International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 5, Sept-Oct-2023, ISSN (Online): 2395-566X

A Comparative Study Of Gain And Phase Margin To Analyze Stability Of A System From Its Corrosponding Transfer Functions And The Plot Of Power Spectral Density Curve By Using Neumerical Results Obtained Fromcircular And Linear Convolutions And Fast Fourier Transformof Real And Complex Sets Of Neumerical Data Sequences

Abir Chakraborty Senior Research Fellow
Vellore Institute Of Technology
Vellore,Tamil Nadu,CH

Abstract- This work is an application where we have tried to combine the basic concept of control system engineering with digital signal processing engineering. Actually there are so many application available regarding digital signal processing, out of all the applications we have selected only one basic terminology that is "CONVOLUTION" and "FAST FOURIER TRANSFORM". That's why we have used two very common convolution and transformation techniques Linear and Circular convolutions. The results of these two convolutions that we have generated have been used in control system engineering purpose. For our convenience we have used both real and complex data's for both types of LINEAR and CIRCULAR convolutions as well as fast Fourier transformation then the outcomes that we have got after convolutions have been used as inputs for generating transfer functions. Then only from that transfer functions we have calculated the GAIN and PHASE margin related parameters and pole-zero values. In this way we have coupled one concept of digital signal processing with control system to show which data sets(REAL DATAS or COMPLEX DATAS)generate a stable transfer function. So our main aim is to judge the stability of a systemby using POLES and RELATIVE STABILITY [3] ANALYSIS. Side by side we have plotted the power spectral density of convoluted datas obtained after circular and linear convolutions.]

Keywords- Convolution , GAIN And PHASE , RELATIVE STABILITY

I. INTRODUCTION

As we have already mentioned that there are so many mathematical applications are there in digital signal processing; but remember one thing that we are not focusing on Analog signals because for our paper purpose we need digitally discrete numerical data's not the continuous domain analog signal values. Now we have selected two separate mathematical applications 'LINEAR CONVOLUTION' [1] and 'CIRCULAR CONVOLUTION '[2] and 'FAST FOURIER TRANSFORM' [3] because of the fact that we need to generate numerical data's that we will use as inputs in order to generate transfer functions. From that transfer function we can judge the location of poles and gain as well as phase margin values to realize the stability of the used transfer function. So let's talk about convolutions, suppose we have taken two different series of data sequences as inputs. Now multiplications of two different sequences of data's in time domain is called Linear Convolution and in frequency domain multiplication is called circular convolution. Here we have tested our codes by two different sequences of data's ; Real data's and Complex data's. Now you must take sequence of data's as 2 or 2^n that's means multiplications of 2 format. For our programming purpose we have use a set 8 real and 8 complex data's as inputs. So if both the sequences (ONE sequence called L, other one is M) have 8 data's then after circular convolution result will have number of data's=Max(L,M)[1] and after linear convolution result will have number of data's=L+M-1[2];But in case of fast Fourier transformation we use butterfly structure with the help of 2 or 2^n number of input data's and the same procedure will be used to generate the transfer functions with the help of data's converted from complex(x+ j*y) domain to modulus or magnitude format.

Now after generating the data's by using circular and linear convolution we will move in control system programming purpose ok, here by we will compare the values of GAIN and PHASE margin[3] then we will find out and compare how many poles are located at the 'S'=sigma + j*w[3] plane. If in both the transfer functions (generated fromlinear and circular convolutions by using prime real and complex data sequences) we have found out that in both the cases some poles are located in Right half of 'S' plane) then only we will compare the numerical values of GAIN and PHASE margins. From the values of Gain and Phase margin we will have to take decisions of stability analysis of both the system transfer function.

Then only we can conclude that which sequence of data's (Real or Complex) can generate a stable system. Here while generating the transfer function we have converted the complex data's (generated from circular convolution) in modular or magnitude format for the ease of matlab coding purpose. Now let us discuss about the Power Spectral Densityof a function; it means power consumed or dissipated per unit bandwidth. Now we can calculate the PSD for convoluted or auto-correlated signals; autocorrelation means convolution of a signal along with the time delayed version of the signal itself. Here for real and complex both the data's we have plotted the power spectral density spectrum of linear and circularly convoluted signals to investigate in which case we are getting uniformly distributed Gaussian Curveor not.

3.Steps Used In Matlab Programming

STEP 1: Take two normal or real data sequences and two complex conjugate data sequences

STEP 2: Perform linear and circular convolution on two real and complex data sequences separately

STEP 3: Now you will get one set means two separate data sequences each after performing both circular and linear convolutions; total four data sequences two in real data sequence format and rest two in complex data sequence format

STEP4: Now perform modulus operation on complex data sequences, we will get the magnitude only(sqrt(x^2+y^2)), now we will have to find out the transfer function by using mat-lab coding on both the sequences that means taking two datasets generated from original real and complex data sequences each separately in a group.

STEP 5: Now one by one find out gain margin, phase margin, gain and phase crossover frequencies, pole-zero values and at last plot of location of poles of both the transfer functions in graphical method.

STEP 6: Find out the power spectral density of four different circular and linear convolution data's of real and complex sets.

II.MATLAB PROGRAMMING CODES

1. Matlab Codes Of Linear And Circular Convolutions

```
syms x y x1 y1 h
 x=[1 2 3 4 5 6 7 8];
 y=[9 10 11 12 13 14 15 16];
 x1=[1+2i 3+4i 5+6i 7+8i 6+9i 4+7i 5+6i 2+6i];
 y1=[2+3i 6+5i 7+8i 8+5i 3+9i 3+6i 4+6i 7+6i];
subplot(2,2,1)
 h=conv(x,y);
 stem(h);
title('linear convolution');
subplot(2,2,2)
 xpad = [x zeros(1,8-length(x))];
 ypad = [y zeros(1,8-length(y))];
 ccirc = ifft(fft(xpad).*fft(ypad));
 stem(ccirc);
 title('circular convolutionwith real data');
subplot(2,2,3)
h1=conv(x1,y1);
stem(h1);
title('linear convolution with complex data');
subplot(2,2,4)
xpad = [x1 zeros(1,8-length(x1))];
ypad = [y1 zeros(1,8-length(y1))];
ccirc = ifft(fft(xpad).*fft(ypad));
stem(ccirc);
title('circular convolution with complex data');
Continuous-time transfer function.
```

2. Matlab Codes Of Fast Fourier Transform

```
 x=[1 2 3 4 5 6 7 8];
 x1=[1+2i 3+4i 5+6i 7+8i 6+9i 4+7i 5+6i 2+6i];
 y=fft(x);
 y1=fft(x1);
 subplot(2,2,1)
 stem(y);
 title('fast fourier transform of real data');
 subplot(2,2,2)
 stem(y1);
 title('fast fourier transform of complex data');
```

3. Matlab Code For Gain Mergin,Phase Margin, Pole,Zero Values Calculations

```
subplot(2,2,1)
 x11=[9 28 58 100 155 224 308 408 427 428 410 372 313 232 128];
 x12=[436 456 468 472 468 456 436 408];
c12=tf(x11,x12)
 plot(pole(c12),'*');
title('plot of poles of real data transfer function'
subplot(2,2,2)
x14=[.0806 .354 .90 1.725 2.655 3.516 4.17 4.69 4.73 4.183 3.54 2.582 1.68 1.118 .583];
 x15=[4.81 4.537 4.437 4.263 4.31 4.63 4.746 4.68];
 c16=tf(x14,x15); plot(pole(c16),'+');
 title ('plot of poles of complex data transfer function');
 [Gm,Pm,Wcg,Wcp] = margin(c16);
```

4. Matlab Code For Power Spectral Density

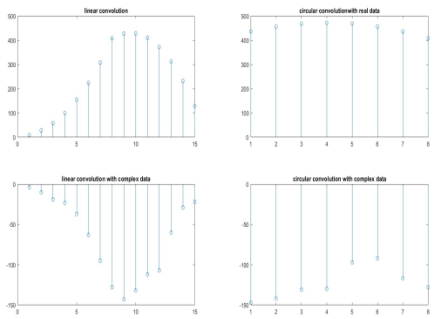
Calculation

```
Subplot (2,2,1)
x1=[9 28 58 100 155 224 308 408 427 428 410 372 313 232 128];
[H,F] = freqz(1,x1,[],1);
plot(F,20*log10(abs(H)));
xlabel('Frequency (Hz)');
```

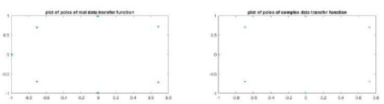
```
ylabel('PSD (dB/Hz)');
title('power spectral density of linear convolution real data');
subplot(2,2,2)
x2=[436 456 468 472 468 456 436 408];
[H,F] = freqz(1,x2,[],1);
plot(F,20*log10(abs(H)));
xlabel('Frequency (Hz)');
ylabel('PSD (dB/Hz)');
title('power spectral density of circular convolution real data');
subplot(2,2,3)
x3=[.0806 .354 .90 1.725 2.655 3.516 4.17 4.69 4.73 4.183 3.8 2.582 1.68 1.118 .583];
[H,F] = freqz(1,x3,[],1);
plot(F,20*log10(abs(H)));
xlabel('Frequency (Hz)');
ylabel('PSD (dB/Hz)');
title('power spectral density of linear convolution complex data');
subplot(2,2,4)
x4=[4.81 4.537 4.437 4.263 4.31 4.63 4.746 4.68];
[H,F] = freqz(1,x4,[],1);
plot(F,20*log10(abs(H)));
xlabel('Frequency (Hz)');
ylabel('PSD (dB/Hz)');
title('power spectral density of circular convolution complex data');
```

V.RESULTS

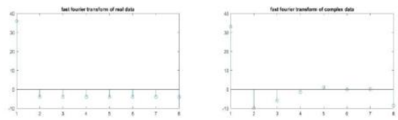
1. Circular And Linear Convoltion Plot



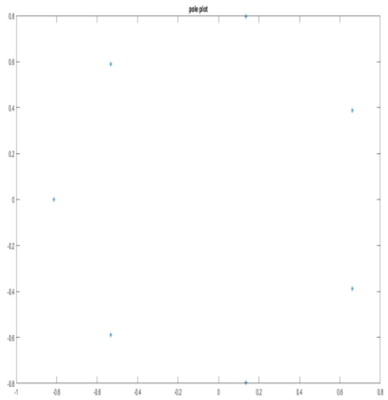
2.Graphical Pole Plots



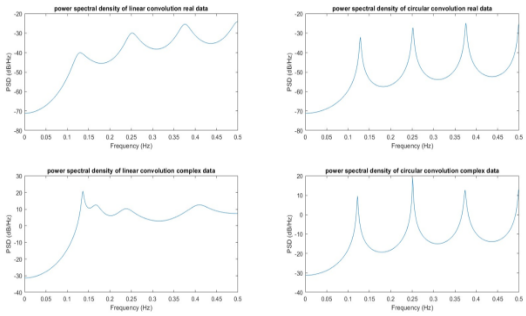
3. Graphical Plot Of Fast Fourier Transformation Of Real And Complex Datas



4. Graphical Pole Plot Of Transfer Function Of Fast Fourier Transformation



5. Power Spectral Density Plot In Four Different Cases Of Convolutions



6. Gain Mergin,Phase Margin, Pole,Zero Values With Table Of Calculations

For Real Set Of Datas	
Transfer Function	9 S^14 + 28 S^13 + 58 S^12 + 100 S^11 + 155 S^10 + 224 S^9 + 308 S^8 + 408 S^7 + 427 S^6 + 428 S^5 + 410 S^4 + 372 S^3 + 313 S^2 + 232 S + 128 -- 436 S^7 + 456 S^6 + 468 S^5 + 472 S^4 + 468 S^3 + 456 S^2 + 436 S + 408
Zeros	-1.0164 + 0.9455i -1.0164 - 0.9455i -1.3995 + 0.0000i -1.0863 + 0.0000i -0.7692 + 0.7670i -0.7692 - 0.7670i

	-0.0946 + 1.3479i -0.0946 - 1.3479i -0.0038 + 1.0860i -0.0038 - 1.0860i 0.8107 + 0.9836i 0.8107 - 0.9836i 0.7606 + 0.7731i 0.7606 - 0.7731i		
Poles	0.6849 + 0.7163i 0.6849 - 0.7163i -0.9903 + 0.0000i -0.7031 + 0.6974i -0.7031 - 0.6974i -0.0096 + 0.9904i -0.0096 - 0.9904i		
Gain Mergin	Phase Mergin	Gain Crossover Frequency	Phase Crossover Frequency
7.2770	29.2128	1.4831	1.8323
For Complex Set Of Datas			
Transfer Functions	0.806 S^14 + 0.354 S^13 + 0.9 S^12 + 1.725 S^11 + 2.655 S^10 + 3.516 S^9 + 4.17 S^8 + 4.69 S^7 + 4.73 S^6 + 4.183 S^5 + 3.8 S^4 + 2.582 S^3 + 1.68 S^2 + 1.118 S+ 0.58 --- 4.81 S^7 + 4.537 S^6 + 4.437 S^5 + 4.263 S^4 + 4.31 S^3 + 4.63 S^2 + 4.746 S + 4.68		
Zeros	0.9802 + 0.9462i 0.9802 - 0.9462i -0.5253 + 0.8391i -0.5253 - 0.8391i -0.9096 + 0.2486i -0.9096 - 0.2486i -0.6774 + 0.4631i -0.6774 - 0.4631i -0.0394 + 0.9595i -0.0394 - 0.9595i 0.3726 + 0.8409i 0.3726 - 0.8409i 0.5793 + 0.7182i 0.5793 - 0.7182i		
Poles	0.7251 + 0.6982i 0.7251 - 0.6982i -0.0076 + 0.9960i -0.0076 - 0.9960i -0.9901 + 0.0000i -0.6941 + 0.7041i -0.6941 - 0.7041i		
Gain Mergin	Phase Margin	Gain Cross Over Frequency	Phase Crossover Frequency
Inf	-115.6054	Nan	0.9853

Datas Used In Fast Fourier Transfomation			
Transfer Function	36 S^7 + 10.73 S^6 + 5.656 S^5 + 4.329 S^4 + 4 S^3 + 4.329 S^2 + 5.656 S + 10.73 -- 58.25 S^7 + 16.66 S^6 + 6.05 S^5 + 6.461 S^4 + 2.236 S^3 + 0.638 S^2 + 3 S + 11.5		
Zero	0.7180 + 0.4636i 0.7180 - 0.4636i 0.1049 + 0.8315i 0.1049 - 0.8315i -0.8343 + 0.0000i -0.5547 + 0.6234i -0.5547 - 0.6234i		
Poles	-0.8140 + 0.0000i -0.5324 + 0.5883i -0.5324 - 0.5883i 0.1349 + 0.7975i 0.1349 - 0.7975i 0.6615 + 0.3890i 0.6615 - 0.3890i		
Gain Mergin	Phase Mergin	Gain Cross Over Frequency	Phase Crossover Frequency
Inf	Inf	Nan	Nan

6. Datas Used In Transfer Function Calculations

Datas Used In Transfer Function 1	Input1=[9 28 58 100 155 224 308 408 427 428 410 372 313 232 128] Input2=[436 456 468 472 468 456 436 408]
Datas Used In Transfer Function 2	Input1 = [-0.0400 + 0.0700i , -0.1000 + 0.3400i , -0.1900 + 0.8800i , -0.2300 + 1.7100i , -0.3700 + 2.6300 ,0.6300 + 3.4600i , -0.9500 + 4.0600i , -1.2800 + 4.5100i , -1.4300 + 4.5100i , -1.3200 + 3.9700i -1.1200 + 3.3600i , -1.0700 + 2.3500i , -0.6000 + 1.5700i , -0.2900 + 1.0800i, -0.2200 + 0.5400i] Input2=[-1.4700 + 4.5800i -1.4200 + 4.3100i -1.3100 + 4.2400i -1.3000 + 4.0600i -0.9700 + 4.2000i -0.9200 + 4.5400i -1.1700 + 4.6000i -1.2800 + 4.5108]
Datas Used In Transfer Function 3	INPUT1=[36.0000 + 0.0000i , -4.0000 + 9.6569i , -4.0000 + 4.0000i , -4.0000 + 1.6569i , -4.0000 + 0.0000i, -4.0000 - 1.6569i, 4.0000 - 4.0000i , -4.0000 - 9.6569i] INPUT2=[33.0000 +48.0000i,-9.9497 - 13.3640i,-6.0000 + 1.0000i, -1.4645 - 6.2929i,1.0000 - 2.0000i,-0.0503 - 0.6360i ,0.0000 - 3.0000i,-8.5355 - 7.7071i]

III.DISCUSSIONS ON RESULTS

Take a close look on the table no 5.3 and observe the neumerical values poles both in complex and real sets of data we can see very carefully that in each cases 2 poles are lying right half of the s-plain and if you consider the transfer function generated by Fast Fourier Transformation datas there also you can observe that 4 poles are lying at the right half of thr complex plane.

Now see the pole plots very carefully for the cases of linear and circular convolution datas;there also you can see that 2 poles are lying at the right half of the s plain; so both systems are unstable according to ROUTH-HIRWITZ CRITAERION.Now lets go to table number 5.3 if you observe the GAIN and PHASE margin of both real and complex data systems you can easily say that gain margin of complex data system is infinity and phase margin of complex data system is negetive but in case of real data sets both gain and phase margins are having a definete finite positive values.

So we can say that transfer fuction genrated from the convolution results of real data sets is more stable than transfer function genrated from the convolution results of complex data sets. But if we observe the 'Gain Margin' and 'Phase Margin' of the FAST FOURIER TRANSFORMATION (FFT) we can easily conclude that the transfer function that we have used for the purpose of operations is not STABLE at all. If we observe the PSD functions in all four cases we can say that for the PSD of circular convolution datas one principal peak exists and two side peaks exist.But not in a single case we have seen that power spectral density function is uniformly distributed as a gaussian curve.

IV.FUTURE SCOPE OF WORK

Here we have have used a set of real and complex set of data sequences to proceed our convolution programming but we didn't realise the scope of application of 'CAUSAL'OR'MEMORYLESS' signals. So in future scope of work our aim will be to design a set of 'CAUSAL'OR'MEMORYLESS' signals by using a set of real and complex data sequences then we will perform DFT(Discrete fourier transform) to generate new data sequences. That new data sequences we will use to generate the transfer functions , then our final aim will be to use the characteristics equation of the transfer functions to build 'ROUTH-HIRWITZ' tables by using matlab codes . At last from that 'ROUTH-HIRWITZ' tables we will determine or judge the stability of the original real and complex data sets after DFT techniques.

V.CONCLUSIONS

Here we have used the concept of " BODE PLOT "ananlysis to calculate gain and phase margin of the systemso that at the end we can judge the stability of the two above mentioned systems; our request is that try the two original real and complex sequences in any othe types of convolution syste if the matlab codes are available for that operations. Then you can use NICHOLE'S CHART TRCHQUINE to justify the stability of the transfer functiongenerated from the convoluted datas.We hope that the results should not be changed.

REFERENCES

1."Digital Signal Processing"-Ramesh Babu , Latest Addtion
2."Signals And System"-H.P.Shu
3."Linear Control System"-B.S.Manke