A

Project Report

On

IMPLEMENTATION OF PLL USING MATLAB

Submitted to

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
RK VALLEY, KADAPA

in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

Y RAJESH R200012

K SATHVIKA R200527

P KARTHIK R200036

Under the Guidance of

Mr. K HARINATH REDDY sir, Assistant Professor

RGUKT RK VALLEY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
RKVALLEY,KADAPA 516330

2024-2025

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
RK VALLEY,KADAPA 516330

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DECLARATION

We hereby declare that the project report entitled **"Implementation of PLL using MATLAB"**submitted to the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING** in partial fulfillment of requirements for the award of the degree of **BACHELOR OF TECHNOLOGY**. This project is the result of our own effort and that it has not been submitted to any other University or Institution for the award of any degree or diploma other than specified above.

Y. RAJESH R200012
K.SATHVIKA R200527
P. KARTHIK R200036

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
RK VALLEY,KADAPA 516330

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **Implementation of PLL Using MATLAB** a bonafIde record of the project work done and submitted by

Y. RAJESH R200012
K SATHVIKA R200527
P KARTHIK R200036

for the partial fulfment of the requirements for the award of B.Tech. Degree in **ELECTRONICS AND COMMUNICATION ENGINEERING,** RGUKT,RK VALLEY.

GUIDE
Mr K.Harinath Reddy
Assistant Professor
RGUKT,RK VALLEY
Kadapa-516330

Head Of the Department
Mr.Y Arun kumar Reddy
Assistant Professor
RGUKT,RK VALLEY
Kadapa-516330

ACKNOWLEDGEMENT

We are thankful to our guide **Mr K HARINATH REDDY,** for his valuable guidance and encouragement. His helping attitude and suggestions have helped us in the successful completion of the project.

We would like to express our gratefulness and sincere thanks to **Y ARUN KUMAR REDDY**, Head of the Department of ELECTRONICS AND COMMUNICATION ENGINEERING, for his kind help and encouragement during the course of our study and in the successful completion of the project work. Successful completion of any project cannot be done without proper support and encouragement. We sincerely thanks to the management for providing all the necessary facilities during the course of study.

We would like to thank our parents and my friends who have the greatest contributions in all our achievements, for the great care and blessings in making us successful in all our endeavors.

Y. RAJESH R200012
K. SATHVIKA R200527
P KARTHIK R200036

ABSTRACT

KEY WORDS**:** Phase locked loop(PLL), phase detector(PD), voltage controlled oscillator(VCO),Loop filter(LF).

This project focuses on implementing a phase locked loop using MATLAB. A phase locked loop is a control system that geberates an output signal whose phase is related to the phase of an input signal.The project aims to design and simulate a PLL system in MATLAB to achieve frequency and phase synchronisation.The project will explore behaviour of the PLL under different conditions and evaluates its effectiveness in maintaining phase and frequency coherence.Through this implementation, the project seeks to deeper understanding of PLL systems and their applications in communication and control systems.

CONTENTS

CHAPTER 1

INTRODUCTION

INTRODUCTION

Concept of PLL was introduced in 1930s and since then the scope of PLL application has attracted many designers. With the advancement of the technology new designs, new problems and non idealities are emerging and hence the motivation to work in this field. PLL has vast application in the area of electronics and communication. PLL can be used for clock generation for a microprocessor, as a frequency synthesizer in a mobile, etc.

The PLL is a feedback control system that automatically adjusts the phase of a locally generated signal to match the phase of input signal.PLLs' operate by producing an oscillator frequency to match the frequency of input signal.in this locked condition,any slight change in the input signal first appears as change in phase between input signal & oscillator frequency.this phase shift then acts as an error signal to change frequency of the local PLL oscillator to match input signal.the locking onto a phase relationship between the input signal and the local oscillator accounts for the name Phase locked loop.

CHAPTER 2

PHASE LOCKED LOOP

2.1PHASE LOCKED LOOP:

A PLL is a closed-loop feedback system that sets fixed phase relationship between its output clock phase and the phase of a reference clock. A PLL is capable of tracking the phase changes that falls in this bandwidth of the PLL. A PLL also multiplies a low-frequency reference clock CKref to produce a high-frequency clock CKout this is known as clock synthesis.

A PLL has a negative feedback control system circuit. The main objective of a PLL is to generate a signal in which the phase is the same as the phase of a reference signal. This is achieved after many iterations of comparison of the reference and feedback signals. In this lock mode the phase of the reference and feedback signal is zero. After this, the PLL continues to compare the two signals but since they are in lock mode, the PLL output is constant.

The basic block diagram of the PLL is shown in the Figure 1. In general a PLL consists of three main blocks:

1. Phase Detector or Phase Frequency Detector (PD or PFD)

2. Low Pass Filter (LPF)

3. Voltage Controlled Oscillator (VCO)

The basic structure of the PLL can be understood from the block diagram below .

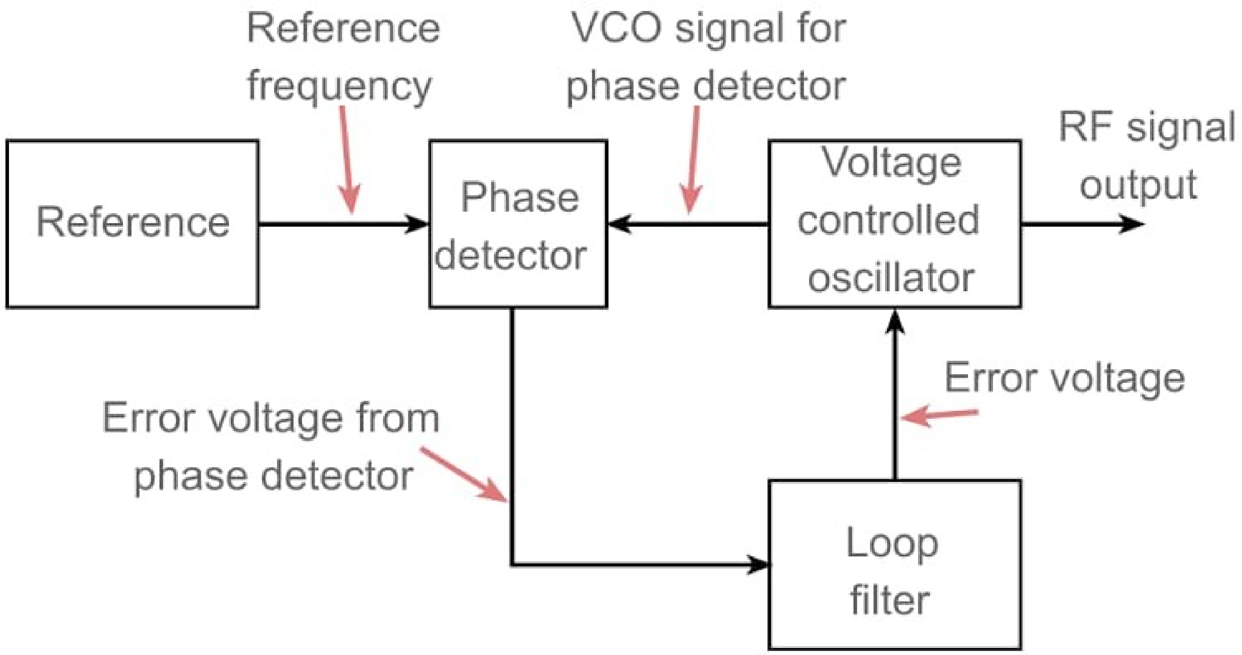


Fig. Block diagram of PLL.

The "Phase frequency Detector" (PFD) is one of the main parts in PLL circuits. It compares the phase and frequency difference between the reference clock and the feedback clock.Depending upon the phase and frequency deviation, it generates two output signals "UP" and "DOWN". The phase and frequency of the "Voltage Controlled Oscillator" (VCO) output depends on the generated DC control voltage. If the PFD generates an "UP" signal, the error voltage at the output of LPF increases which in turn increase the VCO output signal frequency. On the contrary, if a "DOWN" signal is generated, the VCO output signal frequency decreases. The output of the VCO is then fed back to the PFD in order to recalculate the phase difference, and then we can create closed loop frequency control system.

2.2 PLL Architecture

A PLL comprises ofseveral components. They are (1) phase or phase frequency detector, (2) loop filter, (3) voltage-controlled oscillator each block is briefly explained below.

2.2.1 Phase Frequency Detector

The “Phase frequency Detector” (PFD) is one of the main part in PLL circuits. It compares the phase and frequency difference between the reference clock and the feedback clock.Depending upon the phase and frequency deviation, it generates two output signals “UP” and“DOWN”.

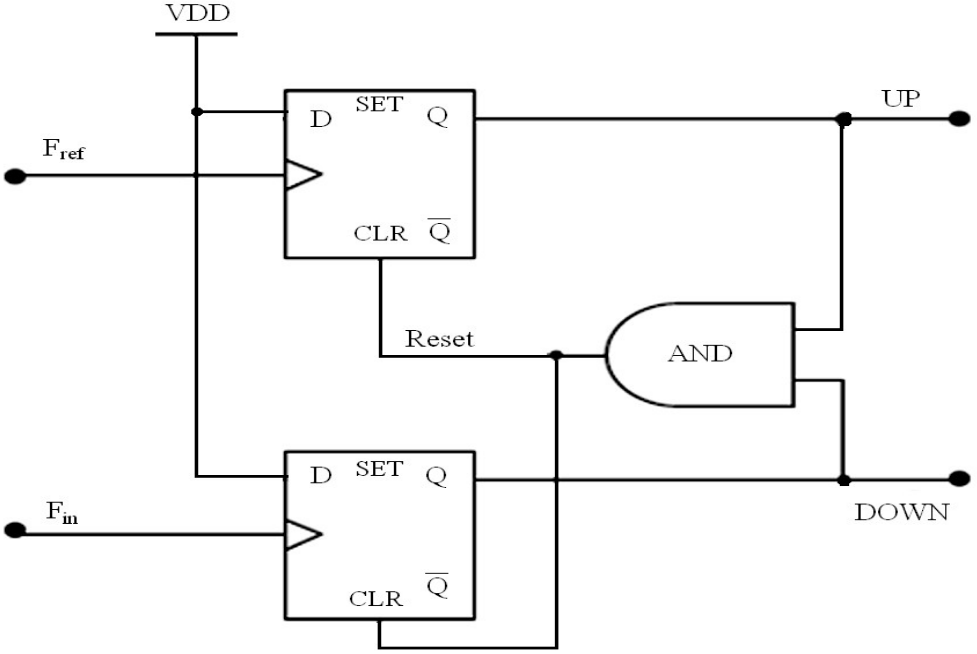
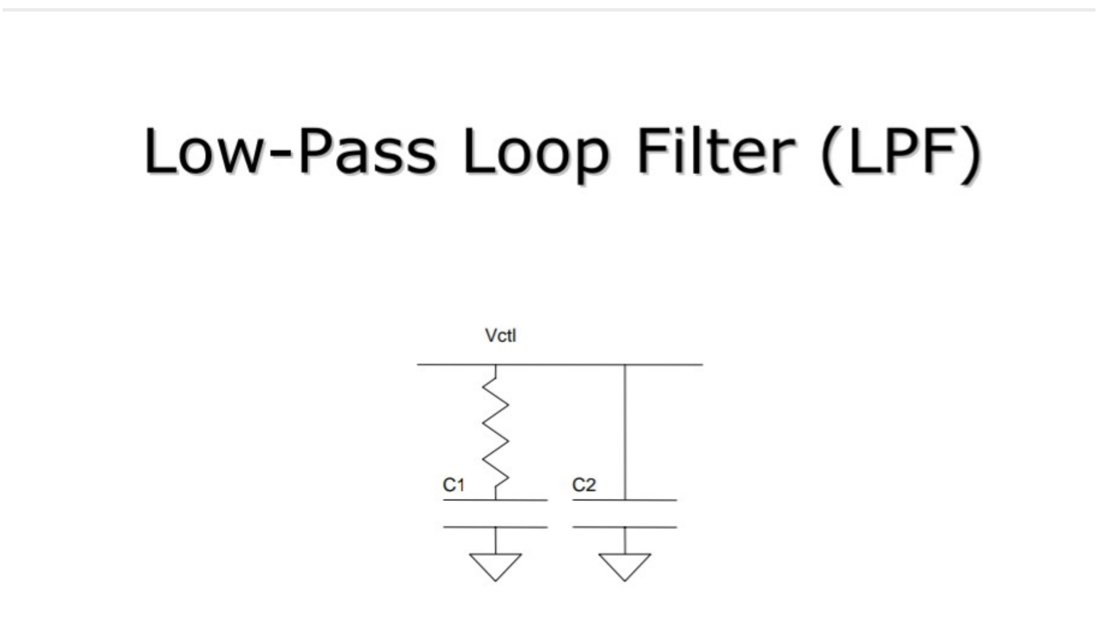


Fig: Block diagram of a traditional PFD circuit

If there is a phase difference between the two signals, it will generat e “UP” or “DOWN”synchronized signals. When the reference clock rising edge leads the feedback input clock rising edge “UP” signal goes high while keeping “DOWN” signal low. On the other hand if the feedback input clock rising edge leads the reference clock rising edge “DOWN” signal goes high and “UP” signal goes low. Fast phase and frequency acquisition PFDs [6-7] are generally preferred over traditional PFD.

2.2.2 Loop Filter

Low Pass Filter is used to block the high frequency signals from the PFD and allows only low frequency signals. It generates DC voltage by which the output is varied. This DC voltage is used to control the output frequency of the Voltage controlled oscillator. The output of LPF is fed to VCO input and it is given by the equation. The DC voltage of LPF depends on input and reference frequency. If both are same and so they are locked then it maintains constant value. A simple low pass filter circuit is shown in Fig.



Low-Pass Loop Filter (LPF)

Fig: Loop filter

2.2.3 Voltage Controlled Oscillator

An oscillator is an autonomous system which generates a periodic output without any input.The most popular type of the VCO circuit is the current starved voltage controlled oscillator (CSVCO). Here the number of inverter stages is fixed with 5. The simplified view of a single stage current starved oscillator is shown in the Figure.

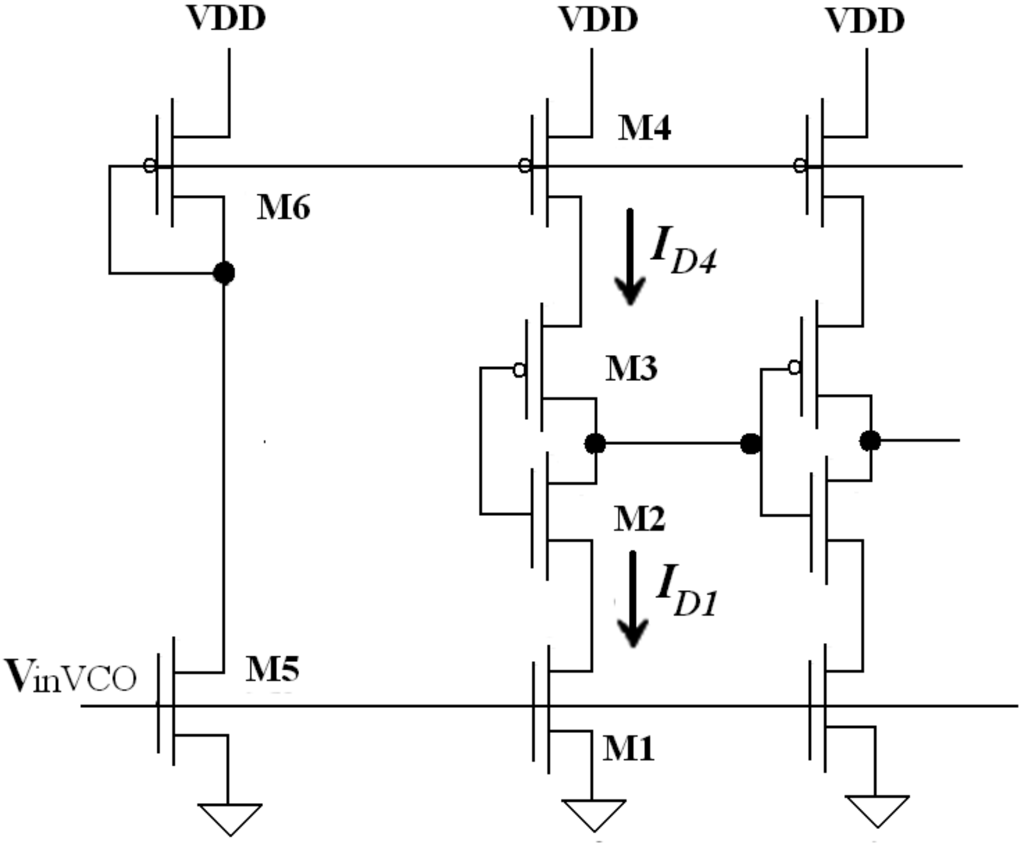


fig:Simplified view of a current starved VCO

10 Transistors M2 and M3 operate as an inverter while M1 and M4 operate as current sources.The current sources, Ml and M4, limit the current available to the inverter, M2 and M3; in other words, the inverter is starved for current. The desired center frequency of the designed circuit is 1GHz with a supply of 1.8V. The CSVCO is designed both in usual manner.

The general circuit diagram of the current starved voltage controlled oscillator is shown in the Figure 2.6.

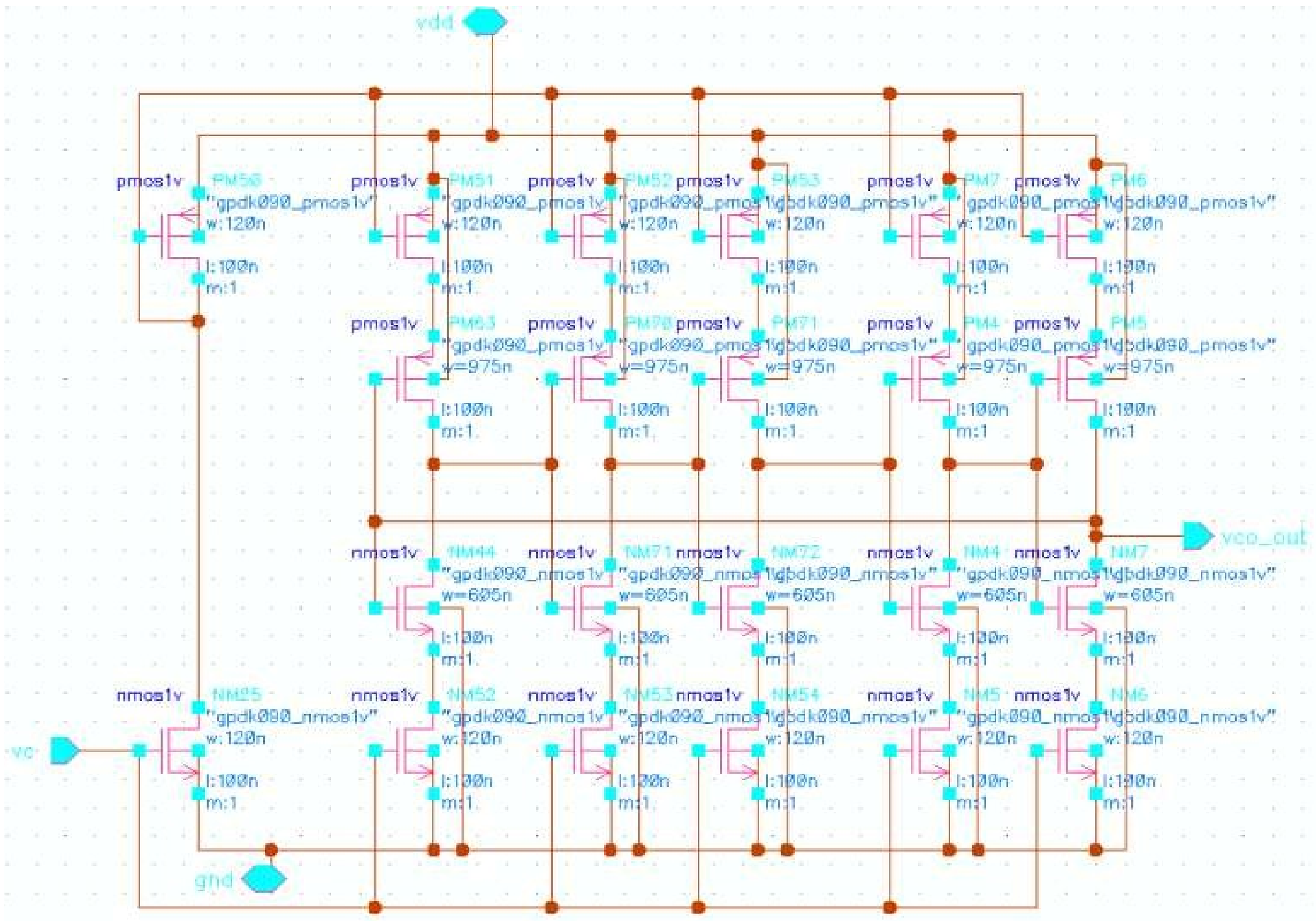


Fig: Circuit diagram of a current starved VCO

2.3 Types of PLL

There are mainly 4 types of PLL are available. They are

1. Liner PLL
2. Digital PLL
3. All Digital PLL
4. Soft PLL

Linear PLL: A Linear Phase-Locked Loop operates in the analog domain. It uses linear components like resistors, capacitors, and operational amplifiers to process the phase error between the input and output signals. The linear PLL is effective for applications where the signals are continuous and can vary smoothly. It's commonly used in radio frequency applications.

Digital PLL: A Digital Phase-Locked Loop uses digital components and processes the signals in a digital format. It typically involves a phase detector, a digital loop filter, and a voltage-controlled oscillator (VCO) that is controlled digitally. Digital PLLs are commonly used in digital communication systems because they can handle discrete signals and are more robust to noise.

All Digital PLL: An All Digital PLL is a specific type of digital PLL where all components, including the phase detector, loop filter, and oscillator, are implemented using digital circuits. This means everything is processed in the digital domain, which allows for greater integration, flexibility, and ease of implementation in digital systems. They are often used in applications like clock recovery and frequency synthesis in digital systems.

Soft PLL: A Soft PLL is a type of PLL that can adapt its parameters dynamically based on the conditions of the input signal. This adaptability allows it to provide better performance in varying environments, especially in wireless communications where the signal conditions can change rapidly. The "soft" aspect refers to its ability to adjust and optimize its operation for better tracking and stability.

- These PLLs differ mainly in their implementation (analog vs. digital) and their adaptability to changing conditions. Linear PLLs are more traditional analog systems, while digital and all-digital PLLs are designed for modern digital applications, and soft PLLs offer dynamic adaptability for improved performance.

2.4 Terms in PLL

2.4.1 Lock in Range

Once the PLL is in lock state what is the range of frequencies for which it can keep itself locked is called as lock in range. This is also called as tracking range or holding range.

2.4.2 Capture Range

When the PLL is initially not in lock, what frequency range can make PLL lock is called as capture range. This is also known as acquisition range. This is directly proportional to the LPF bandwidth. Reduction in the loop filter bandwidth thus improves the reejection of the out of band signals, but at the same time the capture range decreases, pull in time becomes larger and phase margin becomes poor.

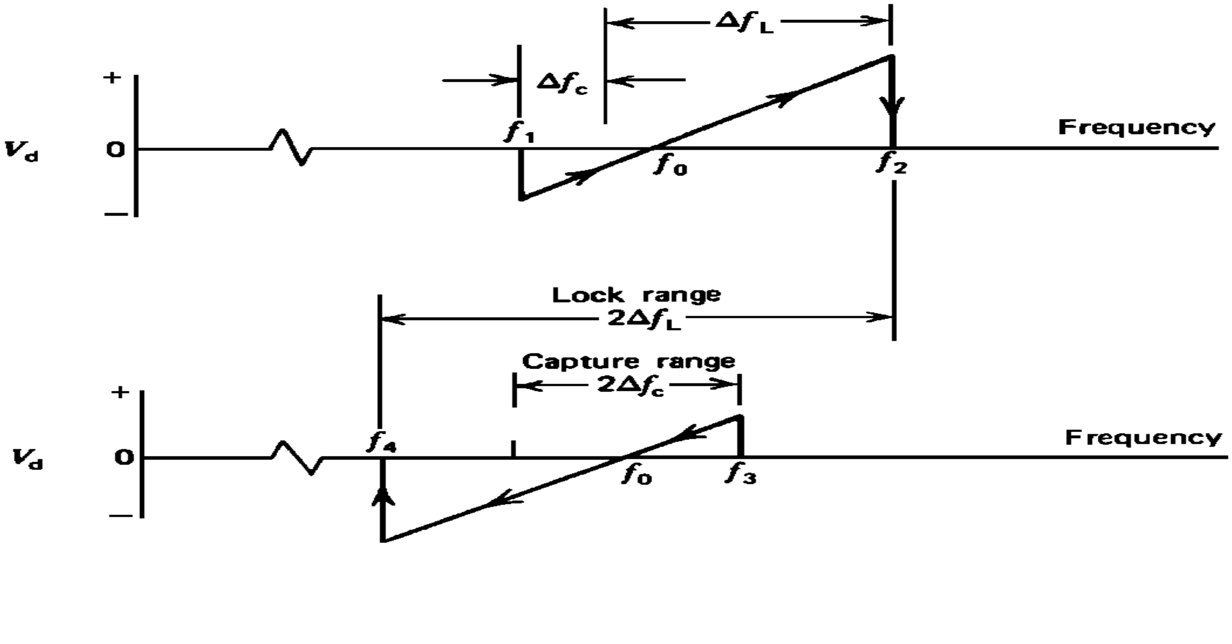


Fig: Illustration of lock and capture range

2.4.3 Pull in Time

The total time taken by the PLL to capture the signal (or to establish the lock) is called as Pull in Time of PLL. It is also called as Acquisition Time of PLL.

2.4.4 Bandwidth of PLL

Bandwidth is the frequency at which the PLL begins to lose the lock with reference.

CHAPTER 3

DESIGN AND SYNTHESIS OF PLL

3.2 Design Procedure

3.2.1 VCO Design

Since VCO is the heart of the whole PLL system, it should be designed in a proper manner. The design steps for the current starved VCO are as follows.

Step 1:

Find the value of the propagation delay for each stage of the inverter in the VCO circuit using the following equation.

Step 2:

Find the ratio for the transistors in the different inverter stages using the equation.

Step 3:

After finding the ratio, find the values for W and L.

Step 4:

Find the value of the total capacitance form the expression is the width and length of the PMOS and NMOS transistors in the inverter stages.

Step 5:

Calculate the value of drain current for the center frequency.

Step 6:

Find the ratio for the current starving transistors in the circuit from the drain current expression .

3.2.2 Design of Phase Locked Loop

The value of the charge pump current and the component parameters of the loop filter play a major role in the design of the phase locked loop circuit. The value of the lock time mainly depends upon these parameters. So while designing the circuit proper care should be taken in calculating these parameters. For the given values of reference(Fref) and output frequency(Fout) as well as the lock in range, the following steps to be carried out in designing the filter circuit.

Step 1: Find the value of the divider circuit to be used which is given by

Step 2: Find the value of the natural frequency from the lock in range .

Step 3:Find the value of the charge pump gain from the charge pump current used in the circuit.

Step 4:Find the value of the gain of the VCO circuit from the characteristics curve using the following expression.

Step 5 :Find the values of the loop filter component parameters using the following expressions.

3.3 Design Specifications and Parameters

3.3.1 VCO Design Specification

The current starved VCO design specifications are mentioned in the following table.

Table1: VCO design specifications

Parameter	Value
Center frequency	1GHz
No. of inverter stage	5
Inverter delay	100ps
Load capacitance	65F
Supply voltage	1.8V

3.3.2 VCO Design Parameters

Table2: List of design parameters of the CSVCO circuit

Parameter	Value
Width of Current starved PMOS(WPCS)	2.33µm
Width of Current Starved NMOS(WnCS)	140nm
Width of PMOS in Inverter(WP)	2.44µm
Width of NMOS in Inverter(Wn)	150nm
LPCS = LnCS = LP = Ln = L	100nm

3.3.3 PLL Design Parameters

The whole PLL system design specifications and parameters are shown in the table below.

Table 3.PLL design specifications and parameters

Parameter	value
Reference frequency(Fref)	500 MHz
output frequency(Fout)	1 GHz
Lock in range	100 MHz
Supply voltage	1.8V
Capacitor (C1)	15 pf
Capacitor (C2)	1.5pf
Resistor (R)	1.384Ω

CHAPTER 4

SIMULATION AND RESULTS

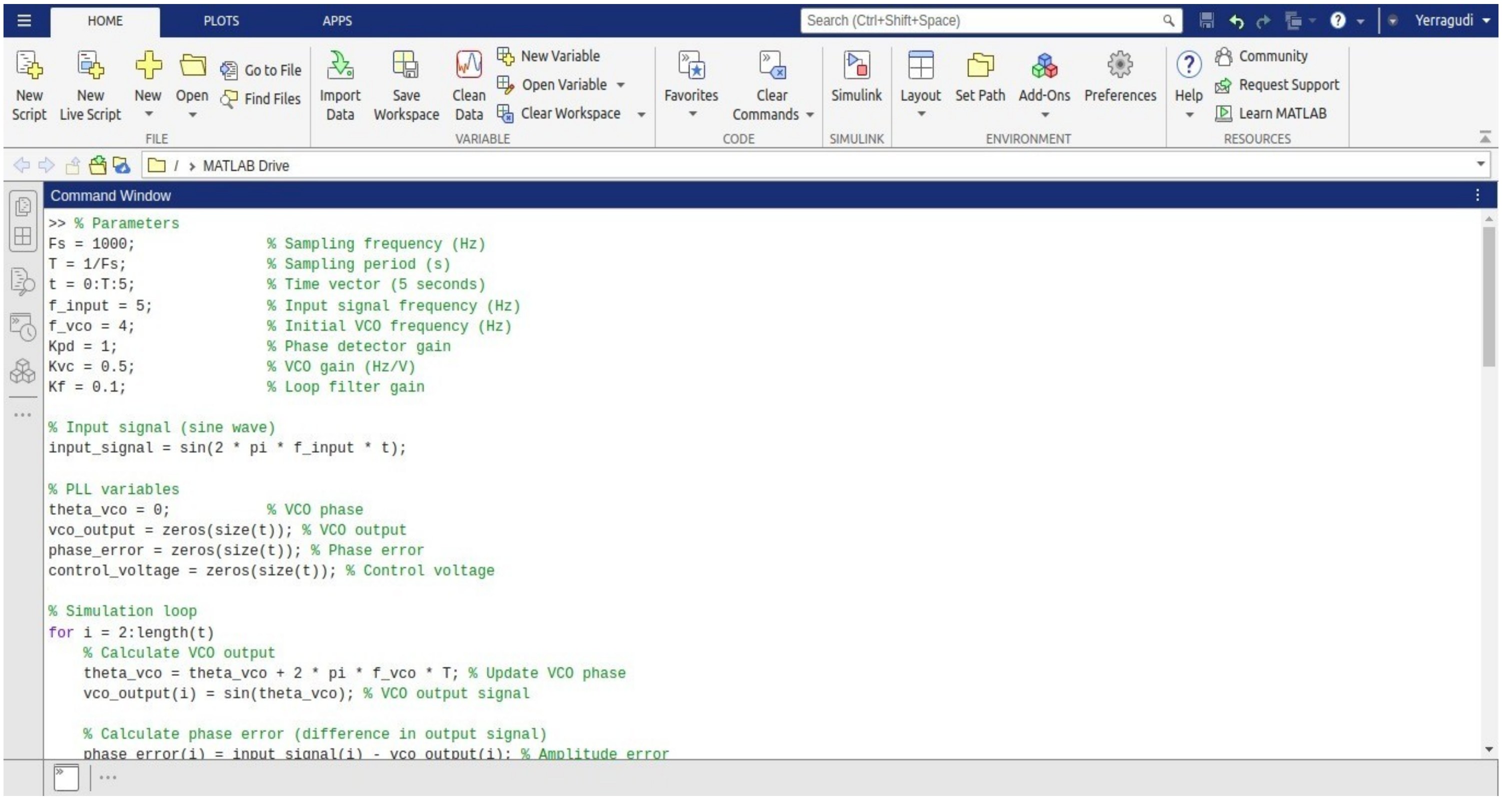
4.1 MATLAB code to implement PLL

Here is the MATLAB code to implement PLL.

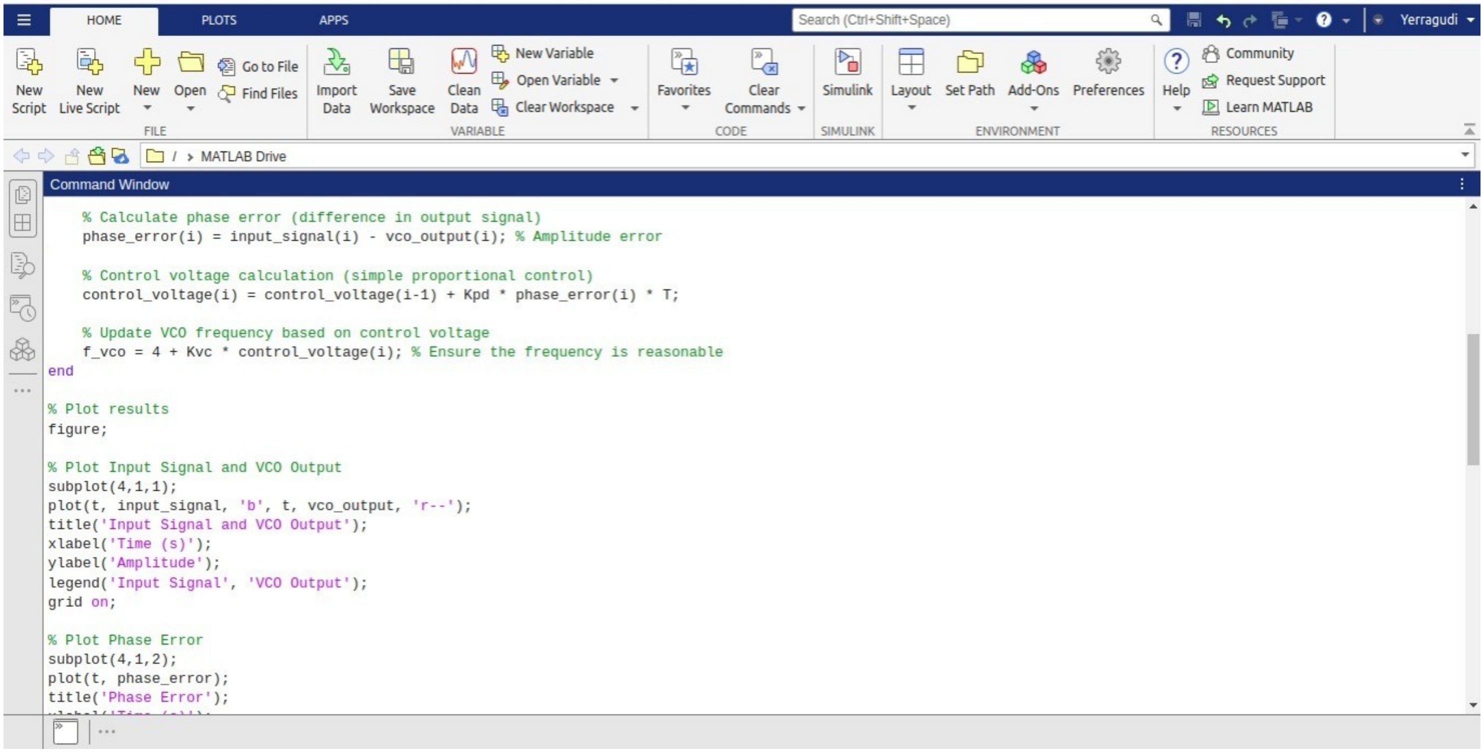
Code:

```
% Parameters
Fs = 1000;              % Sampling frequency (Hz)
T = 1/Fs;               % Sampling period (s)
t = 0:T:5;              % Time vector (5 seconds)
f_input = 5;            % Input signal frequency (Hz)
f_vco = 4;              % Initial VCO frequency (Hz)
Kpd = 1;                % Phase detector gain
Kvc = 0.5;              % VCO gain (Hz/V)
Kf = 0.1;               % Loop filter gain
% Input signal (sine wave)
input_signal = sin(2 * pi * f_input * t);
% PLL variables
theta_vco = 0;          % VCO phase
vco_output = zeros(size(t)); % VCO output
phase_error = zeros(size(t)); % Phase error
control_voltage = zeros(size(t)); % Control voltage
% Simulation loop
for i = 2:length(t)
    % Calculate VCO output
    theta_vco = theta_vco + 2 * pi * f_vco * T; % Update VCO phase
    vco_output(i) = sin(theta_vco); % VCO output signal
 % Calculate phase error (difference in output signal)
    phase_error(i) = input_signal(i) - vco_output(i); % Amplitude error
 % Control voltage calculation (simple proportional control)
   control_voltage(i) = control_voltage(i-1) + Kpd * phase_error(i) * T;
    % Update VCO frequency based on control voltage
    f_vco = 4 + Kvc * control_voltage(i); % Ensure the frequency is reasonable
end
% Plot results
figure;
% Plot Input Signal and VCO Output
subplot(4,1,1);
plot(t, input_signal, 'b', t, vco_output, 'r--');
title('Input Signal and VCO Output');
xlabel('Time (s)');
ylabel('Amplitude');
legend('Input Signal', 'VCO Output');
grid on;
% Plot Phase Error
subplot(4,1,2);
plot(t, phase_error);
title('Phase Error');
xlabel('Time (s)');
ylabel('Phase Error');
grid on;
% Plot Control Voltage
subplot(4,1,3);
```

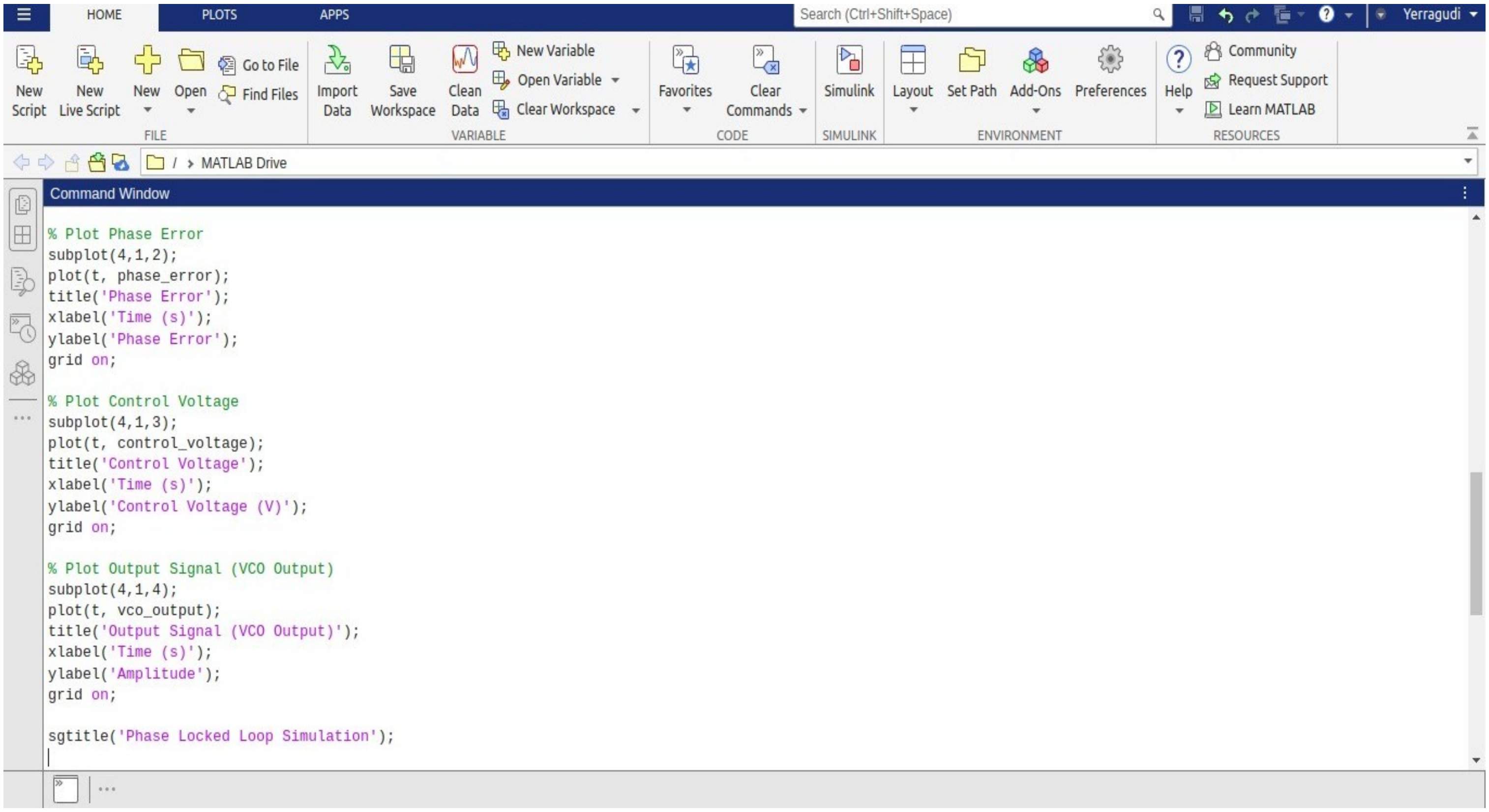
```
plot(t, control_voltage);

title('Control Voltage');

xlabel('Time (s)');

ylabel('Control Voltage (V)');

grid on;

% Plot Output Signal (VCO Output)

subplot(4,1,4);

plot(t, vco_output);

title('Output Signal (VCO Output)');

xlabel('Time (s)');

ylabel('Amplitude');

grid on;

sgtitle('Phase Locked Loop Simulation');
```



```
>> % Parameters
Fs = 1000;              % Sampling frequency (Hz)
T = 1/Fs;               % Sampling period (s)
t = 0:T:5;              % Time vector (5 seconds)
f_input = 5;            % Input signal frequency (Hz)
f_vco = 4;              % Initial VCO frequency (Hz)
Kpd = 1;                % Phase detector gain
Kvc = 0.5;              % VCO gain (Hz/V)
Kf = 0.1;               % Loop filter gain

% Input signal (sine wave)
input_signal = sin(2 * pi * f_input * t);

% PLL variables
theta_vco = 0;          % VCO phase
vco_output = zeros(size(t)); % VCO output
phase_error = zeros(size(t)); % Phase error
control_voltage = zeros(size(t)); % Control voltage

% Simulation loop
for i = 2:length(t)
    % Calculate VCO output
    theta_vco = theta_vco + 2 * pi * f_vco * T; % Update VCO phase
    vco_output(i) = sin(theta_vco); % VCO output signal

    % Calculate phase error (difference in output signal)
    phase_error(i) = input_signal(i) - vco_output(i); % Amplitude error
```



```
    % Calculate phase error (difference in output signal)
    phase_error(i) = input_signal(i) - vco_output(i); % Amplitude error

    % Control voltage calculation (simple proportional control)
    control_voltage(i) = control_voltage(i-1) + Kpd * phase_error(i) * T;

    % Update VCO frequency based on control voltage
    f_vco = 4 + Kvc * control_voltage(i); % Ensure the frequency is reasonable
end

% Plot results
figure;

% Plot Input Signal and VCO Output
subplot(4,1,1);
plot(t, input_signal, 'b', t, vco_output, 'r--');
title('Input Signal and VCO Output');
xlabel('Time (s)');
ylabel('Amplitude');
legend('Input Signal', 'VCO Output');
grid on;

% Plot Phase Error
subplot(4,1,2);
plot(t, phase_error);
title('Phase Error');
```



```
% Plot Phase Error
subplot(4,1,2);
plot(t, phase_error);
title('Phase Error');
xlabel('Time (s)');
ylabel('Phase Error');
grid on;

% Plot Control Voltage
subplot(4,1,3);
plot(t, control_voltage);
title('Control Voltage');
xlabel('Time (s)');
ylabel('Control Voltage (V)');
grid on;

% Plot Output Signal (VCO Output)
subplot(4,1,4);
plot(t, vco_output);
title('Output Signal (VCO Output)');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;

sgtitle('Phase Locked Loop Simulation');
```

Code Explanation:

This MATLAB code simulates a Phase-Locked Loop (PLL) system, which is used for synchronizing an output signal's frequency and phase with that of an input signal.

Parameters:

1. Fs = 1000;

Sampling frequency (Hz). This defines how many samples per second are taken from the signals

2. T = 1/Fs;

Sampling period (s). This is the time interval between each sample, calculated as the inverse of the sampling frequency.

3. t = 0:T:5;

Time vector (5 seconds). This creates a time vector that ranges from 0 to 5 seconds, with intervals of T.

4. f_input = 5;

Input signal frequency (Hz). This is the frequency of the sine wave input signal that the PLL will try to lock onto.

5. f_vco = 4;

Initial VCO (Voltage Controlled Oscillator) frequency (Hz). This is the starting frequency of the VCO before it begins to adjust to the input signal.

6. Kpd = 1;

Phase detector gain. This determines how much the phase error affects the control voltage applied to the VCO.

7. Kvc = 0.5;

VCO gain (Hz/V). This indicates how much the output frequency of the VCO changes in response to a control voltage.

8. Kf = 0.1;

Loop filter gain. In this simplified model, it's not directly used but typically would affect the smoothing of the control voltage.

Variables:

1. input_signal = sin(2 * pi * f_input * t);

The input signal is a sine wave generated at the frequency defined by f_input.

2. theta_vco = 0;

VCO phase, which is initialized to zero. This keeps track of the current phase of the VCO output.

3. vco_output = zeros(size(t));

This initializes an array to hold the output of the VCO over time.

4. phase_error = zeros(size(t));

This initializes an array to store the phase error at each time step.

5. control_voltage = zeros(size(t));

This initializes an array for the control voltage, which will be used to adjust the VCO frequency.

Simulation Loop:

The loop iterates over the time vector to simulate the behavior of the PLL:

1. VCO Output Calculation:

- theta_vco = theta_vco + 2 * pi * f_vco * T;

  Updates the VCO phase by adding the phase corresponding to the current VCO frequency.

- vco_output(i) = sin(theta_vco);

  Computes the VCO output signal based on the updated phase.

2. Phase Error Calculation:

- phase_error(i) = input_signal(i) - vco_output(i);

  Calculates the phase error as the difference between the input signal and the VCO output.

3. Control Voltage Calculation:

- control_voltage(i) = control_voltage(i-1) + Kpd * phase_error(i) * T;

Updates the control voltage based on the previous control voltage and the current phase error, simulating a proportional controller.

4. VCO Frequency Update:

- f_vco = 4 + Kvc * control_voltage(i);

Updates the VCO frequency based on the control voltage, ensuring it stays at a reasonable value.

Summary:

This code effectively simulates a basic PLL system, demonstrating how the VCO adjusts its frequency to minimize the phase error between the input signal and its own output. The parameters control the behavior and dynamics of the PLL, while the variables track the state of the system throughout the simulation.

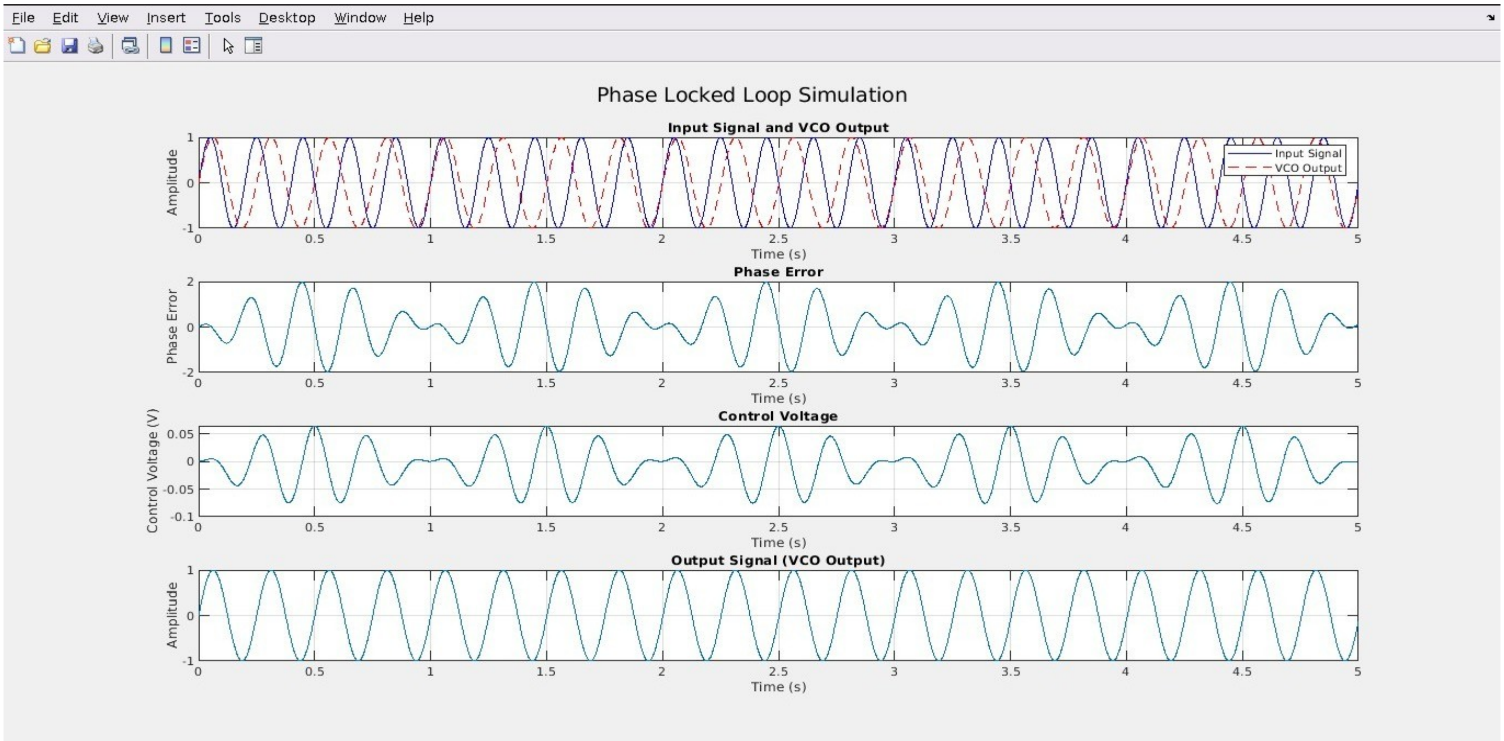


fig: Phase Locked loop simulation

Graph Explanation:

This figure shows the simulation results for a Phase-Locked Loop (PLL) system, broken down into four plots, each representing a different aspect of the PLL's operation:

1. Top Plot (Input Signal and VCO Output)

This plot shows two waveforms: the input signal (likely the reference signal for the PLL) and the output of the Voltage-Controlled Oscillator (VCO). The VCO output tries to match the frequency and phase of the input signal. As the PLL locks, you can observe that the two signals become closely aligned in both frequency and phase over time.

2. Second Plot (Phase Error)

This plot shows the phase difference (or error) between the input signal and the VCO output. Initially, the phase error may vary significantly, but as the PLL locks, the error stabilizes and oscillates around a smaller amplitude. A lower phase error indicates that the PLL is locked and the VCO output is synchronized with the input signal.

3. Third Plot (Control Voltage)

This plot shows the control voltage signal fed to the VCO. The PLL uses this voltage to adjust the VCO frequency, effectively minimizing the phase error. You may notice that the control voltage fluctuates initially but then stabilizes as the PLL locks. The variations in this voltage indicate the adjustments the system makes to synchronize the VCO with the input signal.

4. Bottom Plot (Output Signal - VCO Output)

This plot shows the final VCO output signal. Ideally, after lock-in, this output matches the frequency and phase of the input signal shown in the top plot. The VCO output represents the PLL’s final synchronized output signal, which, once locked, follows the input signal’s characteristics closely.

Together, these plots show how the PLL works to bring the VCO output into phase alignment with the input signal, effectively locking onto its frequency and phase.

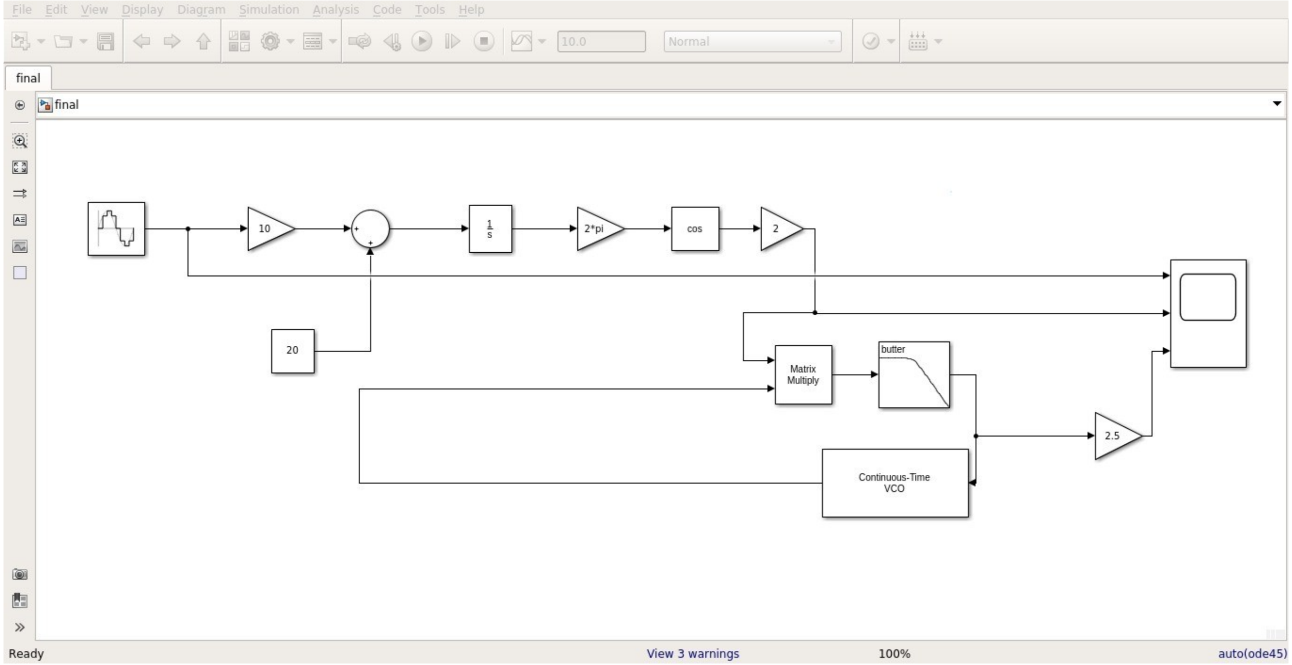
CHAPTER 5

APPLICATIONS AND CONCLUSION

5.1Applications of PLL:

The demand of the PLL circuit increases day by day because of its wide application in the area of electronics, communication and instrumentation. The recent applications of the PLL circuits are in memories, microprocessors, hard disk drive electronics, RF and wireless transceivers, clock recovery circuits on microcontroller boards and optical fiber receivers. Some of the PLL applications are mentioned below.

1.Frequency Demodution:



This Simulink model represent a frequency demodulation system using a Phase-Locked Loop (PLL).

Blocks:

1. Input Signal: The input is a modulated signal whose frequency varies with the information being transmitted. This signal is fed into the loop for demodulation.

2. Gain Block (10): This block scales the input signal by a factor of 10. Scaling helps in adjusting the signal's amplitude to be compatible with the rest of the blocks in the loop.

3. Adder Block (+): The adder combines the scaled input signal with a feedback signal from the VCO (Voltage Controlled Oscillator). This step is part of the feedback control mechanism that adjusts the VCO frequency to match the input signal's frequency.

4. Integrator Block (1/s): The integrator accumulates the phase of the signal. In PLL systems, the integrator effectively transforms frequency information into phase information, which helps in tracking the input signal frequency.

5. Multiplication by (2 \pi): This block converts the frequency in hertz to angular frequency (radians per second). It is necessary to keep the units consistent in the PLL loop.

6. Cosine Block (cos): This block calculates the cosine of the integrated signal. It likely serves as a phase reference signal to drive the feedback loop, ensuring that the PLL can lock onto the input frequency by phase comparison.

7. Gain Block (2): Another scaling operation, likely to ensure the feedback signal is at an appropriate level for comparison with the input signal.

8. Matrix Multiply Block: This block multiplies the output of the feedback with the signal fed to the low-pass filter. This could be to extract the frequency deviation component.

9. Butterworth Filter Block (butter): This low-pass filter removes high-frequency components, isolating the frequency deviation from the input signal. This component is crucial in filtering out noise and obtaining the demodulated signal.

10. Continuous-Time VCO Block: The VCO adjusts its output frequency based on the control voltage from the feedback loop. In a PLL, the VCO generates a frequency that attempts to match the input frequency. The feedback from the VCO ensures that the loop remains locked to the input signal’s frequency changes.

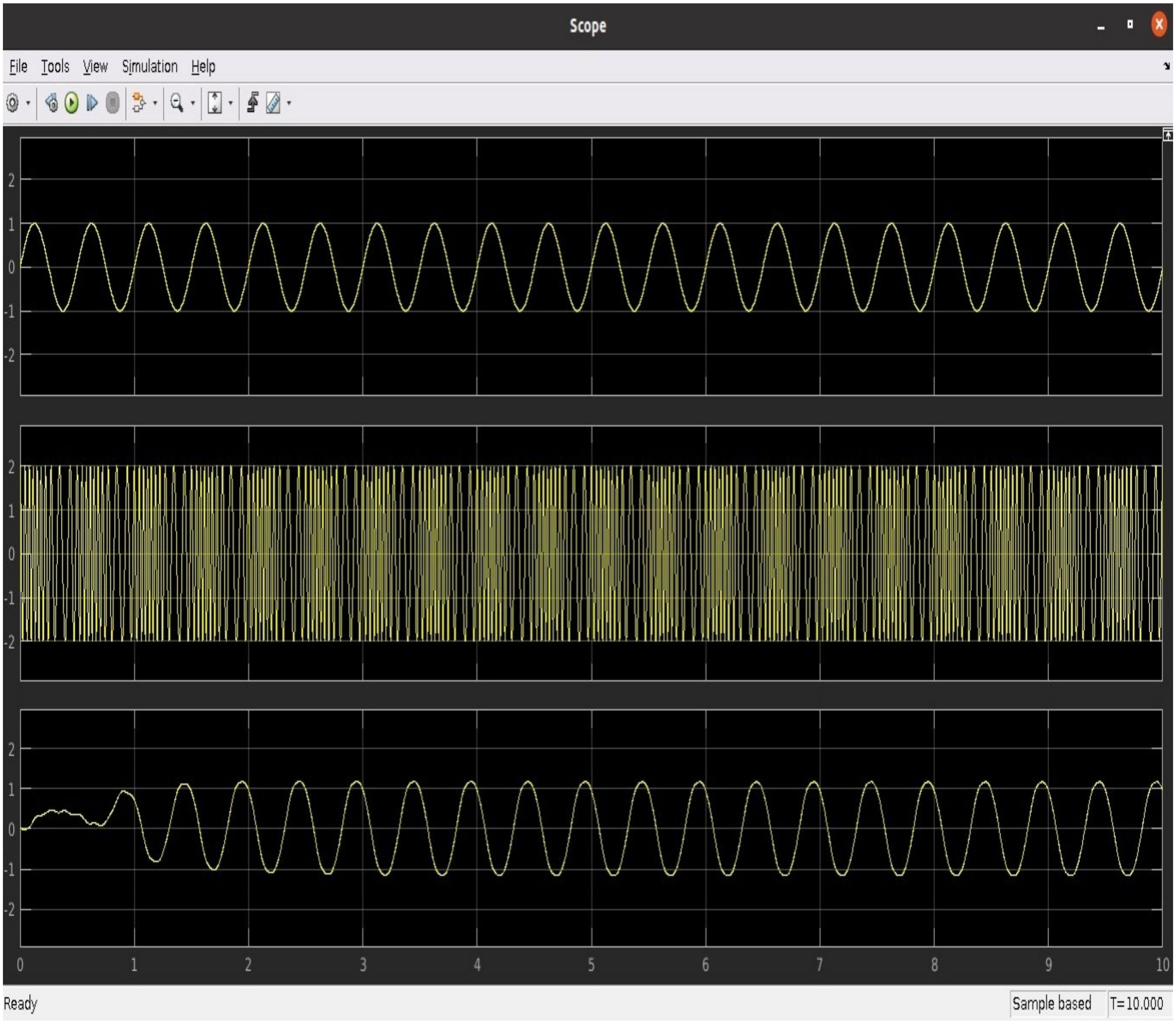
11. Output Display Block (Scope): The scope shows the demodulated output signal, which should correspond to the original information signal.

12. Gain Block (2.5): This final scaling block might be used to adjust the amplitude of the demodulated signal for display or further processing.

Summary

This model implements a PLL-based frequency demodulation system where the loop tracks the input frequency through feedback control. As the input frequency varies, the loop adjusts the VCO to match it, generating a phase-shifted signal proportional to the input frequency changes. The low-pass filter extracts the low-frequency information from the feedback loop, yielding the demodulated signal.

Plot:



2.Frequency Synthesis:

A frequency synthesizer is an electronic system for generating a range of frequencies from a single fixed time base or oscillator.

3. Clock Generation

Many electronic systems include processors of various sorts that operate at hundreds of megahertz. Typically, the clocks supplied to these processors come from clock generator PLLs, which multiply a lower-frequency reference clock (usually 50 or 100 MHz) up to the operating frequency of the processor. The multiplication factor can be quite large in cases where the operating frequency is multiple GHz and the reference crystal is just tens or hundreds of megahertz.

4. Carrier Recovery (Clock Recovery)

Some data streams, especially high-speed serial data streams (such as the raw stream of data from the magnetic head of a disk drive), are sent without an accompanying clock. The receiver generates a clock from an approximate frequency reference, and then phase-aligns to the transitions in the data stream with a PLL. This process is referred to as clock recovery.

5. Skew Reduction

This is one of the very popular and earliest uses of PLL. Suppose synchronous pair of data and clock lines enter a large digital chip. Since clock typically drives a large number of transistors and logic interconnects, it is first applied to large buffer. Thus, the clock distributed on chip may suffer from substantial skew with respect to data. This is an undesirable effect which reduces the timing budget for on-chip operations.

6.. Jitter and Noise Reduction

One desirable property of all PLLs is that the reference and feedback clock edges be brought into very close alignment. The average difference in time between the phases of the two signals when the PLL has achieved lock is called the static phase offset. The variance between these phases is called tracking jitter. Ideally, the static phase offset should be zero,and the tracking jitter should be as low as possible.

5.2CONCLUSION:

we can observe various signals and plots that demonstrate the behaviour and performance of PLL system.These outputs(plots) may include:

1. **Error signals**(plots): plots showing error signals generated by phase detector indicating how well PLL is adjusting output to match input.
2. **Step response:**Response of PLL to sudden change in input which shows how quickly system adjust and maintain synchronization.
3. Phase and frequency synchronisation.
4. Voltage-controlled oscillator(VCO).