Table of Contents


```
clc
clear all;
close all;
```

program to calculate the diffrential equations solutions

```
%y[n]+2y[n-1]=3y[n-2]=x[n]
%to find natural response yn[n], forced response yf[n] and total response
%yt[n] with initial conditons [-n] and y[-2] and input (1/2)^n u(n)
```

input sequence

```
N=10
n=0:N-1
disp('input is')
x=(1/2).^n
a=[1 2 3] %coefficients of y
b=1       %coefficients of x
ic=[1,-1] %initial conditions
```

```
N =

    10


n =

     0     1     2     3     4     5     6     7     8     9

input is

x =

  Columns 1 through 7

    1.0000    0.5000    0.2500    0.1250    0.0625    0.0313    0.0156

  Columns 8 through 10
```

```
    0.0078    0.0039    0.0020

a =

     1     2     3

b =

     1

ic =

     1    -1
```

natural response

```
yn(1)= -2*ic(1)-3*ic(2);
yn(2)= -2*yn(1)-3*ic(1);

for i=3:N
    yn(i)= -2*yn(i-1)-3*yn(i-2);
end
disp('**********************************************************************
disp('natural response=');
disp(yn);
```

```
        **********************************************************************
        natural response=
             1    -5     7     1   -23    43   -17   -95   241  -197
```

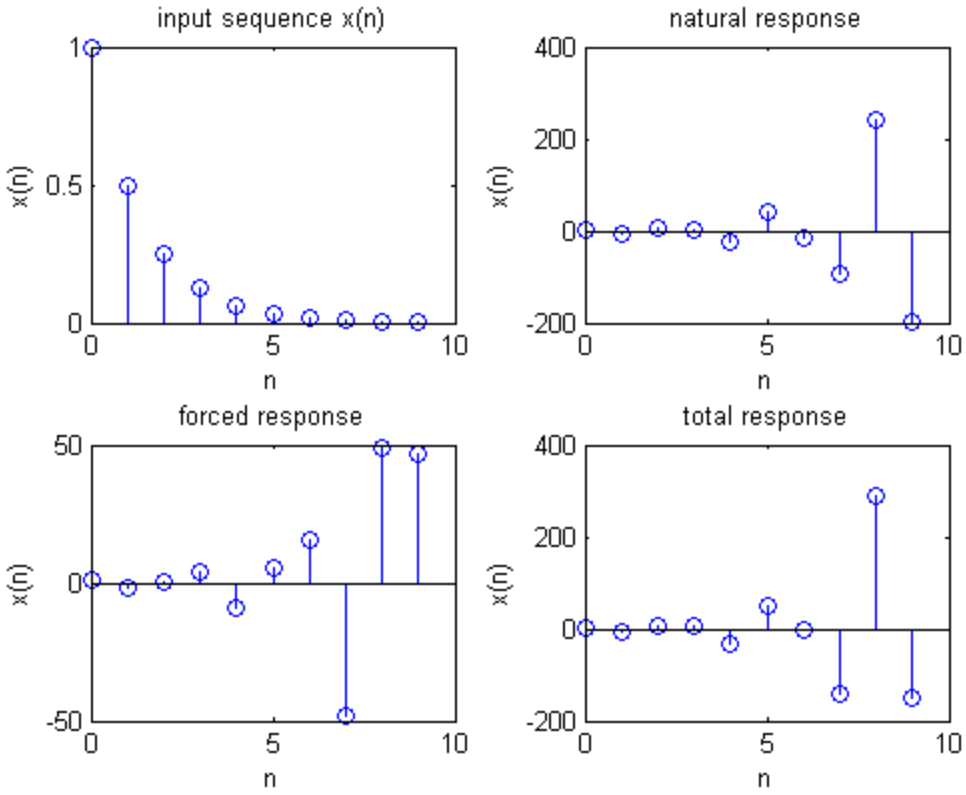
forced response

```
yf(1)=x(1);
yf(2)=x(2)-2*yf(1);

for k=3:N
    yf(k)=x(k)-2*yf(k-1)-3*yf(k-2);

end
disp('**********************************************************************
disp('forced response=');
disp(yf);

% % total response

for d=1:N
    yt(d)=yn(d)+yf(d);
```

```
end

disp('**************************************************************************
disp('total response=');
disp(yt);
disp('##########################################################################
p=filtic(b,a,ic);
ynatural=filter(b,a,zeros(1,N),p);
disp('**************************************************************************
disp('using inbuilt natural is=');
disp(ynatural);

yforced=filter(b,a,x);
disp('**************************************************************************
disp('using inbuilt forced is=');
disp(yforced);

for f=1:N
    ytotal(f)=ynatural(f)+yforced(f);
end
disp('**************************************************************************
disp('using inbuilt toatl is=');
disp(ytotal);

disp('##########################################################################

%%to plot
n=0:9;
subplot(2,2,1);
stem(n,x);
xlabel('n');
ylabel('x(n)');
title('input sequence x(n)');

subplot(2,2,2);
stem(n,yn);
xlabel('n');
ylabel('x(n)');
title('natural response');

subplot(2,2,3);
stem(n,yf);
xlabel('n');
ylabel('x(n)');
title('forced response');

subplot(2,2,4);
stem(n,yt);
xlabel('n');
ylabel('x(n)');
title('total response');
```

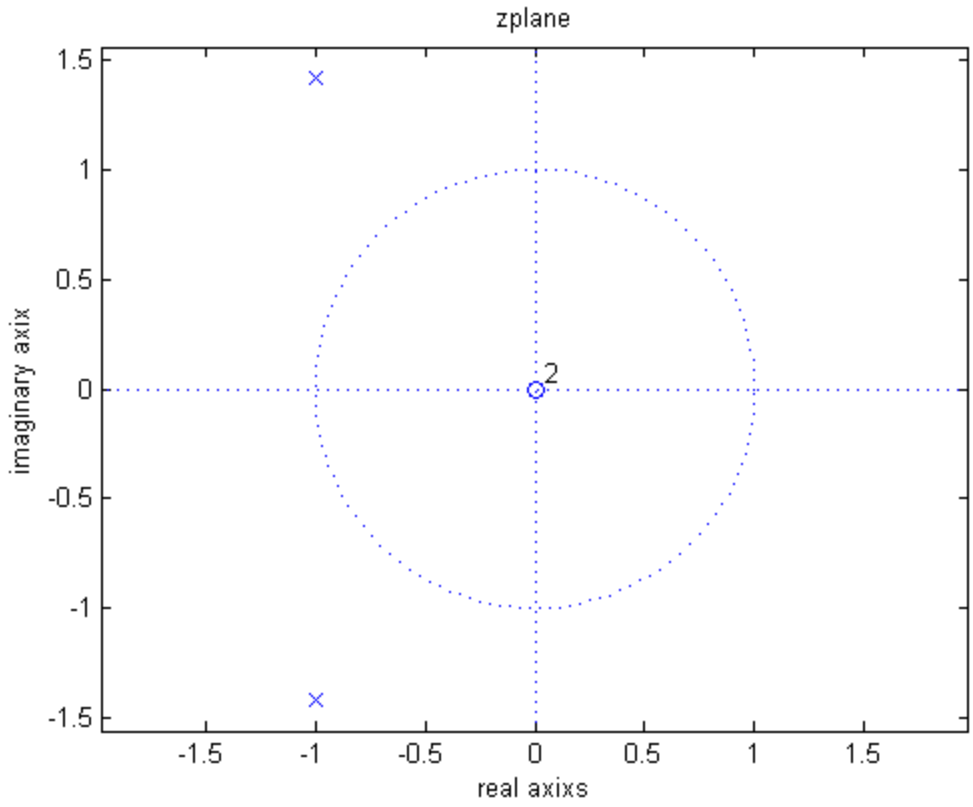
```
        **************************************************************************
        forced response=
```

```
  Columns 1 through 7

    1.0000   -1.5000    0.2500    4.1250   -8.9375    5.5313   15.7656

  Columns 8 through 10

  -48.1172   48.9414   46.4707

**************************************************************************
total response=
  Columns 1 through 7

    2.0000   -6.5000    7.2500    5.1250  -31.9375   48.5313   -1.2344

  Columns 8 through 10

 -143.1172  289.9414 -150.5293

##########################################################################
**************************************************************************
using inbuilt natural is=
     1    -5     7     1   -23    43   -17   -95   241  -197

**************************************************************************
using inbuilt forced is=
  Columns 1 through 7

    1.0000   -1.5000    0.2500    4.1250   -8.9375    5.5313   15.7656

  Columns 8 through 10

  -48.1172   48.9414   46.4707

**************************************************************************
using inbuilt toatl is=
  Columns 1 through 7

    2.0000   -6.5000    7.2500    5.1250  -31.9375   48.5313   -1.2344

  Columns 8 through 10

 -143.1172  289.9414 -150.5293

##########################################################################
```



zplot

```
figure(2);
zplane(b,a);
xlabel('real axixs');
ylabel('imaginary axix');
title('zplane');
```



stablity on basis of roots of a

```
q=roots(a);
if abs(q)<1
    disp('system is stable ');
else disp('system is unstable ');
end
```

```
        system is unstable
```

Published with MATLAB® R2014a