

Lehigh University Lehigh Preserve

Theses and Dissertations

2012

PLL FM Demodulator with Synchronous Filter

Shaohui Huang
Lehigh University

Follow this and additional works at: <http://preserve.lehigh.edu/etd>

Recommended Citation

Huang, Shaohui, "PLL FM Demodulator with Synchronous Filter" (2012). *Theses and Dissertations*. Paper 1285.

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

PLL FM Demodulator with Synchronous Filter

by

Shaohui Huang

A Thesis

Presented to the Graduate and Research Committee

Of Lehigh University

In Candidacy for the Degree of

Master of Science

in

Department of Electrical and Computer Engineering

Lehigh University

December 2011

This thesis is accepted and approved in partial fulfillment of the requirements for the
Master of Science.

Date

Thesis advisor
Prof. Douglas R Frey

Chairperson of Department
Prof. Filbert Bartoli

Acknowledgment

As with any endeavor, this research project could not have been completed without the help and support of others. Foremost, I wish to thank my advisor Prof. Douglas R. Frey. His guidance throughout the course of this project is greatly appreciated. I also wish to thank my family for their support and encouragement. I dedicate this work to all of you!

Table of Contents

Chapter1	:	PLL FM Demodulator Introduction	2
1.1	:	PLL Communication Applications and Basic PLL Block Diagram	2
1.2	:	Introduction of PLL FM Demodulator and Building Block	3
Chapter 2	:	Description of the PLL Block	7
2.1	:	Phase Detector	7
2.1.1	:	Phase Detector Types	7
2.1.2	:	Phase Frequency Detector with Voltage Output	12
2.2	:	Low Pass Filter (LPF)	14
2.2.1	:	First Order LPF and MATLAB Code	14
2.2.2	:	Second Order Butterworth Low Pass Filter	16
2.3	:	Loop Filter	17
2.3.1	:	Loop Filter Types and Order of the PLL	17
2.3.2	:	MATLAB Code of the Loop Filter	18
2.4	:	Voltage control Oscillator (VCO)	19

2.4.1	:	VCO General Theory	19
2.4.2	:	VCO Architecture and Equation	20
2.4.3	:	MATLAB Code of VCO1 and VCO2	21
Chapter 3	:	System Level Description of the PLL	23
3.1	:	Operation of the Three Basic Functional Blocks	23
3.2	:	PLL Transfer Function and MATLAB Simulation Result	25
3.3	:	Loop Parameters	28
Chapter 4	:	How This Research Adds To Basic PLL	31
		-Optimum Time Varying Filter	
4.1	:	Optimum Filter and Optimum Time Varying Filter	
		with Band Pass Filter	31
4.2	:	Synchronous filtering idea and Mathematical framework	32
4.3	:	Optimum FM signal filtering	36
4.3.1	:	Theory	36
4.3.2	:	Optimum time varying FM signal BPF of this project	37
Chapter 5	:	Simulation Result	40

Chapter 6	: Research Conclusion and Possible Future Work	46
Reference		48
Appendix		49
A	: PLL Simulation MATLAB Code	49
A1	: PLL (with multiplier Phase detector) simulation MATLAB code	49
A2	: PLL (with Phase frequency detector) simulation MATLAB code	51
B	: BPF Simulation MATLAB Code	54
C	: Whole Research (PLL FM Demodulator) Simulation	
	MATLAB Code	55
Vita		61

Abstract of the Thesis

Phase locked demodulators are widely used for reception of amplitude modulation(AM), phase modulation(PM), and frequency modulation(FM). In this research, we concentrated on phase locked FM demodulators. Our main two parts are the PLL block and the Optimum Time Varying Filter block. We started our work by building up three basic blocks of the PLL; the PFD, the Loop Filter and the VCO. By carefully studying the structure and the operation of these blocks, we were able to make good choices regarding the types of each block and their parameters. As a result, we got the desired results in our PLL simulations. After that, in order to improve the quality of the FM signal, we worked on the Optimum Time Varying Filter which is the main research part in the overall thesis. We studied the relationship of the optimum filter (or matched filter), the optimum time-varying filter and our band pass filter. We then derived the necessary equations to realize our synchronous filter. Based on the theory and equation discussion, we were able to design the band pass filter. After we put all of the blocks together to realize our FM PLL demodulator, we simulated the whole demodulator and recovered the FM demodulated signal which showed that we had achieved our research goal-namely, to improve the quality of the PLL FM Demodulator.

Chapter 1: PLL FM Demodulator Introduction

1.1: PLL Communication Applications and Basic PLL Block Diagram

The Phase-Locked Loop (PLL) has become an essential component in wireless communication systems. To begin with, analog or digital data can be transmitted using a PLL over a data link. Historically, the PLL was developed to be used in the analog domain. The inventor Henri de Bellescize designed a Vacuum tube-based synchronous demodulator for an AM receiver.

Phase-Locked loops are used for demodulation of many kinds of modulated signals. Applications include coherent amplitude detectors (product detectors), phase demodulators (PM detectors), and frequency demodulators (FM discriminators).

A PLL is a circuit that causes a system to track with another one. More precisely, a PLL is a circuit to synchronize an output signal (generated by VCO) with an input signal in frequency as well as in phase. In the “locked” state, the phase error between the VCO output signal and the input signal is zero, or it remains constant. A small phase error can often be tolerated. A locked PLL is also said to track the phase information at its input signal. A basic PLL Block Diagram is shown in figure 1.1.

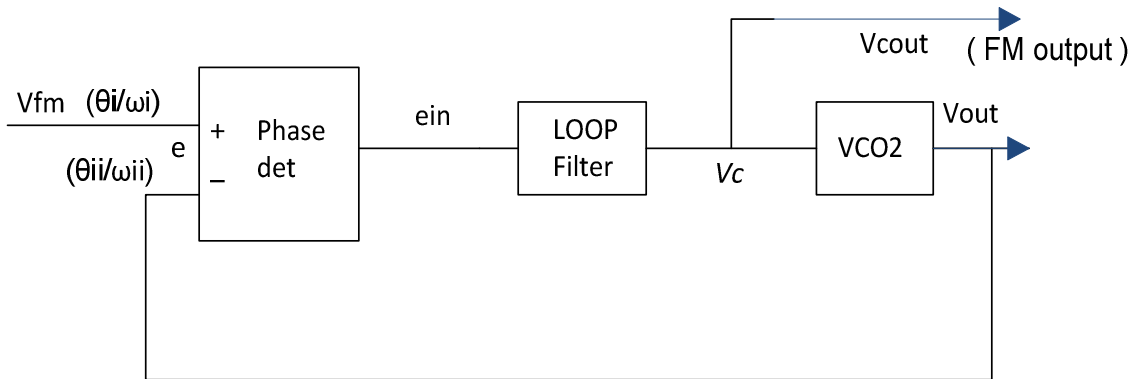


Figure 1.1 Phase Locked Loop (PLL) Block

A Phase locked loop (PLL) contains three basic elements (figure 1.1). (1) a phase detector(PD),(2) a loop filter(LF), and (3) a voltage controlled oscillator(VCO). A Phase detector compares the phase of an input signal with the phase of the VCO output signal; the output of the PD is a measure of the phase error between its two inputs. The error voltage (e_{in}) is then filtered by the loop filter, whose output (a control voltage) is applied to the VCO. The control voltage changes the VCO frequency in a way that reduces the phase error between the input signal and the VCO.

In this way, the three basic PLL blocks work together. This thesis describes research aimed at improving a PLL FM demodulator.

1.2: Introduction of PLL FM Demodulator and Building Blocks

Suppose that a frequency-modulated input signal is applied to a PLL. For the loop to remain in a lock, it is necessary that the frequency of the VCO track the incoming frequency very closely. The frequency of the VCO is proportional to the control voltage

V_c , so the control voltage must be a close replica of the original modulation on the signal. Modulation may therefore be recovered from the VCO control voltage. This is the principle of the PLL FM demodulator (PLD). The PLD is a modulation-tracking loop. Consider the following definition where θ_i is the input phase and θ_{ii} is the phase output of the PLL.

$$\text{PLL closed loop transfer function: } H(s) = \frac{\theta_{ii}(s)}{\theta_i(s)}$$

$$\text{The instantaneous frequency modulation } m(t) \text{ in rad/sec, } m(t) = \frac{d\theta_i(t)}{dt}$$

$$\text{Taking the Laplace transform: } M(s) = S\theta_i(s)$$

$$\text{VCO transfer function: } H_{vco}(s) = \frac{\theta_{ii}(s)}{V_c(s)} = \frac{K_{vco}}{S}$$

$$V_c(s) = \frac{S\theta_{ii}(s)}{K_{vco}} = \frac{SH(s)\theta_i(s)}{K_{vco}} = \frac{M(s)H(s)}{K_{vco}}$$

This shows the transfer function between the original frequency modulation 'm(t)' and the resulting VCO control voltage 'V_c(t)'. The message recovered is equivalent to the original message, filtered by the closed-loop transfer function H(s) and divided by the VCO gain factor K_{vco}. V_c(t) is a reproduction of m(t). That is why we can demodulate our FM signal from V_c(t). Using this basic idea, we pursued our research goal which is to improve the quality of this PLL FM Demodulator.

Everyone knows that if one wants to design a system, the system structure is very important. So, before discussing any details, we need to construct the "Phase

locked FM Demodulator (PLD) Building Block”, which is the basis of the research discussed in this thesis.

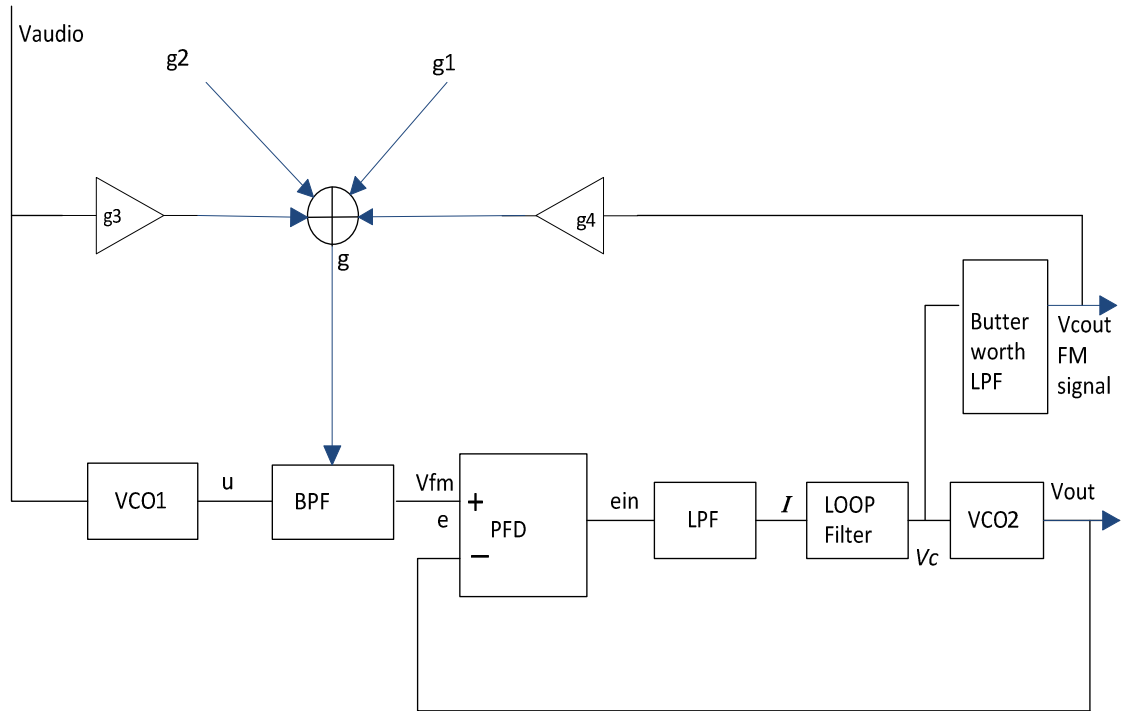


Figure 1.2 PLL FM Demodulator with BPF Building Block

A conceptual block diagram of a PLD in this project is shown in figure1.2. It consists of seven blocks which include voltage control oscillator (VCO1), VCO2, band pass filter (BPF), phase detector (PD), low pass filter (LPF), loop filter (LF) and second order Butterworth low pass filter. VCO1 and BPF (Part 1) forms high frequency FM signal (v_{fm}). PD, LPF, Loop Filter and VCO2 (Part 2) is Phase Lock Loop. The VCO1 block is for producing the FM signal. The Band Pass Filter (BPF) block is the new block we have researched to get a high quality input signal for the PLL. PD, LP (or with LPF) and VCO blocks that make up the PLL.

Comparing to other RF front-end circuits, the PLD is a far more complicated system, and a combination of many individual small blocks, especially the Optimum Time Varying Filter (Band Pass Filter) block. In the following chapters more details of each block will be given.

During the research part of this thesis, the Optimum Time Varying Filter (Band Pass Filter) block received the most attention of any of the seven blocks. In the following chapters, we begin by discussing the PLL which includes the PD, LPF, Loop Filter and VCO blocks, and then later the band pass filter block.

Chapter 2: Description of the PLL Block

This chapter discusses each of the building blocks in a conventional PLL. Different realizations are discussed along with their basic operation.

2.1: Phase Detector

2.1.1: Phase Detector Types

In mixed signal PLLs, there are two broad classes of phase detectors: multiplier (or combinatorial) devices and sequential devices. It is useful to see how each type of Phase detector works in a PLL; then we can make a choice of which type of detector is better in this design.

Type 1: Combinatorial devices – Multiplier

Input to PD: the multiplier type phase detectors has two inputs, denoted $u_i(t)$ and $u_o(t)$, where the input signal $u_i(t)$ is typically a sine wave given by

$$u_i(t) = U_s \sin(\omega_{it} + \theta_i)$$

The second input signal $u_o(t)$ is feedback from the VCO

of the PLL, and is usually a symmetrical square wave signal having the form

$$u_o(t) = U_o \text{rect}(\omega_{oit} + \theta_o)$$

Note that the Fourier series of the square wave signal is:

$$u_o(t) = U_o \left[\frac{4}{\pi} \cos(\omega_{oit} + \theta_o) + \frac{4}{3\pi} \cos(3\omega_{oit} + \theta_o) + \dots \right]$$

We get $u_d(t)$, the output of the multiplier phase detector by multiplying two input signals $u_i(t)$ and $u_o(t)$ of this PD block.

$$u_d(t) = u_i(t) * u_o(t)$$

$$= U_i U_o \sin(\omega_i t + \theta_i) \left[\frac{4}{\pi} \cos(\omega_o t + \theta_o) + \frac{4}{3\pi} \cos(3\omega_o t + 3\theta_o) \dots \right]$$

Let us analysis these phase-detector products. First when the PLL is locked, the frequencies ω_i and ω_o are identical, and $u_d(t)$ becomes:

$$u_d(t) = U_i U_o \left[\frac{2}{\pi} \sin \theta_e \dots \right], \text{ where } \theta_e = \theta_i - \theta_o \text{ is the phase error.}$$

The first term of this series is the desired "DC" term. The higher-frequency terms will be mostly eliminated by the loop filter. So we get $u_d(t) \approx K_d \sin(\theta_e)$, where

detector gain $K_d = 2U_i U_o / \pi$ with K_d having dimensions of V/rad, when θ_e is small, we get $u_d(t) \approx K_d \theta_e$. Secondly, when the PLL is out of lock, the radian frequencies ω_i and

ω_o are different. The output signal of the multiplier can be written as

$$u_d(t) = K_d \sin(\omega_i t - \omega_o t + \theta_i - \theta_o) + \text{higher harmonics}$$

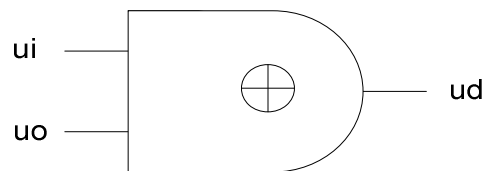
The higher harmonics are attenuated by the loop filter; however, the difference frequency of the AC term is the difference $\omega_i - \omega_o$. Because the output u_d is an AC signal, its average is zero. This means that the average output signal of the loop filter would also be zero. This would make it impossible for the loop to acquire lock, because the frequency of the VCO signal would remain permanently hung up at its free running frequency ω_{fr} , with a superimposed frequency modulation. However, the AC signal $u_d(t)$ is actually an asymmetric 'sine wave' – that is, the durations of the positive and negative

half waves are different. This can be seen by looking at the harmonics. So, there will be a nonzero DC component that will pull the average output frequency of the VCO up or down, until lock is acquired,

$$\omega_i - \omega_{ii} < \Delta\omega_p \quad \Delta\omega_p : \text{ pull-in range}$$

The pull-in process is quite slow.

Type 2: Sequential devices - EXOR phase detector



$$u_d = u_i \oplus u_o$$

$$\theta_e = \theta_i - \theta_o$$

The operation of the EXOR phase detector is similar to that of the linear multiplier. The signals in DPLLs are always binary signals. We assume for the moment that both signals u_i and u_o are symmetrical square waves. Let's discuss different phase errors θ_e .

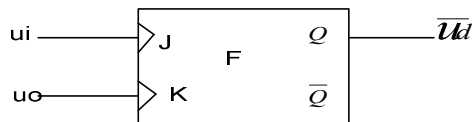
First, at zero phase error, the signals u_i and u_o are out of phase by exactly 90° . Then the output signal u_d is square wave whose frequency is twice the reference frequency; the duty cycle of u_d signal is exactly 50 percent. Because the high-frequency component of this signal will be filtered out by loop filter, we consider only the average

value of u_d . The average value $\overline{u_d}$ is the arithmetic mean of the two logical levels; it will be $\overline{u_d} = 0$.

When the output signal 'u_o' lags the reference signal u_i , the phase error θ_e becomes positive. Now the duty cycle of u_d becomes larger than 50 percent – in another words, the average of u_d is considered positive. Clearly, the mean of u_d reaches its maximum value for a phase error of $\theta_e = 90^\circ$ and its minimum value for $\theta_e = -90^\circ$. Within a phase error range of $-90^\circ < \theta_e < 90^\circ$, $\overline{u_d}$ is exactly proportional to ω_e and can be written as $\overline{u_d} = K_d \theta_e$. The EXOR phase detector can maintain phase tracking when the phase error θ_e is $-90^\circ < \theta_e < 90^\circ$.

When the EXOR phase detector in the unlocked state of the PLL, $\omega_i - \omega_{ii} \neq 0$. The output signal of the EXOR then contains an AC term whose fundamental radian frequency is the difference $\omega_i - \omega_{ii}$. The higher harmonics of which will be filtered out by the loop filter. When $\omega_i - \omega_{ii} < \Delta \omega_p$, the acquisition is realized. And the pull-in process is slow. And it is similar to that described above.

Type 3: Sequential devices – JK-flipflop phase detector



$$Q^* = J * \overline{Q} + \overline{K} * Q$$

This type of JK-flipflop is edge-triggered. A positive edge appearing at the J input triggers the flipflop into its “high” state $Q = 1$, and a positive edge at the K input triggers

the flipflop into its “low” state $Q = 0$. Let’s discuss the phase error “ Θ_e ”. First, with no phase error ($\Theta_e = 0$), u_i and u_o have opposite phase. The output signal u_d then represents a symmetrical square wave whose frequency is identical with the reference frequency. The condition is considered as $\overline{u_d}$ being zero. Secondly, if the phase error becomes positive, the duty cycle of the u_d signal becomes greater than 50 percent, in another words, $\overline{u_d}$ becomes positive. Clearly, when the phase error reaches 180° , $\overline{u_d}$ becomes maximum and when the phase error is -180° , $\overline{u_d}$ is minimum.

$$-180^\circ < \Theta_e < 180^\circ, \quad \overline{u_d} = K_d \Theta_e$$

When $-180^\circ < \Theta_e < 180^\circ$, the JK-flipflop phase detector can maintain phase tracking. In the unlocked state, $\omega_i - \omega_{ii} \neq 0$.

As $\omega_i - \omega_{ii} < \Delta \omega_p$, the PLL can realize the locked state

Type 4: Sequential devices-phase/frequency detector (PFD)

There is another type of phase detector (phase-frequency detector) that enables much faster acquisition, because its output signal is not only phase sensitive, but also frequency sensitive (in the unlocked state). For the reason we chose this type of PFD in our PLL.

PFD’s output signal does not only depend on Θ_e (phase error), but also on a $\Delta\omega = \omega_i - \omega_{ii}$ (frequency error). The PFD can tell whether the ω_i of the input signal $u_i(t)$ is higher or lower than the ω_{ii} of the output signal $u_o(t)$. As a result it allows the PLL to get locked in the most adverse situation – that is, for arbitrarily large frequency offsets between the two input signals.

Have being considered all these types of phase detector. The phase frequency detector becomes the first choice in this PLL FM demodulator.

2.1.2: Phase frequency detector with voltage output

(1): PFD configuration and state diagram

A basic PFD consists of a pair of D flip-flop plus an AND gate as shown in figure

2.1.2a. One output of D-flipflop is "UP", another is "DN".

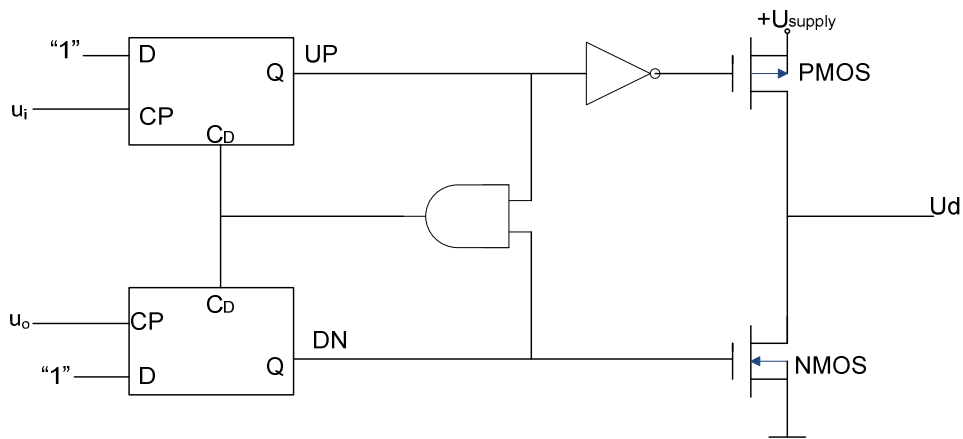


Figure 2.1.2a Phase-Frequency Detector

The PDF acts as a tristable device, and can be in three states: -1, 0, +1.

DN = 1, UP = 0 : state = -1

UP = 0, DN = 0 : state = 0

DN = 0, UP = 1 : state = 1

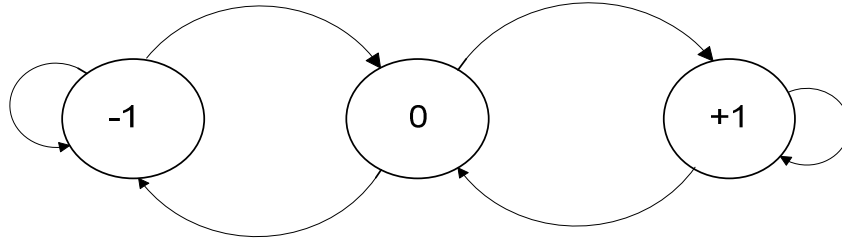


Figure 2.1.1b State Diagram for a PFD

The figure 2.1.2a shows how the u_d signal is generated. When the UP signal is high, the P-channel MOS transistor conducts, so u_d equals the positive voltage U_{supply} . When the DN signal is high, the N-channel MOS transistor conducts, so u_d is at ground potential. If neither signal is high, both MOS transistors are off, and the output signal floats – in other words, it is in the high-impedance state. So, the output signal u_d represents a tristate signal.

(2): PFD Operation in PLL

To see how the PFD works in a real PLL system, we start by assuming that the phase error is zero. It is also assumed that the PFD has been in the 0 state initially. The signals u_i and u_o are “exactly” in phase here; both positive edges of u_i and u_o occur “at the same time”; hence, their effects will cancel. The PFD then will stay in the 0 state forever. When u_i leads, the PFD now toggles between the state 0 and 1. If u_i lags, the PFD toggles between state -1 and 0. It is very clear that u_d becomes largest when the phase error is positive and approaches 360° , and smallest when the phase error is negative and approaches -360° . When the phase error θ_e exceeds 2π , the PFD behaves

as if the phase error recycled at zero; hence, the PFD period is 2π . When the phase error is restricted to the range $-2\pi < \theta_e < 2\pi$, the average signal $\overline{u_d}$ becomes: $\overline{u_d} = K_d \theta_e$.

In analogy to the JK-flipflop, phase detector gain is computed by $K_d = \frac{U_{sup}}{4\pi}$. So, it is a phase detector.

To recognize the bonus offered by the PFD, we assume the PLL is unlocked initially. Then we make the assumption that the reference frequency ω_i higher than the output frequency ω_{ii} . The u_i signal then generates more positive transitions per unit of time than the signal u_o . The PFD can toggle only between the state 0 and 1 under this condition but will never go into the -1 state. If ω_i is much higher than ω_{ii} , the PFD will be in the 1 state most of the time. When ω_i is smaller than ω_{ii} , the PFD will toggle between the -1 and 0. When ω_i is much lower than ω_{ii} , the PFD will be in the -1 state most of the time. We conclude that the average output signal u_d of the PFD varies monotonically with the frequency error $\Delta\omega = \omega_i - \omega_{ii}$ when the PLL is out of lock. It is the record a frequency detector. Because $\overline{u_d}$ of the PFD depends on the phase error in the locked state of the PLL and on the frequency error in the unlocked state, a PLL which uses the PDF will lock under any condition.

2.2: Low Pass Filter

2.2.1: First order LPF and MATLAB Code

A filter is a circuit that processes signals on a frequency-dependent basis. The frequency response is expressed in terms of the transfer function $H(j\omega)$, where $\omega = 2\pi f$. The low-pass response is really characterized by a cutoff frequency ω_c , such that

$|H| = 1$ for $\omega < \omega_c$ and $|H| = 0$ for $\omega > \omega_c$, indicating that the input signals with frequency less than ω_c go through the filter with unchanged amplitude while signals with $\omega > \omega_c$ undergo complete attenuation.

In this PLL, a first order low pass filter is used for the removal of high-frequency noise from the output signal (e_{in}) of the PFD. Because the frequency of V_{audio} is $\omega_{audio} = 2\pi \times 3$, the LPF's cutoff frequency ω_c needs to be higher than ω_{audio} . In figure 1.2, we put our LPF after the PD block. Following the analysis in chapter 2.1.1, the output signal $u_d(t)$ of the phase detector consists of a number of terms. In the locked state of the PLL, the first DC component is roughly proportional to the phase error θ_e ; the remaining terms are AC components having frequencies of $2\omega_i, 4\omega_i, \dots$. These higher frequencies are unwanted signals. They are filtered out by the low pass filter block and the loop filter block. So we choose $\omega_c = 2\pi \times 2e4$

In here, we need to have the LPF transfer function which will be shown in the MATLAB simulation later.

The transfer function is:
$$H(s) = \frac{I_1}{e_{in}} = \frac{\omega_c}{s + \omega_c}$$

Apply "Forward Euler" to transfer this function from "S" field to "Z" field:

$$S \Rightarrow \frac{Z-1}{T} I_1(z) * \left(\frac{z-1}{T} + \omega_c \right) = \omega_c * e_{in}(z)$$

$$z I_1(z) = I_1(z) * (1 - \omega_c * T) + T * \omega_c * e_{in}(z)$$

Reversing the "Z" transform, we get: $I_1(k+1) = (1 - \omega_c * T) I_1(k) + T * \omega_c * e_{in}(k)$

Then the MATLAB code is : $I_1(k) = (1 - \omega_c * T) I_1(k-1) + T * \omega_c * e_{in}(k-1)$

2.2.2: Second Order Butterworth Low Pass Filter

Figure1.2 shows that we already get our FM signal from the PLL V_c signal. But, there is still high-frequency noise left after the loop filter. A second order Butterworth LPF could be used to remove this noise if we want better V_{cout} (FM signal), and its transfer function becomes part of our simulation.

a. For a second order Butterworth LPF, the $|H(j\omega)|$ curve is maximally flat and becomes rounded around near ω_c and rolls off at an ultimate rate of -40db/dec in the stop band. This makes it work well in this demodulator.

b. The transfer function of the second order Butterworth LPF is

$$H(s) = \frac{1}{(s / \omega_o)^2 + 2\xi(s / \omega_o) + 1} = \frac{1}{(s / \omega_o)^2 + \frac{1}{Q}(s / \omega_o) + 1} \quad Q = \frac{1}{2\xi}$$

$$Q = \frac{1}{\sqrt{2}} = 0.707$$

$$\omega_o = \omega_{cout} = \omega_{3db} = 2\pi * 4\text{khz}$$

$$H(s) = \frac{\omega_o}{s^2 + \frac{\omega_o}{Q}s + \omega_o^2} \quad a = \frac{\omega_o}{Q} \quad b = \omega_o$$

$$H(s) = \frac{b}{s^2 + as + b}$$

Applying the "Forward Euler" to mapping this transfer function from $S \rightarrow Z$

$$S \Rightarrow \frac{Z-1}{T} H(z) = \frac{V_{-cout}(z)}{V_{-c}(z)} = \frac{b^2 T^2}{(z-1)^2 + a(z-1) + b^2 T^2}$$

$$z^2 * V_{-cout}(z) = z * V_{-cout}(z) * (z - aT) + V_{-cout}(z) * (aT - b^2 T^2 - 1) + V_{-c}(z) * b^2 T^2$$

Reversing the transform yields the difference equation that we implement in MATLAB

$$V_{-cout}(k+2) = (2 - aT) * V_{-cout}(k+1) + (aT - b^2 T^2 - 1) * V_{-cout}(k) + b^2 T^2 * V_{-c}(k)$$

2.3 : Loop Filter

2.3.1: Loop filter types and order of the PLL along with loop stability

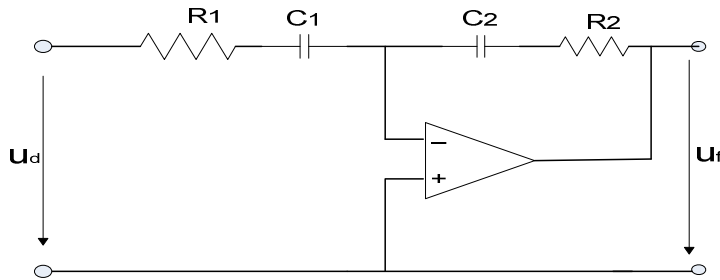
The output signal $u_d(t)$ of the PD consists a number of terms.

$$u_d(t) = U_i U_o \sin(\omega_i t + \theta_i) \left[\frac{4}{\pi} \cos(\omega_{ii} t + \theta_o) + \frac{4}{3\pi} \cos(3\omega_{ii} t + \theta_o) \dots \right]$$

In the locked state of the PLL, $\omega_i = \omega_{ii}$

$$u_d(t) = U_i U_o \left[\frac{2}{\pi} \sin \theta_e \dots \right] \approx K_d \sin(\theta_e)$$

Because these high frequencies are unwanted signals, they are filtered out by the loop filter. A first-order low-pass filter is used in most PLL. Three different types of loop filters are mostly used in PLL circuits: the passive lead-lag filter, the active lead-lag filter, and the active Proportional-plus-Integral filter. In this project, we chose active lead-lag filter as the loop filter. One typical circuit realization of an active lead-lag filter shown below:



$$\text{Transfer function: } F(s) = k_a \frac{1 + s\tau_2}{1 + s\tau_1} \quad \tau_1 = R_1C_1\tau_2 = R_2C_2 \quad k_a = C_1/C_2$$

Generally, the order of the PLL is always higher by 1 than the order of the loop filter. Because higher-order loop filters offer better noise cancellation, a loop filter of order 2 and higher are used in critical application. With higher-order loop filters, however, loop stability becomes an issue. Getting stable operation with a second-order PLL was easy because the open-loop transfer function had two poles and one zero. A pole creates a phase shift of -90° at higher frequencies, and a zero creates a phase shift of $+90^\circ$. When the poles and zero are properly located, the overall phase shift never comes close to -180° ; hence, the loop stays stable. If the loop filter has two or more poles, the phase shift can become larger than 180° . Hence the poles and zeros of the loop filter must be placed such that stability is maintained.

2.3.2: MATLAB code of the loop filter

We start from the transfer function of the loop filter:

$$H(s) = F(s) = k_a \frac{1 + s\tau_2}{1 + s\tau_1} = \frac{s + \omega_1}{s + \omega_0}$$

$$\omega_1 = \frac{1}{\tau_2}, \omega_0 = \frac{1}{\tau_1}, \quad k_a = \frac{\omega_1}{\omega_0}$$

Applying the “Forward Euler” to mapping this transfer function from S → Z

$$S \Rightarrow \frac{Z-1}{T} \frac{v_c(z)}{I(z)} = \frac{z-1+\omega_1 T}{z-1+\omega_0 T}$$

$$V_c(z) * (z-1) + V_c(z) * \omega_0 T = I(z) * (z-1) + \omega_1 T * I(z)$$

$$z * V_c(z) + V_c(z) * (\omega_0 T - 1) = z * I(z) + I(z) * (\omega_1 T - 1)$$

Reversing the transform yields the difference equation that we implement in MATLAB

$$V_c(k+1) + V_c(k) * (\omega_0 T - 1) = I(k+1) + I(k) * (\omega_1 T - 1)$$

$$V_c(k+1) = (1 - \omega_0 T)V_c(k) + I(k+1) + (\omega_1 T - 1)I(k)$$

So, the MATLAB code for the loop filter is: $V_c(k) = (1 - \omega_0 T)V_c(k-1) + I(k) - (1 - \omega_1 T)I(k-1)$

2.4 : Voltage Controlled Oscillator (VCO)

2.4.1: VCO General Theory

In PLLs, two fundamentally types of controlled oscillators are used: voltage control oscillator and current control oscillator. Figure 2.4.2 shows the simplified schematic of a VCO.

The operation of this circuit is as follows (figure 2.4.2): First, the input control signal is converted into a current signal. The NOR gates form an RS latch. Assume that the output signal of the left NOR gate is high (H) and the output signal of the right NOR gate is Low (L). Consequently, the P-channel MOS transistor P1 is on and the N-channel MOS transistor N₁ is off; furthermore, P₂ is off and N₂ is on. Therefore, the right terminal

of capacitor C is grounded, and the output current of the voltage-current converter flows into the left terminal of C; thus the voltage at that terminal ramps up in a positive direction. The upper threshold of the Schmitt triggers is set at half of the supply voltage $V_{DD}/2$. When the voltage at the left side of C exceeds that threshold, the RS latch changes state. Now the left terminal of C becomes grounded, and the output current of the voltage-current converter flows into the right terminal of C. The voltage at the right side of C ramps up now, until the right Schmitt triggers switches to the High state. This process repeats infinitely. If measured, the differential voltage across the capacitor, we would observe a triangular waveform.

2.4.2: VCO Architecture and Equation

The radian frequency ω_1 of the VCO output signal is proportional to the control signal V_c

$$\omega_1 = \omega_0 + K_0 V_c$$

K_0 is VCO gain; units are $\text{rad s}^{-1}\text{v}^{-1}$.

ω_0 is the radian center frequency of the PLL.

Most VCOs are powered from a unipolar power supply V_{DD} .

$$\text{So, } \omega_1 = \omega_0 + K_0(V_c - V_{DD}/2)$$

In the MATLAB simulation of this project:

$$\text{Set: } \omega_{\text{vco}} = \omega_1, \quad \omega_{\text{fr}} = \omega_0, \quad k_{\text{vco}} = K_0, \quad V_{\text{cfr}} = V_{DD}/2$$

$$\text{So, } \omega_{\text{vco}} = \omega_{\text{fr}} + k_{\text{vco}}(V_c - V_{\text{cfr}})$$

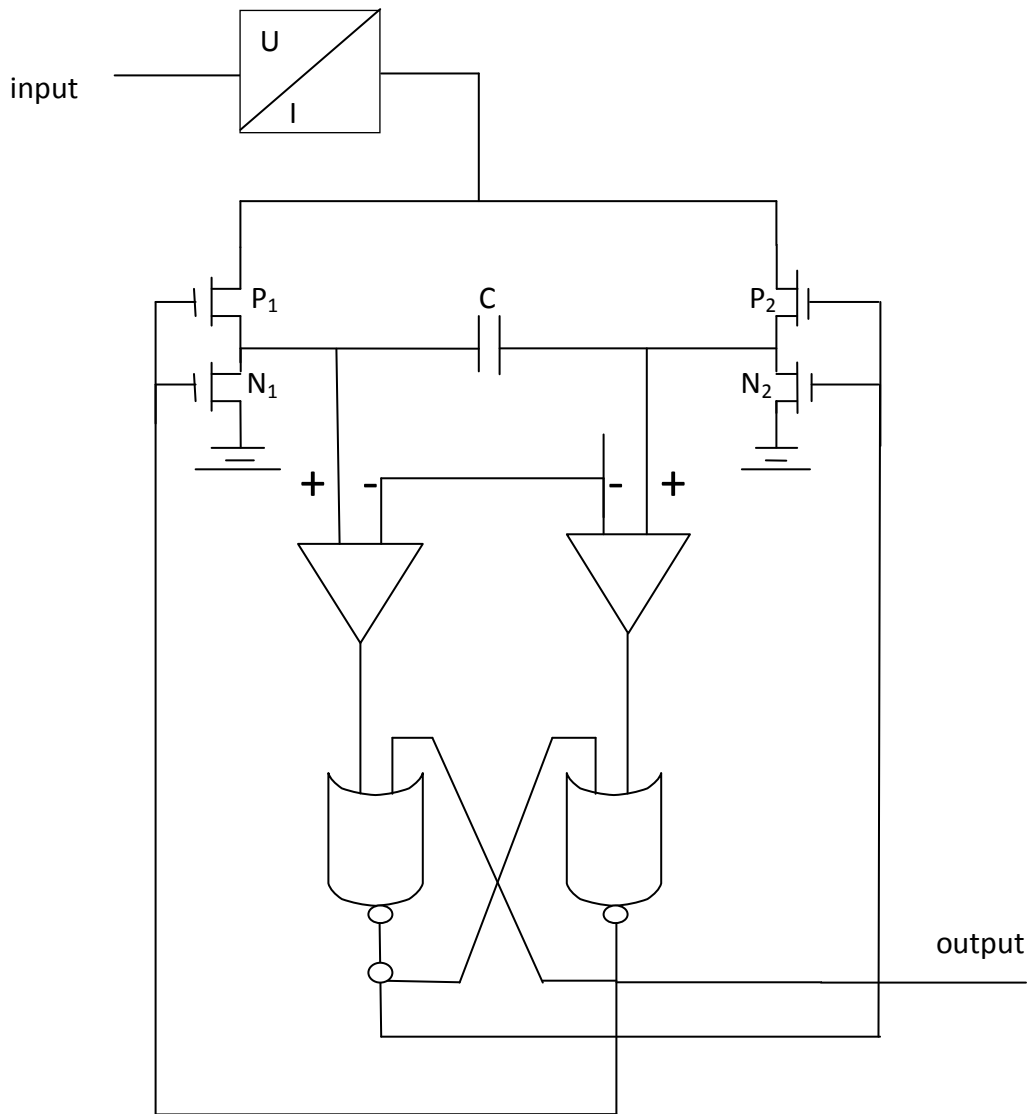


Figure 2.4.2

2.4.3 MATLAB code of VCO1 and VCO2

We start from the equation below,

$$\text{since } \phi_{out}(t) = \int \omega_{vco2}(t) dt$$

Applying the Laplace transform to transfer it from the time domain to the frequency domain, we get:

$$phi_out(s) = \frac{1}{S} \omega_{vco2}(s)$$

Apply "Forward Euler" to transfer from "S" field to "Z" field:

$$phi_out(z) = \frac{T}{z-1} \omega_{vco2}(z)$$

$$z * phi_out(z) = phi_out(z) + T \omega_{vco2}(z)$$

Reversing the "Z" transform, we get: $phi_out(k+1) = phi_out(k) + \omega_{vco2}(z) * T$

So, the MATLAB code for VCO1 and VCO2 is:

$$\omega_{vco1} = \omega_{fr} + k_{vco} * (V_{audio} - V_{cfr})$$

$$\omega_{vco2}(k) = \omega_{fr} + k_{vco} * (V_c(k) - V_{cfr})$$

$$phi_out(k+1) = phi_out(k) + \omega_{vco2}(z) * T$$

Chapter3 : System Level Description of the PLL

3.1: Operation of the three basic functional blocks

Every PLL is a nonlinear system. Fortunately, most PLLs can be analyzed using linear techniques when in a locked condition. The basic PLL block diagram is shown in Figure 1.1 and contains three essential functional blocks: a phase detector (PD), a loop filter (LF) and a voltage-controlled oscillator (VCO) that were described in chapter2. And normally, it is useful to add a low pass filter before loop filter to also remove high frequency noise in the loop. Let's look at the operation of the three functional blocks of figure 1.1.

(1) The Phase Detector (PD) compares of the output signal with the phase of the input signal and develops an output signal $e_{in}(t)$.

$$e_{in}(t) = k_d \theta_e \quad \text{where: } \theta_e \text{ is phase error: } \theta_e = \theta_i - \theta_{ij}$$

k_d is the gain of PD in v/rad

(2) The Loop Filter (LF) attenuates the superimposed AC component of $e_{in}(t)$.

$$e_{in}(t) = \text{DC component} + \text{superimposed AC component}$$

(3) The VCO oscillates at an angular frequency:

$$\omega_{ij} = \omega_{vco} = \omega_{fr} + k_{vco}(V_c - V_{cfr})$$

where ω_{fr} is the free running frequency of the VCO

k_{vco} is the VCO gain in $s^{-1}v^{-1}$

Let us now see how the three building blocks work together. First, we assume the angular frequency of the input signal $u_i(t)$ is equal to the free running frequency ω_{fr} . The VCO then operates at its free running frequency ω_f . As we see, the phase error θ_e is zero. If θ_e is zero, the output signal u_d of PD must also be zero. Consequently, the output signal of the loop filter u_c will be zero. This is the condition that permits the VCO to operate at its free running frequency.

If the phase error θ_e were not zero initially, the PD would develop a nonzero output signal u_d . After some delay, the loop filter would also produce a finite signal u_c . This would cause the VCO to change its operating frequency in such a way that the phase error finally vanishes.

Assume now that the frequency of the input signal is changed suddenly at time by the amount $\Delta\omega$. The phase of the input signal then starts leading the phase of the output signal. A phase error is built up and increases with time. The PD develops a signal $u_d(t)$, which also increase with time. With a delay given by the loop filter, $u_c(t)$ will also rise. This causes the VCO to increase its frequency. The phase error becomes smaller now, and after some settling time the VCO will oscillate at a frequency that is exactly the frequency of the input signal. Depending on the type of loop filter used, the final phase error will have been reduced to zero or to a finite value. The VCO now operates at a frequency which is greater than its free running frequency ω_{fr} by an amount $\Delta\omega$. This will force the signal $u_c(t)$ to settle at a final value of $u_c(t) = \Delta\omega/K_{vco}$

If the free running frequency of the input signal is frequency-modulated by an arbitrary low-frequency signal, then the demodulated signal is the output signal of the loop filter. u_c = FM output, that is why the PLL can be used as an FM detector.

3.2: PLL transfer function and MATLAB code

(1) Phase detector output: $e_{in} \approx k_d \theta_e$ $\theta_e = \theta_i - \theta_{ii}$

PD (with multiplier) transfer function: $\frac{e_{in}(s)}{\theta_e(s)} = k_d$

PD(multiplier) MATLAB code: $e_{in}(k) = V_{ref}(k) * V_{out}(k)$

PFD(with flipflop) MATLAB code is:

```
%phase detector
V_ref = 0.5 + 0.5*sign(V_fm);
for k = 2:N-1;
% ein(k)=V_fm(k)* V_out(k);
% ein(k)=u(k)* V_out(k);
R(k) = Q1(k-1)*Q2(k-1);
if (( V_ref(k-1)== 0 ) && ( V_ref(k)== 1))
Q1(k)=1;
elseif ((R(k-1)==0) && (R(k)==1))
Q1(k)= 0;
else Q1(k) = Q1(k-1);
end;
if (( V_out(k-1)== 0 ) && ( V_out(k)== 1))
```



```

Q2(k)=1;

elseif ((R(k-1)==0) && (R(k)==1))

Q2(k)= 0;

else Q2(k) = Q2(k-1);

end;

ein(k) = Q1(k)-Q2(k);

```

(2) Loop filter

active lead-lag filters: $F(s) = k_a \frac{1 + s\tau_2}{1 + s\tau_1}$

In this project, we have: $F(s) = H(s) = \frac{s + \omega_1}{s + \omega_0}$

MATLAB code of the loop filter is discussed in §2.3.2:

$$V_c(k) = (1 - \omega_0 T)V_c(k-1) + I(k) - (1 - \omega_1 T)I(k-1)$$

(3) VCO transfer function: $H_{vco}(s) = \frac{k_{vco}}{s}$

$$\omega_{vco}(t) = \omega_{fr} + k_{vco}u_c(t)$$

As we discussed in §2.4.2 and §2.4.3, we get:

$$\text{VCO1 MATLAB code: } \omega_{vco1} = \omega_{fr} + k_{vco} * (v_{audio} - v_{cfr})$$

$$\text{VCO2 MATLAB code: } \omega_{vco2}(k) = \omega_{fr} + k_{vco} * (v_c(k) - v_{cfr})$$

$$phi_out(k+1) = phi_out(k) + \omega_{vco2}(z) * T$$

(4) Simulation Result (PLL with PD and PLL with PFD):

The simulation code (PLL with PD) is on the appendix A₁.

In Figure 3.2.1, we could see signal V_c of loop filter output tracked the input signal V_{audio} after a limited time delay. V_c has almost the same frequency and amplitude as the input V_{audio} . Of course, signal V_c has some high frequency noise. It is not as the same clean as the signal V_{audio} .

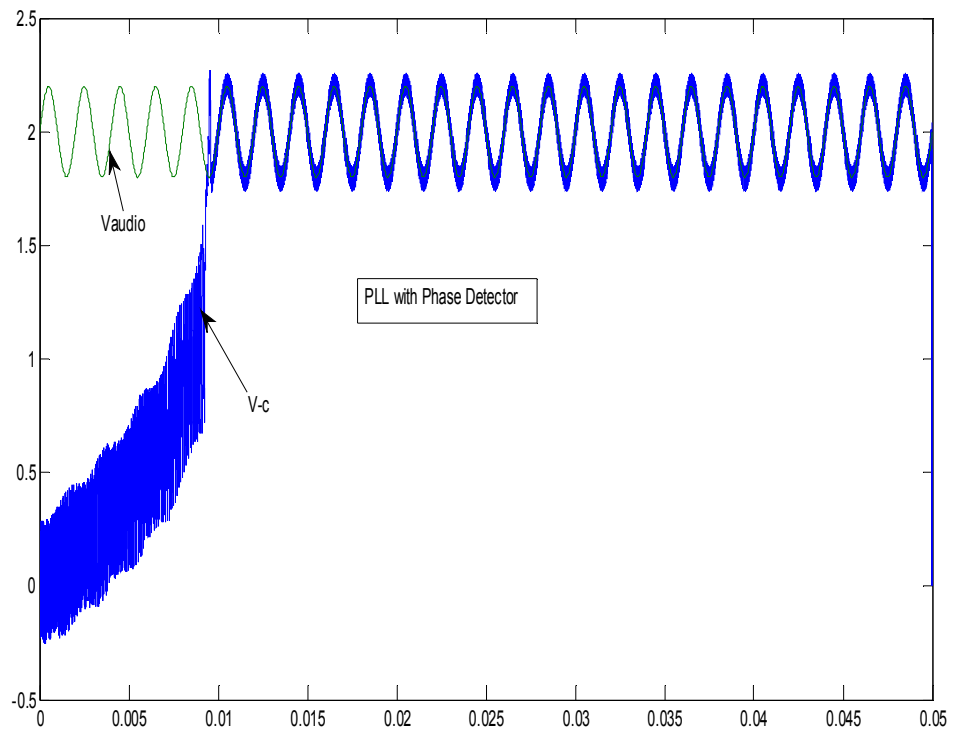


Figure 3.2.1 V_{audio} versus V_c (PLL with PD)

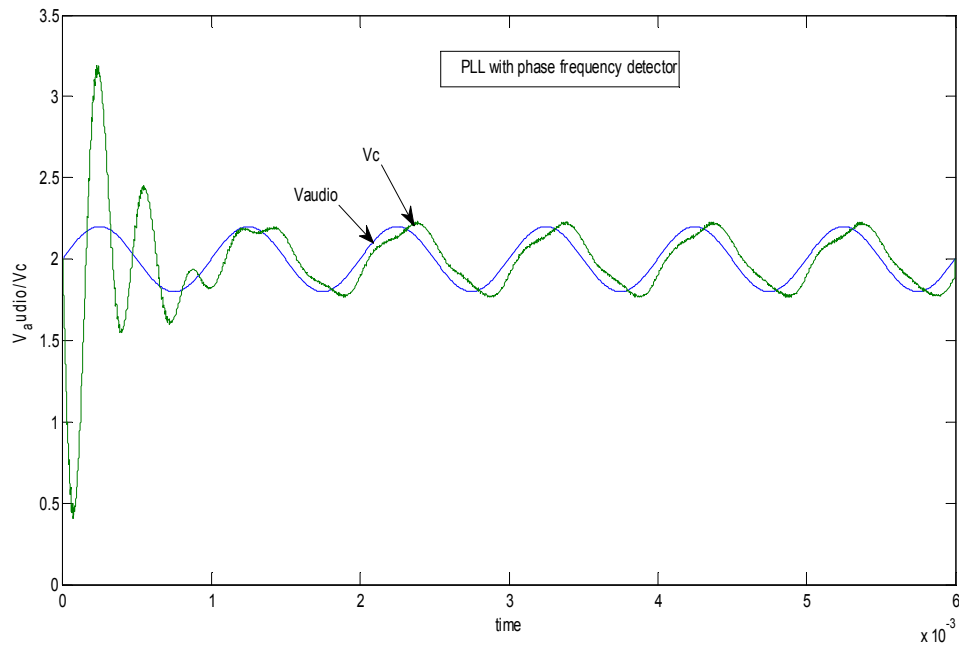


Figure 3.2.2 V_{audio} versus V_c (PLL with PFD)

The simulation code (PLL with PFD) is on the appendix A₂.

From figure 3.2.2, the loop filter output signal V_c tracked input signal V_{audio} after a limit time delay. Even the wave of signal V_c has some deform, because of noise.

3.3: Loop parameter

From figure 1.1, we can get the PLL open loop transfer function: $G(s) =$

$$\frac{k_d k_{vco} F(s)}{s} \text{ and closed loop transfer function: } H_{\text{PLL}}(s) = \frac{G(s)}{1 + G(s)} =$$

$$\frac{k_d k_{vco} F(s)}{s + k_d k_{vco} F(s)}$$

From open loop transfer function $G(s)$ and closed loop transfer

function $H_{\text{PLL}}(s)$, we can derive loop parameters such as natural frequency, damping factor and loop gain.

(1). Natural frequency and damping factor

The denominators of the transfer function in normalized form:

Denominators = $s^2 + 2 \xi \omega_n s + \omega_n^2$ where ω_n is the natural frequency

ξ is the damping factor

For the active PI (Proportional- Integral) filter:

$$\omega_n = \sqrt{\frac{k_d k_{vco}}{\tau_1}}, \quad \xi = \frac{\omega_n \tau_2}{2}$$

The transfer function reduces to

$$H_{PLL}(s) = \frac{2 \xi \omega_n s + \omega_n^2}{s^2 + 2 \xi \omega_n s + \omega_n^2}$$

Natural frequency and damping are a convenient description of the properties of a pole pair and well suited for second-order loops.

if $\xi < 1$, the poles are a complex-conjugate pair

if $\xi = 1$, the poles are real and coincident

if $\xi > 1$, the poles are real and separate

The typical values of ξ is: $0.2 < \xi < 2$ and the preferred value is: $\xi = 0.707$

$$10^{-5} < \omega_n < 10^8 \text{ rad/s}$$

The second-order PLL is actually a low-pass filter for input phase signals $\Theta_i(t)$.

The damping factor ξ has an important influence on the dynamic performance of the PLL.

(2). Loop gain k , for a second-order PLL:

$$k = k_d k_{vco} k_1 = \frac{k_d k_{vco} \tau_2}{\tau_1}$$

$$\xi = \frac{1}{2} \sqrt{k \tau_2} \quad \omega_n = \sqrt{\frac{k}{\tau_2}}$$

The 3-db bandwidth of a second-order PLL can be calculated as:

$$\omega_{3db} = K \left(\frac{1}{2} + \frac{1}{4\xi^2} + \frac{1}{2} \sqrt{1 + \frac{1}{\xi^2} + \frac{1}{2\xi^4}} \right)^{\frac{1}{2}}$$

K is a good indication of the low pass corner frequency of $H(s)$. And k also has a dominant influence on the speed of response and bandwidth of the PLL.

Chapter 4: How This Research Adds To Basic PLL

- Optimum Time Varying Filter

In figure 1.2, we include a band pass filter block in front of the PLL to improve the quality of the FM signal. In this chapter, we will discuss how to realize the optimum time varying filter as the band pass filter.

4.1: Optimum filter and optimum time varying filter related to the band pass filter

An optimum filter is a filter which mimics the desired time domain or frequency domain response of a system. The goal of optimum filtering is as follows: If a certain signal is corrupted by noise in a channel, the optimum filter which may be applied to the composite signal is that which maximizes the S/N ratio at its output. Suppose the target input signal is a amplitude, phase, or frequency modulated signal whose spectrum occupies a symmetrical bandwidth centered about some carrier frequency. If we were to assume that the modulating signal was such that the spectrum were essentially flat over the signal bandwidth, then the matched filter would be an ideal band pass filter whose bandwidth matched that of the modulated signal.

Suppose that an input signal is an FM signal. At any instant in time we could say that the signal is a single frequency. Assume a band pass filter were designed that would move its center frequency in exact synchronization with the input. Then the output of

this filter would still be FM input, and since the band width of the filter, wherever it is centered, could be made arbitrarily small, the output noise could be essentially zero. As noise is closer to zero, the S/N ratio equals to infinite.

4.2: Synchronous filtering idea and mathematical framework

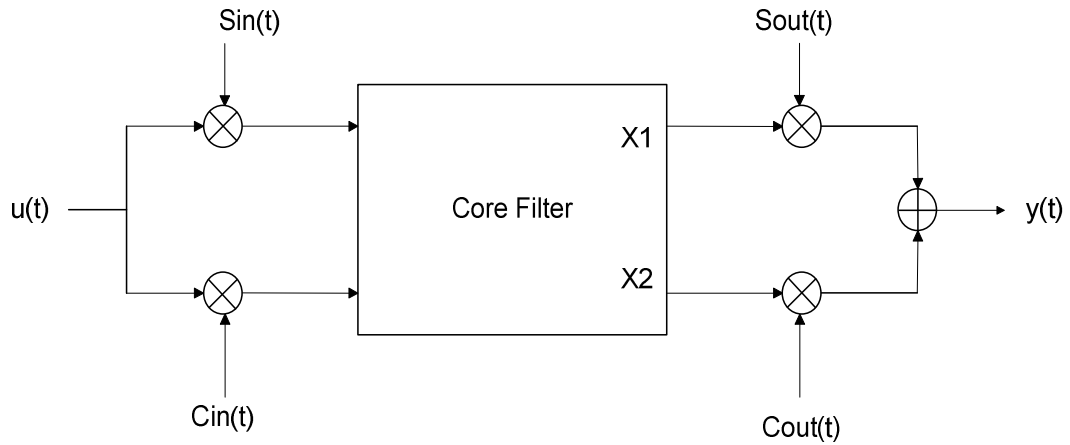


Figure 4.2 Synchronous Filter

Figure 4.2 shows a diagram of a synchronous filter, also called a complex filter. A complex filter constitutes the heart of integrated filtering circuitry in modern RF electronics. The input 'u' is processed via multiplication with the signals, $S_{in}(t)$ and $C_{in}(t)$, resulting in a pair of inputs to the core filter. In a complex filter these signals are sinusoidal signals in quadrature. The output of the overall filter is derived by adding the post-processed core filter outputs. The post- processing is accomplished via respective multipliers driven by another pair of signals, $S_{out}(t)$ and $C_{out}(t)$ [cited in Reference 1 and 2]. We may write down the dynamical equations of the band pass filter:

$$\begin{aligned} \frac{d \bar{x}}{d t} &= A \bar{x} + \bar{b} u \\ y &= \bar{c}^T \bar{x} + d u \end{aligned} \quad [1]$$

$$A = \begin{bmatrix} -\frac{\omega_o}{2Q} & -\omega_A \\ \omega_A & -\frac{\omega_o}{2Q} \end{bmatrix} ; \quad \bar{b} = \bar{c} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad [2]$$

If $Q \gg 1$ $\omega_A \approx \omega_o$ would be center frequency of the filter.

$$\text{Let } \bar{x}' = e^{J\phi(t)} \bar{x} ; \quad \bar{x} = e^{-J\phi(t)} \bar{x}'$$

$$\text{Where } J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} ;$$

$$\text{So, } e^{J\phi(t)} = \begin{bmatrix} \cos(\phi(t)) & \sin(\phi(t)) \\ -\sin(\phi(t)) & \cos(\phi(t)) \end{bmatrix}$$

$$\begin{aligned} \frac{d \bar{x}'}{d t} &= \frac{d}{d t} (e^{J\phi(t)} \bar{x}) = \frac{d}{d t} (e^{J\phi(t)}) \bar{x} + e^{J\phi(t)} \frac{d \bar{x}}{d t} \\ &= \frac{d}{d t} (e^{J\phi(t)}) \bar{x} + e^{J\phi(t)} (A \bar{x} + \bar{b} u) \end{aligned}$$

$$= \left(\frac{d}{d t} (e^{J\phi(t)}) + e^{J\phi(t)} A \right) \bar{x} + e^{J\phi(t)} \bar{b} u \quad [3a]$$

$$\text{Since } \bar{x} = e^{-J\phi(t)} \bar{x}'$$

Replace it into equation [3a]. We get:

$$\begin{aligned} \frac{d \bar{x}'}{d t} &= \left(\frac{d}{d t} (e^{J\phi(t)}) + e^{J\phi(t)} A \right) e^{-J\phi(t)} \bar{x}' + e^{J\phi(t)} \bar{b} u \\ &= e^{J\phi(t)} J \dot{\phi}(t) e^{-J\phi(t)} \bar{x}' + A \bar{x}' + e^{J\phi(t)} \bar{b} u \end{aligned}$$

So
$$\frac{d \bar{x}}{d t} = (A + J \dot{\phi} (t)) \bar{x} + e^{J \phi (t) } \bar{b} u \quad [3]$$

$$y = \bar{c}^T \bar{x} + d u = \bar{c}^T e^{-J \phi (t) } \bar{x} + d u$$

It has been assumed that A and J commute in the simplification of the state matrix, which is an easy condition to ensure if one employs typical filter topologies [5]. The form of A in [2] satisfies this condition. Observe that if $\phi(t) = \omega_m t$ then the new state matrix is $A + \omega_m J$, which is constant. The new off-diagonal terms are $\pm(\omega_m - \omega_A)$ whose magnitude suggests the center frequency of the transformed core filter, which may be engineered to be much lower than that of the original. This formulation leads naturally to complex filters, which are a subset of synchronous filters [cited in reference 1 and 2]. Here is the mathematics to approve it.

Using equation [3]
$$A = \begin{bmatrix} -\frac{\omega_o}{2Q} & -\omega_A \\ \omega_A & -\frac{\omega_o}{2Q} \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\phi (t) = \omega_m t$$

We get three parameters: (1), $A + J \dot{\phi} (t)$ (2), $e^{J \phi (t) } \bar{b}$ (3), $\bar{c}^T e^{-J \phi (t) }$

Let's analyze each of these:

$$(1), A + J \dot{\phi} (t) = A + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \omega_m = A + \begin{bmatrix} 0 & \omega_m \\ -\omega_m & 0 \end{bmatrix}$$

$$= A + \begin{bmatrix} -\frac{\omega_o}{2Q} & -\omega_A + \omega_m \\ \omega_A - \omega_m & -\frac{\omega_o}{2Q} \end{bmatrix}$$

So, the center frequency of the transformed core filter is

$$\omega_o = \pm(\omega_m - \omega_A)$$

$$\begin{aligned} (2), \quad e^{J\phi(t)} \overline{b} &= \begin{bmatrix} \cos(\phi(t)) & \sin(\phi(t)) \\ -\sin(\phi(t)) & \cos(\phi(t)) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\ &= \begin{bmatrix} b_1 \cos(\phi(t)) + b_2 \sin(\phi(t)) \\ b_2 \cos(\phi(t)) - b_1 \sin(\phi(t)) \end{bmatrix} \\ &= b \begin{bmatrix} \sin(\phi(t) + \alpha) \\ \cos(\phi(t) + \alpha) \end{bmatrix} \\ &= \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix} \quad [4] \end{aligned}$$

$$\text{Where } b = \sqrt{b_1^2 + b_2^2}; \quad \alpha = \tan^{-1}\left(\frac{b_1}{b_2}\right)$$

These operators show clearly how the input is processed using quadrature multiplication to become a pair of inputs to the core filter. A similar computation on the transformed term in the input-output equation shows how the state variable filter outputs are processed using quadrature multiplication.

$$(3), \quad \overline{c}^T e^{-J\phi(t)} = [c_1 \quad c_2] e^{-J\phi(t)}$$

$$\begin{aligned}
&= c [\sin(\phi(t) + \beta) \quad \cos(\phi(t) + \beta)] \\
&= [s_{out}(t) \quad c_{out}(t)] \quad [5]
\end{aligned}$$

Where $c = \sqrt{c_1^2 + c_2^2}$; $\beta = \tan^{-1} \left(\frac{c_1}{c_2} \right)$

These parameters process the input and produce the output, showing that the synchronous filter implements a complex filter.

If input signal $u(t)$ were a pure sine wave plus some noise, we could filter this with an arbitrarily sharp band pass filter centered at the input signal frequency and remove essentially all of the noise. The mathematics guarantees that the composite synchronous filter would perform the desired narrow band filtering.

4.3 : Optimum FM signal filtering

4.3.1: General Theory

$u = \sin(\omega t + f(t))$, A phase modulated signal is now being applied to the core filter. The input will be processed by constant frequency quadrature signals. The core filter will be time-varying. The signal will be filtered in an effectively narrow band pass filter, removing essentially all noise.

The time varying state space transformation is in the following way:

Let $\phi(t) = \omega_m t + f(t)$

state matrix: $A + J \dot{\phi}(t) = A + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (\omega_m + \dot{f}(t))$

$$= \begin{bmatrix} -\frac{\omega_o}{2Q} & -\omega_A \\ \omega_A & -\frac{\omega_o}{2Q} \end{bmatrix} + \begin{bmatrix} 0 & \omega_m + \dot{f}(t) \\ -(\omega_m + \dot{f}(t)) & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{\omega_o}{2Q} & (\omega_m + \dot{f}(t) - \omega_A) \\ -(\omega_m + \dot{f}(t) - \omega_A) & -\frac{\omega_o}{2Q} \end{bmatrix}$$

The off-diagonal term is the center frequency of the transformed core filter, which is a time varying term. So, the filter is now a time varying filter. An FM signal input with an arbitrarily sharp filter produces an output that is free of noise.

4.3.2: Optimum time varying FM signal BPF of this project

From the discussion before, we may construct the BPF of this project.

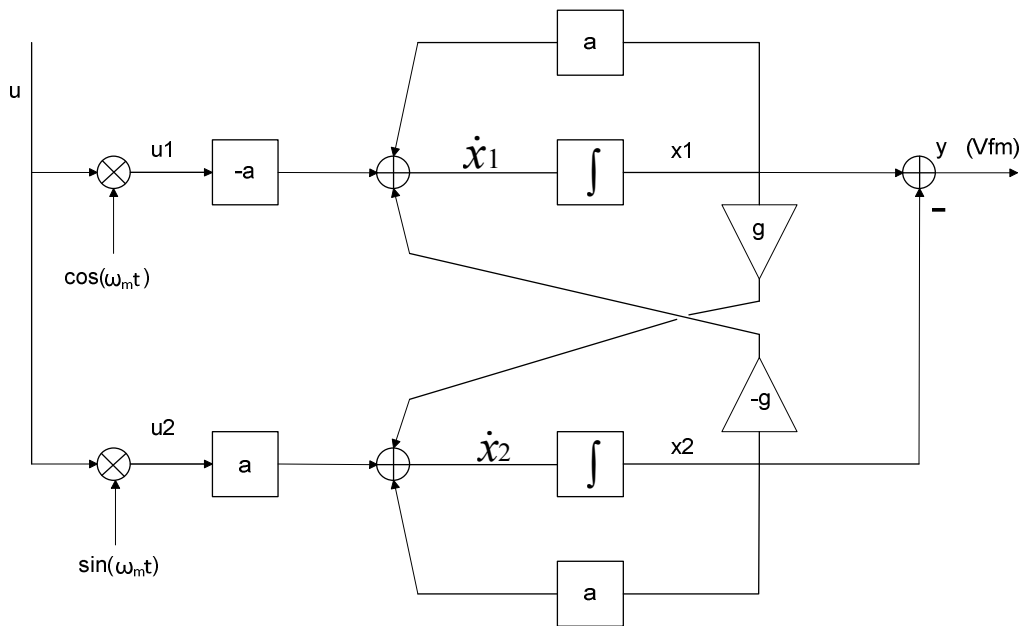


Figure 4.3.2 Band Pass Filter (time-varying)

$$\dot{x}_1 = ax_1 - gx_2 - au_1$$

$$\dot{x}_2 = ax_2 + gx_1 + au_2$$

$$y = x_1 - x_2$$

$$u_1 = u \cos(\omega_m t)$$

$$u_2 = u \sin(\omega_m t)$$

$$g(t) = g_1 + k_{vco}[c(V_{audio}(t) - 2) + (1 - c)V_{out}(t)]$$

$$sx_1 = ax_1 - gx_2 - au_1$$

$$sx_2 = ax_2 + gx_1 + au_2$$

Using the Forward Euler mapping, we have: $\frac{z-1}{T}x_1 = ax_1 - gx_2 - au_1$

$$\frac{z-1}{T}x_2 = ax_2 + gx_1 + au_2$$

$$zx_1 = x_1(1 + aT) - gTx_2 - aTu_1$$

$$zx_2 = gTx_1 + (1 + aT)x_2 + aTu_2$$

$$x_1(k) = (1 + aT)x_1(k-1) - gTx_2(k-1) - aTu_1(k-1)$$

$$x_2(k) = gTx_1(k-1) + (1 + aT)x_2(k-1) + aTu_2(k-1)$$

$$y(k) = x_1(k) - x_2(k)$$

Now, we start to discuss the parameter “g” associated with the band pass filter.

Since,

$$g(t) = g_1 + k_{vco}[c(V_{audio}(t) - 2) + (1 - c)V_{out}(t)]$$

First, if we set $k_{vco} = 0$, then, $g(t)$ becomes a constant equal to g_1 , and the BPF is time invariant. Secondly, If we set $k_{vco} \neq 0$, parameter “g” becomes a function of time, $g = g(t)$, and we get a time-varying band pass filter.

(c), BPF (with $g(t)=g_1$) simulation result

The simulation code is in the appendix B.

From figure 4.3.2, we could see that after a time delay, we get the output signal ‘Y’ of the band pass filter that is a stable sine wave whose amplitude is the same as the input signal ‘u’.

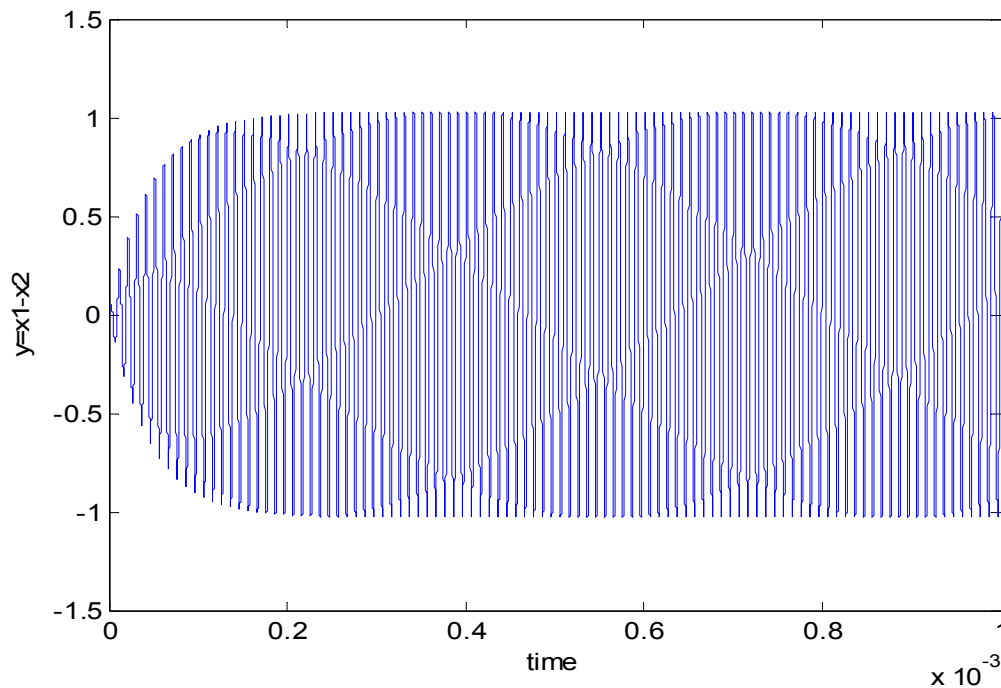


Figure 4.3.2 BPF output signal y

Chapter 5: Simulation Result

In the MATLAB code, we have a feedback parameter, “ g ”, having four parts: g_1 , g_2 , g_3 and g_4 , shown in figure 5.1 which is a reproduction of Figure1.2.

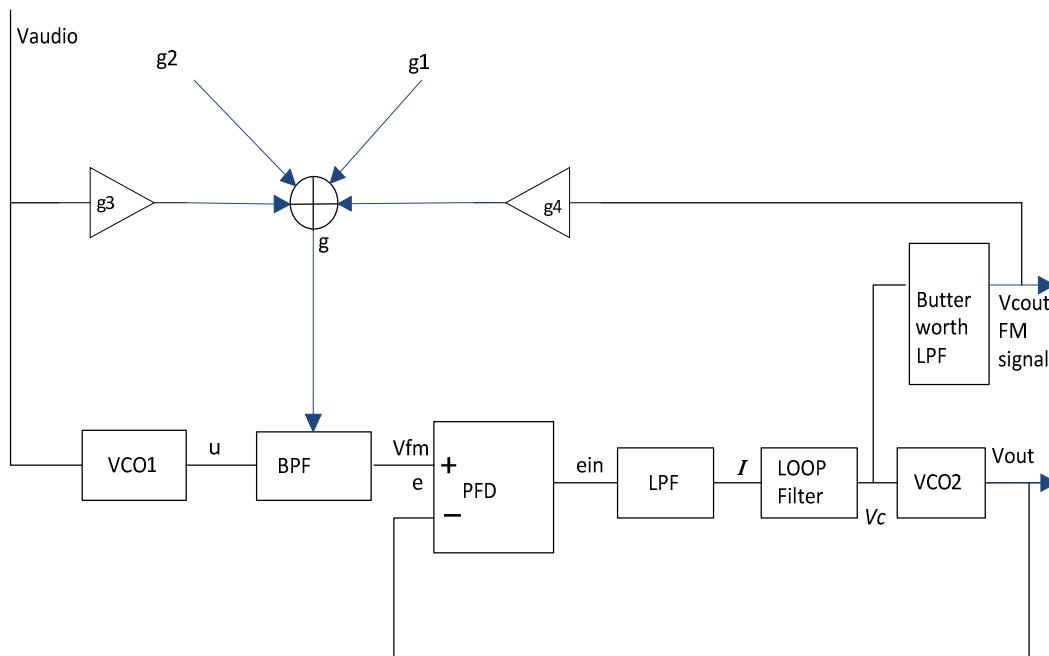


Figure 5.1 PLL FM Demodulator with BPF Building Block

By controlling “ g ”, we were able to make the Band Pass Filter be a time-varying filter or invariant filter. This gave us the chance to directly compare the simulation results for two kinds of demodulator (PLL with time-varying BPF and PLL with the invariant BPF). Here is the equation for $g(t)$ showing the varying components.

$$g(t) = g_1 + K_{vco} * [C * (V_{audio}(t) - 2) + (1 - C) * V_{cout}(t)]$$

$$= g_1 - 2CK_{vco} + CK_{vco}V_{audio}(t) + (1 - C)K_{vco}V_{cout}(t)$$

$$= g_1 + g_2 + g_3V_{audio}(t) + g_4V_{cout}(t)$$

where g_1 is constant, $g_2 = -2CK_{vco}$, $g_3 = CK_{vco}$, $g_4 = (1 - C)K_{vco}$

The parameter 'C' was added to allow more control of the overall loop.

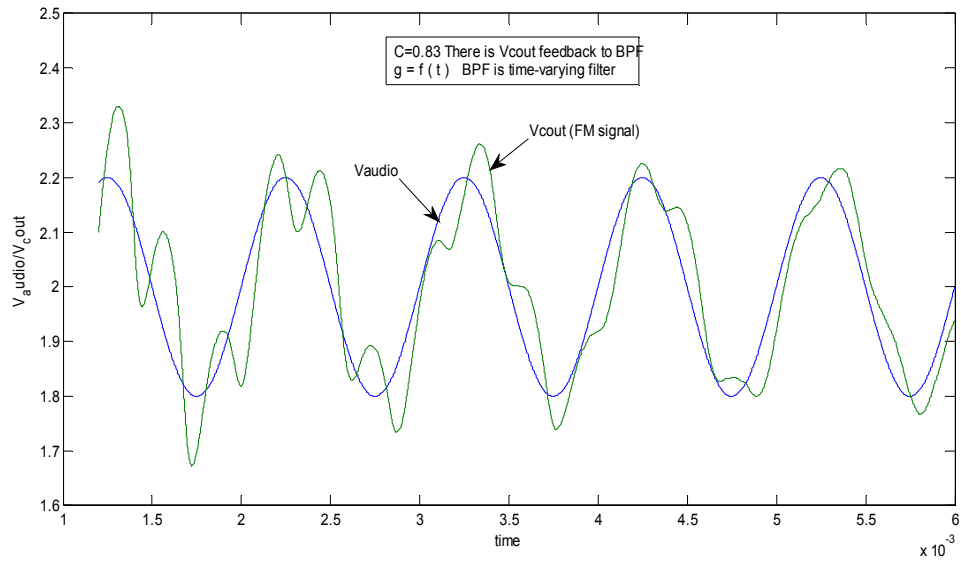


Figure 5.2 Input V_{audio} versus V_{cout} (output FM signal)

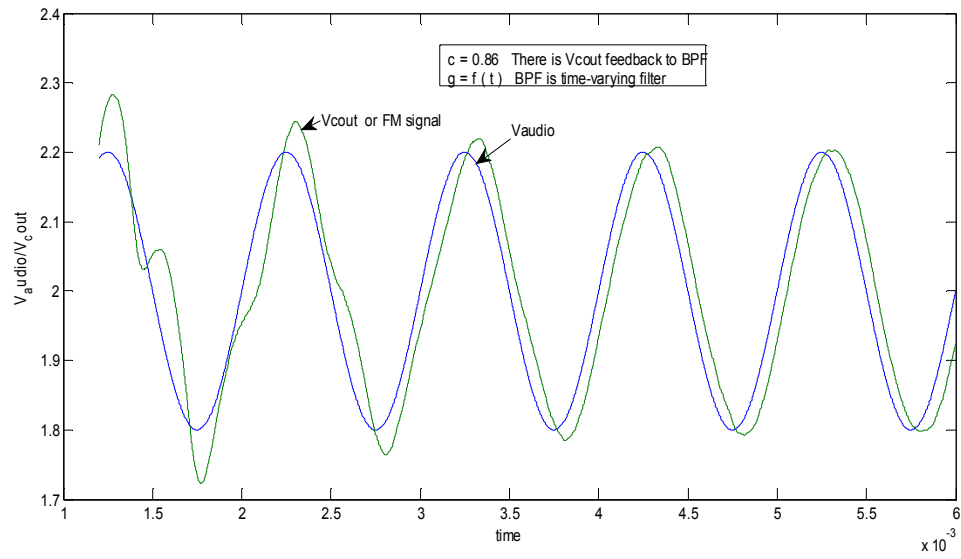


Figure 5.3 Input V_{audio} versus $V_{\text{c out}}$ (output FM signal)

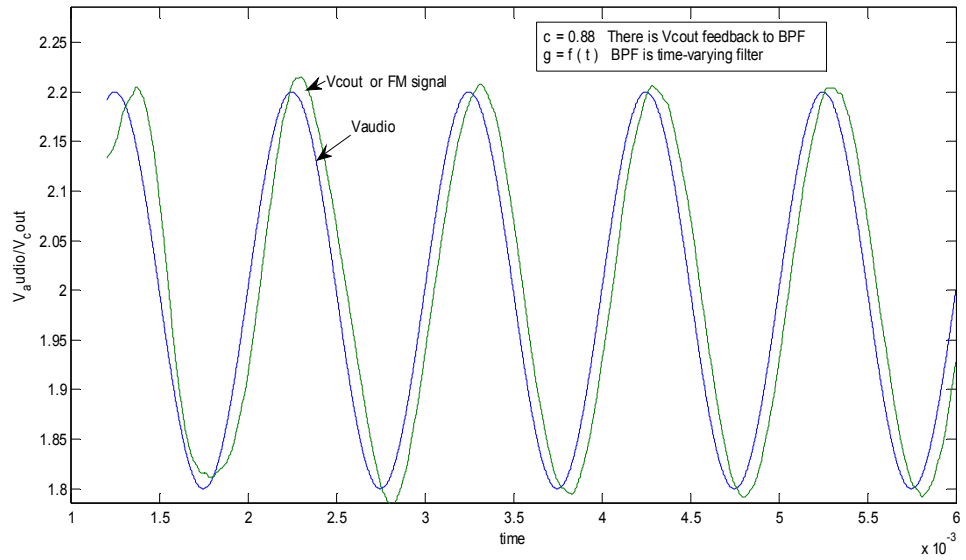


Figure 5.4 Input V_{audio} versus $V_{\text{c out}}$ (output FM signal)

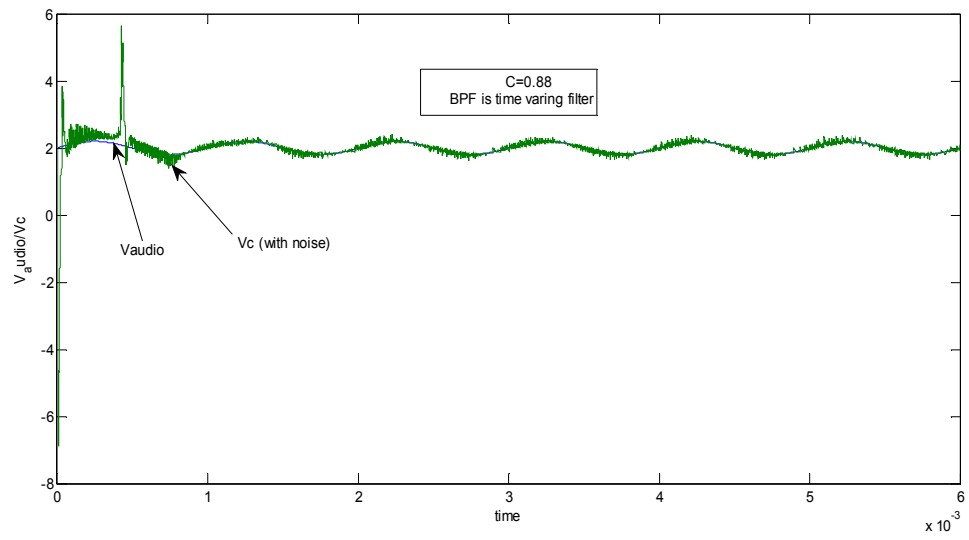


Figure 5.5 Input V_{audio} versus V_c (with high frequency noise)

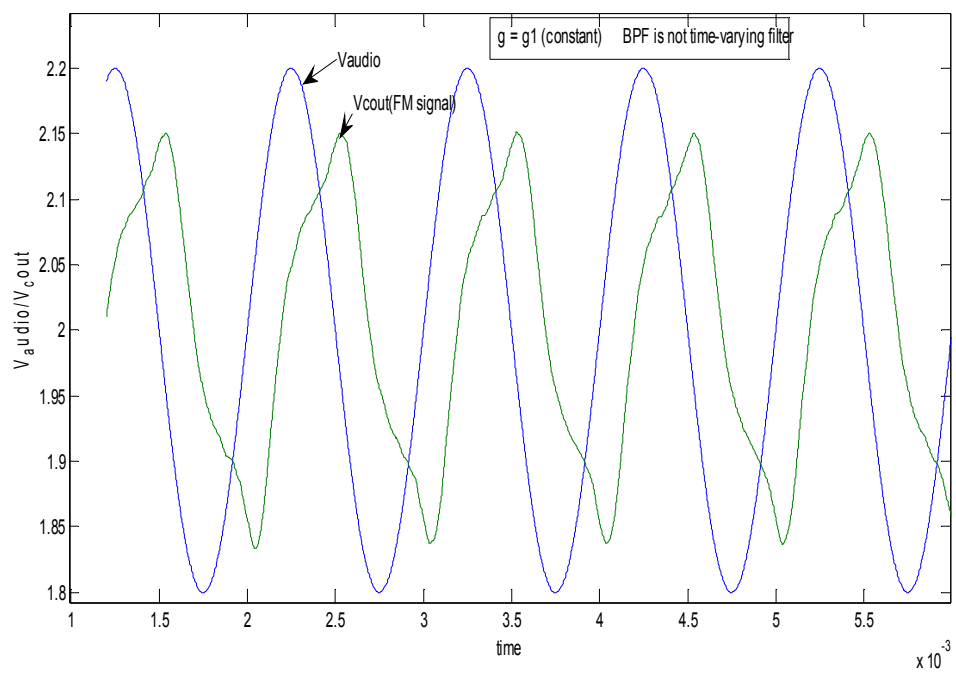


Figure 5.6 Input V_{audio} versus $V_{\text{c,out}}$ (output FM signal)

Figures 5.2 - 5.4 show the comparison between the original audio modulation signal and the recovered signal as our feedback parameter 'C' is varied. A value of $C = 1$ for loop feedback proved to be too big. The results for $C = 0.83$, 0.86 , and 0.88 are shown in Figures 5.2, 5.3, and 5.4, respectively. Note the high quality achieved with $C = 0.88$ in Figure 5.4.

Figure 5.5 shows the loop filter output V_c versus V_{audio} . We can see that there is significant high frequency noise in V_c . This kind of noise is undesirable in our FM demodulated signal. This shows that the Second Order Butterworth Low Pass Filter is necessary in this system if we want to have a clean FM signal (or V_{cout}) from the demodulator. Its position is shown in Figure 5.1. Figures 5.2 - 5.4 showed results that includes this filter.

Figures 5.6 shows the demodulator output signal V_{cout} (FM signal) compared to the original audio signal when the time varying filter is made time-invariant. Clearly, the output FM signal has become deformed. The optimum time-varying filter plays a very important role in the PLL FM Demodulator.

It is worth noting that some problems occurred during our simulations. For example, when we first put together the whole demodulator, we didn't consider having the Butterworth LPF block. But the simulation results shown in Figure 5.5 made the second order Butterworth LPF clearly necessary, allowing us to get a clean FM demodulated signal from the output of this Butterworth LPF. We also noticed that too

much feedback – that is, $C \approx 1$ – did not provide good results. We are not fully sure why this happened.

Chapter 6: Research Conclusion and Possible Future Work

Phase locked demodulators are widely used for reception of amplitude modulation(AM), phase modulation(PM), and frequency modulation(FM). In this research, we concentrated on phase locked FM demodulators. Our main two parts are the PLL block and the Optimum Time Varying Filter block. We started our work by building up three basic blocks of the PLL; the PFD, the Loop Filter and the VCO. By carefully studying the structure and the operation of these blocks, we were able to make good choices regarding the types of each block and their parameters. As a result, we got the desired results in our PLL simulations. After that, in order to improve the quality of the FM signal, we worked on the Optimum Time Varying Filter which is the main research part in the overall thesis. We studied the relationship of the optimum filter (or matched filter), the optimum time-varying filter and our band pass filter. We then derived the necessary equations to realize our synchronous filter. Based on the theory and equation discussion, we were able to design the band pass filter. After we put all of the blocks together to realize our FM PLL demodulator, we simulated the whole demodulator and recovered the FM demodulated signal which showed that we had achieved our research goal-namely, to improve the quality of the PLL FM Demodulator.

These results were very satisfying when I finished the simulations. At the same time, I noticed the distortion problem in the V_{cout} signal which was affected by the feedback of the band pass filter. The choice of constant "C" is critical. If I had more time, I would work on this and find a way to get rid of the V_{audio} signal in the BPF feedback and only keep the V_{cout} signal as part of the feedback. In that case, if we still could get a good quality FM signal (V_{cout}), it would be perfect.

Reference

- 1, D. R. Frey, "Optimal Filtering of AM and FM signal", submitted to Iscas 2009
- 2, D. R. Frey, "Synchronous Filtering", *Trans. Circuits and Syst, I: Reg. papers*, vol. 53, no. 8, 2006, pp.1772-1782
- 3, D. H. Wolaver, Phase-Locked Loop Circuit Design, Prentice Hall, Englewood Cliffs, NJ, 1991
- 4, W. R. Evans, Control-System Dynamics, McGraw-Hill, New York, 1954
- 5, H. W. Bode, Network Analysis and Feedback Amplifier Design, Van Nostrand, New York, 1945
- 5, A. Blanchard, Phase-Locked Loops, Wiley, New York, 1976
- 6, J.W.M. Bergmans, "Effect of Loop Delay on stability of Discrete-time PLL", *IEEE Trans, Circuits & Sys, I*, 42, 229-231, Apr.1995

Appendix

A : PLL Simulation MATLAB Code and Simulation Result

A1: PLL (with multiplier Phase detector) simulation

```
% VCO Block + PLL(PD) Block

T = 1e-7;

T_end = 0.05; N = fix(T_end/T); time = T*[0:N-1];

w_in = 2*pi*500;

w_fr = 2*pi*50e3;

V_cfr = 2; K_vco = 2*pi*10e3;

w_0 = 2*pi*10;

w_1 = 2*pi*1e3;

w_c = 2*pi*10e3;

phi_out = zeros(1,N);

w_vco2 = zeros(1,N);

V_c = zeros(1,N);

V_out = zeros(1,N);

phi_in = zeros(1,N);
```



```

ein = zeros(1,N);
I = zeros(1,N);
V_audio = 2 + 0.2*sin(w_in*time);
w_vco1 = w_fr + K_vco*(V_audio-V_cfr);
%vco1
for k = 1:N-1;
phi_in(k+1) = phi_in(k) + T*w_vco1(k);
end;
V_ref= sin(phi_in);
%PLL
for k = 2:N-1;
%phase detector
ein(k)=V_ref(k)* V_out(k);
I(k)= (1-w_c*T)*I(k-1)+w_c*T*ein(k-1);
%loop filter
V_c(k)=(1-w_0*T)*V_c(k-1)+ I(k)-(1-w_1*T)*I(k-1);
%vco2
    w_vco2(k) = w_fr + K_vco*(V_c(k)-V_cfr);
phi_out(k+1) = phi_out(k) +w_vco2(k)*T;
V_out(k+1) = sin(phi_out(k+1));
end;
plot(time,V_c,time,V_audio);

```

A2: PLL (with Phase frequency detector) simulation

```
% VCO + BPF + PLL(digital PFD) Block

T = 1e-7;

T_end = 0.006; N = fix(T_end/T); time = T*[0:N-1];

%VCO

w_audio = 2*pi*1e3;

w_fr = 2*pi*1e5;

w_fr = 2*pi*1e6;

V_cfr = 2; K_vco = 2*pi*10e3;

%BPF

w_m = 2*pi*900e3;

a = -2*pi*5e3;

g = 2*pi*100e3;

u1 = zeros(1,N);

u2 = zeros(1,N);

x1 = zeros(1,N);

x2 = zeros(1,N);

V_fm = zeros(1,N);

phi_fm = zeros(1,N);

%PLL

R = zeros(1,N);
```

```

Q1 = zeros(1,N);
Q2 = zeros(1,N);
w_c = 2*pi*10e2;      %LPF
w_0 = 2*pi*1;        %Loop Filter
w_1 = 2*pi*1e2;      %Loop Filter
w_pll = w_fr - w_m;
phi_out = zeros(1,N);
w_vco2 = zeros(1,N);
V_c = 2*ones(1,N);
V_out = zeros(1,N);
ein = zeros(1,N);
I1 = zeros(1,N);
I = zeros(1,N);
%vco1
V_audio = 2 + 0.2*sin(w_audio*time);
w_vco1 = w_fr + K_vco*(V_audio-V_cfr);
for k = 1:N-1;
phi_fm(k+1) = phi_fm(k) + T*w_vco1(k);
end;
u = sin(phi_fm);
%BPF
for k = 1:N-1;
u1(k) = (cos(w_m*k*T)) * u(k);

```

```

u2(k)=(sin(w_m*k*T))*u(k);
x1(k+1)=(1+a*T)*x1(k)-g*T*x2(k)-T*a*u1(k);
x2(k+1)=g*T*x1(k)+(1+a*T)*x2(k)+T*a*u2(k);
V_fm(k+1)=x1(k+1)-x2(k+1);
end;

%PLL

%phase detector
V_ref = 0.5 + 0.5*sign(V_fm);
for k = 2:N-1;
% ein(k)=V_fm(k)* V_out(k);
% ein(k)=u(k)* V_out(k);
R(k) = Q1(k-1)*Q2(k-1);
if (( V_ref(k-1)== 0 ) && ( V_ref(k)== 1))
Q1(k)=1;
elseif ((R(k-1)==0) && (R(k)==1))
Q1(k)= 0;
else Q1(k) = Q1(k-1);
end;

if (( V_out(k-1)== 0 ) && ( V_out(k)== 1))
Q2(k)=1;
elseif ((R(k-1)==0) && (R(k)==1))
Q2(k)= 0;

```

```

else    Q2(k) = Q2(k-1);
end;

ein(k) = Q1(k)-Q2(k);

%LPF

% I(k) = (1-w_c*T)*I(k-1)+w_c*T*2*ein(k-1);
I1(k+1) = (1-w_c*T)*I1(k)+ein(k)*w_c*T;
I(k+1) = 2*pi*I1(k+1);

%loop filter

    V_c(k) = (1-w_0*T)*V_c(k-1) + I(k) - (1-w_1*T)*I(k-1);

%vco2

    % w_vco2(k) = w_fr + K_vco*(V_c(k)-V_cfr);
    w_vco2(k) = w_pll + K_vco*(V_c(k)-V_cfr);

phi_out(k+1) = phi_out(k) + w_vco2(k)*T;
V_out(k+1) = 0.5 + 0.5 * sign(sin(phi_out(k+1)));

end;

plot(time,V_audio,time,V_c)
ylabel('V_audio/Vc'); xlabel('time');

```

B : BPF Simulation MATLAB Code and Simulation Result(with $g(t) = g_1$)

```

% Band Pass Filter Block( No time-varying filter)

T = 50e-9;

T_end = 1e-3; N = fix(1+T_end/T); time = T*[0:N-1];

```

```

w_in = 2*pi*1e6;
w_m = 2*pi*900e3;
a = -2*pi*5e3;
g = 2*pi*100e3;
u=sin(w_in*time);
u1 = zeros(1,N);
u2 = zeros(1,N);
x1 = zeros(1,N);
x2 = zeros(1,N);
y = zeros(1,N);
for k = 1:N-1;
u1(k)=(cos(w_m*k*T))*u(k);
u2(k)=(sin(w_m*k*T))*u(k);
x1(k+1)=(1+a*T)*x1(k)-g*T*x2(k)-T*a*u1(k);
x2(k+1)= g*T*x1(k)+(1+a*T)*x2(k)+T*a*u2(k);
y(k+1)= x1(k+1)-x2(k+1);
end;
plot(time,y);
ylabel('y=x1-x2'); xlabel('time');

```

C : Whole Research (PLL FM Demodulator) Simulation MATLAB Code

% VCO + BPF + PLL(digital)+ Butterworth(LPF) Block

```

T = 0.5e-7;
T_end = 0.006; N = fix(T_end/T); time = T*[0:N-1];
%VCO
w_audio = 2*pi*1e3;
%w_fr = 2*pi*1e5;
w_fr = 2*pi*1e6;
V_cfr = 2; K_vco = 2*pi*10e3;
%vco1
V_audio = 2 + 0.2*sin(w_audio*time);
    w_vco1 = w_fr + K_vco*(V_audio-V_cfr);
phi_fm = zeros(1,N);
for k = 1:N-1;
phi_fm(k+1) = phi_fm(k) + T*w_vco1(k);
end;
w_noise = 2*pi*(1e6 + 2000);
% u= sin(phi_fm);
u= sin(phi_fm) + 0.01*(randn(1,N).*sin(w_fr*time) +
randn(1,N).*cos(w_fr*time));
    u1 = zeros(1,N);
    u2 = zeros(1,N);
    x1 = zeros(1,N);
    x2 = zeros(1,N);
V_fm = zeros(1,N);

```

```

V_cout = zeros(1,N);

R = zeros(1,N);
Q1 = zeros(1,N);
Q2 = zeros(1,N);
w_c = 2*pi*20e3;      %LPF
w_0 = 2*pi*1;        %Loop Filter
w_1 = 2*pi*1e3;      %Loop Filter make 8e3

phi_out = zeros(1,N);
w_vco2 = zeros(1,N);
V_c = 2*ones(1,N);
V_out = zeros(1,N);
ein = zeros(1,N);
I1 = zeros(1,N);
I = zeros(1,N);
w_m = 2*pi*900e3;
a = -2*pi*2e3;
c = 0.88;
g1 = 2*pi*100e3;
for k = 2:N-2;
    %BPF
    %g = w_audio*0.2*cos(w_audio*time)+ g1 ;

```



```

%g = K_vco*0.2*sin(w_audio*time)+ g1 ;
g(k) = K_vco*[c* (V_audio(k)-2)+(1-c)* V_cout(k)]+ g1 ;
u1(k)=(cos(w_m*k*T))*u(k);
u2(k)=(sin(w_m*k*T))*u(k);
x1(k)=(1+a*T)*x1(k-1)-g(k)*T*x2(k-1)-T*a*u1(k-1);
x2(k)= g(k)*T*x1(k-1)+(1+a*T)*x2(k-1)+T*a*u2(k-1);
V_fm(k)= x1(k)-x2(k);
%w_cout = 2*pi*3e3;
%PLL
%phase detector
V_ref(k) = 0.5 + 0.5*sign(V_fm(k));
% ein(k)=V_fm(k)* V_out(k);
% ein(k)=u(k)* V_out(k);
R(k) = Q1(k-1)*Q2(k-1);
if (( V_ref(k-1)== 0 ) && ( V_ref(k)== 1))
Q1(k)=1;
elseif ((R(k-1)==0) && (R(k)==1))
Q1(k)= 0;
else Q1(k) = Q1(k-1);
end;
if (( V_out(k-1)== 0 ) && ( V_out(k)== 1))
Q2(k)=1;
elseif ((R(k-1)==0) && (R(k)==1))

```

```

Q2(k) = 0;

else    Q2(k) = Q2(k-1);

end;

%  if (( V_fm(k-1)== 0 ) && ( V_fm(k)== 0))

%    Q1(k)=Q2(k);

% end;

ein(k) = Q1(k)-Q2(k);

%LPF

% I(k)= (1-w_c*T)*I(k-1)+w_c*T*2*ein(k-1);

I1(k)= (1-w_c*T)*I1(k-1)+ein(k-1)*w_c*T;

I(k) = 4*pi*I1(k);

%loop filter

V_c(k)=(1-w_0*T)*V_c(k-1)+ I(k)-(1-w_1*T)*I(k-1);

%vco2

w_pll = w_fr - w_m;

%w_vco2(k) = w_fr + K_vco*(V_c(k)-V_cfr);

w_vco2(k) = w_pll + K_vco*(V_c(k)-V_cfr);

phi_out(k+1) = phi_out(k) +w_vco2(k)*T;

V_out(k+1) = 0.5 + 0.5 * sign(sin(phi_out(k+1)));

% V_cout(k+1) = (1 - w_cout*T)*V_cout(k) + w_cout*T*V_c(k);

%Butterworth LBF(second order)

```

```

Q = 0.707;
w_cout = 2*pi*4e3;
    a1 = w_cout/Q;
    b1 = w_cout;
V_cout(k+2) = (2-a1*T)*V_cout(k+1) + [a1*T-(b1*T)^2-
1]*V_cout(k) + V_c(k)*[(b1*T)^2];
end;

x = [N/5:N]; time1 = time(x); V_audio1 = V_audio(x); V_c1
= V_c(x); V_cout1 = V_cout(x);

plot(time,V_audio,time,V_c)
ylabel('V_audio/Vc'); xlabel('time');
figure;
plot(time1,V_audio1, time1,V_cout1);
ylabel('V_audio/V_cout'); xlabel('time');

```

Vita

Shaohui Huang

Graduate student of ECE Department

Place and Date of Birth: March 15, 1965, Chongqing City, China

Names of Parents: Zhide Huang (Father)

Tongyucheng(Mother)

Education:

Lehigh University, Graduate student of ECE Department (2008-2011)

Northampton Community College, Bethlehem, PA(2007)

Wuhan University of Technology, China (1982-1986)

Bachelor of Engineering in Electronic Engineering

Professional Experience:

Inchcape Testing Services (ETL Testing Laboratories), China

Marketing Executive (1993 - 1999)

Cooperated with ETL Testing Lab and FCC, successfully introduced Chinese electrical products such as touch screens and electric tooth brushes into international business market.

Guangzhou Peugeot Automobile Co LTD, China, Electrical Engineer (1986 – 1993)

I used to work at localization of automobile electrical systems, such as turn signal lights, headlights, radio and speakers, batteries, windshield wipers, relays, and circuit breakers. Through research and testing, I successfully helped local electrical product manufacturers to improve their technology and quality.