the nature of the signal or a suitable criterion such as entropy (The Mathwork, Inc. 2004). In general, the higher the decomposition level, the better the de-noising performance will be. However, after a certain level, the improvement is not very significant. Particularly for real time applications, the higher decomposition level will take longer time to process.

Considering the above factors, 10 decomposition levels are chosen in this thesis. The test results are shown below.

7.2.1.3 Results from Adaptive Bandwidth Algorithm

As discussed in Chapter 6, in order to adjust the PLL bandwidth, the first task is to estimate the dynamics of the incoming signal. Figure 7.20 and 7.22 show the signal acceleration and jerk derived from the pseudorange rate given by the simulator**.** Figure 7.21 and 7.23 show the estimated acceleration and jerk using the adaptive bandwidth algorithm. These figures indicate that the adaptive bandwidth algorithm can estimate the incoming signals dynamics. However, the estimated acceleration and jerk are noisy. Ignoring the clock errors (caused by vibration and Allan deviation), the PLL bandwidth is adapted according to the acceleration (second order loop) and jerk (third order loop) and signal-to-noise ratio. The noisy dynamics estimation decreases the accuracy of the adaptive bandwidth.

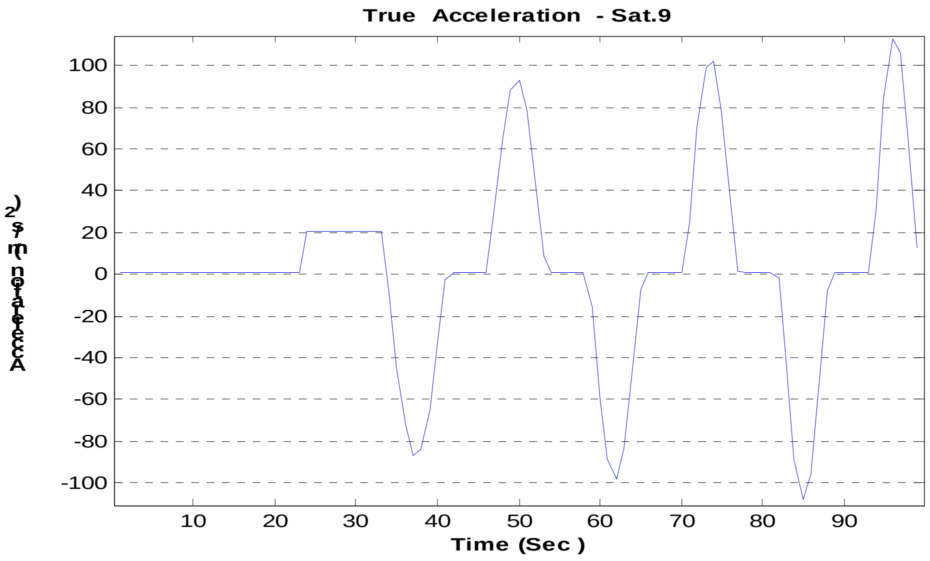

Figure 7.20: True acceleration from the simulator for satellite 9

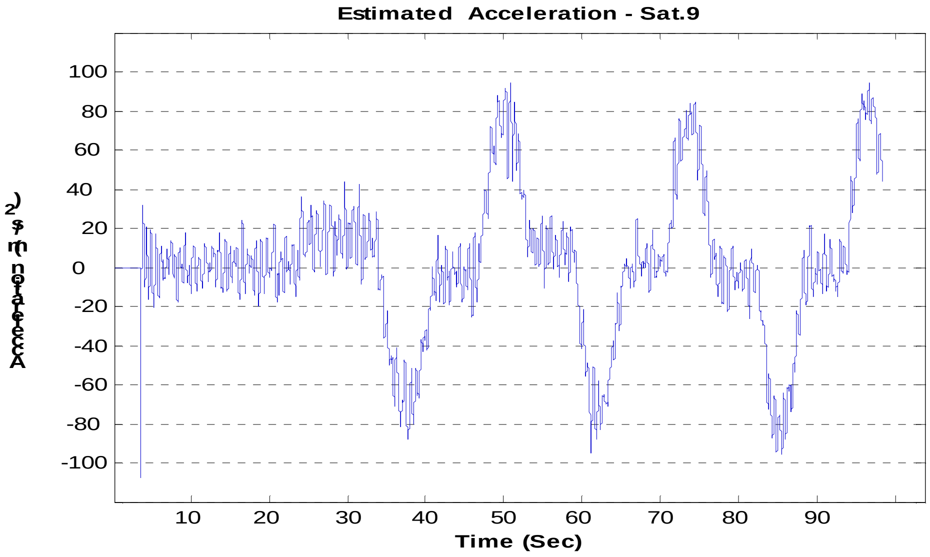

Figure 7.21: Estimated acceleration using the adaptive bandwidth algorithm for satellite 9

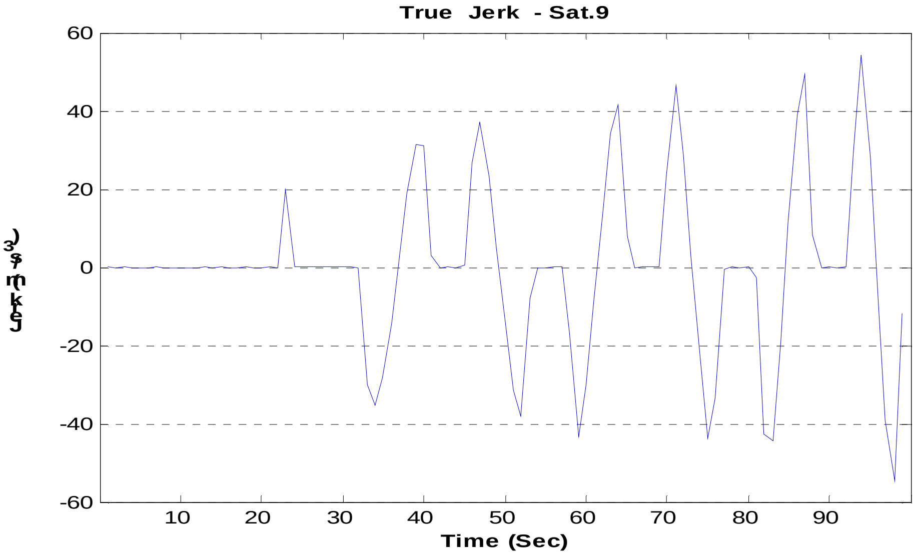

Figure 7.22: True jerk from the simulator for satellite 9

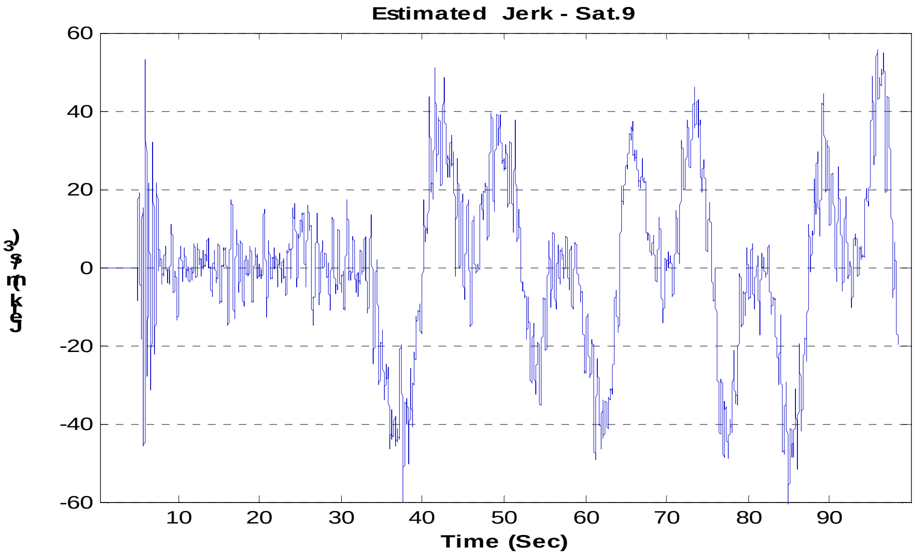

Figure 7.23: Estimated jerk using the adaptive bandwidth algorithm for satellite 9

Figure 7.24 shows the Doppler frequency after adapting the bandwidth for satellite 9 and when the C/N_O is 45 dB-Hz. A comparison of Figure 7.24 and 7.5 shows the improvement. Figure 7.25 shows the adaptive bandwidth which changes with the estimated dynamics. Figure 7.26 shows the PLL lock indicator results. The figure demonstrates that sometimes the PLL still loses lock even though the bandwidth is adaptable. There are two factors that affect this phenomenon. First when computing the adaptive bandwidth, the clock effects are not taken into account. The resulting bandwidth is not as accurate as it should be. Secondly under high dynamics situations, the bandwidth must change fast to catch up with the fast variations in the incoming signal. However the PLL does not allow its bandwidth to change quickly, otherwise the loop filter will output impulses which will cause the PLL to lose lock (Legrand & Macabiau 2000).

A simple method to deal with this problem is to insert a low-pass filter which will delay the change rate of the PLL bandwidth. The response time of the low-pass filter is critical for the abrupt variation in an incoming signal's jerk which requires fast response in the bandwidth. However, too short a response time can cause the loop to lose lock. This is a major problem for the adaptive bandwidth algorithm.

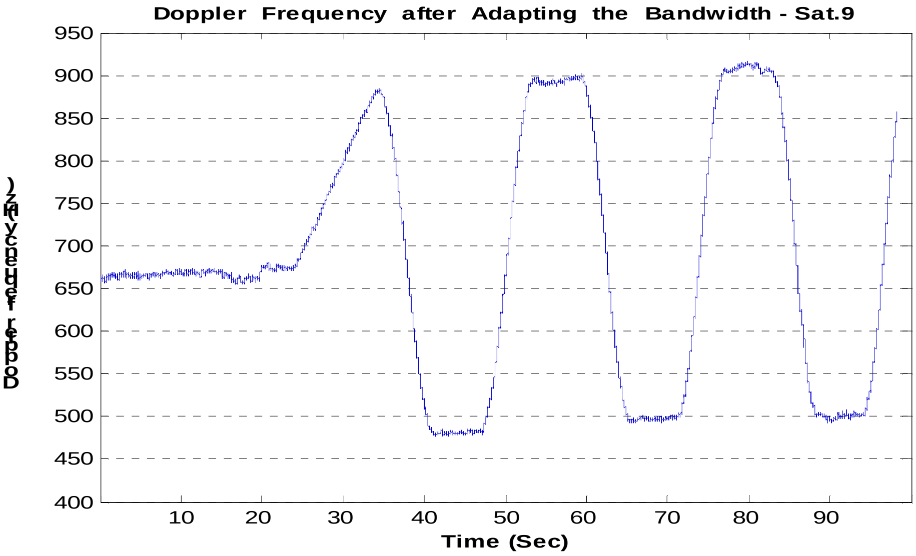

Figure 7.24: Doppler frequency after adapting the bandwidth for satellite 9 (C/N$_O$ is 45 dB-Hz)

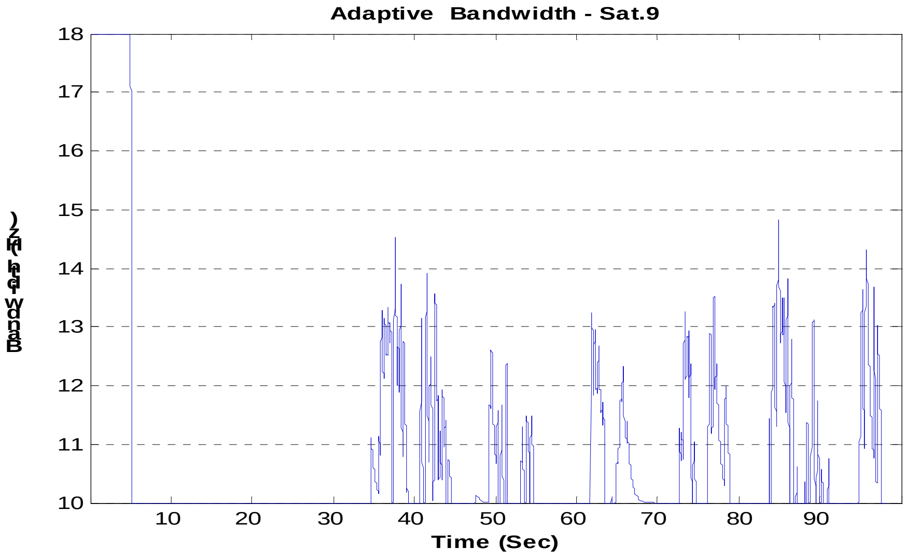

Figure 7.25: Adaptive bandwidth for satellite 9

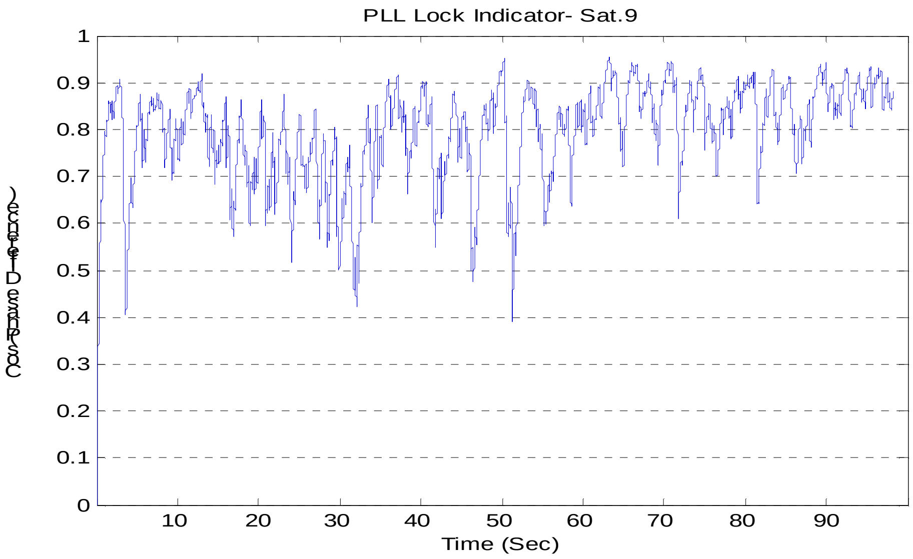

Figure 7.26: PLL lock indicator for satellite 9

7.2.2 Post Processing

In this thesis, all results from the three algorithms were processed again using the wavelet de-noising technique. For this mode, all data involves in the de-noising processing. This means except the last moment result, all other moment results are derived using the previous, present and future information. The results indicate that better results are obtained in post processing compared to real time processing.

7.2.2.1 Results from the Kalman filter-based tracking algorithm

The post processed results for the Kalman filter-based tracking algorithm are shown in Figure 7.27, 7.28 and 7.29. The results clearly indicate an improvement, particularly for the high noise level and wide bandwidth case.

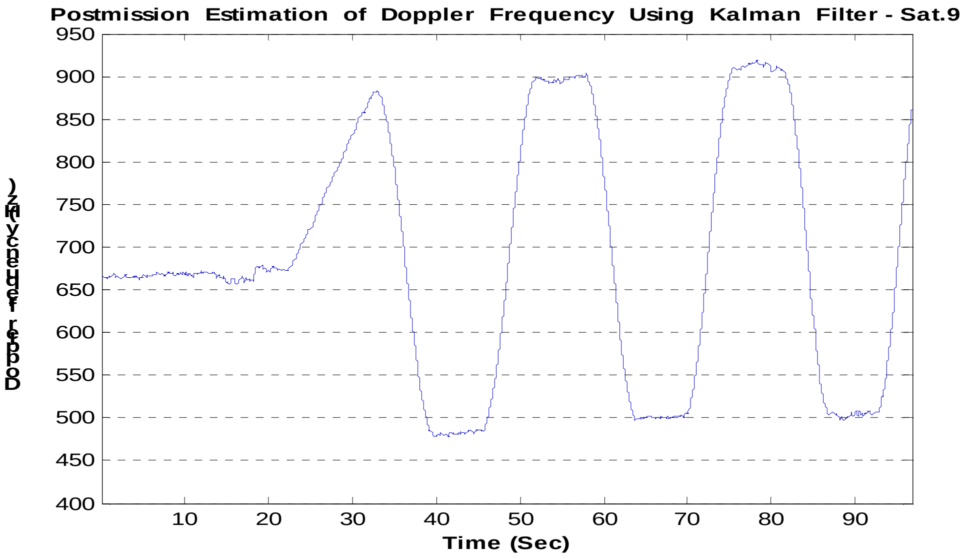

Figure 7.27: Doppler frequency from Kalman filter based tracking algorithm for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

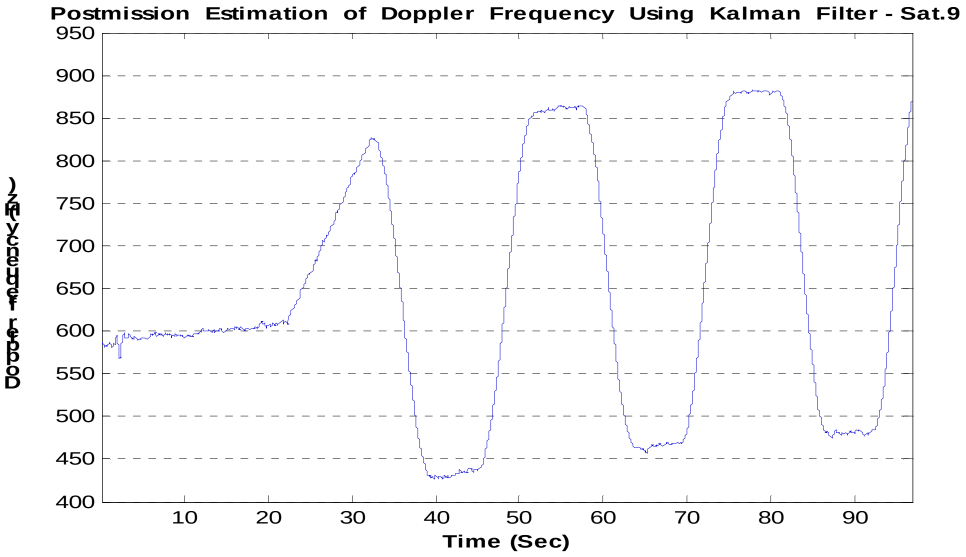

Figure 7.28: Doppler frequency from Kalman filter-based tracking algorithm for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

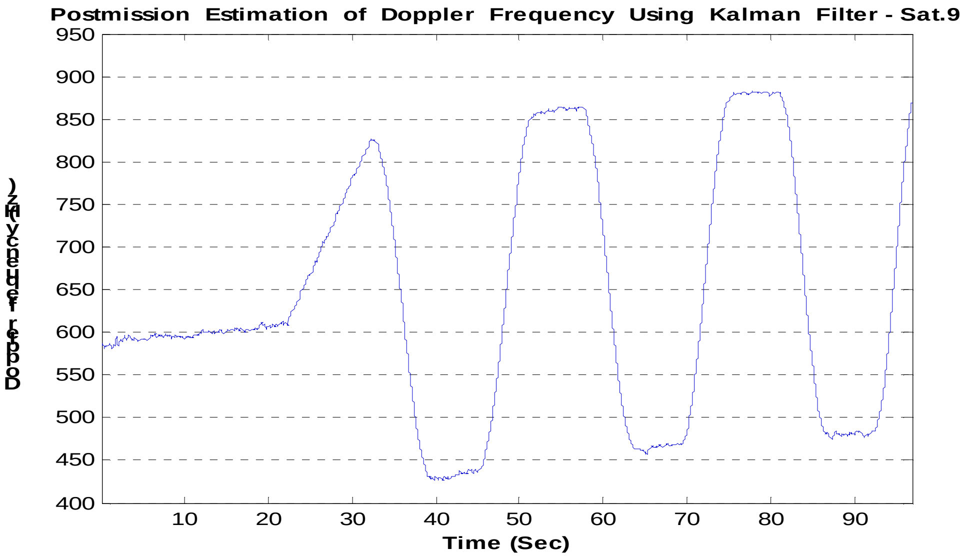

Figure 7.29: Doppler frequency from Kalman filter-based tracking algorithm for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

7.2.2.2 Results from applying the wavelet de-noising technique in PLL

The post processed results, after applying the wavelet de-noising technique, are shown in Figure 7.30, 7.31 and 7.32. The results show that the improvement is not as obvious as with the Kalman filter. However compared with the real time results reported earlier in Figure 7.14, 7.15 and 7.16, these post-processed results do provide some improvements.

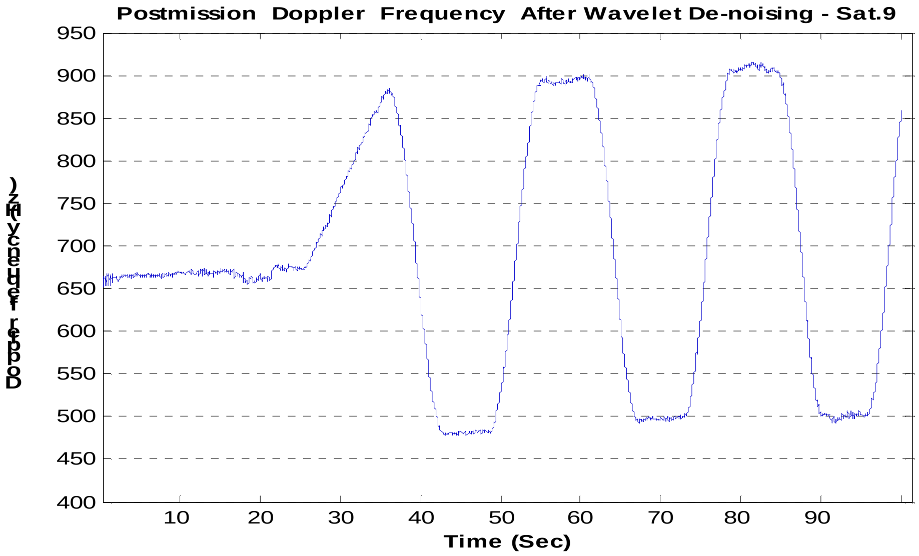

Figure 7.30: Doppler frequency after applying wavelet de-noising technique for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

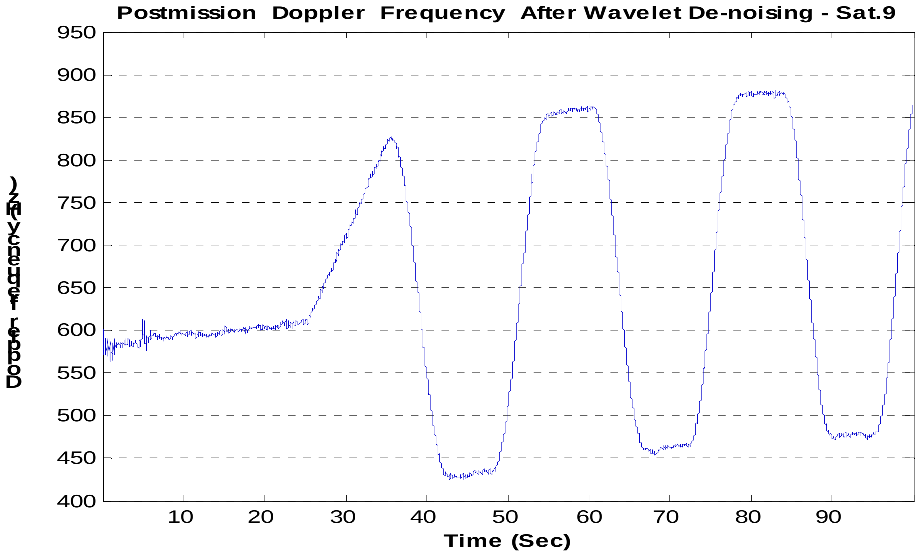

Figure 7.31: Doppler frequency after applying wavelet de-noising technique for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

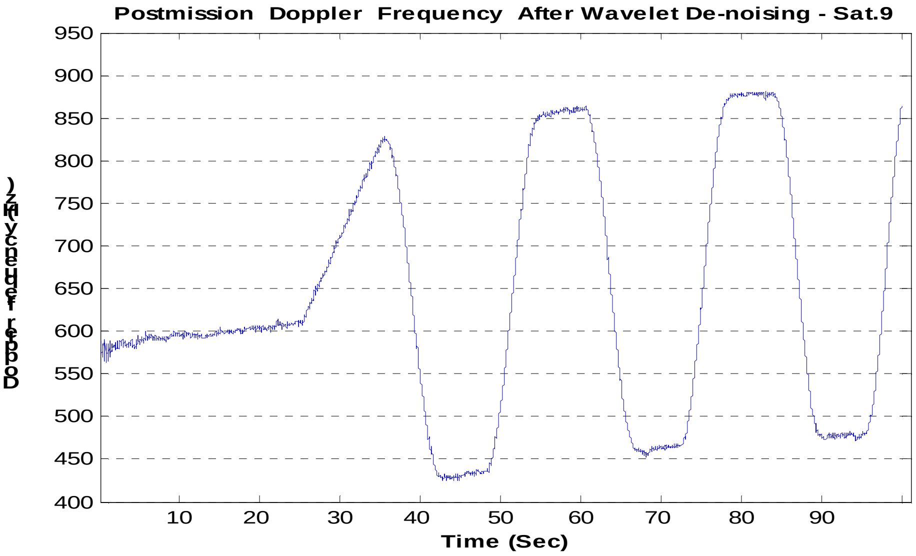

Figure 7.32: Doppler frequency after applying wavelet de-noising technique for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

7.2.2.3 Results from Adaptive Bandwidth Algorithm

The post processed result for the adaptive bandwidth algorithm is shown in Figure 7.33. Compared to the real time result of Figure 7.24, the improvement is significant.

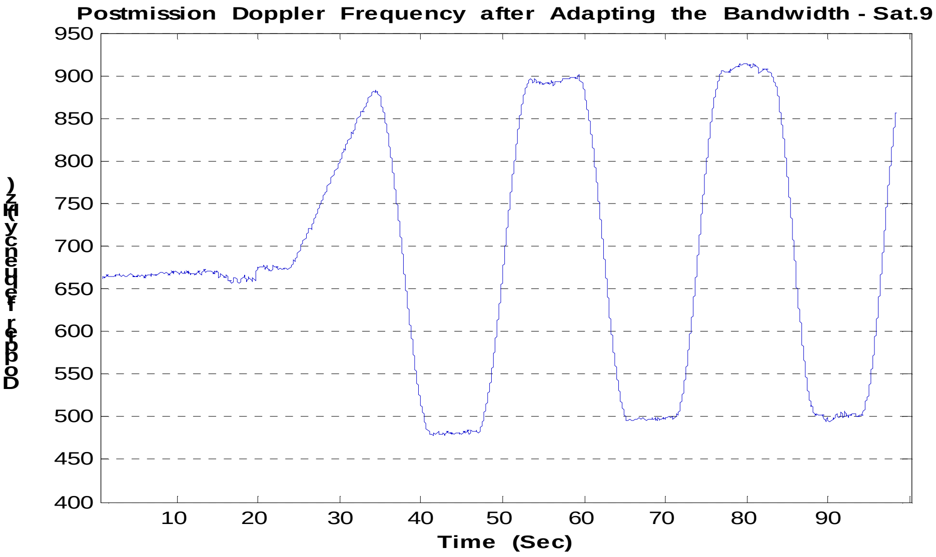

Figure 7.33: Doppler frequency after adapting the PLL bandwidth for satellite 9 (C/N_O is 45 dB-Hz)

7.3 3-Dimensional Motion Test, Results and Analysis

In this test, a three dimensional motion with a 360° turn and a 3.8 g lateral acceleration in the horizontal plane and simultaneously, 5 m/s velocity in the height direction, is simulated. The horizontal and 3-D trajectories are shown in Figure 7.34 and 7.35.

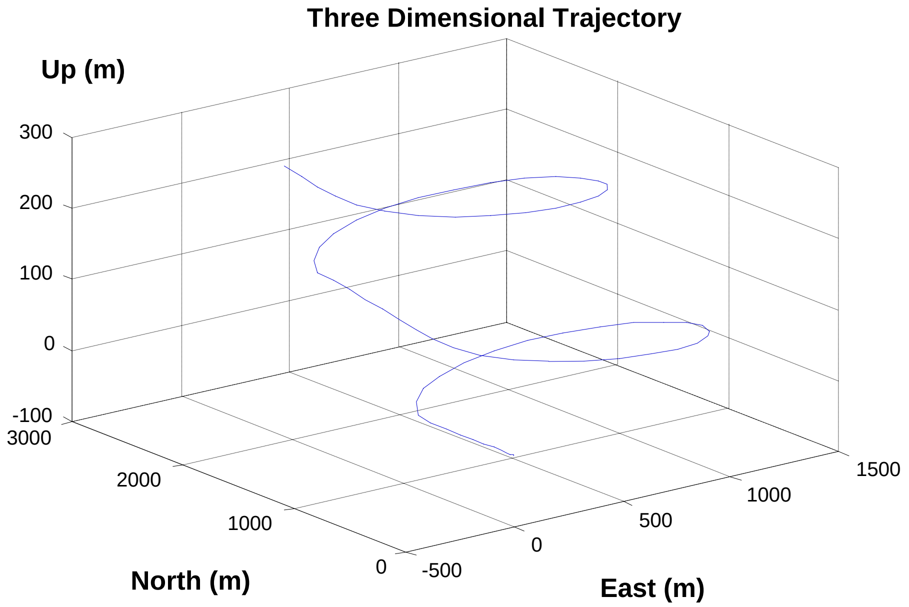

Figure 7.34: 3-Dimensional trajectory 360º turns and 3.8 g lateral acceleration in horizontal plane and 5 m/s velocity in height direction.

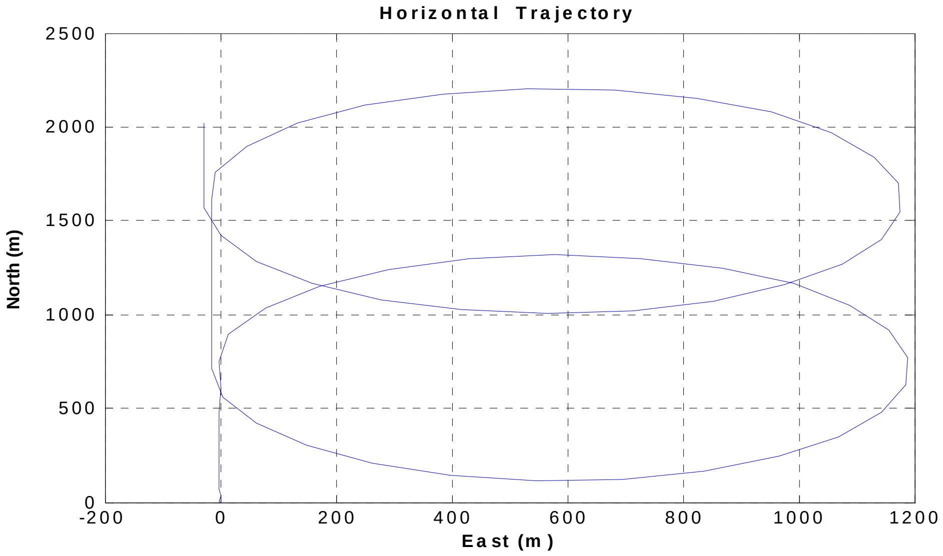

Figure 7.35: Horizontal trajectory with 360° turns and 3.8 g lateral acceleration

Similarly, the Doppler frequency can be derived from the pseudorange rate given by the simulator. That for satellite 9 is shown in Figure 7.36.

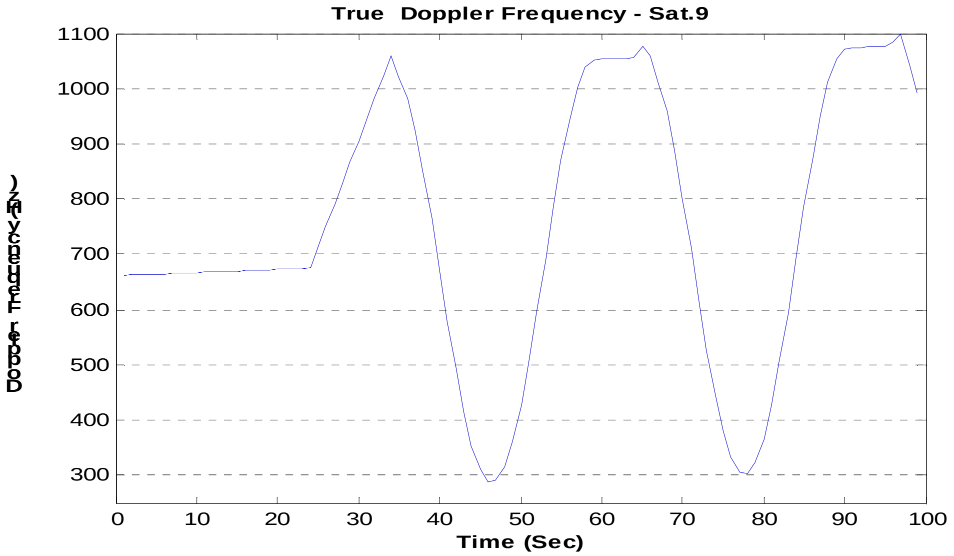

Figure 7.36: Doppler frequency from the simulator for satellite 9

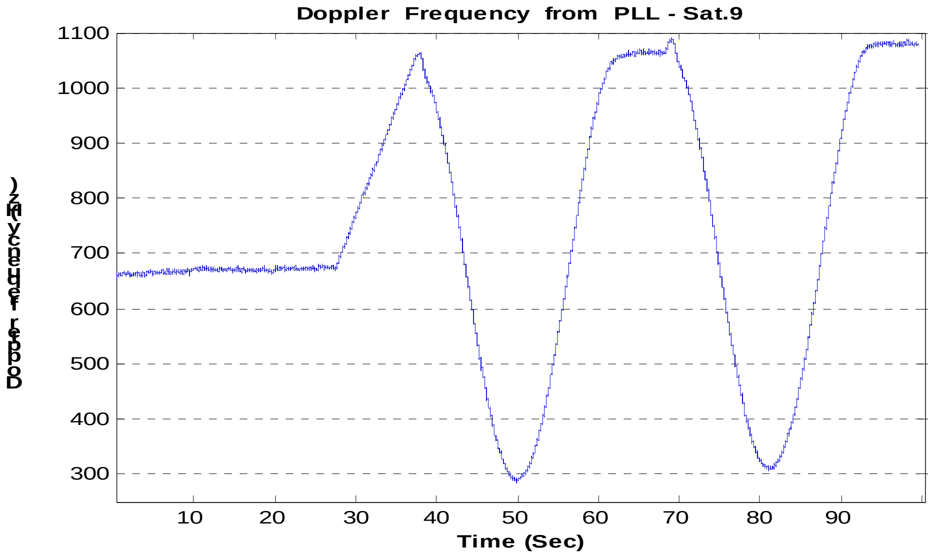

Figure 7.37: Doppler frequency from an ordinary PLL for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

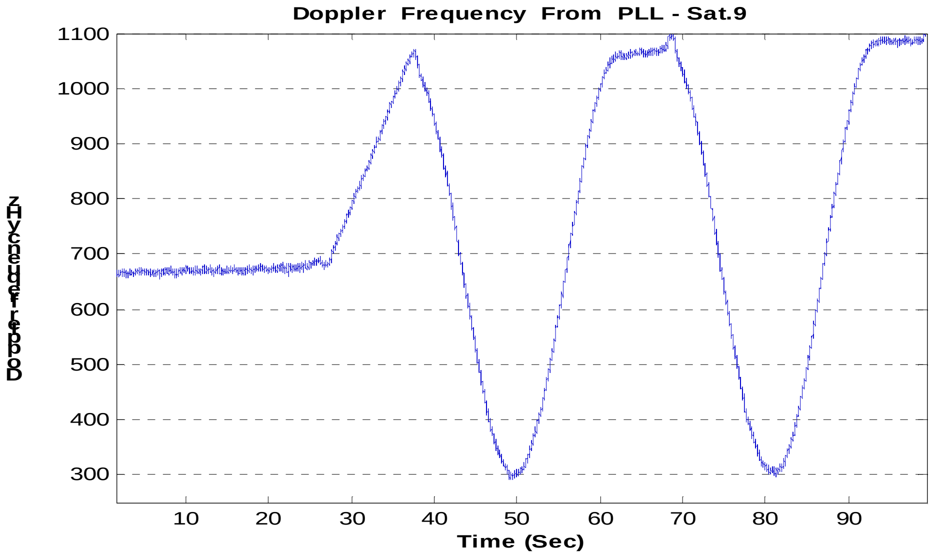

Figure 7.38: Doppler frequency from an ordinary the PLL for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

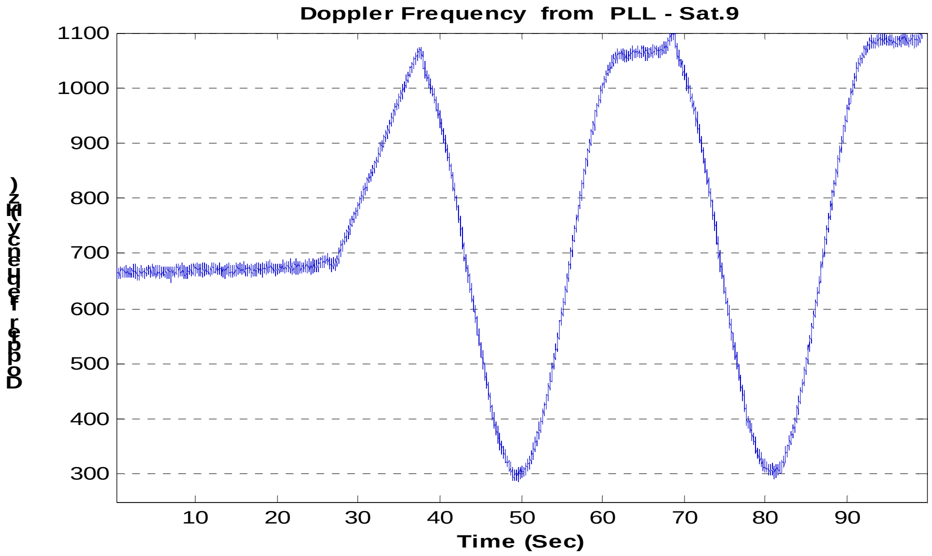

Figure 7.39: Doppler frequency from an ordinary the PLL for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

Again, the test is carried out for three scenarios. The Doppler frequencies from the ordinary PLL for these three scenarios are shown in Figure 7.37, 7.38 and 7.39. Similar to the test 1, Doppler frequencies become very noisy with the increase of the noise power and the loop bandwidth.

7.3.1 Real Time Processing

7.3.1.1 Results from the Kalman Filter-Based Tracking Algorithm

Figure 7.40, 7.41 and 7.42 show the Doppler frequencies from the Kalman filtered-based tracking algorithm and Figure 7.43, 7.44 and 7.45, their corresponding PLL lock indicators. The maximum jerk in this test is higher than the one in the horizontal motion. However, due to the PLL existing, the Doppler frequency from the Kalman filter-based tracking algorithm can still track the input Doppler frequency correctly. Even though the PLL sometimes lose phase lock, as long as it keeps frequency tracking, the Kalman filter-based tracking algorithm can still obtain the correct phase different measurements from the output of the PLL discriminator. This ensures accurate Doppler frequency measurements, a process which is essential for carrier phase measurements.

The same phenomena exist for this test. As the noise power increases, the Doppler tracking performance does not deteriorate very much. This also applies to the scenario of a 30 Hz PLL bandwidth. Obviously the Doppler frequency from the ordinary PLL is much noisier due to a wider bandwidth. However, the Kalman filter-based tracking algorithm generates almost the same result as for the 18 Hz PLL bandwidth case. This leads to the conclusion that the Kalman filter-based tracking algorithm still behaves well in low signal-to-noise ratio and wider bandwidth situations.

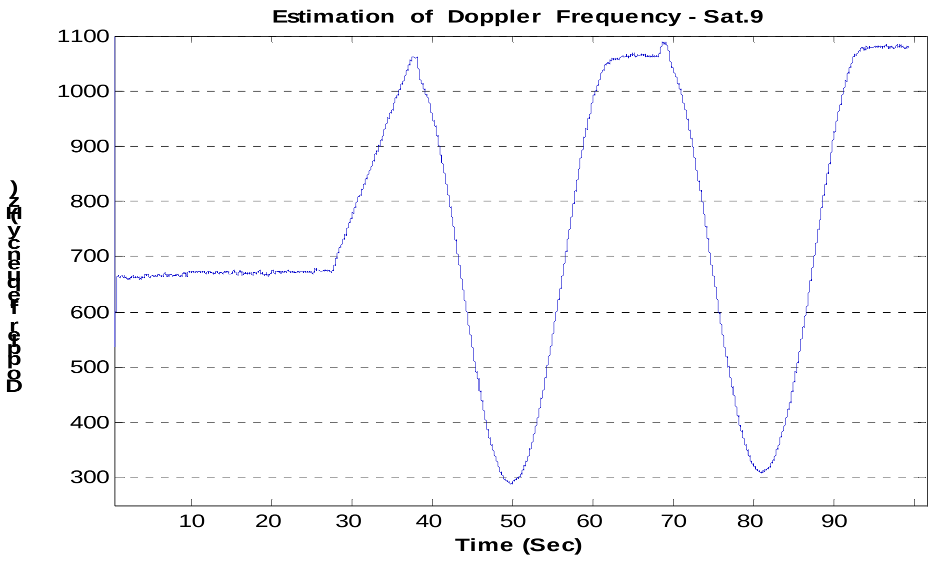

Figure 7.40: Doppler frequency from the Kalman filter-based tracking algorithm for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

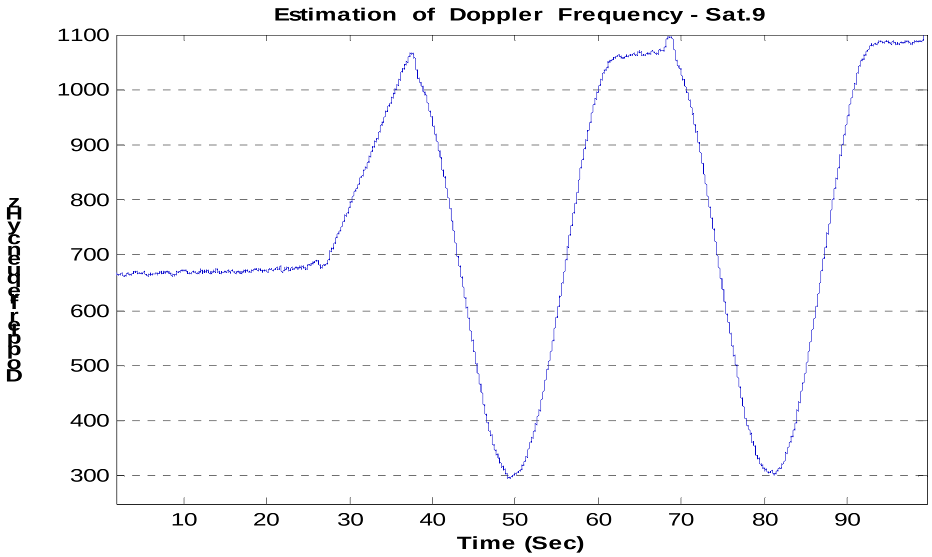

Figure 7.41: Doppler frequency from the Kalman filter-based tracking algorithm for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

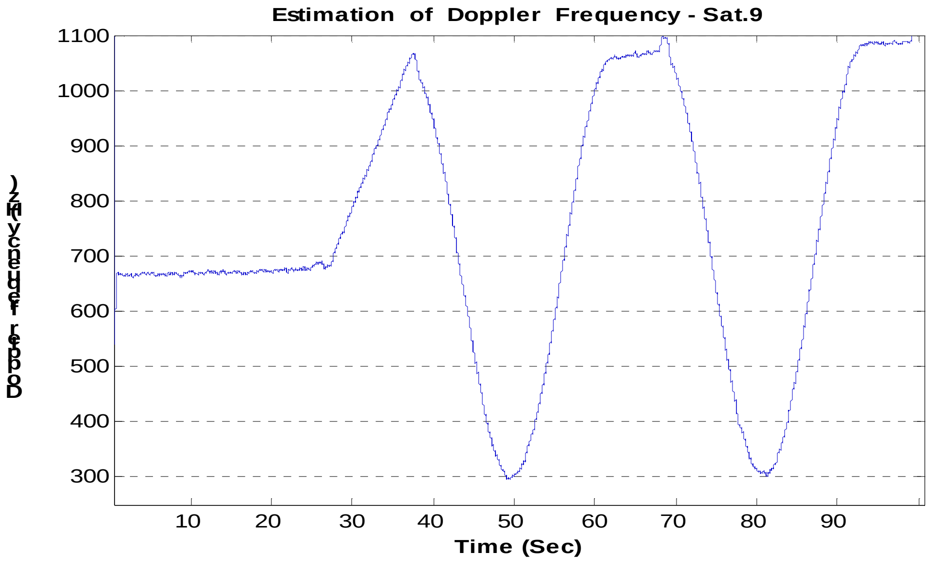

Figure 7.42: Doppler frequency from the Kalman filter-based tracking algorithm for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

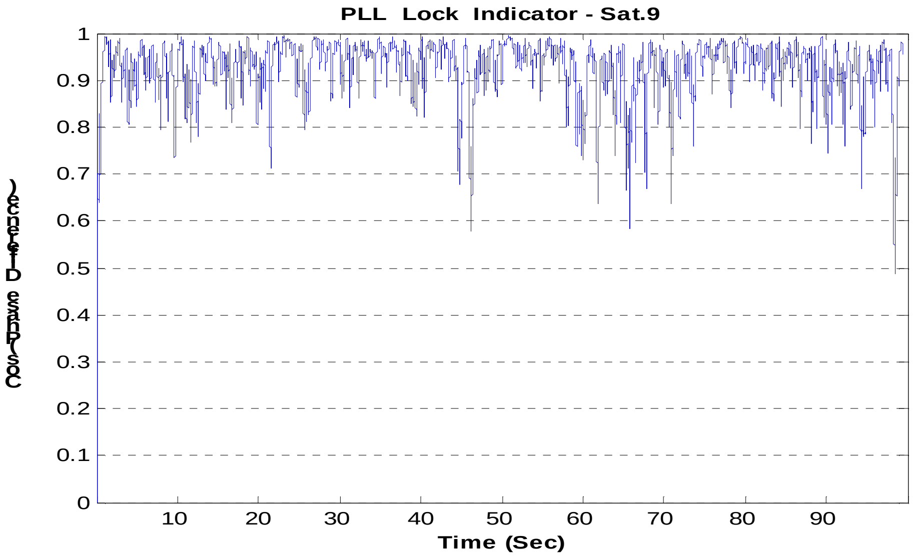

Figure 7.43: The PLL lock indicator for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

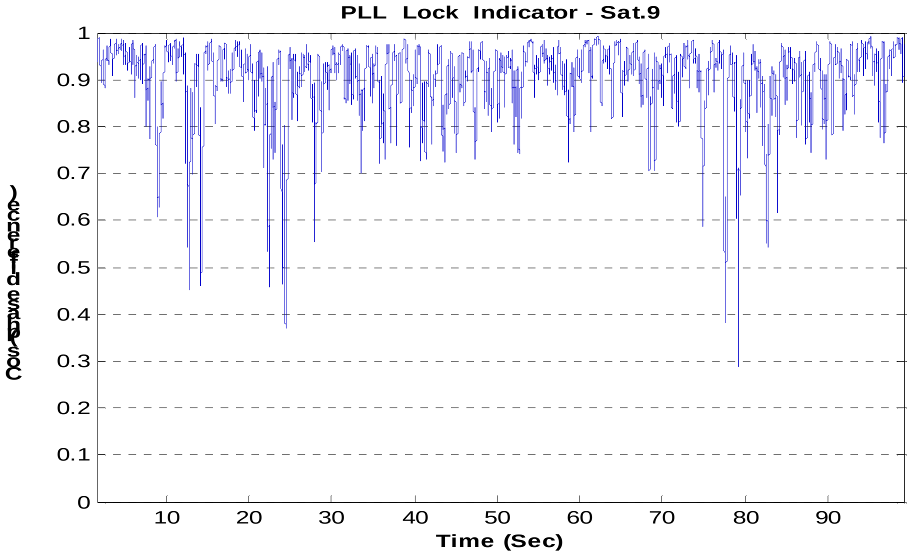

Figure 7.44: The PLL lock indicator for satellite 9 (PLL Bandwidth = 18 Hz, C/N_O = 39 dB-Hz)

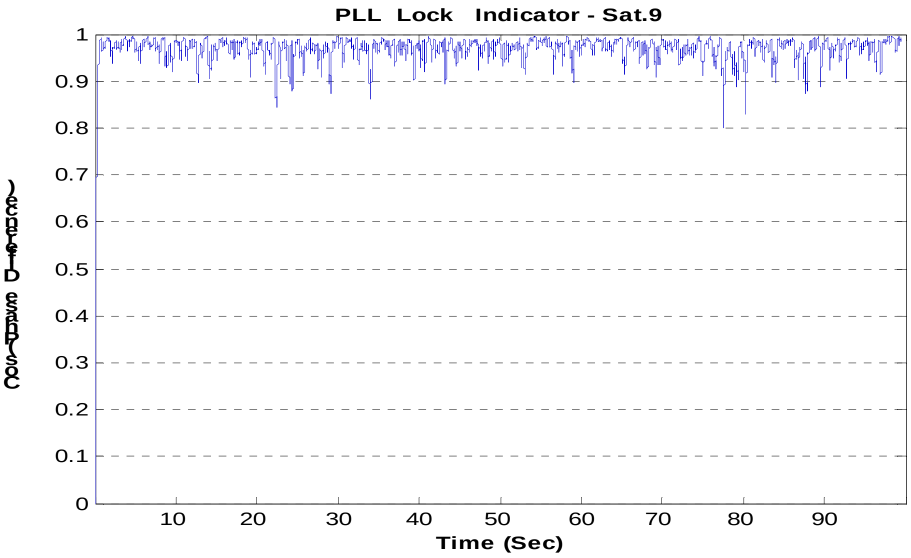

Figure 7.45: The PLL lock indicator for satellite 9 (PLL Bandwidth = 30 Hz, C/N_O = 39 dB-Hz)

7.3.1.2 Results with the Wavelet De-noising Technique

Figures 7.46, 7.47 and 7.48 show the Doppler frequencies when applying wavelet de-noising technique in the PLL and Figures 7.49, 7.50 and 7.51 their corresponding PLL lock indicators. Again, these figures show that the PLL tracking performance improve significantly with an increase of the noise power and bandwidth and the PLL more easily loses phase lock when the noise power increases and the bandwidth narrows.

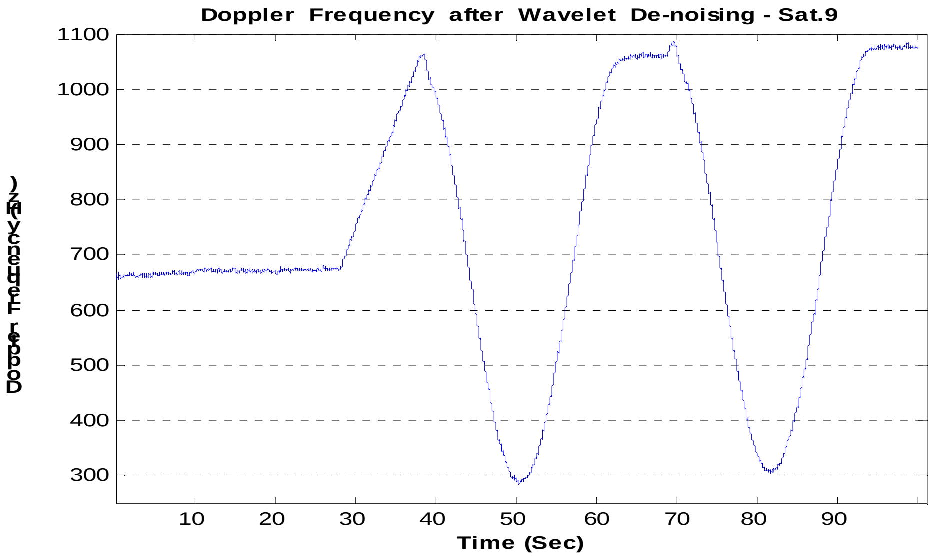

Figure 7.46: Doppler frequency after applying wavelet de-noising technique in PLL for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

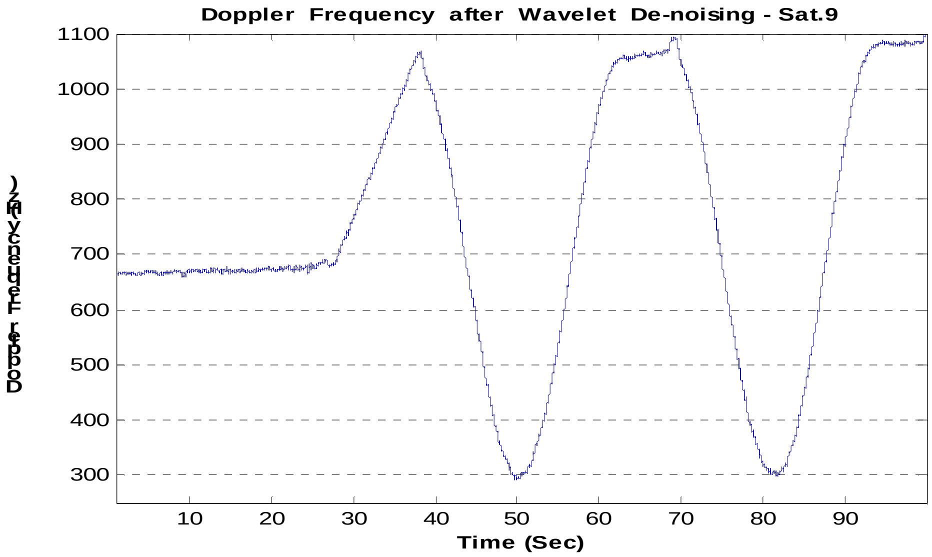

Figure 7.47: Doppler frequency after applying wavelet de-noising technique in PLL for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

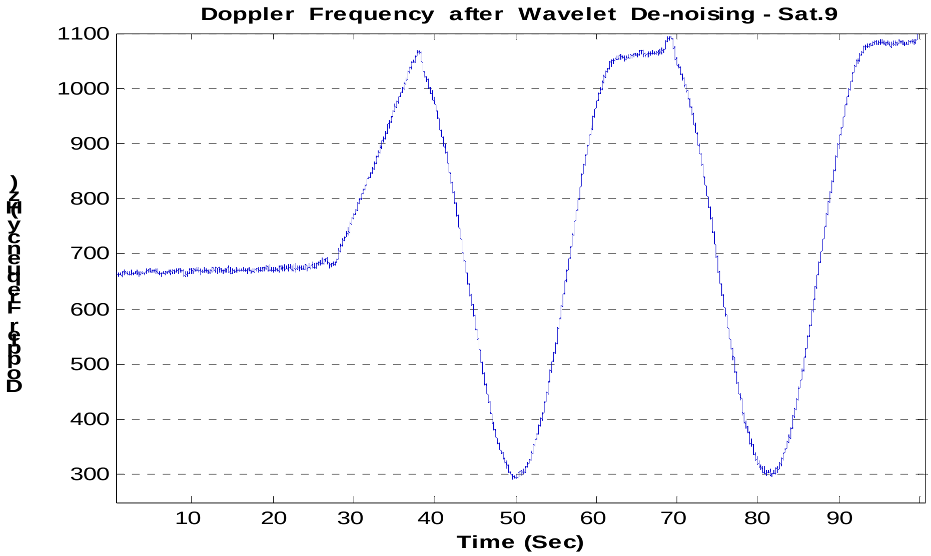

Figure 7.48: Doppler frequency after applying wavelet de-noising technique in PLL for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

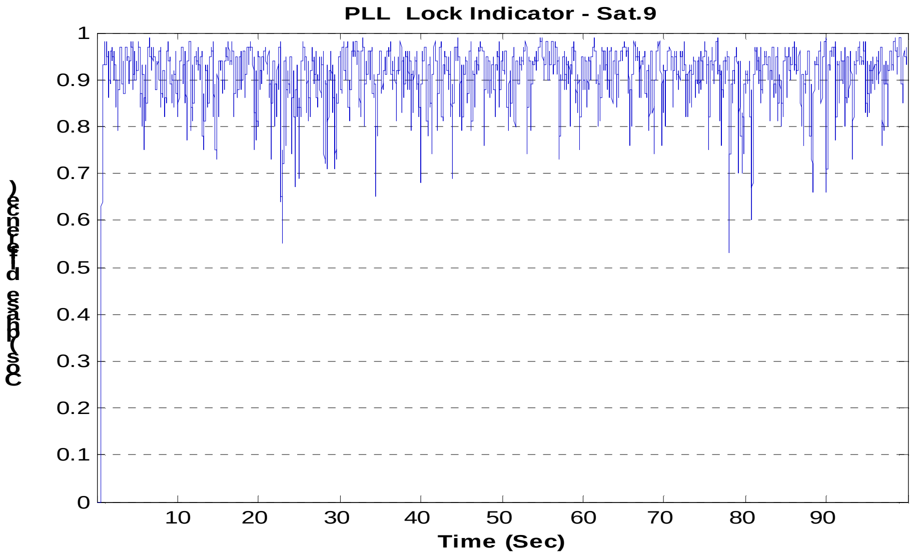

Figure 7.49: The PLL lock indicator for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

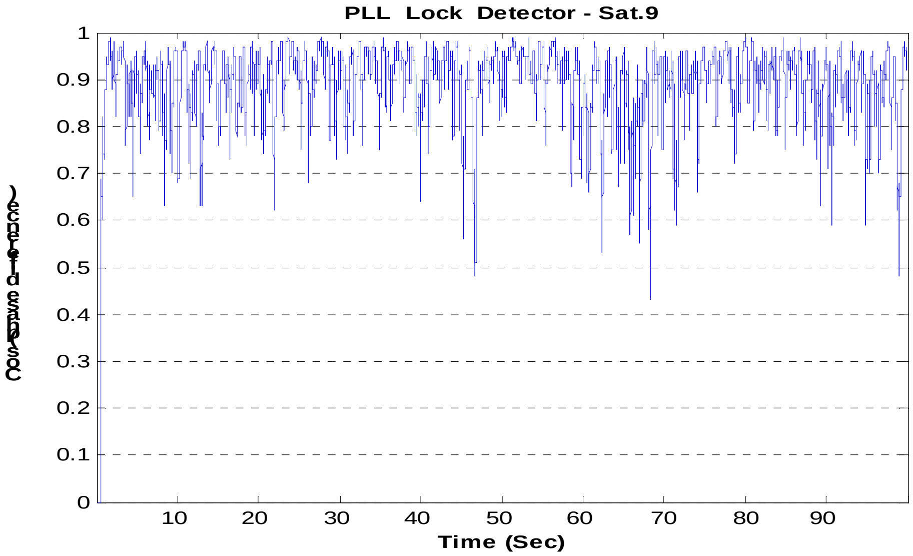

Figure 7.50: The PLL lock indicator for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

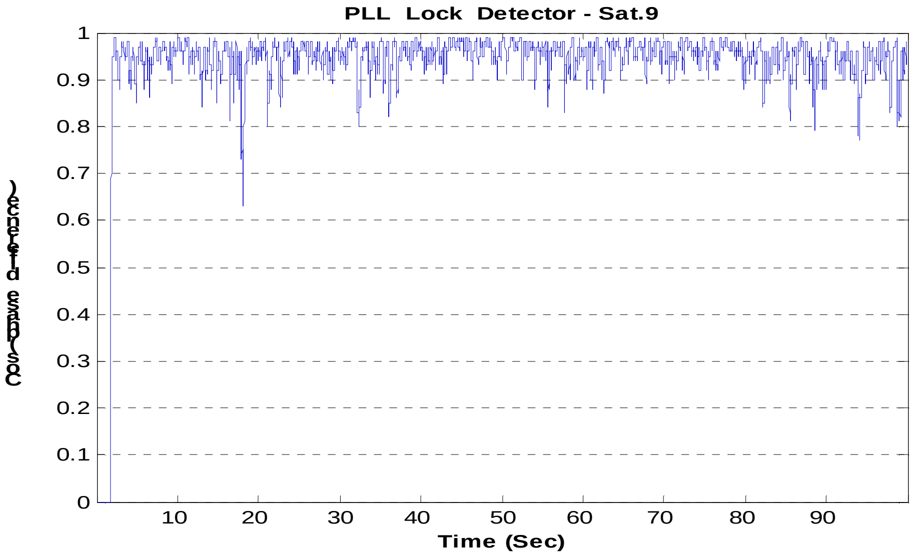

Figure 7.51: The PLL lock indicator for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

7.3.1.3 Results from the Adaptive bandwidth algorithm

Figures 7.52 and 7.54 show the true acceleration and jerk derived from the pseudorange rate given by the simulator and Figures 7.53 and 7.55 show the estimated acceleration and jerk using the adaptive bandwidth algorithm. Obviously the estimated acceleration and jerk can follow the changes in the real acceleration and jerk quite well. The biggest problem is that both estimations are noisy - especially the jerk, which is derived from acceleration. Even though the jerk in this test is not, on the average, higher than the jerk in test 1, the rate change of the jerk in test 2 is sometimes much higher than the one in test 1. Therefore it is more difficult for the adaptive bandwidth to keep lock.

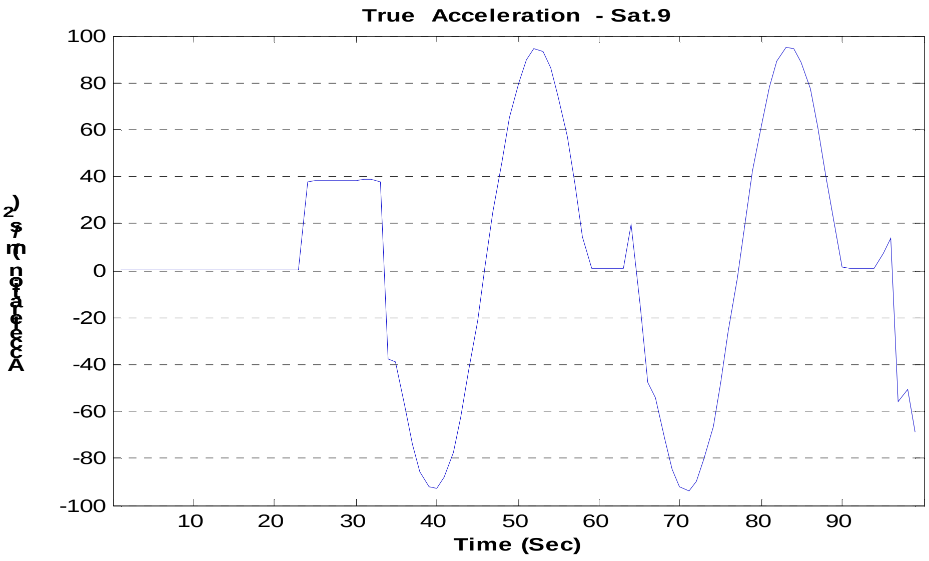

Figure 7.52: The true acceleration from the simulator for satellite 9

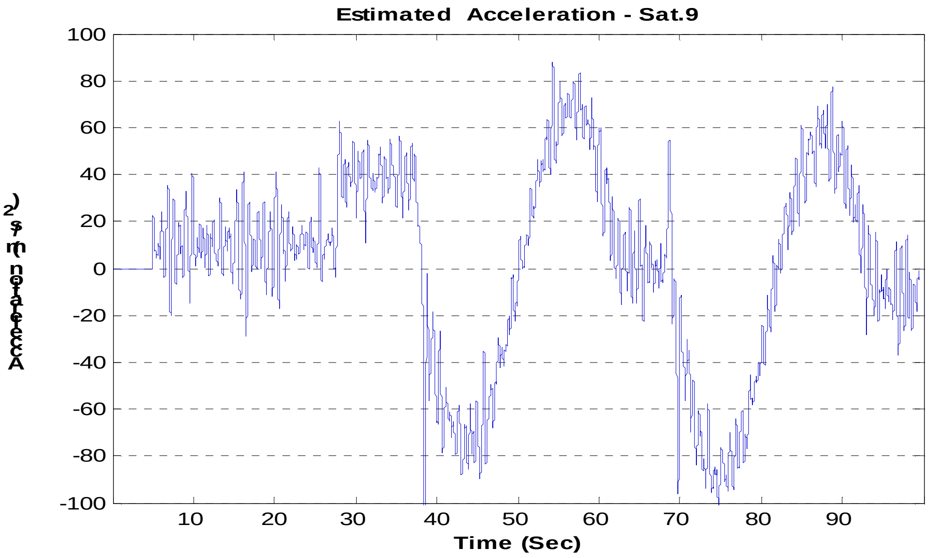

Figure 7.53: The estimated acceleration using the adaptive bandwidth algorithm for satellite 9

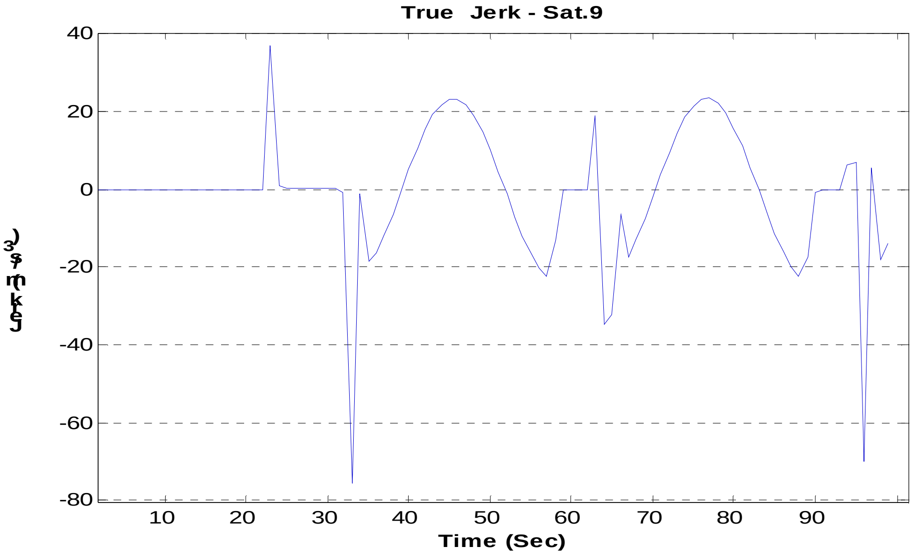

Figure 7.54: The true jerk from the simulator for satellite 9

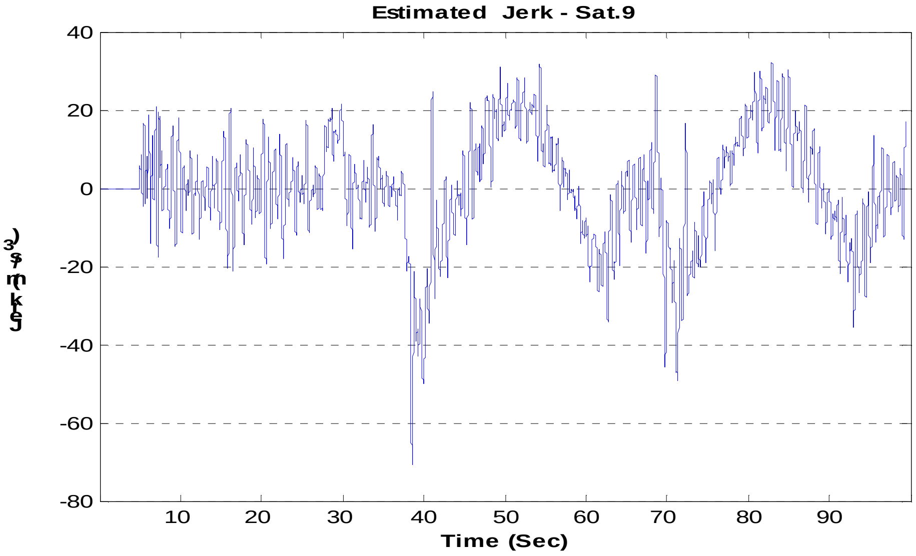

Figure 7.55: The estimated acceleration using the adaptive bandwidth algorithm for satellite 9

Figure 7.56 and 7.57 show the Doppler frequency after adapting the bandwidth. Figure 7.57 indicates that the PLL bandwidth changes with the incoming dynamics.

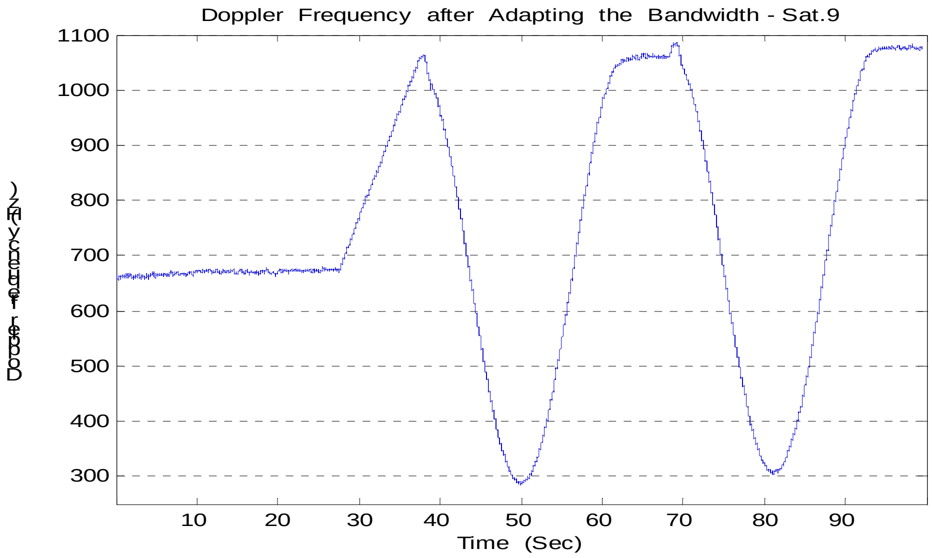

Figure 7.56: The Doppler frequency after adapting the bandwidth for satellite 9 (C/N_O is 45 dB-Hz)

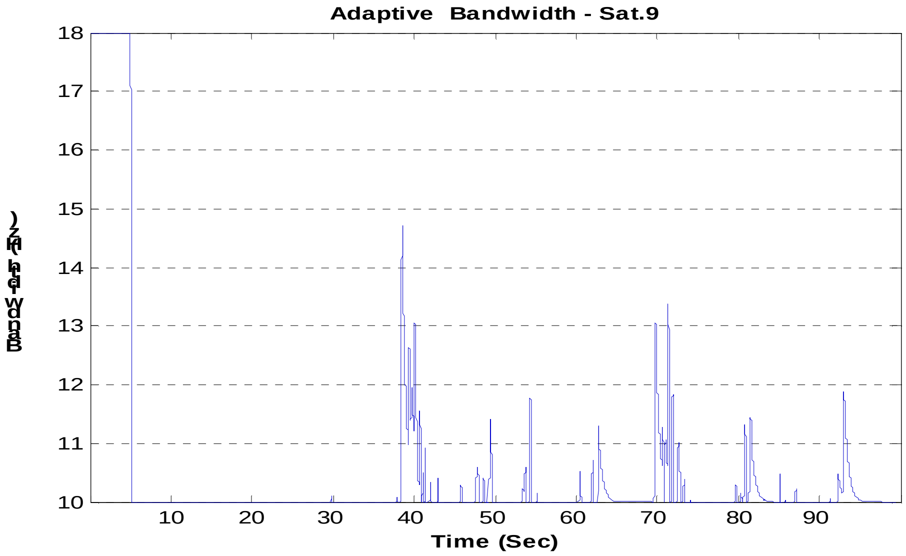

Figure 7.57: Adaptive bandwidth for satellite 9

7.3.2 Post Processing

7.3.2.1 Results from the Kalman Filter-Based Tracking

The post processed results for the Kalman filter-based tracking algorithm are shown in Figure 7.58, 7.59 and 7.60. The results clearly indicate improvement particularly for a high noise level and wide bandwidth case.

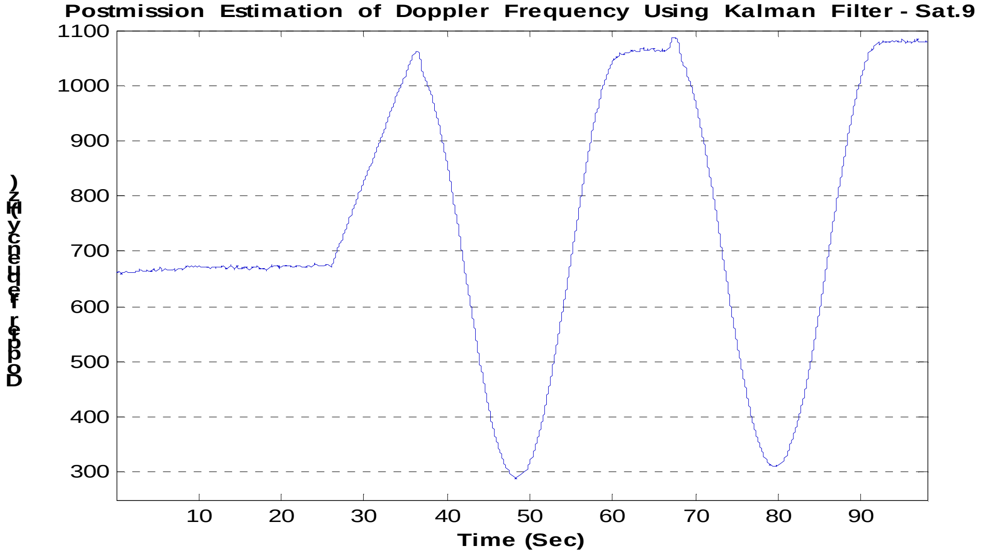

Figure7. 58: Doppler frequency from Kalman filter based tracking loop for satellite 9 (PLL Bandwidth = 18 Hz, C/N_O = 45 dB-Hz)

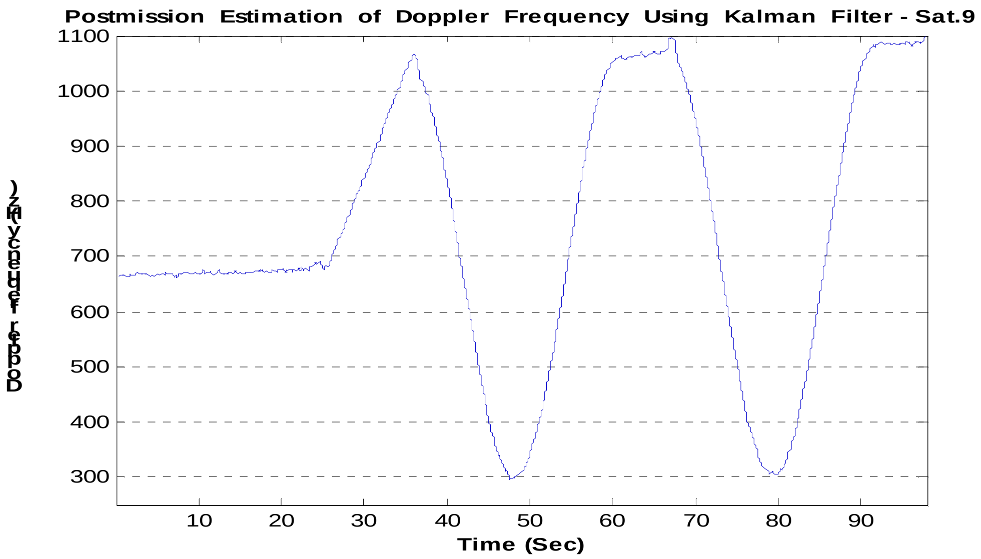

Figure7. 59: Doppler frequency from Kalman filter based tracking loop for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 39 dB-Hz)

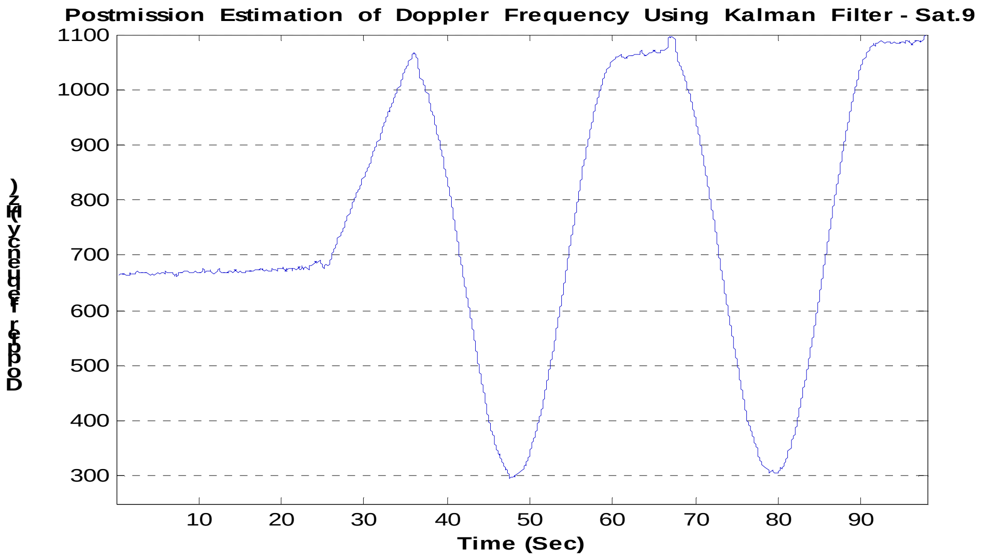

Figure 7.60: Doppler frequency from Kalman filter based tracking loop for satellite 9 (PLL Bandwidth = 30 Hz, C/N$_O$ = 39 dB-Hz)

7.3.2.2 Results from Applying the Wavelet De-noising Technique

The post processed results for applying the wavelet de-noising technique are shown in Figure 7.61, 7.62 and 7.63. The results show improvement particularly for the high noise power and wider bandwidth compared to Figure 7.46, 7.47 and 7.48.

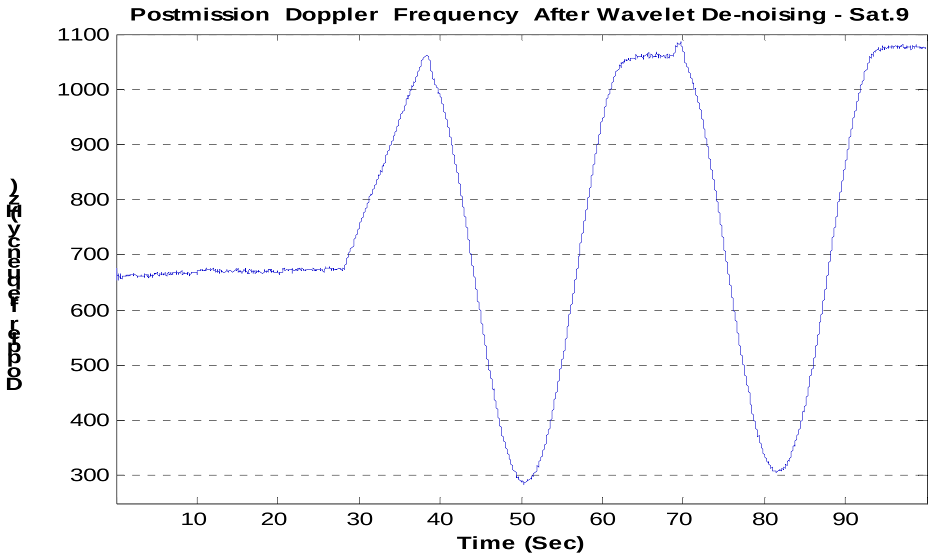

Figure 7.61: Doppler frequency after applying wavelet de-noising technique for satellite 9 (PLL Bandwidth = 18 Hz, C/N$_O$ = 45 dB-Hz)

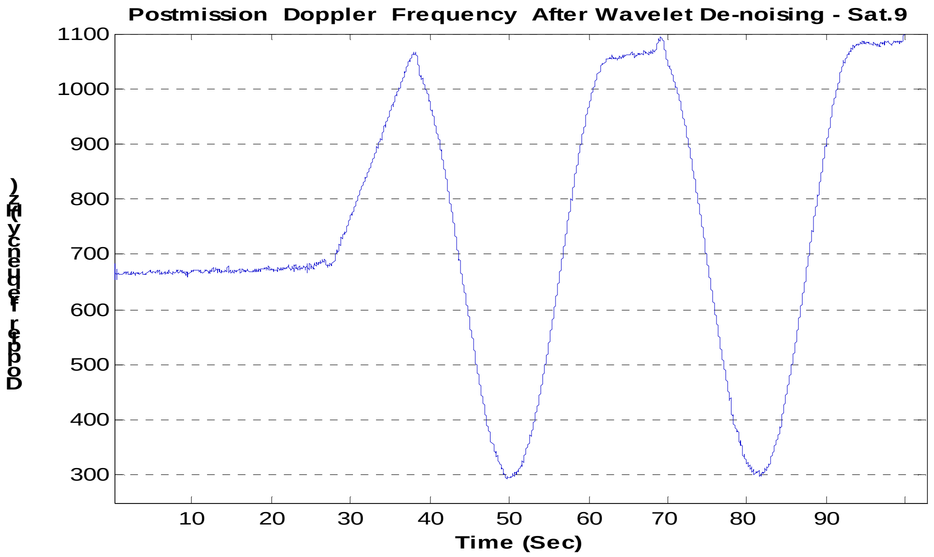

Figure 7.62: Doppler frequency after applying wavelet de-noising technique for satellite 9 (PLL Bandwidth = 18 Hz, C/N_O = 39 dB-Hz)

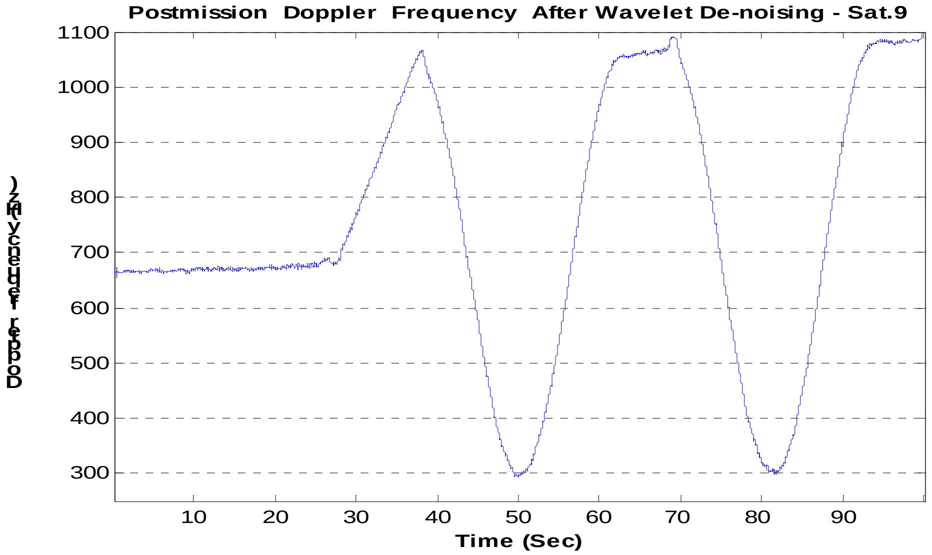

Figure 7.63: Doppler frequency after applying wavelet de-noising technique for satellite 9 (PLL Bandwidth = 30 Hz, C/N_O = 39 dB-Hz)

7.3.2.3 Results from the Adaptive Bandwidth Algorithm

The post processed result for the adaptive bandwidth algorithm is shown in Figure 7.64.

Compared to the real time result of Figure 7.56, the improvement is significant.

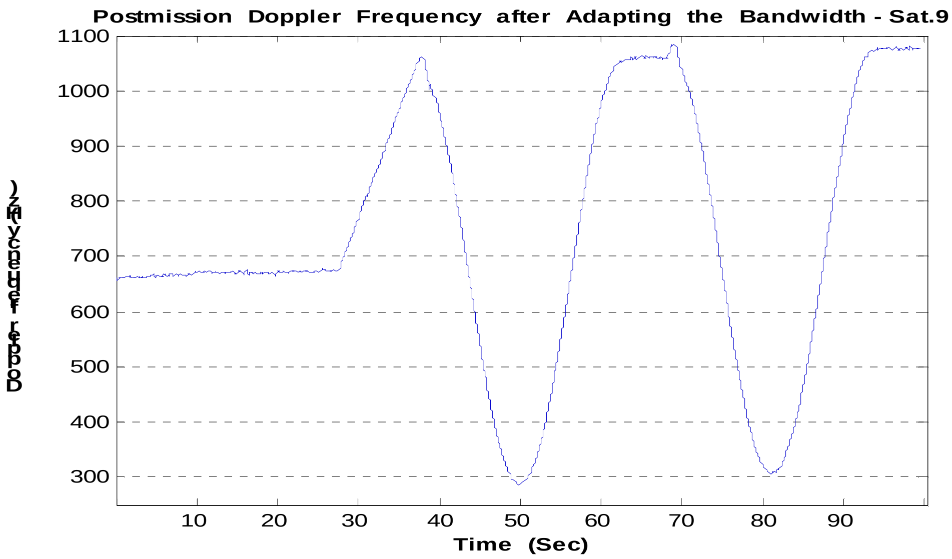

Figure 7.64: Doppler frequency from adaptive bandwidth algorithm for satellite 9 (C/N_O is 45 dB-Hz)

7.4 Summary

This section presents statistic results for the two tests and scenarios. Table 7.4 and 7.5 shows the real-time results for the horizontal motion and the three dimensional motion, respectively. The results for an ordinary PLL and the algorithms proposed in this thesis for three scenarios are presented. Both tables show a similar trend for the two tests. The ordinary PLL produces the highest maximum and minimum Doppler frequency error. The Kalman filter-based tracking algorithm demonstrates the best performance for the above two cases. The wavelet de-noising technique provides medium improvement, while the adaptive bandwidth algorithm provides little improvement. For both tests, the Doppler frequency error standard deviation for an ordinary PLL and the proposed algorithms deteriorates with an increase in the noise and PLL bandwidth. Again, the

Kalman filter-based tracking algorithm shows the best improvement, which becomes significant with an increase in the noise and PLL bandwidth. This result matches well those shown in Figure 7.8, 7.9 and 7.10. The wavelet de-noising algorithm also shows improvement, which are however less than those obtained with the Kalman filter-based tracking algorithm, but have similar trends.

Table 7. 4: Real-Time Statistical Results for the Horizontal Motion

Doppler Frequency Error			Ordinary PLL (Hz)	Kalman Filter (Hz)	Wavelet De-noising (Hz)	Adaptive Bandwidth (Hz)
Horizontal Motion	Bn=18 Hz C/N_O=45 dB-Hz	Max	9.7	4.4	6.4	9.2
		Min	-9.8	-5.1	-7.2	-8.7
		Mean	-0.1	0.0	0.1	0.0
		Std Dev	2.6	1.7	1.9	2.4
	Bn=18 Hz C/N_O=39 dB-Hz	Max	11.7	5.5	7.2	
		Min	-11.9	-5.1	-8.3	
		Mean	-0.1	0.0	-0.1	
		Std Dev	3	1.7	2.0	
	Bn=30 Hz C/N_O=39 dB-Hz	Max	18.5	5.6	8.0	
		Min	-17.7	-6.0	-8.2	
		Mean	0.0	-0.1	0.3	
		Std Dev	4.0	1.7	2.3	

Table 7. 5: Real-Time Statistic Results for the Three Dimensional Motion

Doppler Frequency Error			Ordinary PLL (Hz)	Kalman Filter (Hz)	Wavelet De-noising (Hz)	Adaptive Bandwidth (Hz)
Three Dimensional Motion	Bn=18 Hz C/N_O=45 dB-Hz	Max	9.3	5.0	7.1	8.0
		Min	-8.3	-4.7	-5.4	-7.3
		Mean	0.0	0.0	-0.1	-0.1
		Std Dev	2.3	1.5	1.8	2.1
	Bn=18 Hz C/N_O=39 dB-Hz	Max	14.3	6.5	7.5	
		Min	-13.2	-6.2	-7.3	
		Mean	0.1	0.0	0.0	
		Std Dev	3.4	2.1	2.4	
	Bn=30 Hz C/N_O=39 dB-Hz	Max	17.8	6.5	8.0	
		Min	-18.1	-6.0	-8.2	
		Mean	0.0	0.0	-0.2	
		Std Dev	4.4	2.1	2.4	

Post processed results are shown in Table 7.6 and 7.7. It is obvious that a much better performance is obtained, compared to the real-time results. The Kalman filter-based tracking algorithm still demonstrates the best performance. However, the wavelet-de-noising algorithm does not provide a large improvement compared to the other two algorithms. This indicates that if the wavelet de-noising technique is applied to a signal which is already processed by the wavelet de-noising technique in real-time, not too much improvement can be expected.

Table 7. 6 : Post Processed Statistic Results for the Horizontal Motion

Doppler Frequency Error			Kalman Filter (Hz)	Wavelet De-noising (Hz)	Adaptive Bandwidth (Hz)
Horizontal Motion	Bn=18 Hz C/N_O=45 dB-Hz	Max	2.5	4.2	3.8
		Min	-3.3	-4.5	-4.2
		Mean	-0.1	0.0	0.0
		Std Dev	1.2	1.6	1.3
	Bn=18 Hz C/N_O=39 dB-Hz	Max	3.4	4.7	
		Min	-4.1	-4.7	
		Mean	0.0	0.0	
		Std Dev	1.2	1.8	
	Bn=30 Hz C/N_O=39 dB-Hz	Max	3.8	5.2	
		Min	-4.0	-4.8	
		Mean	0.0	0.2	
		Std Dev	1.2	1.8	

Table 7. 7: Post Processed Statistic Results for the Three Dimensional Motion

Doppler Frequency Error			Kalman Filter (Hz)	Wavelet De-noising (Hz)	Adaptive Bandwidth (Hz)
Three Dimensional Motion	Bn=18 Hz C/N_O=45 dB-Hz	Max	3.0	4.3	3.2
		Min	-2.8	-4.1	-4.0
		Mean	0.0	-0.1	-0.1
		Std Dev	1.0	1.6	1.3
	Bn=30 Hz C/N_O=39 dB-Hz	Max	4.1	4.5	
		Min	-3.5	-5.3	
		Mean	0.0	0.0	
		Std Dev	1.4	2.0	
	Bn=30 Hz C/N_O=39 dB-Hz	Max	4.0	4.1	
		Min	-4.4	-4.6	
		Mean	-0.1	0.0	
		Std Dev	1.4	2.1	

Table 7.8 and 7.9 present the Doppler frequency improvement obtained from the two tests and different scenarios. In general, all results comply with the results from previous figures. Better performances are obtained from the Kalman filter-based tracking algorithm and the wavelet de-noising algorithm. However the adaptive bandwidth algorithm yields a poor improvement particularly for the real-time processing. This is because the minimum PLL bandwidth is set to 10 Hz which is not narrow enough in order to obtain a better improvement. The reason for using a 10 Hz minimum PLL

bandwidth is to guarantee PLL tracking reliability. As discussed before, a PLL does not allow its bandwidth to be changed too fast otherwise it will lose lock. However, a high dynamic signal requires a rapid variation of the PLL bandwidth, and a 10 Hz PLL bandwidth can prevent this from happening.

Table 7. 8: Real-Time Doppler Frequency Improvement

Scenarios		Doppler Frequency Improvement		
		Kalman Filter	Wavelet De-noising	Adaptive Bandwidth
Horizontal Motion	Bn=18 Hz C/N_O=45 dB-Hz	35%	26%	7%
	Bn=18 Hz C/N_O=39 dB-Hz	41%	30%	
	Bn=30 Hz C/N_O=39 dB-Hz	56%	43%	
Three Dimensional Motion	Bn=18 Hz C/N_O=45 dB-Hz	34%	23%	10%
	Bn=18 Hz C/N_O=39 dB-Hz	39%	30%	
	Bn=30 Hz C/N_O=39 dB-Hz	52%	44%	

Table 7. 9: Post-Processing Doppler Frequency Improvement

Scenarios		Doppler Frequency Improvement		
		Kalman Filter	Wavelet De-noising	Adaptive Bandwidth
Horizontal Motion	Bn=18 Hz C/N_O=45 dB-Hz	55%	39%	50%
	Bn=18 Hz C/N_O=39 dB-Hz	58%	43%	
	Bn=30 Hz C/N_O=39 dB-Hz	69%	56%	
Three Dimensional Motion	Bn=18 Hz C/N_O=45 dB-Hz	55%	32%	42%
	Bn=18 Hz C/N_O=39 dB-Hz	59%	40%	
	Bn=30 Hz C/N_O=39 dB-Hz	68%	52%	

CHAPTER EIGHT: CONCLUSIONS AND RECOMMENDATIONS

Three algorithms to improve the tracking performance of the PLL are presented in Chapter 4, 5 and 6. These algorithms were developed and implemented as a part of a GPS software receiver. Then they were tested and compared to the results using an conventional GPS software receiver.

8.1 Conclusions

The Kalman filter-based tracking algorithm uses the tracking ability of an ordinary PLL as a basis and augments it with a carrier phase dynamics model to estimate the phase difference, the Doppler frequency and its rate change. The purpose of keeping the ordinary PLL is to guarantee a good tracking reliability. Even though the output of the PLL discriminator is very noisy, the carrier phase dynamic model can still extract valuable information from it. By varying the weight on the measurement according to its noise power level, the Kalman filter-based tracking algorithm demonstrates a significant improvement. The estimated Doppler frequency and phase difference can be used as input to derive the carrier phase measurement. The test results shown in Chapter 7 demonstrate that this algorithm works especially well under low signal-to-noise ratios and high dynamic applications that require wide bandwidths.

Applying a wavelet de-noising technique in the PLL helps reducing the noise within the bandwidth of the loop filter and therefore, a less noisy tracking performance can be obtained by the NCO. Even though the noise is reduced, the PLL is still not 100%

effective due to the effect of high dynamic stress. However this disadvantage can be overcome by increasing the bandwidth. Results from Chapter 7 indicate that, under the conditions of wide bandwidth and a low signal-to-noise ratio, this algorithm produces superior tracking performance. Even though the improvement is not as significant as that resulting from the Kalman filter-based tracking algorithm, applying the wavelet de-noising technique in the PLL is still an effective method to improve the PLL tracking performance, particularly in post processing mode.

The key premise of the successful use of the adaptive bandwidth algorithm is the correct estimation of incoming signal dynamics. The adaptive bandwidth algorithm can estimate the dynamics quite well under normal signal-to-noise ratio and high dynamics but with relatively less random variation. As a result, the bandwidth can be adapted correctly according to the dynamics and the tracking performance is improved specifically for the period of low dynamics. However, if the signal-to-noise ratio is low or dynamics changes are very fast in a short interval, some problems will arise. Because the measurements used for the estimation are noisy, the estimation cannot avoid the impact of the noise and the estimation accuracy will decrease. Consequently the bandwidth will be adapted in an incorrect way. On the other hand, if dynamics changes very fast and randomly, the PLL bandwidth has to be changed fast to follow this change of trend. However as discussed in Chapter 6, the PLL does not allow its bandwidth to change fast otherwise lost of phase lock will result. In the worst case, the PLL will not only lose phase lock but also will lose frequency lock. These two problems restrict the application of the adaptive bandwidth algorithm.

Post processing results show better improvements especially for the Kalman filter-based tracking algorithm and adaptive bandwidth algorithm than the results for the real time processing case. These improvements further show that the wavelet de-noising technique can be applied to improve noisy measurements such as pseudoranges.

8.2 Recommendations

Based on the test results in the thesis, the following recommendations can be made: The Kalman filter-based tracking algorithm provides better improvements than the other two algorithms. The combination of the PLL and the carrier phase dynamic model improves the tracking reliability particularly in the case of high and random dynamics applications. Based on the theory, the measurement is not restricted at the output of the PLL discriminator. Measurements from other points are possible such as the Doppler frequency from the output of the NCO.

Applying the wavelet de-noising technique in the PLL is another promising method. Actually this algorithm is not limited in improving the tracking performance applications but also other applications that relate to noisy measurements such as pseudoranges. In addition, the selection of the wavelet and decomposition levels also plays some role in the final performance. Further, the combination of the wavelet de-noising technique and the other two algorithms offers potential. In addition, the de-noising technique can also be applied at the output of the loop filter.

Using an adaptive bandwidth algorithm must be carefully considered under a low signal-to-noise ratio. If considering all tracking errors including clock errors caused by vibration and Allan deviation, perhaps a more accurate bandwidth can be obtained.

The combination of the Kalman filter-based tracking algorithm with the wavelet de-noising technique may provide further improvement. In this scheme, the wavelet de-noising should be applied immediately after the PLL discriminator and the result after wavelet de-noising used as the phase difference measurement for the carrier phase dynamic model. The less noisy measurement is likely to provide a significant improvement.

Another possible combination is between the adaptive bandwidth algorithm and the wavelet de-noising technique. Again, the wavelet de-noising technique should be applied immediately after the discriminator. In this way, a less noisy phase difference is provided to the adaptive bandwidth algorithm. From previous discussion, it is known that the phase difference is an important measurement for the adaptive bandwidth algorithm to estimate the dynamics. Therefore combining these two algorithms together will improve the performance of the adaptive bandwidth algorithm in the case of low signal-to-noise power ratios

There are two tests that could be done to enhance the results presented in this thesis. In all three algorithms, the clock influence is not taken into account. If its effect can be removed, better tracking performance could be obtained. Particularly for the adaptive bandwidth algorithm, a more accurate bandwidth estimation is possible.

Integrating GPS with other sensors is another area which has been studied in high dynamic applications. The biggest challenge for this algorithm is the accuracy of the aiding Doppler frequency which mainly depends on the quality of the sensors. The accurate aiding Doppler can effectively narrow the bandwidth of the PLL and consequently improve its tracking performance. In this case, the errors related to the clock become the dominant error sources.

References

Accord Software & Systems Private Limited (2004) *GPS Signal Tap User's Guide*

Analogdevices Company (2004), Direct Digital Synthesizer, retrived from www.analog.com May. 2004

Best R. E. (1999) *Phase-Locked Loops, Desigins Simulation, and Applications.* McGraw-Hill, fourth edition.

Brown, R. G. and P. Y. C. Hwang (1992) *Introduction to Random Signal and Applied Kalman Filtering*. John Wiley & Sons, Inc., second edition.

Coifman R. R. and Donoho D. L. (1995) Translation-Invariant De-Noising, retrived from URL http://www-stat.stanford.edu/~donoho/Reports/1995/TIDeNoise.pdf

Corbell P. M. and M. M. Miller (2000) Design and Analysis of A Matlab Based Digitized IF GPS Signal Simulator and a Simulink Based Configurable GPS Receiver. Proceeding of ION GPS 2000, pp. 1906-1915. Institute of Navigation.

Demoz G.-E., A. Razavi, P. Enge, J. Gautifer, D. Akos, S. Pullen and B. Pervan (2003) *Doppler Aided Tracking Loops for SRGPS Integrity Monitoring.* Proceeding of ION GPS 2003, pp. 2515-2523. Institute of Navigation.

Donoho D. L. (1995) *De-Noising by Soft-thresholding. IEEE transactions on information theory*. 1995, pp.613-627, IEEE.

Egan W. F. (1998) *Phas –Lock Basics,* John Wiley & Sons, Inc.

Gelb, A. (1974) *Applied Optimal Estimation.* The M.I.T Press.

Gold K. and A. Brown (2004) *Architecture and Performance Testing of a Software GPS Receiver for Space-based Applications.* Proceeeding s of IEEEAC, 2004, pp.1-12.

Hu G. S. (2001) *Digital Signal Processing,* Tsinghua University Press.k

Jovancevic, A., A. Brown, S. Ganguly, J. Goda, M. Kirchner and S. Zigic (2003) *Real – Time Dual Frequency Software Receiver*. Proceeding of ION GPS 2003, pp. 2572-2583. Institute of Navigation.

Jung H., Psiaki M. L. and Powell S. P. (2003) *Kalman-Filter-Based Semi-Codeless Tracking of Weak Dual-Frequency GPS Signals*. Proceeding of ION GPS 2003, pp. 2515-2523. Institute of Navigation.

KAPLAN E. D. (1996) *Understanding GPS: Principles and Applications.* Artech House Publishers, Norwood.

Krumvieda K., P. Madhani, C. Cloman, E. Olson, J. Thomas, P.a Axelrad, W. Kober (2001) A Complete IF Software GPS Receiver: A Tutorial about the Details. Proceeding of ION GPS 2003, pp. 789-811. Institute of Navigation.

Legrand F. and Macabiau C. (2000) *Improvement of Pseuforange Measurements Accuracy By Using Fast Adaptive Bandwidth Lock Loops*. Proceeding of ION GPS 2000, pp. 2346-2356. Institute of Navigation.

Legrand F. and Macabiau C. (2001) *Results of the implementation of the Fast Adaptive Bandwidth Lock Loops on a real GPS receiver in a high dynamics context.* GNSS 2001 International Symosium, pp. 1-6.

Legrand F. and Macabiau C. (2001) *Real-time minimization of the total tracking error in phase and delay lock loops – a second approach of the Fast Adaptive Bandwidth Algorithm. Proceeding of the ION 57th Annual Meeting & Cigtf 20th Biennial Guidance Test Symposium, 2001.*

Ma, C., G. Lachapelle, and M.E. Cannon, (2004) Implementation of a Software GPS Receiver. Proceedings of GNSS 2004 (Session A3, Long Beach, CA), 21-24 September), The Institute of Navigation, Fairfax, VA, pp. 8-19.

Macabiau C. and F. Legrand (2000) *GPS L5 Receiver Implementation Issues.* Proceeding of ION GPS 2003, pp. 153-163. Institute of Navigation.

Malat S. (1998) *A Wavelet tour of signal processing*. Academic Press.

Mallat S. (1989) A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Transactions on Pattern and Machine Intelligence. July, 1989, IEEE.

Mertins A. (1999) *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications.* John Wiley & Sons, Inc.

Parkinson B. W. and Spilker Jr J. J.(1996) *Global Positioning System: Theory and Applications.* American Institute of Aeronautics and Astronautics, Inc.

Psiaki M. L. (2000) *Attitude Sensing Using a Global – Positioning – System Antenna on A Turntable.* URL http://www.mae.cornell.edu/Psiaki/rot_ant_gps_attitude.pdf

Psiaki M. L. (2001) *Smoother-Based GPS Signal Tracking in a Software Receiver*. Proceeding of ION GPS 2001, pp. 2900-2913. Institute of Navigation.

Psiaki M. L. and H. Jung (2002) *Extended Kalman Filter Methods for Tracking Weak GPS Signals.* Proceeding of ION GPS 2002, pp. 2539-2553. Institute of Navigation.

Ray J. (2003) *Advanced GPS Receiver Technology*. Engo 699.73 Lecture Notes. Department of Geomatics Engineering, The University of Calgary.

Rioul O. and P. Duhamel (1992) *Fast Algorithms for Discrete and Continuous Wavelet Transforms.* IEEE Transactions on Information Theory, March, 1992, IEEE.

Salychev S. O. (1998) *Inertial Systems in Navigation and Geophysics.* Bauman MSTU Press.

Taswell C. (2000) *The What, How, and Why of Wavelet Shrinkage Denoising*. Computing in Science &Engineering, May-June, 2000, IEEE.

The MathWorks, Inc. (2004) Matlab Help – Wavelet Toolbox

Vetterli M. and C. Herley (1992) Wavelets and Filter Banks: Theory and Design. IEEE Transactions on Signal Processing, Sep. 1992, pp. 2207-2232

Yang C. (2003) *Tracking of GPS Code Phase and Carrier Frequency in the Frequency Domain.* Proceeding of ION GPS 2003, pp. 628-637. Institute of Navigation.

Zhang Y., Y. Y. Wang, W. Q. Wang and B. Liu (2001) *Doppler Ultrasound Signal Denoising Based on Wavelet Frames.* IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, May, 2001, pp. 709-716. IEEE.

Ziedan N. I. and Garrison J. L. (2003) *Bit Synchronization and Doppler Frequency Removal at Very Low Carrier to Noise Ratio Using a Combination of the Viterbi Algorithm with an Extended Kalman Filter.* Proceeding of ION GPS 2003, pp. 2515-2523. Institute of Navigation.

Appendix A

DERIVATION OF THE STEADY-STATE ERROR

From the Section 6.2 we obtain

$$\theta_e(\infty) = \lim_{z\to 1}\left[(z-1)\theta_e(z)\right] = \lim_{z\to 1}\left[(z-1)\frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K\sum_{n=0}^{N-1} b_n z^{-n-1}}\sum_{m=1}^{\infty}\frac{\theta_0^{(m)}}{m!}Z[k^m]\right] \tag{A.1}$$

From the characteristic of the Z transform, we get

$$Z[nx(n)] = -z\frac{d}{dz}Z[x(n)] \tag{A.2}$$

For $x(n) = u(n)$, Equation C.2 can be rewritten as

$$Z[nu(n)] = -z\frac{d}{dz}Z[u(n)] \tag{A.3}$$

Because $Z[u(n)] = \dfrac{z}{z-1}$

$$\begin{aligned}
Z[nu(n)] &= \frac{z}{(z-1)^2}\\
Z[n^2u(n)] &= \frac{z(z-1)}{(z-1)^3}\\
Z[n^3u(n)] &= \frac{-z^3+2z+5}{(z-1)^4}\\
Z[n^4u(n)] &= \frac{z(-z^3-3z^2+6z+22)}{(z-1)^5}
\end{aligned} \tag{A.4}$$

$$\theta_e(\infty) = \lim_{z\to 1}\left\{(z-1)\frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K\sum_{n=0}^{N-1} b_n z^{-n-1}}\left[\frac{z\theta_0^{(1)}}{(z-1)^2} + \frac{z(z+1)\theta_0^{(2)}}{2(z-1)^3} + \frac{(-z^3+2z+5)\theta_0^{(3)}}{6(z-1)^4} + \right.\right.$$

$$\left.\left. + \frac{z(-z^3-3z^2+6z+22)\theta_0^{(4)}}{24(z-1)^5} + \cdots\right]\right)$$

$$= \lim_{z\to 1}\left\{(z-1)\frac{K_d(1-z^{-1})^N}{(1-z^{-1})^N + K\sum_{n=0}^{N-1} b_n z^{-n-1}}\left[\frac{\theta_0^{(1)}}{(z-1)^2} + \frac{\theta_0^{(2)}}{(z-1)^3} + \frac{\theta_0^{(3)}}{(z-1)^4} + \frac{\theta_0^{(4)}}{(z-1)^5} + \cdots\right]\right\}$$

(A.5)

From the above equation, it is obvious that for the items

$$\theta_e(\infty) = \begin{cases} 0 & m < N \\ \dfrac{\theta_0^N}{K_O\sum_{n=0}^{N-1} b_n} & m = N \\ \infty & m > N \end{cases} \tag{A.6}$$