Phase-Locked Loop using a comb filter with fractional delay

Umair N. Mughal; Robert Griñó

ACES: Control Avançat de Sistemes d'Energia

Phase Locked Loop using a comb filter with fractional delay

Umair N. Mughal; Robert Griñó

May 2011

Abstract

A Phase Locked Loop is a feedback system combining a Voltage Controlled Oscillator and a Phase Comparator These are connected so that the oscillator maintains a constant phase angle relative to a reference signal. Phase locked loops can be used, for example to generate stable output frequency signals from a fixed frequency signal.
A Comb Filter is a kind of Notch Filter (Non Recursive Filter) that is normally used to remove the harmonic terms from a particular signal.

In this Design, a PLL is implemented using a Comb Filter because the main frequencies that are considered noise in the PLL loop are the harmonics of the fundamental frequency.
There are 2 objectives of this project report,

1. Implementing Comb Filter Using Fractional Delay

2. Comparision of Frame Based and Typical Discrete Sample Based Implementation of a Comb Filter

Contents

List of Figures

List of Tables

Chapter 1

STUDY of Phase Locked Loop and Comb Filter

1.1 Phase Locked Loop

1.1.1 Introduction

The phase locked loops have many different applications and it comes to support from the *heritage of control and vibration theory where they are used to describe free body behavior of mechanical systems.* Typically in communication systems PLL are broad application spectrum as like,

1. Carrier synchronization

2. Carrier Recovery

3. Frequency Division and multiplication

4. Demodulation

The PLL have generally three classes,

1. The linear or analog PLL (LPLL)

2. The digital PLL (DPLL)

3. All digital PLL (ADPLL)

The conceptual basis behind each of these is the same and they are all specified by the same standard parameters such as loop bandwidth, damping factor etc.
Also a PLL has three core components, they are,

1. Phase Detector (PD) or the multiplier

2. The loop Filter (LF)

3. Voltage Controlled Oscillator or VCO

1.1.2 Mathematical Explanation

Now lets look into the mathematical describtion of all the core components of PLL.

Phase Detector

In simpler terms, the phase detector is a multiplier. Lets start with two sinosoids, $s_1(t)$ and $s_2(t)$ Both have same frequency but are phase shifted by 90°. Now the multiple of these signals,
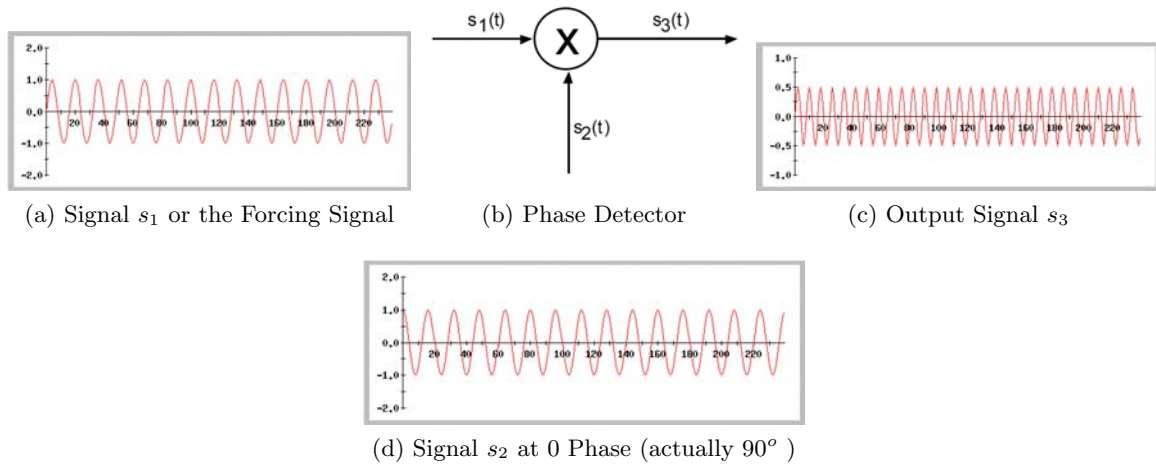
$$s_3(t) = s_1(t)s_2(t) \tag{1.1}$$

(a) Signal s_1 or the Forcing Signal   (b) Phase Detector   (c) Output Signal s_3

(d) Signal s_2 at 0 Phase (actually 90°)

Figure 1.1: Phase Detector and its Signal Sequences

$$s_1(t) = A_1 \sin(\Omega t + \phi_1(t)) \tag{1.2}$$

$$s_2(t) = A_2 \cos(\Omega t + \phi_2(t)) \tag{1.3}$$

The output of the multiplier is

$$s_3(t) = K_d A_1 A_2 \sin(\Omega t + \phi_1(t)) \cos(\omega t + \phi_2(t)) \tag{1.4}$$

where K_d is the gain of the multiplier. With a little trignometric manipulation, we can put this equation in a form which is far more illuminating,

$$s_3(t) = \frac{K_d A_1 A_2}{2} \sin(\phi_1(t) - \phi_2(t)) + \frac{K_d A_1 A_2}{2} \sin(2\Omega t + \phi_1(t) + \phi_2(t)) \tag{1.5}$$

In this form, we see that the multiplier signal consists of two parts, the first one is function of only the phase difference of the two signals, and the second term is function of frequency and phase difference but this frequency is twice the signal frequency (note the $2\omega t$ term) plus the sum of the two phases.

We can use the equation 1.5 to develop the PLL by recognizing that the output signal of the multiplier is a *function of the phase difference of the two input signals*. This is useful information and we can use it to synchronize the two signals. The second part of this equation at twice the frequency is a harmonic term which needs to be filtered out since it does not offer anything we need. The effect of higher order harmonics in $s_1(t)$ will introduce same in $s_3(t)$ hence it develops the need of a harmonic filter.

Loop Filter

Now we need to add a filter at the output of the multiplier signal. It is required that its bandwidth should be quite smaller so that it knocks out noise and unwanted harmonic terms. When there is no
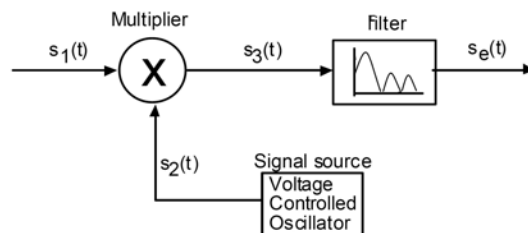

Figure 1.2: Loop Filter

phase difference (we are starting with a 90° phase difference which is required to make this whole thing

work), then the signal out of the filter is just the first part of equation 1.5. We call this part the error signal (also called the control signal).

$$s_e(t) = \frac{K_d A_1 A_2}{2} \sin(\phi_1(t) - \phi_2(t))$$
$$s_e(t) = \frac{K_d A_1 A_2}{2} (\phi_1(t) - \phi_2(t)) \qquad where \ \sin(\theta_{small}) \simeq \theta_{small} \tag{1.6}$$

If a phase difference is 0° (actually 90°) then we would expect the signal to be zero, which is the desired and the locked state of the PLL. It the phase between the two signals $s_1(t)$ and $s_2(t)$ varies from that, then we would expect the filtered $s_3(t)$ signal to change as we will see in the implementation part.

The Filter that is used in this Design is Comb Filter which will be discussed in detail in the subsequent section.

Voltage Controlled Oscillator

The error signal provides us an indication that what is happening to the input phase. We want the error signal to have zero amplitude and we can do that only by changing the phase of signal $s_2(t)$ to match the phase of signal $s_1(t)$. VCO which we use to produce the signal allows us to do that.

It produces a periodic signal, the frequency of which changes based on the control signal applied externally. If the error signal is zero, then the VCO produces just its quiscent frequency (center frequency). But if the error signal is something other than zero, then it responds by changing its operating frequency. A constant of K_o represents the sensitivity of the VCO. It represents the change
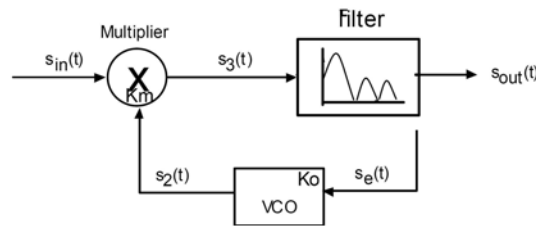

Figure 1.3: PLL with all major 3 components

in the instantaneous frequency of the VCO as a fucntion of the error signal amplitude such that

$$K_o = \frac{d\Omega_i}{dv} \tag{1.7}$$

The signal out of the VCO is given by

$$s_2(t) = A_2 \cos(\Omega_c(t) + \phi_2(t)) \tag{1.8}$$

The units of K_o are hertz per volts. Given a certain input voltage, it will produce a change in the output signal frequency by the following relationship

$$\Omega_{out} = \Omega_c(t) + K_o v(t) \tag{1.9}$$

Where Ω_c is the center or operating frequency. Also we know that for some periodic signal $p(t)$, its frequency in Hz is equal to the rate of change of phase of 2π segments, or

$$F_i(t) = \frac{1}{2\pi} \frac{d\phi_1(t)}{dt} \tag{1.10}$$

and conversely, phase is the integral of the frequency over a certain period of time.

$$\phi_i(t) = 2\pi \int_0^t F_i(t)dt \tag{1.11}$$

These relationships apply to all periodic signals, even those that are non sinosoidal

$$p(t) = \cos(2\pi F t) = \cos(2\pi \int_0^t F_i(t)dt) \tag{1.12}$$

We can now write the phase of the feed back signal as,

$$\phi_2(t) = 2\pi K_o \int_0^t s_e(t)dt = 2\pi K_o s_e(t)t \tag{1.13}$$

So as long as the error signal has a non zero amplitude, the phase of the VCO signal will keep on increasing until such time as it is decreased to zero. Now substituting equation 1.13 into equation 1.6 we get

$$s_e(t) = \frac{K_m A_1(t)A_2(t)}{2} \sin(\phi_1(t) - \phi_2(t)) \tag{1.14}$$

$$s_e(t) = \frac{K_m A_1(t)A_2(t)}{2} sin(\phi_1(t) - 2\pi K_o \int_0^t s_e(t)dt) \tag{1.15}$$

The equation of s_e for very small θ can be linearised by making the following assumption,

$$\sin(\phi_1(t) - \phi_2(t)) = \phi_1(t) - \phi_2(t) \tag{1.16}$$

Therefore 1.15 can be written as

$$s_e(t) = \frac{K_m A_1(t)A_2(t)}{2}(\phi_1(t) - 2\pi K_o s_e(t)t) \tag{1.17}$$

Now in the conclusion we can say that as long as the error signal is present, the phase keeps on changing linearly. However, as the phase of the signal out of the VCO changes, the new difference in phase decreases and the error signal decreases at the next go around. This decreases the phase change further untill the error signal amplitude has gone to zero.
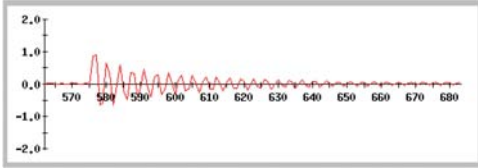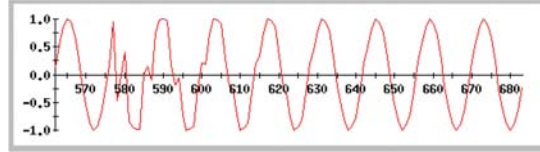
(a) Error Signal (b) Output Signal of VCO

Figure 1.4: Error Signal and Resulting Signal Out of VCO

1.2 Comb Filter

1.2.1 Introduction

A Comb Filter is a kind of Notch Filter. Before studying Comb Filter it is important to consider few properties of FIR and Notch Filters.

- FIR Filter

  1. Stability[7]

     An important property of FIR Filter is that it is always stable because stability requires that there should be no poles outside the unit circle and hence this filter has no poles.

  2. Phase[7]

     (a) Another property of non recursive filter is that we make filters with exactly linear phase. Note that the ability to have an exactly linear phase response is one of the most properties of the LTI System (Filter).

     (b) When a signal passes through a filter, it is modified in amplitude and/or phase.

     (c) The nature and extent of the modification of the signal is dependent on the amplitude and phase characteristics of the filter.

     (d) A filter is said to have a linear phase response if its phase satisfies one of the following relationships

     $$\phi(\omega) = -a\omega \qquad (1.18)$$

     $$\phi(\omega) = b - a\omega \qquad (1.19)$$

     Where a and b are constants.

  3. Phase Jump[7]

     It is also important to understand the phase jumps,

     (a) A jump of $\pm 2\pi$ occurs to maintain the phase function within the principal range of $[+\pi$ and $-\pi]$

     (b) A jump of $\pm\pi$ occurs when $H(\omega)$ undergoes a change of sign

- Notch Filter

  1. When a zero is placed at a given point on the unit circle, the frequency response will be zero gain at the corresponding point. A pole on the other hand produces a infinte peak at the corresponding frequency point.

  2. Poles that are close to the unit circle give rise to large peaks, whereas zeros close to or on the unit circles produces troughs of minima. Thus by strategically placing poles and zeros on the z plane, we can obtain sample low pass or other frequency selective filters (notch filters).

  3. Now if we want to remove a particular frequency we need to have an angle and radius to place a pole on the unit circle, e.g. to remove F_1 such that the $3db$ width of notch is δF and the sampling frequency is F_s then we have

$$r = 1 - \pi \frac{\delta F}{F_s} \qquad (1.20)$$

and the angle to place the pole and is determined as,

$$\theta = 360^o \frac{F_1}{F_s} \qquad (1.21)$$

1.2.2 Comb Filter

Comb Filter using Integer Delay

In our case the comb filter has only two nonzero values in its impulse response. A single delayed version of an input sequince is added to or subtracted from the undelayed version of the input sequence to form the output. For the case of a single sample of delay, the impulse response would be $[1, 1]$ when the delayed signal is added to the original, or $[1, -1]$ when it is subtracted. For two samples of delay, it would be $[1, 0, 1]$ or $[1, 0, -1]$ and so forth for different delays. The simple FIRs are comb filters which may also be characterised, respectively, as lowpass, highpass, bandpass and notch filters. When the second non zero value in the impulse responses does not have unity magnitude, the null depth does not go to zero amplitude. e.g. $[1, 0.9], [1, 0, -0.7]$ etc. Comb filters are useful in certain types of applications. Suppose, for example that you had an audio signal polluted with a 60 Hz fundamental wave with very high harmonic amplitudes extending into 10^4 or higher frequency range. A Comb filter is ideal for suppressing such a harmonic series, exhibiting economny and simplicity[5]. Now as we have the harmonic term in our PLL which we should remove therefore the delay of Comb Filter need to designed in such away that it produces notches to remove those harmonic terms. A typical comb filter will have function

$$H(z) = 1 - z^{-D} \tag{1.22}$$

Now we also know that

$$z = e^{J\omega} \tag{1.23}$$

where

$$\omega = 2\pi \frac{F_a}{F_s} \tag{1.24}$$

Therefore

$$H(z) = 1 - z^{-D} = 1 - e^{-J2\pi D \frac{F_a}{F_s}} \tag{1.25}$$

$$H(z) = 2_J e^{(-J\pi D \frac{F_a}{F_s})} \frac{e^{J\pi D \frac{F_a}{F_s}} - e^{-J\pi D \frac{F_a}{F_s}}}{2_J} \tag{1.26}$$

$$H(z) = 2 e^{J(\frac{\pi}{2} - \pi D \frac{F_a}{F_s})} (2 \sin(\pi D \frac{F_a}{F_s})) \tag{1.27}$$

Hence

$$|H(\omega)| = 2 \sin(\pi D \frac{F_a}{F_s}) \tag{1.28}$$

Now our demand is that $2 \sin(\pi D \frac{F_a}{F_s}) = 0$, which therefore gives,

$$D = \frac{F_s}{F_a} \tag{1.29}$$

Also from equation A.3

$$\angle H(\omega) = \frac{\pi}{2} - \pi D \frac{F_a}{F_s} \tag{1.30}$$

It is therefore clear that it has linear phase with $b = \frac{\pi}{2}$ and $a = \frac{D}{2} = \frac{F_s}{2F_a}$
But we also need to have a pole at $\theta = 0$ so that we at least allow the phase to pass, therefore our new transfer function becomes

$$H(z) = \frac{1 - z^{-D}}{1 - z^{-1}} \tag{1.31}$$

Here we are considering that $\frac{F_s}{F_a}$ will always give us an integer quantity but this may not be possible due to the noise/variation in f_a. Hence for a good filter design we need to consider the fractional variations in delays. The subsequent section is dedicated to this issue.

Comb Filter using Fractional Delay

In the previous section we considered a Comb Filter with $D = D_I$ which is an ideal case. Practically $D = D_I + D_f$ where D_f is the fractional delay element. Hence in this section we will only consider D_f as D_I is already covered[4].

Fractional delay filter is a device which has find application in numerous fields as like signal processing, including communiations, array processing, speech processing, etc. As we already know that one fundamental advantage of DSP Technology over traditional analog methods is the easy implementation of a constant delay: the signal samples are simply stored in buffer memory for a given time. This technique works perfectly as long as the desired delay is a multiple of the used sample interval[4].

The fractional delay can be viewed a time domain interpolation problem, often leading to a modification of standard sampling rate conversion maethods, or to the use of the traditional Lagrange interpolation technique. The problem D_f can be solved by viewing a delay as a resampling process. The desired solution can obtained by first reconstructing the continuous bandlimited signal and then resampling it after shifting. The task is thus related to interpolation in multirate filter design techniques or sampling rate conversion in general[4]. There has already been a lot of research done on these filters and many approximated fractional delay filters are found in literature. e.g.

1. Fractional Delay Approximation Using FIR Filters

    (a) Least Squared Error Solutions for FIR Filters
        i. Direct Least Squared Integral Error FIR Design
        ii. Windowing Methods for FIR Filter Design
        iii. FIR Filter Design with Smooth Transition Band Function
        iv. General Least Squares FIR Approximation of a Complex Frequency Response
        v. Least Squares FIR Design on a Discrete Frequency Grid
        vi. Stochastic Least Mean Squared Error, FIR Interpolation

    (b) Maximally Flat FIR FD Filter Design
        i. Lagrange Interpolation

    (c) Minimax Design of FIR Fractional Delay Filters

    (d) Controlling the Delay of Arbitrary FIR Filters
        i. Forier Transform Based Methods
        ii. Matrix Transform Method Based on Zeros of the Error Function
        iii. Farrow Structure for Fractional Delay FIR Filters

2. Fractional Delay Approximation Using Allpass Filters

    (a) Least Square Design of All Pass Filters
        i. Approximate LS PHase Error Design
        ii. LS Phase Delay Error Design of All Pass Filters

    (b) Maximally Flate Group Delay Design of All Pass Filters

    (c) Minimax or Equiripple Design of All Pass Filters
        i. Equiripple Phase Error Design
        ii. Equiripple Phase Delay Error Design

    (d) Controlling the Delay of All Pass Filters
        i. Matrix Transform Method Based on Zeros of the Error Function
        ii. Recursive Farrow Structure for All Pass Filters

In this project report only two fractional delay filters are considered for implementation, due to their simplicity. The two approximated filters are *Lagrange Interpolation Filter (Maximally Flat FIR Fractional Delay Filter)* and *Thiran Filter (Maximally Flat Group Delay All Pass Fractional Delay Filter)*. These two filters are safe solutions in most cases when one is not quite sure about how to state the magnitude and phase delay constrains as in the case of noise.

- Lagrange Interpolation Filter

    It is an easiest way to approximate fractional delay FIR filter. Its co-efficients are adopted from[4],

    $$h(n) = \prod_{\substack{k=0 \\ k \neq n}}^{N} \frac{D-k}{n-k} \quad for \ \ n = 0, 1, 2 ... N \tag{1.32}$$

    where N is the order of the filter. The coefficients for the Lagrange FD Filters of length L = 2,3 and 4 are given in the following table.

Table : Coefficients of the Lagrange FD Filters of Order N = 1,2 and 3			
	$h(0)$	$h(1)$	$h(2)$	$h(3)$
$N=1$	$(1-D)$	D	-	-
$N=2$	$\frac{(D-1)(D-2)}{2}$	$-D(D-2)$	$\frac{D(D-1)}{2}$	-
$N=3$	$-\frac{(D-1)(D-2)(D-3)}{6}$	$D\frac{(D-2)(D-3)}{2}$	$-\frac{D(D-1)(D-3)}{2}$	$\frac{D(D-1)(D-3)}{6}$

Table 1.1: Coefficients of the Lagrange FD Filters of Order N = 1,2 and 3

$$H_1(z) = (1-D) + Dz^{-1}$$
$$H_2(z) = \left(\frac{(D-1)(D-2)}{2}\right) + \left(\frac{(-D)(D-2)}{1}\right)z^{-1} + \left(\frac{(D)(D-1)}{2}\right)z^{-2}$$
$$H_3(z) = \left(\frac{-D(D-1)(D-2)(D-3)}{2}\right) + \left(\frac{D(D-2)(D-3)}{2}\right)z^{-1} + \left(\frac{-D(D-1)(D-3)}{2}\right)z^{-2} + \left(\frac{D(D-1)(D-2)}{6}\right)z^{-3} \tag{1.33}$$

- Thiran Filter

    The thiran method can be viewed as a recursive counterpart of Lagrange interpolation. It is the simplest of all All Pass Filter Approximations. It is characterised by maximally flat group delay at the zero frequency. Its coefficients are obtained from the following equation[4]

    $$a_k = (-1) \begin{pmatrix} N \\ k \end{pmatrix} \prod_{N=0}^{N} \frac{D-N+n}{D-N+k+n} \quad for \ \ k = 0, 1, 2 ... N \tag{1.34}$$

    where N is the order of the All Pass Filter. The denominator coefficients of low order all pass transfer function (N=1,2 and 3) are given in the following table.

Table : Coefficients of the Thiran FD All Pass Filters of Order N = 1,2 and 3			
	a_1	a_2	a_3
$N=1$	$(1-D)(1+D)$	-	-
$N=2$	$-2(D-2)(D+1)$	$\frac{(D-1)(D-2)}{(D+1)(D+2)}$	-
$N=3$	$-3\frac{(D-3)}{(D+1)}$	$3\frac{(D-2)(D-3)}{(D+1)(D+2)}$	$-\frac{(D-1)(D-2)(D-3)}{(D+1)(D+2)(D+3)}$

Table 1.2: Coefficients of the Thiran FD All Pass Filters of Order N = 1,2 and 3

$$H_1(z) = \frac{(1-D)(1+D)+z^{-1}}{1+(1-D)(1+D)z^{-1}}$$
$$H_2(z) = \frac{\left(\frac{(D-1)(D+2)}{(D+1)(D+2)}\right)+\left(-\frac{2(D-2)}{(D+1)}\right)z^{-1}+z^{-2}}{1+\left(-\frac{2(D-2)}{(D+1)z^{-1}}\right)z^{-1}+\left(\frac{(D-1)(D+2)}{(D+1)(D+2)}\right)z^{-2}}$$
$$H_3(z) = \frac{\left(-\frac{(D-1)(D-2)(D-3)}{(D+1)(D+2)(D+3)}\right)+\left(\frac{3(D-2)(D-3)}{(D+1)(D+2)}\right)z^{-1}+\left(-\frac{3(D-3)}{(D+1)}\right)z^{-2}+z^{-3}}{1+\left(-\frac{3(D-3)}{(D+1)}\right)z^{-1}+\left(\frac{3(D-2)(D-3)}{(D+1)(D+2)}\right)z^{-2}+\left(-\frac{(D-1)(D-2)(D-3)}{(D+1)(D+2)(D+3)}\right)z^{-3}} \tag{1.35}$$

Appendix A

Phase Locked Loop Blocks in Simulink

Different people have approached in different ways to design PLL, but the main components of the loop are typically same.

I was assigned the task to implement this PLL using Comb Filter in Simulink and using Sample Based Technique. Therefore I decided to start with a simple analogue sinosoidal source which was operating on Fundamental Frequency of 50 Hz (Grid Frequency). Keeping in mind the Nyquist Criteria, a sampling frequency of 1510^3 Hz was selected. The complete PLL Loop can be seen in the figure and Now starting
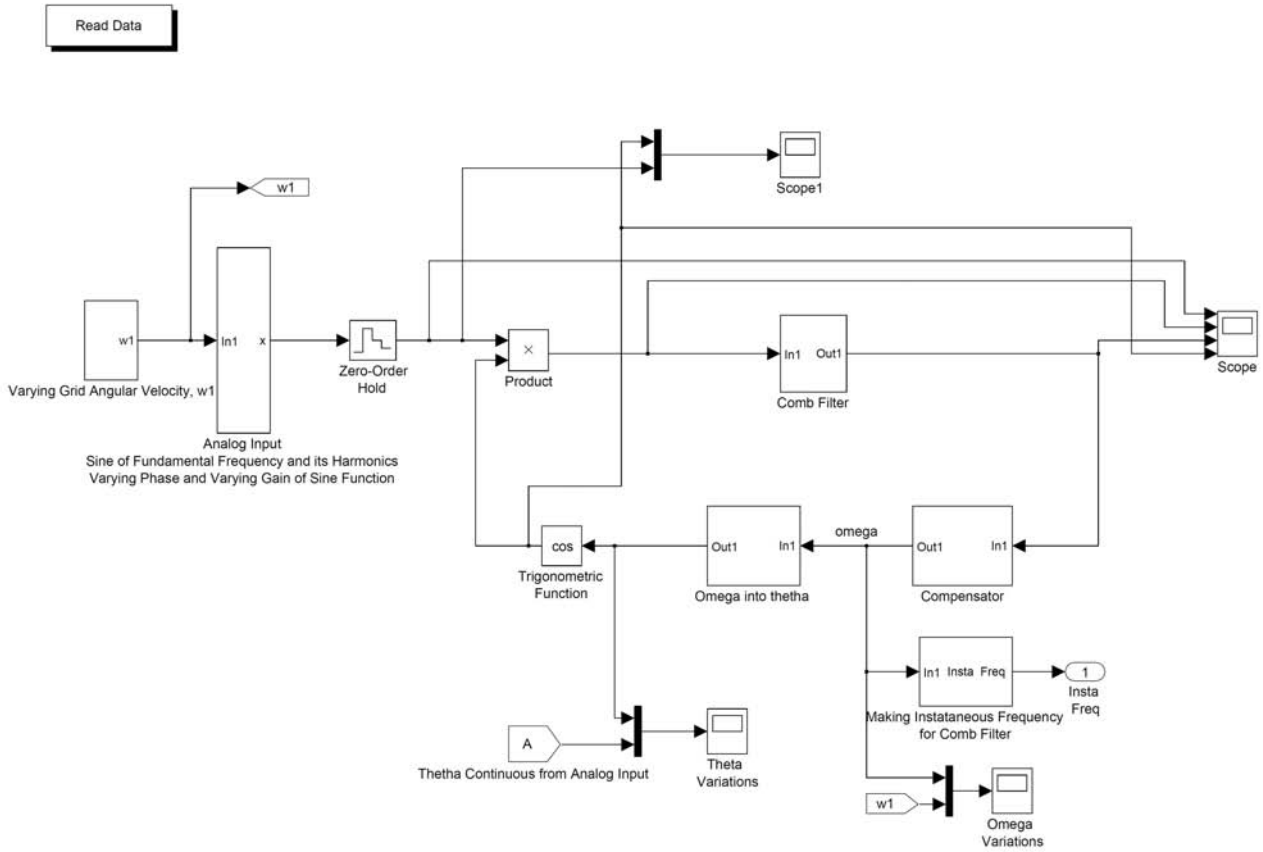
Figure A.1: Complete PLL Loop

from the Grid Frequency Variations, the input sequency of grid frequency is assumed as follow,

$$\omega(t) = \begin{cases} 0.3 & \frac{\text{rad}}{\text{s}} & 0.1 \text{ s} \leq \text{ t} \\ -0.6 & \frac{\text{rad}}{\text{s}} & 0.4 \text{ s} \leq \text{ t} \\ 0.3 & \frac{\text{rad}}{\text{s}} & 0.6 \text{ s} \leq \text{ t} \\ 2\pi50 & \frac{\text{rad}}{\text{s}} & 0.0 \text{ s} \leq \text{ t} \end{cases} \qquad (A.1)$$

This signal is then passed through the analog input sequence block which contain the fundamental
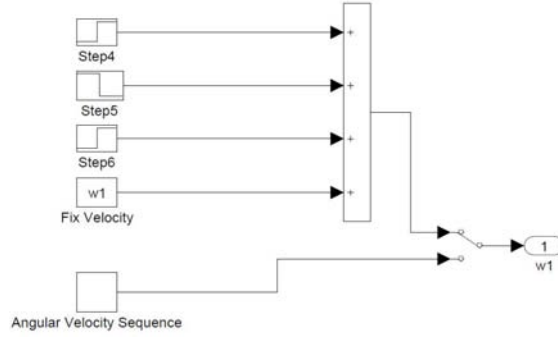
Figure A.2: Variations in Grid Velocity

forcing function alongwith its 4 harmonics and White Noise. The amplitude of white noise is assumed to be 0.01 and its sampling time is assumed to be 10^{-5} This block is shown in the following figure, Now
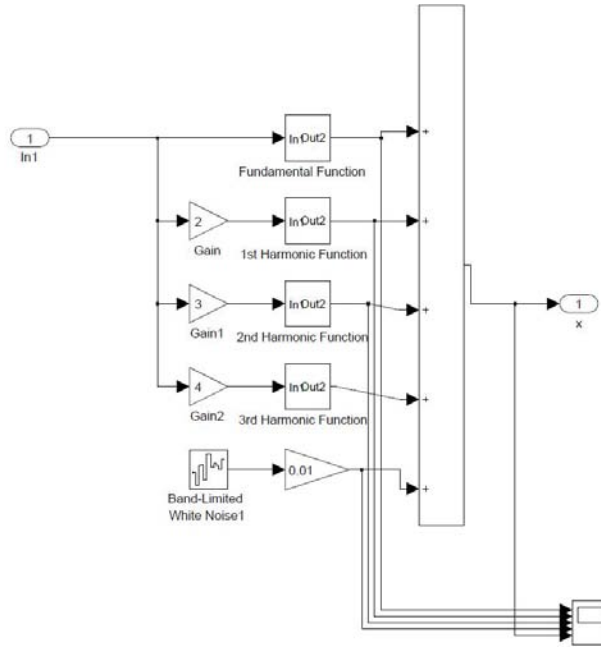
Figure A.3: Sequence of s_1 *Forcing Signal*

the fundamental forcing function also contain the variation in phase and amplitude. These variations are incorporated in the loop with so that the designed PLL should be able to behave properly at maximum variations. These blocks are given as, Also assumed ϕ variations from $s_1(t) = A\sin(\omega t + \phi)$ are,

$$\phi(t) = \begin{cases} \frac{\pi}{6} & \text{rad} & 0.3\,\text{s} \leq \text{ t} \\ \frac{\pi}{3} & \text{rad} & 0\,\text{s} \leq \text{t} \\ \frac{-\pi}{6} & \text{rad} & 0.5\,\text{s} \leq \text{ t} \end{cases} \tag{A.2}$$

And assumed $A(Amplitude)$ variations from $s_1(t) = A\sin(\omega t + \phi)$ are,

$$A(t) = \begin{cases} 1 & 0.3\,\text{s} \leq \text{ t} \\ 1 & 0\,\text{s} \leq \text{ t} \\ -1 & 0.5\,\text{s} \leq \text{ t} \end{cases} \tag{A.3}$$

The signal after after passing from these 2 blocks is then sampled using Zero Order Hold Block. After this the signal is then supposed to be filtered to remove all the harmonic terms using Comb Filter and Fractional Delay. The inside of these Comb Filter and Different Type of Fractional Delay Filter are shown in the following figures.   The signal after passing through the Filter is then passed through the Compensator for the Stabililization. A Discrete PI Compensator is choosen for this purpose whose
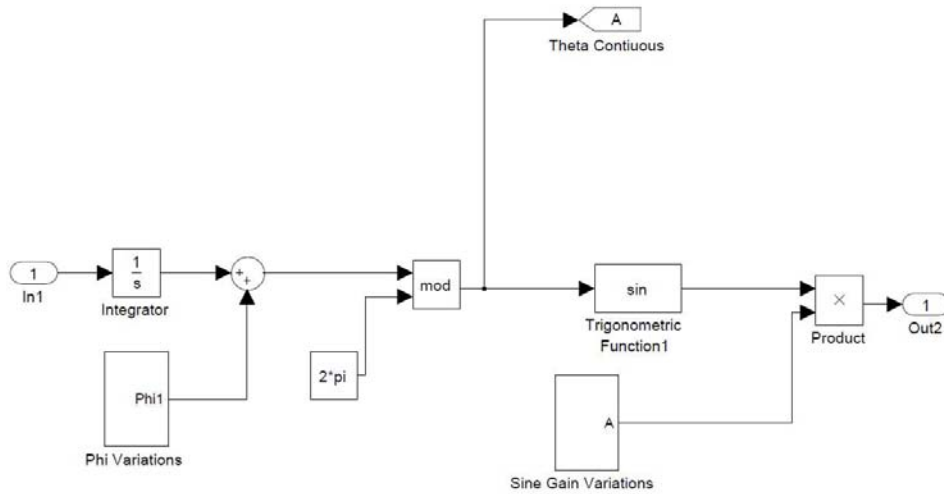
Figure A.4: Fundamental Forcing Function s_1

(a) Variations in the Gain of s_1 emphForcing Signal

(b) Varying Phase in the Fundamental Forcing Funtion
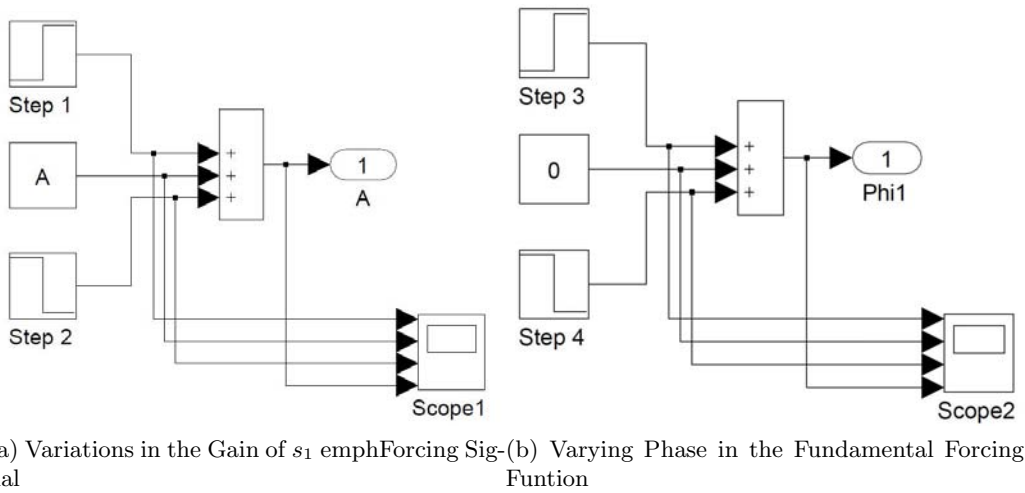
Figure A.5: Variation in Phase and Amplitude of Forcing Function

gains are adjusted as $K_p = 120$ and $K_i = 2800$. The inside of this compensator is given as, The output from this compensator takes 2 ways,

1. An indirect way is to go to a low pass filter, so that this output again goes into the Comb Filter for stability of Phase Locked Loop.

2. The direct way is to go to an integrator, the output of which is θ which then goes into VCO and hence into the Phase Detector.
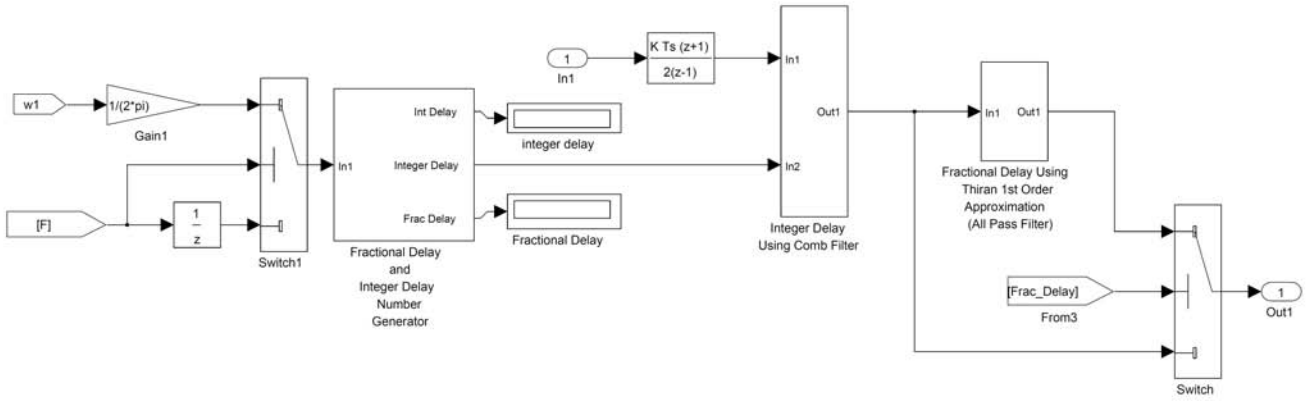
Figure A.6: Complete Loop of PLL Filter
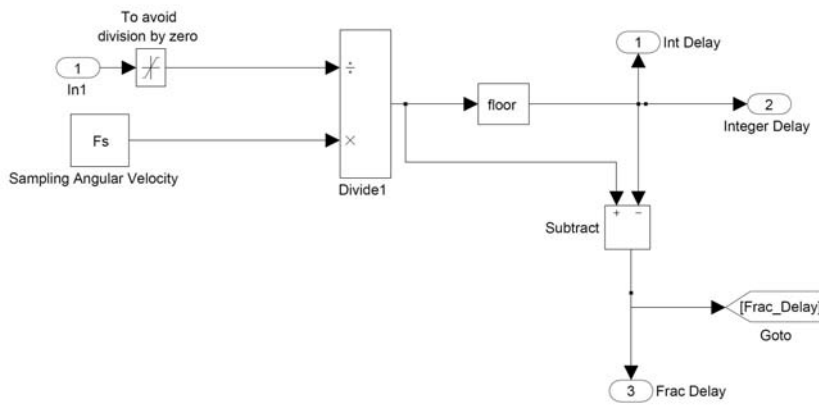
Figure A.7: Finding Integer and Fractions to be used in Comb Filter and Approximated Fractional Delay Loops Respectively
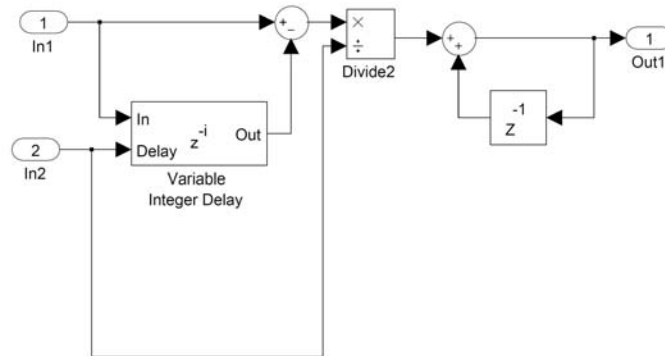
Figure A.8: Comb Filter

16

(a) Length 1

(b) Length 2
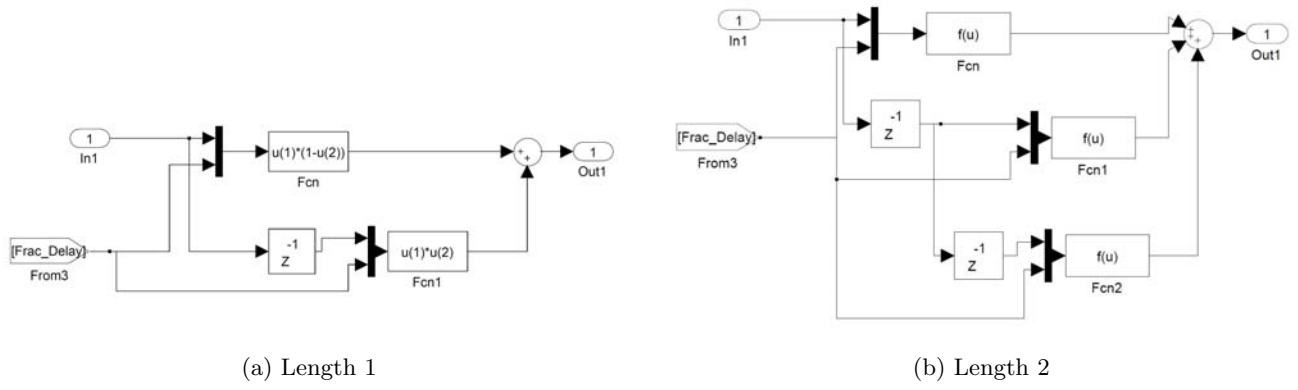
Figure A.9: Fractional Delay Loop of Lagrangian Approximators

(a) Order 1

(b) Order 2
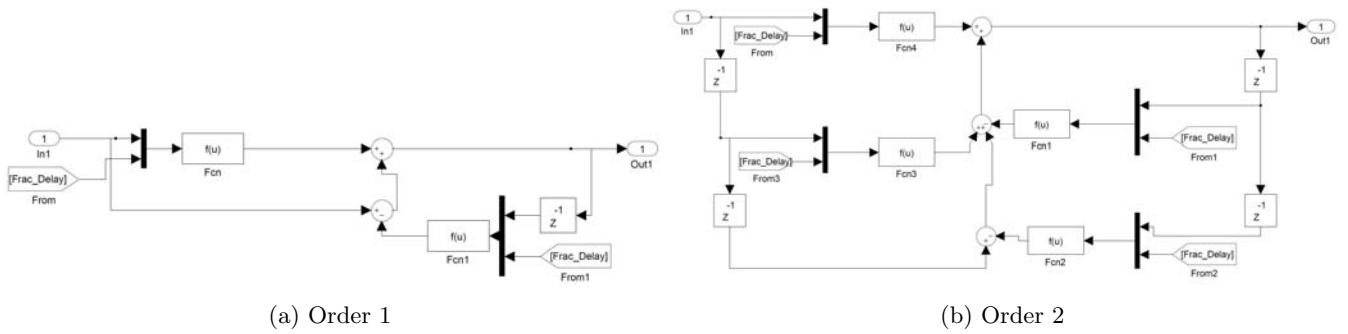
Figure A.10: Fractional Delay Loop of Thiran Approximators
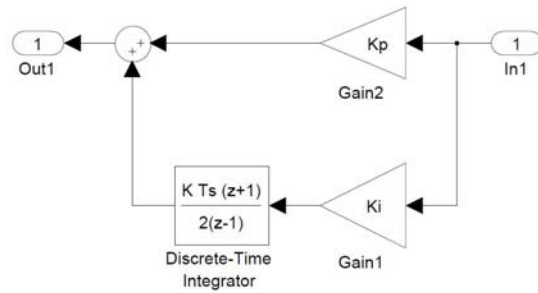
Figure A.11: Discrete Compensator

(a) Filtering High Frequency Noise

(b) Discrete Transformation omega into theta
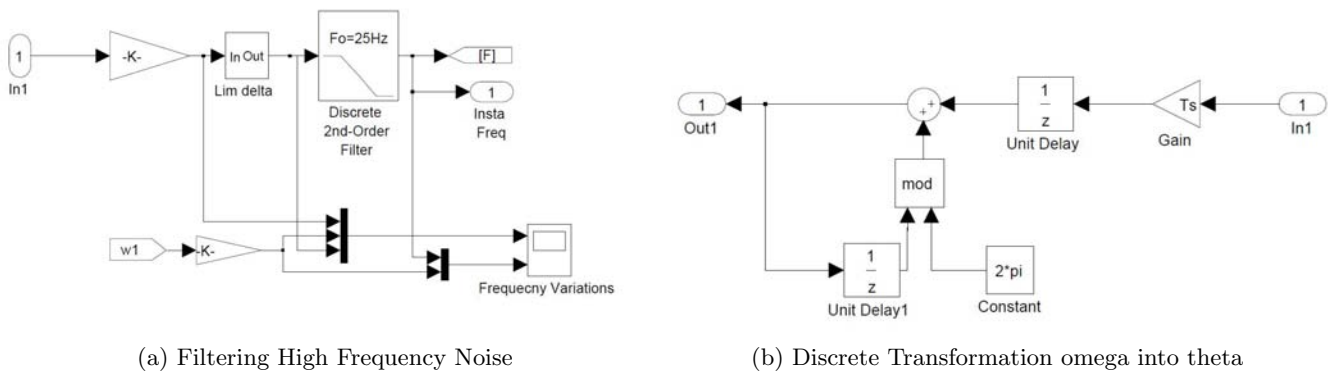
Figure A.12: Output Directions of Compensator

Appendix B

Phase Locked Loop

Now the most important responses of this loop are given below. Also the phase characteristics of different order of Lagrangian and Thiran Approximators were compared with the original phase of fractional delay. It is observed that,

1. Thiran 1st order Approximator matches best with the Original Phase of Fractional Delay.

2. It is also observed from these results that as we are increasing the Fractional Part towards unity, the phase of all approximators are matching closely with the original.

There results are also shown in the corresponding figures,
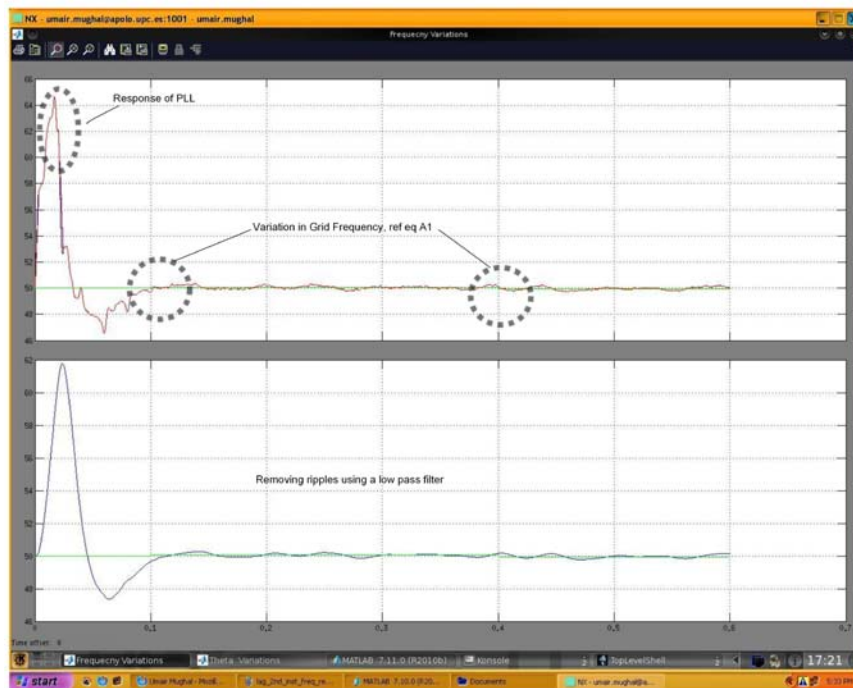
Figure B.1: Frequency Stabilization Using Comb Filter (without any Fractional Delay Filter) in the PLL Loop only by Varying Grid Frequency of s_1
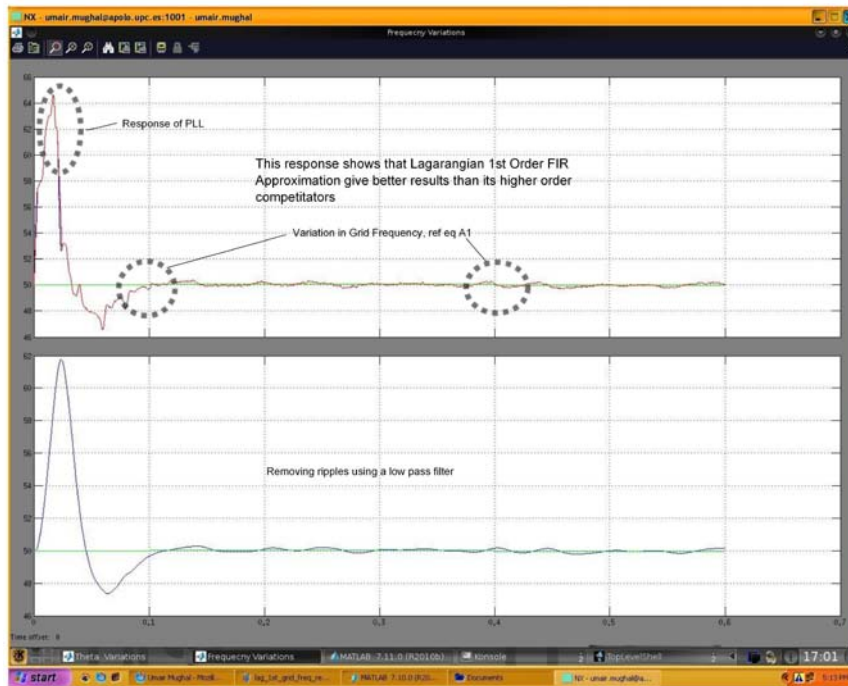
Figure B.2: Frequency Stabilization Using Comb Filter and Lagrangian Approximator by Varying Frequency

Figure B.3: Frequency Stabilization Using Comb Filter and Lagrangian, Length 1 Approximator by Varying Phase and Gain of s_1
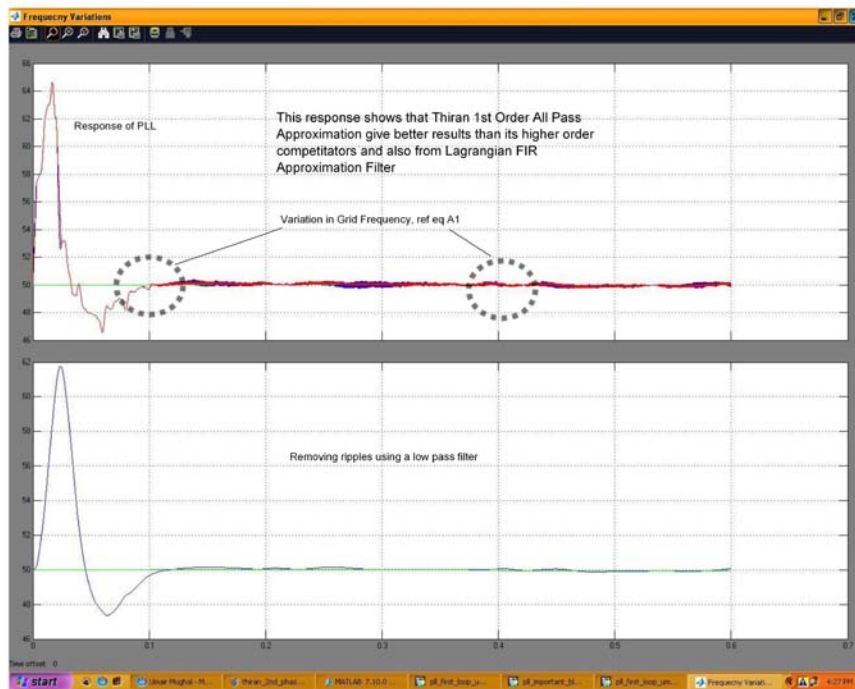
Figure B.4: Frequency Stabilization Using Comb Filter and Thiran, Order 1 Approximator by Varying Grid Frequency of s_1
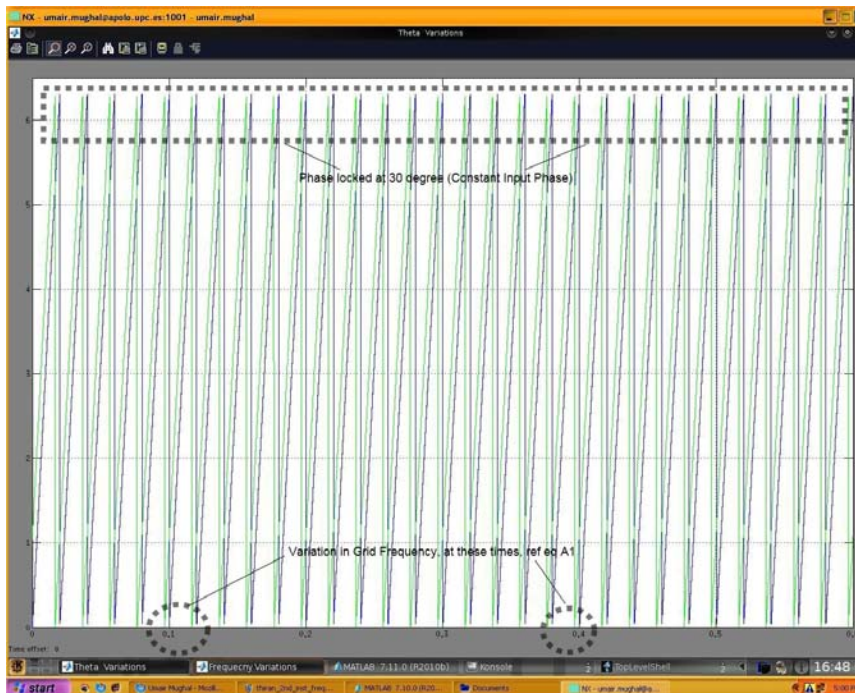

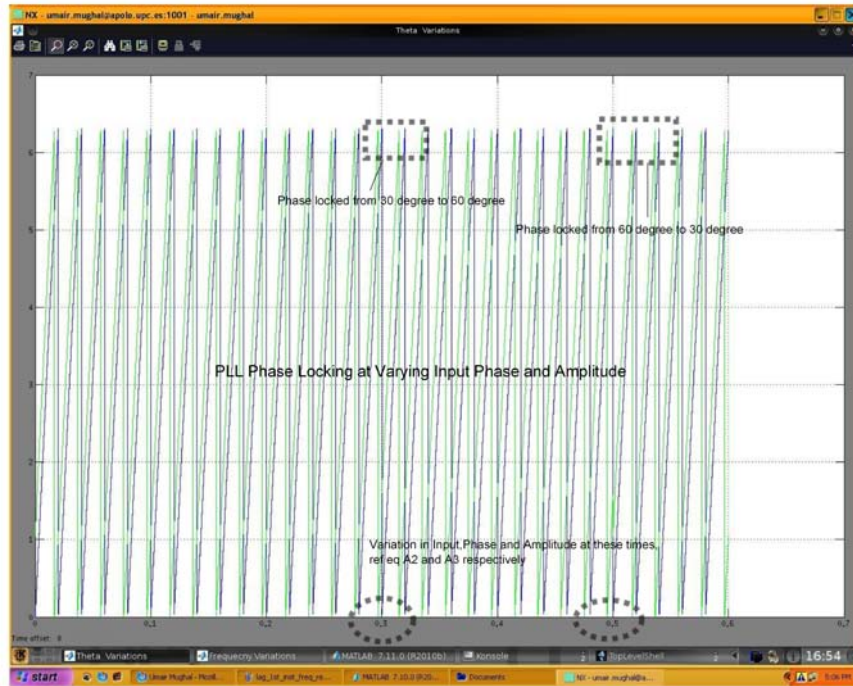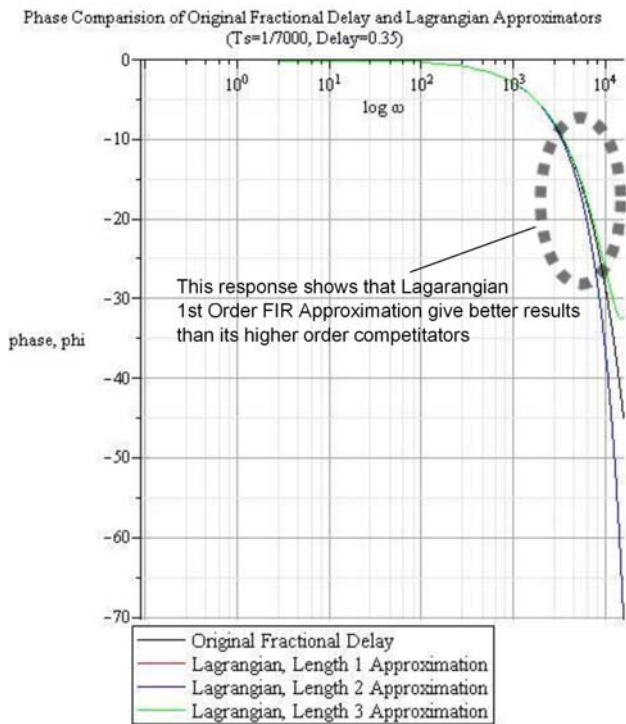Figure B.5: Phase Locking in the PLL Loop by Varying the Grid Frequency but keeping the Phase Constant

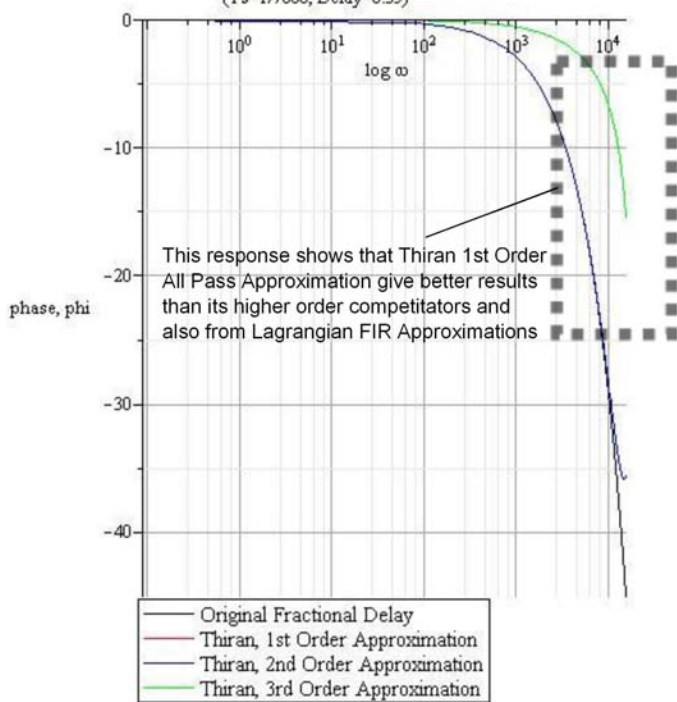Figure B.6: Phase Locking in the PLL Loop by Varying the Phase but keeping the Grid Frequency Constant

(a) D = 0.35
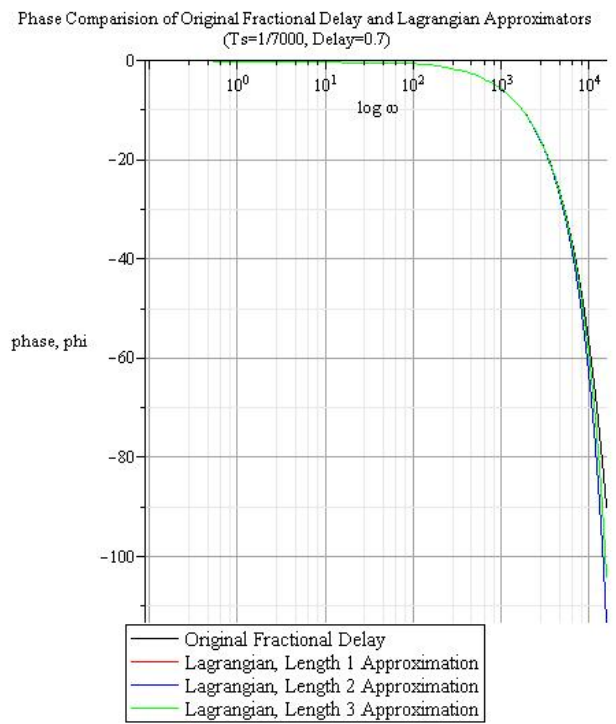
(b) D = 0.70

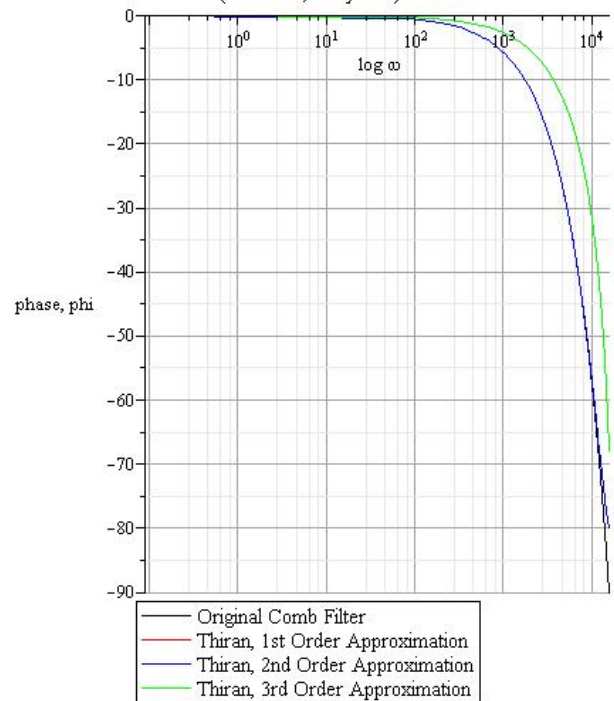Figure B.7: Phase Comparision of Original Phase Delay and Lagrangian Approximators using different Delays, D

(a) D = 0.35

(b) D = 0.70

Figure B.8: Phase Comparision of Original Phase Delay and Thiran Approximators using different Delays, D

Appendix C

Comb Filter Frame Based Implementation in Simulink

As it is well understood that if we have a large sample size then due to the delay logic of Comb Filter, there will be lot of computation which may result in the memory errors. Hence we are bound to find some alternative for an efficient implementation of this kind of filter. One option is to use a Buffer Blocks (available in Signal Management in Signal Processing Blockset of MatLab). This will allow us for a Frame Based Implementation of Comb Filter which will reduce the Computation Time (Memory Saving) by utilizing the First and Last Delay Sample (required for Comb Filter). I applied this Frame Based Implementation and compared the results with the typical sample based approach and as expected the computation time was reduced. A Delay Line Block together with Sub Matrix Blocks were used for this implementation. The simulink models of this implementation alongwith the response are shown here,
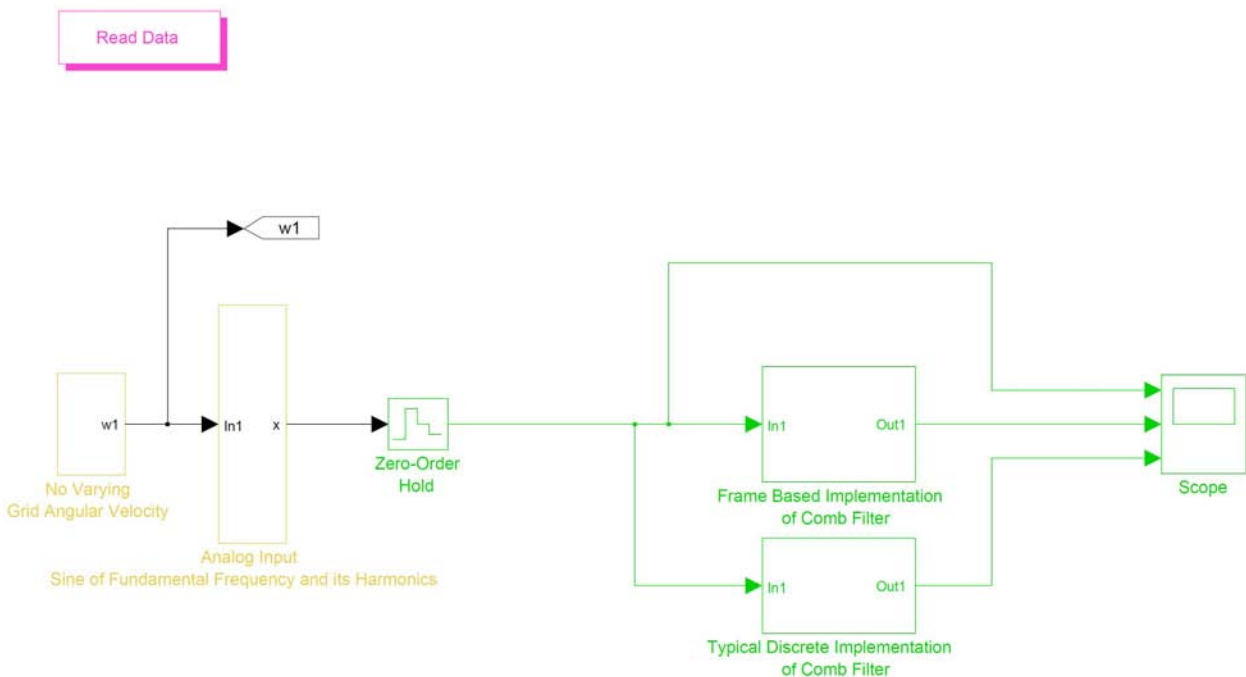

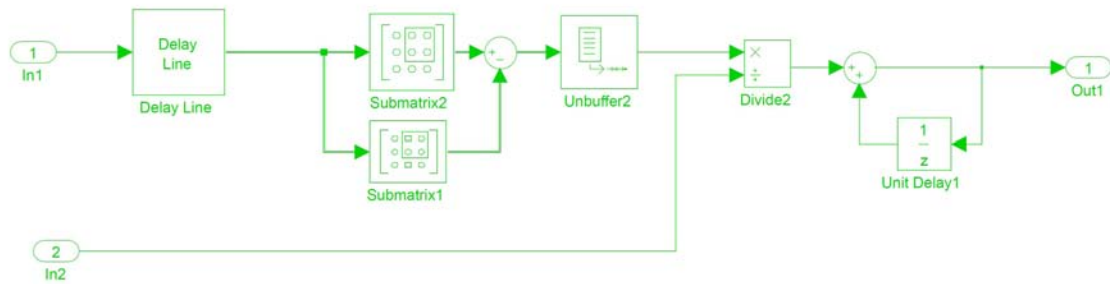Figure C.1: Main Loop for Comparision of 2 Methods
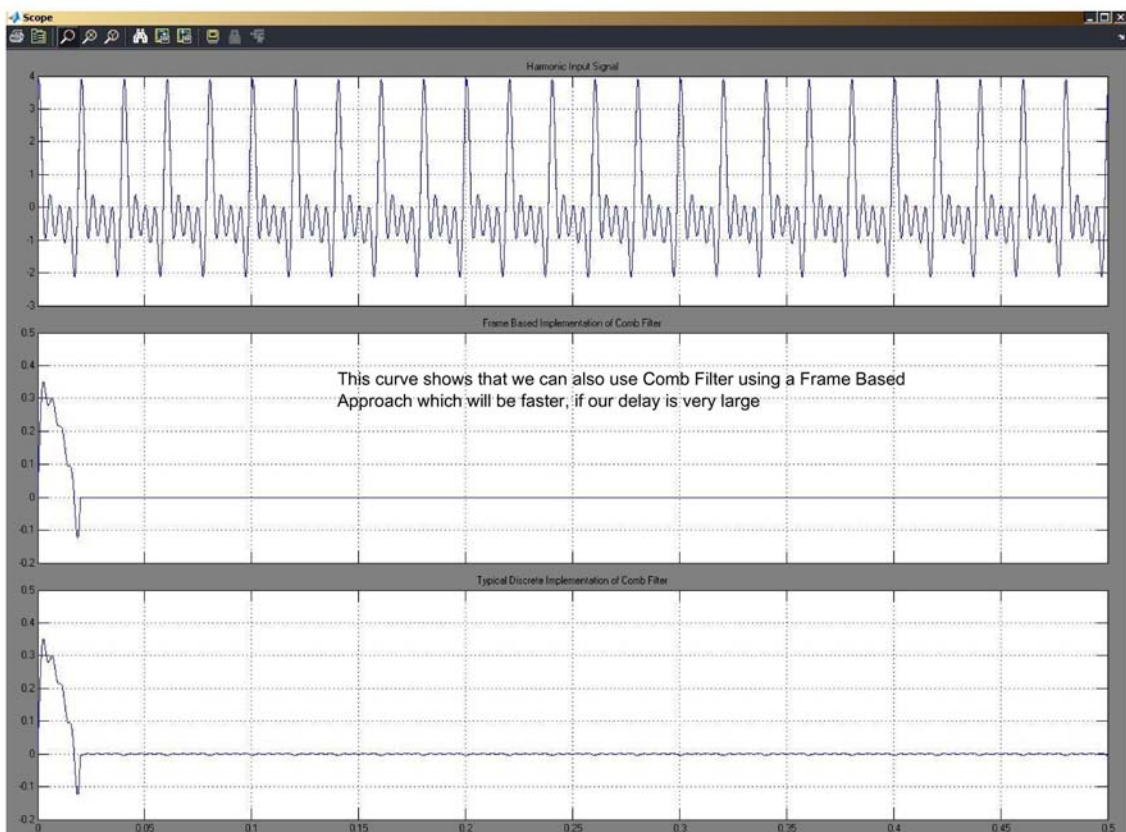
Figure C.2: Frame Based Comb Filter

Figure C.3: Response of Frame Based and Typical Discrete Comb Filter

Bibliography

[1] Farzad Nekoogar and Gene Moriarty, ¨Digital Control Using Digital Signal Processing¨, Prentice Hall Information and System Sciences Series Thomas Kailath, Series Editor

[2] Roland E. Best, ¨Phase Locked Loops; Design, Simulation and Applications¨, 5th Edition, McGraw Hill

[3] Soo-Chang Pei and Chien-Cheng Tseng, ¨A Comb Filter Design, Using Fractional Sample Delay¨, 1997 IEEE International Symposium on Circuits and Systems, June 9-12, 1997, Hong Kong

[4] Timo I. Laakso, Vesa Valimaki, Matti Karjalainen, and Unto K. Laine, ¨Splitting; Tools for fractional delay filter design¨, IEEE Signal Processing Magazine, January 1996

[5] Forestor W. Isen, ¨DSP for MatLab and LabView¨, Volume I, II, III and IV, Synthesis Lectures on Signal Processing, Morgan and Claypool Publishers

[6] http://www.complextoreal.com/chapters/pll.pdf

[7] Online Lectures of Prof. E. Ambikairajah, UNSWelearning on www.youtube.com