極寬捕獲範圍的一位元取樣數位鎖相迴路

研究生：李信德　　　　　　　　　　　指導教授：高銘盛 教授

國立交通大學

電信工程學系碩士班

摘　要

在此篇論文中，我們提出具有極寬捕獲範圍的一位元取樣數位鎖相迴路。該數位鎖相迴路主要經由兩個步驟達成極寬捕獲範圍：一、資訊蒐集：進行多次的相位差偵測，將偵測到的平均相位值結合數位電路的特性，可迅速將數位振盪器的頻率逼近輸入訊號頻率。二、頻率微調：利用傳統鎖相迴路的概念，將數位振盪器的頻率微調以鎖住輸入訊號頻率。

此外，我們將捕獲範圍分成：低頻、中頻以及高頻三個頻段，並個別給予不同的關鍵參數。依循以上步驟，我們實現極寬捕獲範圍的目標。由電腦模擬的結果顯示，在無雜訊干擾的環境中，自然頻率為 10K Hz 的一位元取樣數位鎖相迴路，可以達到的捕獲範圍為 50 Hz ~ 10K Hz，此捕獲範圍為自然頻率的 99.5%。

One Bit Quantized Digital Phase-Locked Loop with Ultra-Wide Capture Range

Student: Hsin-Te Lee Advisor: Prof. Ming-Seng Kao

Department of Communication Engineering

National Chiao Tung University

Abstract

In this thesis, we propose the idea of one bit quantized digital phase-locked loop (PLL) with ultra-wide capture range. To achieve the goal, we devise two steps: 1. Information collection: we estimate the phase difference many times and average the results to obtain an accurate phase estimation. Combining the accurate estimation and the feature of digital PLL, we transfer the output frequency of numerically-controlled oscillator (NCO) to around the input signal frequency. 2. Fine frequency adjustment: based on the concept of phase locking, we slightly adjust the NCO output frequency until the system is locked.

In addition, we separate the capture range into lower frequency, middle frequency and higher frequency ranges. Depending on the frequency range, we change some key factors in order to achieve phase locking over a wide frequency range. As a result, we can realize a digital PLL with ultra-wide capture range. From the simulation results, we can lock input frequency whose range is from 50 Hz to 10K Hz in noiseless environment, with an initial NCO natural frequency of 10K Hz. The system achieves almost 99.5% capture range of the natural frequency.

Acknowledgements

I would like to show my deepest appreciation to my advisor, Dr. Ming-Seng Kao, for his enthusiastic guidance and patience. It is absolutely lucky to be his student and his positive attitude affects me a lot. Besides, I would like to thanks all the members in our laboratory for their support, especially Wan-Hsin Hsieh. He gives me a lot of directions of the work during the two years. Last but not the least; I have to thanks my family, my grandparents, parents and sisters. All of their love and support are the strongest energy for me. Without their encouragement, this work could not be possible.

Content

List of figure

Chapter 1.
Introduction

A. PLL overview

PLL is a technique that has a great contribution to the technology advancement in communications and motor servo control systems in the past 40 years. The introduction of PLL is beneficial to coherent communication systems, too. In the late 1970's, PLL did not be widely used because of the difficulty in circuit realization. With the rapid development of integrated circuits (ICs), the applications of PLL were widespread in modern communication systems. Since then, the PLL has made much progress and turned its use in high-precision apparatus into consumer electronics. It makes modern electronic systems be able to improve performance and reliability, especially in common electronic applications used daily [1].

A PLL contains three basic functional blocks: 1. Phase detector (PD), 2. Loop filter (LF), 3. Voltage-controlled oscillator (VCO). The first PLL ICs appeared around 1965 and were purely analog devices. In the analog PLL, an analog multiplier was used as the PD, the LF was built from a passive or active RC filter, and the VCO was used to generate the output signal of the PLL. This type of PLL is referred to as the "linear PLL" (LPLL) today. In the following years, the PLL shifted slowly but steadily into digital domain. The very first digital PLL (DPLL), which appeared around 1970, was in effect a hybrid device: only the phase detector was built from a digital circuit, but the remaining blocks were still analog. A few years later, the "all-digital" PLL (ADPLL) was invented. The ADPLL was exclusively built from digital function blocks and hence didn't contain any passive components like resistors and capacitors.

In this thesis, we focus on the capture range of PLL. Definition of capture range is given below.

Capture range: Consider a PLL that is not in lock, then the input signal frequency slowly approaches the free running frequency of the VCO, the maximum frequency range in which the PLL finally becomes in lock is called the capture range [3].

Normally, no matter what kind of PLL it would be, the capture range is usually narrow, even though we implement circuit by high-order PLLs [4]. For some special design of ADPLL, they propose Locking Status Indicator (LSI) in a new structure. When the PLL becomes out of lock, the PLL increases the loop bandwidth and achieve fast locking. Hence, they can also achieve wider capture range [5].

B. Our work

In this thesis, we propose a new method of ADPLL which is called "one bit quantized digital PLL". The concept of one bit quantized digital PLL is originated from typical arctangent phase discriminator (APD) of GPS [6]. We utilize the idea and extend the system to achieve an ultra-wide capture range. In addition, we do not have to change our loop bandwidth and do not have any indicator of the input frequency. In other words, we simplify the architecture of ADPLL and reach the goal of ultra-wide capture range.

C. Thesis organization

The thesis is organized as below. In Chapter 2, we will introduce the operating principles of PLL, from LPLL to ADPLL. Then, the concept of one bit quantized digital PLL will be described. Next, in Chapter 3, we analyze the affect of every parameter in one bit quantized digital PLL and devise the method to achieve ultra-wide capture range. Then, we study the system performance under noisy environment in Chapter 4. Finally, we summarize the important ideas of our design and show the future work in Chapter 5.

Chapter 2.
One Bit Quantized Digital PLL System Overview

As we know that the multi-bit analog-to-digital-conversion (ADC) is utilized in most applications. For example, all digital phase-locked loop is one of the applications. In order to reduce the complexity of the quantization design, the one-bit ADC has been investigated. In this chapter, we would like to firstly introduce the general concept of PLL and one-bit ADC. Then, we combine PLL with one-bit ADC and bring the idea - one bit quantized digital phase-locked loop.

2.1. Operating principles of PLL

A PLL is a circuit that makes a particular oscillator to track another one. To be more specific, a PLL is a circuit synchronizing an output signal (generated by an oscillator) with an input or reference signal in frequency as well as in phase. In the synchronized—often called locked—state, the phase error between the output signal and the input signal remains constant or zero.

If a phase error builds up, a control mechanism acts on the oscillator in such a way that the phase error is again reduced to a minimum. In such a control system, the phase of the output signal is actually locked to the phase of the input signal. This is the reason why it is regarded as a phase-locked loop [5].

The basic operating principle of PLL is explained by the example of the linear PLL. Its block diagram is shown in Fig. 2.1. The PLL contains three basic functional blocks:

1. Phase detector.

2. Loop filter.

3. Voltage-controlled oscillator.

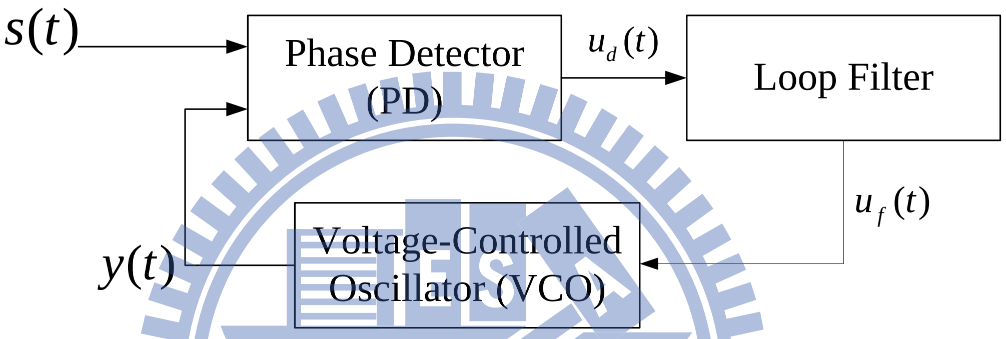

Fig. 2.1 PLL block diagram.

The signals within the PLL circuit are defined as follows:

- The input (or reference) signal $s(t)$ with angular frequency ω_c.
- The output signal $y(t)$ of the VCO with angular frequency ω_o.
- The output signal $u_d(t)$ of the phase detector ($\bar{u}_d$ is the average value of $u_d(t)$).
- The output signal $u_f(t)$ of the loop filter.
- The phase error θ_e, defined as the phase difference between signals $s(t)$ and $y(t)$
- The natural frequency of VCO (ω_n).

Phase detector

The PD is referred to as a phase comparator, which compares the phase of $y(t)$

with the phase of $s(t)$ and develops an output signal $u_d(t)$ that is approximately proportional to the phase error $\theta_e(t)$, given as

$$u_d(t) = K_d \cdot \theta_e(t), \tag{2.1}$$

where K_d (volts/rad) is the gain of the PD. Fig. 2.2 is a graphical representation of Eq.(2.1).

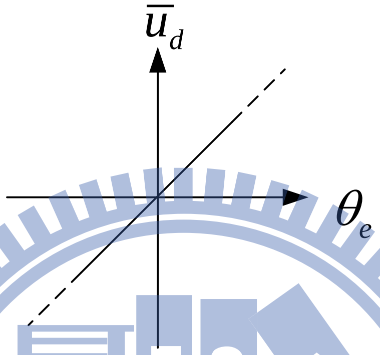

Fig. 2.2 Transfer function of the PD.

Loop filter

The output signal $u_d(t)$ of the PD contains a dc component and an ac component. The phase error is contained in the dc component, while the ac component is undesired. Therefore, the loop filter is used to eliminate ac component. Besides, another purpose of the loop filter is to attenuate signals at the reference frequency (ω_c) and its harmonics. In most cases, a first-order low-pass filter is used.

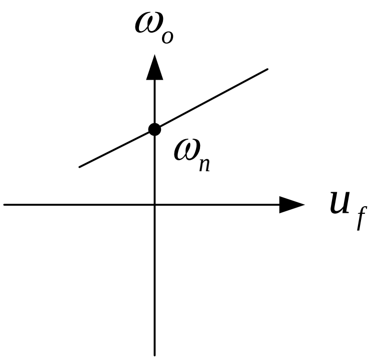

Fig. 2.3 Transfer function of the VCO.

Voltage-controlled oscillator

A typical characteristic of a voltage-controlled oscillator is shown in Fig. 2.3. Here, the VCO oscillates at an frequency ω_o, which is determined by the input signal $u_f(t)$. The frequency ω_o is given by

$$\omega_o(t) = \omega_n + K_o u_f(t), \tag{2.2}$$

where K_o is the VCO gain in $rad \cdot s^{-1} \cdot V^{-1}$. Actually, the curve of Fig. 2.3 may be nonlinear, but it usually simplifies the PLL design if the slope is the same everywhere.

2.2. One bit quantized digital PLL

From analog to digital:

An all digital phase-locked loop (ADPLL) works similar to the analog PLL, but it is implemented completely by digital circuits. ADPLL is easier to implement and design, and is less sensitive to voltage variation than analog PLL; however it typically suffers from the higher phase noise due to quantization error. Normally, a higher number of quantization bits leads to the lower quantization error. Therefore, most of the ADPLL system nowadays is quantized by multi-bit ADC to reduce quantization error. However, the complexity of multi-bit quantization increases either in the hardware design or software design. For reducing the complexity of multi-bit quantization, we introduce the idea of “one bit quantized digital PLL”.

Structure of one bit quantized PLL:

A general operation block diagram of one bit quantized PLL is shown in Fig. 2.4. As we see, the structure is similar to analog PLL. An obvious difference between one bit quantized digital PLL and analog PLL is that the signals are always one-bit-quantized before they are processed.

Comparing with analog PLL, the PD of one bit quantized PLL can not just be a mixer because the output of one bit quantization is merely +1 and -1. If we implement the PD of one bit quantized PLL by a mixer, the possible results— +1 or -1 —can not express the phase difference between input signal and output signal correctly. Therefore, we have to establish a new PD for one bit quantized system. The following is the introduction of the new PD.

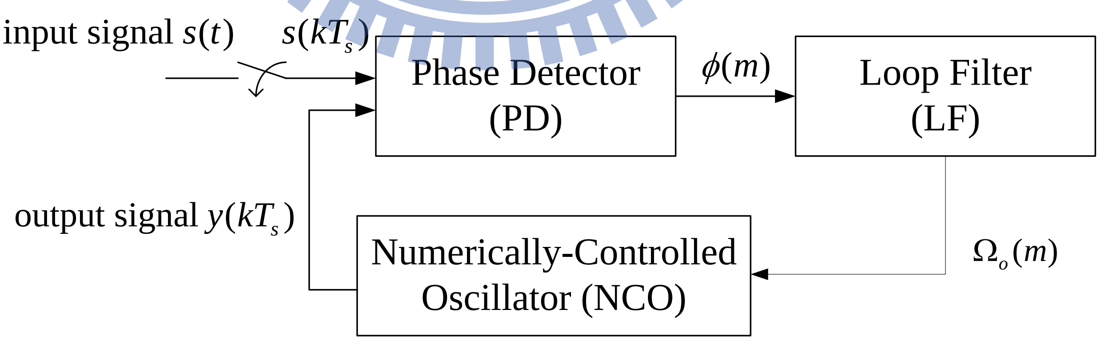

Fig. 2.4 The block diagram of one bit quantized digital PLL, where $\phi(m)$ is the estimated phase difference between $s(kT_s)$ and $y(kT_s)$, while $\Omega_o(m)$ is the input of NCO.

2.2.1. The PD of one bit quantized digital PLL

Digital phase discriminator [7]

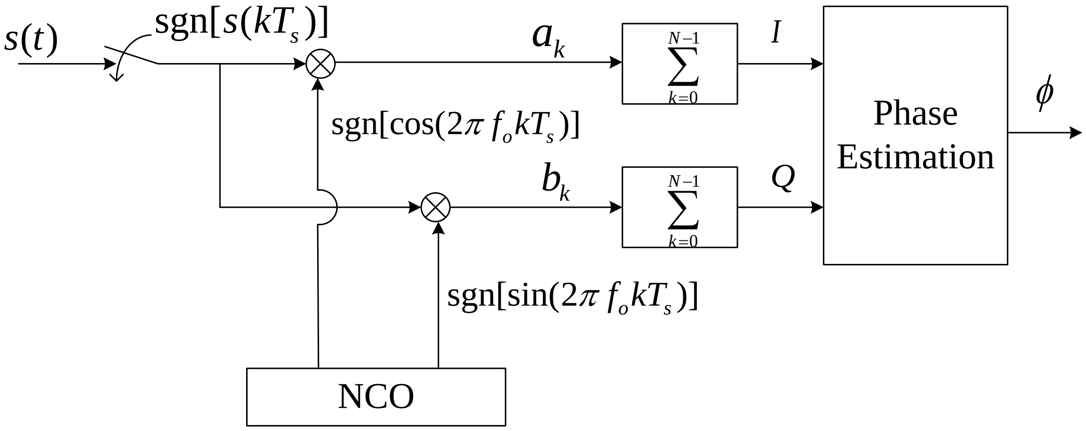

Fig. 2.5 One bit quantized DPD.

In order to design an ADPLL, we utilize the digital phase discriminator (DPD) instead of APD for the one bit quantized digital PLL. Before analyzing the system in Fig. 2.5, we would like to explain its operation principle. First, for phase estimation, it is usual to calculate the phase difference between the input signal and the output signal via the inphase branch (I) and the quadrature branch (Q). However, the most special setting for one bit quantized DPD is to accumulate one-bit samples a_k and b_k . As mentioned previously, we can not use only one bit to distinguish the phase between the input signal and the output signal. For gaining more information from one bit quantization, we accumulate a_k and b_k N times to obtain I and Q. Therefore, I and Q will have enough information for accurate phase estimation.

Consider an input signal $s(t)$ given as

$$s(t) = \cos(2\pi f_c t + \theta), \tag{2.3}$$

where f_c is the carrier frequency, θ is an unknown initial phase. Besides, we assume that f_o is equal to f_c, where f_o is the output frequency of NCO. In Fig. 2.5, the signal a_k is given as

$$\begin{aligned} a_k &= \operatorname{sgn}[s(kT_s)]\cdot\operatorname{sgn}[\cos(2\pi f_c kT_s)] \\ &= \operatorname{sgn}[\cos(2\pi f_c kT_s+\theta)\cdot\cos(2\pi f_c kT_s)] \\ &= \operatorname{sgn}[\cos\theta+\cos\varphi_k], \end{aligned} \tag{2.4}$$

where $\operatorname{sgn}[x]$ is the polarity function, i.e., $\operatorname{sgn}[x]=1$ if x>0, $\operatorname{sgn}[x]=-1$ if x<0 and $\operatorname{sgn}[x]=0$ if x=0; $\varphi_k = 4\pi f_c kT_s+\theta \mod 2\pi$, k=0,1…$N$-1 and N is the total number of samples within T. In Eq. (2.4), a_k can be referred to the polarity of a discrete sinusoid plus a DC bias specified by $\cos\theta$. It can be proved φ_k's are uniformly distributed over $[0,2\pi]$. Define the ratio of negative samples to the total number of samples within T, given by

$$\eta_i = \frac{N_-}{N}, \tag{2.5}$$

where N_- is the number of samples with $a_k=-1$.

From Fig. 2.6, when $0<\theta\le\pi$, $a_k=-1$ only if φ_k is within the interval $[\pi-\theta,\pi+\theta]$. Since φ_k's are uniformly distributed over $[0,2\pi]$, if N is large enough, the parameter η_i is approximated by

$$\eta_i \approx \frac{2\theta}{2\pi} = \frac{\theta}{\pi}. \tag{2.6}$$

In contrast, when $-\pi\le\theta\le 0$, $a_k=-1$ only if φ_k is within the interval $[\pi+\theta,\pi-\theta]$. In this case,

$$\eta_i \approx \frac{-2\theta}{2\pi} = \frac{-\theta}{\pi}. \tag{2.7}$$

From Eq. (2.6) and Eq. (2.7), θ can be derived from η_i with polarity ambiguity. The ambiguity can be resolved with the Q-channel signal. In the similar process, we obtain

$$b_k = \operatorname{sgn}[-\sin\theta+\sin\varphi_k], \tag{2.8}$$

where $\varphi_k = 4\pi f_c kT_s+\theta \mod 2\pi$ and k=0,1…N-1.

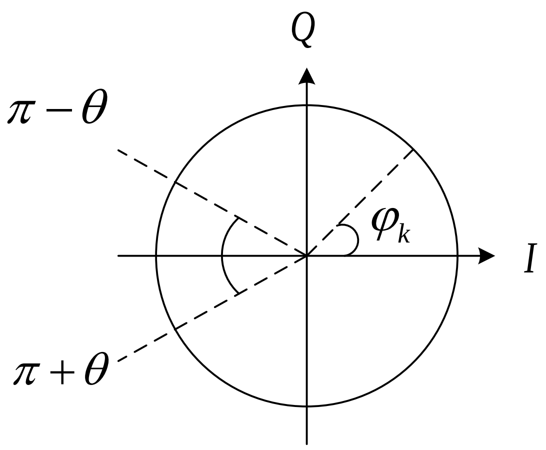

Fig. 2.6 The phase discriminator of one bit quantized software-defined receiver

Define

$$\eta_q = \frac{N'_-}{N}, \tag{2.9}$$

where N'_- is the number of samples with $b_k = -1$. We obtain

$$\eta_q \approx \frac{1}{2} + \frac{\theta}{\pi} \quad \text{if} \quad -\frac{\pi}{2} \le \theta < \frac{\pi}{2}; \tag{2.10}$$

$$\eta_q \approx \frac{3}{2} - \frac{\theta}{\pi} \quad \text{if} \quad \frac{\pi}{2} \le \theta < \frac{3\pi}{2}. \tag{2.11}$$

From Eq. (2.10) and Eq. (2.11), we have

$$\eta_q > \frac{1}{2} \quad \text{if} \quad 0 < \theta \le \pi; \tag{2.12}$$

$$\eta_q \le \frac{1}{2} \quad \text{if} \quad -\pi < \theta \le 0. \tag{2.13}$$

As a result, the polarity of θ can be acquired directly from η_q. From Eqs. (2.6)-(2.7) and Eqs. (2.12)-(2.13), the phase θ can be obtained by

$$\theta = \operatorname{sgn}(\eta_q - \frac{1}{2}) \cdot \eta_i \pi. \tag{2.14}$$

In addition, from the implementation point of view, the two negative sample counters can be obtained from summation output I and Q in Fig. 2.5, that is,

$$\eta_i = \frac{N-I}{2N} \quad , \quad \eta_q = \frac{N-Q}{2N}. \tag{2.15}$$

Thus, the estimated phase can be derived by

$$\theta = -\mathrm{sgn}(Q)\cdot\frac{\pi}{2}[1-\frac{I}{N}],\ \ \theta\in[-\pi,\pi] \tag{2.16}$$

which makes the proposed approach easily fit in the traditional structure.

Furthermore, even though the initial frequency of input signal and NCO output signal are different, we still can distinguish the phase difference which is caused by the different frequency of them. We can apply this characteristic to the one bit quantized PLL.

When f_o is not equal to f_c, the estimated phase can be given by

$$\phi = [\frac{2\pi(f_c - f_o)NT_s}{2}+\theta] = -\mathrm{sgn}(Q)\cdot\frac{\pi}{2}[1-\frac{I}{N}],\ \ \phi\in[-\pi,\pi] \tag{2.17}$$

where $\frac{2\pi(f_c - f_o)NT_s}{2}$ is the average phase difference caused by the frequency difference between input signal and NCO.

2.2.2. Digital loop filter

In analog PLL, we implement loop filter by low-passed filter in most cases. Normally, we use an integrator to implement a low-passed filter. For a digital PLL, since all data are discrete, we replace the integrator with an accumulator [8]. As shown in Fig. 2.7, the relationship between input and output is written as

$$\Omega_o(m) = \Omega_i(m) + \Omega_o(m-1)\,, \tag{2.18}$$

where $\Omega_i(m)$ is the input of the accumulator at time *m*, $\Omega_o(m)$ is the output and $\Omega_o(m-1)$ is the output of the accumulator at time *m*-1.

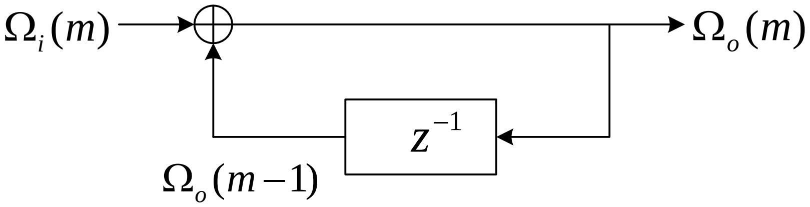

Fig. 2.7 Configuration of $\Omega_o(m) = \Omega_i(m) + \Omega_o(m-1)$.

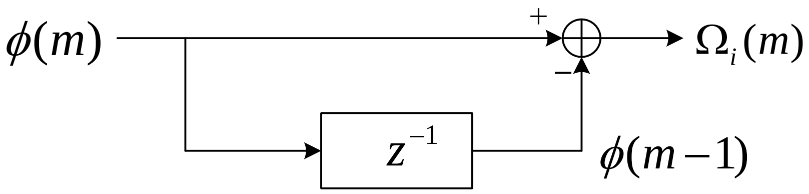

Fig. 2.8 Configuration of $\Omega_i(m) = \phi(m) - \phi(m-1)$.

As will be clear later, the purpose of the accumulator is to attenuate signals that emanated from the phase estimator, and adjust the input of NCO in finer resolution. Moreover, we add a function block between the DPD and the loop filter, as shown in Fig. 2.8.

The polarity of $\Omega_i(m)$ is determined by the relative magnitude of $\phi(m)$ and $\phi(m-1)$. In other words, the polarity of $\Omega_i(m)$ provides an information for the NCO to adjust its frequency at time *m*. The further details will be presented on sec. 3.1.

Finally, we combine above blocks and construct the one bit quantized PLL, as shown in Fig. 2.9,

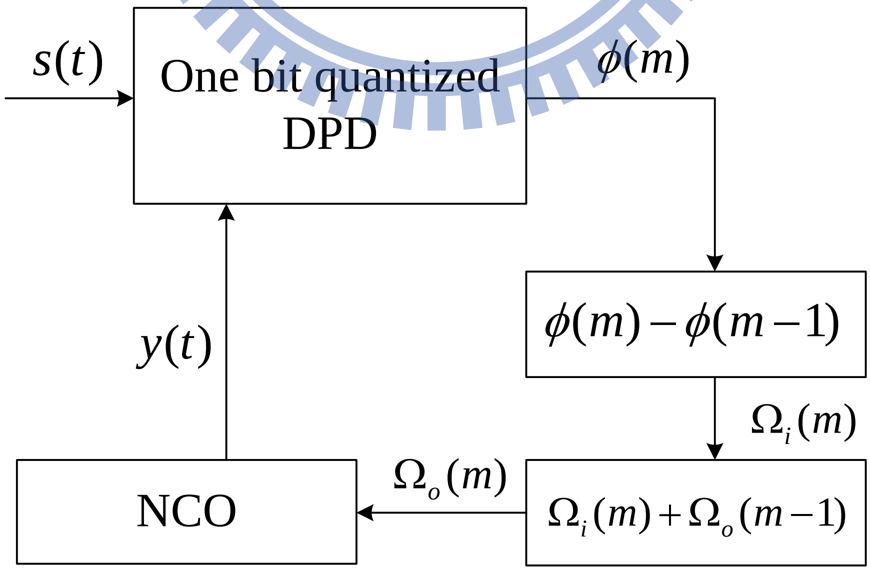

Fig. 2.9 One bit quantized digital PLL.

2.3. Summary

In this chapter, we introduce a general block diagram of PLL and explain the function of each block. Besides, we know that ADPLL is easier to design than the analog PLL. For reducing the complexity of ADPLL, we employ one bit quantization. Then, we discuss the one bit quantization in PLL and explain one bit phase estimation.

As we know, PLLs are widely used in tracking systems, and they usually have narrow capture range. The characteristic is based on analog PLL. However, even though ADPLL had been investigated, their capture range is still limited. Next, we will study how to use ADPLL to achieve ultra-wide capture range.

Chapter 3.

One Bit Quantized Digital PLL in Noiseless Environment

As we know, most of the PLLs work in a limited capture range. Take the popular PLL IC CD4046B for example; its capture range is about 10% of the natural frequency. In this chapter, we focus on the capture range and propose a new method to reach an ultra-wide capture range in one-bit quantized PLL. The work is done without any prior information about input signal frequency.

3.1. Frequency detection

By definition, the phase error of PLL is zero or constant when in lock, and the frequency of input signal and output signal (of NCO) are the same as well. Here, we discuss the method to adjust the NCO frequency so as to lock the input signal frequency.

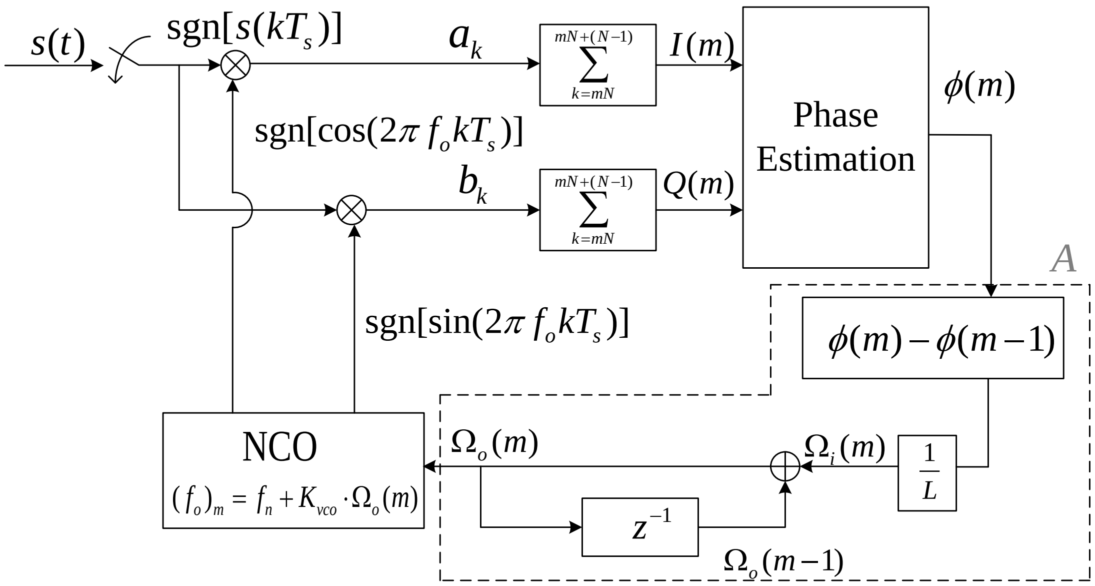

Fig. 3.1 One bit quantized digital PLL in details.

The signals within the one-bit quantized PLL circuit are defined as follows:

- The input signal $s(t)=\cos(2\pi f_c t+\theta)$, where f_c is the carrier frequency and θ is the initial phase.
- The output signals of NCO are $\cos(2\pi f_o t)$ and $\sin(2\pi f_o t)$, where f_o is the NCO output frequency.
- T_s is the sampling period.
- The multiplication of quantized input signal and quantized output signal (cosine branch) is a_k, where $k = mN + i$, $i = 0,1,\ldots, N-1$ and $m = 0,1,2,\ldots$.
- The multiplication of quantized input signal and quantized output signal (sine branch) is b_k, where $k = mN + i$, $i = 0,1,\ldots, N-1$ and $m = 0,1,2,\ldots$.
- $I(m)$ is the I-channel output, $I(m)=\sum_{k=mN}^{mN+(N-1)} a_k$.
- $Q(m)$ is the Q-channel output, $Q(m)=\sum_{k=mN}^{mN+(N-1)} b_k$.
- The phase difference (phase error) between input and output signal is $\phi(m)$, given by

$$\phi(m) = -\mathrm{sgn}[Q(m)] \cdot \frac{\pi}{2}[1 - \frac{I(m)}{N}] \text{ over } [-\pi, \pi]. \tag{3.1}$$

- $\Omega_i(m)$ is $\frac{1}{L}$ times of the difference between $\phi(m)$ and $\phi(m-1)$, where $\frac{1}{L}$ is the attenuation parameter. In our implement, $\frac{1}{L} = \frac{1}{10}$ is taken.
- $\Omega_o(m)$ is the input of NCO.
- f_n is the natural frequency of NCO and K_{VCO} is the conversion gain. Here we take $f_n = 10K(Hz)$ and $K_{VCO} = 10 \quad (rad^{-1})$.

As we see in Fig. 3.1, there is an algorithm for part A (enclosed by dashed line) to adjust the output frequency to match the input frequency. Note that the range of phase estimation is over $[-\pi, \pi]$ in Eq. (2.16). For better understanding, we illustrate the algorithm graphically as in Fig. 3.2-Fig. 3.4. First, we demonstrate the method as two runners on the playground. We would like to measure which one of the runners is faster or slower, and then adjust the velocity of the runner. Our purpose is to keep the two runners at the same velocity.

In the above case, suppose we can measure the distance between the two runners but can not measure the velocity of them. Besides, the velocity of runner B is adjustable while that of the runner A is fixed. Assume that ● is runner A and ◍ is the runner B whose velocity is adjustable.

Case Ⅰ:

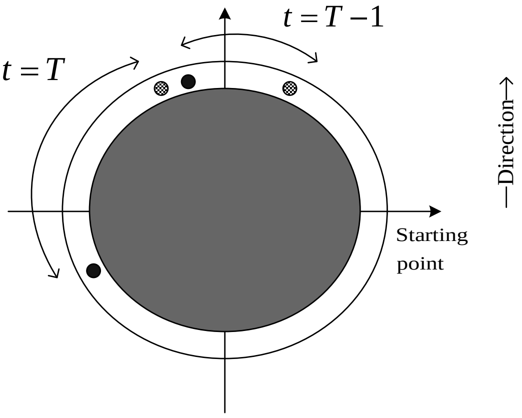

Fig. 3.2 Graphical illustration of Case Ⅰ.

As shown in Fig. 3.2, we observe that the distance between runner A and B at time T is longer than the one at time T-1. It means runner B is slower than runner A. To keep the two runners at the same pace, we increase the speed of runner B.

Case Ⅱ:

In contrast, as shown in Fig. 3.3, the distance of the two runners at time T is less than that at time T-1. To keep them in the same velocity, we slow down runner B.

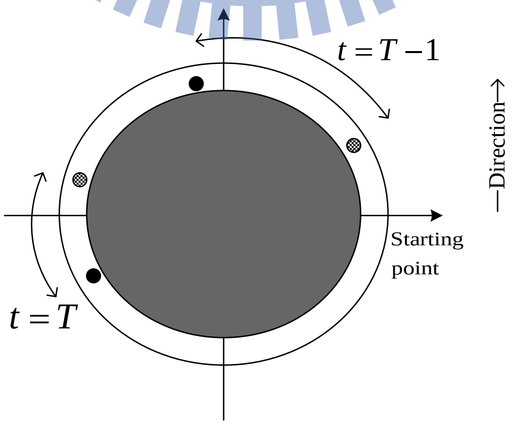

Fig. 3.3 Graphical illustration of Case Ⅱ.

Case Ⅲ:

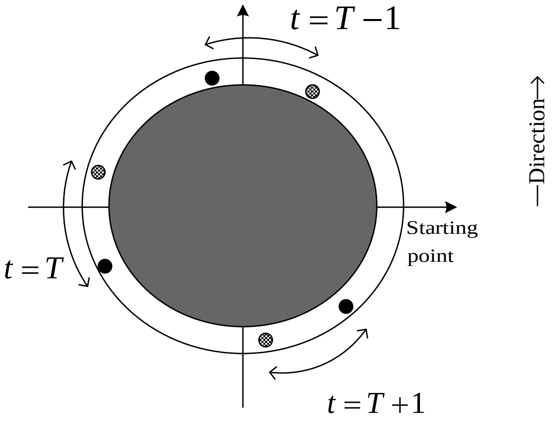

Fig. 3.4 Graphical conception of CaseⅢ.

In the case of Fig. 3.4, we notice that the distance between the runners is the same at any time. It means that the velocity of the runners is the same, so no modification is made.

Corresponding to Fig. 3.2-Fig. 3.4, we describe the phase-adjustment method in Fig. 3.5-Fig. 3.7. Similarly, the one bit quantized digital PLL can only measure:

1. The phase difference between input signal and NCO output signal at time m, which is presented by $\phi(m)$.
2. The difference between $\phi(m)$ and $\phi(m-1)$.

In the case of Fig. 3.5-Fig. 3.7, we assume the phase moves counterclockwise and the origin is at phase zero. Besides, the frequency of input signal is fixed and our goal is to adjust the frequency of NCO output signal until it matches input signal. Here, we may regard the input signal as the runner A, and the output signal of NCO as the runner B. The phase of input signal at time T is presented by the distance from the starting point to runner A at time T; the phase of NCO output signal corresponds to runner B. Therefore, the phase error $\phi(m)$ is the distance between the two runners at time m. The velocity is referred to the signal frequency. We describe the system again in these notations.

Case Ⅰ:

In Fig. 3.5, it shows that the phase error at time m ($\phi(m)$) is larger than the phase error at time m-1 ($\phi(m-1)$). It means the frequency of NCO is lower than the frequency of input signal, so we have to increase the frequency of NCO.

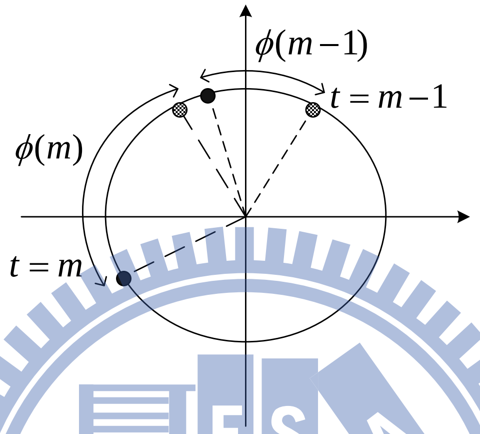

Fig. 3.5 The phase variation with time, where the angle of ● is the phase of input signal, the angle of ▨ is the phase of NCO output signal.

Case Ⅱ:

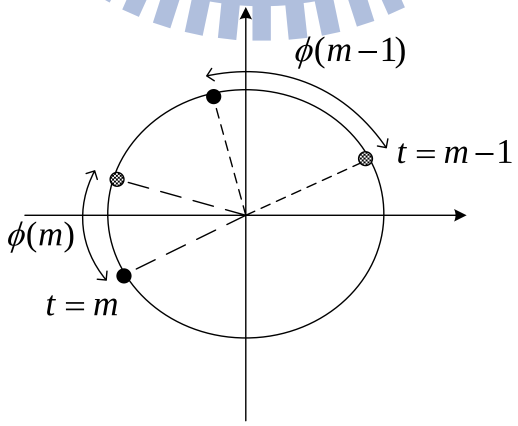

Fig. 3.6 The phase variation with time, where the angle of ● is the phase of input signal , the angle of ▨ is the phase of NCO output signal.

In this case, $\phi(m)$ is smaller than $\phi(m-1)$ as shown in Fig. 3.6. The phenomenon represents that the frequency of NCO output signal is higher than the frequency of input signal. Hence, we decrease the frequency of NCO.

Case Ⅲ:

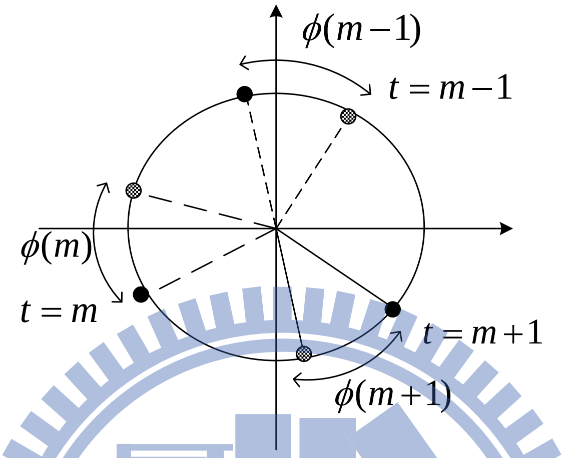

Fig. 3.7 The phase variation with time, where the angle of ● is the phase of input signal, the angle of ◍ is the phase of NCO output signal.

In this case**,** Fig. 3.7 shows that the phase error keeps the same at time m-1, m and m+1. According to the definition of locked state, it means the PLL is in lock. Thus, we do not have to adjust NCO frequency.

As a result, the operation principle of part A in Fig. 3.1 can be shown as

$$\text{if } \ \phi(m)-\phi(m-1)\ge 0 \ \text{ then } \ \Omega_i(m)\ge 0, \tag{3.2}$$

$$\text{if } \ \phi(m)-\phi(m-1)< 0 \ \text{ then } \ \Omega_i(m)< 0. \tag{3.3}$$

If $\Omega_i(m)\ge 0$, the frequency of NCO will be slightly increased after $\Omega_i(m)$ passes through an accumulator which is made for a finer resolution. On the contrary, if $\Omega_i(m)<0$, the frequency of NCO will be slightly decreased. Besides, there is a limitation on $\phi(m)-\phi(m-1)$. To distinguish the polarity of $\phi(m)-\phi(m-1)$, we have to limit $\phi(m)-\phi(m-1)$ over $[-\pi,\pi]$. In other words, we have to revise Eq. (3.2)-(3.3) as

$$\text{if } \ 0 \le \phi(m) - \phi(m-1) < \pi \ \text{ then } \ \Omega_i(m) \ge 0, \tag{3.4}$$

$$\text{if } \ -\pi \le \phi(m) - \phi(m-1) < 0 \ \text{ then } \ \Omega_i(m) < 0. \tag{3.5}$$

However, because $\phi(m)$ is over $[-\pi, \pi]$, therefore $\phi(m) - \phi(m-1)$ may have problems when the adjacent $\phi(m)$ and $\phi(m-1)$ cross the boundary. Thus, we modify the value of $\phi(m) - \phi(m-1)$ if it is out of $[-\pi, \pi]$. The modifications are

$$\text{if } \ -2\pi \le \phi(m) - \phi(m-1) < -\pi$$

$$\text{then } \ \phi(m) - \phi(m-1) = \phi(m) - \phi(m-1) + 2\pi, \tag{3.6}$$

and

$$\text{if } \ \pi < \phi(m) - \phi(m-1) \le 2\pi$$

$$\text{then } \ \phi(m) - \phi(m-1) = \phi(m) - \phi(m-1) - 2\pi. \tag{3.7}$$

3.2. Analysis of one bit digital quantized PLL with ultra-wide capture range

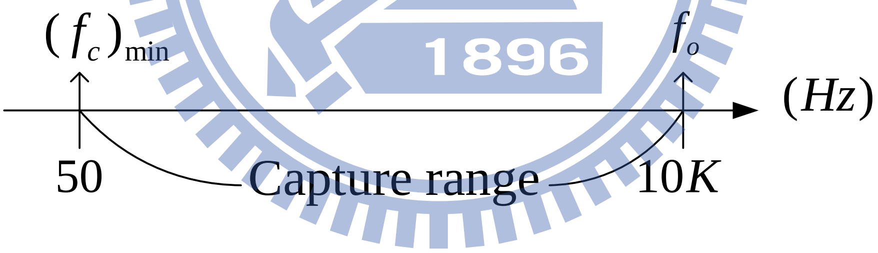

Fig. 3.8 The expected capture range of the thesis.

In the thesis, we would like to implement a one bit quantized digital PLL with ultra-wide capture range. Specifically, our purpose is to obtain the capture range to be over 90% of the natural frequency.

As shown in Fig. 3.8, f_c is the input frequency we would like to track, f_o is the NCO output frequency and the initial value of f_o is 10K Hz. In our system, f_o can track f_c without any prior information about f_c and f_c could be any value between 50 Hz and 10K Hz.

The tracking procedure of our system is mainly separated into two steps:

Step 1. Information collection:

The frequency detection $\phi(m)-\phi(m-1)$ is not always accurate, so in the beginning we run the one bit quantized digital PLL many cycles (in our case, we take 200 cycles) without changing the output frequency f_o. The average frequency deviation of the first 200 cycles will be much more accurate.

Step 2. Fine frequency adjustment:

After step 1, we obtain an estimate of $\phi(m)-\phi(m-1)$. Then we take the advantage of digital PLL, which transfers the output frequency f_o from 10K Hz to around f_c at once. In other words, f_o is near f_c at the 201st cycle. Thus, we can use the method in Section 3.1 to slightly adjust NCO output frequency until f_o is in lock with f_c.

Parameter determination:

A. Sampling frequency:

The sampling frequency f_s is the inverse of sampling period T_s. We take f_s as about 320K Hz, for a fine resolution of initial f_o.

B. Accumulate N times in the beginning:

The value N is adaptive for different input and output frequency variation. However, we take *N* as a fixed value for the first 200 cycles mentioned in step 1. The followings are the process of the determination of *N*.

In our system, the first phase difference is shown as

$$\phi(1) = \frac{1}{2} \cdot 2\pi(f_o - f_c)NT_s + \theta \quad , \tag{3.8}$$

which contains the average phase detection from t=T_s to t=NT_s and the initial phase difference θ. The second phase difference is

$$\phi(2) = \frac{1}{2} \cdot 2\pi(f_o - f_c)NT_s + \phi(1), \tag{3.9}$$

where the initial phase becomes $\phi(1)$, and $\frac{1}{2} \cdot 2\pi(f_o - f_c)NT_s$ is the average phase deviation between t=$2NT_s$ and t=$(N+1)T_s$.

Besides, we assume f_o is always higher than f_c at the beginning, and the maximum range of $\phi(m) - \phi(m-1)$ is 2π. Therefore, we can infer that

$$\begin{aligned} &\phi(2) - \phi(1) = 2\pi(f_o - f_c)NT_s < 2\pi \\ &\Rightarrow (f_o - f_c)NT_s < 1 \\ &\Rightarrow N < \frac{1}{(f_o - f_c) \cdot T_s} \\ &\Rightarrow N < \frac{f_s}{(f_o - f_c)} \\ &\Rightarrow N < \frac{f_s}{f_o} \end{aligned} \tag{3.10}$$

If we take $N = \left\lfloor \frac{f_s}{f_o} \right\rfloor - 1$, then $\phi(m) - \phi(m-1) = 0$ may happen in the experiment. The result may cause a serious error in phase estimation. To avoid the problem, we take N as

$$\phi(2) - \phi(1) = 2\pi(f_o - f_c)NT_s < \pi + \frac{5}{8}\pi, \tag{3.11}$$

then

$$N = \left\lfloor \frac{13 f_s}{16 f_o} \right\rfloor. \tag{3.12}$$

In our implementation, if N is an even number, a serious problem may happen, that is, the phase estimation $\phi(m)$ would possibly be zero owing to

$$\operatorname{sgn}[Q(m)=0]=0,\ where\ Q(m)=\sum_{k=mN}^{mN+(N-1)} b_k\ . \tag{3.13}$$

Therefore, we take N as the maximum odd number of Eq. 3.12 in the simulation.

C. 200 cycles of information collection:

We would like to make sure that the number of sampling is sufficient for detecting the phase. At least, the total sampling time has to reach almost one period of every signal. The step makes sure that we can collect the phase information of four quadrants in every signal. In our case, the lowest possible frequency is 50 Hz, so we take 200 times information collection to confirm that we still have enough frequency variation information even if the frequency is only 50Hz. From Eq. (3.12).

$$2\pi NT_s \times \left(f_c\right)_{\min} \times 200 \approx 2\pi\ . \tag{3.14}$$

3.3. The tracking procedure

In this section, we will discuss the whole process of one bit quantized digital PLL, including details of every parameter. However, the different input frequency results in the different process. Here, we divide the input frequency into three parts and discuss the process of each situation.

Note that the following discussion is based on the assumption of f_o=10K Hz and $f_s \approx 320K$ Hz

A. Middle input frequency ragne

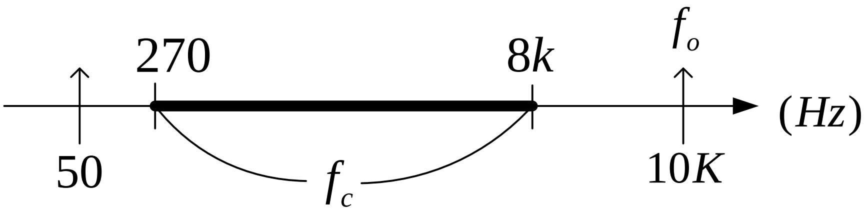

Fig. 3.9 General range of middle input frequency.

The range of middle input frequency is shown in Fig. 3.9, which is about 270 Hz to 8K Hz . The method we deal with middle input frequency is as mentioned in the beginning of Section 3.2.

Step 1. Information collection

The function of step 1 has been introduced previously, so we directly take N=25 by Eq. (3.11). The concept can be illustrated graphically in Fig. 3.10.

As shown in Fig. 3.10, we observe that the output frequency is unchanged when m=1~199. In the period, the system keeps accumulating $\phi(m)-\phi(m-1)$. Until m=200, we finish our last accumulation and measure an accurate value of Δf . We employ Δf in Eq. 3.15 as the input of NCO. After adjusted, the output frequency f_o is quite close to the input frequency f_c, where

$$\Delta f = \frac{1}{2\pi N T_s}\cdot\left\{\frac{1}{200}\cdot\sum_{m=1}^{200}[\phi(m)-\phi(m-1)]\right\}. \tag{3.15}$$

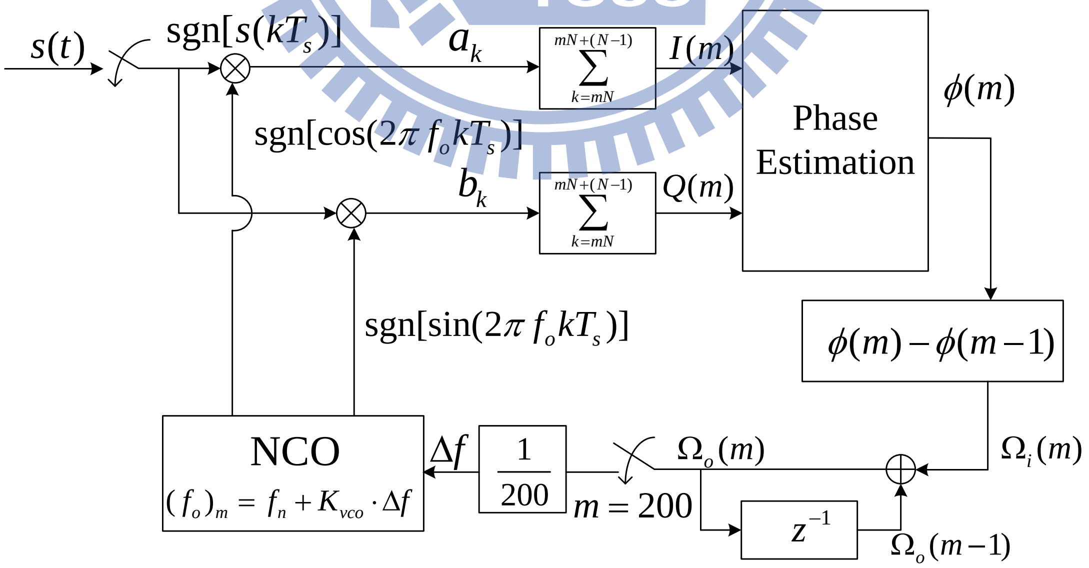

Fig. 3.10 The block diagram when m=1~200.

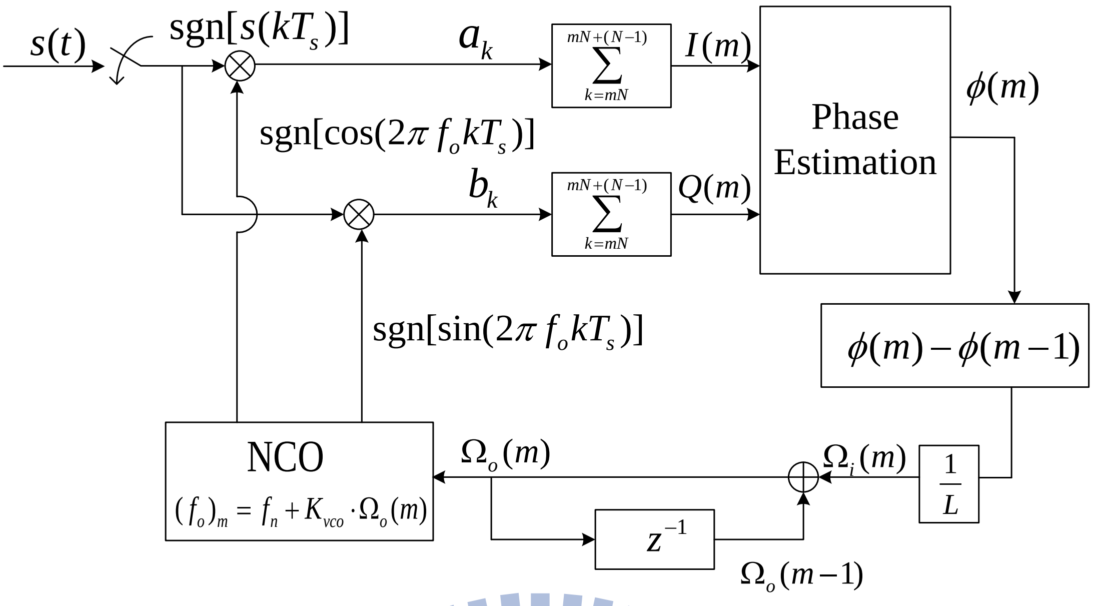

Fig. 3.11 The block diagram when m>200.

Step 2. Fine frequency adjustment:

Starting from m=201, the system enters another state which is one bit quantized digital PLL with narrow capture range as shown in Fig. 3.11. One parameter which we have to emphasize is N. Now, f_o is quite close to f_c, so we do not have to worry about the limitation of $\phi(m)-\phi(m-1)$. As a consequence, we take N=1001 to achieve a much finer resolution and assure the sampling time in one cycle is sufficient for phase estimation.

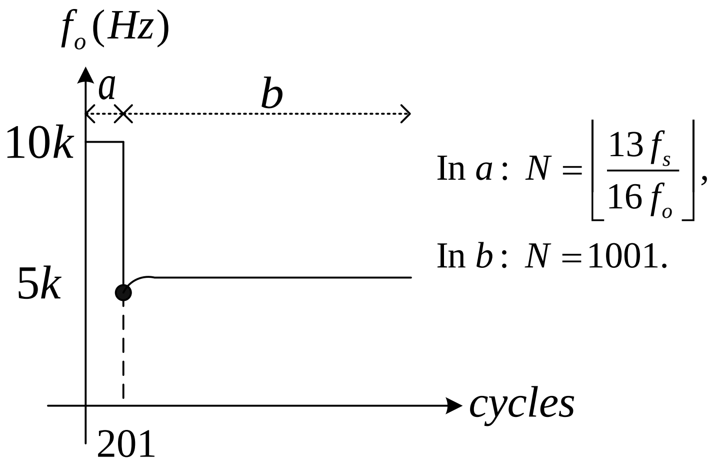

Fig. 3.12 Expected outcome with middle input frequency.

As shown in Fig. 3.12, the ideal outcome should graphically be similar to the figure.

B. Lower input frequency range

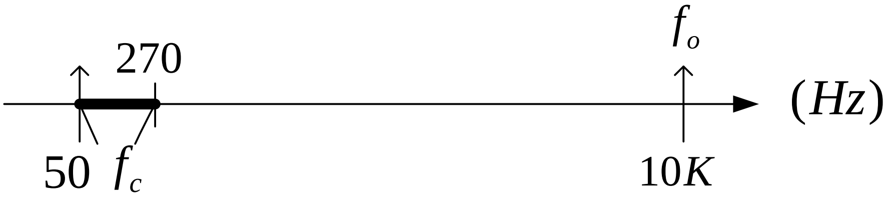

Fig. 3.13 General range of middle input frequency.

In this case, the range of lower input frequency as shown in Fig. 3.13 is around 50Hz to 270 Hz. The process of tracking the lower input frequency is quite similar to part A of Section 3.3.

Step 1. Information collection

As shown in Fig. 3.10, we observe that the output frequency is unchanged when m=1~199. In the period, the system keeps accumulating $\phi(m)-\phi(m-1)$.

Until m=200, we obtain an accurate value of Δf. We use Δf to be the input of NCO as Fig. 3.10. Afterwards, the output frequency f_o is very close to the input frequency f_c.

Step 2. Fine frequency adjustment:

Starting from m=201, the system enters another state which is one bit quantized digital PLL with narrow capture range as shown in Fig. 3.11. Still, we have to emphasize the parameter N. Now, f_o is quite close to f_c, so we do not need to consider the limitation of $\phi(m)-\phi(m-1)$. However, if we take N=1001, the sampling time in one cycle is not sufficient for lower frequency. For instance, assume that the input frequency f_c is 100 Hz and f_o is around 100 Hz. We want to sample the signal of 100 Hz to nearly a period in one cycle. Because the sampling frequency f_s is about 320K Hz, we have to take almost 3200 times. Obviously, N=1001 is not sufficient for lower frequency.

In this case, the parameter N is given by

$$N = \left\lfloor \frac{13 f_s}{16(f_o)_{201}} \right\rfloor \tag{3.16}$$

where $(f_o)_{201}$ is the value of f_o when m=201.

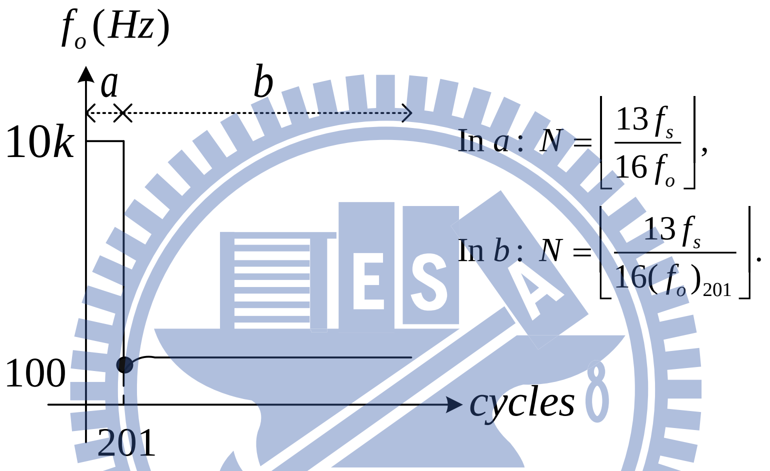
$f_o(Hz)$
a
b
$10k$
In a: $N = \left\lfloor \frac{13 f_s}{16 f_o} \right\rfloor$,
In b: $N = \left\lfloor \frac{13 f_s}{16(f_o)_{201}} \right\rfloor$.
100
$cycles$
201

Fig. 3.14 Expected outcome of one bit digital quantized PLL with lower input frequency.

As shown in Fig. 3.14, the ideal outcome should be graphically similar to the figure.

C. Higher input frequency range

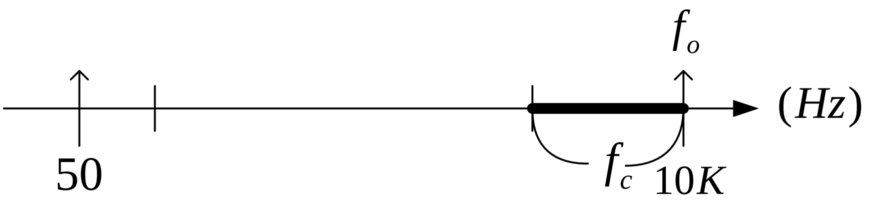

Fig. 3.15 General range of higher input frequency.

The following discussion is based on the input frequency which is located around 8K Hz to 10K Hz as shown in Fig. 3.15. The process of tracking higher input frequency is mainly split into three parts.

Step 1. Information collection

Repeating the process of step 1 in part A of Section 3.3, we would like to measure an accurate frequency difference between input and output signals. However, with higher input frequency, some problems may occur.

When m=1~200, we take N=25 as usual. It is a coarse resolution for phase estimation. This is also the reason why we have to take an average of phase estimation on the first 200 cycles. Consider the value of $\phi(m)-\phi(m-1)$, given by

$$\phi(m)-\phi(m-1)=2\pi(f_o-f_c)NT_s \text{ over } [0,2\pi]. \qquad (3.17)$$

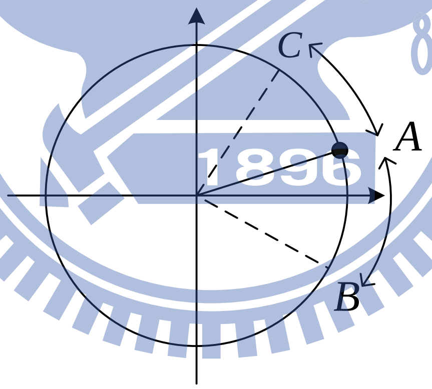

Fig. 3.16 Phase diagram of $\phi(m)-\phi(m-1)$.

For higher input frequency, the value of $\phi(m)-\phi(m-1)$ is quite small comparing with middle and lower input frequencies. Therefore, we might misjudge $\phi(m)-\phi(m-1)$. As shown in Fig. 3.16, the value A is the correct value of $\phi(m)-\phi(m-1)$. In the process of estimation, the misjudged value might locate between value B and value C. We take the value B for instance. Note that $\phi(m)-\phi(m-1)$ is only over $[0,2\pi]$. As long as we misjudge value A to value B, the system will present, for example, A= 0.2π but B= 1.8π. If we take the misjudged value into the estimation system, the outcome of Δf will be erroneous.

Step 2. Information re-collection

Therefore, we set a counter in the first 200 cycles. The counter is to calculate the number of $\phi(m)-\phi(m-1)\leq 50^{\circ}$. Once the number of $\phi(m)-\phi(m-1)\leq 50^{\circ}$ is over 70 within the first 200 cycles, we assume that the outcome of Δf will not be accurate.

To be more specific, we regard the input frequency as sufficiently high when the number of $\phi(m)-\phi(m-1)\leq 50^{\circ}$ is over 70 within the first 200 cycles. To avoid the mistake, we drop the outcome of Δf and re-estimate $\phi(m)-\phi(m-1)$ with longer sampling times in each cycle. In other words, we increase N as follows:

$$\begin{aligned}&\phi(m)-\phi(m-1)=2\pi(f_o-7.8K)NT_s<2\pi\\&\Rightarrow N=\left\lfloor\frac{f_s}{10K-7.8K}\right\rfloor-1.\end{aligned}\tag{3.18}$$

If we modify N as $2\pi(f_o-8K)NT_s<2\pi$, the value of $\phi(m)-\phi(m-1)$ might cross the boundary 0 and 2π when $f_c\approx 8K(Hz)$. On the other hand, when $f_c\approx 10\text{K}\ (Hz)$, N should be large enough to assure $\phi(m)-\phi(m-1)$ is sufficient. To satisfy the tradeoff, we take N with Eq. (3.18).

Besides, in the re-estimate part, we do not need 200 cycles to estimate $\phi(m)-\phi(m-1)$ because we have already known that input frequency is sufficient high. In other words, $(f_c)_{\min}$ in Eq. (3.14) is sufficient large. Therefore, we take 100 cycles to estimate $\phi(m)-\phi(m-1)$ in this case. The re-estimate part is graphically depicted in Fig. 3.17.

In Fig. 3.17, we observe that the output frequency is still fixed as original value when m=201~299. In the period, the system keeps accumulating $\phi(m)-\phi(m-1)$ and estimate $\Delta f'$ as

$$\Delta f' = \frac{1}{2\pi N T_s} \cdot \left\{ \frac{1}{100} \cdot \sum_{m=201}^{300} [\phi(m) - \phi(m-1)] \right\}. \tag{3.19}$$

When m=300, we finish our last accumulation and measure an accurate value of $\Delta f'$. We employ $\Delta f'$ as the input of NCO. After adjusted, the output frequency f_o is quite close to input frequency f_c.

In summary, there are two advantages to increase N in Step 2:

1. Get finer resolution: decrease the range of misjudging.

2. Keep the value of $\phi(m) - \phi(m-1)$ away from the boundary 0 and 2π.

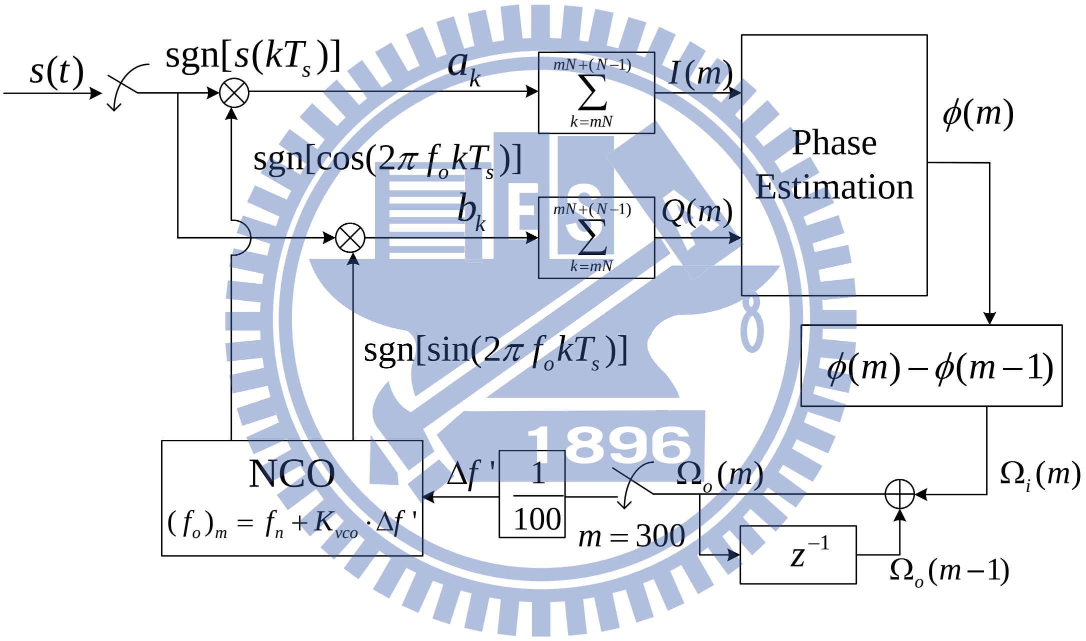

Fig. 3.17 The block diagram when m=201~300.

Step 3. Fine frequency adjustment:

The block diagram is shown in Fig. 3.11. The only parameter we revise here is that the block diagram occurs when m>300. Starting from m=301, the system is in another state which is one bit quantized digital PLL with narrow capture range. In the period, we take N=1001 for the same reason as before. After a few adjustments, the output frequency is supposed to be in lock with input frequency. As shown in Fig. 3.18, the ideal outcome should be graphically similar to the figure.

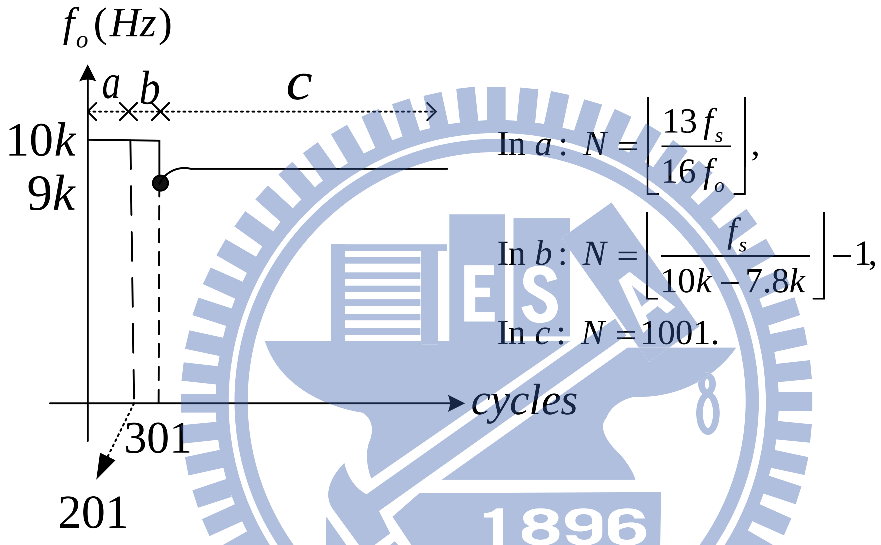
$f_o(Hz)$
a
b
c
$10k$
$9k$
In a: $N=\left\lfloor \frac{13f_s}{16f_o} \right\rfloor$,
In b: $N=\left\lfloor \frac{f_s}{10k-7.8k} \right\rfloor-1$,
In c: $N=1001$.
$cycles$
301
201

Fig. 3.18 Expected outcome of one bit quantized digital PLL with higher input frequency.

D. Summary

Fig. 3.19 is the flowchart of distinguishing f_c in the first 200 cycles. Define

m: number of cycles,

N: number of samples in each cycle,

counter: number of $\phi(m)-\phi(m-1)\le 50^\circ$ within the first 200 cycles.

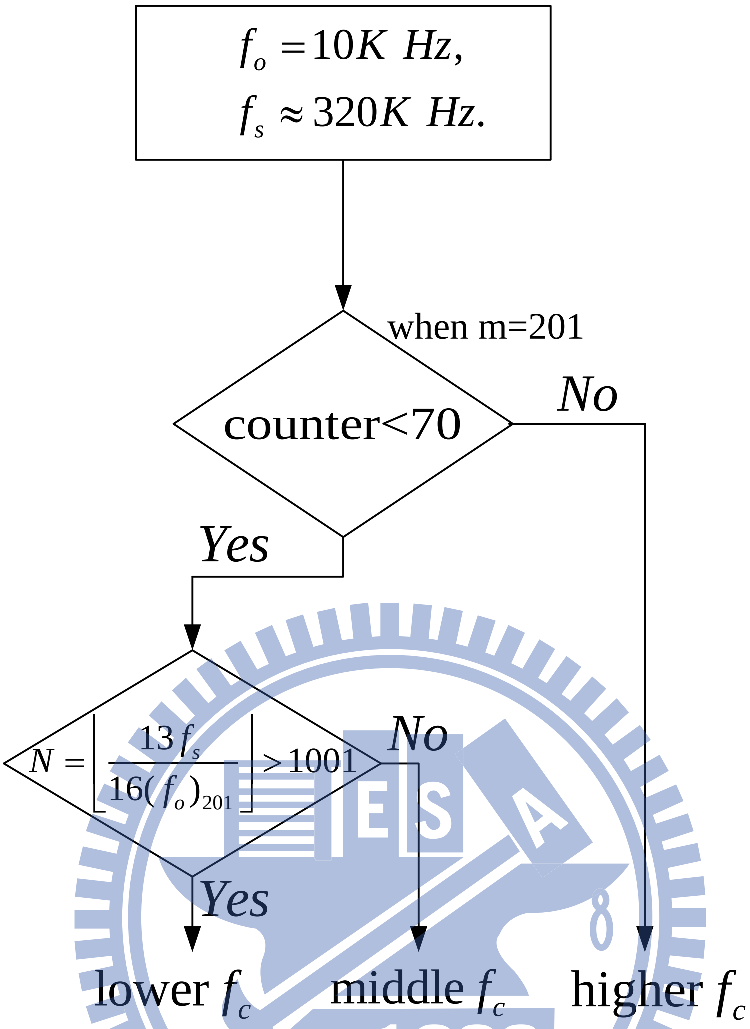

Fig. 3.19 Flowchart of distinguishing input frequency

After distinguishing the input frequency during the first 200 cycles, we start the process of fine frequency adjustment. Finally, we achieve the one bit quantized digital PLL with ultra-wide capture range.

3.4. Simulation Results

We simulate the system with MATLAB. The followings are simulation results of the one bit quantized digital PLL.

A. Middle input frequency range

Here we illustrate the tracking process for four values of f_c over middle frequency range and show the NCO output frequency at a specific cycle where X is the number of cycles and Y means the value of NCO output frequency in Hz.

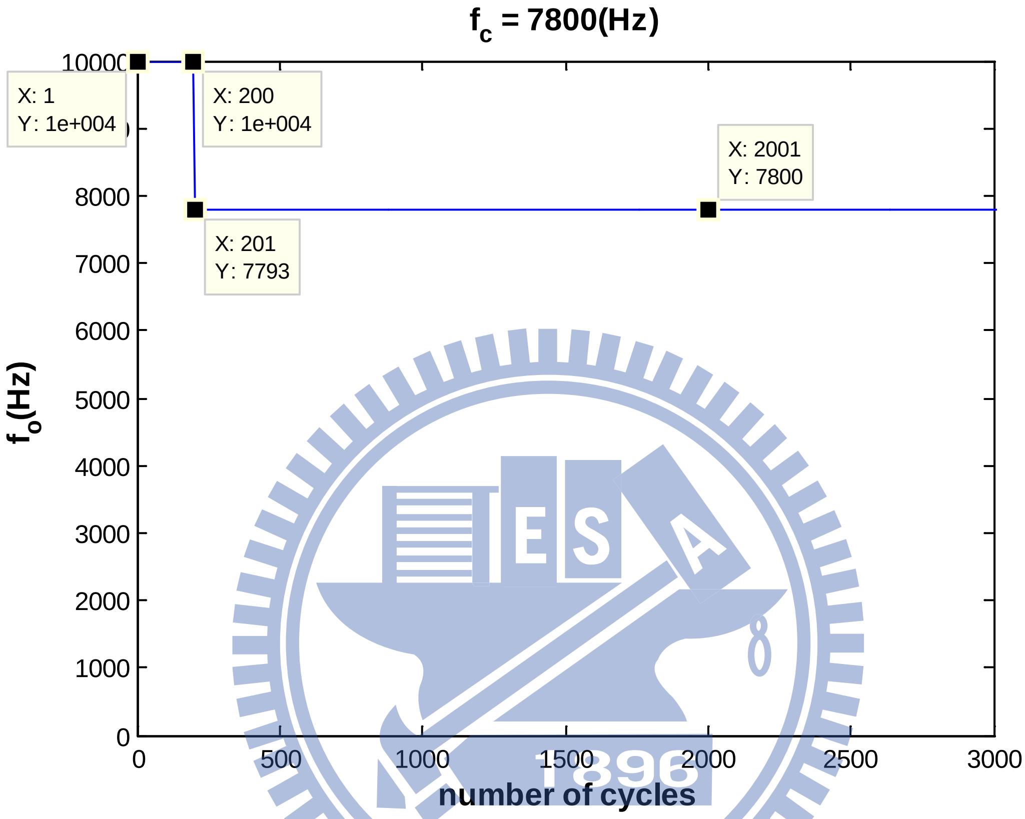

Fig. 3.20 The locking process of f_c 7800 Hz.

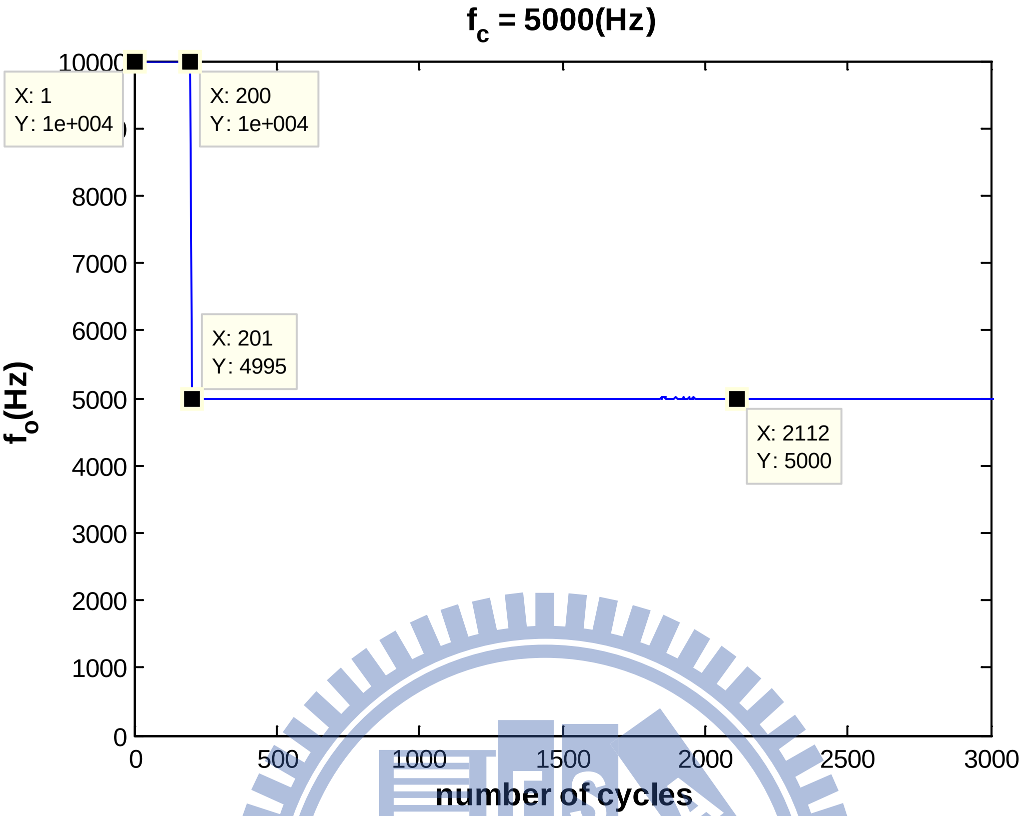

Fig. 3.21 The locking process of f_c 5000 Hz.

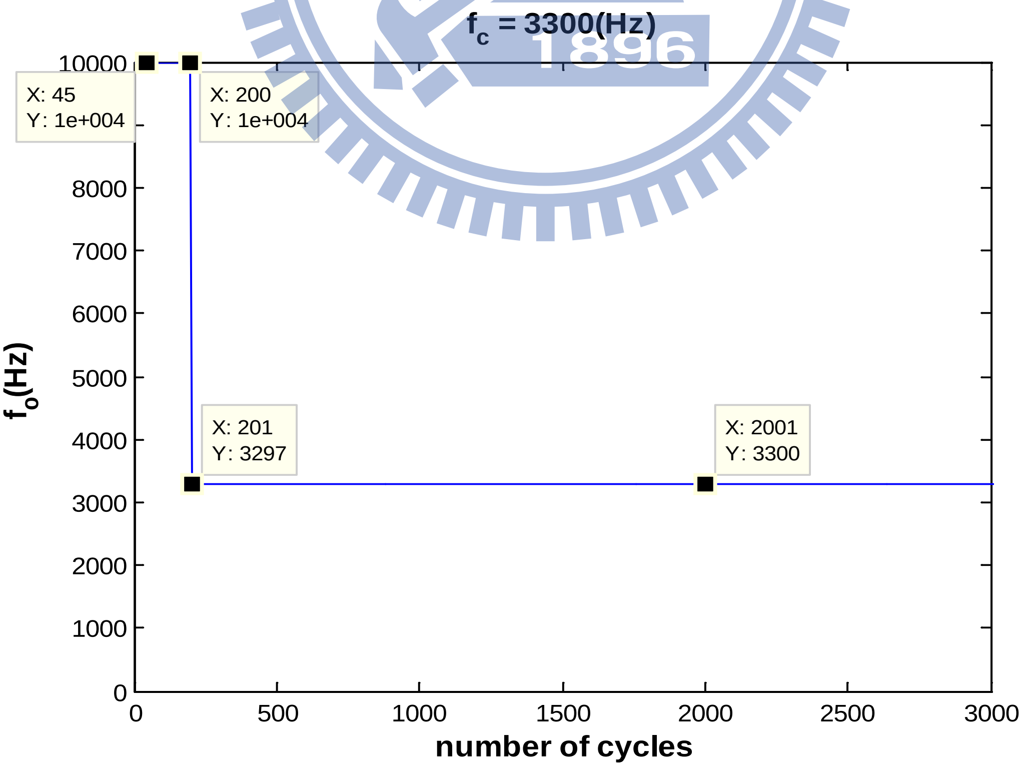

Fig. 3.22 The locking process of f_c 3300 Hz.

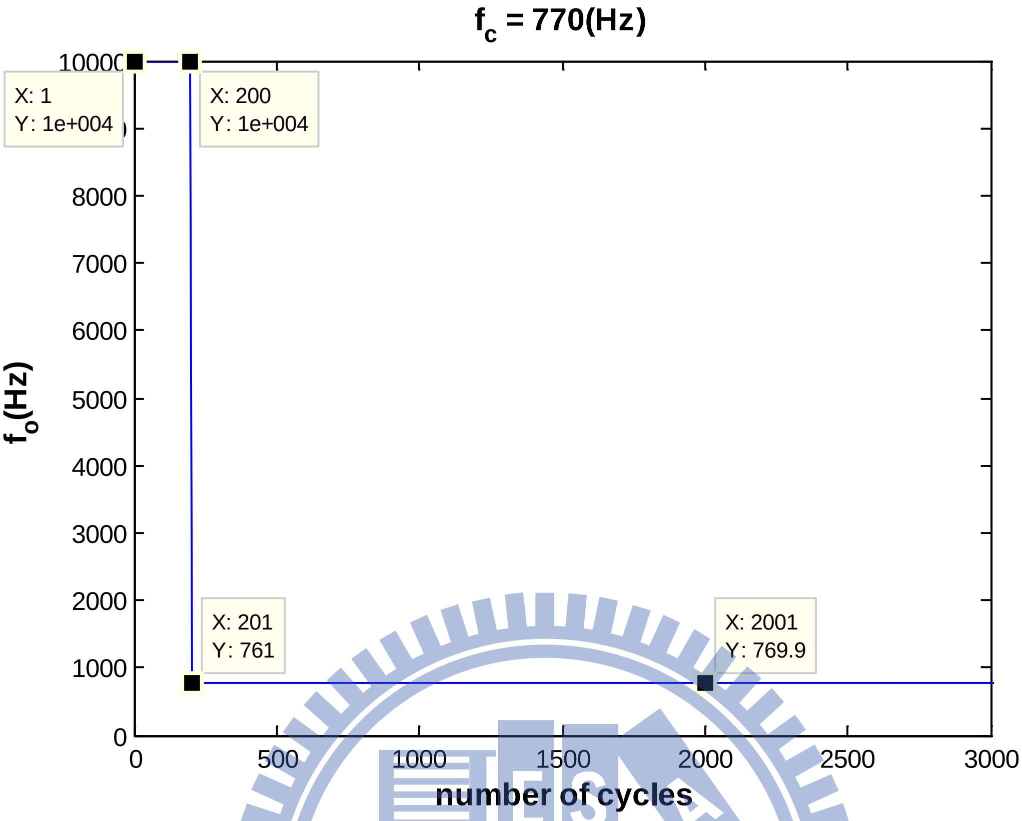

Fig. 3.23 The locking process of f_c 770 Hz.

B. Lower input frequency range

Here, we depict the locking procedure for four values of f_c over lower frequency range and show the output frequency at a specific cycle where X is the number of cycles and Y means the value of NCO output frequency in Hz.

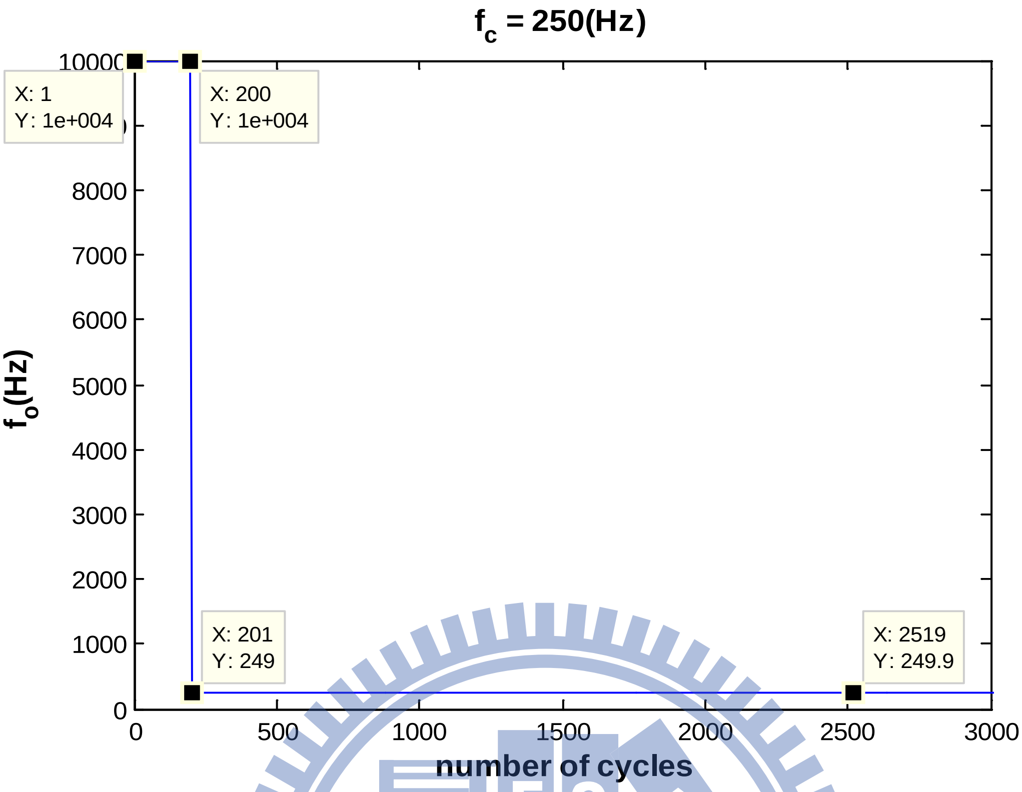

Fig. 3.24 The locking process of f_c 250 Hz.

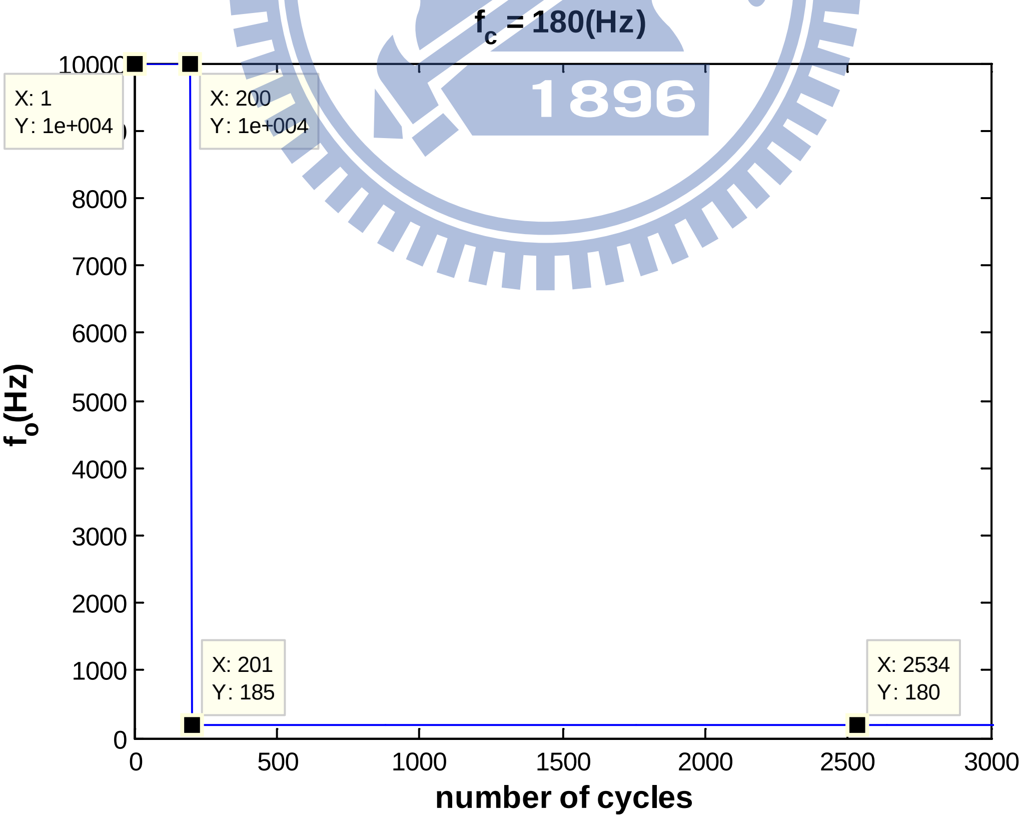

Fig. 3.25 The locking process of f_c 180 Hz.

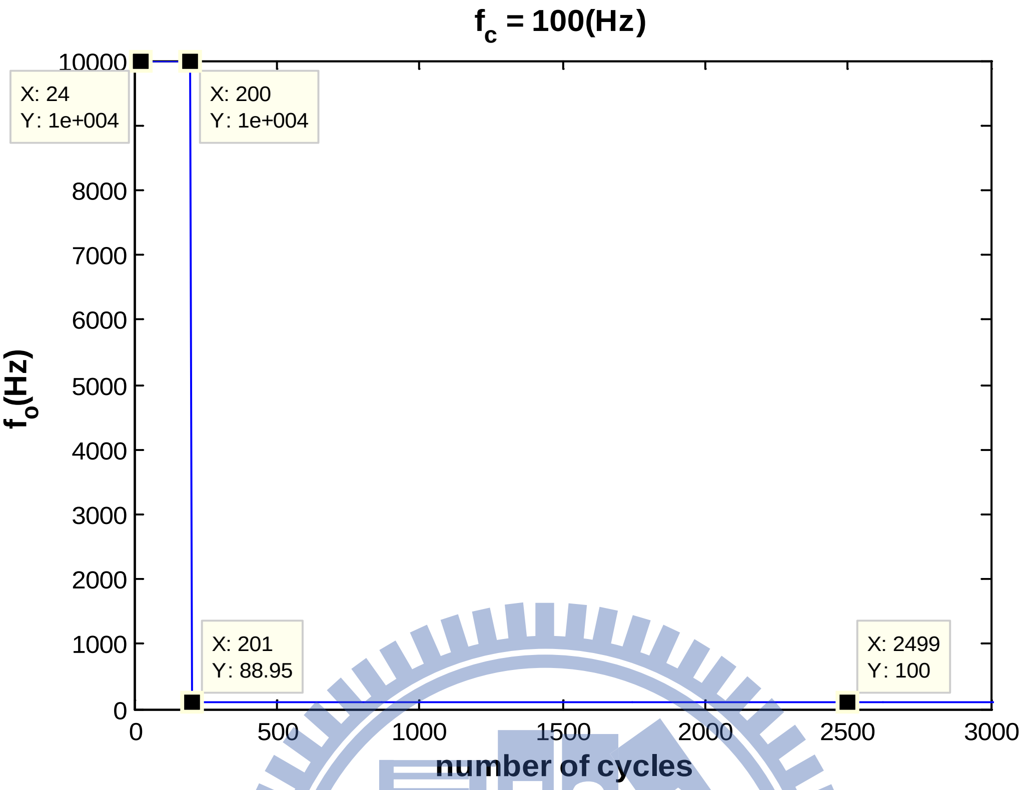

Fig. 3.26 The locking process of f_c 100 Hz.

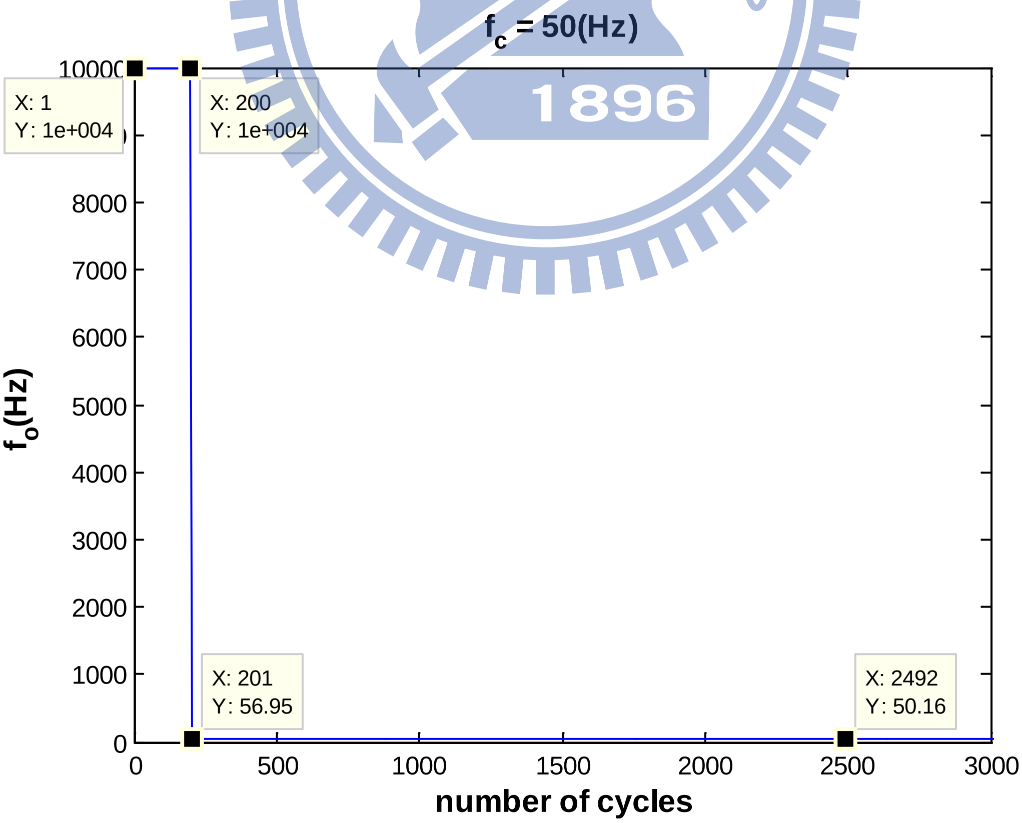

Fig. 3.27 The locking process of f_c 50 Hz.

C. Higher input frequency range

We illustrate the tracking process for four values of f_c over higher frequency and show the output frequency specifically at a specific cycle.

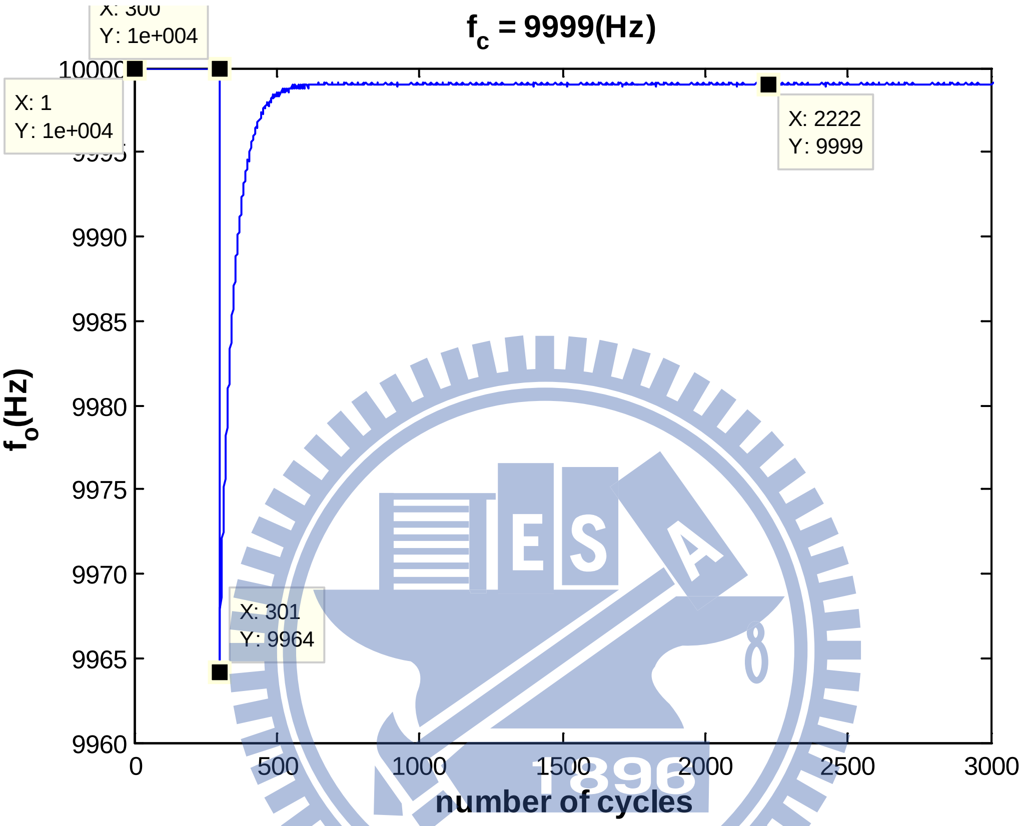

Fig. 3.28 The locking process of f_c 9999 Hz.

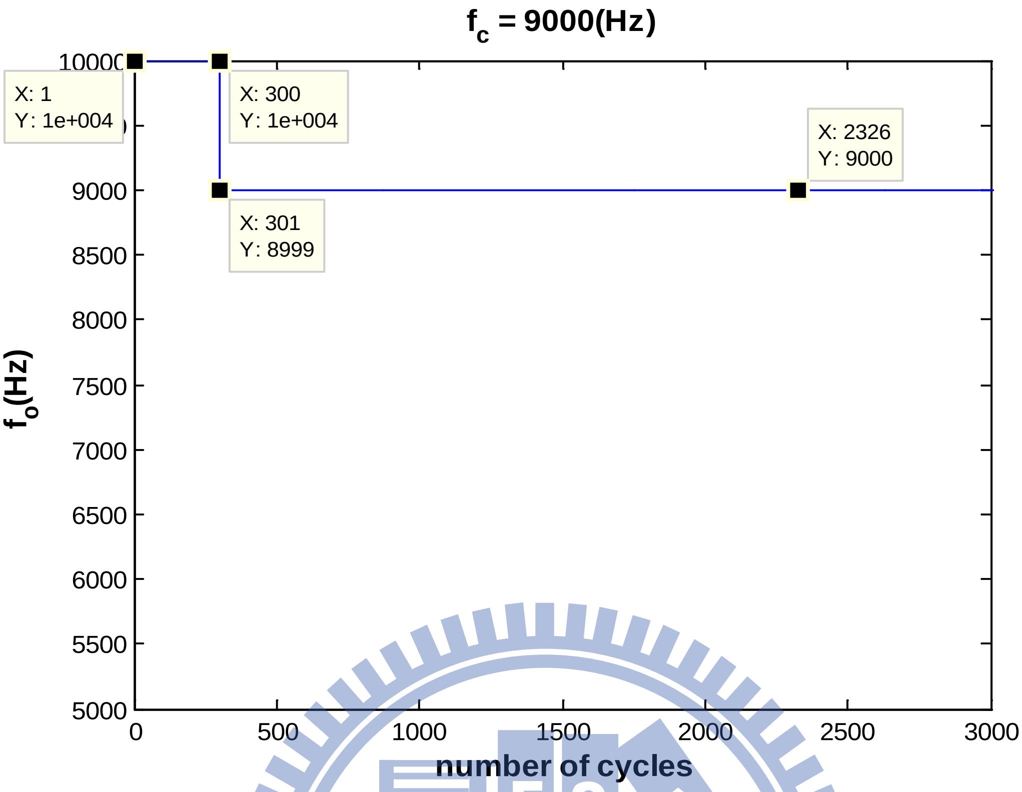

Fig. 3.29 The locking process of f_c 9000 Hz.

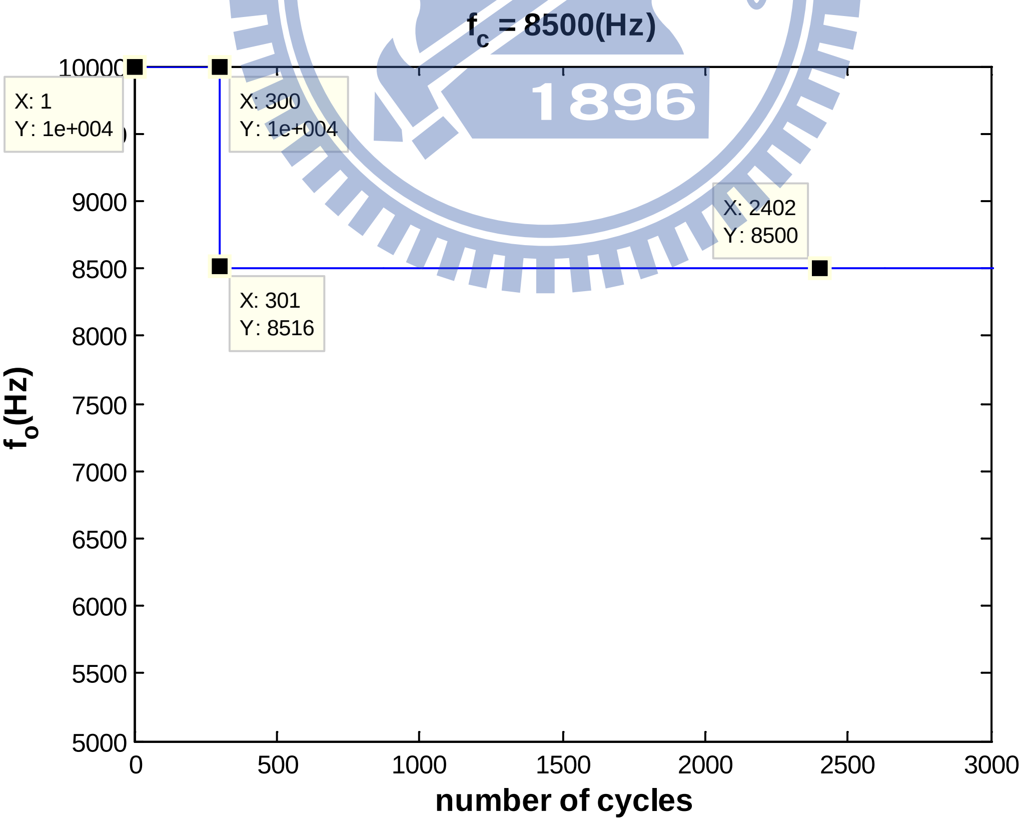

Fig. 3.30 The locking process of f_c 8500 Hz.

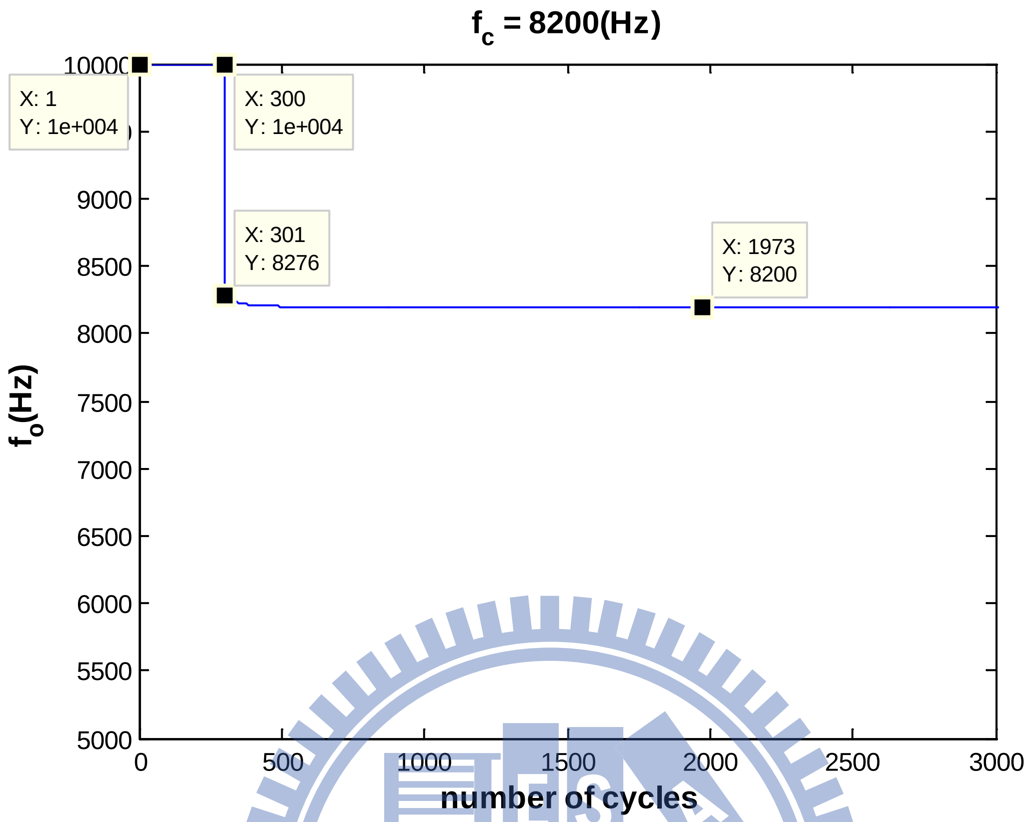

Fig. 3.31 The locking process of f_c 8200 Hz.

D. Simulation results of a ultra-wide capture range

Finally, we observe the locking results of a wide input frequency range. Instead of showing the locking process, we define that if the final value of f_o is located within $[f_c - 1, f_c + 1]$ when m=3000, the system is called “locked”. We just have to observe the value of f_o at m=3000, and then we can determine whether the system is in-lock or not.

We increase f_c logarithmically from 50 Hz to 9976 Hz in Fig. 3.32. On Y-axis, Y=1 indicates the system is locked and Y=0 shows the system fails to lock. As we see, our design can lock all the frequencies from 50 Hz to 9976 Hz successfully in noiseless environment.

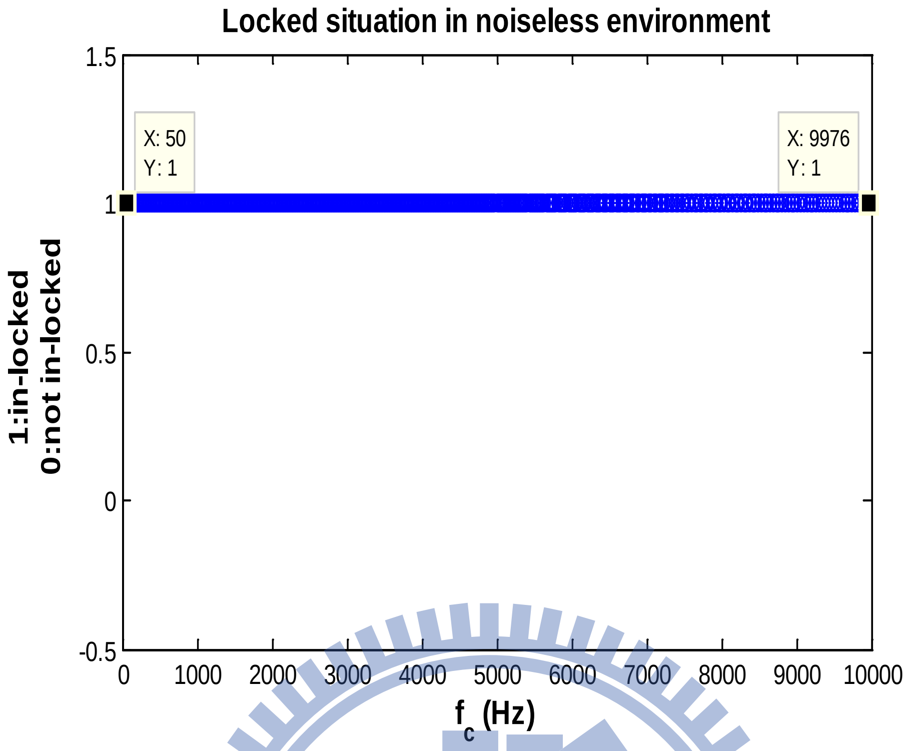

Fig. 3.32 Locked situation in noiseless environment.

3.5. Summary

In this chapter, we propose methods to deal with different input frequency signals and then demonstrate one bit quantized digital PLL with ultra-wide capture range. In the process, the number N is quite important for either the resolution or the error of phase estimation. Therefore, we modify N to fit different situations. The unique features of our system are:

1. f_c is an unknown frequency over [50 Hz, 10K Hz].

2. In noiseless environment, the system achieves almost 99.5% capture range of the natural frequency.

The outcomes of simulation justify our design in noiseless environment.

Chapter 4.

Noise effect

It is known that most of the PLL systems would be affected by noise in reality. In this chapter, we analyze the performance of one bit quantized digital PLL in noisy environment. Our discussion is based on the simulation outcomes of different signal-to-noise ratio (SNR) in dB. We use additive white Gaussian noise (AWGN) to simulate the noise affect in our system.

4.1. Overview

The model of one bit quantized digital PLL with noise is the same as that in Chapter 3. The only change is the input signal $s(t)$, given by

$$s(t) = \cos(2\pi f_c t + \theta) + n(t), \tag{4.1}$$

where $n(t)$ is the additive white Gaussian noise.

In addition, we relax the setting of determining whether the system is in-locked or not. In noiseless case, we find that different input frequency can almost be locked after m=1000. Therefore, the following discussion is based on calculating the mean-square-error (MSE) [9] between input frequency and output frequency from m=1001 to m=3000. If the value of MSE is smaller than 1 Hz, it is called “locked”. In other words, when the system satisfies

$$\sqrt{\frac{1}{2000}\left(\sum_{m=1001}^{3000}[(f_o)_m - f_c]^2\right)} \le 1\ (Hz), \tag{4.2}$$

then the system will be locked.

Besides, we transform the algorithm of phase estimation from Eq. (3.1) into

$$\phi(m) = -\text{sgn}[Q(m)] \cdot \frac{\pi}{2}[1 - \frac{I(m)}{|I(m)| + |Q(m)|}] \tag{4.3}$$

which has more accurate estimation in noisy environment than Eq. (3.1) [10].

Simulation results

We test f_c increasing logarithmically from 50 Hz to 9976 Hz in Fig. 4.1-Fig. 4.5. On Y axis of those figures, Y=1 means locked successfully and Y=0 means unlocked.

.

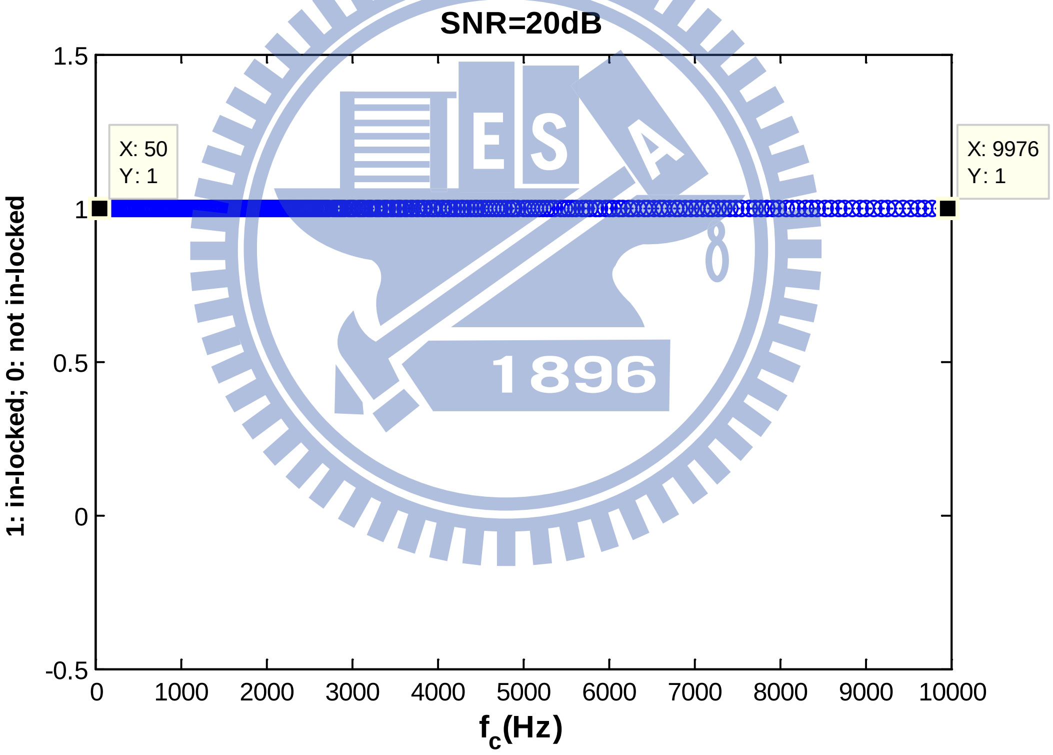

Fig. 4.1 Simulation results with SNR=20 dB.

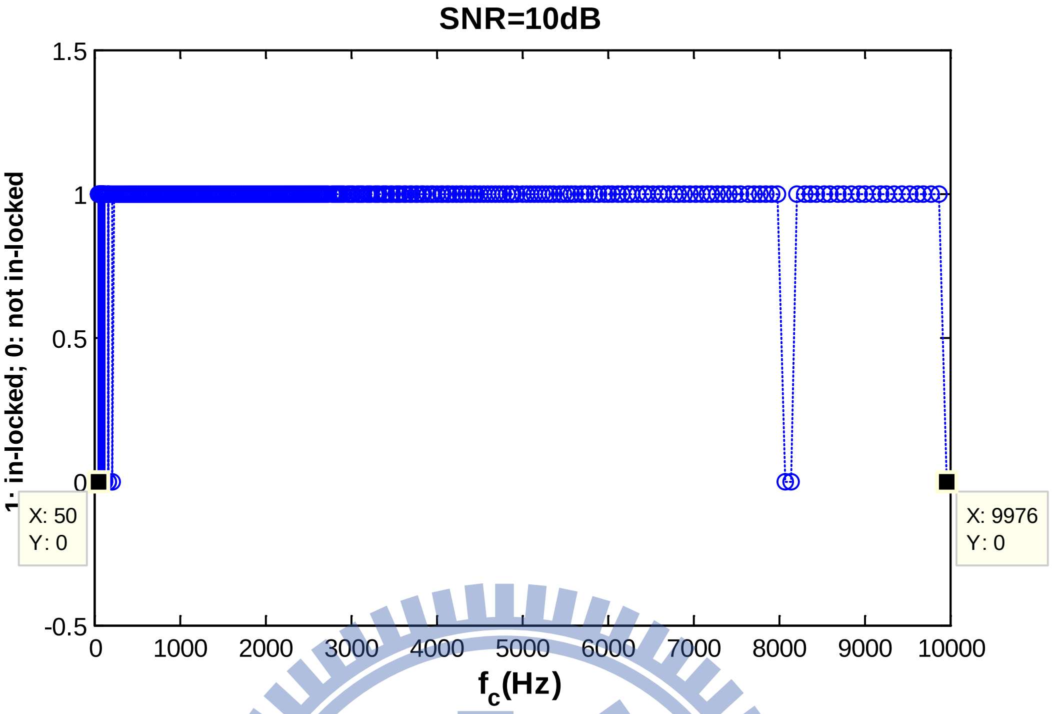

Fig. 4.2 Simulation results with SNR=10 dB.

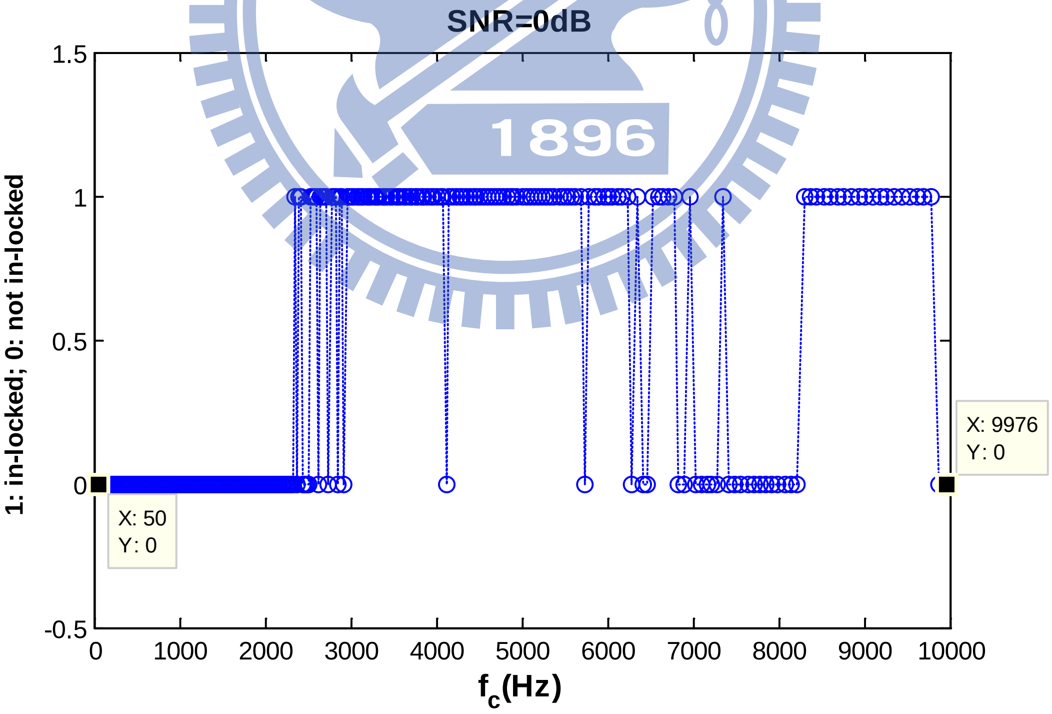

Fig. 4.3 Simulation results with SNR=0 dB.

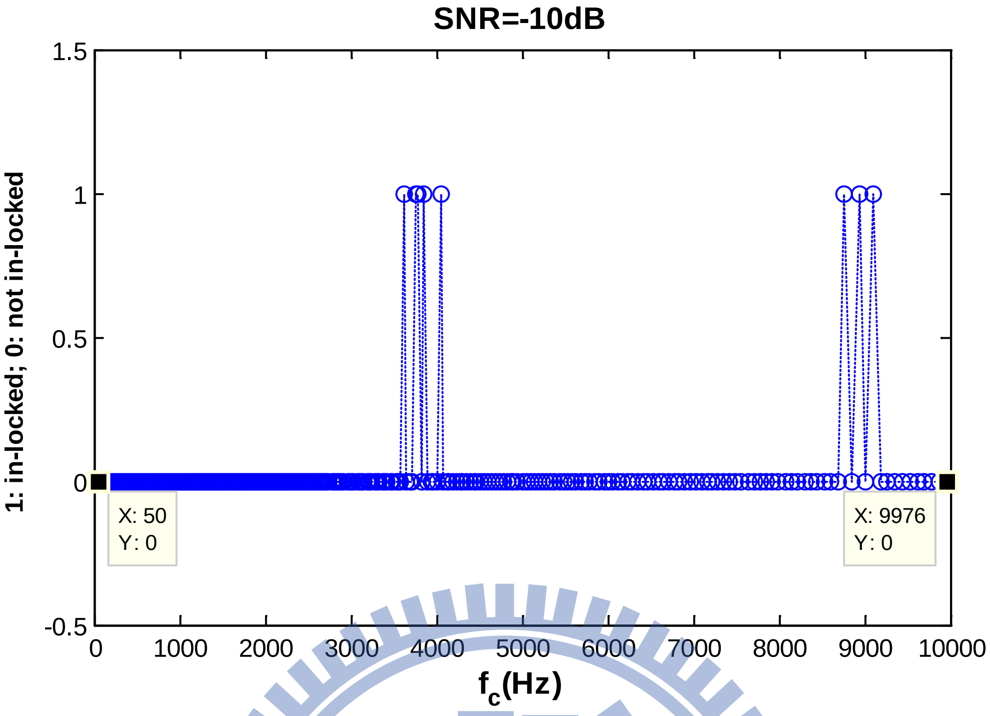

Fig. 4.4 Simulation results with SNR=-10 dB.

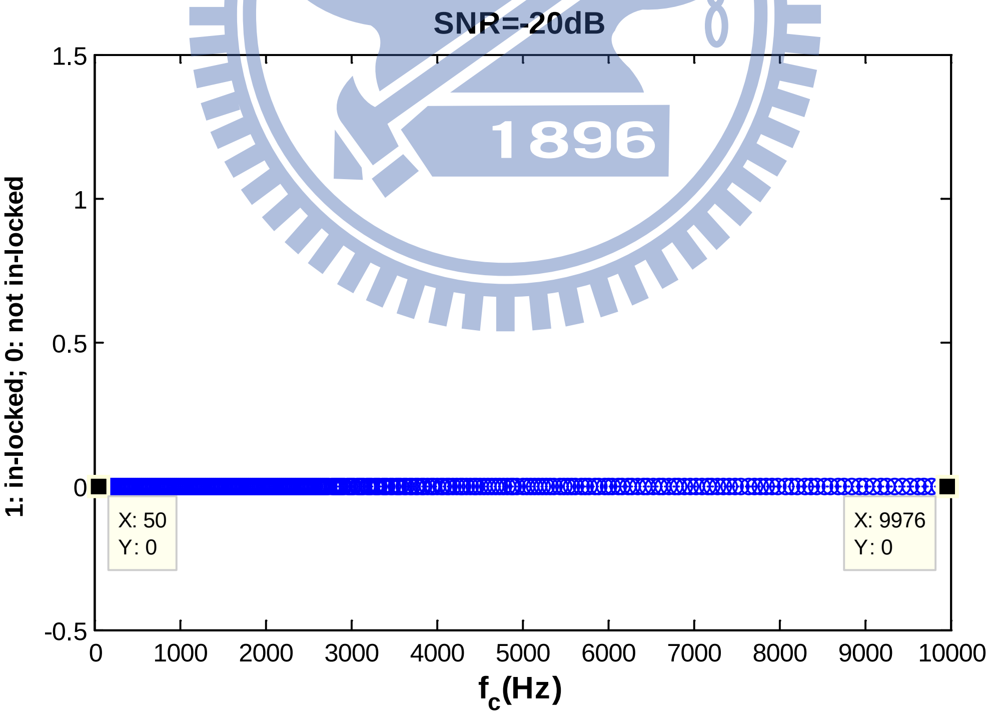

Fig. 4.5 Simulation results with SNR=-20 dB.

As we see, the higher the SNR, the better the performance. We will discuss the details in the following section.

4.2. Noise effect

As shown in Fig. 4.1, we found that f_o between 50~10K Hz could be locked in our system when SNR=20 (dB). In Fig. 4.2, the number of successfully locked system decreases but it only occurs in particular range of f_o. When SNR=10 dB, there are mainly two parts of f_o being unlocked.

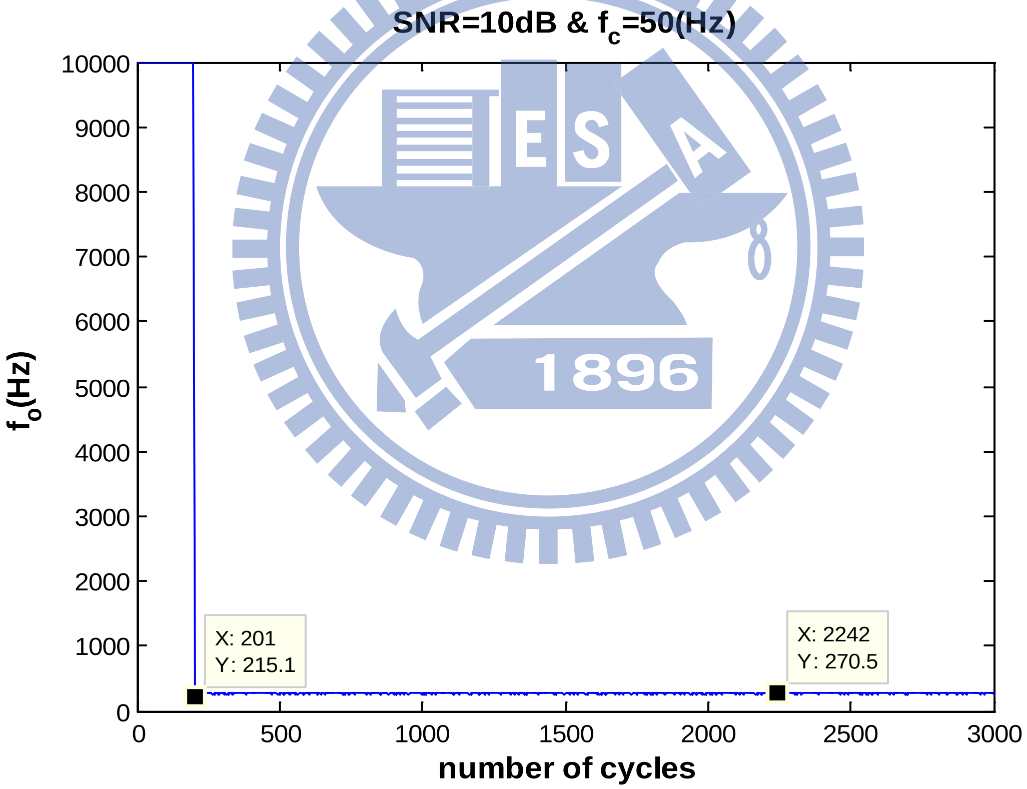

Fig. 4.6 The unlocked process of f_c = 50 Hz with SNR=10 dB.

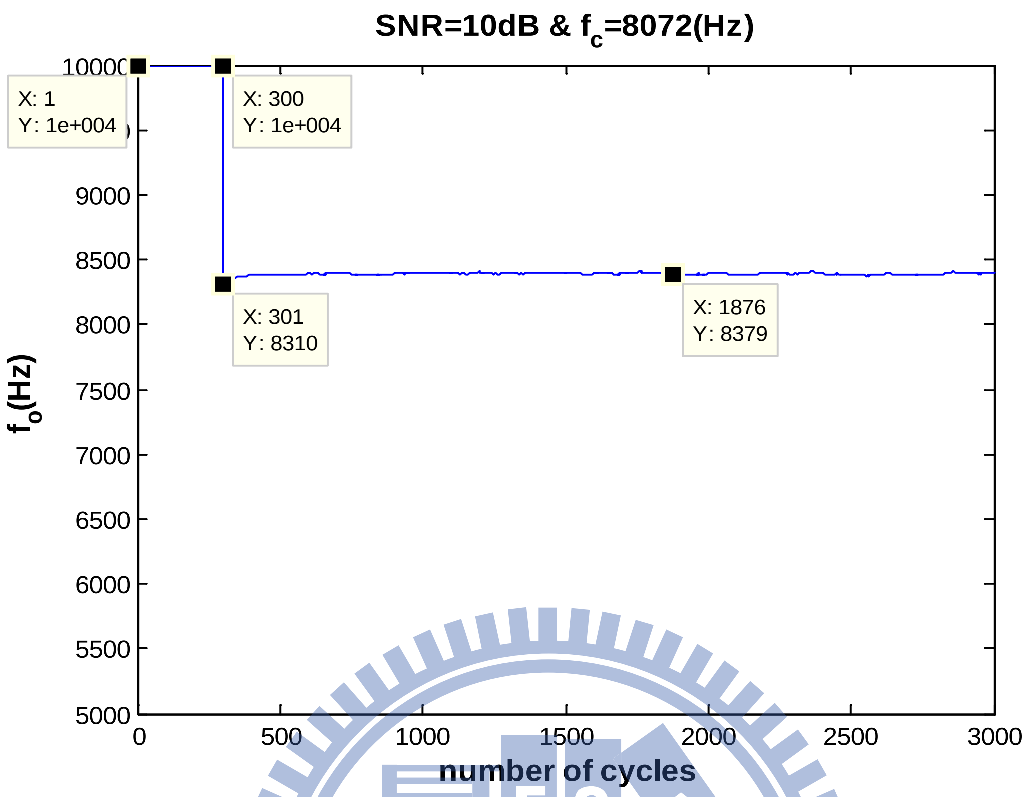

Fig. 4.7 The unlocked process of f_c = 8072 Hz with SNR=10 dB.

The first part is around the lowest frequency 50 Hz. As shown in Fig. 4.6, the system can not track and lock the input frequency 50 Hz with SNR=10dB. We observe that the phase estimation by the first 200 cycles is not sufficient, so $(f_o)_{201}$ is still away from 50 Hz. Obviously, the first step is not able to accurately estimate Δf and the system will not be locked eventually. In mathematics, we found that the first 200 cycles plays an important role in the system. As discussed in Section 3.2, the number of 200 cycles is determined by the lowest input frequency to make sure even the lowest frequency could be sampled almost one period. Nevertheless, the one period sampling provides insufficient information for phase estimation in noisy environment. This is the reason why unlock frequencies are close to 50 Hz in Fig. 4.2.

When SNR=10 dB, we found that there are a few unlocked frequencies around 8K Hz. The detail of unlocked process is shown in Fig. 4.7 for $f_c = 8072\ (Hz)$. We analyze the value of f_o and find that the accurate phase estimation of the first 200 cycles is the key factor. In this case, the reason for a wrong estimation of the first 200 cycles is concerning with the accumulated value of counter and the value of $\phi(m)-\phi(m-1)$. As discussed in part C of Section 3.3, if $\phi(m)-\phi(m-1)$ is small, the system might have a wrong estimation of $\phi(m)-\phi(m-1)$. In a noiseless environment, the value of $\phi(m)-\phi(m-1)$. However, in a noisy environment, there are many serious misjudgments. Those misjudgments make the phase estimation inaccurate. At the same time, the value of counter is not larger than 70 for the case of f_c = 8072 Hz, so the system can not enter "the higher frequency judgment" mode. Hence, the system is unlocked.

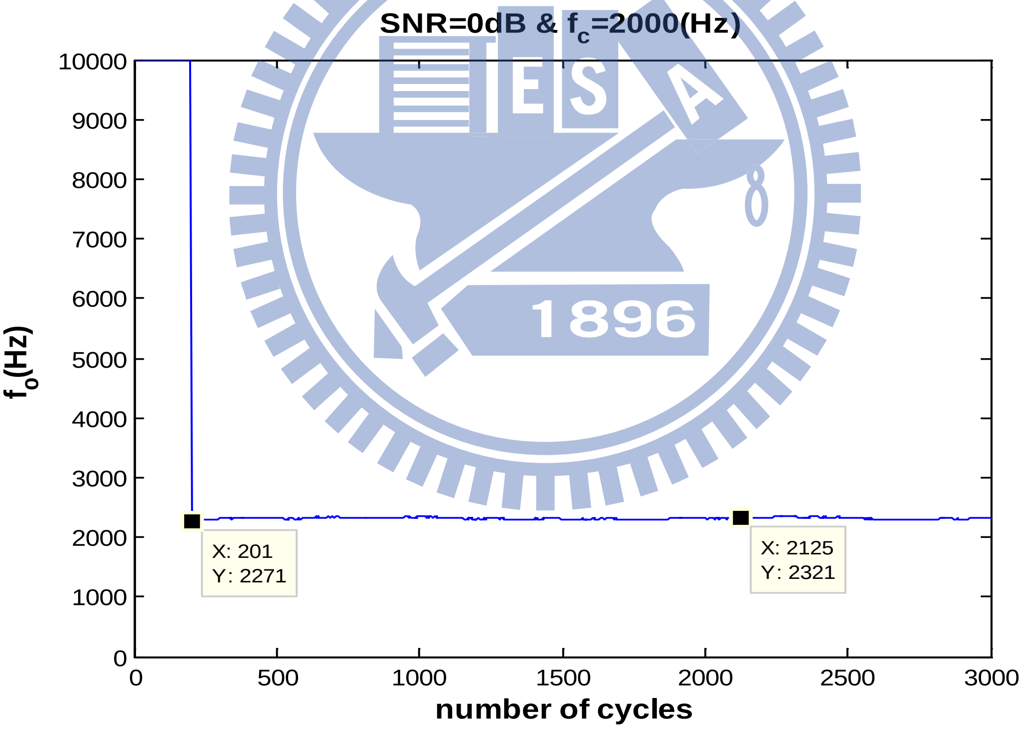

Fig. 4.8 The unlocked process of f_c = 2000 Hz with SNR=0 dB.

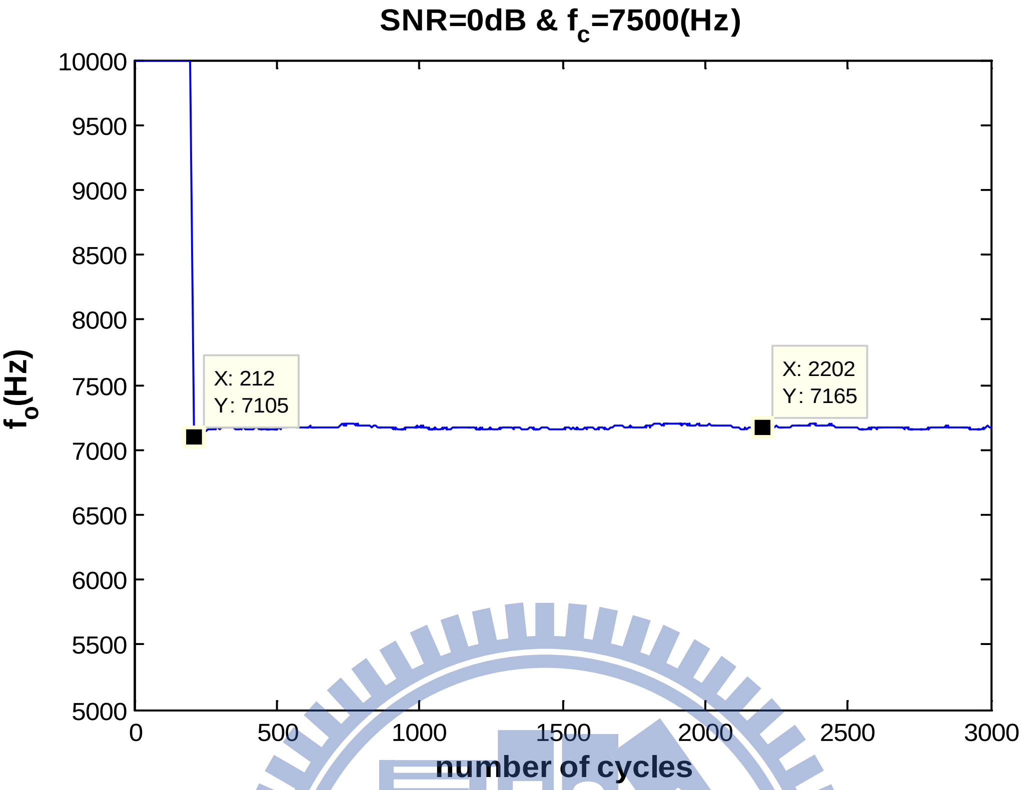

Fig. 4.9 The unlocked process of f_c = 7500 Hz with SNR=0 dB.

In the case of SNR=0 dB, we test two unlocked frequencies which is shown in Fig. 4.8 and 4.9. The main reason for Fig. 4.8 is as discussed in the case of Fig. 4.6. The whole sampling time in the first 200 cycles is insufficient. In this case, even though we have sampled about 30 periods for f_c =2000 Hz, we can not make an accurate phase estimation due to worse SNR.

For Fig. 4.9, there is the same reason as Fig. 4.6. The value of $\phi(m)-\phi(m-1)$ for f_c = 7500 Hz is too small to make an accurate estimation when SNR=0 dB. At the same time, the value of counter is not larger than 70 as well. Those reasons cause the system unlock.

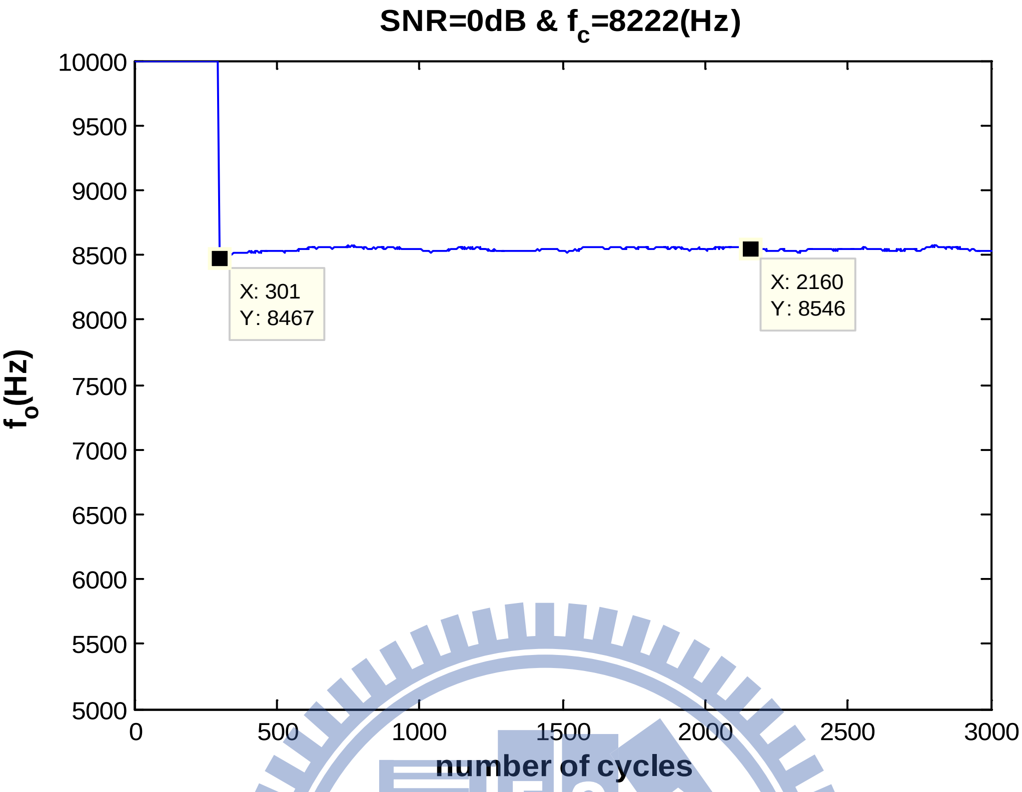

Fig. 4.10 The unlocked process of f_c = 8222 Hz with SNR=0 dB.

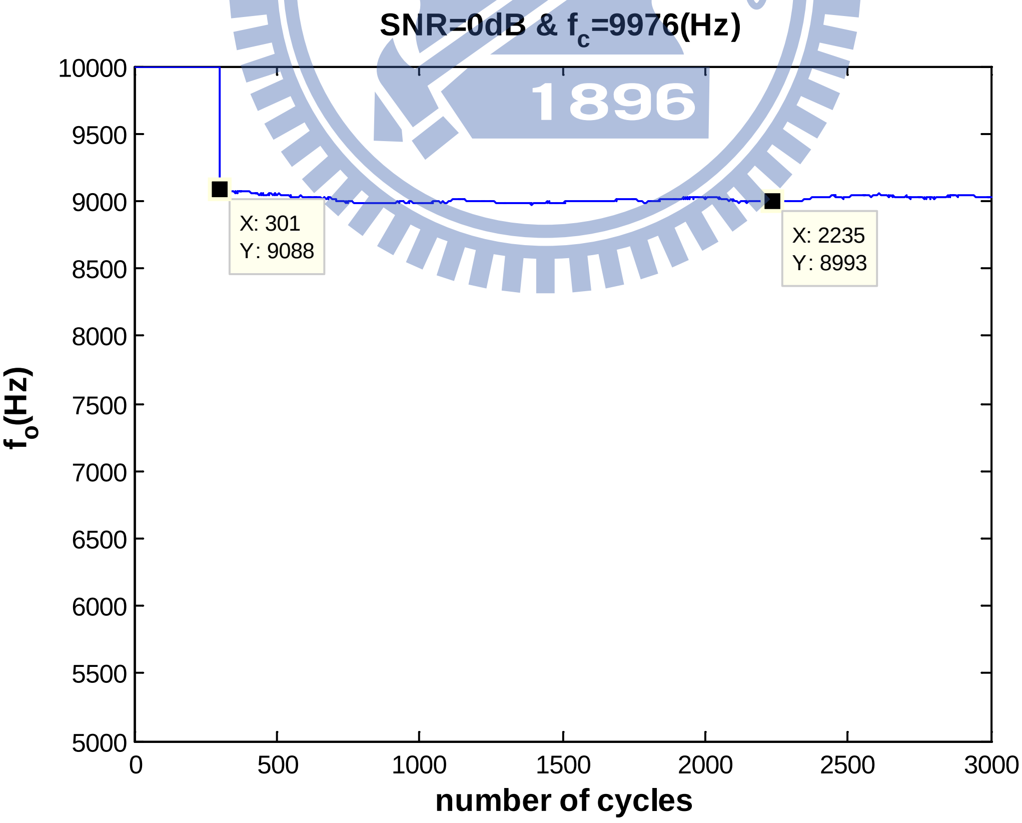

Fig. 4.11 The unlocked process of f_c = 9976 Hz with SNR=0 dB.

In Fig. 4.10, the first problem of higher input frequency appears. We discuss the reason from m=201 to m=300, that is, the step of information re-collection in part C of Section 3.3. Our purpose of information re-collection is to increase the sampling number N of every cycle in m=201~300, so as to keep the value of $\phi(m)-\phi(m-1)$ away from the boundary 0 and 2π. As depicted in Eq. 3.18, we take N as $\left\lfloor \frac{f_s}{10K-7.8K} \right\rfloor -1$. However, the step only makes sure that the value of $\phi(m)-\phi(m-1)$ is not small, but does not assure it will not be close to 2π. In other words, the value of $\phi(m)-\phi(m-1)$ might be close to 2π when input frequency is around 8K Hz.

In Fig. 4.11, another problem of higher input frequency happens while input frequency is close to 10K Hz. We also discuss the phenomenon from the step of information re-collection. As we see, we take N as $\left\lfloor \frac{f_s}{10K-7.8K} \right\rfloor -1$ in this step. In this case, the value of N is insufficient. Again, our purpose of information re-collection is to keep the value of $\phi(m)-\phi(m-1)$ away from the boundary 0 and 2π. As shown in Eq. (3.17), the value of $\phi(m)-\phi(m-1)$ is determined by $2\pi(f_o-f_c)NT_s$. If f_c is very close to f_o, N has to increase to ensure sufficiently large value of $\phi(m)-\phi(m-1)$.

However, there is a tradeoff between the case in Fig. 4.10 and 4.11. In Fig. 4.10, the problem is due to a comparatively large N. In contrast, the issue is owing to the insufficient N in Fig. 4.11. Consequently, the unlocked result is inevitable with worse SNR.

4.3. Summary

In this chapter, we analyze some unlocked examples. Apparently, the low SNR causes the worse performance in our system. In the case of SNR=10dB and SNR=0dB, we discuss the unlocked situations and find out the reasons individually. Because the system is adaptive without giving a general location of input frequency, those features result in unlocked conditions in low SNR cases.

Chapter 5.

Conclusions

In this thesis, we consider a one bit quantized digital PLL to achieve ultra-wide capture range. In Chapter 2, we introduce the operation principle of PLL. Meanwhile, we compare the traditional analog PLL and digital PLL and introduce the idea of digital PLL with is one bit quantization. The architecture of one bit quantized digital PLL is similar to traditional digital PLL. The obvious difference is the DPD in one bit quantization. We explain the phase estimation algorithm of DPD in this chapter.

In Chapter 3, we analyze details in one bit quantized digital PLL, including the choice of sampling number N, the method of phase/frequency estimation, the restriction of different input frequencies. We further establish a flowchart for distinguishing different input frequencies. Following the flowchart, we could separate input frequency into three ranges: lower, middle and higher input frequencies. Each range has its own steps to track the input frequency. As shown in simulation, we can lock input frequency whose range is from 50 to 10K Hz in noiseless environment, with an initial NCO output frequency $f_o = 10K\ Hz$.

In Chapter 4, we add AWGN to the input signal and show the tracking results for SNR ranging from 20 dB to -20 dB. The outcomes reveal the weakness of our system. In noisy environment, we find that the lower input frequency and the boundary between middle and higher input frequency have natural restrictions which may cause inaccurate phase/frequency estimation.

In summary, we have successfully implemented one bit digital quantized PLL with ultra-wide capture range. The system achieves almost 99.5% capture range of the natural frequency in noiseless environment. This is impossible for analog PLL or conventional digital PLL. However, if we want to implement the system in practice, we must take the noise affect into consideration. The noise affect is a problem worthy of further investigation in the future.

References

[1] G. C. Hsieh; J. C. Hung, "Phase-locked loop techniques. A survey," *IEEE Trans. Ind. Electron.*, Vol. 43, No. 6, Dec. 1996.

[2] R. E. Best, "Phase-locked loops: design, simulation, and applications," 5th ed., McGraw-Hill, New York, 2003.

[3] D. H. Wolaver, “Phase-Locked Loop Circuit Design,” Prentice-Hall Inc., New Jersey, 1991.

[4] T. M. Almeida; M. S. Piedade, “High performance analog and digital PLL design,” *Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on*, Vol. 4, pp. 394-397, 1999.

[5] Y. S. Choi; H. H. Choi; T. H. Kwon, “An adaptive bandwidth phase locked loop with locking status indicator,” *Science and Technology, 2005. KORUS 2005. Proceedings*, pp.826-829, 2005.

[6] J. B. Y. Tsui, “Fundamentals of Global Positioning System Receivers : A Software Approach,” 2nd ed., John Wiley & Sons Inc.,2005

[7] C. F. Chang; M. S. Kao, “High-Accuracy Carrier Phase Discriminator in One-Bit Quantized Software-Defined Receivers,” Signal Processing Letters, IEEE, Vol. 15, pp.397-400, 2008.

[8] A. V. Oppenheim, R. W. Schafer, “Digital signal processing,” 2nd ed., Prentice-Hall Inc., New Jersey, 1999.

[9] J. G. Proakis, “Digital Communications,” 3rd ed., McGraw Hill, 1995.

[10] C. F. Chang; R. M. Yang; M. S. Kao, “Implementation of an innovative phase discriminator for improved tracking performance in one-bit software GPS receiver,” ION NTM 2008, San Diego, January 2008.