Exp: 15 FIND STABILITY OF A SYSTEM USING ROUTH HURWITZ CRITERION

Program:

```
clear;
clc;
xdel(winsid());
mode(0);
s=%s;
H=s^4+2*s^3+3*s^2+4*s+5;
//H=s^5+7*s^4+6*s^3+42*s^2+8*s+56;
disp(H,'The given characteristics equation 1-G(s)H(s)=');
c=coeff(H);
len=length(c);
r=routh_t(H);
disp(r,"Rouths table=");
x=0;
for i=1:len
if(r(i,1)<0)
x=x+1;
end
end
if(x>=1)
printf("From Rouths table, it is clear that the system is unstable.")
else
printf("From Rouths table, it is clear that the system is stable.")
end
```

17.Generation of Common Discrete Time Signals

Program:

```
//UNIT IMPULSE SIGNAL
clear all;
close ;
N =5; //SET LIMIT
t1 = -5:5;
x1 =[ zeros (1 , N ) ,ones (1 ,1) ,zeros (1 , N ) ];
subplot (2 ,4 ,1) ;
plot2d3 ( t1 , x1 )
xlabel ( ' tim e ' ) ;
ylabel ( ' Ampli tude ' ) ;
title ( ' Uni t im p ul s e s i g n a l ' ) ;
//UNIT STEP SIGNAL
t2 = -5:5;
x2 =[ zeros (1 , N ) ,ones (1 , N +1) ];
subplot (2 ,4 ,2) ;
plot2d3 ( t2 , x2 )
xlabel ( ' tim e ' ) ;
ylabel ( ' Ampli tude ' ) ;
title ( ' Uni t s t e p s i g n a l ' ) ;
//EXPONENTIAL SIGNAL
t3 =0:1:20;
x3 =exp( - t3 ) ;
subplot (2 ,3 ,3) ;
plot2d3 ( t3 , x3 ) ;
xlabel ( ' tim e ' ) ;
ylabel ( ' Ampli tude ' ) ;
title ( ' E x p o n e n t i a l s i g n a l ' ) ;
//UNIT RAMP SIGNAL 4
t4 =0:20;
```

```
x4 = t4 ;
subplot (2 ,3 ,4) ;
plot2d3 ( t4 , x4 ) ;
xlabel ( ' tim e ' ) ;
ylabel ( ' Ampli tude ' ) ;
title ( ' Uni t ramp s i g n a l ' ) ;
//SINUSOIDAL SIGNAL
t5 =0:0.04:1;
x5 =sin (2* %pi * t5 ) ;
subplot (2 ,3 ,5) ;
plot2d3 ( t5 , x5 ) ;
title ( ' S i n u s o i d a l S i g n a l ' )
xlabel ( ' tim e ' ) ;
ylabel ( ' Ampli tude ' ) ;
//RANDOM SIGNAL
t6 = -10:1:20;
x6 = rand (1 ,31) ;
subplot (2 ,3 ,6) ;
plot2d3 ( t6 , x6 ) ;
xlabel ( ' tim e ' ) ;
ylabel ( ' Ampli tude ' ) ;
title ( ' Random s i g n a l ' ) ;
```

18.DIT-FFT and DIF-FFT Algorithm

Program

```
clear;
clc ;
close ;
x = [1,-1,-1,-1,1,1,1,-1];
//FFT Computation
```

```
X = fft (x , -1);

disp(X,'X(z) = ');
```

19.Analog Filter design Using Transformation method

Program

```
clear;

clc ;

close ;

s=%s;

z=%z;

HS=(s^2+4.525)/(s^2+0.692*s+0.504);

T=1;

HZ=horner(HS,(2/T)*(z-1)/(z+1));

disp(HZ,'H(z) =');
```

ii).program

```
clear;

clc ;

close ;

s=%s;

T=0.2;

HS=10/(s^2+7*s+10);

elts=pfss(HS);

disp(elts,'Factorized HS = ');

//The poles comes out to be at -5 and -2

p1=-5;

p2=-2;
```

```
z=%z;
HZ=T*((-3.33/(1-%e^(p1*T)*z^(-1)))+(3.33/(1-%e^(p2*T)*z^(-1))))
disp(HZ,'HZ = ');
```

20,Analog Butterworth Filter

```
//First Order Butterworth Low Pass Filter
clear;
clc;
close;
s = poly(0,'s');
Omegac = 0.2*%pi;
H = Omegac/(s+Omegac);
T =1;//Sampling period T = 1 Second
z = poly(0,'z');
Hz = horner(H,(2/T)*((z-1)/(z+1)))
HW  =frmag(Hz(2),Hz(3),512);
W = 0:%pi/511:%pi;
plot(W/%pi,HW)
a=gca();
a.thickness = 3;
a.foreground = 1;
a.font_style = 9;
xgrid(1)
xtitle('Magnitude Response of Single pole LPF Filter Cutoff frequency = 0.2*pi','Digital Frequency--
>','Magnitude');
Disp(“Hz”,Hz);
```