ELP311
Communication Engineering Laboratory

Experiment 6
Phase-Locked Loop for Synchronization

Ishvik Kumar Singh
2018EE10616

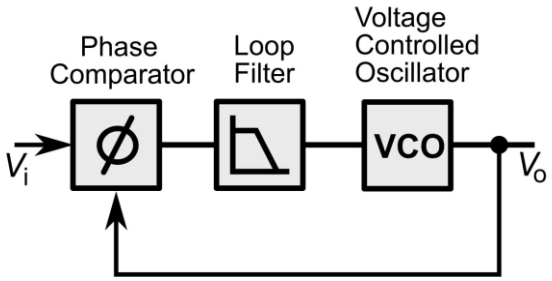
Objective

Modeling of Phase-Locked Loop system using MATLAB

Theory

Coherent detection techniques improve the reliability of any communication system. For coherent detection to work, the receiver radio must synchronize the clock of its local oscillator with that of the carrier signal. This can be achieved using phase-locked loops. In PLLs, the phase of the local oscillator adapts to that of the reference signal. PLLs mainly comprise of

i. A multiplier (used as phase comparator)
ii. A low-pass filter (loop filter)
iii. A voltage-controlled oscillator (VCO)



PLL Block Diagram

The basic principle of PLL is to first multiply the baseline signal of VCO with the transmitted signal, and then feed the result to a low-pass filter. The output of the low pass filter is used to control the VCO, which in turn alters its phase and align it with that of the carrier signal. This is an iterative process that must ensure that the phases of the two signals are aligned after a few iterations.

$$x_1(t) = cos(2\pi ft + \varphi_1), x_2(t) = cos(2\pi ft + \varphi_2)$$
$$y(t) = x_1(t)x_2(t) = \frac{1}{2}cos(\varphi_1 - \varphi_2) + \frac{1}{2}cos(4\pi ft + \varphi_1 + \varphi_2)$$

The low pass filter removes the high frequency component giving the control voltage as

$$v(t) \approx \frac{1}{2}cos(\varphi_1 - \varphi_2)$$

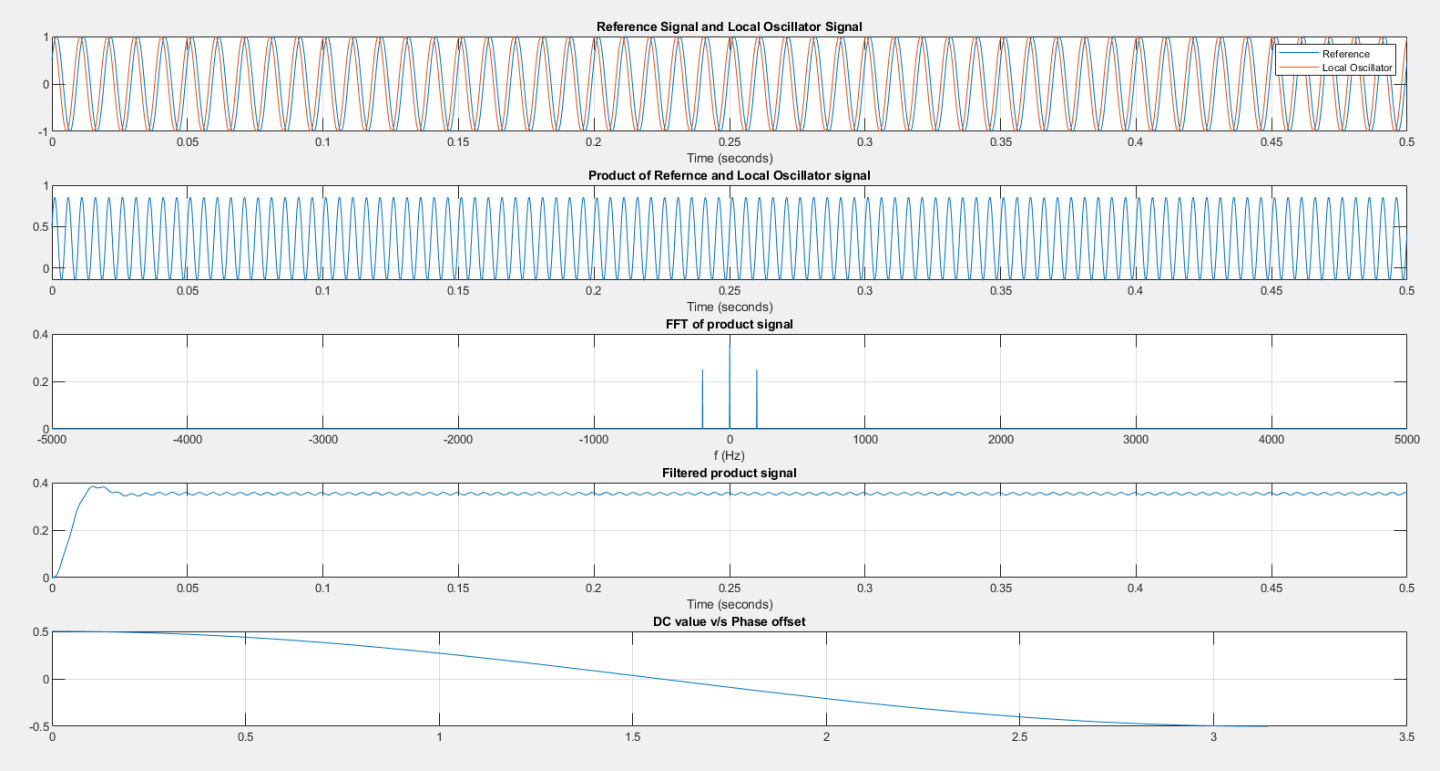
$v(t)$ can be used to correct the phase offset.

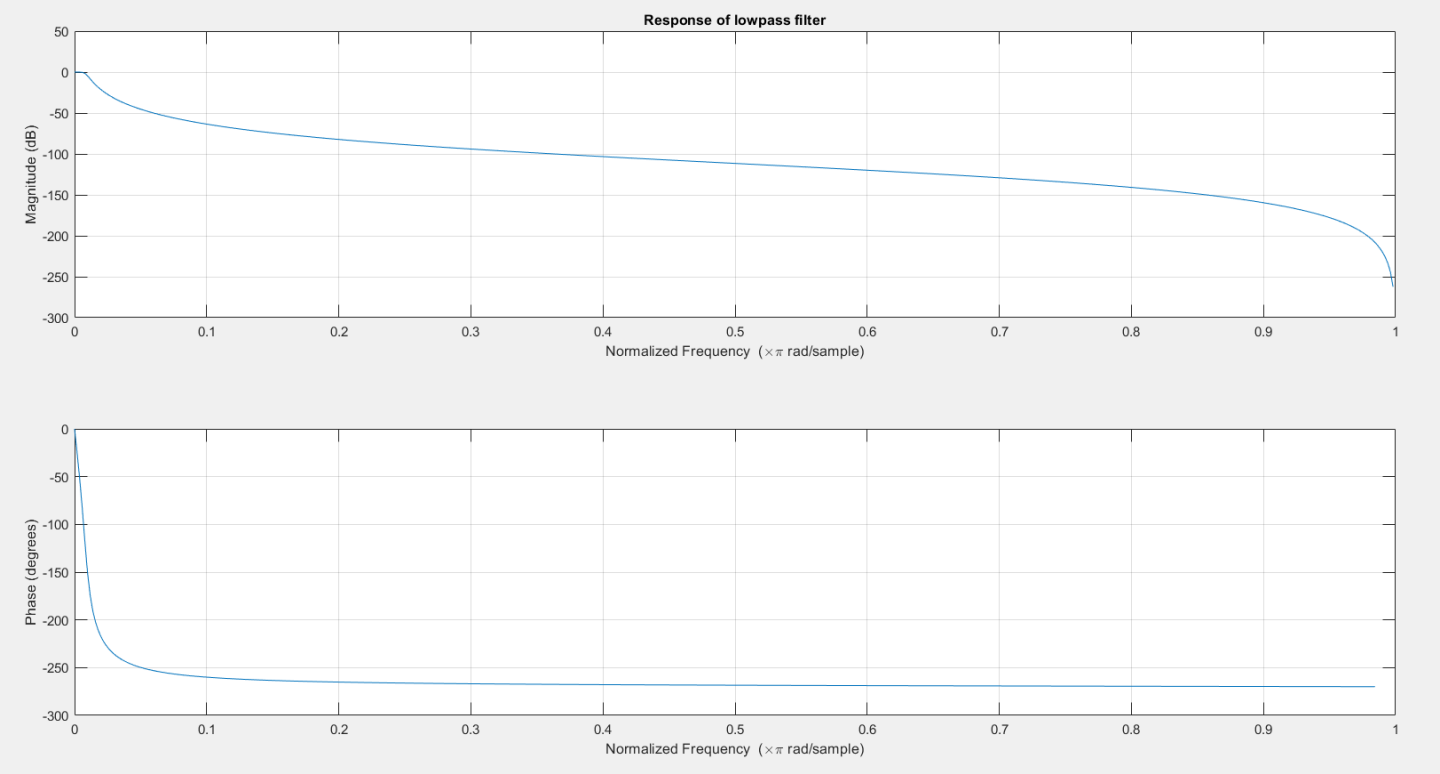
MATLAB code

```
close all;
startplot = 1;
endplot = 5000;

fs = 10000;   % Sample frequency
N = 5000;     % Number of samples

Ts = 1/fs;
t = (0:Ts:N*Ts-Ts);

fm = 100;       % Refernce frequency
phi_ref = pi/6; % Refernce phase
phi_los = phi_ref + pi/4;

ref = sin(2*pi*fm*t + phi_ref); % refernce signal
los = sin(2*pi*fm*t + phi_los); % local oscillator signal

figure(1);
subplot(5,1,1);
plot(t(startplot:endplot), ref(startplot:endplot), t(startplot:endplot), los(startplot:endplot));
title('Reference Signal and Local Oscillator Signal');
xlabel('Time (seconds)');
legend('Reference', 'Local Oscillator');
grid;

product = ref .* los; % product of ref and los

subplot(5,1,2);
plot(t(startplot:endplot), product(startplot:endplot));
title('Product of Refernce and Local Oscillator signal');
xlabel('Time (seconds)');
grid;

lm = length(product);
f = linspace(-fs/2,fs/2,lm);
fourier = fft(product,lm);
fft_product = abs(fftshift(fourier))/lm;    % fft of product

subplot(5,1,3);
plot(f,fft_product)
title('FFT of product signal');
xlabel('f (Hz)');
grid;

% butterworth filter
Wp = 2*20/fs;
Ws = 2*50/fs;
[n,Wn] = buttord(Wp,Ws,0.1,5);
[a,b] = butter(n,Wn);

filtered_signal = filter(a, b, product);    % passing product signal through low pass Butterworth filter

subplot(5,1,4);
plot(t(startplot:endplot), filtered_signal(startplot:endplot));
title('Filtered product signal');
xlabel('Time (seconds)');
grid;

phi = 0:pi/180:pi;
DCvalues = zeros(1, 180);

for i = 1:length(phi)
    los = sin(2*pi*fm*t + phi_ref + phi(i));
    product = ref .* los;

    DCvalues(i) = mean(product);
end

subplot(5,1,5);
plot(phi, DCvalues);
title('DC value v/s Phase offset');
grid;
```

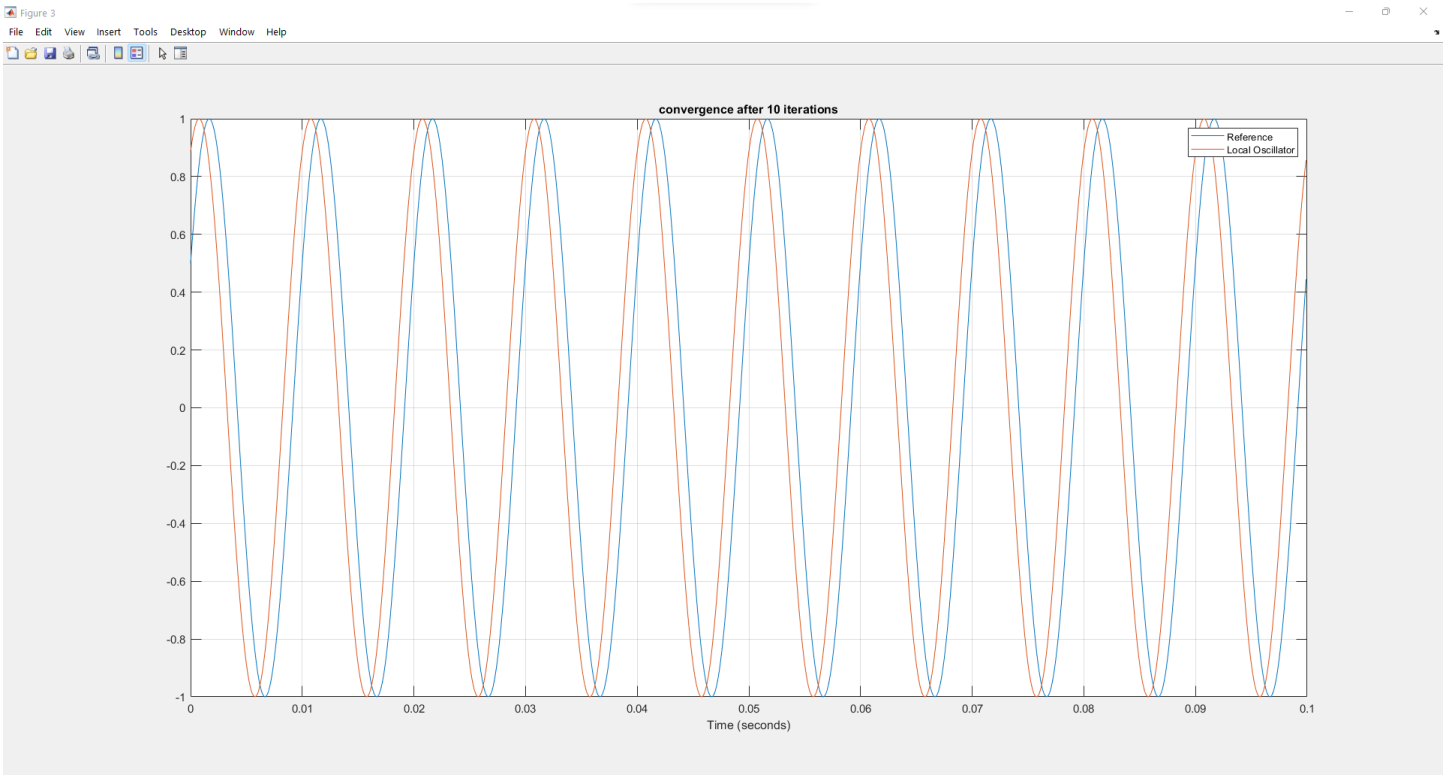
```
figure (2)
freqz(a, b);
title('Response of lowpass filter');

kp = 0.3;
phi1 = 83 * pi/180;
count = 0;

while(1)
    los = sin(2*pi*fm*t + phi1);
    product = ref .* los;

    DC = mean(product);
    phi1 = phi1 +  kp*(DC-0.5);
    count = count + 1;

    if or((rem(count, 200) == 0), (count == 10))
        figure;
        plot(t(startplot:endplot/5), ref(startplot:endplot/5), t(startplot:endplot/5), los(startplot:endplot/5));
        hold on
        title(sprintf("convergence after %d iterations", count));
        xlabel('Time (seconds)');
        legend('Reference', 'Local Oscillator');
        grid;
    end

    if abs(DC-0.5) < 1e-4
        break
    end
end

figure;
plot(t(startplot:endplot/5), ref(startplot:endplot/5), t(startplot:endplot/5), los(startplot:endplot/5));
hold on
title(sprintf("Final convergence after %d iterations", count));
xlabel('Time (seconds)');
legend('Reference', 'Local Oscillator');
grid;
```

Plots

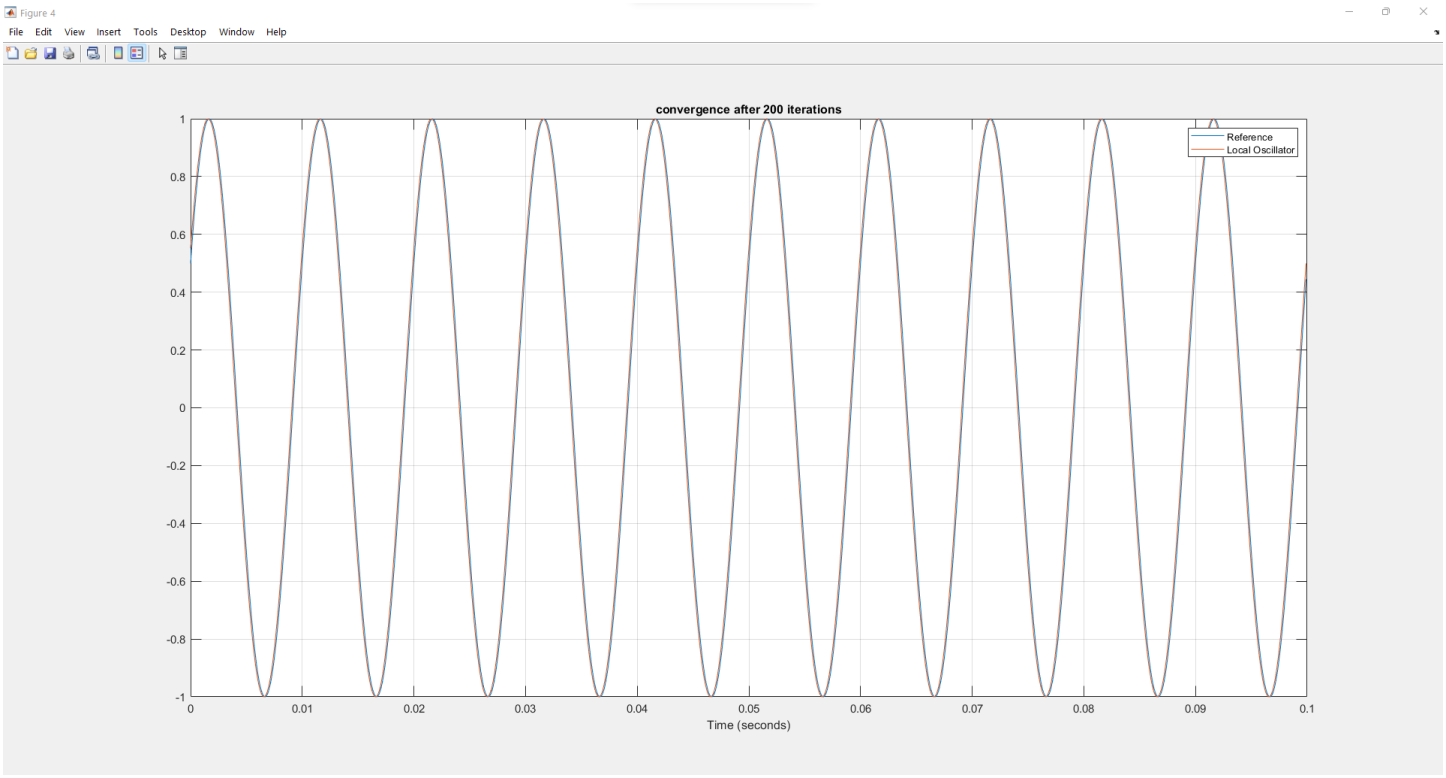




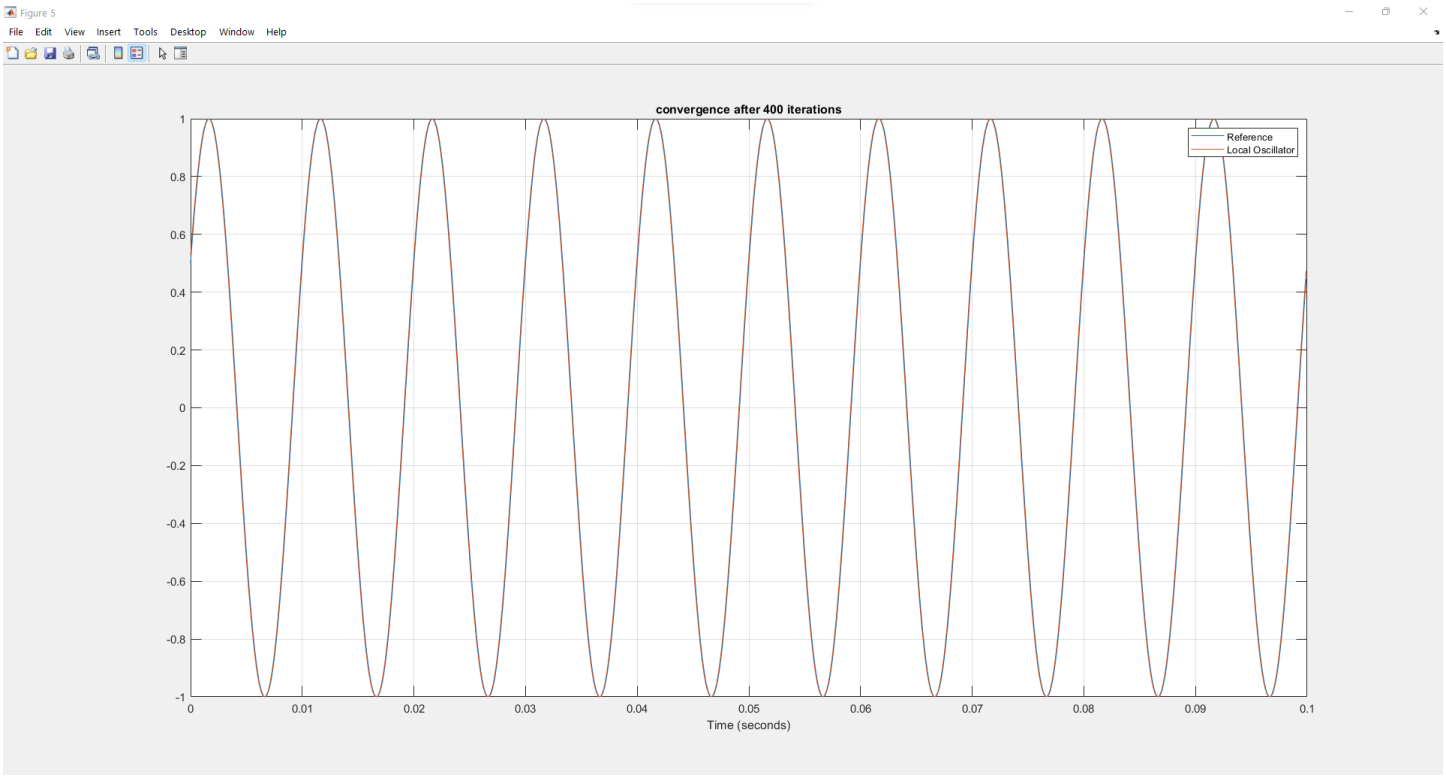
Frequency Response of Low-pass Filter



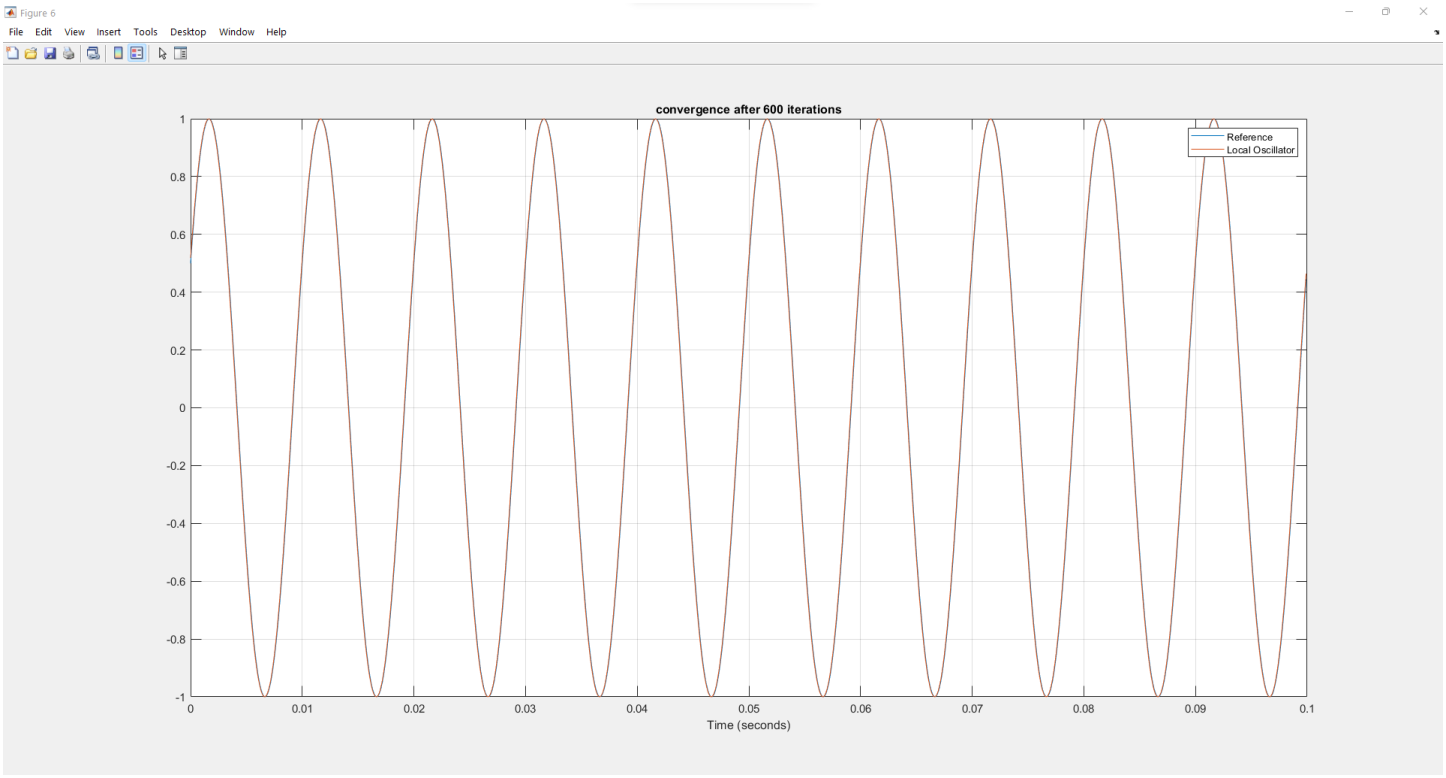
Reference and Local oscillator signal after 10 iterations



Reference and Local oscillator signal after 200 iterations

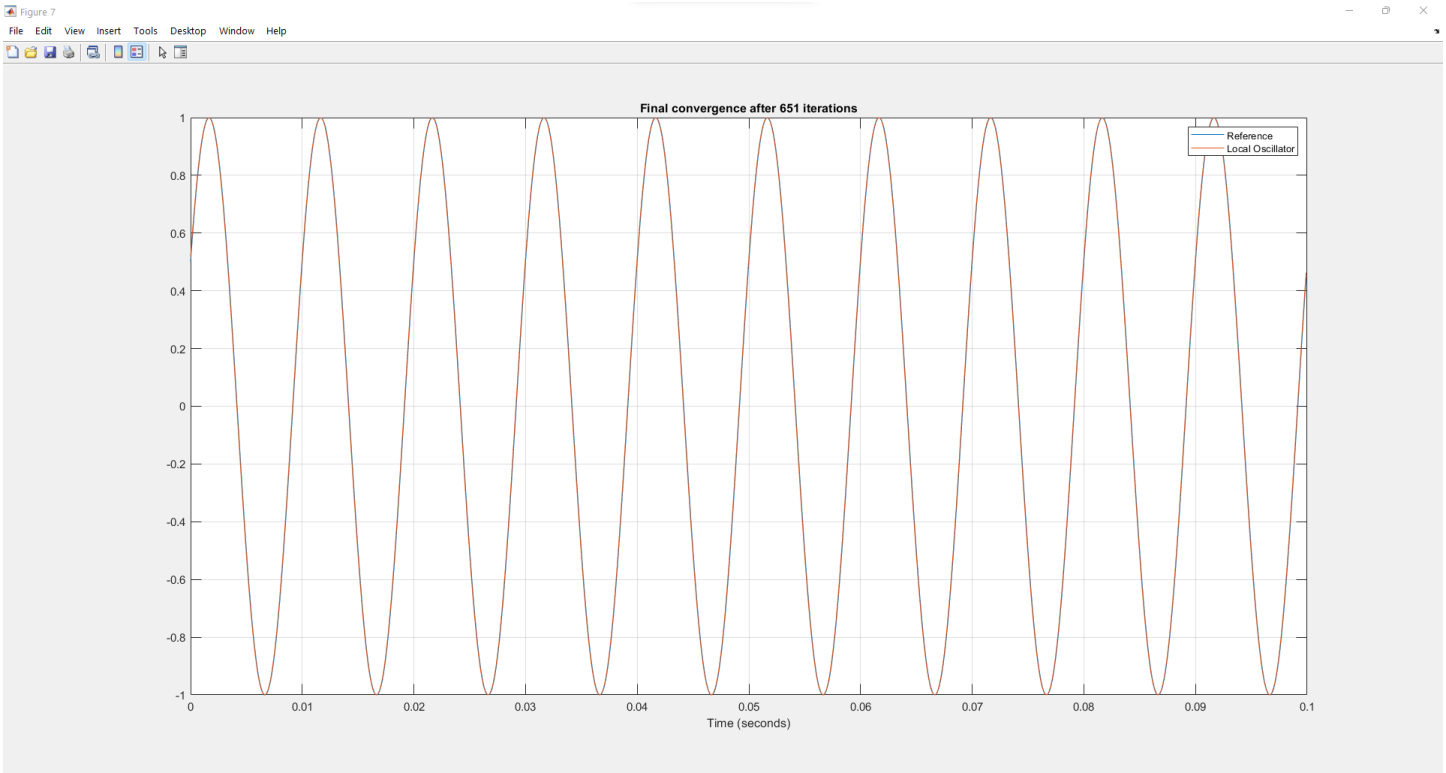


Reference and Local oscillator signal after 400 iterations



Reference and Local oscillator signal after 600 iterations

Final convergence is observed after 651 iterations when the phase offset between the reference signal and the local oscillator signal is less than the tolerance limit of 10^{-4}.



Reference and Local oscillator signal after 651 iterations

Tutorial Questions

1. **What would be the DC value at the output of multiplier when two cosine signals are locked?**

Ans: The DC value at the output of the multiplier when the two cosine signals are locked will $1/2$.

2. **Determine the theoretical capture range (range of frequencies PLL can acquire) for your designed PLL model.**

Ans: The capture range of a PLL is characterized by the capture frequency ω_c which is given as

$$\omega_c = \pm\sqrt{A_0 K_{PD} K_{VCO} \omega_{LP}}$$

Here, A_0 is the gain of the LPF, K_{PD} is the constant of proportionality in the output of phase detector, K_{VCO} is the sensitivity of the VCO, and ω_{LP} is the cutoff frequency of the LPF. The theoretical capture range is given as $2\omega_c$.

Now, in this experiment, $A_0 = 1$, $K_{VCO} = 0.3$, $K_{PD} = 1$, $\omega_{LP} = 276.2736$
Therefore, capture range is $18.2079\ rad/s$.

3. **How long does your designed algorithm takes to lock to the phase with the reference signal? It depends on what factors.**

Ans: The designed algorithm takes 651 iterations for convergence when the phase offset between the reference signal and the local oscillator signal is less than the tolerance limit of 10^{-4}. The time (or iterations) taken for PLL to lock to the phase with the reference signal depends on the initial phase offset between the two signals as well as on the overall loop gain and loop filter characteristics.

4. **How does the parameters of the LPF affect the performance of your PLL model?**

Ans: Following are the parameters of the LPF and their affect on the performance of the PLL:

i. **Bandwidth:** The bandwidth of a low-pass filter is critical in determining the lock time and settling time. Further, the bandwidth can be controlled to ensure that spurious frequency components are eliminated. The loop bandwidth also has an impact on the integrated phase noise.

ii. **Phase Margin:** For optimal lock time, the phase margin is around 45°.If the phase margin is too low, there will be excessive ringing, or even instability. If the phase margin is too high, there will be no ringing, but the frequency will settle in much slower. In designs for optimal integrated phase noise, phase margin chosen higher, up to 80°. This causes the transfer function to be flatter at the expense of lock time. It also increases the VCO noise suppression near the loop bandwidth**.**

iii. **Gamma Parameter:** This parameter is based on the concept of maximizing the phase margin at the loop bandwidth. A gamma value slightly greater than one corresponds to maximizing the phase margin at a frequency less than the loop bandwidth, and a gamma value of less than one corresponds to maximizing the phase margin at a frequency greater than the loop bandwidth.

5. How does the phase sensitivity affect the algorithm?

Ans: Phase sensitivity can used to control the rate of convergence of the PLL algorithm. In general, higher the phase sensitivity, greater will be the rate of convergence, i.e., the phases will lock quickly. However, it must be noted that after a certain value of phase sensitivity the algorithm may never terminate, i.e., the two may never lock. Therefore, K_p must neither be too small or too large.

6. What should be the minimum sampling rate of your model? Should it be just the twice of frequency under consideration?

Ans: As per Shanon-Nyquist criterion, the minimum sampling frequency should be twice the frequency under consideration. However, the sampling rate is usually greater than just the twice of frequency under consideration (oversampling). This is because oversampling can improve resolution and signal-to-noise ratio and can be helpful in avoiding aliasing and phase distortion.

7. As algorithm approaches the convergence, how does the granularity of phase update changes?

Ans: As the algorithm approaches convergence the magnitude of phase update reduces and becomes very small when the algorithm is close to convergence.

Observations and Conclusions

1. The nature of DC values v/s phase difference is same as $\frac{1}{2}\cos\theta$ v/s θ. This was predicted theoretically.
2. For a phase difference of 90°, the DC value in the product signal is 0. Therefore, the frequency spectrum of the filtered product signal shows the passband of the low-pass filter, as it only filters white noise from the digital signal.
3. For lower values of K_p convergence is slower, however this ensures that convergence will be observed for lower tolerance limits also.
