**İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY**

# PID APPROACH TO PLL SYSTEMS

**M.Sc. Thesis by**

**H. Uğur UYANIK, B.Sc.**

**Department : Electronics and Communication Engineering**

**Programme : Electronics Engineering**

**JUNE 2007**

**İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY**

# PID APPROACH TO PLL SYSTEMS

**M.Sc. Thesis by**

**H. Uğur UYANIK, B.Sc.**
**(504051212)**

**Date of submission : 7 May 2007**

**Date of defence examination: 13 June 2007**

**Supervisor (Chairman): Assis. Prof. Dr. Nil TARIM**

**Members of the Examining Committee Prof.Dr. Melih PAZARCI (İTÜ)**

**Assis. Prof.Dr. Mehmet Turan SÖYLEMEZ (İTÜ)**

**JUNE 2007**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# FAZ KİLİTLEMELİ SİSTEMLERE PID YAKLAŞIMI

**YÜKSEK LİSANS TEZİ**

**Müh. H. Uğur UYANIK**
**(504051212)**

**Tezin Enstitüye Verildiği Tarih : 7 Mayıs 2007**
**Tezin Savunulduğu Tarih : 13 Haziran 2007**

**Tez Danışmanı : Yrd. Doç.Dr. Nil TARIM**
**Diğer Jüri Üyeleri Prof.Dr. Melih PAZARCI (İTÜ)**
**Yrd. Doç. Dr. Mehmet Turan SÖYLEMEZ (İTÜ)**

**HAZİRAN 2007**

**ACKNOWLEDGEMENT**

First of all, I would like to thank to my supervisor Assis. Prof. Nil TARIM for her patience, support and help during my M.Sc. thesis.

I would like to thank to M.Sc. Burçin PAK and Ph.D. Devrim Yılmaz AKSIN for their valuable advices about my thesis and to my colleagues M.Sc. Tolga KAYA and M.Sc. E. Özgür ATEŞ for their support and friendship.

I am also grateful to my family for their encouragement and support in every period of my life and to my friends for their moral support.

Finally, I would like to thank to G. Selda KURUOĞLU for her patience and calmness during my whole M.Sc. period.

**May, 2007** **H. Uğur UYANIK**

**TABLE OF CONTENTS**


**ACRONYMS** v
**FIGURE LIST** vi
**SYMBOL LIST** viii
**ÖZET** ix
**SUMMARY** x

**1. INTRODUCTION** 1

**2. BASIC PROPERTIES OF PHASE LOCKED LOOPS** 5
2.1. Block Diagram of a PLL 5
2.2. Loop Orders and Types 6
2.2.1. Order of a PLL 6
2.2.2. Type of a PLL 6
2.2.3. Steady State Errors 6
2.2.3.1. Phase Steps 7
2.2.3.2. Frequency Steps 8
2.2.3.3. Frequency Ramps 9
2.3. Transient Response of PLLs 10
2.3.1. Transient Response of a First Order PLL 11
2.3.1.1. Phase Steps 11
2.3.1.2. Frequency Steps 12
2.3.1.3. Frequency Ramps 14
2.3.2. Transient Response of a Second Order PLL 15
2.3.2.1. Phase Steps 16
2.3.2.2. Frequency Steps 18
2.3.2.3. Frequency Ramps 20
2.4. Working Ranges of PLLs 22
2.5. Noise Performance 25

**3. THE PROPOSED PHASE-LOCKED LOOP STRUCTURE** 27
3.1. Charge Pump PLL 27
3.1.1. Phase Frequency Detector 27
3.1.2. Charge Pump 29
3.1.3. Loop Filter 30

3.1.4. Noise of Loop Filter 31
3.1.5. Steady State Errors and Transient Response 32
3.2. PID Implemented PLL Structure 33
3.2.1. 1st Order Filter 35
3.2.2. PD Controller 37
3.2.3. Comparator and Finite State Machine 40
3.2.4. Enable / Disable Switch 42
3.2.5. Layout of the Overall Controller 42
3.3. The Analysis of the Proposed PLL 43

**4. SIMULATION RESULTS 50**
4.1. 2.4 GHz CMOS Frequency Synthesizer 50
4.2. A 1.5V 2.4GHz PLL for WLAN Applications 53
4.3. Overall Results 56

**5. CONCLUSION 57**

**REFERENCES 58**

**BIOGRAPHY 59**

**ACRONYMS**

| | |
|---|---|
| **DDS** | : Direct Digital Sysnthesis |
| **PLL** | : Phase-Locked Loop |
| **ROM** | : Read Only Memory |
| **VCO** | : Voltage Controlled Oscillator |
| **PID** | : Proportional-Integral-Derivative |
| **PD** | : Proportional-Derivative |
| **PFD** | : Phase Frequency Dedector |
| **PLL** | : Phase-Locked Loop |
| **ROM** | : Read Only Memory |
| **VCO** | : Voltage Controlled Oscillator |
| **PID** | : Proportional-Integral-Derivative |
| **PSD** | : Power Spectral Density |
| **AMS** | : Austria Micro Systems |
| **OTA** | : Operational Transconductance Amplifier |
| **WLAN** | : Wireless Local Area Network |

**FIGURE LIST**

**Page No:**


**Figure 1.1** : DDS Block Diagram ..... 1
**Figure 1.2** : PLL Block Diagram .....…..... 2
**Figure 2.1** : Basic Block Diagram of a PLL ..... 5
**Figure 2.2** : Phase Step Input .....……..... 7
**Figure 2.3** : Frequency Step Input.....…..... 8
**Figure 2.4** : Input Phase Function for a Frequency Step Input ..... 8
**Figure 2.5** : Input Frequency Change for a Frequency Ramp Input ..... 9
**Figure 2.6** : Phase Error of a First Order PLL for a Phase Step Input ..... 12
**Figure 2.7** : Phase Error of a First Order PLL for a Frequency Step Input .... 13
**Figure 2.8** : Frequency Error of a First Order PLL for a Frequency Step Input ..... 13
**Figure 2.9** : Phase Error of a First Order PLL for a Frequency Ramp Input.. 14
**Figure 2.10** : Frequency Error of a First Order PLL for a Frequency Ramp Input ..... 15
**Figure 2.11** : Phase Error of a Second Order PLL for a Phase Step Input ..... 17
**Figure 2.12** : Phase Error of a Second Order PLL for a Frequency Step Input 19
**Figure 2.13** : Frequency Error of a Second Order PLL for a Frequency Step Input ..... 20
**Figure 2.14** : Phase Error of a Second Order PLL for a Frequency Ramp Input ..... 21
**Figure 2.15** : Frequency Error of a Second Order PLL for a Frequency Ramp Input ..... 22
**Figure 2.16** : Working Ranges of PLLs …..... 24
**Figure 2.17** : Definition of Equivalent Rectangular Noise Bandwidth ..... 25
**Figure 3.1** : Charge Pump PLL ..... 27
**Figure 3.2** : PFD Circuit ..... 28
**Figure 3.3** : PFD Waveforms ..... 28
**Figure 3.4** : Charge Pump Circuit..... 29
**Figure 3.5** : PFD – Charge Pump Pair Waveform ..... 29
**Figure 3.6** : Charge Pump Loop Filter with Two Capacitors ..... 31
**Figure 3.7** : Noise Model for Loop Filter ..... 32
**Figure 3.8** : Controller Implemented PLL Structure ..... 34
**Figure 3.9** : Proposed Controller Structure ..... 34
**Figure 3.10** : Used 1$^{st}$ Order Filter ..... 35
**Figure 3.11** : Schematic of the OTA ..... 36
**Figure 3.12** : Schematic of the Source Follower ..... 36
**Figure 3.13** : Derivative Block ..... 37
**Figure 3.14** : Summing Amplifier ..... 37
**Figure 3.15** : Linearized OTA Input Stage ..... 38
**Figure 3.16** : Transconductance vs. Tail Current ..... 38
**Figure 3.17** : Overall PD Controller ..... 37
**Figure 3.18** : Overall Controller without Enable/Disable Circuitry..... 40

**Figure 3.19 :** Employed Comparator Topology ..... 40
**Figure 3.20 :** Final Comparator Configuration ..... 41
**Figure 3.21 :** The Finite State Machine ..... 42
**Figure 3.22 :** Enable / Disable Switch ..... 42
**Figure 3.23 :** Layout of the Overall Controller ..... 43
**Figure 3.24 :** Overall PLL curcuit ..... 43
**Figure 3.25 :** Settling of the Real Waveform with Envelopes ..... 46
**Figure 3.26 :** Normalized Overshoot vs. Damping Ratio ..... 48
**Figure 3.27 :** Effect of $K_P$ ..... 49
**Figure 3.28 :** Effect of $K_D$ ..... 49
**Figure 4.1 :** Transient Response of the First Design Example ..... 51
**Figure 4.2 :** Ideal Step Responses for the First Design Example with and without the Controller ..... 52
**Figure 4.3 :** Transient Response of the First Design Example with MOS Blocks ..... 52
**Figure 4.4 :** The Pull-In Behaviour of the First Design Example ..... 53
**Figure 4.5 :** Transient Response of the Second Design Example ..... 54
**Figure 4.6 :** Ideal Step Responses for the Second Design Example with and without the Controller ..... 54
**Figure 4.7 :** The Transient Response of the Second Design Example with MOS Blocks ..... 55
**Figure 4.8 :** The Pull-In Behaviour of the Second Design Example ..... 55

**SYMBOL LIST**

| | |
|---|---|
| $K_{PD}$ | : Phase detector gain |
| $F_L(s)$ | : Loop filter transfer function |
| $K_{VCO}$ | : VCO gain |
| $N$ | : Frequency divide ratio |
| $H(s)$ | : Closed loop transfer function of a PLL |
| $\omega_o$ | : Output frequency |
| $\omega_i$ | : Input frequency |
| $\Phi_e$ | : Phase error of a PLL |
| $\Phi_i$ | : Input phase |
| $\Phi_o$ | : Output phase |
| $G(s)$ | : Open loop transfer function of a PLL |
| $U(t)$ | : Unit step function |
| $K_v$ | : Velocity error constant |
| $K_a$ | : Acceleration error constant |
| $K$ | : Open loop DC gain |
| $m$ | : Slope of the frequency ramp input |
| $\omega_p$ | : Pole frequency |
| $\omega_z$ | : Zero frequency |
| $\omega_n$ | : The natural frequency |
| $\zeta$ | : Damping ratio |
| $\omega_d$ | : The oscillation frequency |
| $\Phi_{PDMAX}$ | : The maximum input phase for which the PD behaves linear |
| $B_L$ | : The noise bandwidth |
| $T_p$ | : The pull-in time for a PLL |
| $I_{CP}$ | : Current of the charge pump |
| $F(s)$ | : Filter function |
| $Z(s)$ | : Fitler transimpedance function |
| $e_{n,2}$ | : Noise voltage generated by resistor |
| $e_{n,leak}$ | : Noise voltage generated by leakage resistor |
| $I_{TAIL}$ | : Tail current |

## FAZ KİLİTLEMELİ SİSTEMLERE PID YAKLAŞIMI

## ÖZET

Bu çalışmada PID kontrolörü faz kilitlemeli çevrim (PLL) sistemlerine entegre etme yolu araştırılmıştır. Temel PLL yapıları incelenmiş ve s domeni transfer fonksiyonları kullanılarak analiz edilmiştir. Sürekli hal hataları ve basamak cevapları ters Laplace dönüşümü kullanılarak formüle edilmiştir. PLL'lerin çalışma aralıkları ve gürültü performansları verilmiştir. Yük pompası kullanan PLL yapıları detaylı olarak incelenmiştir. Bu PLL'in alt blokları açıklanmış, basamak cevabı tasarım parametrelerinin bir fonksiyonu olarak verilmiştir. Yük pompası içeren PLL'lerin hali hazırda integratör içerdiği göz önünde bulundurularak PID kontrolörü PD kontrolörü olarak değiştirilmiştir. Bu amaca uygun bir PD kontrolörü önerilmiştir. Önerilen PLL yapısının s domeni transfer fonksiyonu çıkartılmış ve bu transfer fonksiyonundan yararlanılarak basamak cevabına ait formüller hesaplanmıştır. Kontrolör parametrelerinin basamak cevabına etkisi incelenmiştir. Önerilen kontrolör yapısı 0.35μm AMS CMOS prosesi kullanılarak gerçeklenmiştir. Daha önceden önerilen iki PLL yapısının parametreleri önerilen kontrolör kullanılarak iyileştirilmiştir. Hesaplanan sonuçlar simülasyonlarla uyumluluk göstermiştir.

**PID APPROACH TO PLL SYSTEMS**

## SUMMARY

In this work, a proper way to implement PID controllers into PLL systems is examined. Basic PLL structures are explained and analyzed using their s domain transfer functions. Steady state errors and transient responses are formulated in detail with the help of inverse Laplace transforms of s domain functions. Working ranges and noise performances of PLLs are given. Charge pump PLL structure is examined in detail. The building blocks of charge pump PLL such as the phase frequency detector and the loop filter are explained. Transient response of a charge pump PLL is given as a function of the design parameters. Considering that the charge pump PLL system already has an integrator, PID controller is configured as a PD controller. A PD controller suitable for integrating into charge pump PLLs is proposed. The s domain transfer function of the proposed PLL structure is formulated and using the transfer function, transient response of the proposed PLL is extracted. Effects of controller parameters on the transient response are investigated. Proposed controller is realized using 0.35μm CMOS process of AMS. The parameters of the two previously proposed PLLs are tuned with the proposed controller and the calculations shows consistent results with the simulations.

## 1. INTRODUCTION

Synthesizing a frequency from a given periodic signal has always been a typical problem of engineering. Its roots go back to the industrial revolution when the engineers faced with a simple equation that they had to implement in their mechanical systems [1]:

$$\omega_{out} = \xi_x \, \omega_{in} \tag{1.1}$$

where $\xi_x$ is the frequency ratio. Although they had the opportunity to efficiently solve this problem with simple gearboxes, frequency synthesis in electronics is somewhat difficult, especially when frequency ratio is larger than 1. Thus, many circuits were proposed for multiplying a frequency of a given input signal, which is mostly a reference signal with high spectral purity. In today's technology, two major methods are utilized for frequency synthesis: Direct digital synthesis (DDS) and phase locked loops (PLL).

The main idea behind the DDS principle is to synthesize a sine wave using sampled sinusoidal data. An accumulator counts up using the frequency word, then the output is changed to sampled sine wave using the data stored in ROM. The resulting wave is then converted to analog, and filtered to reduce frequency spurs.

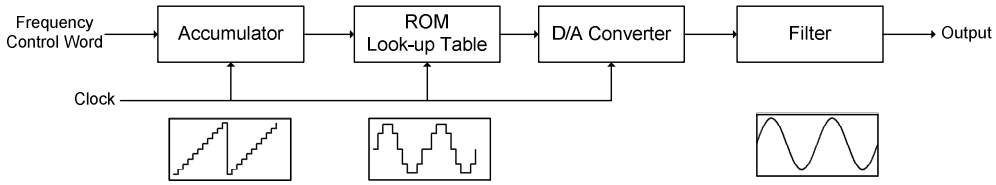


**Figure 1.1**: DDS Block Diagram

PLL, on the other hand, is a widely used circuit technique for over half a century. Its main ability is to generate a waveform which is in phase with its input. Because of this ability, PLL is a solution to many problems in synchronization, clock recovery and frequency synthesis [2-4].

In the earliest decades of the past century, superheterodyne receivers were being used. However, there were many tuned stages in superheterodyne receivers, so there was a search for a simpler method. The search resulted in 1932, when the concept of homodyne receivers was proposed. The idea was simpler than superheterodyne; when the input signal and a local oscillator signal with the same phase and frequency were multiplied, the result was the exact replica of the modulated waveform. The idea seemed to have no problem, until the experiments showed that the frequency of the local oscillator was drifting after some time. In order to solve this problem, a method used in servo motors was implemented; the phase difference of two signals was measured and used as a correction voltage for the local oscillator. This idea is still the main essence of phase locked loops [5].

Mainly, the PLL is a loop employing two main blocks. The first block has the purpose of measuring the phase difference between the generated waveform and the input wave, and using the measured phase difference, the second block generates the waveform fed back into the first block. Using this feedback method, the PLL eventually removes the phase difference between the input and output signals [5].

In most cases, a loop filter is inserted into the forward gain path in order to achieve the desired transfer function. Also in frequency generator applications, a frequency divider is inserted into the feedback path so that the output frequency is a product of the input frequency and the divide ratio. So the final basic blocks of a PLL can be summarized as a phase-error detector, a loop filter, a voltage controlled oscillator (VCO) and a divider.

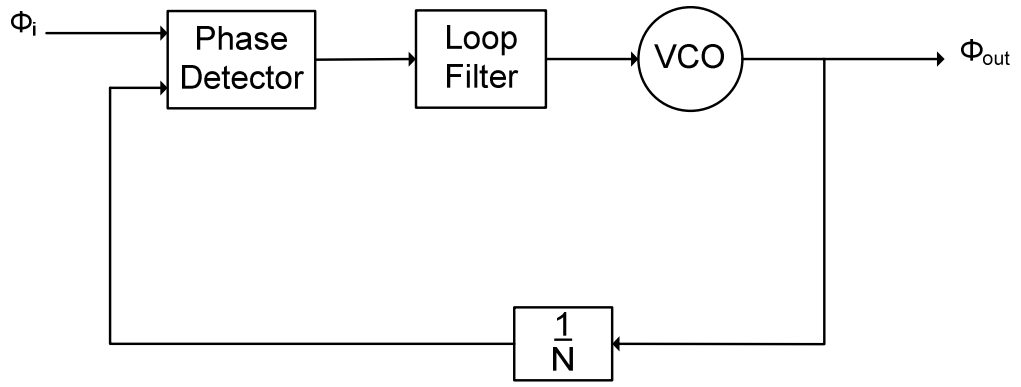


**Figure 1.2**: PLL Block Diagram

PLL design is a trade-off cycle between parameters such as bandwidth, settling time and noise. Since there are many different areas where the PLL is used, the importance of these parameters varies within the context of certain limits. For example, settling time is not the primary concern when the PLL is employed in a system which demands constant frequency. However it becomes important in areas such as wireless communications, where frequency hopping demands quick changes in the frequency of the generated signal.

Taking settling time as a figure of merit, there are numerous things that a PLL designer must deal with. Increasing the loop bandwidth is the first and easiest thing that comes into mind. However, increasing loop bandwidth also increases noise bandwidth, which means that less noise is filtered in the loop. The same trade-off is also valid for damping ratio, which plays a critical role in the transient response [1].

In order to overcome the trade-offs, there are many proposed circuits which change loop dynamics during the transition time and restore the originals afterwards. Just like a gearbox, most of these circuits are based on the idea of accelerating and decelerating the loop. This is primarily done by discretely switching a loop parameter with the help of the data coming from a lock detector [1]. These circuits are sometimes hard to design, mostly not configurable, and due to their nonperiodical discrete behaviors, nearly impossible to be modeled. Designing an additional block to decrease the settling time, employing basic, configurable circuits with precise models is the main idea of control theory.

Considering that the PLL is a feedback control loop, it can be analyzed in terms of all control theory. However traditional PLLs lack a decent controller used in all modern control systems. The absence of a controller makes PLL designers deal with trade-offs when tuning PLLs' transient performances, namely settling time, overshoot and ripple. In order to overcome this issue, a way to implement a controller into the PLL without disturbing its initial properties, especially noise, should be investigated.

The basic controller in control theory is the proportional, integral and derivative controller, called the PID controller. Implementing a PID controller into the PLL would loosen the trade-off margins and produce better results in terms of transient response when compared with the conventional PLL design.

Due to their phase lock ability, PLLs are widely used for PID systems, however PID implemented PLL structure has not been found in the literature.

The goal of this thesis is to implement a PID controller into PLL in order to tune the transient response without disturbing its noise properties.

Chapter 2 presents the basic properties, s domain transfer functions, steady state errors, transient responses and the noise properties of PLLs.

Chapter 3 presents charge pump PLLs and the implementation of the PID concept into charge pump PLLs.

Chapter 4 presents simulation results after implementing the controller into two previously proposed PLLs.

Chapter 5 is the review of the thesis and the conclusion.

## 2. BASIC PROPERTIES OF PHASE-LOCKED LOOPS

In this chapter, the basic properties of phase-locked loops are investigated. Steady state errors and time domain analyses are formulated step by step using proper s domain transfer functions, including the effect of divider ratio.

### 2.1 Block Diagram of a PLL

As mentioned in the first chapter, the PLL consists of four main blocks, namely phase detector, loop filter, VCO and divider. The signal flow graph of such PLL is shown in Figure 2.1.

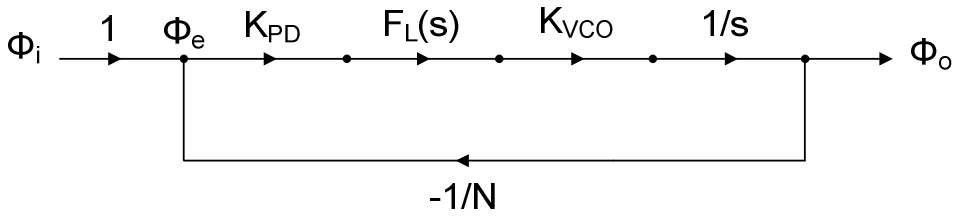


**Figure 2.1**: Basic Block Diagram of a PLL

In the linear model of a PLL, $K_{PD}$, $F_L$, $K_{VCO}$ and N are the phase detector gain, loop filter transfer function, VCO gain and frequency divide ratio, respectively. Defining G(s) as forward gain results with:

$$G(s) = \frac{K}{s} F_L(s) \tag{2.1}$$

where $K = K_{PD} K_{VCO}$ . Using (2.1), the closed loop gain becomes:

$$H(s) = \frac{G(s)}{1 + \frac{1}{N} G(s)} = \frac{\frac{K F_L}{s}}{1 + \frac{K F_L}{sN}} = N \frac{\frac{K F_L}{N}}{s + \frac{K F_L}{N}} \tag{2.2}$$

## 2.2 Loop Orders and Types

PLLs are classified by two parameters; their orders and their types. This classification gives the designer a starting point in PLL design, since PLLs with same orders and same types have similar transfer functions, thus they behave similarly, regardless of the blocks they are built with.

### 2.2.1 Order of a PLL

Order of a PLL is defined as the order of the denominator of the closed loop transfer function [1]. In the simplest case, where $F_L = 1$, the PLL becomes first order loop and using (2.2), the closed loop transfer function becomes:

$$H(s) = N\frac{\omega_0}{s+\omega_0} \tag{2.3}$$

where $\omega_0 = \frac{K}{N}$. Similarly, higher order loop filters increases the order of a PLL.

### 2.2.2 Type of a PLL

The type of a PLL is simply the number of integrators in the loop, because in systems like the PLL, where steady state errors are the main concern, the number of integrators in the loop has a special importance. Using this definition, the type of a PLL can easily be determined by the number of poles in the open loop transfer function [1]. Since every PLL has a VCO, which acts like an integrator, orders of all PLLs are larger than zero.

### 2.2.3 Steady State Errors

Including N into the equations derived in [1,6], phase error of a PLL is defined as:

$$\Phi_e(s) = \frac{\Phi_o(s)}{G(s)} = \Phi_i(s)\frac{1}{1+\frac{G(s)}{N}} \tag{2.4}$$

where $\Phi_i$, $\Phi_o$ and $\Phi_e$ are the input phase, the output phase and the phase error respectively.

G(s) can be defined as the ratio of two polynomials, such as:

$$G(s) = \frac{A(s)}{s^n B(s)} \tag{2.5}$$

where n is the type of PLL.

Substituting (2.5) into (2.4) gives:

$$\Phi_e(s) = \Phi_i(s) \frac{Ns^n B(s)}{A(s) + Ns^n B(s)} \tag{2.6}$$

Applying Laplace limit theorem to (2.6) results with:

$$\lim_{t\to\infty}\left[\phi_e(t)\right] = \lim_{s\to 0}\left[\Phi_i(s)\frac{Ns^{n+1}B(s)}{A(s)+Ns^n B(s)}\right] = \lim_{s\to 0}\left[\Phi_i(s)\frac{Ns^{n+1}B(s)}{A(s)}\right] \tag{2.7}$$

Using (2.7), response of a PLL to different inputs can be formulated.

#### 2.2.3.1 Phase Steps

As shown in Figure 2.2, for the case where a phase step is applied, the input function can be defined as:

$$\phi_i(t) = \Delta\phi_i U(t) \tag{2.8}$$

where U(t) is the unit step function.

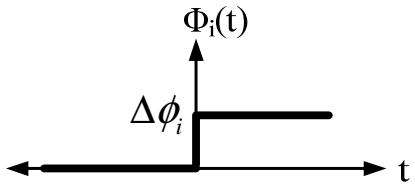


**Figure 2.2:** Phase Step Input

Using (2.8), Laplace transform of the input becomes:

$$\Phi_i(s) = \frac{\Delta\phi_i}{s} \tag{2.9}$$

Substituting (2.9) into (2.7) gives:

$$\lim_{t\to\infty}\left[\phi_e(t)\right]=\lim_{s\to 0}\left[\Delta\phi_i\frac{Ns^nB(s)}{A(s)}\right]=0 \tag{2.10}$$

Equation (2.10) is a simple result but it gives fundamental information for PLLs; the steady state phase error of a phase step input is zero.

#### 2.2.3.2 Frequency Steps

For a frequency step, the frequency of the input signal is:

$$\omega_i(t)=\Delta\omega_i U(t) \tag{2.11}$$

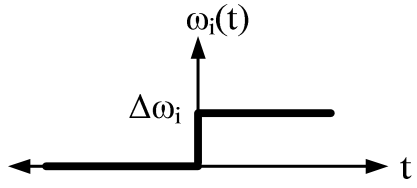


**Figure 2.3:** Frequency Step Input

Using (2.11), the input phase becomes:

$$\phi_i(t)=\int_{-\infty}^{t}\omega_i(\tau)\,d\tau=\Delta\omega_i\,tU(t) \tag{2.12}$$

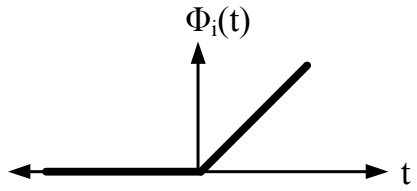


**Figure 2.4:** Input Phase Function for a Frequency Step Input

Taking the Laplace transform of the input shown in (2.12) results with:

$$\Phi_i(s)=\frac{\Delta\omega_i}{s^2} \tag{2.13}$$

Substituting (2.13) into (2.7) gives:

$$\lim_{t\to\infty}\left[\phi_e(t)\right]=\lim_{s\to 0}\left[\Delta\omega_i\frac{Ns^{n-1}B(s)}{A(s)}\right] \tag{2.14}$$

Equation (2.14) shows that the steady state error for a frequency step depends on the type of the PLL. For type I PLLs (2.14) becomes:

$$\lim_{t\to\infty}\left[\phi_e(t)\right]=N\frac{\Delta\omega_i}{KF_L(0)}=\frac{\Delta\omega_i}{K_v} \tag{2.15}$$

where $K_v$ is the parameter called the velocity error constant in control systems terminology. For type II PLLs, the steady state error for a frequency step input is zero. These results show that after a frequency step at the input, a type I PLL tracks the input phase with a constant error and a type II PLL with zero error, but the output frequency value has no errors in both cases, since constant phase error means zero frequency error.

#### 2.2.3.3 Frequency Ramps

As shown in Figure 2.5, for the case where the rate of change of the frequency is constant, the input frequency is:

$$\omega_i(t)=mtU(t) \tag{2.16}$$

where m is the slope of the frequency ramp.

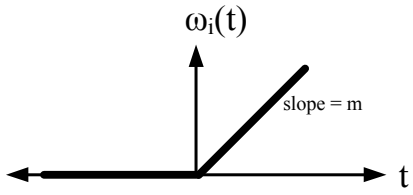


**Figure 2.5:** Input Frequency Change for a Frequency Ramp Input

Using (2.16), the input phase becomes:

$$\phi_i(t)=\int_{-\infty}^{t}\omega_i(\tau)\,d\tau=m\frac{t^2}{2}U(t) \tag{2.17}$$

Laplace transform of (2.17) is:

$$\Phi_i(s) = \frac{m}{s^3} \tag{2.18}$$

Substituting (2.18) into (2.7) gives:

$$\lim_{t\to\infty}\left[\phi_e(t)\right] = \lim_{s\to 0}\left[m\frac{Ns^{n-2}B(s)}{A(s)}\right] \tag{2.19}$$

Similar to the frequency step input, the steady state error of frequency ramp input depends on the type of the PLL. As shown in (2.20), for type I PLLs, although the phase difference is infinite for a frequency ramp input, the frequency difference is constant.

$$\lim_{t\to\infty}\left[\omega_e(t)\right] = \lim_{t\to\infty}\left[\frac{d}{dt}\phi_e(t)\right] = \lim_{s\to 0}\left[m\frac{Ns^{n-1}B(s)}{A(s)}\right] = N\frac{m}{KF_L(0)} \tag{2.20}$$

For type II PLLs, the steady state phase difference is constant for a frequency ramp, such as:

$$\lim_{t\to\infty}\left[\phi_e(t)\right] = mN\left[\frac{B(0)}{A(0)}\right] = \frac{m}{K_a} \tag{2.21}$$

where $K_a$ is the acceleration error constant. Equations (2.20) and (2.21) show that after a frequency ramp is applied, type I PLL tracks the input frequency with a constant frequency error whereas type II PLL tracks the input frequency with zero error but finite phase error. Type III PLLs can track the frequency ramp with zero frequency and phase error, but they are used exceptionally.

### 2.3 Transient Response of PLLs

Steady state errors mentioned so far deal with the situations occurring after infinite time passes. However, temporary behaviors of PLLs under sudden changes of input signals have special importance in areas where settling time is critical. Although s domain models are not accurate enough for large values of changes in the input phase, they provide the starting point for tuning a PLL's transient response. Besides,

their accuracy increases as the change in the input gets smaller, thus they are sufficiently suitable for small changes in input phase or input frequency.

### 2.3.1 Transient Response of a First Order PLL

The open loop transfer function of a first order PLL is:

$$G(s) = \frac{K}{s} \tag{2.22}$$

where $K = K_{PD}K_{VCO}$ . Using (2.6), the phase error of the system is defined as:

$$\Phi_e(s) = \Phi_i(s)\frac{sN}{sN + K} \tag{2.23}$$

Changing the input signal in (2.23), behavior of any first order PLL can be investigated.

#### 2.3.1.1 Phase Steps

In the case where a phase step is applied to the input, the Laplace transform of the input is shown in (2.9). Substituting (2.9) into (2.23) gives the following transfer function:

$$\Phi_e(s) = \Delta\phi_i \frac{N}{sN + K} \tag{2.24}$$

Taking the inverse Laplace transform of (2.24) gives:

$$\phi_e(t) = \Delta\phi_i e^{-\frac{Kt}{N}} \tag{2.25}$$

Equation (2.25) shows that any first order PLL gives an exponential phase error response to a phase step input, which approaches to zero as time goes to infinity. This statement concludes to the same result with (2.10).

Equation (2.25) also shows that the input phase error settles faster as K increases or N decreases. Graphical representation of rate of change in the phase error is shown in Figure 2.6.

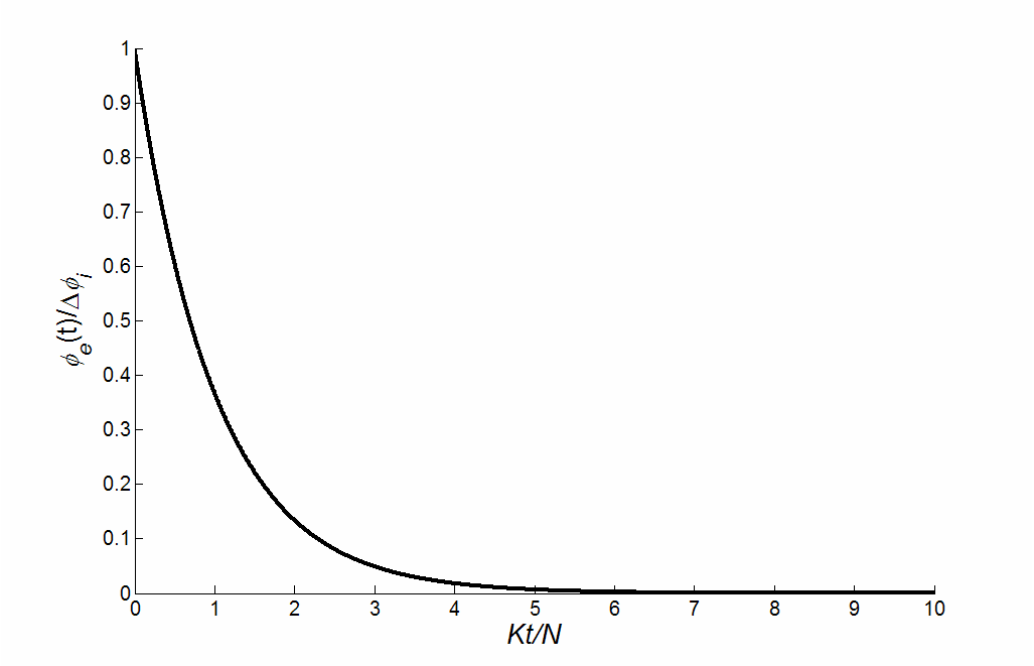


**Figure 2.6:** Phase Error of a First Order PLL for a Phase Step Input

#### 2.3.1.2 Frequency Steps

The Laplace transform of the input phase when a frequency step is applied is shown in (2.13). Substituting (2.13) into (2.23) gives the following transfer function:

$$\Phi_e(s) = \frac{\Delta\omega_i}{s}\frac{N}{sN+K} \tag{2.26}$$

Taking the inverse Laplace transform of (2.26) gives:

$$\phi_e(t) = \frac{\Delta\omega_i}{K}N\left(1-e^{-\frac{Kt}{N}}\right) \tag{2.27}$$

Equation (2.27) shows that in the case of frequency step input, the phase error reduces exponentially, but approaches to a finite value as time goes to infinity, rather than zero. It also shows that the steady state error decreases as K/N increases. This is the same conclusion with (2.15). Graphical representations of the change in the phase error for different K/N values are shown in Figure 2.7.

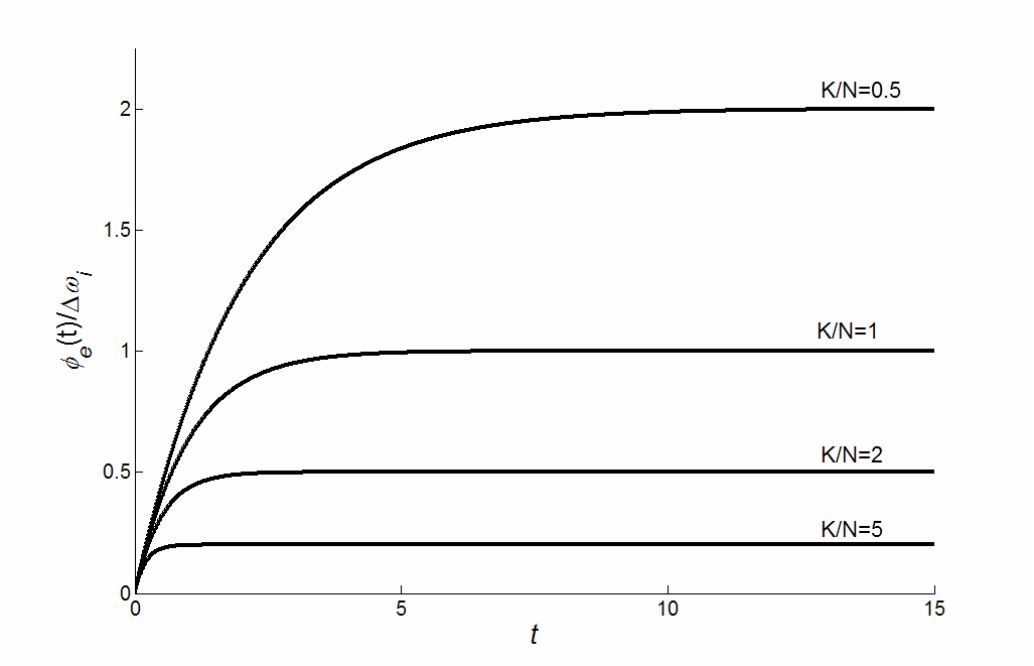


**Figure 2.7:** Phase Error of a First Order PLL for a Frequency Step Input

The derivatives of the curves in Figure 2.7, which correspond to the frequency error for a frequency step input, are shown in Figure 2.8.

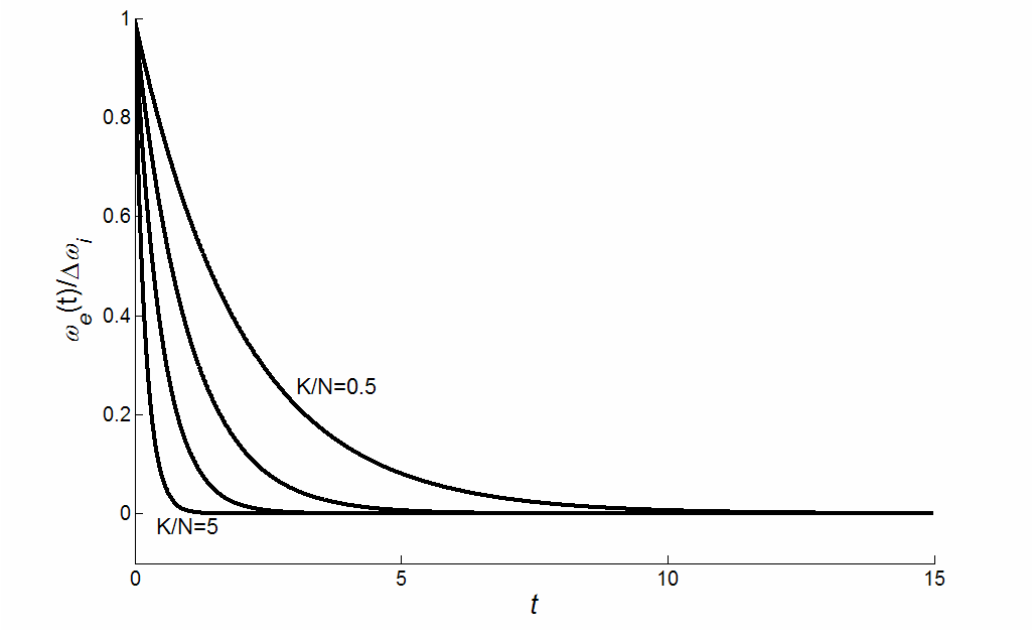


**Figure 2.8:** Frequency Error of a First Order PLL for a Frequency Step Input

As seen in Figure 2.8, frequency error for a frequency step input starts from 1 and goes to 0 as expected. It can also be seen that increasing K/N values decreases the settling time.

### 2.3.1.3 Frequency Ramps

Substituting (2.18) into (2.23) gives the following transfer function:

$$\Phi_e(s) = \frac{m}{s^2}\frac{N}{sN+K} \tag{2.28}$$

Taking the inverse Laplace transform of (2.28) gives:

$$\phi_e(t) = \frac{m}{K^2}N\left[N\left(e^{-\frac{Kt}{N}}-1\right)+Kt\right] \tag{2.29}$$

Equation (2.29) shows that in the case of frequency ramp input, the phase error becomes linear as time goes to infinity; hence, even though the phase error is increasing as the time increases, the frequency error at the output remains constant. This result is confirmed by (2.20).

Phase errors and frequency errors for different K/N values are shown in Figure 2.9 and Figure 2.10, respectively. Similar to phase and frequency step inputs, increasing K/N value decreases the steady state errors and decreases the system's response time.

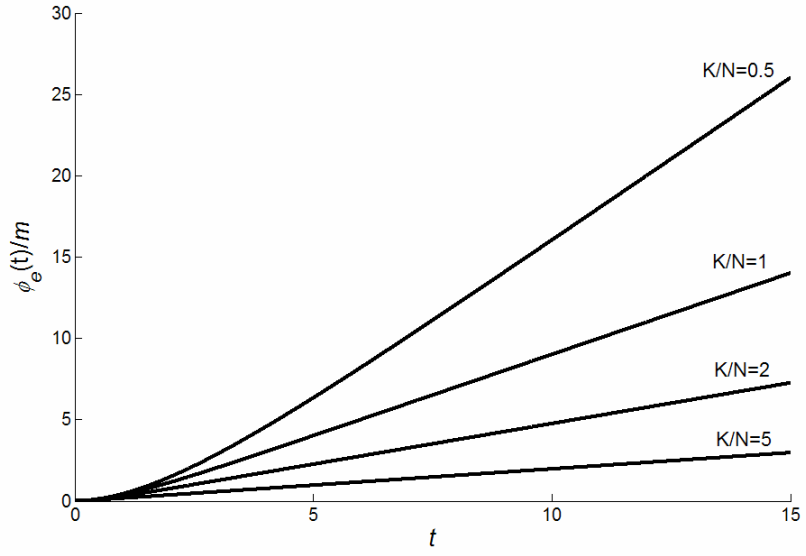


**Figure 2.9:** Phase Error of a First Order PLL for a Frequency Ramp Input

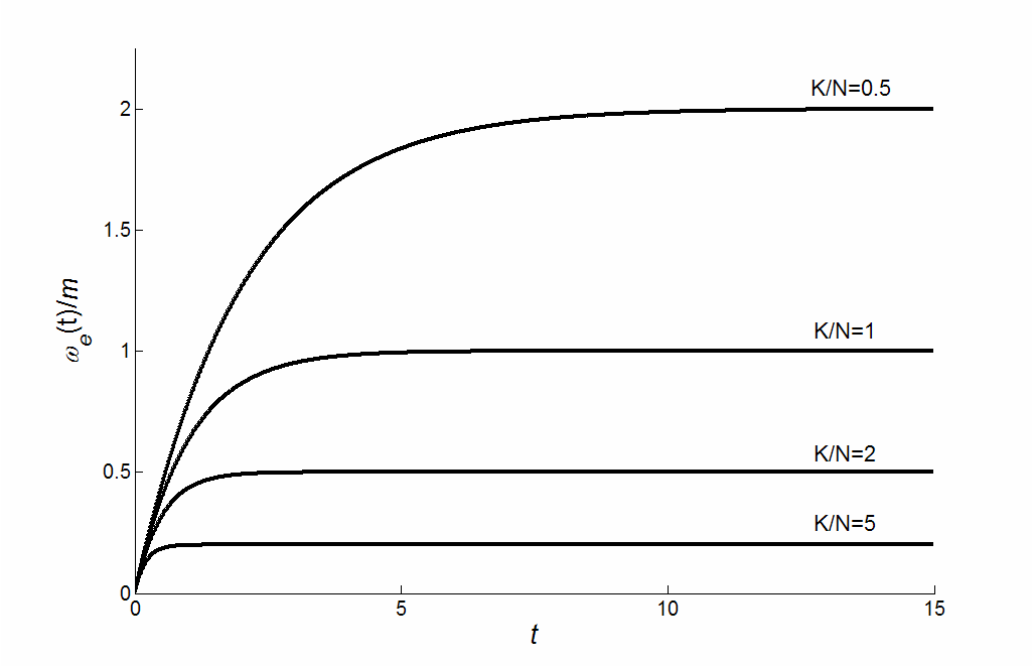


**Figure 2.10:** Frequency Error of a First Order PLL for a Frequency Ramp Input

### 2.3.2 Transient Response of a Second Order PLL

The open loop transfer function of a general second order PLL is as follows:

$$G(s) = \frac{K}{s}\frac{s+\omega_z}{s+\omega_p} \tag{2.30}$$

where $K = K_{PD}K_{VCO}$ . Using (2.6) the phase error of the system is defined as:

$$\Phi_e(s) = \Phi_i(s)\frac{Ns(s+\omega_p)}{s^2N + s(N\omega_P + K) + K\omega_z} \tag{2.31}$$

Defining ζ and $\omega_n$ as new variables and doing some math converts (2.31) into a more familiar form:

$$\Phi_e(s) = \Phi_i(s)\frac{s(s+\omega_p)}{s^2 + 2\zeta\omega_n s + \omega_n{}^2} \tag{2.32}$$

where

$$\omega_n = \sqrt{\frac{K\omega_z}{N}} \tag{2.33}$$

and

$$\zeta = \frac{\left(\omega_P + \frac{K}{N}\right)}{2\sqrt{\frac{K\omega_z}{N}}} \tag{2.34}$$

Using (2.32), transient response of any second order PLL can be calculated. It can easily be noted that when $\omega_p = 0$, the system becomes a typical second order type II PLL.

Equation (2.32) shows that unlike first order PLLs, where forward gain is the only degree of freedom, second order PLLs have two main transfer function parameters, namely $\omega_n$, which is the natural frequency, and $\zeta$, which is the damping ratio.

#### 2.3.2.1 Phase Steps

Substituting (2.9) into (2.32) gives the following transfer function:

$$\Phi_e(s) = \Delta\phi_i \frac{\left(s + \omega_p\right)}{s^2 + 2\zeta\omega_n s + \omega_n{}^2} \tag{2.35}$$

Taking the inverse Laplace transform of (2.35) gives:

$$\phi_e(t) = \Delta\phi_i\, e^{-\zeta\omega_n t}\left[\cosh\left(\omega_n t\sqrt{\zeta^2-1}\right) + \frac{\frac{\omega_p}{\omega_n} - \zeta}{\sqrt{\zeta^2-1}}\sinh\left(\omega_n t\sqrt{\zeta^2-1}\right)\right] \tag{2.36}$$

which can be simplified as follows when $\zeta = 1$, which makes the system critically damped, which is also considered as the optimum value for transient response [7]:

$$\phi_e(t) = \Delta\phi_i\, e^{-\zeta\omega_n t}\left[1 + t\left(\omega_p - \omega_n\right)\right] \tag{2.37}$$

However, when $\zeta < 1$, the system is underdamped and the equation (2.36) becomes:

$$\phi_e(t) = \Delta\phi_i\, e^{-\zeta\omega_n t}\left[\cos\left(\omega_n t\sqrt{1-\zeta^2}\right) + \frac{\frac{\omega_p}{\omega_n} - \zeta}{\sqrt{1-\zeta^2}}\sin\left(\omega_n t\sqrt{1-\zeta^2}\right)\right] \tag{2.38}$$

which further simplifies to:

$$\phi_e(t) = \Delta\phi_i\, e^{-\zeta\omega_n t}\sqrt{1 + \frac{\left(\frac{\omega_p}{\omega_n} - \zeta\right)^2}{1-\zeta^2}}\sin\left[\omega_n t\sqrt{1-\zeta^2} + \arctan\left(\frac{\sqrt{1-\zeta^2}}{\frac{\omega_p}{\omega_n} - \zeta}\right)\right] \tag{2.39}$$

A general form of a damping oscillation is:

$$f(t) = Ae^{-\zeta\omega_n t}\sin\left(\omega_d t + \phi_i\right) \tag{2.40}$$

where $Ae^{-\zeta\omega_n t}$ is the envelope of the oscillation, $\omega_d$ is the oscillation frequency and $\phi_i$ is the initial phase.

Equation (2.39) shows that when $\zeta < 1$, the output is simply a damped oscillation, which is an expected result. As shown in Figure 2.11, for every positive damping ratio, the output phase error approaches to zero as time goes to infinity.

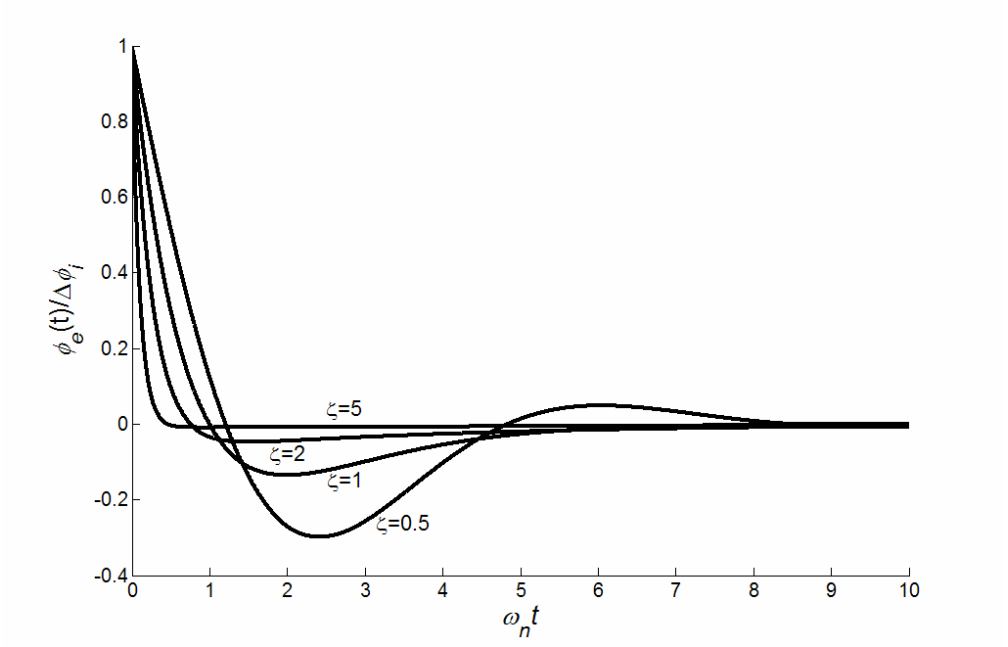


**Figure 2.11:** Phase Error of a Second Order PLL for a Phase Step Input

For type II PLLs, equations (2.36), (2.37) and (2.39) are easily calculated by taking $\omega_p$ zero. The change of phase errors for a phase step input of a second order type II system with respect to normalized time is shown in Figure 2.11. As seen in Figure 2.11, increasing the damping ratio decreases the settling time. Considering that the horizontal axis is the normalized time, it can be seen that increasing the natural frequency decreases the settling time. However, in PLL systems, increasing the natural frequency also increases the noise bandwidth, which increases the amount of noise passing to the output.

#### 2.3.2.2 Frequency Steps

Substituting (2.13) into (2.32) gives the following transfer function:

$$\Phi_e(s) = \frac{\Delta\omega_i}{s}\frac{(s+\omega_p)}{s^2+2\zeta\omega_n s+\omega_n{}^2} \tag{2.41}$$

Taking the inverse Laplace transform of (2.41) gives:

$$\phi_e(t) = \Delta\omega_i \frac{\omega_p}{\omega_n{}^2}\left\{1-e^{-\zeta\omega_n t}\left[\cosh\left(\omega_n t\sqrt{\zeta^2-1}\right)+\frac{\zeta-\dfrac{\omega_n}{\omega_p}}{\sqrt{\zeta^2-1}}\sinh\left(\omega_n t\sqrt{\zeta^2-1}\right)\right]\right\} \tag{2.42}$$

Taking $\omega_p = 0$ gives:

$$\phi_e(t) = \frac{\Delta\omega_i}{\omega_n}\frac{e^{-\zeta\omega_n t}}{\sqrt{\zeta^2-1}}\sinh\left(\omega_n t\sqrt{\zeta^2-1}\right) \tag{2.43}$$

Equation (2.42) can be simplified as follows when $\zeta = 1$:

$$\phi_e(t) = \Delta\omega_i\left\{\frac{\omega_p}{\omega_n{}^2}+e^{-\omega_n t}\left[t\left(1-\frac{\omega_p}{\omega_n}\right)-\frac{\omega_p}{\omega_n{}^2}\right]\right\} \tag{2.44}$$

When $\omega_p = 0$, (2.44) becomes:

$$\phi_e(t) = \Delta\omega_i e^{-\omega_n t} t \tag{2.45}$$

However, when $\zeta < 1$, equation (2.42) becomes:

$$\phi_e(t) = \Delta\omega_i \frac{\omega_p}{\omega_n^2}\left\{1 - e^{-\zeta\omega_n t}\left[\cos\left(\omega_n t\sqrt{1-\zeta^2}\right) + \frac{\zeta - \frac{\omega_n}{\omega_p}}{\sqrt{1-\zeta^2}}\sin\left(\omega_n t\sqrt{1-\zeta^2}\right)\right]\right\} \quad \textbf{(2.46)}$$

which further simplifies to:

$$\phi_e(t) = \Delta\omega_i \frac{\omega_p}{\omega_n^2}\left\{1 - e^{-\zeta\omega_n t}\sqrt{1 + \frac{\left(\zeta - \frac{\omega_n}{\omega_p}\right)^2}{1-\zeta^2}}\sin\left[\omega_n t\sqrt{1-\zeta^2} + \arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta - \frac{\omega_n}{\omega_p}}\right)\right]\right\} \quad \textbf{(2.47)}$$

For the case which $\zeta < 1$ and $\omega_p = 0$, (2.43) becomes:

$$\phi_e(t) = \frac{\Delta\omega_i}{\omega_n}\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin\left(\omega_n t\sqrt{1-\zeta^2}\right) \quad \textbf{(2.48)}$$

Figure 2.12 represents (2.48) for different values of damping coefficients.

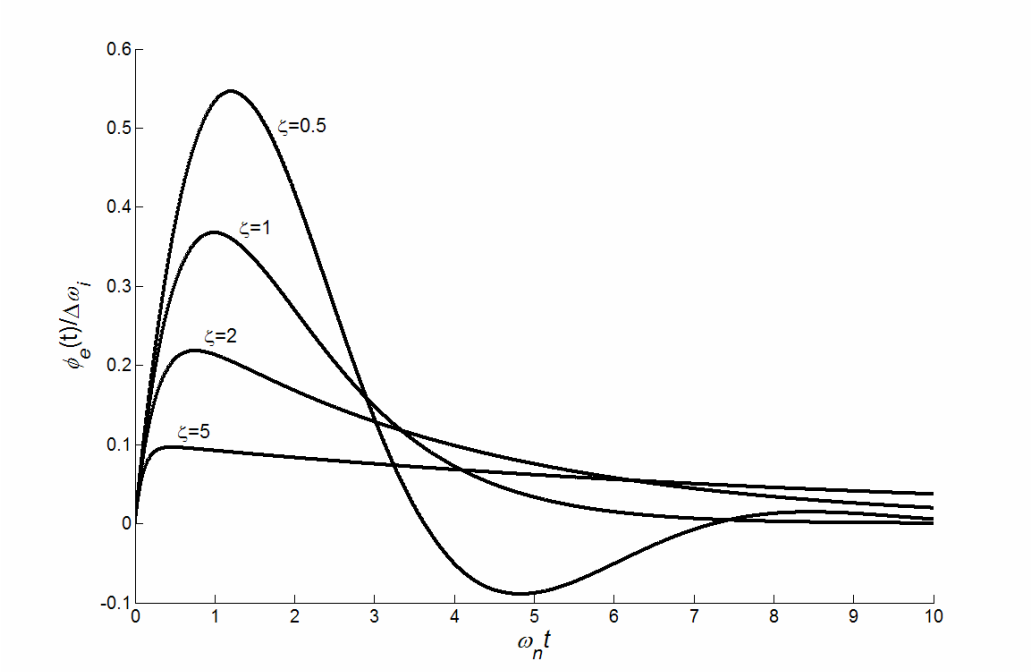


**Figure 2.12:** Phase Error of a Second Order PLL for a Frequency Step Input

Taking the derivative of the curves in Figure 2.12 gives the frequency errors, which are shown in Figure 2.13.

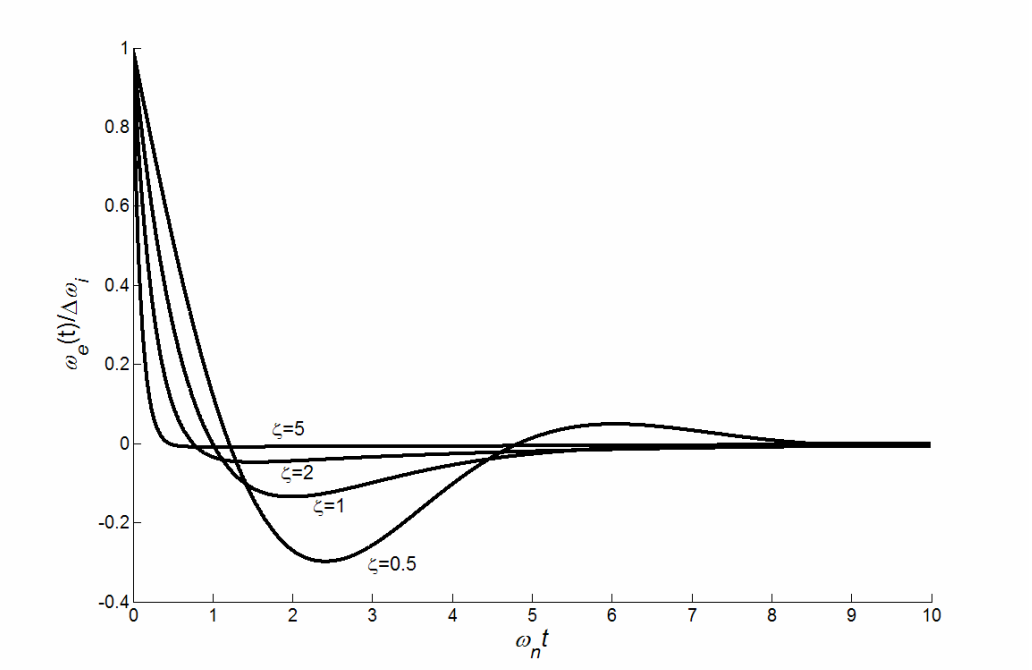


**Figure 2.13:** Frequency Error of a Second Order PLL for a Frequency Step Input

In Figure 2.13, it is noticeable that the frequency error starts from 1 and approaches to 0 as expected. It can also be noted that both of the plots in Figure 2.12 and 2.13 are in the form of damping oscillations with the same oscillation frequency and envelope but different initial phases.

### 2.3.2.3 Frequency Ramps

Substituting (2.18) into (2.32) gives the following transfer function:

$$\Phi_e(s) = \frac{m}{s^2}\frac{(s+\omega_p)}{s^2+2\xi\omega_n s+\omega_n{}^2} \tag{2.49}$$

Taking the inverse Laplace transform of (2.49) gives:

$$\phi_e(t) = \frac{m}{\omega_n{}^3}\left\{t\omega_n\omega_p - A + e^{-\zeta\omega_n t}\left[A\cosh\left(\omega_n t\sqrt{\zeta^2-1}\right) + \frac{B}{\sqrt{1-\zeta^2}}\sinh\left(\omega_n t\sqrt{\zeta^2-1}\right)\right]\right\} \tag{2.50}$$

where $A = 2\zeta\omega_p - \omega_n$ and $B = 2\zeta^2\omega_p - \zeta\omega_n - \omega_p$.

Equation (2.50) can be simplified as follows when $\zeta = 1$:

$$\phi_e(t) = \frac{m}{\omega_n^3}\left\{\omega_n - 2\omega_p + t\omega_n\omega_p + e^{-t\omega_n}\left[t\left(\omega_n\omega_p - \omega_n^2\right) - \omega_n + 2\omega_p\right]\right\} \quad \textbf{(2.51)}$$

When $\zeta < 1$, equation (2.50) becomes:

$$\phi_e(t) = \frac{m}{\omega_n^3}\left\{t\omega_n\omega_p - A + e^{-\zeta\omega_n t}\left[A\cos\left(\omega_n t\sqrt{1-\zeta^2}\right) + \frac{B}{\sqrt{1-\zeta^2}}\sin\left(\omega_n t\sqrt{1-\zeta^2}\right)\right]\right\} \quad \textbf{(2.52)}$$

which further simplifies to:

$$\phi_e(t) = \frac{m}{\omega_n^3}\left\{t\omega_n\omega_p - A + e^{-\zeta\omega_n t}\sqrt{A^2 + \frac{B^2}{1-\zeta^2}}\sin\left[\omega_n t\sqrt{1-\zeta^2} + \arctan\left(\frac{A}{B}\right)\right]\right\} \quad \textbf{(2.53)}$$

Equation (2.53) shows that as time increases, the phase error for a frequency ramp input becomes linear for a type I PLL, thus the frequency error becomes constant, which is consistent with (2.20). Similarly, as shown in (2.21), for a type II PLL (2.53) becomes constant. Graphical representation of (2.53) for a type II PLL is shown in Figure 2.14.

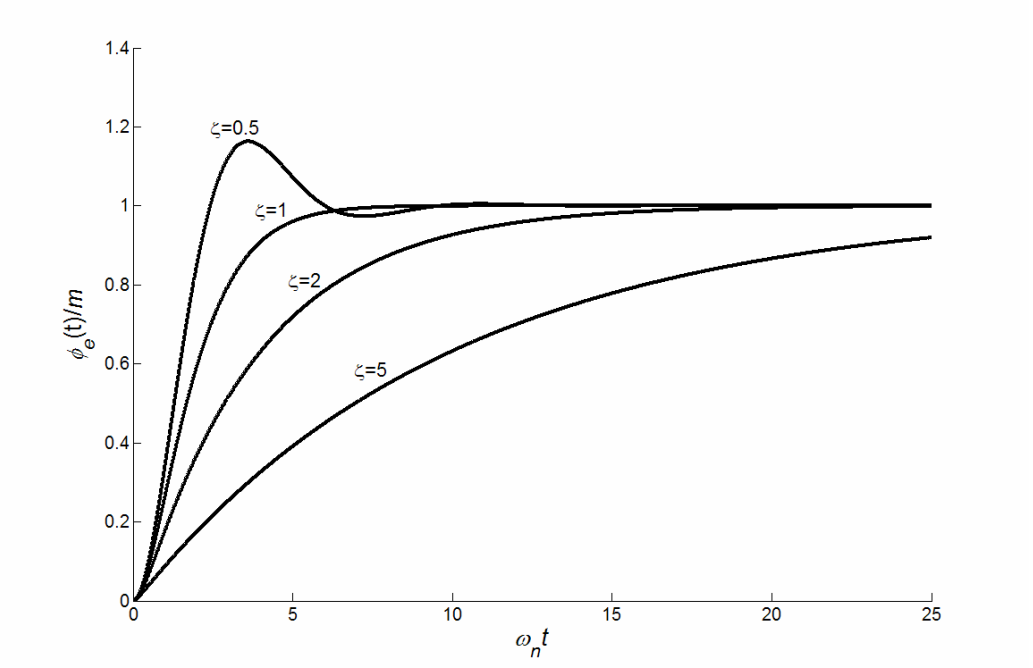


**Figure 2.14:** Phase Error of a Second Order PLL for a Frequency Ramp Input

As shown in Figure 2.15, taking the derivative of the curves in Figure 2.14 gives the frequency error for a frequency ramp input.

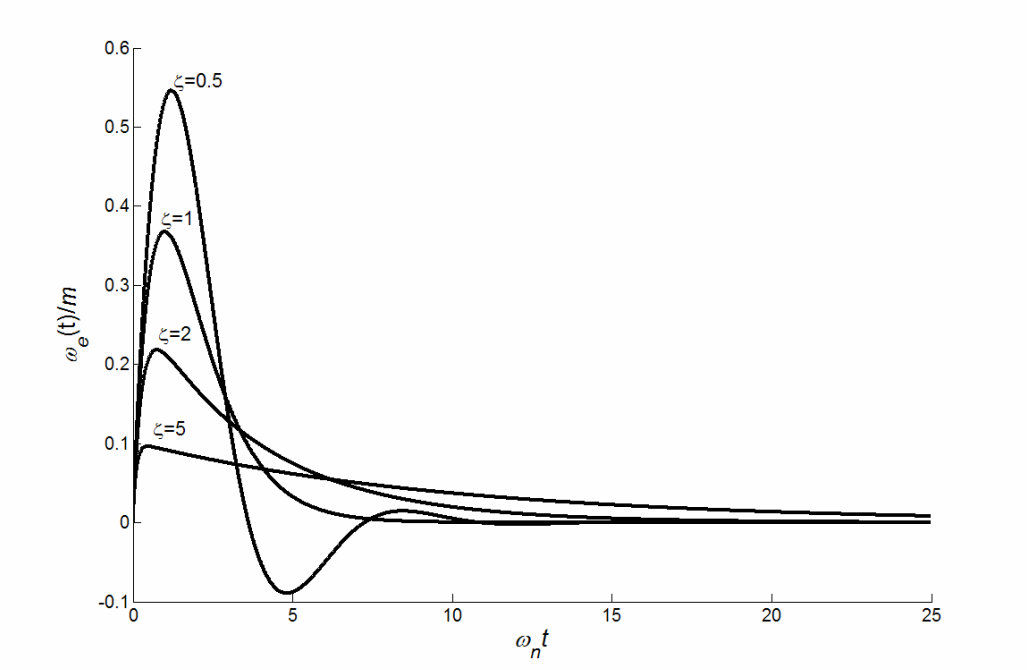


**Figure 2.15:** Frequency Error of a Second Order PLL for a Frequency Ramp Input

## 2.4 Working Ranges of PLLs

In previous sections, all blocks of the PLL are assumed to be operating in the linear region, thus s domain representations were accurate enough to model their behaviors. However, when the amplitude of the step or ramp input is large, it may cause these blocks enter into non-linear region, where s domain transfer functions are no longer valid. At this point, it's necessary to divide the operating regions of PLLs into different ranges [1,5].

The main block which determines whether the PLL is operating in the linear or non-linear region is the phase detector. Phase detectors must behave non-linear by definition, since phase differences of any two signals are in the range of [-2π, 2π]. When the phase detector, thus the PLL, is operating in the non-linear region, the phase difference caused by non-zero frequency difference can reach to 2π, which is called cycle slipping.

Lock-in range is defined as the maximum frequency range for which a PLL locks without cycle slipping. PLL behaves in a linear fashion when the input frequency step is inside the lock-in range, thus the s domain representations are valid.

In order to find the lock-in range, the maximum phase error for a frequency step input should be defined. For a common case, which is a type II second order PLL with unity damping ratio, the first derivative of the phase error defined in (2.45) becomes:

$$\frac{d\phi_e(t)}{d(\omega_n t)} = \frac{\Delta\omega_L}{\omega_n}\left(e^{-\omega_n t} - \omega_n e^{-\omega_n t} t\right) \tag{2.54}$$

which becomes zero for $\omega_n t = 0$. In this condition, the maximum phase error becomes:

$$\phi_e(t)_{\max} = \frac{\Delta\omega_L}{e\omega_n} \tag{2.55}$$

Solving for $\Delta\omega_L$ and substituting $\phi_e(t)_{\max}$ with $\phi_{PD\max}$ ,which is the maximum input phase for which the PD behaves linear, gives:

$$\Delta\omega_L = \phi_{PD\max} e\omega_n \tag{2.56}$$

Equation (2.56) shows that the lock-in range is defined by the PD type and the natural frequency. Using (2.56), the maximum output frequency step for linear region becomes:

$$\Delta\omega_{o\max} = N\Delta\omega_L = N\phi_{PD\max} e\omega_n \tag{2.57}$$

which is the maximum output frequency step for a frequency synthesizer without causing cycle slips.

Hold-in range is the maximum frequency range for which an initially locked PLL does not loses lock. It is defined as the static stability limit and depends on the DC loop gain only. It is derived from the maximum output frequency that the VCO can produce, neglecting the saturation of the frequency characteristics.

$$\Delta\omega_H = \pm\frac{K_{PD}K_{VCO}F(0)}{N} \quad \textbf{(2.58)}$$

For a type I loop, F(0) is a finite value. However, type II loops have two integrators one of which is inside the loop filter. Thus the DC gain of the loop filter is infinite, resulting with an infinite hold-in range. In a more realistic point of view, type II loops have hold-in ranges equal to the frequency ranges of their VCOs.

Pull-in range is the maximum frequency range for which a PLL can acquire lock eventually. It is defined as the dynamic stability limit. Pull-in range is determined using the differential equations of PLLs. For a type II second order PLL, pull-in range is infinite, and the pull-in time is defined as:

$$T_P = \frac{(\Delta\omega_i)^2}{2\zeta\omega_n^3} \quad \textbf{(2.59)}$$

Equation (2.59) shows that the pull-in time is inversely proportional to $\omega_n^3$.

Pull-out frequency is the maximum frequency step that can be applied to an initially locked loop without causing to lose lock. Its value is found empirically using the phase plane of PLLs. Its value is approximately:

$$\Delta\omega_{PO} \cong 1.8\omega_n(\zeta+1) \quad \textbf{(2.60)}$$

for $\zeta$ between 0.5 and 1.4.

Overall summary of working ranges of PLL is shown in Figure 2.18.

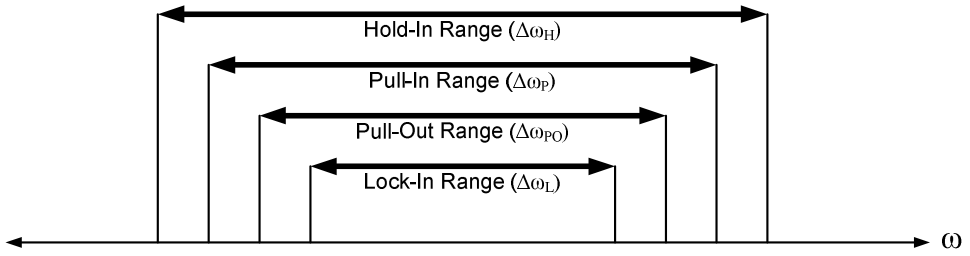


**Figure 2.16:** Working Ranges of PLLs

As seen in Figure 2.16 working ranges of PLLs can be ordered as below:

$$\Delta\omega_L < \Delta\omega_{PO} < \Delta\omega_P < \Delta\omega_H \quad \textbf{(2.61)}$$

## 2.5 Noise Performance

Although PLLs are complicated blocks with many interesting properties, their s domain transfer functions show that they also behave like low-pass filters. In order to measure the ability of a filter to reject noise, an equivalent rectangular noise bandwidth is defined as shown in Figure 2.17, which can be applied to PLL systems [1,8].

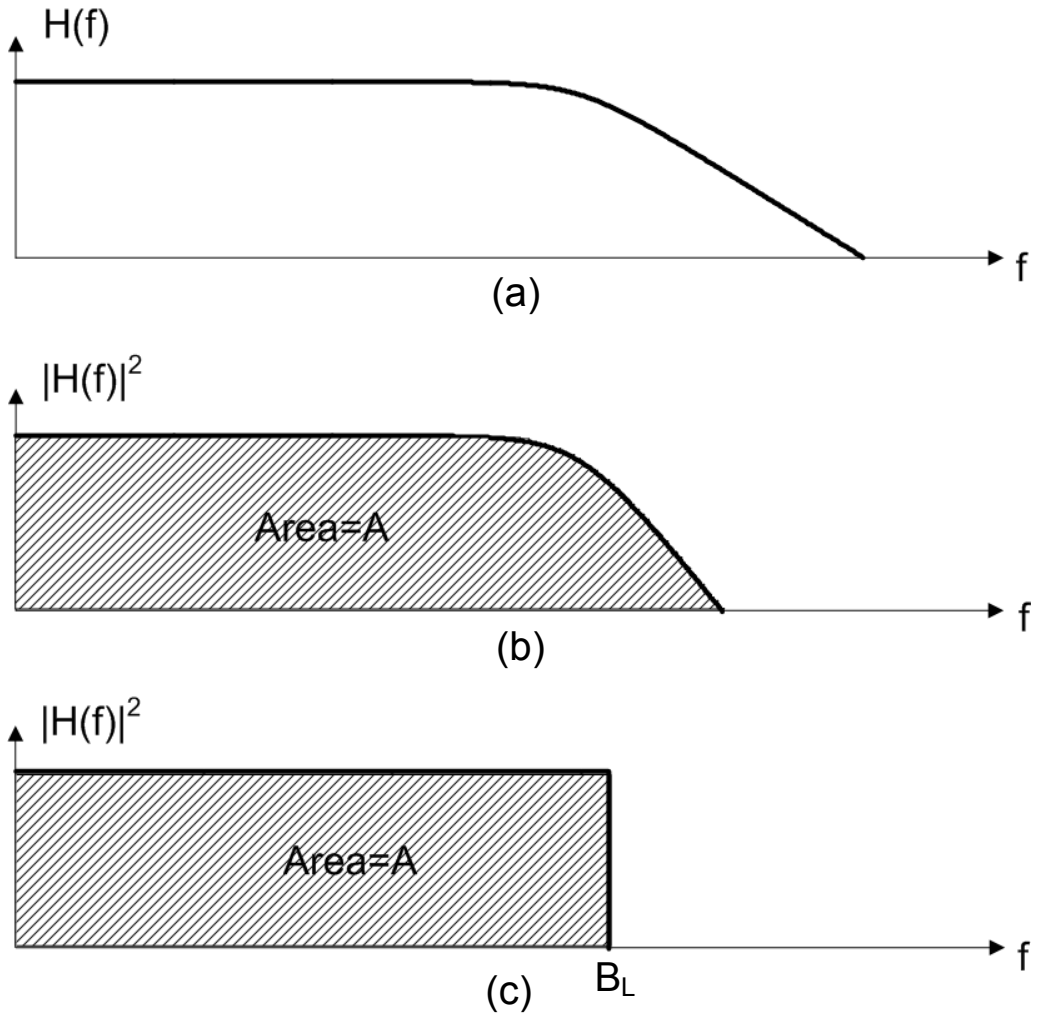


**Figure 2.17:** Definition of Equivalent Rectangular Noise Bandwidth

A typical low-pass filter response and square of the response are shown in Figure 2.17a and Figure 2.17b respectively. The noise bandwidth $B_L$ is simply defined as the

frequency makes the area under the curves of Figure 2.17b and Figure 2.17c equal. Using this definition, $B_L$ is formulated as:

$$B_L = \int_0^\infty \left| G(f) \right|^2 df \tag{2.62}$$

For an ideal filter with infinitely sharp cutoff, the noise bandwidth of the filter is equal to the cutoff frequency. For any filter of finite degree, the noise bandwidth is larger than the cutoff frequency.

Substituting a typical first order filter transfer function into (2.62) gives:

$$B_L = \int_0^\infty \left| \frac{\omega_0}{j\omega + \omega_0} \right|^2 \frac{d\omega}{2\pi} = \frac{\omega_0}{4} = \frac{\pi}{2} f_0 \tag{2.63}$$

Equation (2.63) also shows the noise bandwidth for a first order PLL.

Applying a general second order filter function to (2.62) gives:

$$B_L = \frac{1}{2\pi} \int_0^\infty \left| \frac{\phi_o(\omega)}{N\phi_i(\omega)} \right|^2 d\omega = \frac{1}{2\pi} \int_0^\infty \left| \frac{2\zeta\omega_n j\omega + \omega_n{}^2}{-\omega^2 + 2\zeta\omega_n j\omega + \omega_n{}^2} \right|^2 d\omega = \pi f_n \left( \zeta + \frac{1}{4\zeta} \right) \tag{2.64}$$

which shows that the noise bandwidth is proportional to the natural frequency.

Taking the first derivative of (2.64) gives:

$$\frac{dB_L}{d\zeta} = \pi f_n \left( \frac{4\zeta^2 - 1}{4\zeta^2} \right) \tag{2.65}$$

which is equal to zero for $\zeta = 0.5$. Using this statement, it can be concluded that minimum noise bandwidth is achieved when the natural frequency is as low as possible and the damping ratio is 0.5.

## 3. THE PROPOSED PHASE-LOCKED LOOP STRUCTURE

The proposed PLL structure is based on a typical charge pump PLL since it is the most widely used structure in PLL design. Thus, first the basic properties of the charge pump PLL are given and the transient formulas extracted in Chapter 2 are modified for its design parameters. However it's worth mentioning that the idea can be implemented into every PLL structure.

### 3.1 Charge Pump PLL

As seen in Figure 3.1, a typical charge pump PLL consists of 5 blocks, namely phase frequency detector (PFD), charge pump, loop filter, VCO and divider.

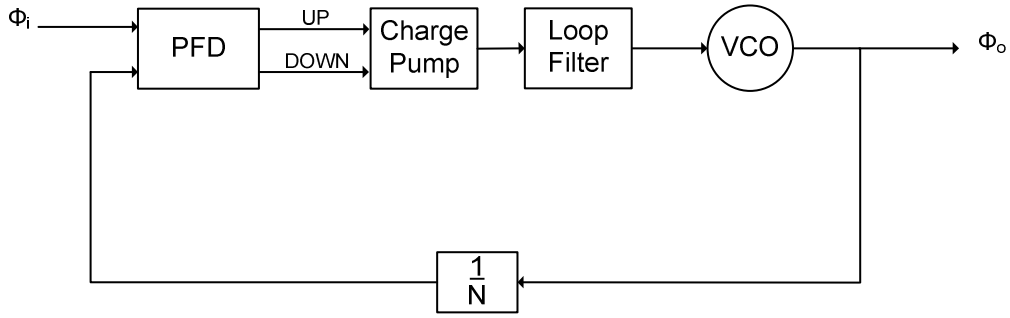


**Figure 3.1:** Charge Pump PLL

#### 3.1.1 Phase Frequency Detector

The most widely used PFD structure is shown in Figure 3.2. It is simply a sequential circuit which produces UP and DOWN outputs, telling the charge pump to increase or decrease the VCO input voltage [9].

As seen in Figure 3.3, the main principle of the PFD is to measure the time between the rising edges of the inputs, and to change the duty cycle of the output accordingly, providing 100% duty cycle for a maximum phase difference of $\pm 2\pi$.

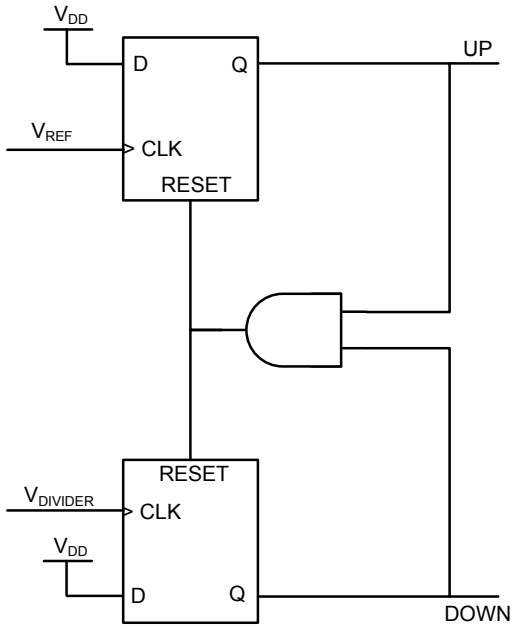


**Figure 3.2:** PFD Circuit

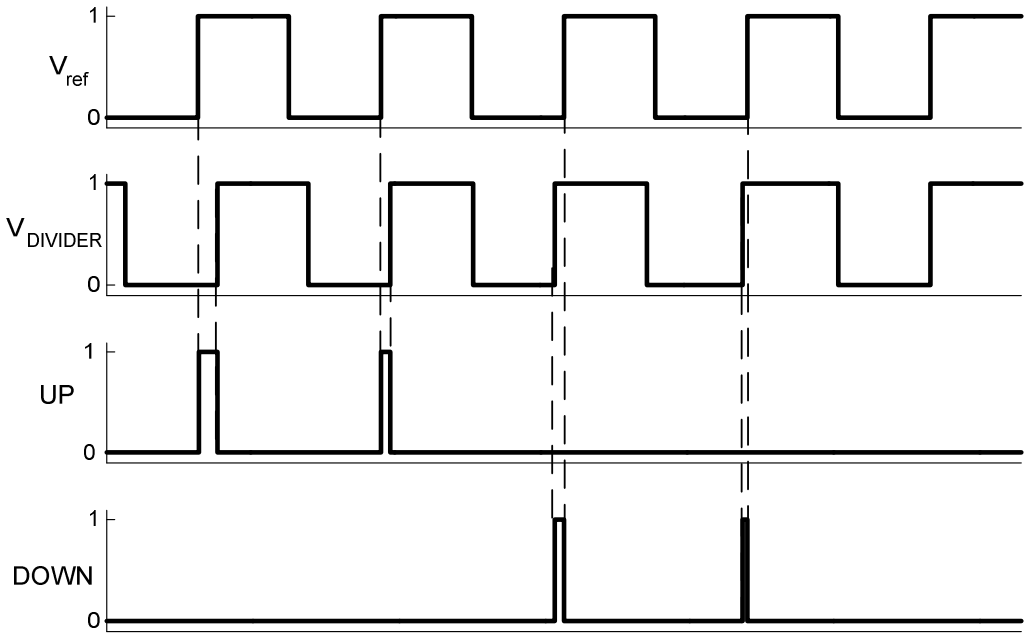


**Figure 3.3:** PFD Waveforms

If $V_{REF}$ triggers the PFD first, then the output frequency should be increased, so the UP output becomes logic high until $V_{DIVIDER}$ rises, otherwise the DOWN output becomes high until $V_{REF}$ rises.

### 3.1.2 Charge Pump

An ideal representation of the charge pump circuit is shown in Figure 3.4.

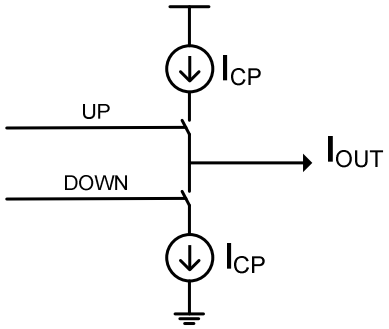


**Figure 3.4:** Charge Pump Circuit

The idea behind the charge pump circuit is simply to push current out if UP is high and pull current in if DOWN is high [10].

Taking the average output current for a PFD – Charge Pump pair results with the output current vs. phase difference graph shown in Figure 3.5 [3].

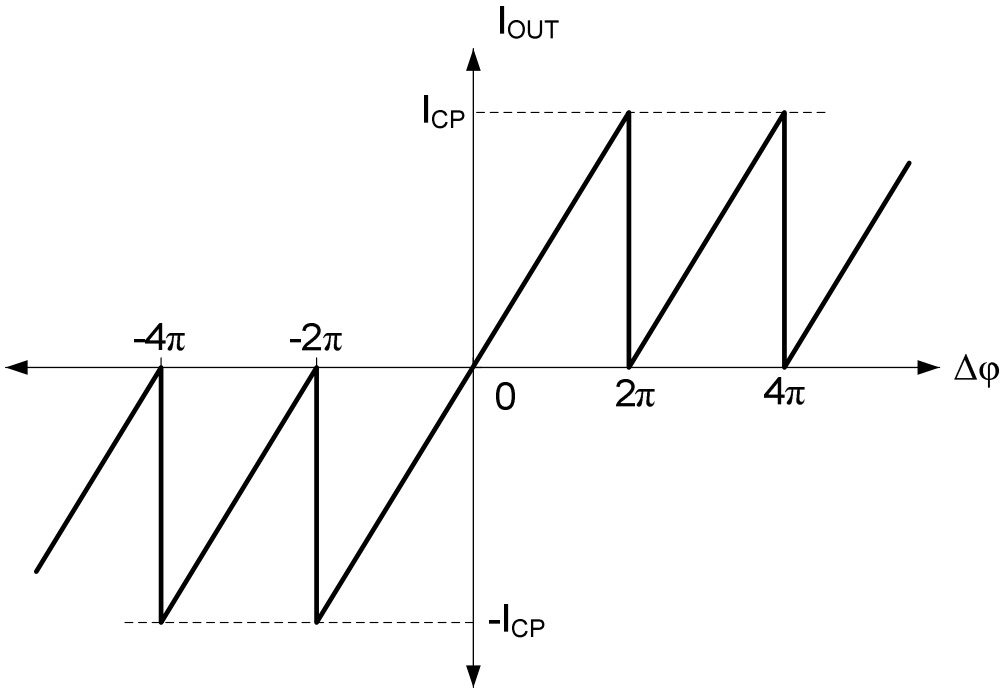


**Figure 3.5:** PFD – Charge Pump Pair Waveform

As seen in Figure 3.5, for a phase difference of [-2π, 2π] the PFD behaves linearly, thus taking the derivative in the linear region results with the PFD – Charge Pump pair gain:

$$K_{PD} = \frac{I_{CP}}{2\pi}[A/rad] \tag{3.1}$$

### 3.1.3 Loop Filter

Since the VCO needs a control voltage but the PD output is in current form, the transfer function of loop filter circuits are in the transimpedance form.

The capacitor is the simplest loop filter for a charge pump PLL. In the case where only a capacitor is used as a loop filter, the filter function becomes:

$$F(s) = Z(s) = \frac{1}{sC} \tag{3.2}$$

Using (3.2) and (2.2) gives:

$$H(s) = \frac{\frac{K}{s^2 C}}{1 + \frac{K}{s^2 CN}} = N \frac{\frac{K}{CN}}{s^2 + \frac{K}{CN}} \tag{3.3}$$

Equation (3.3) is a typical function of an oscillator [3]. Physically, this result means that the PLL never settles if only a capacitor is used as a loop filter. From a control theory point of view, this system is controlled by an integral-only controller, which is known to be unstable.

In order to stabilize the system in (3.3), a zero must be added to transfer function [3]. This can be achieved by changing the loop filter into a resistor and capacitor connected in series. The resulting filter function becomes:

$$F(s) = Z(s) = R + \frac{1}{sC} = \frac{sRC + 1}{sC} \tag{3.4}$$

Substituting (3.4) into (2.1) gives the open loop transfer function:

$$G(s) = \frac{K}{s^2 C}(sRC + 1) = \frac{KR}{s^2}\left(s + \frac{1}{RC}\right) \tag{3.5}$$

The corresponding closed loop transfer function is:

$$H(s) = N\frac{\frac{K}{N}\left(\frac{sRC+1}{sC}\right)}{s+\frac{K}{N}\left(\frac{sRC+1}{sC}\right)} = KR\frac{s+\frac{1}{RC}}{s^2+s\frac{KR}{N}+\frac{K}{NC}} \qquad \textbf{(3.6)}$$

Equation (3.6) shows that the charge pump PLL is a second order PLL. Considering that the capacitor in the loop filter acts like an integrator, it can also be stated that the charge pump PLL is a type II loop.

For the filter in (3.4), the output ripple is unacceptably high. In order to suppress the output ripple, a small capacitor $C_2$ is added. The final loop filter is shown in Figure 3.6.

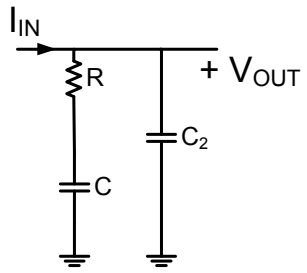


**Figure 3.6:** Charge Pump Loop Filter with Two Capacitors

As a rule of thumb, this second capacitor is chosen to be one twentieth of the original charge pump capacitor. This rule supplies acceptable output ripple while still maintaining stability. For the sake of simplicity, the second capacitor is not included in the calculations.

### 3.1.4 Noise of Loop Filter

In order to investigate the noise of the loop filter, a more detailed model should be used, which takes leakage current into account [1]. Such model is shown in Figure 3.7.

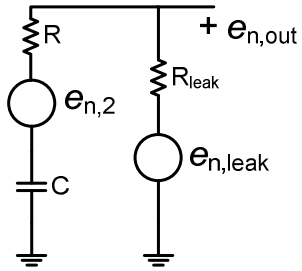


**Figure 3.7:** Noise Model for Loop Filter

The noise voltages generated by resistors are:

$$e_{n,2}^2 = 4kTR \tag{3.7}$$

and

$$e_{n,leak}^2 = 4kTR_{leak} \tag{3.8}$$

Using (3.7) and (3.8), the output noise is approximately:

$$e_{n,out} = \left(e_{n,2} + e_{n,leak}\right)\left(\frac{1+sT_2}{1+sT_{leak}}\right) \tag{3.9}$$

where $T_2$=RC and $T_{leak}$=$R_{leak}$C for $R_{leak}$>>R. Thus the power spectral density (PSD) of the output noise becomes:

$$S(f) = 4kT\left(R + R_{leak}\right)\left|\frac{1+sT_2}{1+sT_{leak}}\right|^2 \tag{3.10}$$

Since the loop filter noise shown in (3.10) directly contributes to the output phase noise, it needs to be as low as possible.

### 3.1.5 Steady State Errors and Transient Response

Since the charge pump PLL is a type II loop, the steady state phase and frequency errors for phase step and frequency step inputs are equal to zero. Using (2.21) and (3.5), the phase error for a frequency ramp input can be calculated as:

$$\lim_{t\to\infty}\left[\phi_e(t)\right] = m\frac{NC}{K} \tag{3.11}$$

In order to use transient response formulas extracted in Chapter 2, $\omega_p$, $\omega_n$ and $\zeta$ constants should be known. As it was mentioned before, $\omega_p$ is equal to zero for type II loops. The other constants can be defined as:

$$\omega_n = \sqrt{\frac{K}{NC}} \tag{3.12}$$

and

$$\zeta = \frac{1}{2}\sqrt{\frac{KR^2C}{N}} \tag{3.13}$$

Using (3.12) and (3.13), transient response of a charge pump PLL can be calculated.

### 3.2 PID Implemented PLL Structure

In the conventional charge pump PLL structure examined previously, in order to change the voltage of the loop filter, a large capacitance should be charged. The optimum settling can be achieved by setting the damping ratio and the natural frequency to proper values. However, the proper values for the damping ratio and the natural frequency increase the noise bandwidth, resulting a trade-off between the transient response and the output noise. In order to overcome this issue, a controller is implemented into PLL so that the transient response is optimized without sacrificing from the noise properties.

There are some main considerations for implementing a controller into the PLL [11,12].

1. The controller should give a designer the ability to tune the transient performance of the PLL.
2. The controller should not alter the PLL's stability.
3. The controller should not increase the noise.

Using the above considerations, the PLL structure shown in Figure 3.8 is proposed. In Figure 3.8, the additional block is shown in dashed line.

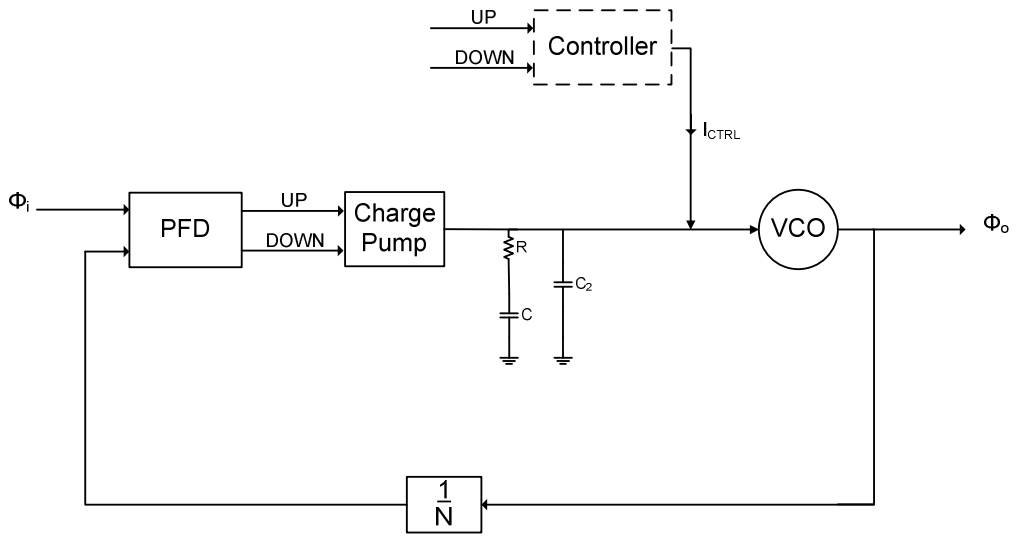


**Figure 3.8:** Controller Implemented PLL Structure

Considering Figure 3.8, below statements can be made:

1. Controller drives the loop filter with a control current, resulting a loop filter current equal to the sum of the charge pump current and the control current.
2. The controller is directly connected to VCO input, thus any noise coming from the controller increases the phase noise. In order to overcome this issue, an enable / disable circuit must be implemented.

The proposed controller structure is shown in Figure 3.9.

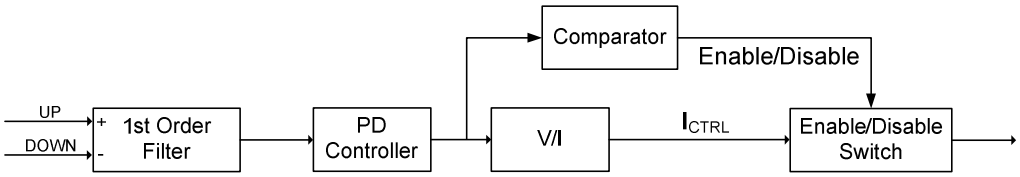


**Figure 3.9:** Proposed Controller Structure

The working principle of the controller shown in Figure 3.9 is as follows:

1. The first order filter takes the voltage difference between the UP and DOWN signals and filters the difference so that the amount of error is achieved.
2. PD controller gets the amount of error as an input. It gives the sum of the first derivative of the input and the input itself, each multiplied with coefficients.
3. V/I converter gets the output voltage of the PD controller and generates the control current.

4. Comparator compares the output of the PD controller with a voltage window and controls the Enable/Disable switch accordingly.
5. Enable/Disable switch simply behaves like a short circuit if the error is high and behaves like an open circuit if the error is low, thus making the PLL more reactive to changes without contributing additional noise.

The controller is designed using 0.35µm CMOS process provided by AustriaMicroSystems (AMS) with typical supply voltage of 3.3V.

### 3.2.1 1st Order Filter

The purpose of the 1st order filter is to filter the difference between the UP and DOWN signals. The used 1st order filter structure is shown in Figure 3.10.

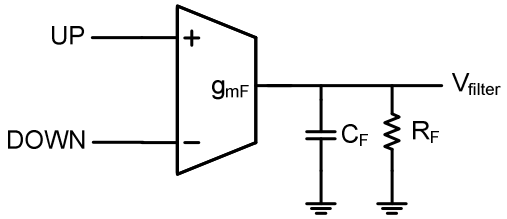


**Figure 3.10:** Used 1st Order Filter

Shown in Figure 3.11, the operational transconductance amplifier (OTA) used in the 1st order filter is a symmetrical cascode OTA with PMOS inputs. The sizes of the MOSFETs in the OTA are given in Table 3.1.

**Table 3.1:** Sizes of MOSFETs in the OTA

| $M_{P1}$,$M_{P2}$ | $M_{N1}$-$M_{N8}$ | $M_{P3}$-$M_{P6}$ |
|---|---|---|
| 25µm/0.35µm | 10µm/0.7µm | 20µm/0.7µm |

Tail current is generated by a cascoded PMOS current mirror with 15µm/0.7µm sized MOSFETs.

Since the difference between the UP and DOWN signals are either $V_{DD}$ or $-V_{DD}$, the maximum output current, rather than the $g_m$ of the OTA, determines the filter behavior. In such case, the OTA behaves like a charge pump, thus the transfer

function is the same as a charge pump with a current equal to the maximum output current of the OTA.

The filter structure has large output impedance, thus a voltage buffer is necessary for proper operation. For this purpose, an NMOS source follower circuit is employed, shown in Figure 3.12. The sizes of NMOSes in source follower are 15µm/0.35µm and 10µm/1µm for $M_{N1}$ and $M_{N2}$ respectively. $V_{bias}$ is generated by a current mirror with a reference current of 20µA.

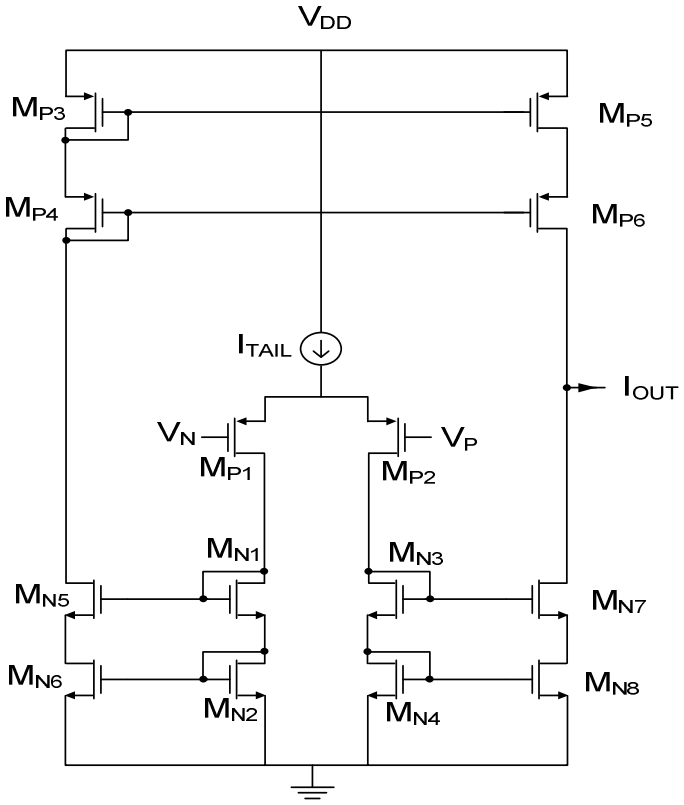


**Figure 3.11:** Schematic of the OTA

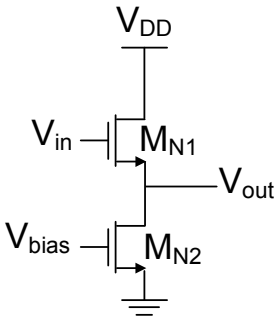


**Figure 3.12:** Schematic of the Source Follower

### 3.2.2 PD Controller

The PD controller gives the sum of the first derivative of the input and the input itself, each multiplied with coefficients. In order to achieve this, a derivative block is used. However, lossless derivative blocks can raise stability issues, thus a lossy derivative block is necessary. The used derivative block is simply a resistor and a capacitor connected in high pass configuration, shown in Figure 3.13.

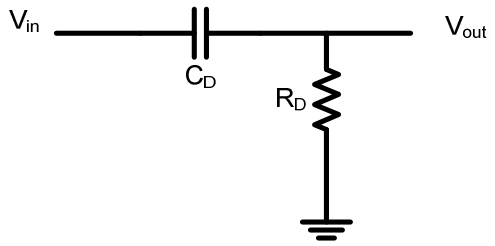


**Figure 3.13:** Derivative Block

The transfer function of the derivative block in Figure 3.13 is:

$$H_D(s) = \frac{sR_DC_D}{sR_DC_D + 1} = \underbrace{sR_DC_D}_{Derivative} \underbrace{\frac{1}{sR_DC_D + 1}}_{1st\ order\ filter} \tag{3.14}$$

Equation (3.14) can be considered as cascade connected first order filter and a lossless derivative block with a gain of $R_DC_D$. For practical reasons, the resistor and the capacitor for the derivative block are chosen to be 25K and 1pF respectively.

In order to implement $K_D$, $K_P$ and the adder blocks, a summing amplifier consisting of OTAs is used.

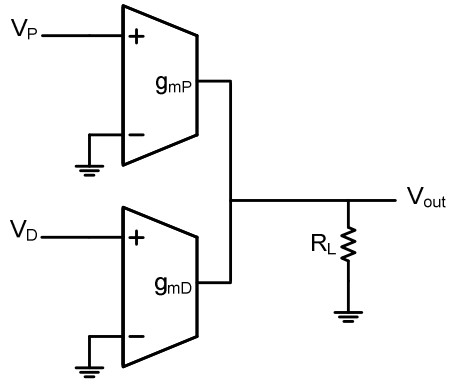


**Figure 3.14:** Summing Amplifier

Since linearity is important for $K_P$ and $K_D$ blocks, OTAs are linearized using method proposed by Nedungadi-Viswanathan [14]. The input stage of the linearized OTAs is shown in Figure 3.15. Sizes of $M_{P1a}$ and $M_{P2a}$ are chosen to be 25µm/0.35µm. The same OTA is employed for the V/I converter at the final stage of the controller.

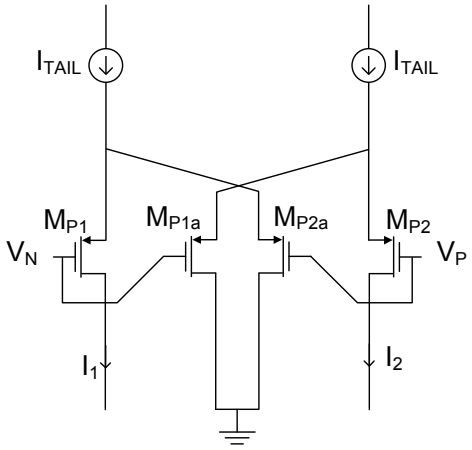


**Figure 3.15:** Linearized OTA Input Stage

The $g_m$ vs. tail current graph of the linearized OTA is given in Figure 3.16.

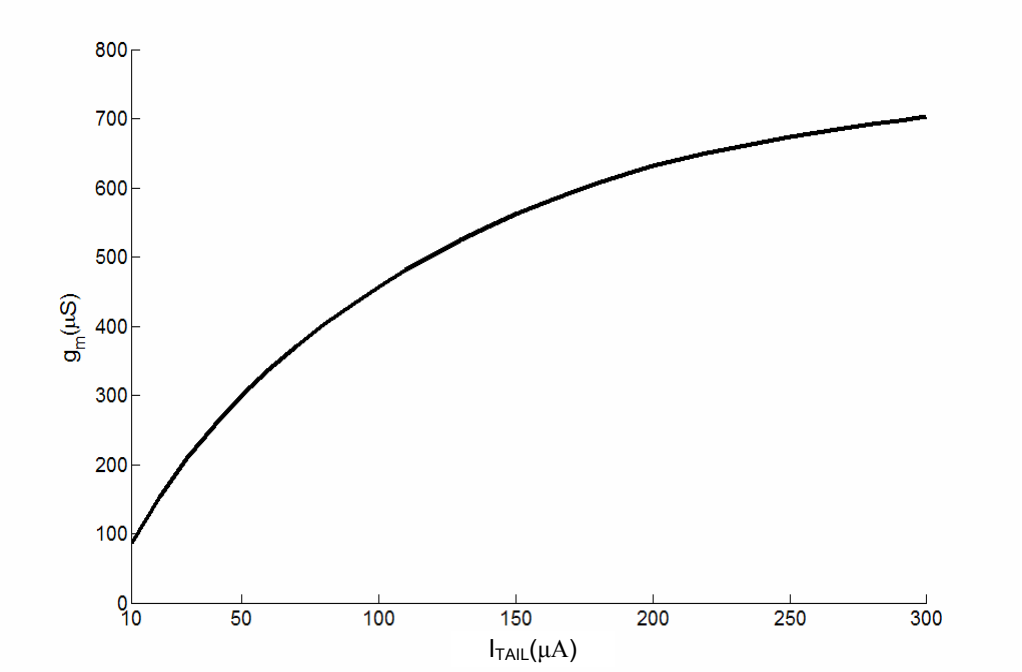


**Figure 3.16:** Transconductance vs. Tail Current

Combining the blocks shown in Figure 3.13 and 3.14 gives the overall PD controller circuit shown in Figure 3.17.

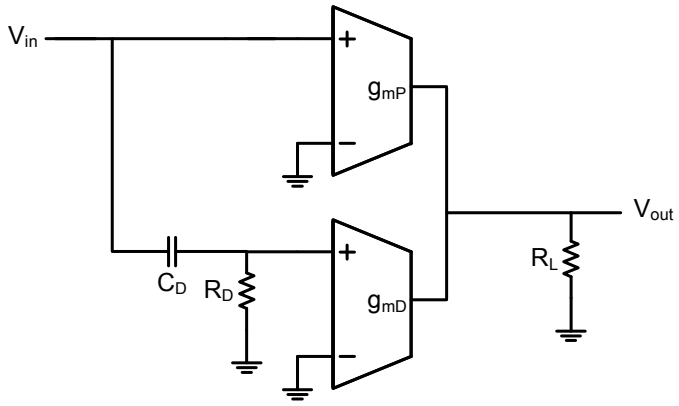


**Figure 3.17:** Overall PD controller

As seen in Figure 3.17, $g_{mD}$ is separated from the DC level of $V_{in}$, whereas $V_{in}$ is directly connected to $g_{mP}$. Recalling Figure 3.12, it can easily be seen that the DC level of $V_{in}$ is shifted. In order to equalize the DC level of both inputs for $g_{mP}$, a duplicate buffer is connected to the negative input of $g_{mP}$, just to ensure the same level shift.

Using Figure 3.17, the transfer function of the PD controller is extracted approximately as follows:

$$H_{PD}(s) = g_{mP}R_L + \frac{sR_DC_D}{sR_DC_D + 1} g_{mD}R_L \qquad \textbf{(3.15)}$$

For input voltages below the cutoff frequency of the 1st order filter shown in (3.14), the derivative block acts like a lossless derivative block with a gain of $R_DC_D$. Since the PD controller deals with relatively low frequencies, (3.15) can further be approximated as:

$$H_{PD}(s) \cong sK_DR_DC_D + K_P \qquad \textbf{(3.16)}$$

where $K_D = g_{mD}R_L$ and $K_P = g_{mP}R_L$.

Using Figures 3.10 3.17, the controller without the decision and enable/disable circuitry is shown in Figure 3.18.

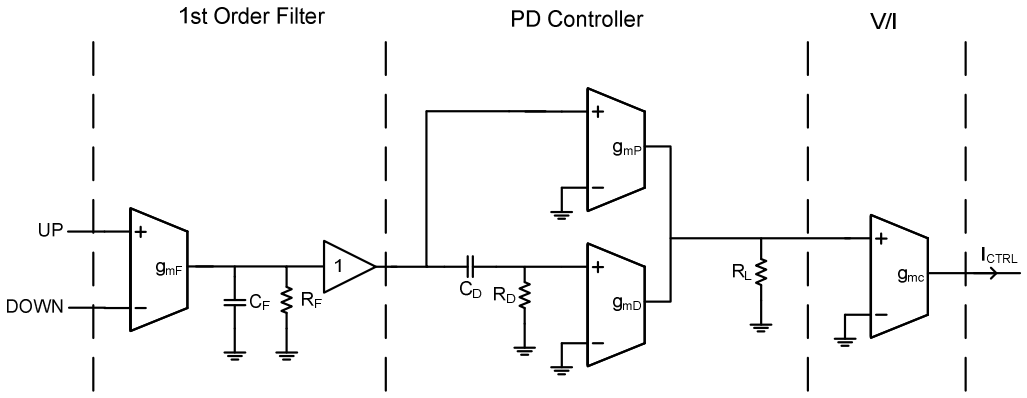


**Figure 3.18:** Overall Controller without Enable/Disable Circuitry

### 3.2.3 Comparator and Finite State Machine

Since the PLL has unique properties in terms of noise rejection, the controller should be disabled when the output frequency reaches a value reasonably close enough to the final output frequency. In order to achieve this, two comparators are connected to the output voltage of PD controller, so that the controller is enabled when the absolute value of PD output is larger than a predefined value. The employed comparator topology is shown in Figure 3.19 [15].

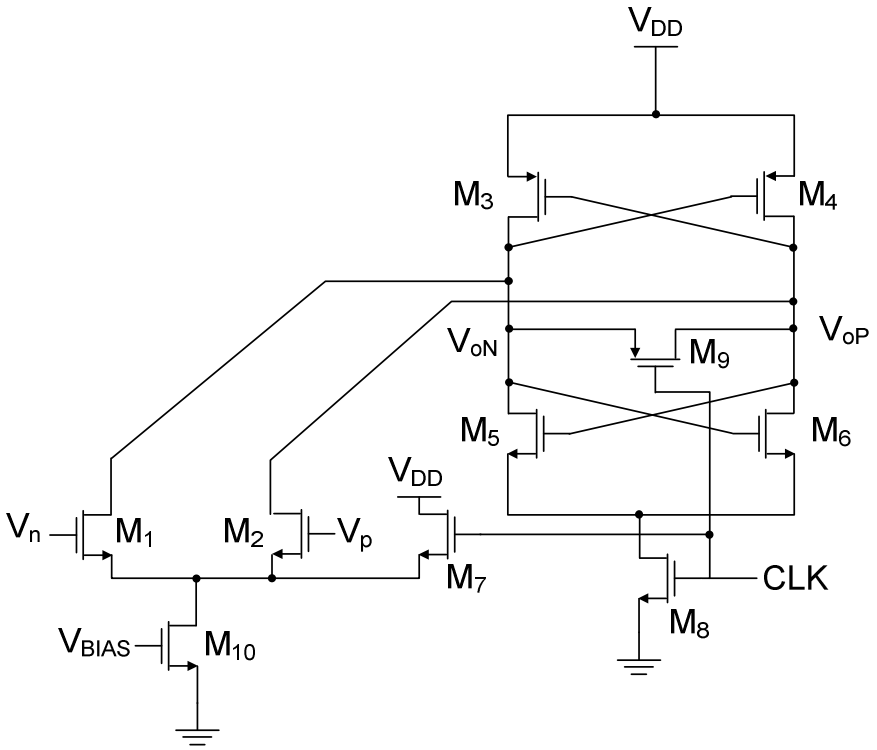


**Figure 3.19:** Employed Comparator Topology

As seen in Figure 3.19, the comparator is mainly a latch loaded differential pair. Triggered by CLK, additional MOSFETs ($M_7$-$M_9$) have secondary purposes such as enabling or disabling the differential pair or the latch. CLK input is connected to the reference CLK of the PLL, so that an additional CLK generator is not necessary. The sizes of MOSFETs in the comparator are shown in Table 3.2. $V_{BIAS}$ is generated by a current mirror with 10µA reference current.

**Table 3.2:** Sizes of MOSFETs in the Comparator

| $M_1$,$M_2$ | $M_3$,$M_4$ | $M_5$-$M_9$ | $M_{10}$ |
|---|---|---|---|
| 5µm/0.35µm | 3µm/0.35µm | 1µm/0.35µm | 5µm/1µm |

The outputs of the comparator settle to logic values when CLK is logic high. However, when CLK is at logic low phase, the output of the comparator is at approximately the threshold value of the inverters, thus it does not have a proper logical value. In order to make the output value constant during CLK changes, a D type flip-flop which is triggered with the inverse of the CLK, is employed. The final comparator configuration is shown in Figure 3.20.

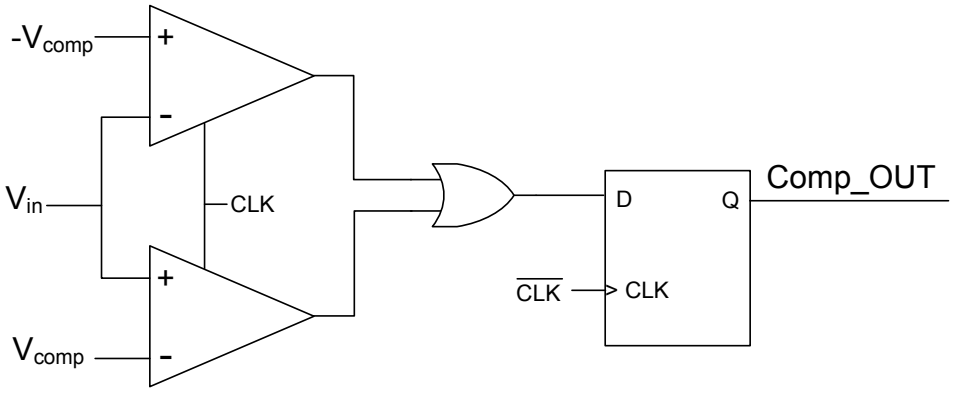


**Figure 3.20:** Final Comparator Configuration

As seen in Figure 3.20, when $V_{in}$, which is the output voltage of the PD controller, is larger than $V_{comp}$ or smaller than $-V_{comp}$, Comp_OUT value becomes logic high. However the output voltage of the PD controller crosses several times through $V_{comp}$ values during transient changes. This situation may result with the controller going on and off several times during the settling process, which is an undesired condition. In order to overcome this issue, a simple finite state machine is connected to

Comp_OUT, so that the controller is disabled only if the Comp_OUT value is logic low for three CLK periods. The finite state machine is shown in Figure 3.21. All logic blocks in Figures 3.20 and 3.21 are used from the standard logic library provided by AMS.

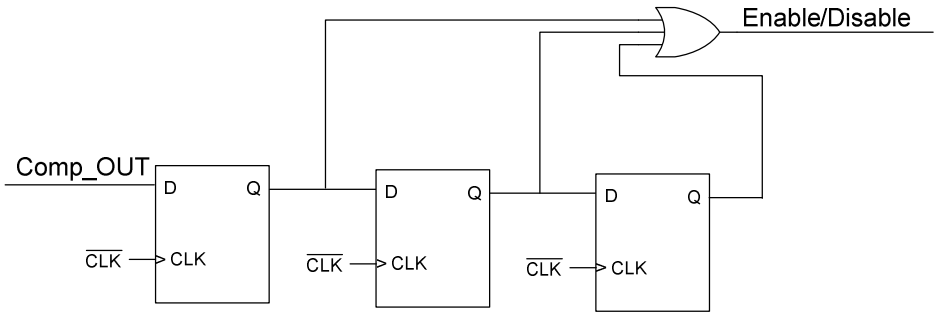


**Figure 3.21:** The Finite State Machine

### 3.2.4 Enable / Disable Switch

The main purpose of the Enable/Disable switch is to enable or disable the control current. This is achieved by a current steering CMOS switch, shown in Figure 3.22. The sizes of MOSFETs are 15µm/0.35µm and 30µm/0.35µm for NMOSes and PMOS, respectively.

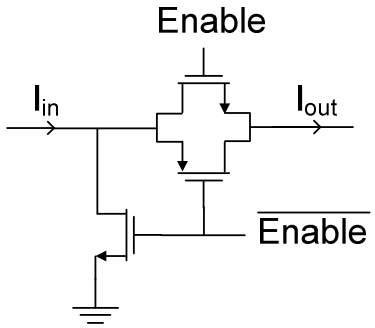


**Figure 3.22:** Enable/Disable Switch

Noise analyses show that when the switch is in off state, there is no additional noise contributed by the controller.

### 3.2.5 Layout of the Overall Controller

Layout of the overall controller is shown in Figure 3.22. The input stages of the OTAs are drawn in common centroid configuration with guard rings in order to reduce the mismatch effects. The size of the circuit is 160µm x 110µm in 0.35µm

CMOS process and the circuit is verified by post-layout simulations, producing the same results with the schematics.

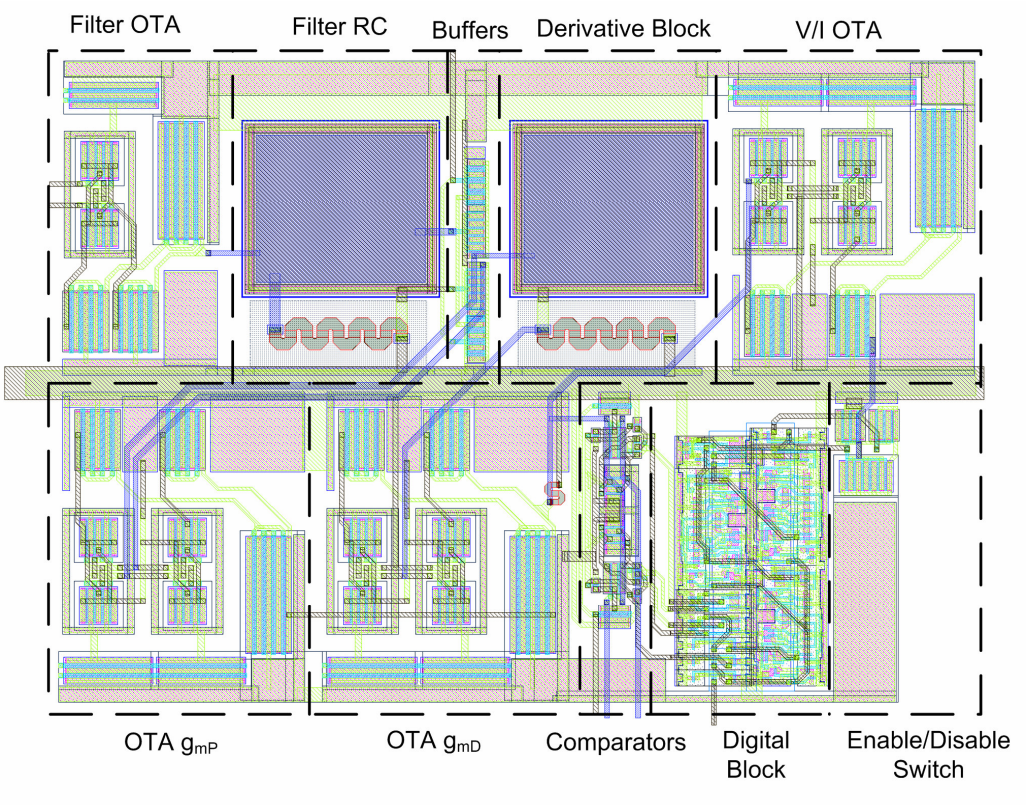


**Figure 3.23:** Layout of the Overall Controller

### 3.3 The Analysis of the Proposed PLL

Using the s domain transfer functions of the blocks mentioned in the previous sections and assuming that the bandwidth of the filter and the derivative block is large enough, the overall PLL circuit is shown in Figure 3.24.

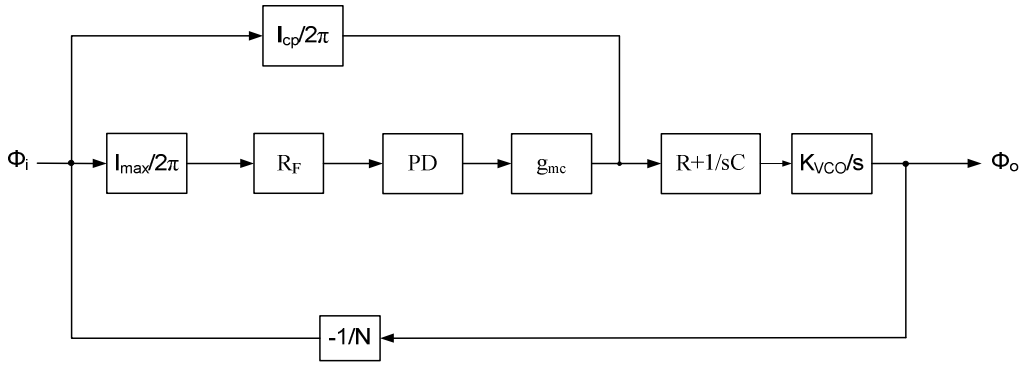


**Figure 3.24:** Overall PLL Circuit

Using Figure 3.24, the open loop transfer function of the proposed PLL becomes:

$$G(s)=\left[K_{CP}+K_{\max}R_F\left(sK_DR_DC_D+K_P\right)g_{mc}\right]\left(\frac{sRC+1}{sC}\right)\frac{K_{VCO}}{s} \quad \textbf{(3.17)}$$

where $K_{CP}$, $K_{max}$ and $g_{mc}$ are the gains of charge pump, the filter OTA and the transconductance of the V/I converter respectively.

The corresponding closed loop function is in the form of:

$$H(s)=N\frac{As^2+2\zeta\omega_n s+\omega_n{}^2}{s^2+2\zeta\omega_n s+\omega_n{}^2} \quad \textbf{(3.18)}$$

where the A constant, damping coefficient and the natural frequency respectively are defined as:

$$A=\frac{R\,K_{VCO}\,K_D{}'}{N+R\,K_{VCO}\,K_D{}'} \quad \textbf{(3.19)}$$

$$\omega_n=\sqrt{\frac{K_{VCO}\left(K_{CP}+K_P{}'\right)}{C\left(N+R\,K_{VCO}\,K_D{}'\right)}} \quad \textbf{(3.20)}$$

$$\zeta=\frac{1}{2}\omega_n\left(RC+\frac{K_D{}'}{K_{CP}+K_P{}'}\right) \quad \textbf{(3.21)}$$

where $K_P{}'=K_PK_{\max}R_Fg_{mc}$ and $K_D{}'=K_DR_DC_DK_{\max}R_Fg_{mc}$.

Using (2.62), (3.18) also shows that the noise bandwidth of the proposed PLL structure is infinite, which shows the necessity of the enable/disable function.

Since the main purpose of the proposed PLL structure is to reduce the settling time of the VCO voltage for frequency step inputs, the transfer function from input phase to VCO control voltage is defined in (3.22).

$$V_{VCO}(s)=\frac{\Phi_i(s)H(s)}{\frac{K_{VCO}}{s}}=\Phi_i(s)\frac{N}{K_{VCO}}\frac{s\left(As^2+2\zeta\omega_n s+\omega_n{}^2\right)}{s^2+2\zeta\omega_n s+\omega_n{}^2} \quad \textbf{(3.22)}$$

For a frequency step input, the VCO control voltage becomes:

$$V_{VCO}(s) = NK_{VCO}\frac{\Delta\omega_i}{s}\frac{As^2 + 2\zeta\omega_n s + \omega_n{}^2}{s^2 + 2\zeta\omega_n s + \omega_n{}^2} \quad \textbf{(3.23)}$$

The inverse Laplace transform of (3.23) is:

$$V_{VCO}(t) = \frac{N\Delta\omega_i}{K_{VCO}}\left\{1-(1-A)e^{-\zeta\omega_n t}\left[\cosh\left(\omega_n t\sqrt{\zeta^2-1}\right) - \frac{\zeta}{\sqrt{\zeta^2-1}}\sinh\left(\omega_n t\sqrt{\zeta^2-1}\right)\right]\right\} \quad \textbf{(3.24)}$$

Taking the limit of (3.24) as time goes to infinity gives:

$$\lim_{t\to\infty}\left[V_{VCO}(t)\right] = \frac{N\Delta\omega_i}{K_{VCO}} \quad \textbf{(3.25)}$$

which is an expected result, since (3.25) shows that the final VCO control voltage is simply the output frequency divided by the VCO gain, which also shows that there is no steady state frequency error.

If $\zeta = 1$, (3.24) becomes:

$$V_{VCO}(t) = \frac{N\Delta\omega_i}{K_{VCO}}\left[1-e^{-\omega_n t}(1-A)(\omega_n t+1)\right] \quad \textbf{(3.26)}$$

For $\zeta < 1$, (3.24) can be rearranged as follows:

$$V_{VCO}(t) = \frac{N\Delta\omega_i}{K_{VCO}}\left\{1+e^{-\zeta\omega_n t}\frac{1-A}{\sqrt{1-\zeta^2}}\sin\left[\omega_n t\sqrt{1-\zeta^2} - \arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)\right]\right\} \quad \textbf{(3.27)}$$

Equation (3.27) is a damped oscillation with its envelope defined as:

$$V_{env}(t) = \frac{N\Delta\omega_i}{K_{VCO}}\left(1 \pm e^{-\zeta\omega_n t}\frac{1-A}{\sqrt{1-\zeta^2}}\right) \quad \textbf{(3.28)}$$

Assuming that the settling time of the envelope is approximately the settling time of the damped oscillation, (3.28) becomes:

$$V_{env}(t_s) = V_{VCO}(\infty)(1 \pm S) = \frac{N\Delta\omega_i}{K_{VCO}}\left(1 \pm (1-A)e^{-\zeta\omega_n t_s}\frac{1}{\sqrt{1-\zeta^2}}\right) \quad \textbf{(3.29)}$$

where $V_{VCO}(\infty)$ is the limit value defined in (3.25) and S is the settling window, such that S=0.02 for calculating the settling time to a 2% of the final value.

Solving (3.29) for S gives:

$$S = e^{-\zeta\omega_n t_s}\frac{(1-A)}{\sqrt{1-\zeta^2}} \tag{3.30}$$

Using (3.30), $t_s$ becomes:

$$t_s = -\frac{(1-A)}{\zeta\omega_n}\ln\left(S\sqrt{1-\zeta^2}\right) = \frac{1}{\zeta\omega_n}\ln\left(\frac{1-A}{S\sqrt{1-\zeta^2}}\right) \tag{3.31}$$

As seen in Figure 3.25, real settling time is very close but often smaller than the one calculated with (3.31).

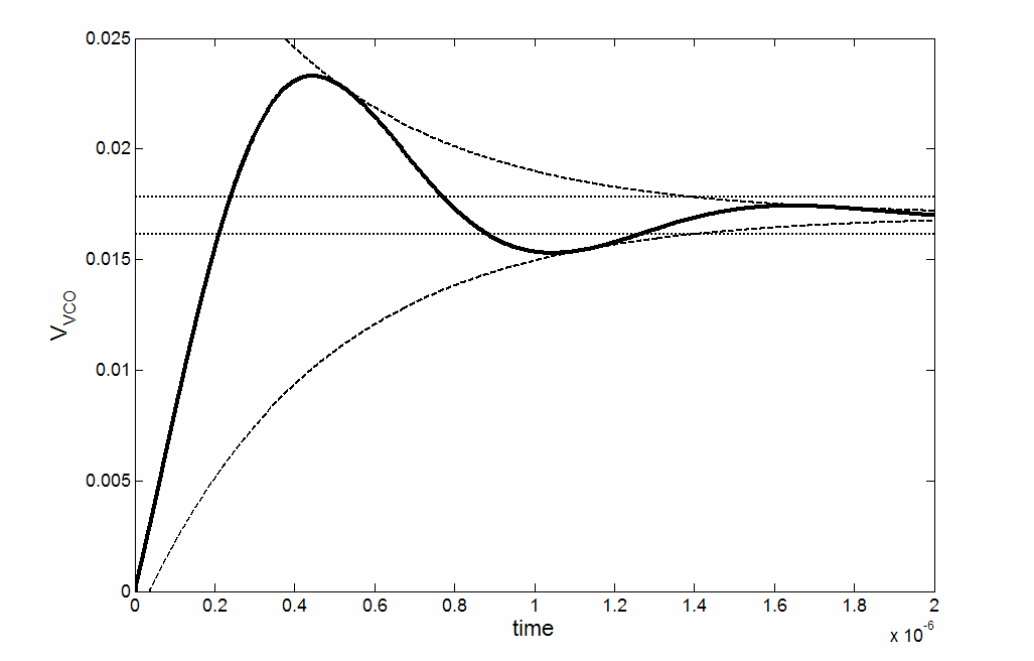


**Figure 3.25:** Settling of the Real Waveform with Envelopes

For $\zeta^2 << 1$, (3.31) simplifies to:

$$t_s = \frac{1}{\zeta\omega_n}\ln\left(\frac{1-A}{S}\right) \tag{3.32}$$

which clearly shows that settling time is inversely proportional to $\zeta\omega_n$. For the proposed charge pump PLL, $\zeta\omega_n$ becomes:

$$\zeta\omega_n = \frac{1}{2C}\frac{RC\left(K_{CP}+K_P'\right)+K_D'}{N+R\,K_{VCO}\,K_D'} \tag{3.33}$$

Equation (3.33) shows that increasing $K_P$ decreases settling time whereas the effect of $K_D$ depends on the value of $K_P$.

Taking the first derivative of (3.27) and doing some math gives:

$$\frac{dV_{VCO}(t)}{d(\omega_n t)} = -\frac{N\Delta\omega_i}{K_{VCO}}e^{-\zeta\omega_n t}\frac{(1-A)}{\sqrt{1-\zeta^2}}\sin\left[\omega_n t\sqrt{1-\zeta^2}-2\arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)\right] \tag{3.34}$$

which becomes zero for:

$$\omega_n t\sqrt{1-\zeta^2}-2\arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)=\pi k \tag{3.35}$$

where k is an integer.

Solving (3.35) for $\omega_n t$ gives:

$$\omega_n t = \frac{1}{\sqrt{1-\zeta^2}}\left[\pi k+2\arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)\right] \tag{3.36}$$

Since the limits of the arctangent function is $(-\pi/2, \pi/2)$, the minimum positive t value satisfying (3.36) occurs for k=0:

$$t = \frac{2}{\omega_n\sqrt{1-\zeta^2}}\arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right) \tag{3.37}$$

which is simply the time of the overshoot. Substituting (3.37) into (3.27) gives:

$$V_{VCO}(t) = \frac{N\Delta\omega_i}{K_{VCO}}\left[1+(1-A)e^{-f(\zeta)}\right] \tag{3.38}$$

where

$$f(\zeta) = \frac{2\zeta}{\sqrt{1-\zeta^2}} \arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right) \tag{3.39}$$

Thus the normalized overshoot is:

$$M_P = (1-A)e^{-f(\zeta)} \tag{3.40}$$

Equation (3.40) shows that the overshoot of the response depends on the damping ratio, which depends on both $K_D$ and $K_P$, and A, which depends on $K_D$.

Graphical representation of (3.40) is shown in Figure 3.26.

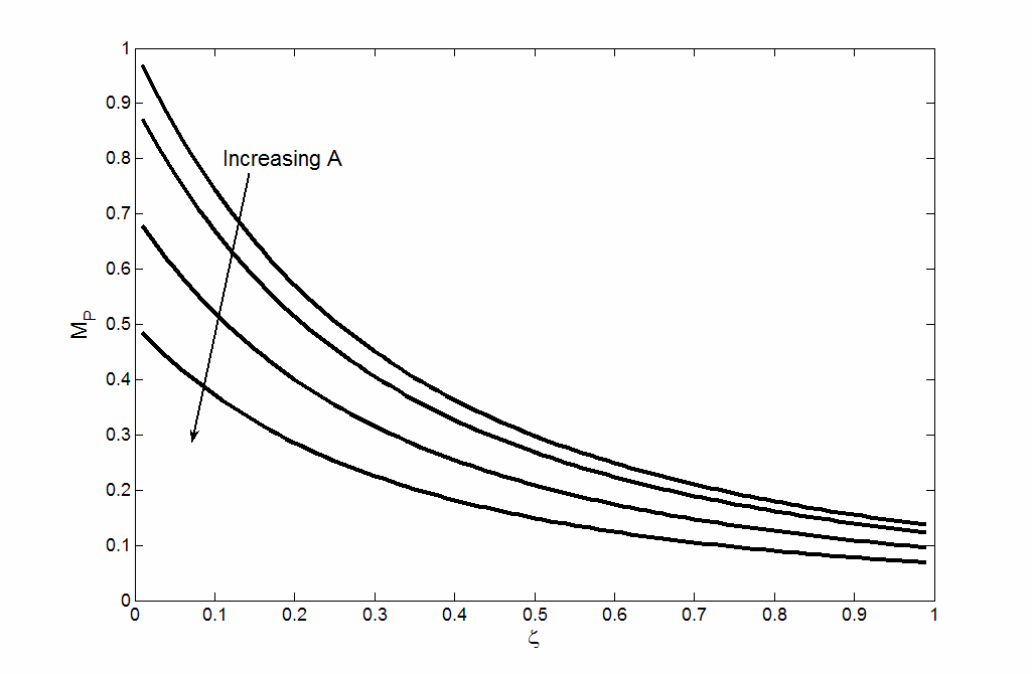


**Figure 3.26:** Normalized Overshoot vs. Damping Ratio

As seen in Figure 3.26, increasing A and increasing damping ratio reduces the overshoot. Thus it can be concluded that increasing $K_D$ decreases overshoot since it increases both A and the damping ratio. Also increasing $K_P$ decreases overshoot by increasing damping ratio. To sum up, increasing $K_D$ decreases overshoot, whereas increasing $K_P$ decreases the settling time and overshoot. As seen in Figure 3.27, increasing $K_P$ decreases both the overshoot and settling time.

In traditional PLL design, increasing R decreases the settling time and overshoot, however it also increases the output ripple. Similarly increasing charge pump

capacitor decreases overshoot, however it also decreases natural frequency which increases the pull-in time.

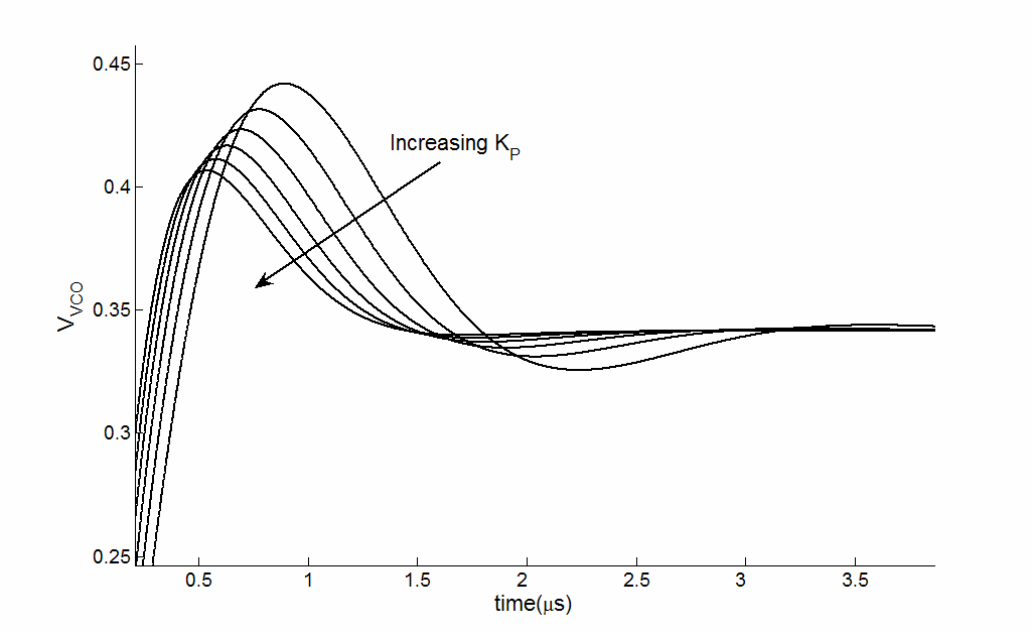


**Figure 3.27:** Effect of $K_P$

As seen in Figure 3.28, increasing $K_D$ decreases the overshoot. Both of the Figure 3.27 and 3.28 show consistent results with the calculations.

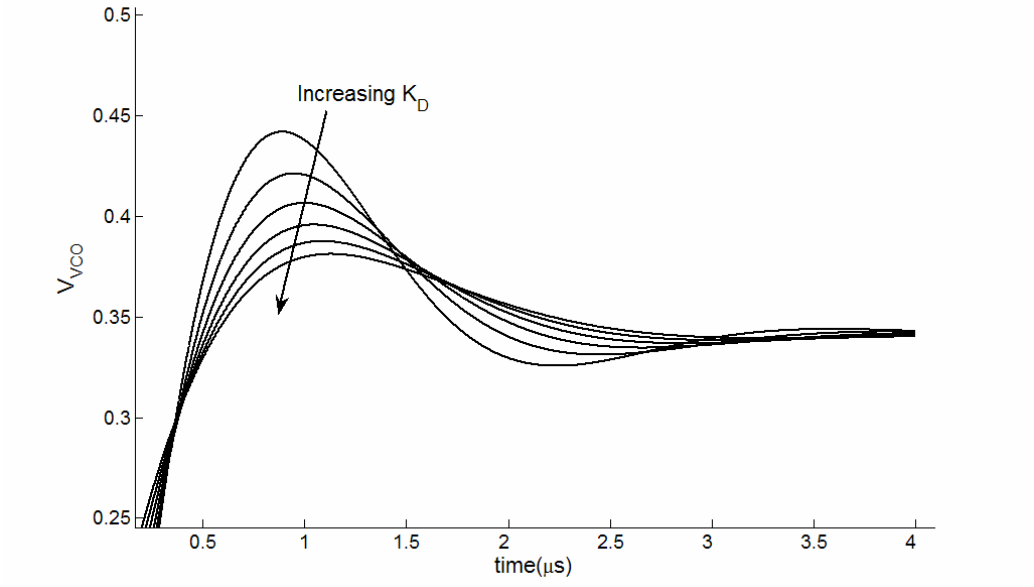


**Figure 3.28:** Effect of $K_D$

## 4. SIMULATION RESULTS

The structure proposed in Chapter 3 is simulated using the VerilogA models of the PFD, charge pump, VCO and divider. The design parameters of the controller are chosen as $I_{max}$=50μA, $C_F$=$C_D$=1pF, $R_F$=$R_D$=25K, $g_{mc}$=200μS and $V_{COMP}$=50mV. Two of previously proposed frequency synthesizer PLLs are examined and enhanced using the proposed controller implemented PLL structure. Both of these frequency synthesizers initially have unity damping ratios. As it was mentioned in Chapter 2, unity damping ratio optimizes the transient response, whereas damping ratio of 0.5 optimizes noise performance. It was also mentioned that decreasing natural frequency decreases noise bandwidth but also increases settling time and pull-in time. Thus both of the designs are configured to have unity damping ratios and increased natural frequencies during the acquisition and 0.5 damping ratios with their natural frequencies restored after phase-locked state.

### 4.1 2.4 GHz CMOS Frequency Synthesizer

The 2.4 GHz CMOS Frequency Synthesizer employs a PLL proposed by Pak [16]. It is designed using the 0.18μm CMOS process of UMC, with a supply voltage of 1.8V and a power consumption of 18mW. The design parameters are $K_{VCO}$=6GHz/V, $I_{CP}$=10μA, $C_1$=28.9pF, $R_2$=12.15K, $C_2$=1.45pF and N=64. Using (3.12) and (3.13), the natural frequency and the damping ratio becomes 5.69 Mrad/s and 1 respectively. Response of the PLL for a frequency step of 5MHz is shown in Figure 4.1. The dotted line and the solid line show the real and the ideal responses respectively.

Although the damping ratio is set to 1 for transient response considerations, in order to reduce the noise, the damping ratio can be set to 0.5 by reducing $R_2$ to 6.075K. The initial noise bandwidth is calculated using (2.64) and equal to 3.55Mrad/s. For the damping ratio of 0.5, the noise bandwidth is reduced to 2.845Mrad/s. Thus, if the system behaves as if the damping ratio is 1 during the acquisition, the optimum transient response would be achieved. Also the natural frequency can be increased during the acquisition so that the settling and pull-in time can be reduced.

Considering the neglected effects such as the finite bandwidth of the derivative block and the 1st order filter, the natural frequency during the acquisition is chosen to be 1.5 times the original one. The controller is disabled after the acquisition, thus the damping ratio returns to 0.5 and the natural frequency reduces to its initial value, yielding an optimum noise performance.

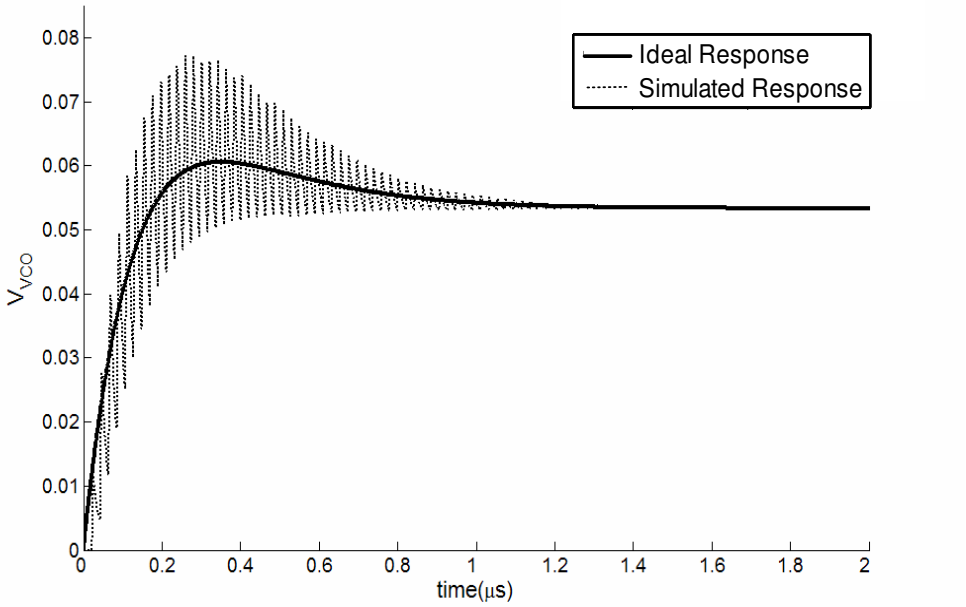


**Figure 4.1:** Transient Response of the First Design Example

Using practical values of $K_P$ and $K_D$ as $K_P$=0.1 and $K_D$=0.1, the damping ratio is calculated as 0.88 and the natural frequency is 9.15Mrad/s. The transient response of the circuit with these values is shown in Figure 4.2. The dashed line shows the step response of the controller implemented PLL and the solid line shows the step response of the PLL without the controller. As seen in the Figure 4.2, the dashed line has a settling behavior as if the damping ratio is unity with increased natural frequency, thus the behavior is scaled in time domain.

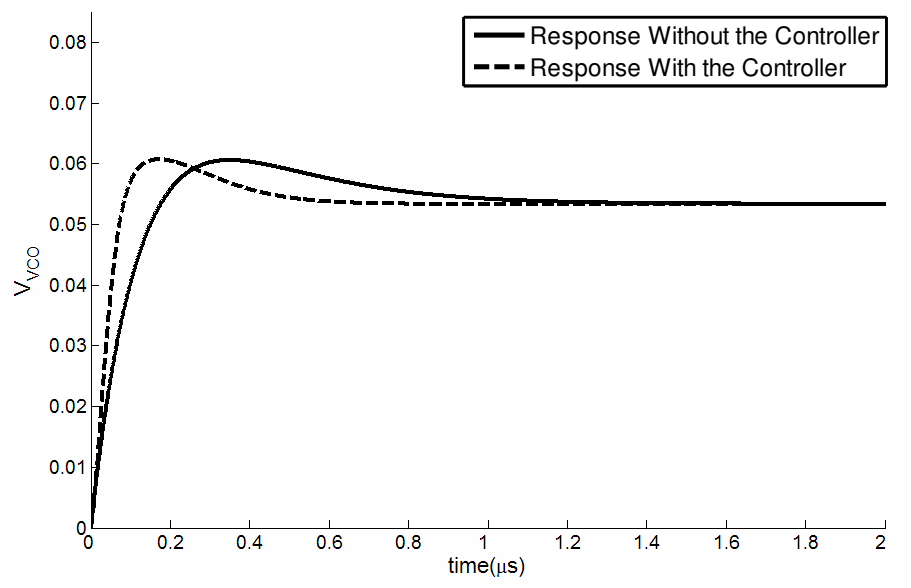


**Figure 4.2:** Ideal Step Responses for the First Design Example with and without the Controller

As shown in Figure 4.3, the step response of the controller implemented PLL shows the same behavior with the formulated step response. The circuit disables itself at around 400ns, making the PLL settle to its final state with the original charge pump and loop filter blocks.

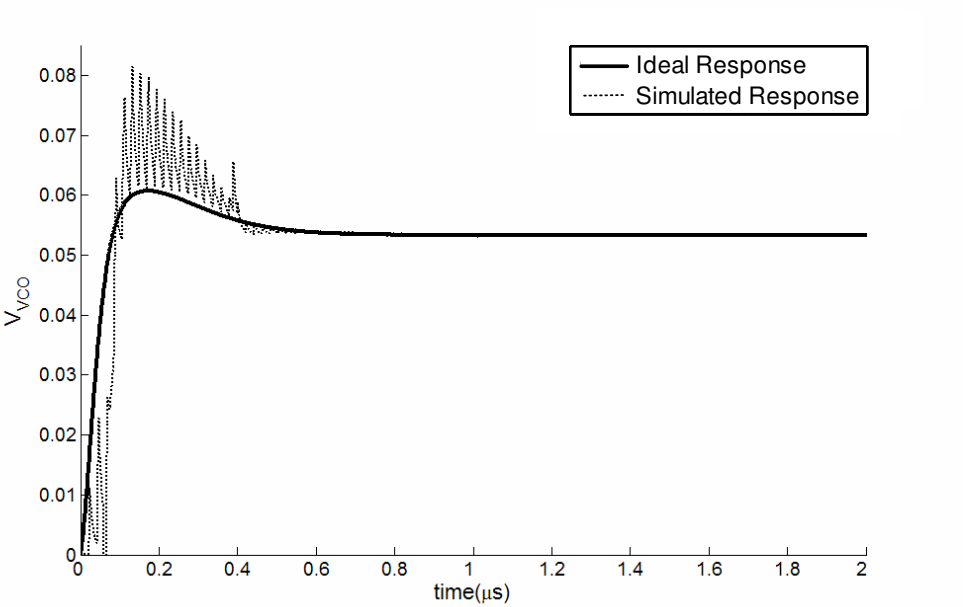


**Figure 4.3:** Transient Response of the First Design Example with MOS Blocks

As it was mentioned, increasing natural frequency decreases pull-in time. Since the natural frequency is increased 1.5 times for the example, the pull-in time is expected to be reduced. Ideally, pull-in time is inversely proportional with $\omega_n^3$. However, as seen on Figure 4.4, although the pull-in time is reduced significantly, since the output current of the OTAs are limited, the simulated pull-in time is not inversely proportional with $\omega_n^3$, and thus it is larger than expected. The darker line shows the pull-in behavior of the original PLL and the lighter line shows the pull-in behavior of the controller implemented PLL.

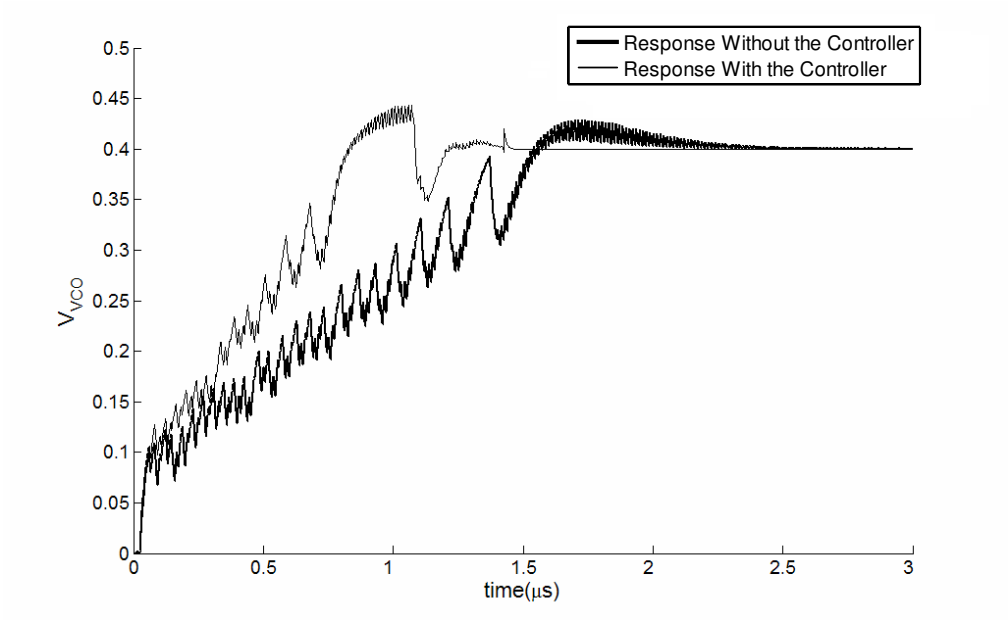


**Figure 4.4:** The Pull-In Behavior of the First Design Example

### 4.2 A 1.5V 2.4GHz PLL for WLAN Applications

Second design example is a PLL with $K_{VCO}$=937.5MHz/V, $I_{CP}$=100μA, $C_1$=100pF, $R_2$=7.5K, $C_2$=7.5pF and N=128 [16]. It is realized with 0.35μm CMOS process of TSMC using 1.5V supply voltage. The damping ratio is calculated as 1.015 and the natural frequency is 2.7 Mrad/s. The noise bandwidth is 1.69Mrad/s. The transient response of the PLL for a 5 MHz step in frequency is shown in Figure 4.5, where the ideal response and the simulation results are drawn in solid line and dotted line respectively.

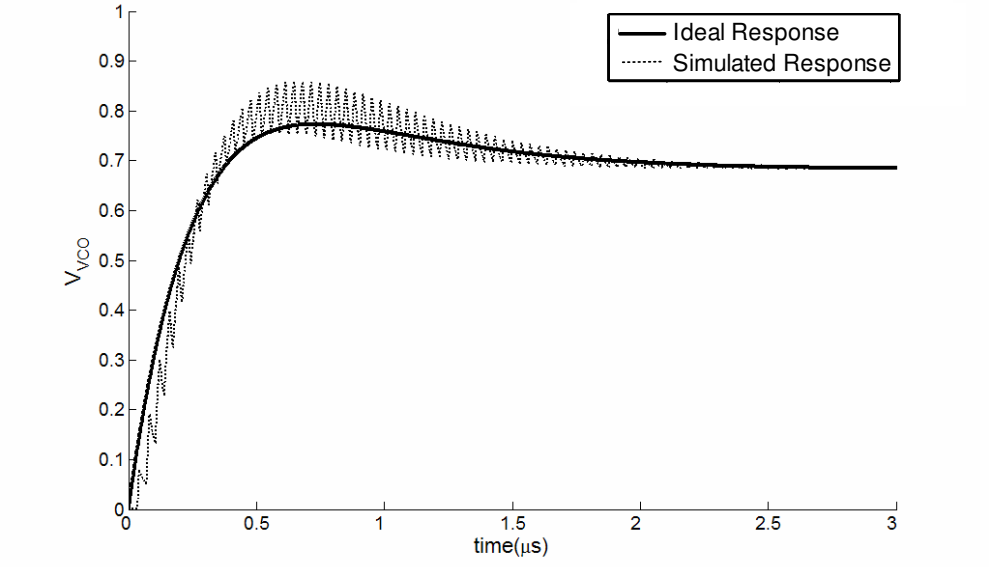


**Figure 4.5:** Transient Response of the Second Design Example

Using the same approach stated in Section 4.1, the loop filter resistor is halved in order to achieve optimum noise bandwidth, which is equal to 1.35Mrad/s. The damping ratio is set to 1 and the natural frequency is set to 1.5 times the original value during the acquisition. For practical values of $K_P$ and $K_D$ as $K_P=K_D=1$, the natural frequency and the damping ratio becomes 5.06Mrad/s and 0.95 respectively. Expected results for the new natural frequency and damping ratio are shown in dashed line in Figure 4.6, where the original settling behavior is shown in solid line.

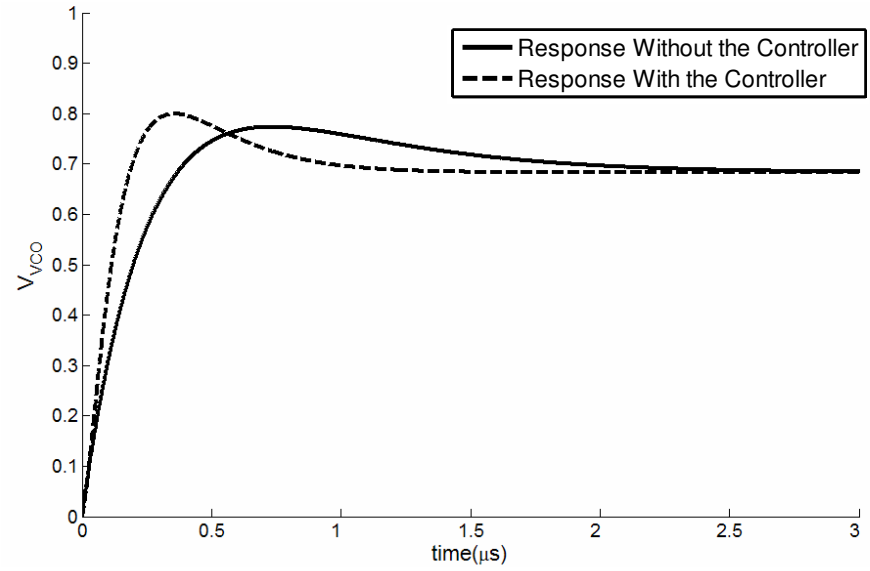


**Figure 4.6:** Ideal Step Responses for the Second Design Example with and without the Controller

Using the controller coefficients as $K_P=K_D=1$, the simulated step response of the proposed PLL structure is shown in Figure 4.7. As seen on Figure 4.7, the behavior of the designed system, which is shown in dotted line, shows great resemblance with the expected results, which is shown in solid line.

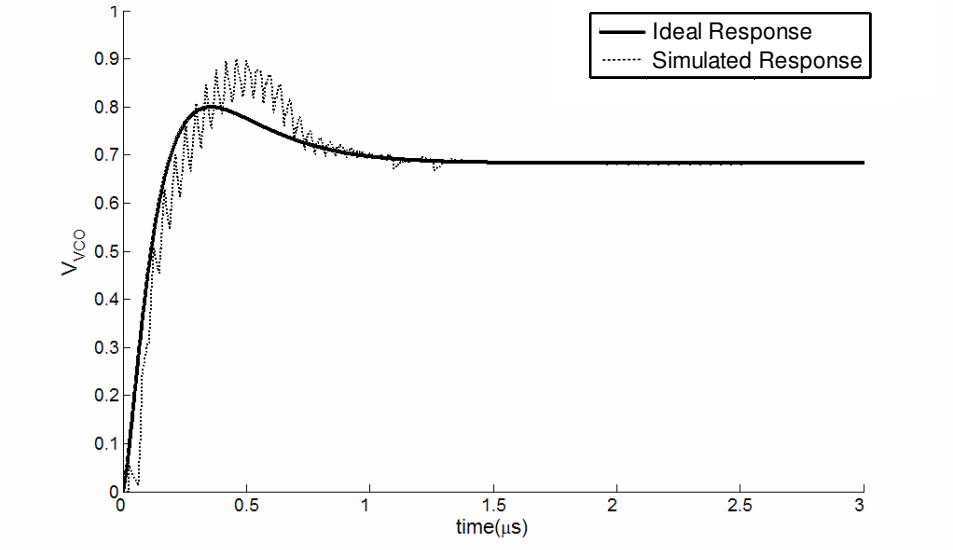


**Figure 4.7:** The Transient Response of the Second Design Example with MOS Blocks

Similar to the first design example, the effect of the increasing natural frequency on the pull-in behavior is shown in Figure 4.8. The darker line shows the pull-in behavior of the original PLL and the lighter line shows the pull-in behavior of the controller implemented PLL.

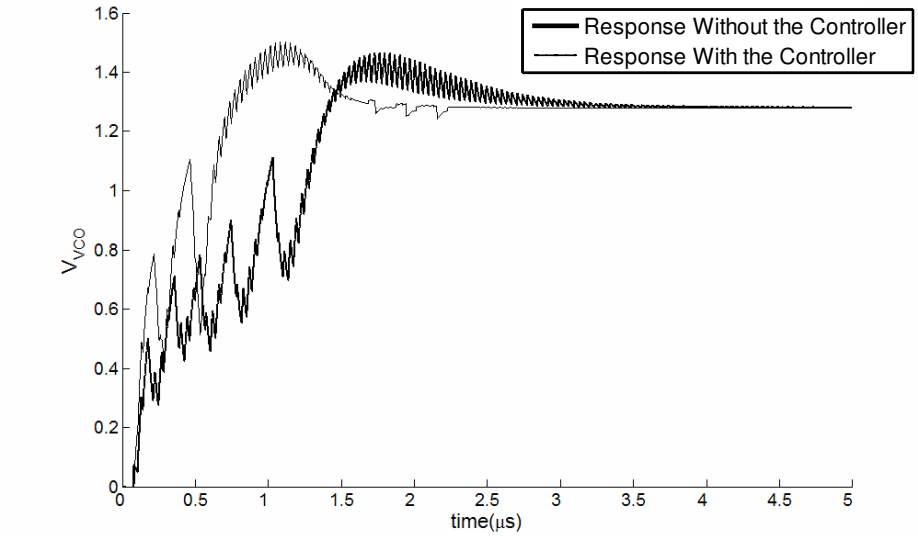


**Figure 4.8:** The Pull-In Behavior of the Second Design Example

### 4.3 Overall Results

The proposed PLL structure is simulated with the VerilogA models of the PLL blocks and the controller designed with 0.35μm CMOS process of AMS. Two of the previously proposed frequency synthesizer PLLs with unity damping ratios are enhanced setting the damping ratios to 0.5 after the phase locked state, thus providing and optimum noise bandwidth, but still making the system settle with unity damping ratios and increased natural frequencies. The simulations show that the transient responses are improved for both pull-in time and settling time. The summary of the simulation results are shown in Table 4.1.

**Table 4.1:** Comparison of the design examples with and without the controller

| | **Settling Time (1%)** | | **Pull-In Time (1%)** | | **Noise Bandwidth** | |
|---|---|---|---|---|---|---|
| | **Original** | **Proposed** | **Original** | **Proposed** | **Original** | **Proposed** |
| **1st Design Example** | 1.29μs | 0.61μs | 2.33μs | 1.44μs | 3.55Mrad/s | 2.85Mrad/s |
| **2nd Design Example** | 2.54μs | 1.37μs | 3.35μs | 2.21μs | 1.69Mrad/s | 1.35Mrad/s |

## 5. CONCLUSION

The goal of this thesis is to implement a controller into a PLL. PLLs, being a feedback control loop, lacks a decent controller, thus the design of a PLL is a trade-off between transient performance and noise. For a traditional control system, unity damping ratio is considered to satisfy the optimum transient response, however the PLL's noise bandwidth reaches to its minimum for a damping ratio of 0.5. Similarly increasing the natural frequency decreases the pull-in time but it also increases the noise. Using a controller enables to change the loop parameters during acquisition and restore afterwards, thus making the loop respond quicker without disturbing its noise. Consequently, the controller can remove the trade-off between the transient response and noise.

The simplest controller in control theory is the PID controller, and it is the controller employed in this work. Theoretically any type of controller can be implemented.

The PID controller simply adds the derivative of the error, the integral of the error and the amount of the error itself, each multiplied with coefficients. Since the PLL is a loop based on integrators, additional integral control is not necessary to implement. Considering this property, a new structure of a charge pump PLL with a PD controller is proposed.

The proposed PLL structure is analyzed in the s domain and the theoretical transient responses are formulated. Formulations show that the natural frequency and the damping ratio of the system can be tuned without changing the initial loop parameters.

Using the VerilogA models of the PFD, VCO, divider and the charge pump, the proposed PLL structure is simulated. The simulations show consistent results with the calculations.

Future work can be summarized as investigation of various controller types that do not consume much area and power, and can also produce better results than the PD controller.

## REFERENCES


[1] **Kroupa, V. F.,** 2003. Phase Lock Loops and Frequency Synthesis, John Wiley and Sons, Inc.

[2] **Gardner, F.,** 1979. Phaselock Techniques, John Wiley and Sons, Inc.

[3] **Razavi, B.,** 1996. Monolithic Phase-Locked Loops and Clock Recovery Circuits, IEEE Press.

[4] **Stensby, J.,** 1997. Phase-Locked Loops Theory and Applications, CRC Press LLC.

[5] **Brennan, P.,** 1996. Phase-Locked Loops Principles and Practice, Donnelley and Sons.

[6] **Best, R. E.,** 1984. Phase-Locked Loops Theory, Design, and Applications, McGraw Hill Book Company

[7] **Banerjee, D.,** 1998. PLL Performance, Simulation, and Design, National Semiconductor Design Notes

[8] **Kroupa, V. F.,** 1982. Noise Properties of PLL Systems, *IEEE Trans. Comm.,* **30,** pp. 2244-2252.

[9] **Sharpe, C. A.,** 1976. A 3-State Phase Detector Can Improve Your Next PLL Design, *EDN Magazine*

[10] **Gardner, F. M.,** 1980. Charge-Pump Phase Lock Loops, *IEEE Trans. Comm.,* **28,** pp. 1849-1858.

[11] **Pak, B.,** 2007. Personal talk.

[12] **Aksın, D. Y.,** 2007. Personal talk.

[13] **Kuntman, H.,** 1997. Analog MOS Tümdevre Tekniği, İstanbul Teknik Üniversitesi Rektörlüğü, İstanbul.

[14] **Nedungadi, A. and Viswanathan, T. R.,** 1984. Design of Linear CMOS Transconductance Elements, *IEEE Trans. Circuits Syst.,* **vol CAS31,** pp. 891-894.

[15] **Razavi, B.,** 1995. Principles of Data Conversion System Design, IEEE Press.

[16] **Pak, B.,** 2002. 2.4GHz CMOS PLL Frequency Synthesizer, *MS Thesis,* ITU Institute of Science and Technology, Istanbul

[17] Build a 1.5V 2.4GHz CMOS PLL, http://www.wirelessnetdesignline.com/howto/180205535

## BIOGRAPHY

Hayri Uğur UYANIK was born in Istanbul, Turkey in 1984. He graduated from Beşiktaş Atatürk Anatolian High School in 2001. In 2005 he received B.Sc. degree in Electronics Engineering from Istanbul Technical University, where he started the M.Sc. program of the same major. He has been working as a research and teaching assistant in the Electronics and Communication Engineering Department of Istanbul Technical University since December 2005. His research interests include modeling, analysis and synthesis of microelectronic circuits.