Practical Workbook
CS-215
Signals and Systems

Name :

Year :

Batch :

Roll No :

Department:

Department of Computer & Information Systems Engineering
NED University of Engineering & Technology

Practical Workbook
CS-215
Signals and Systems

Prepared by:
Dr. Saad Qasim Khan
Syeda Ramish Fatima

Revised in:
February 2019

Department of Computer & Information Systems Engineering
NED University of Engineering & Technology

INTRODUCTION

The workbook has designed for the students of Department of Computer and Information Systems Engineering, NED University of Engineering and Technology enrolled in the subject of Signals and Systems. This course allows the students to analyze in depth the basic idea of signal and systems encountered in engineering and to make the students able to think various dimensions . The new version of the workbook is made to accommodate the requirement of Objective Based Education (OBE). Four lab sessions are directly mapped on CLO-3 '**Practice** simulation of signals and systems using modern tools'. The taxonomy level is P-3. The CLO has been mapped to PLO-5, 'modern tool usage' which includes lab session 6, 7, 8 and 9.

The workbook is divided in three basic sessions. The first session of the workbook which is from lab session 1 to lab session 5 is to aid student to understand signal processing on MATLAB software. The second session is to design electronic circuits for signal processing. This session is mapped onto CLO-3 as stated above. The third and final session is related to Simulink models for basic signal processing.

CONTENTS

Lab Session No.	Title	Page No	Teacher Signature	Date
1	Explore characteristic of various signals using MATLAB	1		
2	Apply elementary operations on signals	9		
3	Examine the aliasing effects on continues time sinusoidal signals	15		
4	Decompose a periodic signal into sum of simple oscillating functions	18		
5	Explore correlation between various signals	21		
6	Implement a passive low-pass filter using discrete components	25		
7	Implement an inverting signal amplifier using discrete components	29		
8	Convert an analog signal to digital signal	33		
9	Implement an active Low-Pass filter using discrete components	39		
10	Implement a band pass filter using switched capacitor filter	43		
11	Signal Processing in Simulink	47		
12	Frequency processing models in Simulink	51		
13	Simulating transfer function of LTI system in Simulink	55		
14	Integration and Differentiation Operations on a Signal	59		
	OBE Evaluation Sheets	62		

Lab Session 01

Explore characteristic of various signals using MATLAB

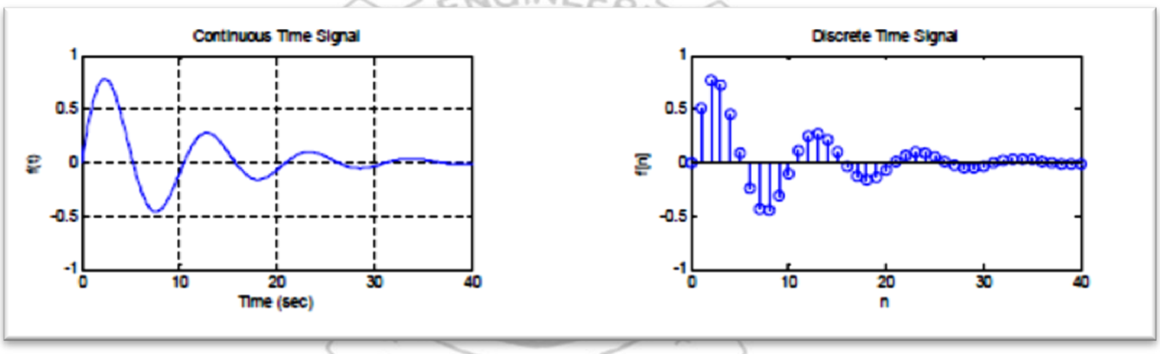
Signals

In the fields of communications, signal processing, and in electrical engineering more generally, a signal is any time- varying or spatial- varying quantity. This variable (quantity) changes in time.

• Speech or audio signal: Sound amplitude that varies in time
• Temperature readings at different hours of a day
• Stock price changes over days etc.

Signals can be classified by continues- time signal and discrete- time signal:

- A discrete signal or discrete- time signal is a time series, perhaps a signal that has been sampled from continuous time signal.
- A digital signal is a discrete- time signal that takes on only a discrete set of values.



MATLAB

Matrix Laboratory (MATLAB) is a powerful high-level programming language for scientific computations. It supports a rich suite of mathematical, statistical and engineering functions and its functionality is extended with interactive graphical capabilities for creating 2D as well as 3D plots. It provides comprehensive toolboxes and various sets of algorithms.

Default Desktop Environment

a) Command Window

The main window in which commands are keyed in after the command prompt ‘>>’.
Results of most printing commands are displayed in this window.

b) Command History Window

This window records all of the executed commands as well as the date and time when these commands were executed. This feature comes very handy when recalling previously executed commands. Previously entered commands can also be re-invoked using up arrow key.

c) Current Directory Window
This window keeps track of the files in the current directory.

d) Workspace
This window is used to organize the loaded variables and displays the information such as size and class of these variables.

Signal's Representation

A signal in MATLAB is represented by a vector:
Examples:

```
x = [2, 3, -5, -3, 1]
n = 2:3:17  %( here step size is 3)
n = 2:17  %( Default Step size 1 is used)
```

Plotting in MATLAB

While plotting in MATLAB one must be careful that a vector is plotted against a vector and lengths of vectors must match. Two functions are used for plotting:

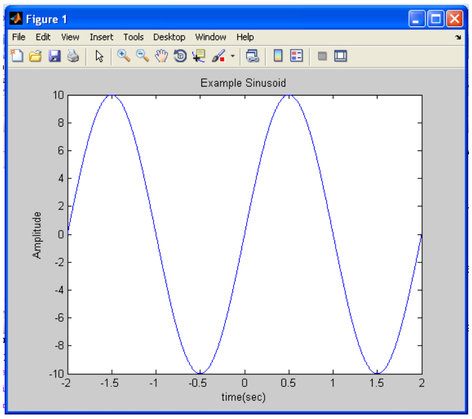
- `plot` (for CT signals)
- `stem` (for DT signals)

Example:

$x = 10\sin\pi t$

MATLAB Commands:

```
t = [-2:0.002:2]
x = 10 * sin (pi * t)
plot(t, x)
title('Example Sinusoid')
xlabel('time(sec)')
ylabel('Amplitude')
```



Multiple Plots

For drawing multiple signals on the same graph, write first signal's x and y axis vectors followed by the next signal.

Syntax:

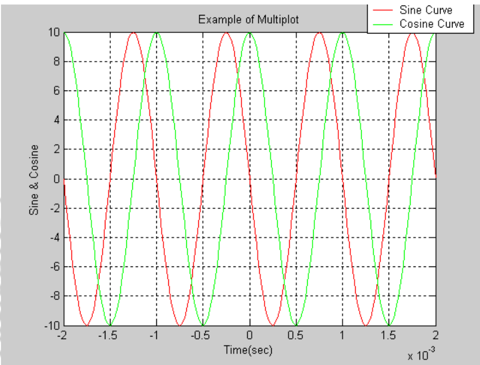
```
plot(X1,Y1,…,Xn,Yn)
```

In order to differentiate them by colors, write line style specifier and color code.

Colours		Line Styles	
y	yellow	.	point
m	magenta	o	circle
c	cyan	x	x-mark
r	red	+	plus
g	green	-	solid
b	blue	*	star
w	white	:	dotted
k	black	-.	dashdot
		--	dashed

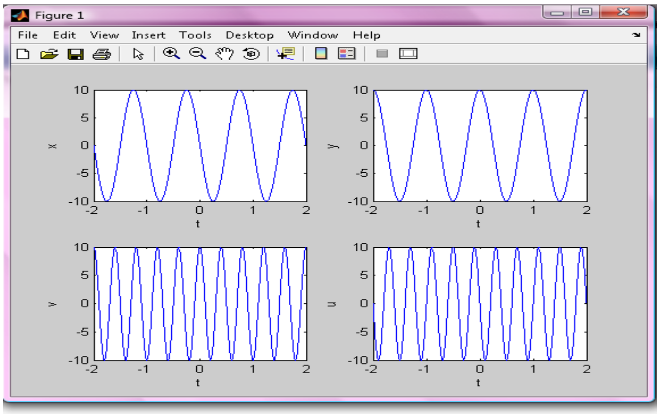
Example

```
plot(t, y, 'r-', t, x, 'g-');
legend('Sine curve', 'Cosine curve');
```



Generating Subplots

```
x=10*sin(-2*pi*t)
y=10*cos(-2*pi*t)
u=10*sin(-5*pi*t)
v=10*cos(-5*pi*t)
t = [-2:0.002:2]
subplot(2, 2, 1), plot(t, x);
xlabel('t'),ylabel('x');
subplot(2, 2, 2), plot(t, y);
xlabel('t'),ylabel('y');
subplot(2, 2, 4), plot(t, u);
xlabel('t'),ylabel('u');
subplot(2, 2, 3), plot(t, v);
xlabel('t'), ylabel('v');
```



DT Plots

Example:

Plot the DT sequences:

```
x = [2, 3, -1, 5, 4, 2, 3, 4, 6, 1]
x = [2, 3, -1, 5, 4, 2, 3, 4, 6, 1]
n = -6:3;
stem(n, x);
```

Zero & One Vectors

To generate zero or one vectors, use following statements:

```
zeros(1, 5)
Output: [0 0 0 0 0]
ones(1, 5)
Output: [1 1 1 1 1]
```

Some Common Types of Signals

For unit step:

```
x=(ones(size(n))).*(n>=0)
```

Note: *size(n)* returns dimension and number of elements in the array

For ramp:

```
x=n.*(n>=0)
```

For exponential:

```
x=(ones(size(n))).*(n>=0) %unit step
y=((0.5).^n).*x
stem(n,y)
```

For Rectangle:

```
t=-1:0.001:1;
y=rectpuls(t);
plot (t,y);
```

Triangle:

```
t=-1:0.001:1;
y=tripuls(t);
plot (t,y);
```

Sawtooth:

```
fs = 10000;
t = 0:1/fs:1.5;
x = sawtooth(2*pi*50*t);
plot(t,x), axis([0 0.2 -1 1]);
```

Square wave:

```
t=0:20;
y=square(t);
plot(t,y)
```

Sinc function:

```
t = -5:0.1:5;
y = sinc(t);
plot(t,y)
```

EXERCISES

1. Write MATLAB code to plot function x = sin(nπx). Generate 8 subplots using for loop. Use step size of 0.05.

2. Write a sequence of MATLAB commands in the space below to plot the curves ***y = cosx*** and ***y = x*** for $0 \le x \le 2$ on the same figure. Then `Zoom in` to determine the point of intersection of the two curves (and, hence, the root of x = cosx) to two significant figures. Your plot must be properly labeled.

3. Draw graphs of the functions for x =0:0.1:10 and label your graph properly.

i. y = sin(x)/x

ii. $u = (1/(x-1)^2)+x$

iii. $v = (x^2+1)/(x^2-4)$

iv. $z = ((10-x)^{1/3}-1)/(4 - x^2)^{1/2}$

4. Write MATLAB commands to plot following elementary DT signals. Properly label your graphs.
   a) Unit Step
   b) Unit Ramp
   c) Real Exponential: $x(n) = 2(0.25)n$, $0 < n < 10$

5. Generate multiple plots with the following data:
Suppose A=1, f=1Hz, t=0:0.01:1: $y(t)=\cos(2\pi t)$ $y(t)=\cos(2\pi t+\pi/2)$ $y(t)=\cos(2\pi t-\pi/2)$ $y(t)=\sin(2\pi t)$ where A is the amplitude of signal. Use colors & line styles to distinguish the plots.

This page is intentionally left blank.

Lab Session 02

Apply elementary operations on signals

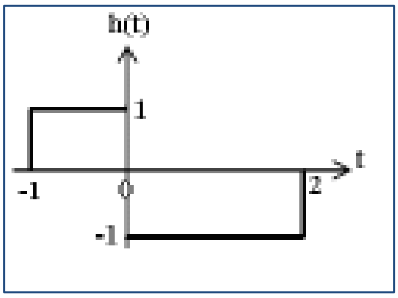
Operation with signals means to add, subtract, multiply, divide, scale, exponentiation, shift, delay or advance and to flip signals. MATLAB allows all these operations but we need to be careful in computation because the vector representation of the signals should have the same time origins and the same number of elements.

Basic Signal Operations:
Given the signals x1 and x2 perform the following operations:
$y1 = x1 + x2$; $y2 = x1 - x2$; $y3 = x1 * x2$; $y4 = x1 / x2$; $y5 = 2\ x1$; $y6 = (x1)^3$

Given is MATLAB code for first six basic operations on sinusoids:

```
x1 = 5*sin((pi/4)*[0:0.1:15]);
x2 = 3*cos((pi/7)*[0:0.1:15]);
% Plotting the signals
subplot(2,4,1), plot(x1)
title('x1 = 5 sin(pi/4)t ')
xlabel(' time (sec) ')
ylabel('x1 (volts) ')
subplot(2,4,2), plot(x2)
title('x2 = 3 cos(pi/7)t ')
xlabel(' time (sec) ')
ylabel('x2 (volts) ')
y1 = x1 + x2; % addition
y2 = x1 - x2; % subtraction
y3 = x1 .* x2; % multiplication
y4 = x1 ./ x2; % division
y5 = 2*x1; % scaling
y6 = x1 .^3; % exponentiation
% Plotting the signals
subplot(2,4,3), plot(y1)
title('y1 = x1 + x2 ')
xlabel(' time (sec) ')
ylabel('y1 (volts) ')
subplot(2,4,4), plot(y2)
title('y2 = x1 - x2 ')
xlabel(' time (sec) ')
ylabel('y2 (volts) ')
subplot(2,4,5), plot(y3)
title('y3 = x1 * x2 ')
xlabel(' time (sec) ')
ylabel('y3 (volts)^2 ')
subplot(2,4,6), plot(y4)
title('y4 = x1 / x2 ')
xlabel(' time (sec) ')
ylabel('x1/x2 ')
subplot(2,4,7), plot(y5)
title('y5 = 2*x1 ')
xlabel(' time (sec) ')
ylabel('y5 (volts) ')
subplot(2,4,8), plot(y6)
title('y6 = x1 ^ 3 ')
xlabel(' time (sec) ')
ylabel('y6 (volts)^3 ')
```

Plotting Piecewise Signals



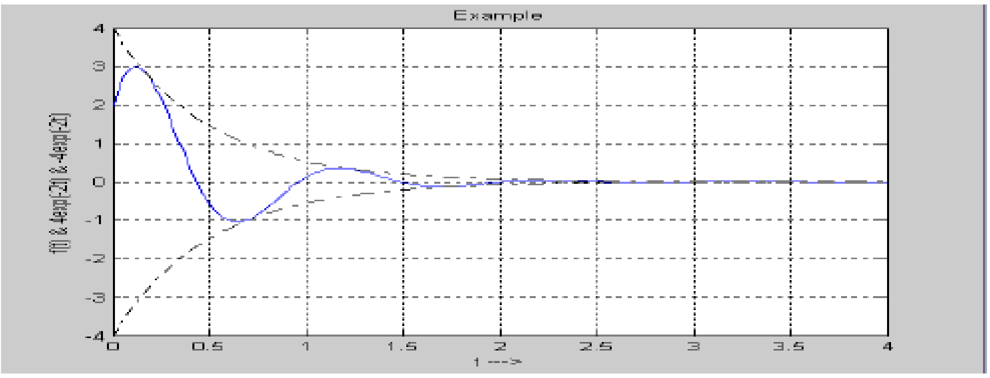
Define a piecewise continuous function h(t)

```
t = [-1:0.01:2];
h=1*[t >= -1 & t <= 0];
h=h+[-1 * (t > 0 & t <= 2)];
plot(t,h,'linewidth',2)
grid
axis([-2 3 -1.2 1.2])
```

Exponentially Varying Sinusoids

Consider the following example:

$$f(t) = 4e^{-2t}\cos(6t - 60^\circ) \qquad 0 \le t \le 4$$



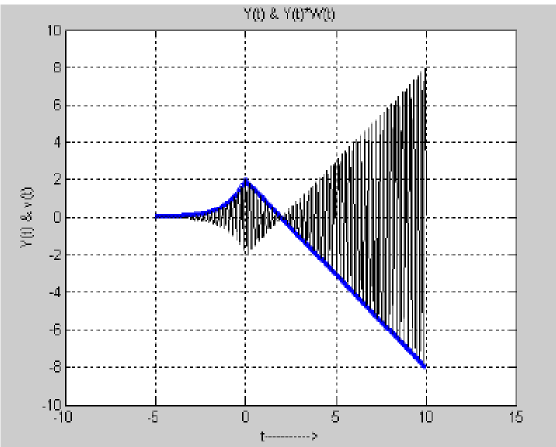
Multiplication of two sinusoids

Consider the following signal

$$Y(t) = \begin{cases} 2e^{t} & -5 \le t \le 0 \\ -t+2 & 0 \le t \le 10 \end{cases}$$

$$W(t) = \cos(30t)$$

$$V(t) = Y(t) * W(t)$$



More on Signal Operations

Time Shifting

One of the most basic operations in a DSP system is to shift the time reference. The shift may be considered either a delay or an advance.

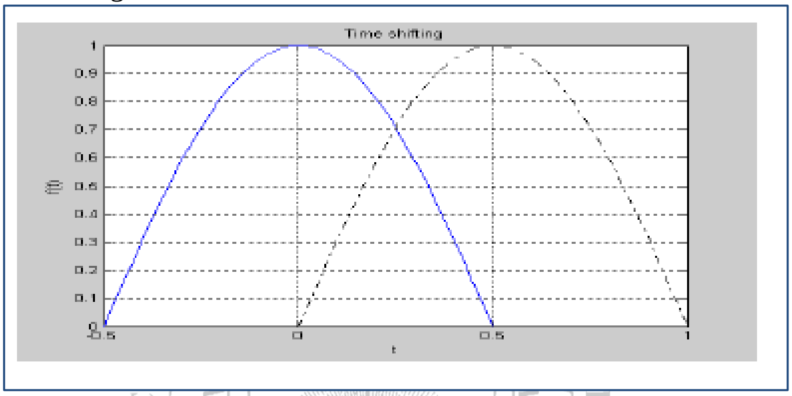
MATLAB function for Signal Shifting:

```
function [y, n] = sigshift(x, n, k)
n = n + k;
y = x;
```

Example:

$$x(t) = cos(\pi t) \qquad -0.5 \leq t \leq 0.5$$

Shift x (t) by 0.5 units on the right



Time Scaling

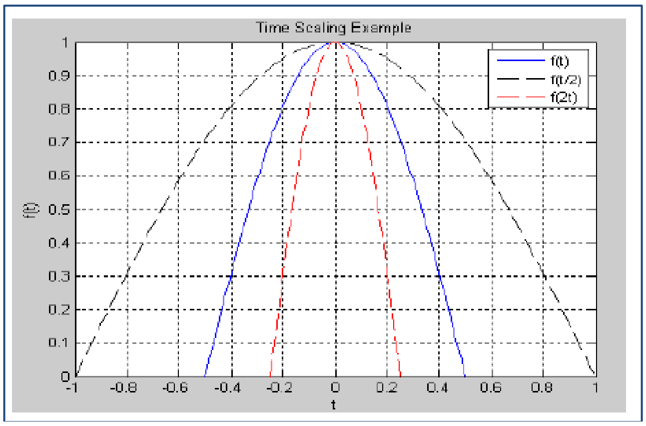
If f(t) is compressed in time by a factor of 'a' where (a>1) then the resulting signal is given by:

ϴ (t) = f(at)

Similarly if f(t) is expanded in time by a factor of 'a' where (a>1) then the resulting signal is given by:

ϴ (t) = f(t/a)

If x = cos(πt); for t = -0.5:0.01:0.5
Find: **ϴ (t) = x(2*t) & ϴ (t) = x(t/2)**

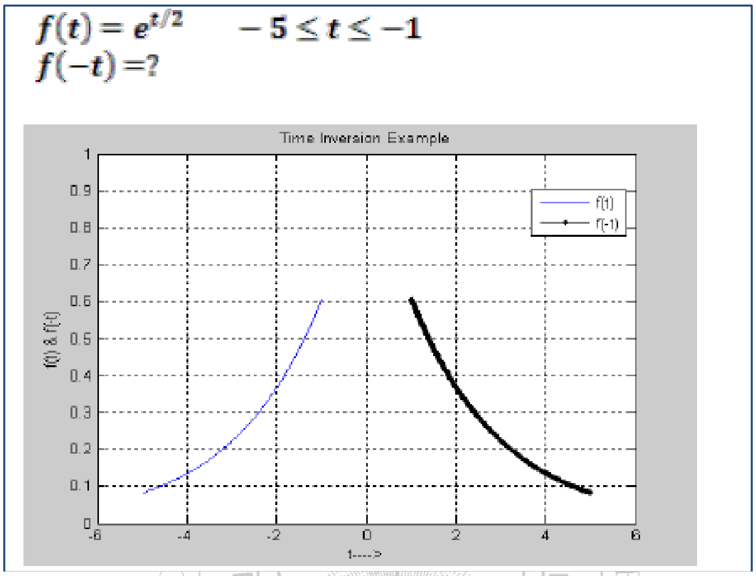


Time Inversion (Folding)
Folding involves the reversal of the time axis.

$$\mathbf{\Theta\ (t) = f(-t)}$$
$$\mathbf{\Theta\ (-t) = f(t)}$$

To invert a signal we replace t by –t.

Consider the following example:



Operations on DT Signals

Signal Addition
This is sample-by-sample addition given by:
$\{x1(n)\}+\{x2(n)\}=\{ x1(n) + x2(n)\}$
Implemented in MATLAB by '+' operator. However, the lengths of *x1*(*n*) and *x2*(*n*) must be same.

Signal Multiplication
This is sample-by-sample multiplication given by:
$\{x1(n)\}\mathbf{.}\{x2(n)\}=\{ x1(n)\, x2(n)\}$
It is implemented in MATLAB by '*' operator. However, the lengths of *x1*(*n*) and *x2*(*n*) must be same.

Delayed (Advanced) Impulse
Formula used is: {
MATLAB function
```
function [y, n] = impseq(n1, n2, k)
n = n1:n2;
y = [(n - k) == 0];
```

Time Shifting, Inversion & scaling are also performed sample by sample on DT Signals.

EXERCISES

1. Write down MATLAB code to perform the signals operations discussed in lab.

2. The operations discussed in lab can also be performed on DT signals (sample by sample). Write MATLAB scripts to generate and plot each of the following signals for the intervals indicated.

a) $x(n) = n^2[u(n+5) - u(n-6)] + 10\delta(n) + 20(0.5)^n[u(n-4) - u(n-10)]$ $\quad -10 \le n \le 10$

b) $x(n) = (0.2)^n[u(n-5) + u(n+6)]\delta(n-5) + 20(0.5)^n[u(n+4)] \qquad -10 \le n \le 10$

Lab Session 03

Examine the aliasing effects on continuous-time sinusoidal signals

Sampling Principle

A CT sinusoid containing a maximum frequency of Fmax must be sampled at a sampling rate of Fs > 2Fmax (Nyquist Rate) to avoid aliasing. If sampling rate is greater than Nyquist Rate, then CT sinusoid can be uniquely recovered from its DT version.
Analog frequencies separated by integral multiple of a given sampling rate are alias of each other
Any CT sinusoid of frequency Fk when sampled at the sampling rate Fs will result in the same DT sinusoid as does the CT sinusoid of frequency F0 sampled at Fs, where:

$$Fk = F0 + kFs \text{ where } k=\pm 1, \pm 2, \pm 3 \ldots$$

Assume two CT sinusoidal signals

$$x1(t) = \cos(2\pi t) \quad F1 = 1 \text{ Hz}$$
$$x2(t) = \cos(6\pi t) \quad F2 = 3 \text{ Hz}$$

These signals can be plotted using the MATLAB code shown:

```
t = -2:0.005:2;
x1 = cos(2*pi*t);
x2 = cos(6*pi*t);
subplot(3,2,1),
plot(t, x1);
axis([-2 2 -1 1]);
grid on;
xlabel('t'), ylabel('cos2\pit');
subplot(3,2,2),
plot(t, x2);
axis([-2 2 -1 1]);
grid on;
xlabel('t'), ylabel('cos6\pit');
```

Plotting DT Sinusoid

```
x1n = cos(2*pi*[-2:1/2:2]); %We will not plot it against n because this will depict
sampled signal incorrectly. Generate another vector from n as follows:
k = -2:length(n)-3;
subplot(3,2,3), stem(k, x1n);
axis([-2 length(n)-3 -1 1]);
grid on;
xlabel('n'),
ylabel('cos\pin');
X2n= cos(6*pi*n)
subplot(3,2,4),stem(k, x2n);
axis([-2 length(n)-3 -1 1]);
grid on;
xlabel('n'),
ylabel('cos3\pin');
```

Reconstruction

D/A conversions are performed using interpolation. There are various approaches to interpolation
◦ Zero-Order-Hold (ZOH)
Step interpolation

◦ First-Order-Hold (FOH)
Samples are connected by straight lines
◦ Cubic Spline Interpolation
It is invoked by spline (n, xn, t)

EXERCISES

1. Write a MATLAB script to carry out the following tasks:
a. Plot of two continuous time sinusoids $x1(t) = \cos 2\pi t$ and $x2(t) = \cos 14\pi t$ for $0 < t < 5$ seconds. Choose a suitable time step.
b. Sample them using a sampling rate of 3 samples/s.
c. Plot the resulting the resulting discrete time sinusoids.
d. Reconstruct the signals
e. You must divide the figure into six subplots so that reconstructed signal and DT version of x1(t) lie in the same column and so are those of x2(t).

2. Does aliasing occur? Briefly explain why?

3. Consider an analog signal $xa(t) = \sin(20\pi t)$, $0 < t < 1$. It is sampled at sampling intervals of 0.01, 0.05, and 0.1 seconds to obtain x(n).
a. Write MATLAB scripts to plot x(n) for each sampling interval
b. Reconstruct the analog signal ya(t) from the samples x(n) using the cubic spline interpolation and determine the frequency in ya(t) from your plot. (Ignore the end effects)
c. Comment on your results.

Lab Session 04

Decompose a periodic signal into sum of simple oscillating functions

In mathematics, a **Fourier series** decomposes periodic functions or periodic signals into the sum of a (possibly infinite) set of simple oscillating functions, namely sines and cosines (or complex exponentials).

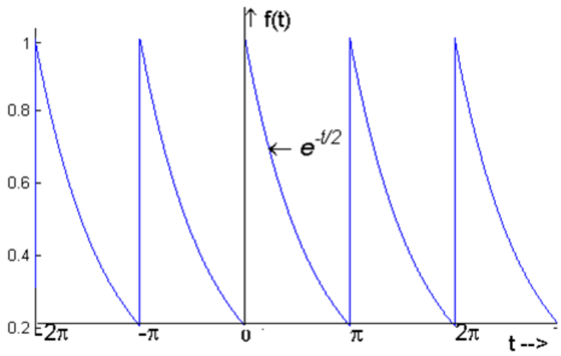
THEOREM:

Let x(t) be a bounded periodic signal with period T. Then x(t) can be expanded as a weighted sum of sinusoids with angular frequencies that are integer multiples of ω0 =2π/T:
X (t) = a0 + a1 cos(ω0t) + a2 cos(2ω0t) + a3 cos(3ω0t) + . . .+ b1 sin(ω0t) + b2 sin(2ω0t) + b3 sin(3ω0t) + . . .
This is the trigonometric Fourier series expansion of x(t).

$$f(t) = a_0 + \sum_{n=1}^{\infty}(a_n \cos\omega nt + b_n \sin\omega nt)$$

$$ao = 1/To \int f(t)dt$$
$$an = 2/To \int f(t) \cos\omega t\, dt$$
$$bn = 2/To \int f(t) \sin\omega t\, dt$$

Consider the given signal. Find the Trigonometric Fourier series coefficients and plot the magnitude and phase spectra for the periodic signals shown below in MATLAB:



Fourier series coefficients for the periodic signal shown above are:

$$ao = 0.504 \qquad an = 0.504\left(\frac{2}{1+16n^2}\right) \qquad bn = 0.504\left(\frac{8n}{1+16n^2}\right)$$

The code for plotting the magnitude and phase spectra for the given co-efficients is given below using for loop:

```
n=1:7;
a0=0.504;
b0=0.504*(8*0/(1+16*0^2)); % b0=0;
Cn=a0;
theta0=atan(-b0/a0);
thetan=theta0;
den=(1+16*n.^2);
N=length(den);
```

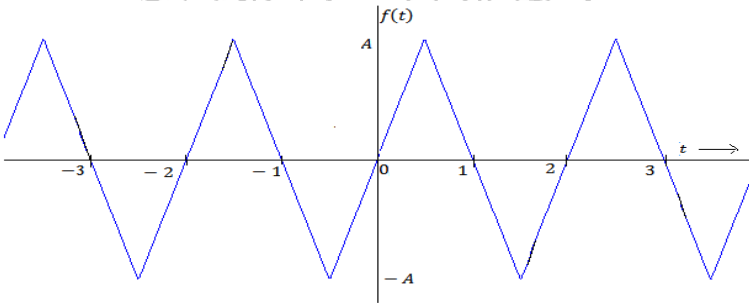
```
for i=1:N
an(i)=0.504*2/den(i);
bn(i)=0.504*8*n(i)/den(i);
cn=sqrt(an(i)^2+bn(i)^2);
Cn=[Cn cn];
theta=atan(-bn(i)/an(i));
thetan=[thetan theta];
end
n=0:7;
subplot(211),plot(n,,'o'),grid, xlabel('n'),ylabel('C_n'),title('Fourier Series')
subplot(212),plot(n,thetan,'o'),grid,xlabel('n'),ylabel('\theta_n (rad)')
```

EXERCISE

1. If f (t) is defined below:

$$f(t) = \begin{cases} 2At, & |t| \le \frac{1}{2} \\ 2A(1-t), & \frac{1}{2} < t \le \frac{3}{2} \end{cases}$$

Find the trigonometric Fourier series coefficients for the periodic signal shown below and use them to plot the magnitude and phase spectra.



Lab Session 05

Explore correlation between various signals

Correlation between two signals may be evaluated by using a mathematical function referred as convolution. There are two type of correlation in signals.

- Auto correlation. (Correlation between a signal and its delayed version).
- Cross correlation. (Correlation between two different signals).

What is Convolution?

Convolution is a mathematical method for combining two signals to form a third signal. It is one of the most important techniques in Signal Processing. Convolution is important because it relates the three signals of interest: the input signal, the output signal, and the impulse response.

Expression for discrete time convolution;

$$y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k].$$

Where; y: output signal
x: input signal
h: impulse response of system

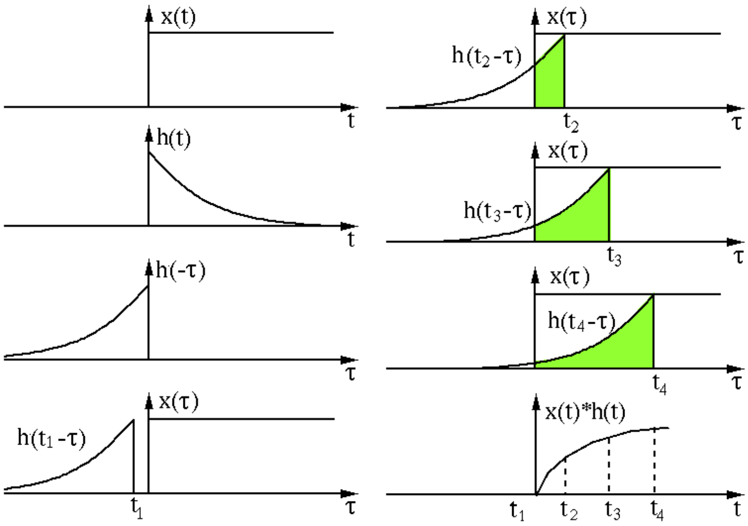
Impulse Response of a System:

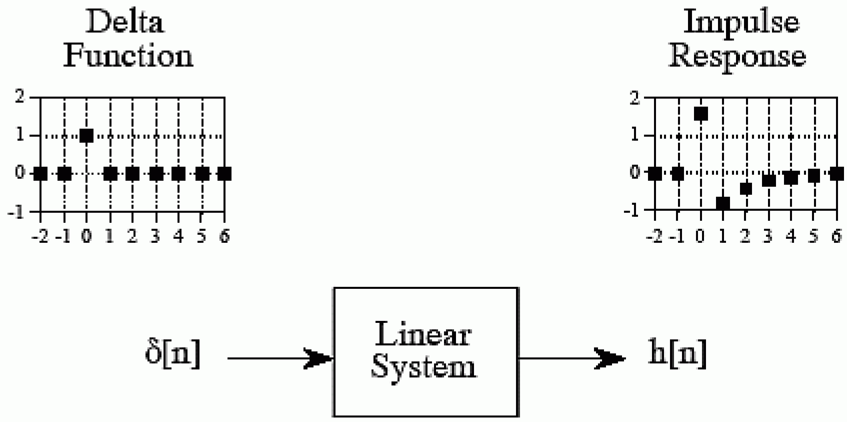
Impulse response of a system can be regarded as the transfer function of a system. If we know a system's impulse response, then we can calculate what the output will be for any possible input signal. This means we know *everything* about the system. There is nothing more that can be learned about a linear system's characteristics.

The impulse response goes by a different name in some applications. If the system being considered is a *filter*, the impulse response is called the **filter kernel**, the **convolution kernel**, or simply, the **kernel**. In image processing, the impulse response is called the **point spread function**. While these terms are used in slightly different ways, they all mean the same thing, the signal produced by a system when the input is a delta function.

How to evaluate the impulse response of a system?

First, the input signal can be decomposed into a set of impulses, each of which can be viewed as a scaled and shifted delta function. Second, the output resulting from each impulse is a scaled and shifted version of the impulse response. Third, the overall output signal can be found by adding these scaled and shifted impulse responses.

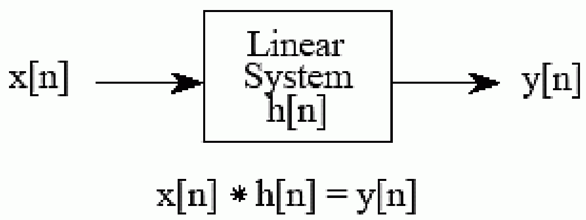




Definition of *delta function* and *impulse response*. The delta function is a normalized impulse. All of its samples have a value of zero, except for sample number zero, which has a value of one. The Greek letter delta, δ[*n*], is used to identify the delta function. The *impulse response* of a linear system, usually denoted by *h*[*n*], is the output of the system when the input is a delta function.

An input signal, *x*[*n*], enters a linear system with an impulse response, *h*[*n*], resulting in an output signal, *y*[*n*]. In equation form: *x*[*n*] * *h*[*n*] = *y*[*n*]. Expressed in words, the input signal convolved with the impulse response is equal to the output signal. Just as addition is represented by the plus, +, and multiplication by the cross, ×, convolution is represented by the star, *.

How convolution is used in DSP. The output signal from a linear system is equal to the input signal *convolved* with the system's impulse response. Convolution is denoted by a star when writing equations.



LAB PERFORMANCE

Convolution in MATLAB

The MATLAB command 'conv' computes the convolution of two vectors.

A single vector can be used to represent a discrete time signal with `t` distance between each sample. Two vectors represent two distinct discrete time signals.

Let two digital signals of different lengths are given as v1 = 1 0 1 1 0 1 1 0 0 and v2= 0 1 0 apply convolution and write down the resultant signal.

MATLAB Code:

```
v1 = [1 0 1 1 0 1 1 0 0];
v2 = [0 1 0];
u = conv(v1, v2);
```

In the above program the value of the resultant signal should be passed to the variable mentioned on the left hand side of equality sign where we have used the *conv* function.

Task: Create subplots for v1, v2 and u signals.

EXERCISES

1. Write MATLAB code to perform convolution between a square pulse [0 0 0 0 4 4 4 4 0 0 0 0] and a triangular wave [0 0 0 1 2 3 4 3 2 1 0 0]. Using stem instruction, generate subplots for the two given vectors and the resultant vector (Attach subplots to this lab session)

2. Write a sequence of MATLAB statements to compute convolution of signal x(t) and impulse response h(t) defined for times t = [-10,+10];

$$x(t) = \begin{cases} 1 & \text{for} \quad 0 \leq t \leq 2 \\ 0 & \text{elsewhere} \end{cases} \qquad h(t) = \begin{cases} 0.5 & \text{for} \quad 0 \leq t \leq 2 \\ 1 & \text{for} \quad 4 \leq t \leq 6 \\ 0 & \text{elsewhere} \end{cases}$$

3. Perform and write down MATLAB code of any program that employs convolution operations by the advice of lab instructor.

4. Write MATLAB commands to perform convolution between signal ‘a’ & ‘b’ for time interval [-20, +20]. Also do attach a printout of resultant signal.

a) Gaussian curve

b) Unit Ramp signal

Lab Session 06

Implement a passive low-pass filter using discrete components

This lab session is mapped on CLO-3 'Practice simulation of signals and systems using modern tools'. The taxonomy level is P-3. The CLO has been mapped to PLO-5, 'modern tool usage'.

Filters:

> *A filter is a circuit capable of passing (or amplifying) certain frequencies while attenuating other frequencies. Thus, a filter can extract important frequencies from signals that also contain undesirable or irrelevant frequencies.*

In circuit theory, a filter is an electrical network that alters the amplitude and/or phase characteristics of a signal with respect to frequency. Ideally, a filter will not add new frequencies to the input signal, nor will it change the component frequencies of that signal, but it will change the relative amplitudes of the various frequency components and/or their phase relationships. Filters are often used in electronic systems to emphasize signals in certain frequency ranges and reject signals in other frequency ranges.

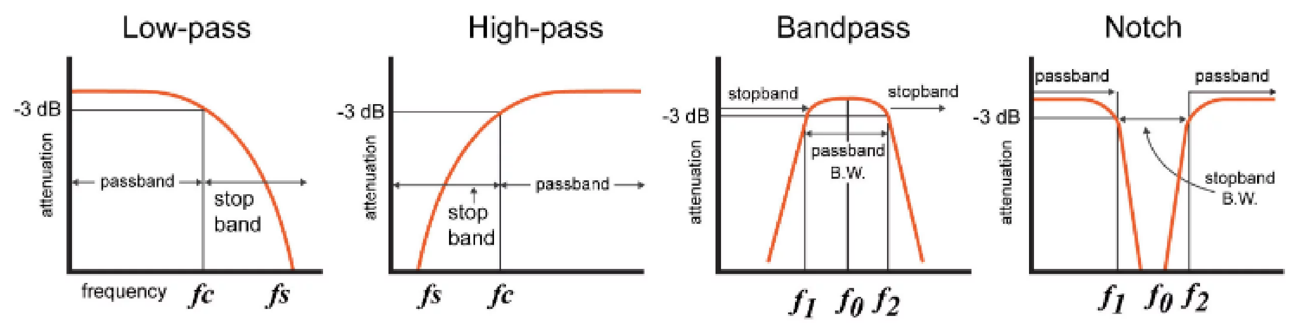
Applications of Filters:

In the field of electronics, there are many practical applications for filters. Examples include:

- ***Radio communications***: Filters enable radio receivers to only "see" the desired signal while rejecting all other signals (assuming that the other signals have different frequency content).
- ***DC power supplies***: Filters are used to eliminate undesired high frequencies (i.e., noise) that are present on AC input lines. Additionally, filters are used on a power supply's output to reduce ripple.
- ***Audio electronics***: A crossover network is a network of filters used to channel low-frequency audio to woofers, mid-range frequencies to midrange speakers, and high-frequency sounds to tweeters.
- ***Analog-to-digital conversion***: Filters are placed in front of an ADC input to minimize aliasing.

Response Curves:

Response curves are used to describe how a filter behaves. A response curve is simply a graph showing an attenuation ratio (V_{OUT} / V_{IN}) versus frequency Attenuation is commonly expressed in units of decibels (dB).



Passive Low Pass Filter:

In low frequency applications (up to 100kHz), passive filters are generally constructed using simple RC (Resistor-Capacitor) networks, while higher frequency filters (above 100kHz) are usually made from RLC(Resistor-Inductor-Capacitor) components.

Passive filters are made up of passive components such as resistors, capacitors and inductors and have no amplifying elements (transistors, op-amps, etc) so have no signal gain, therefore their output level is always less than the input.

Simple First-order passive filters (1st order) can be made by connecting together a single resistor and a single capacitor in series across an input signal, (V_{IN}) with the output of the filter, (V_{OUT}) taken from the junction of these two components.

Cut-off Frequency:

A Low Pass Filter can be a combination of capacitance; inductance or resistance intended to produce high attenuation above a specified frequency and little or no attenuation below that frequency. The frequency at which the transition occurs is called the "cut-off" or "corner" frequency.

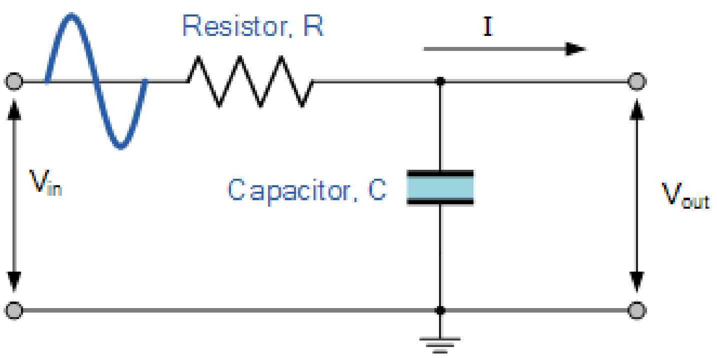
It is calculated as:

$$f_c = 1/\ 2\pi RC$$

Implement a passive low pass filter:

A simple passive RC Low Pass Filter or LPF, can be easily made by connecting together in series a single Resistor with a single Capacitor as shown below. In this type of filter arrangement the input signal (V_{IN}) is applied to the series combination (both the Resistor and Capacitor together) but the output signal (V_{OUT}) is taken across the capacitor only.

This type of filter is known generally as a "first-order filter" or "one-pole filter" because it has only "one" reactive component, the capacitor, in the circuit.



The reactance of a capacitor varies inversely with frequency, while the value of the resistor remains constant as the frequency changes. At low frequencies the capacitive reactance, (XC) of the capacitor will be very large compared to the resistive value of the resistor, R.

This means that the voltage potential, V_C across the capacitor will be much larger than the voltage drop, V_R developed across the resistor. At high frequencies the reverse is true with V_C being small and VR being large due to the change in the capacitive reactance value.

Components Required:

- Function Generator
- Breadboard
- Capacitor 100nF
- Resistor (Variable)
- Oscilloscope

Procedure:

1. Connect resistor and capacitor on breadboard as shown in the diagram.
2. Set Function Generator to 1V AC Sinusoid with frequency lesser than cut-off frequency and connect the positive and ground terminals across the circuit.
3. Connect the output terminals across capacitor to the oscilloscope to see the output signal. The filter will allow the signal to pass and will be seen un-attenuated on the oscilloscope screen.
4. Gradually increase the input signal frequency in the function generator to a value higher than cut-off frequency. The amplitude of the signal should begin to attenuate beyond the cut-off frequency till it fades away completely.

Observation:

Starting with frequency less than cut-off frequency, gradually increase the frequency of the input signal and give your remarks for what you see on the oscilloscope.

Exercise:

1. Implement a low pass filters with cutoff frequency 150 HZ and give your remarks.

2. Change the configuration of the circuit to implement a passive high-pass filter and give your remarks

Lab Session 07

Implement an inverting signal amplifier using discrete components

This lab session is mapped on CLO-3 'Practice simulation of signals and systems using modern tools'. The taxonomy level is P-3. The CLO has been mapped to PLO-5, 'modern tool usage'.

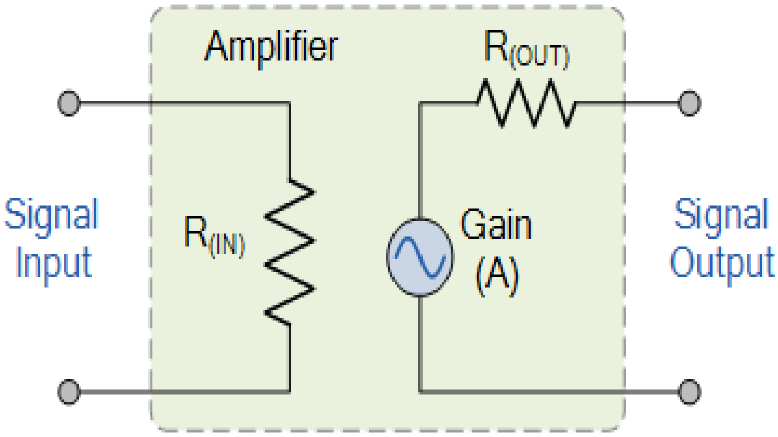
Amplifier:

Amplifier is the generic term used to describe a circuit which produces an increased version of its input signal. However, not all amplifier circuits are the same as they are classified according to their circuit configurations and modes of operation.

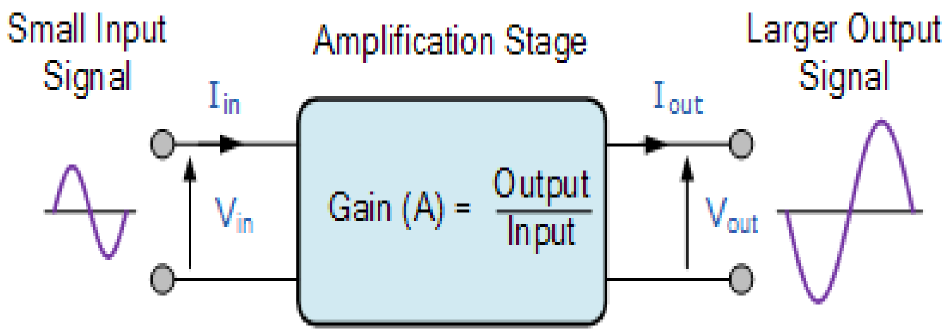
In "Electronics", small signal amplifiers are commonly used devices as they have the ability to amplify a relatively small input signal, for example from a Sensor such as a photo-device, into a much larger output signal to drive a relay, lamp or loudspeaker for example.

There are many forms of electronic circuits classed as amplifiers, from Operational Amplifiers and Small Signal Amplifiers up to Large Signal and Power Amplifiers. The classification of an amplifier depends upon the size of the signal, large or small, its physical configuration and how it processes the input signal, that is the relationship between input signal and current flowing in the load.

Amplifiers can be thought of as a simple box or block containing the amplifying device, such as a Bipolar Transistor, Field Effect Transistor or Operational Amplifier, which has two input terminals and two output terminals (ground being common) with the output signal being much greater than that of the input signal as it has been "Amplified".



The amplified difference between the input and output signals is known as the Gain of the amplifier. Gain is basically a measure of how much an amplifier "amplifies" the input signal. For example, if we have an input signal of 1 volt and an output of 50 volts, then the gain of the amplifier would be "50". In other words, the input signal has been increased by a factor of 50.



Characteristics of Ideal Amplifier

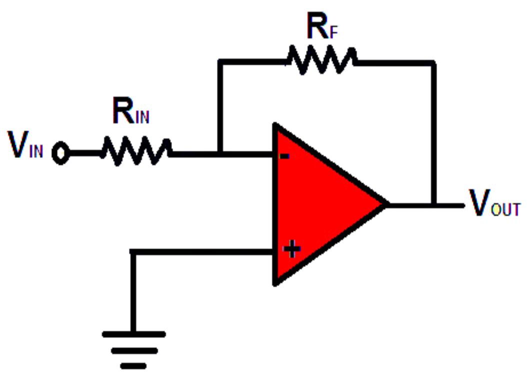
- The amplifiers gain, (A) should remain constant for varying values of input signal.
- Gain is not be affected by frequency. Signals of all frequencies must be amplified by exactly the same amount.
- The amplifiers gain must not add noise to the output signal. It should remove any noise that is already exists in the input signal.
- The amplifiers gain should not be affected by changes in temperature giving good temperature stability.
- The gain of the amplifier must remain stable over long periods of time.

Op-amp as an amplifier:

An operational amplifier (often op-amp or op-amp) is a DC-coupled high-gain electronic voltage amplifier with a differential input and, usually, a single-ended output. In this configuration, an op-amp produces an output potential (relative to circuit ground) that is typically hundreds of thousands of times larger than the potential difference between its input terminals.

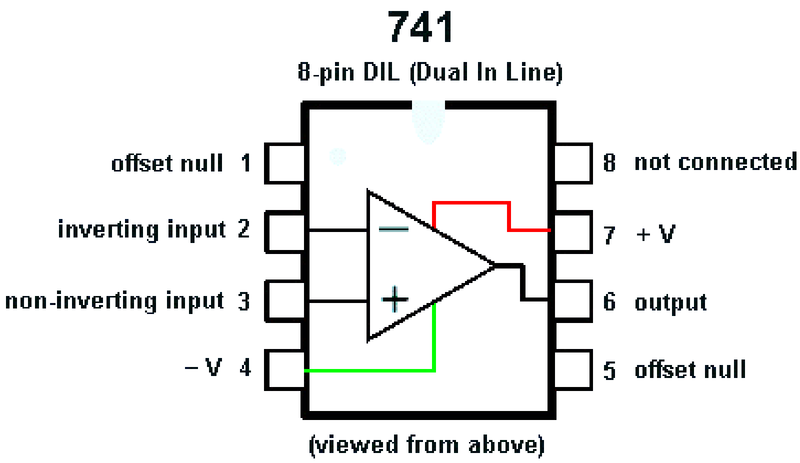
Closed-Loop Amplifier:

If predictable operation is desired, negative feedback is used, by applying a portion of the output voltage to the inverting input. The closed-loop feedback greatly reduces the gain of the circuit. When negative feedback is used, the circuit's overall gain and response becomes determined mostly by the feedback network, rather than by the op-amp characteristics. If the feedback network is made of components with values small relative to the op amp's input impedance, the value of the op-amp's open-loop response AOL does not seriously affect the circuit's performance.



The non-inverting circuit diagram is shown above and the gain of this non-inverting circuit is generally calculated by using the formula:

$$\mathbf{A = 1 + (R_f / R_{in})}$$



Components Required:

- Function Generator
- Breadboard
- Power Supply (+/- 12V)
- Op-amp IC (LM741)
- Resistors (Variable)
- Oscilloscope

Procedure:

1. Connect circuit components IC 741, and resistors on breadboard as shown in the diagram.
2. Connect $\mathbf{R_f}$ =100KΩ between pin 6 and 2.
3. Connect $\mathbf{R_{in}}$=25KΩ between Vin and pin 2.
4. Set Function generator to 1 Volt AC Sinusoid Signal.
5. Connect the output pin 6 and ground to the oscilloscope to see the output signal. The signal must be amplified by the factor of gain.
6. Change the amplitude of input signal and record the output amplitude for the gain calculated.

Observation

Gain= $1 + R_f / R_{in}$ = _______________

S.No.	V_{in}(Volts)	Expected V_{out}	Observed V_{out}
1.			
2.			
3.			

Exercise:

1. Change the values of R_f and R_{in} and repeat the above exercise.

Lab Session 08

Convert an analog signal to digital signal

This lab session is mapped on CLO-3 'Practice simulation of signals and systems using modern tools'. The taxonomy level is P-3. The CLO has been mapped to PLO-5, 'modern tool usage'.

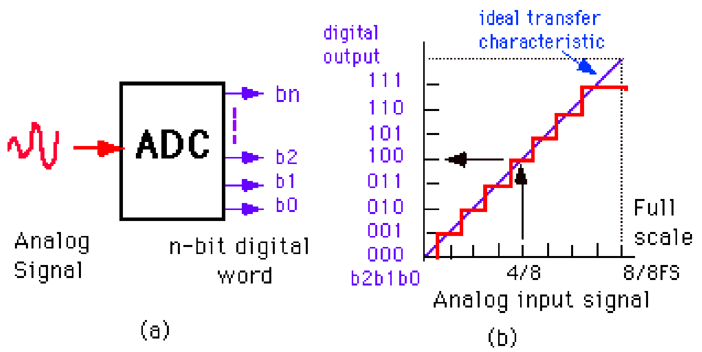
Analog-to-digital conversion is an electronic process in which a continuously variable (analog) signal is changed, without altering its essential content, into a multi-level (digital) signal.

The input to an **analog-to-digital converter (ADC)** consists of a voltage that varies among a theoretically infinite number of values. Examples are sine waves, the waveforms representing human speech, and the signals from a conventional television camera. The output of the ADC, in contrast, has defined levels or states. The number of states is almost always a power of two -- that is, 2, 4, 8, 16, etc. The simplest digital signals have only two states, and are called binary. All whole numbers can be represented in binary form as strings of ones and zeros.

Application of ADC

Digital signals propagate more efficiently than analog signals, largely because digital impulses, which are well-defined and orderly, are easier for electronic circuits to distinguish from noise, which is chaotic. This is the chief advantage of digital modes in communications. Computers "talk" and "think" in terms of binary digital data; while a microprocessor can analyze analog data, it must be converted into digital form for the computer to make sense of it.

A typical telephone modem makes use of an ADC to convert the incoming audio from a twisted-pair line into signals the computer can understand. In a digital signal processing system, an ADC is required if the signal input is analog.



Working

The ADC reports a ratio metric value. An 8-bit ADC can detect 2^8=256 distinct analog levels.

$$\frac{\text{Resolution of the ADC}}{\text{System Voltage}} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

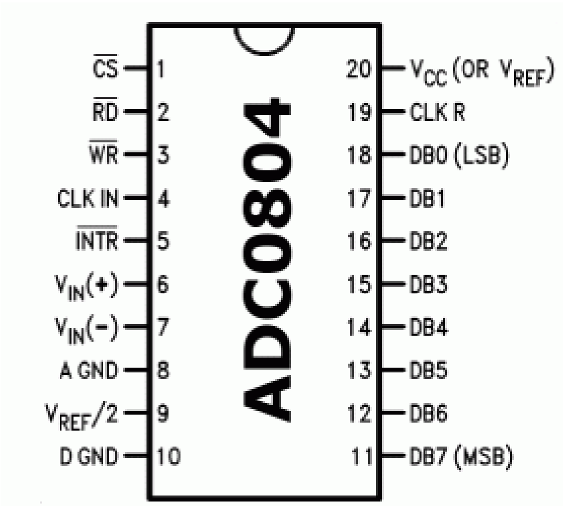
For an 8-bit ADC resolution is 255. If your system voltage is 5V and you want to measure the analog voltage say 2.12 V, ADC reading will be calculated as:

ADC Reading = 255/5*2.12 =108

Now we are going to see an ADC of 0804. It is an 8-bit converter with 5V of power supply. It can take only one analog signal as input.

The digital output varies from 0-255. ADC needs a clock to operate. The time taken to convert the analog to digital value depends on the clock source. An external clock can be given to CLK IN pin no.4. A suitable RC circuit is connected between the clock IN and clock R pins to use the internal clock. Pin2 is the input pin – High to low pulse brings the data from internal register to the output pins after conversion. Pin3 is a Write – Low to high pulse is given to external clock. Pin11 to 18 are data pins from MSB to LSB.

Normal operation is that when the logic high, input has been clocked through the 8-bit shift register, completing the SAR search, on the next clock pulse, the digital word is transferred to the 3-state output. The output of the interrupt is inverted to provide an INTR output that is high during conversion and low when the conversion is completed. When a low is at both CS and RD, an output is applied to the DB0 through DB7 outputs and the interrupt is reset. When either the CS or RD inputs return to a high state, the DB0 through DB7 outputs are disabled (returned to the high-impedance state). The interrupt flip-flop remains reset.



Pin Description of ADC0804:

Pin 1: It is a chip select pin and activates ADC, active low
Pin 2: It is an input pin; high to low pulse brings the data from internal registers to the output pins after conversion
Pin 3: It is an input pin; low to high pulse is given to start the conversion
Pin 4: It is a clock input pin, to give external clock
Pin 5: It is an output pin, goes low when conversion is complete
Pin 6: Analog non-inverting input
Pin 7: Analog inverting input, it's normally ground
Pin 8: Ground (0V)
Pin 9: It is an input pin, sets the reference voltage for analog input
Pin 10: Ground (0V)
Pin 11 – Pin 18: It is an 8-bit digital output pins
Pin 19: Is used with clock IN pin when internal clock source is used
Pin 20: Supply voltage; 5V

Features of ADC0804:

- 0V to 5V analog input voltage range with single 5V supply
- Compatible with microcontrollers, access time is 135 ns
- Easy interface to all microprocessors
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- No zero adjust required
- 0.3[Prime] standard width 20-pin DIP package
- Operates ratio metrically or with 5 VDC, 2.5 VDC, or analog span adjusted voltage reference.
- Differential analog voltage inputs.

Refer to the datasheet of ADC0804 for further details.

Components Required

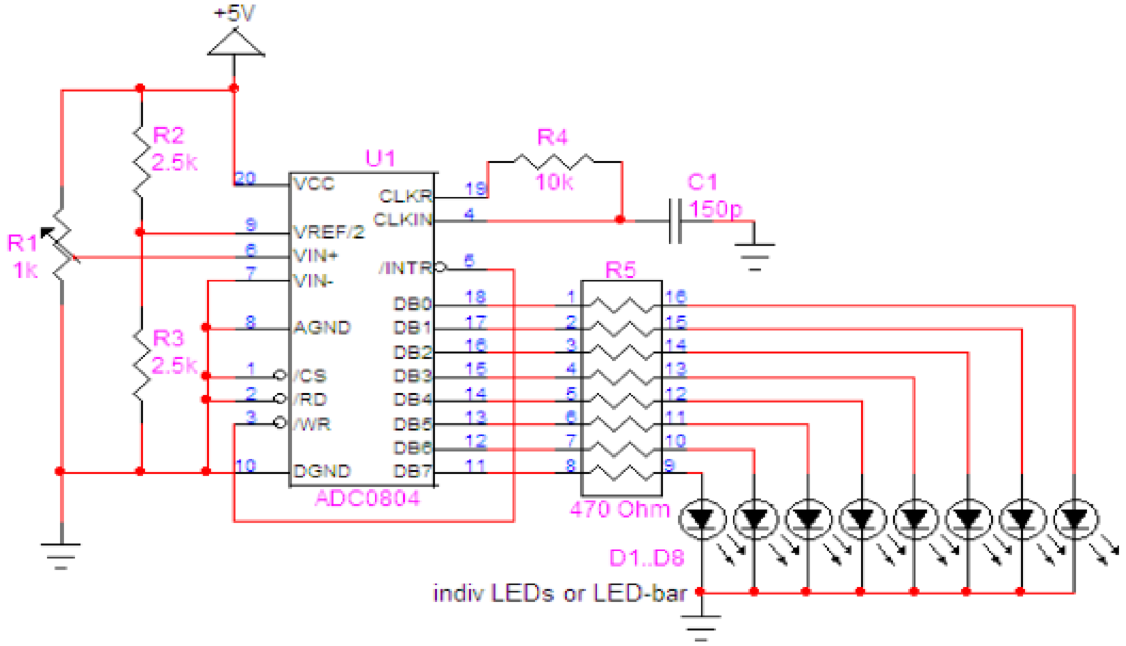
- ADC0804 IC
- LEDs 8
- R-Pack 9x470Ω
- Potentiometer, 1k (or 5k)
- Resistor, 2.5kΩ
- Resistor, 10kΩ
- Capacitor, 150 pF

Procedure

1. Make sure the main components, that is the ADC0804 itself, the LEDs and associated current limiting resistors, are oriented and placed properly on the proto board.
2. Connect power supply, +Vcc (5V) to ADC0804, pin 20, and pin 10 to Gnd.
3. Connect digital outputs D0 ... D7, pins 18 ... 11 (MSB), to the respective LEDs and their other end (the cathodes) via 470 Ω current limiting resistors to Gnd.
4. Next place the 1k potentiometer to the left of the ADC and connect its tap (wiper) to +V-in (pin 6)
5. Connect the voltage divider R2, R3 between +Vcc and Gnd, and its tap (junction between R2 and R3) to V-ref/2 (pin 7)

6. Connect components for internal clock operation (10kΩ, 150 pF), resistor from Clk-R (pin 19) to Clk-in (pin 4), and capacitor from that same pin (pin 4) to Gnd.

Circuit Diagram:



Observation

Your circuit should now be ready to be powered up, and (after having done so) working, i. e. the LEDs should display some binary number, which, if you turn the control on the pot, should change.

Set-up and test in both, single (manual) conversion and continuous mode, where conversion request is retriggered automatically.

Fill in a few entries as shown:

Vin (volts)	Vout (calculated)	Vout (observed)	Error	Comment
0.0000	00000000	00000000	0%	
0.0196	00000001	00000001	0%	
5	11111111	11111111	0%	

Exercise:

1. Write down your observation regarding the mismatch in the calculated value and the observed one (if any). Is that because of noise?

This page is intentionally left blank.

Lab Session 09

Implement an active Low-Pass filter using discrete components

This lab session is mapped on CLO-3 ‘Practice simulation of signals and systems using modern tools’. The taxonomy level is P-3. The CLO has been mapped to PLO-5, ‘modern tool usage’.

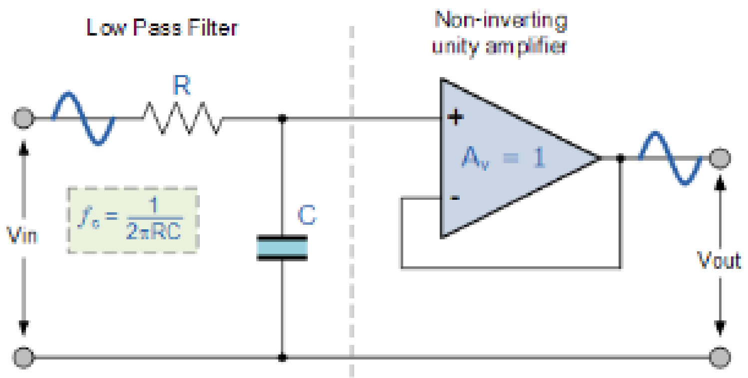
Limitation of Passive Low Pass Filters

In Lab Session 6, we saw how a basic first-order filter circuits, such as the low pass and the high pass filters can be made using just a single resistor in series with a non-polarized capacitor connected across a sinusoidal input signal.

We also noticed that the main disadvantage of passive filters is that the amplitude of the output signal is less than that of the input signal, i.e., the gain is never greater than unity and that the load impedance affects the filters characteristics.

Active Low Pass Filter

By combining a basic RC Low Pass Filter circuit with an operational amplifier we can create an Active Low Pass Filter circuit complete with amplification.



Working of Active Filters

As their name implies, **Active Filters** contain active components such as operational amplifiers, transistors or FET’s within their circuit design. They draw their power from an external power source and use it to boost or amplify the output signal.

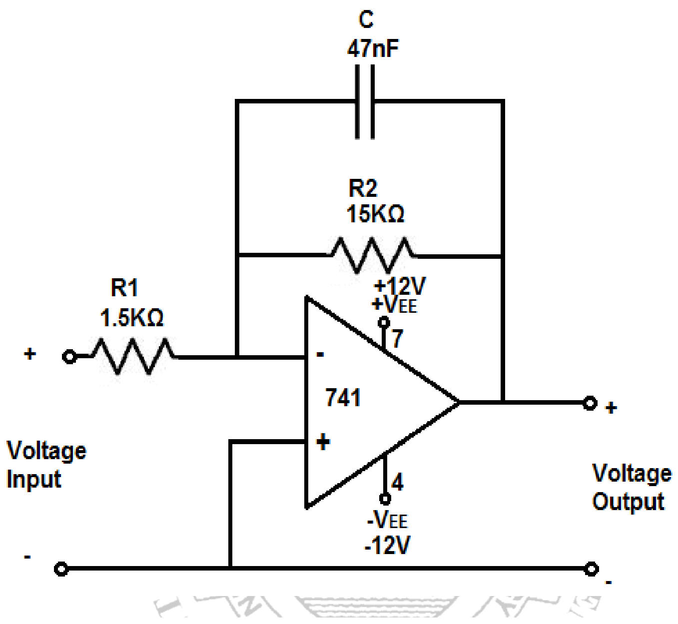
Filter amplification can also be used to either shape or alter the frequency response of the filter circuit by producing a more selective output response, making the output bandwidth of the filter narrower or even wider. Then the main difference between a “passive filter” and an “active filter” is amplification.

This first-order low pass active filter, consists simply of a passive RC filter stage providing a low frequency path to the input of a non-inverting operational amplifier. The amplifier is configured as a voltage-follower (Buffer) giving it a DC gain of one, $A_v = +1$ or unity gain as opposed to the previous passive RC filter which has a DC gain of less than unity.

The advantage of this configuration is that the op-amps high input impedance prevents excessive loading on the filters output while its low output impedance prevents the filters cut-off frequency point from being affected by changes in the impedance of the load.

While this configuration provides good stability to the filter, its main disadvantage is that it has no voltage gain above one. However, although the voltage gain is unity the power gain is very high as its output impedance is much lower than its input impedance. If a voltage gain greater than one is required we can use the following filter circuit.

Designing an Inverting Active Low Pass Filter with Amplification:



The part of the circuit composed of resistor R and capacitor C form the low pass filter. The formula for calculating the cutoff frequency is, frequency= 1/2πR2C= 1/2π(15KΩ)(47nF) = 225.8Hz ≈ 226Hz. So we use a 15KΩ resistor with a 47nF capacitor to form this 226Hz cutoff frequency point. The gain of the op amp is determined by resistors R1 and R2 by the formula, gain (AV) = -R2/R1. Since R2 is 10 times greater than R1, the gain is -10. The negative means that the voltage output is inverted from the voltage input. So while the input voltage is +10V, the output voltage is -10V. They're 180 degrees out of phase. When one is at the positive peak, the other is at its negative peak. A full AC signal is 360 degrees.

Usually phase does not matter. Unless you're dealing with a phase-critical application, it shouldn't matter whether the low pass filter is inverting or non-inverting.

Components Required

- Resistor R1= 1.5 K(or as required).
- Resistor R2= 15 K(or as required).
- Non-polarized Capacitor=C= 47nF (or as required).
- LM741 op-amp IC.
- Power Supply (+/- 12V)
- Function Generator.
- Breadboard.
- Oscilloscope.

Procedure

1. Connect all the circuit components as shown in the circuit diagram above.
2. The LM741 chip is composed of a single op amp. It is an 8-pin chip. We will only use 5 of the pins in our circuit. Pins 7 and 4 are the power pins. Pin 7 is where we connect positive DC voltage to and pin 4 we connect to negative voltage. Pins 2 and 3 are the input pins to the op amp. Pin 2 is the inverting terminal and pin 3 is the non-inverting terminal. Pin 6 is the output of the op amp.
3. Using Function Generator, apply 1Volt Sinusoid input signal with frequency lower than the cut-off frequency f_c.
4. Make necessary connection at output pin 6 of op-amp and ground to the oscilloscope.
5. Gradually start increasing the frequency of the input signal V_{in} beyond the cut-off frequency and observe your output on the oscilloscope screen.

Observation:

Starting with frequency less than cut-off frequency, gradually increase the frequency of the input signal and give your remarks for what you see on the oscilloscope.

Exercise:

1. Design an active low pass filter of 150 HZ and give your remarks.

This page is intentionally left blank.

Lab Session 10

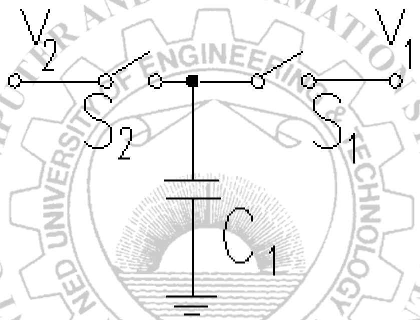
Implement a band pass filter using switched capacitor filter

What is Switched capacitor?

In the last decade or so many active filters with resistors and capacitors have been replaced with a special kind of filter called a switched capacitor filter. The switched capacitor filter allows for very sophisticated, accurate, and tunable analog circuits to be manufactured without using resistors. This is useful for several reasons. Chief among these is that resistors are hard to build on integrated circuits (they take up a lot of room), and the circuits can be made to depend on ratios of capacitor values (which can be set accurately), and not absolute values (which vary between manufacturing runs).

The Switched Capacitor Resistor

To understand how switched capacitor circuits work, consider the circuit shown with a capacitor connected to two switches and two different voltages.



If S_2 closes with S_1 open, then S_1 closes with switch S_2 open, a charge (q is transferred from v_2 to v_1 with

$$\Delta q = C_1(v_2 - v_1).$$

If this switching process is repeated N times in a time (t, the amount of charge transferred per unit time is given by

$$\frac{\Delta q}{\Delta t} = C_1(v_2 - v_1)\frac{N}{\Delta t}.$$

Recognizing that the left hand side represents charge per unit time, or current, and the number of cycles per unit time is the switching frequency (or clock frequency, f_{CLK}) we can rewrite the equation as

$$i = C_1(v_2 - v_1)f_{CLK}.$$

Rearranging we get

$$\frac{(v_2 - v_1)}{i} = \frac{1}{C_1 f_{CLK}} = R,$$

which states that the switched capacitor is equivalent to a resistor. The value of this resistor decreases with increasing switching frequency or increasing capacitance, as either will increase the amount of charge transferred from v_2 to v_1 in a given time.

The Switched Capacitor Integrator

Now consider the integrator circuit. The input-output relationship for this circuit is given by (neglecting initial conditions):

$$v_o(t) = -\frac{1}{RC_2}\int v_i(t)dt = -\omega'\int v_i(t)dt$$

We can also write this with the "s" notation (assuming a sinusoidal input, Ae^{st}, s=jω)

$$V_o(s) = -\frac{\omega'}{s}$$

If you replaced the input resistor with a switched capacitor resistor, you would get

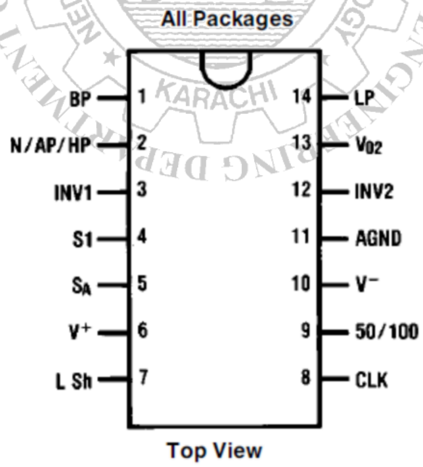
$$\omega' = \frac{1}{RC_2} = f_{CLK}\frac{C_1}{C_2}$$

Thus, you can change the equivalent ω' of the circuit by changing the clock frequency. The value of ω' can be set very precisely because it depends only on the ratio of C_1 and C_2, and not their absolute value.

LAB PERFORMANCE

MF5 Universal Monolithic switched Capacitor Filter

The MF5 is a general-purpose second-order state variable filter whose center frequency is proportional to the frequency of the square wave applied to the clock input (f_{CLK}). By connecting pin 9 to the appropriate DC voltage, the filter center frequency f_O can be made equal to either $f_{CLK}/100$ or $f_{CLK}/50$. f_O can be very accurately set (within ±0.6%) by using a crystal clock oscillator, or can be easily varied over a wide frequency range by adjusting the clock frequency.



Designing a 2nd order Butterworth low-pass filter:

In this lab session, we will design a 2nd order Butterworth low-pass filter with a cutoff frequency of 200 Hz, and a pass band gain of -2. The circuit will operate from a ±5V power supply, and the clock amplitude will be ±5v (CMOS) levels). From the specifications, the filter parameters are: f_O=200 Hz, H_{OLP}=-2, and, for Butterworth response, Q = 0.707.

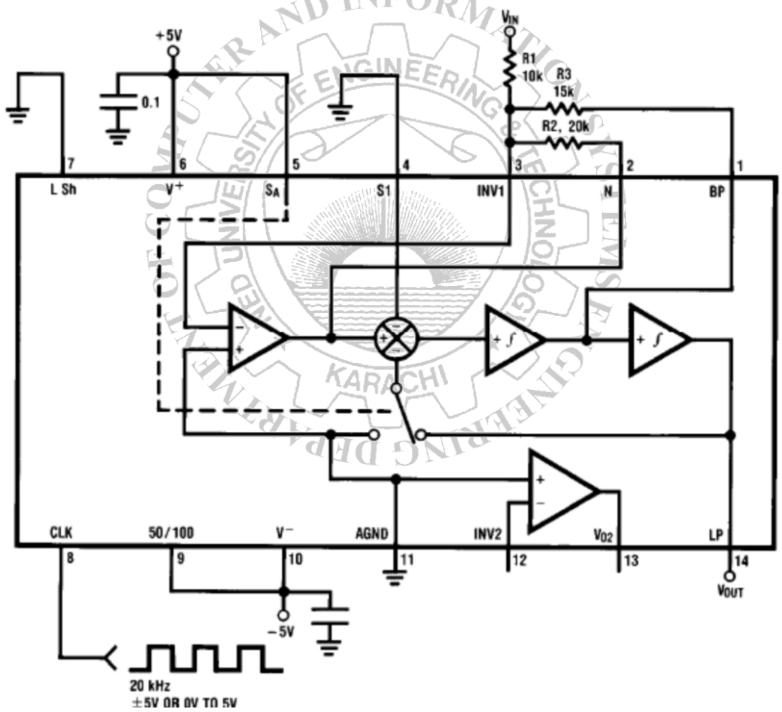
There are several modes of operation for the MF5, each having different characteristics. Some allow adjustment of f_{CLK}/fo, others produce different combinations of filter types, some are inverting while others are non-inverting, etc. To keep the example simple, we will use mode 1, which has notch, band pass, and low pass outputs, and inverts the signal polarity. Three external resistors determine the filter's Q and gain. From the equations accompanying Figure, Q = R3/R2 and the pass band gain H_{OLP} = -R2/R1. Since the input signal is driving a summing junction through R1, the input impedance will be equal to R1. Start by choosing a value for R1. 10k is convenient and gives reasonable input impedance.

For H_{OLP} = -2, we have:

R2 = -R1.H_{OLP} = 10k x 2 = 20k.

For Q = 0.707, we have:
R3 = R2.Q = 20k x 0.707 = 14.14k.Use 15k.

For operation on ±5V supplies, V+ is connected to a5V, V- to -5V, and AGND to ground. The power supplies should be ``clean" (regulated supplies are preferred) and 0.1 µF bypass capacitors are recommended.



EXERCISES

1. Show circuit diagram/ pin assignment to implement a 2nd order Butterworth **Band pass** filter using MF5 IC.

2. Write down your observations for a **2nd order Butterworth low-pass filter** when a frequency less than 200 Hz and more than 200 Hz frequency is applied at the input. What you observe at the output node in both of these cases.

Lab Session 11

Signal Processing in Simulink

This lab session shows how to perform basic signal processing on Simulink Toolbox functionality available in MATLAB. In this lab session, you will simulate the basic signal processing mechanisms such as signal amplification, signal addition in Simulink environment.

Simulink Tool

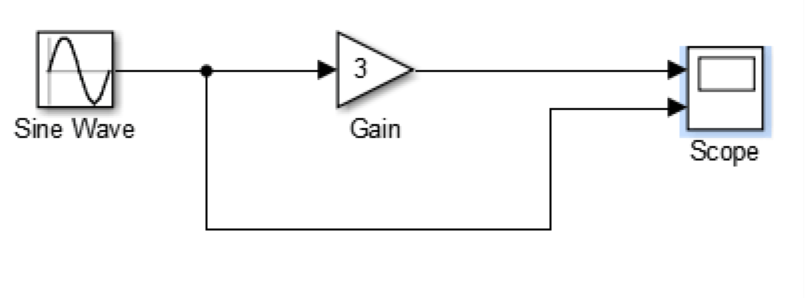
Simulink is a visual programming interface design to make modeling systems intuitive. It offers a way to solve equations numerically using a graphical user interface, rather than requiring code.

Model contains blocks, signals and annotation on a background:

- Blocks are mathematical functions; they can have varying numbers of inputs and outputs.
- Signals are lines connecting blocks, transferring values between them. Signals are different data types, for example numbers, vectors or matrices. Signals can be labeled.
- Annotations of text or images can be added to the model, and while not used in the calculations they can make it easier for others to understand design decisions in the model.

Signal Amplification

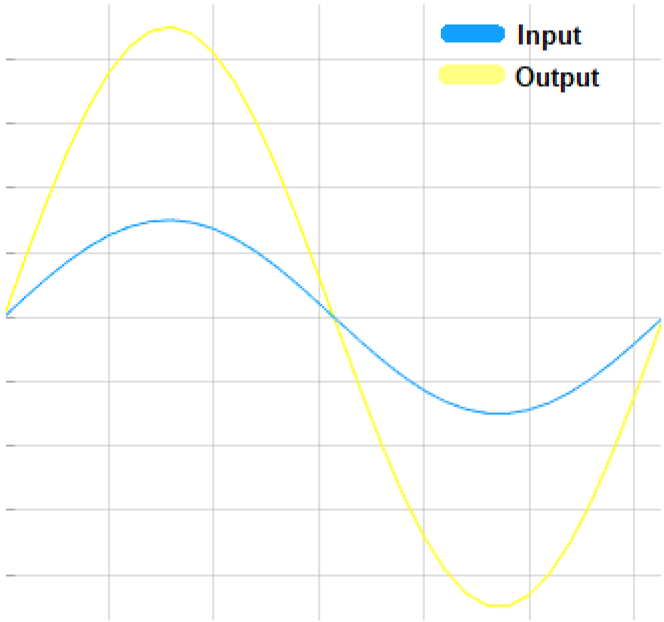
In the above example we are generating a sine wave from sine wave block and amplifying it by a gain factor which you directly check on Simulink by double clicking the scope block.



In order to implement this model, you have to drag and drop three elements from Simulink library, namely;

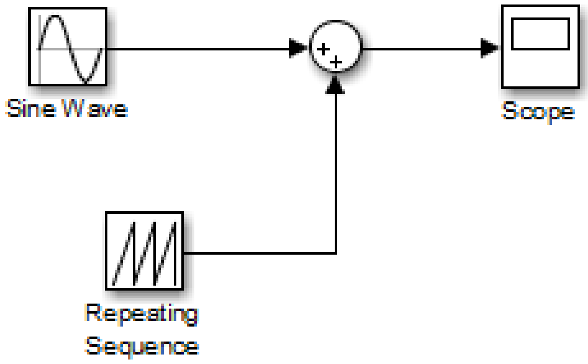
1. Sine wave generator block (*Input device*).
2. Gain block (*System*).
3. Scope block (*Output device*).

The output of both sine wave generator and gain block can be plotted simultaneously to understand the impact of the amplification by the gain block on the scope as shown in the following figure.



Signal Addition

The Simulink model for the summation/addition of two different signals is shown below:



EXERCISE:

1. Model a system that performs addition of a sinusoidal wave and a square pulse. Write down observation regarding output waveform.

This page is intentionally left blank.

Lab Session 12

Frequency processing models in Simulink

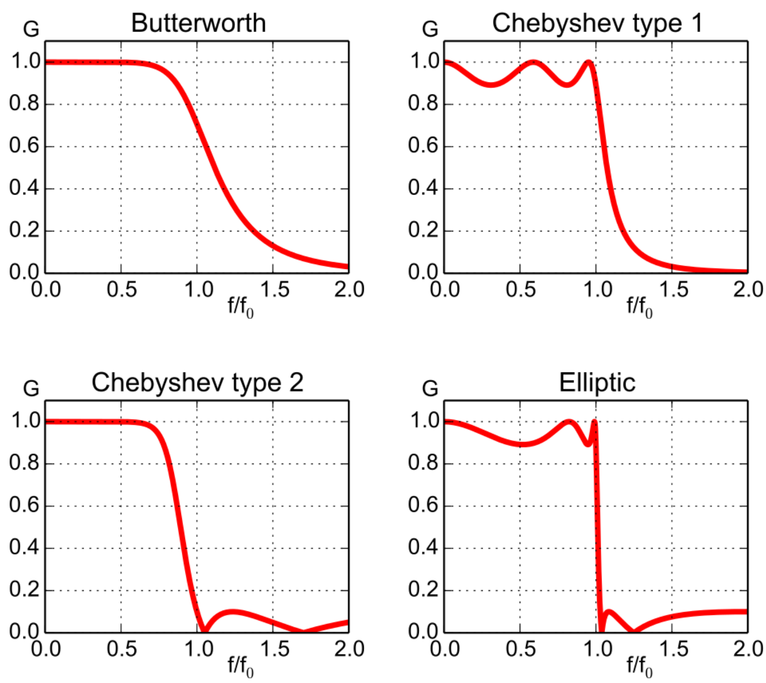
Filters in Simulink

The functioning of filters is covered in detail in the previous lab sessions. In this lab session you are supposed to implement a filter simulation. The different types of analog filters provided in Simulink are listed below:

- Butterworth
- Chebyshev type I.
- Chebyshev type II.
- Elliptic.
- Bessel.

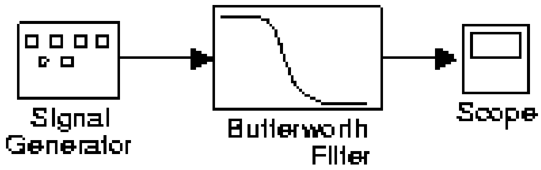
Butterworth Filter

The Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the pass band. It is also referred to as a maximally flat magnitude filter.



Example

A simple example to illustrate the easy migration from simulation to real-time implementation is shown in Figure below. There is a Simulink block diagram for simulation of a low-pass analog Butterworth filter.



The Simulink model shown in above figure has three blocks, namely;

1. Signal Generator (*Input Device*).
2. Butterworth Filter (*System*).
3. Scope (*Output Device*).

The important parameters for the filter block are listed as under:

- **Design method:** The filter design method: Butterworth, Chebyshev type I, Chebyshev type II, Elliptic, or Bessel.
- **Filter type:** The type of filter to design: Low pass, High pass, Band pass, or Band stop.
- **Filter order:** The order of the filter, for low pass and high pass configurations. For band pass and band stop configurations, the order of the final filter is *twice* this value.
- **Pass band edge frequency:** The pass band edge frequency, in rad/s, for the high pass and low pass configurations of the Butterworth, Chebyshev type I, elliptic, and Bessel designs.
- **Lower pass band edge frequency:** The lower pass band frequency, in rad/s, for the band pass and band stop configurations of the Butterworth, Chebyshev type I, elliptic, and Bessel designs.
- **Upper pass band edge frequency:** The upper pass band frequency, in rad/s, for the band pass and band stop configurations of the Butterworth, Chebyshev type I, elliptic, and Bessel designs.
- **Stopband edge frequency:** The stopband edge frequency, in rad/s, for the high pass and low pass band configurations of the Chebyshev type II design.
- **Lower stopband edge frequency:** The lower stopband edge frequency, in rad/s, for the band pass and band stop configurations of the Chebyshev type II design.
- **Upper stopband edge frequency:** The upper stopband edge frequency, in rad/s, for the band pass and band stop filter configurations of the Chebyshev type II design.
- **Pass band ripple in dB:** The pass band ripple, in dB, for the Chebyshev Type I and elliptic designs.
- **Stopband attenuation in dB:** The stopband attenuation, in dB, for the Chebyshev Type II and elliptic designs.

The design and band configuration of the filter are selected from the **Design method** and **Filter type** pop-up menus in the dialog box. For each combination of design method and band configuration, an appropriate set of secondary parameters is displayed.

EXERCISES

1. Model a high pass filter with cutoff frequency of 100KHz in Simulink and write your observation regarding change in output waveform before and after cutoff frequency point.

This page is intentionally left blank.

Lab Session 13

Simulating transfer function of LTI system in Simulink

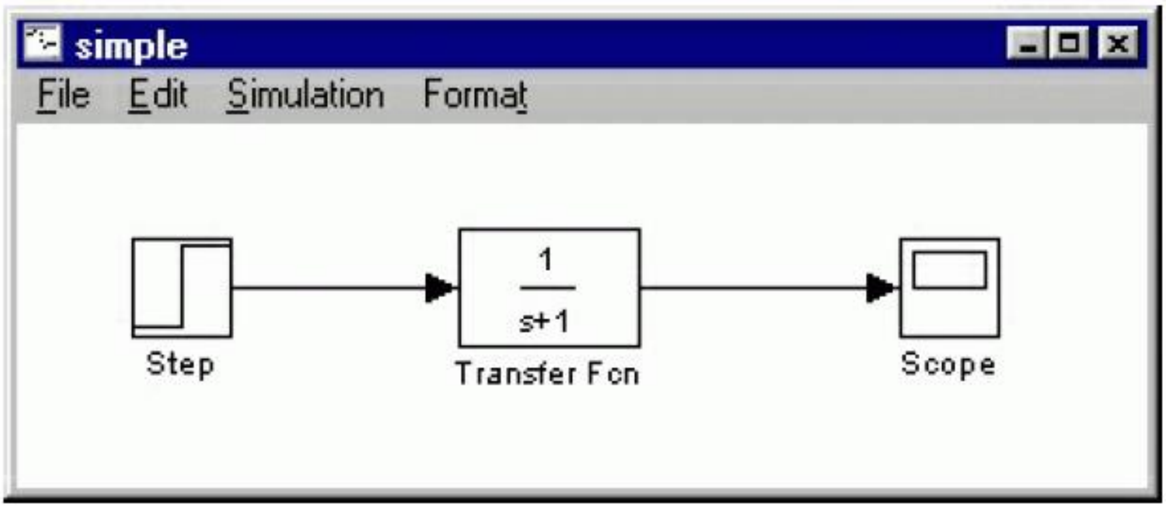
Linear time-invariant systems (LTI systems) are a class of systems used in signals and systems that are both linear and time-invariant. Linear systems are systems whose outputs for a linear combination of inputs are the same as a linear combination of individual responses to those inputs. Time-invariant systems are systems where the output does not depend on *when* an input was applied. These properties make LTI systems easy to represent and understand graphically.

Conversely, the LTI system can also be described by its transfer function. The transfer function is the Laplace transform of the impulse response. This transformation changes the function from the time domain to the frequency domain. This transformation is important because it turns differential equations into algebraic equations, and turns convolution into multiplication. In the frequency domain, the output is the product of the transfer function with the transformed input.

$$Y(S) = H(S)X(S)$$

Simulink Model for Transfer function simulation

The Simulink model for simulating the transfer function is shown below:



The model shown in the figure above is comprised of three blocks, namely:

1. Step signal block (*input signal*).
2. Transfer function (*system*).
3. Scope (*Output block*).

Example

For Linear, Time-Invariant systems (LTI systems), the input and output have a simple relationship in the frequency domain.

$$Out(s) = G(s) * In(s)$$

Where the *transfer function G(s)* can be expressed by the algebraic function

$$G(s) = \frac{num(s)}{den(s)} = \frac{n(0) * s^0 + n(1) * s^1 + n(2) * s^2 + \ldots}{d(0) * s^0 + d(1) * s^1 + d(2) * s^2 + \ldots}$$

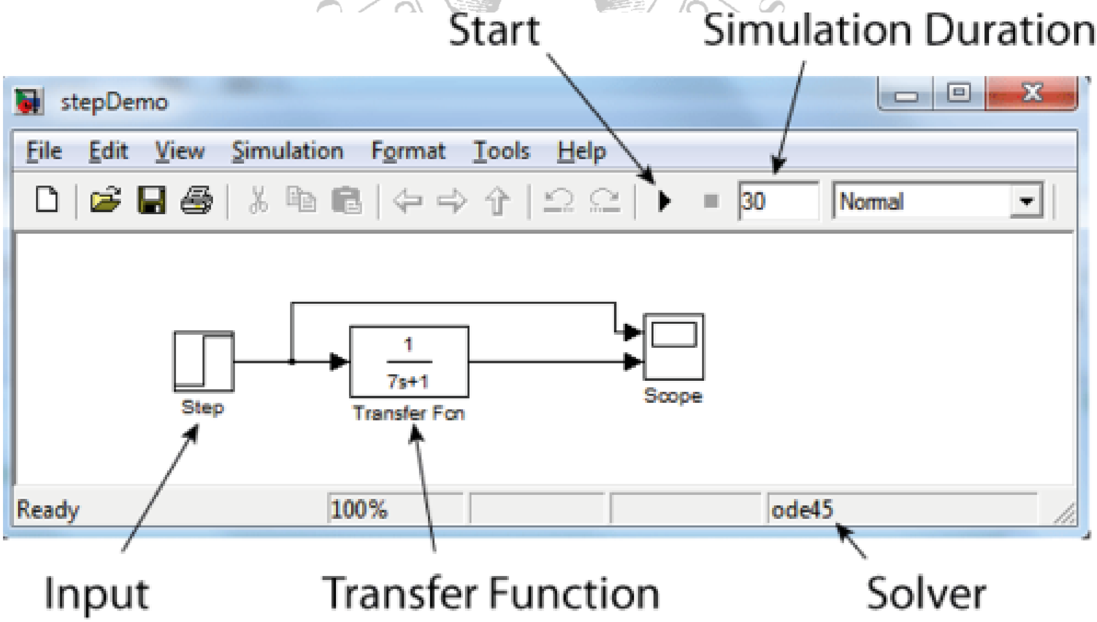
In other words, specifying the coefficients of the numerator (n) and denominator (d) uniquely characterizes the transfer function. This notation is used by some computational tools to simulate the response of such a system to a given input.

Different tools can be used to simulate such a system. For example, the response of a low-pass filter with a time-constant of 7 sec to an input step at 1 sec has the following transfer function

$$G(s) = \frac{1}{7s + 1}$$

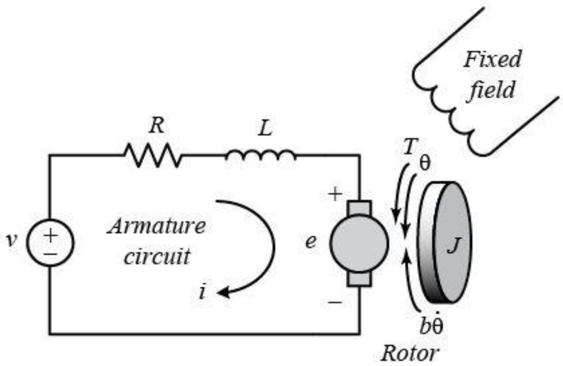
and can be simulated as follows:

.



EXERCISE:

1. The DC motor model is shown in the Fig below:



The transfer function of the DC motor shown above is given as:

$$\frac{\mathbf{0.01}}{\mathbf{0.005\,s^2 + 0.06s + 0.1001}}$$

Simulate the step response of DC motor in Simulink environment.

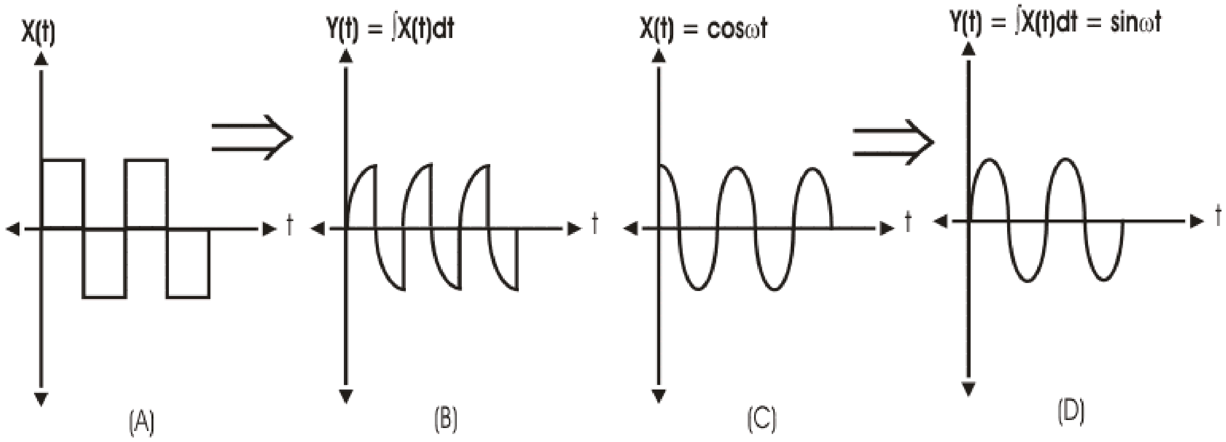
.

This page is intentionally left blank.

Lab Session 14

Integration and Differentiation Operations on a Signal

Differentiation of a Signal



For differentiation of signals, it must be noted that this operation is only applicable for only continuous signals, as a discrete function cannot be differentiated. The modified signal we get on differentiation has tangential values of the parent signal at all instance of time. Mathematically it can be expressed as:

$$Y(t) = \frac{d}{dt}X(t)$$

Differentiation of a standard square and sine wave is shown in the figure above.

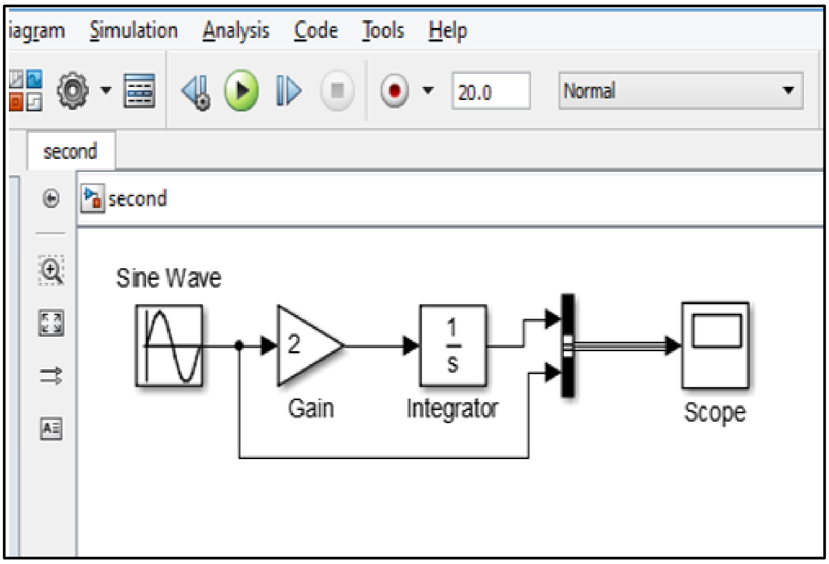
Integration of **a Signal**

Like differentiation, integration of signals is also applicable to only continuous time signals. The limits of integration will be from – ∞ to present instance of time t. It is mathematically expressed as,

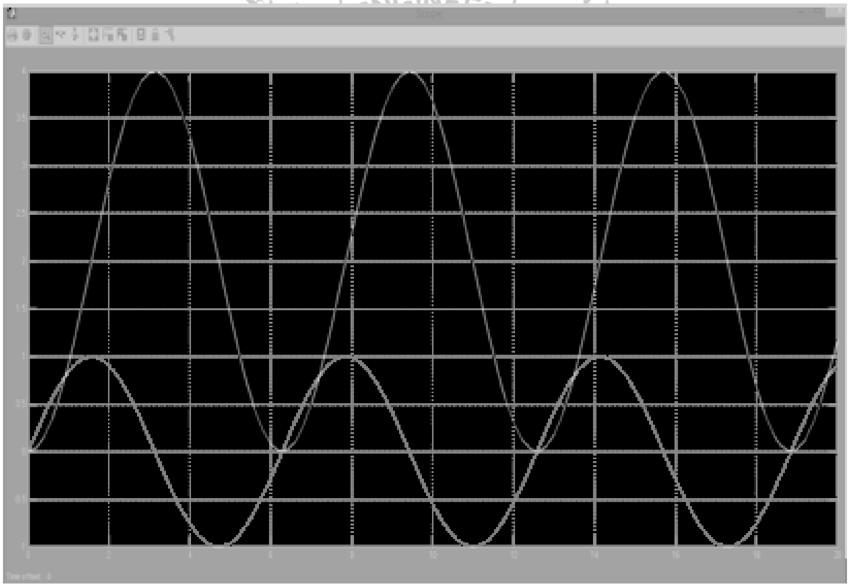
$$Y(t) = \int_{-\infty}^{t} X(t)dt$$

Simulink Model

The Simulink model is shown in the figure below:



The output of the integrator block and the actual wave is shown in figure below:

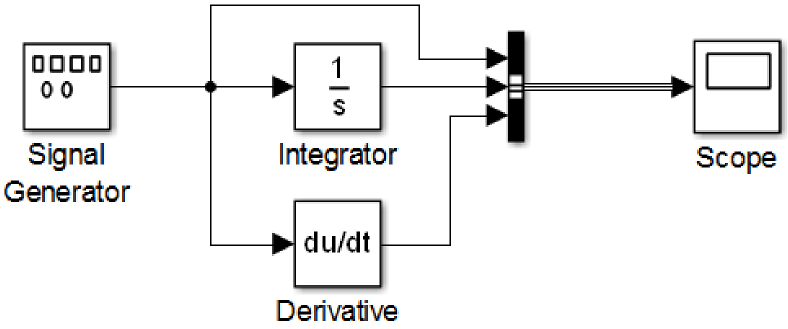


For the above example, we require the following blocks.

- Sine wave
- Integrator
- Scope
- Gain
- Bus creator

EXERCISE:

1. Design the model in the figure below and write down your observations regarding the output waveforms.



DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

RUBRIC

Course Code: _______________ Week #: _______________ Lab #: _________

Assigned task: ____

____

CRITERIA AND SCALES

Criterion 1: Confidence with tool interface

0	1	2	3
The student is unfamiliar with the tool	The student is familiar with the visible features of the tool	The student is familiar with the unexposed features of the tool	The student is proficient with the tool

Criterion 2: Appraisal of lab assignment

0	1	2	3	4
The student did not understand the lab task	The student understood the lab task but had problem with tool usage	The student initiated implementation of the lab task but could not complete it	The student successfully completed the lab task with the tool assigned	The student used an innovative approach for successful completion of the lab task

Criterion 3: Answer to questions related to the tool

0	1	2
The student has no idea about tool application	The student has vague idea about tool application	The student has a clear idea about tool application

Criterion 4: Answer to questions related to the lab task

0	1	2	3
The student did not answer any question regarding lab task	The student answered a few questions regarding lab task	The student answered most of the questions regarding lab task	The student answered all the questions regarding lab task

Criterion 5: How would you grade the interaction of the student with lab resources (lab personnel, participant students, equipment)?

0	1	2
The student took no notice of the lab resources	The student was aware of lab resources for a short period of time but was mostly unconcerned	The student effectively interacted with the lab resources

Total Marks: ________

Teacher's Signature: ___________________

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

RUBRIC

Course Code: _______________ Week #: _______________ Lab #: ________

Assigned task: ____

____

____

CRITERIA AND SCALES

Criterion 1: Confidence with tool interface

0	1	2	3
The student is unfamiliar with the tool	The student is familiar with the visible features of the tool	The student is familiar with the unexposed features of the tool	The student is proficient with the tool

Criterion 2: Appraisal of lab assignment

0	1	2	3	4
The student did not understand the lab task	The student understood the lab task but had problem with tool usage	The student initiated implementation of the lab task but could not complete it	The student successfully completed the lab task with the tool assigned	The student used an innovative approach for successful completion of the lab task

Criterion 3: Answer to questions related to the tool

0	1	2
The student has no idea about tool application	The student has vague idea about tool application	The student has a clear idea about tool application

Criterion 4: Answer to questions related to the lab task

0	1	2	3
The student did not answer any question regarding lab task	The student answered a few questions regarding lab task	The student answered most of the questions regarding lab task	The student answered all the questions regarding lab task

Criterion 5: How would you grade the interaction of the student with lab resources (lab personnel, participant students, equipment)?

0	1	2
The student took no notice of the lab resources	The student was aware of lab resources for a short period of time but was mostly unconcerned	The student effectively interacted with the lab resources

Total Marks: ________

Teacher's Signature: ___________________

DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING

RUBRIC

Course Code: _______________ Week #: _______________ Lab #: _________

Assigned task: ____

____

____

CRITERIA AND SCALES

Criterion 1: Confidence with tool interface

0	1	2	3
The student is unfamiliar with the tool	The student is familiar with the visible features of the tool	The student is familiar with the unexposed features of the tool	The student is proficient with the tool

Criterion 2: Appraisal of lab assignment

0	1	2	3	4
The student did not understand the lab task	The student understood the lab task but had problem with tool usage	The student initiated implementation of the lab task but could not complete it	The student successfully completed the lab task with the tool assigned	The student used an innovative approach for successful completion of the lab task

Criterion 3: Answer to questions related to the tool

0	1	2
The student has no idea about tool application	The student has vague idea about tool application	The student has a clear idea about tool application

Criterion 4: Answer to questions related to the lab task

0	1	2	3
The student did not answer any question regarding lab task	The student answered a few questions regarding lab task	The student answered most of the questions regarding lab task	The student answered all the questions regarding lab task

Criterion 5: How would you grade the interaction of the student with lab resources (lab personnel, participant students, equipment)?

0	1	2
The student took no notice of the lab resources	The student was aware of lab resources for a short period of time but was mostly unconcerned	The student effectively interacted with the lab resources

Total Marks: ________

Teacher's Signature: ___________________