The Islamia University of Bahawalpur

# **U**niversity **C**ollege of **E**ngineering **&T**echnology **D**epartment of **C**omputer **S**ystem **E**ngineering

|  |  |  |
| --- | --- | --- |
| **LAB MANUAL** | **SIGNALS AND SYSTEMS EE-311** | **5thSemester** |

**LAB EXPERIMENT # 11**

**LTIC system and Differential Equation in MATLAB**

|  |  |
| --- | --- |
| **Student Name:** | **Roll No:** |
| **Lab Instructor Signatures:** | **Date:** |

**OBJECTIVE:**

* Compute and plot the frequency response of an LTICsystems using the Laplace transform.
* Solving Ordinary Differential Equations

**Introduction:**

The Laplace transform is used for solving differential and integral equations. The Laplace transform is often interpreted as a transformation from the time-domain, in which inputs and outputs are functions of time, to the frequency-domain, where the same inputs and outputs are functions of complex angular frequency, in radians per unit time.Laplace transform provides an alternative functional description that often simplifies the process of analyzing the behavior of the system, or in synthesizing a new system based on a set of specifications.

**Code:**

clear all; close all;

% a numerator coefficients of transfer function

% b denominator coefficients of transfer function

% z is zero

% p is pole

% k is gain

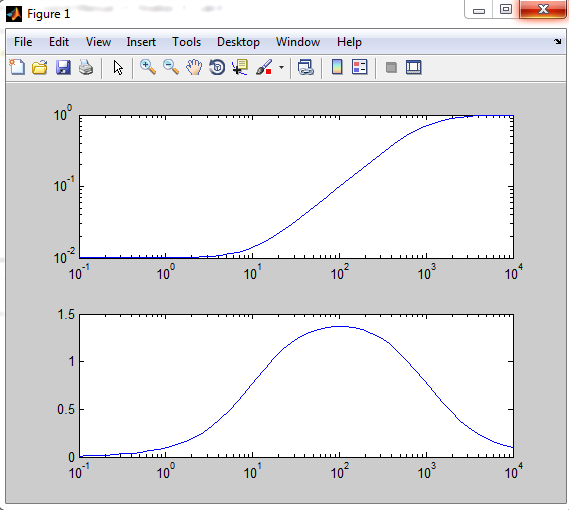
% First Order High pass b=[1 10];

a=[1 1000];

w = logspace(-1,4); h = freqs(b,a,w); mag = abs(h);

phase = angle(h); subplot(2,1,1), loglog(w,mag) subplot(2,1,2), semilogx(w,phase)

# output:



1. **First Order Low pass:**

**Code:**

b=[1 1000];

a=[1 10];

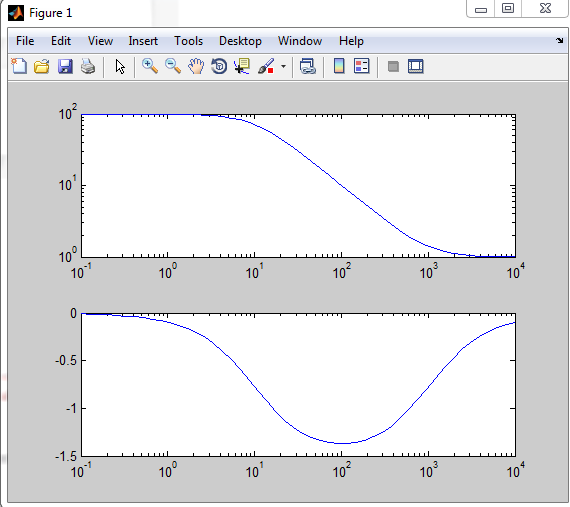
w = logspace(-1,4); h = freqs(b,a,w); mag = abs(h);

phase = angle(h); figure

subplot(2,1,1), loglog(w,mag)

subplot(2,1,2), semilogx(w,phase)

# Output:



1. **Second order low pass transfer function**

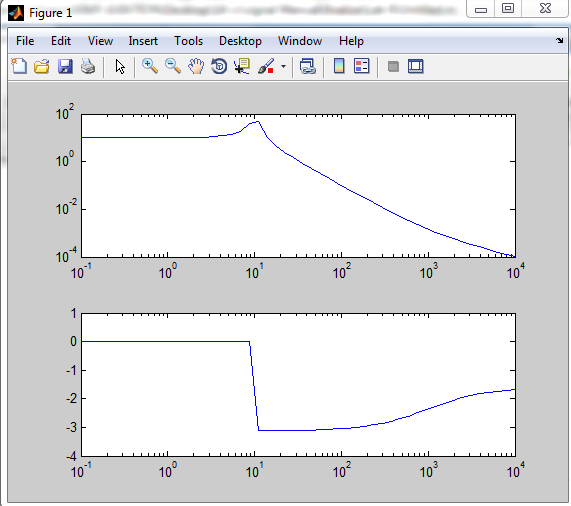
**Code:**

a=[1 1/20 100];

w = logspace(-1,4); h = freqs(b,a,w); mag = abs(h);

phase = angle(h); subplot(2,1,1), loglog(w,mag) subplot(2,1,2), semilogx(w,phase)

# Output:



1. **Second order High pass transfer function:**

**Code:**

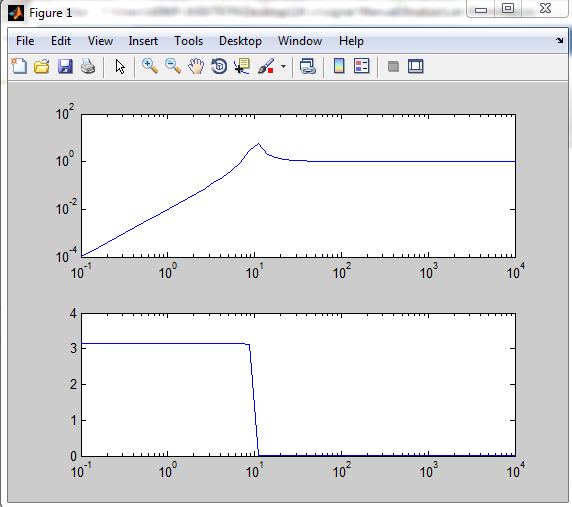
b=[1 0 0];

a=[1 1/20 100];

w = logspace(-1,4); h = freqs(b,a,w); mag = abs(h);

phase = angle(h); subplot(2,1,1), loglog(w,mag) subplot(2,1,2), semilogx(w,phase)

**Output:**



1. **Second order Band pass transfer function**

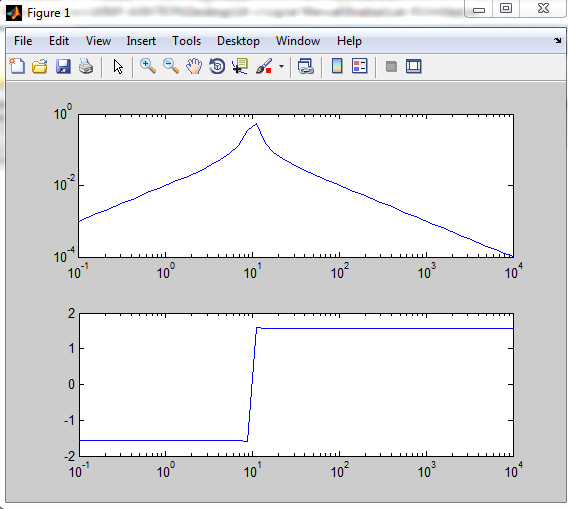
**Code:**

b=[-1 0];

a=[1 1/20 100];

w = logspace(-1,4); h = freqs(b,a,w); mag = abs(h);

phase = angle(h); subplot(2,1,1), loglog(w,mag) subplot(2,1,2), semilogx(w,phase)



1. **Finding Zero and poles from transfer function:**

**Code:**

b=[-1 0];

a=[1 1/20 100];

[z,p,k] = tf2zp(b,a); z

p

k

# Output:

1. **Finding Transfer function from zero and pole:**

**Code:**

k=1;

z=[-10 -1000]';

p=[-100 -10000]';

[b,a] = zp2tf(z,p,k); b

a

* + **Solving Ordinary Differential Equations**:

MATLAB has a command, [**dsolve**], that solves ordinary differential equations (ODEs) symbolically. One form of the dsolvecommand is

dsolve('ODE','initial conditions')

In the ODE string, **D** is used to represent the \_rst derivative and **Dn**represents the nth derivative. To solve the differential equation y**’**(t) = ay(t) subject to the initial condition y(0) = c and assign the solution to y, one gives thecommands

syms a c

y = dsolve('Dy = a\*y','y(0) = c')

**Output**:

y =c\*exp(a\*t)

It is possible to solve much more complicated differential equations this way.