**Lab 3： Fourier Series Representation of Periodic Signals**

|  |  |
| --- | --- |
| **Author** | Name： 谭雍昊 柳嘉伟 Student ID:11910204 11910123 |
| **Introduction**   1. Use MATLAB to calculate DTFS coefficients of a signal by fft(). 2. Use MATLAB to generate the original signal using its DTFS coefficients through ifft(). Use different sets of DTFS coefficients to see the difference. 3. Use MATLAB to calculate frequency response of the signal by freqz() and lsim(). 4. Use MATLAB to verify properties of Fourier series. 5. Use MATLAB to compare the algorithm complexity of different functions.   **Lab results & Analysis**：    **Question 3.5(a)**    **Analysis 3.5(a)**  , here , a0=1.    The result is a real number. Besides, a0 is real. So we expect x[n] to be purely real.  **Question 3.5(b)**    a = [1 2\*exp(-1i\*pi/3) exp(1i\*pi/4) exp(-1i\*pi/4) 2\*exp(1i\*pi/3)];  **Analysis 3.5(b)**    **Question 3.5(c)**    **D:\SUSTECH2020春\信号和系统\lab\HW3\3.5a.png3.5a**  Fig 3.5c  **Analysis 3.5(c)**  Fig 3.5c show us x[n] in one period. Our expectation in Part(a) is correct. x[n] is purely real.  **Question 3.5(d)**      **3.5e**  Fig 3.5d  **Analysis 3.5(d)**  Fig 3.5d shows us x1[n], x2[n], x3[n] over the range of 0≤n≤63.  **Question 3.5(e)**    **D:\SUSTECH2020春\信号和系统\lab\HW3\3.5b.png3.5b**  Fig 3.5e  **Analysis 3.5(e)**  Prediction:  As shown in Fig 3.5e, our predicted values match those obtained with MATLAB.  **Question 3.5(f)**    D:\SUSTECH2020春\信号和系统\lab\HW3\3.5c.png3.5c  Fig 3.5f  **Analysis 3.5(f)**  x3[n] is real, so ak=a\*-k. For x3[n], N=32 so there are 32 ak. The actual order of ak should be -16≤k≤15. However, these values are stored by [a3(1) ... a3(32)]. Therefore, from a-15 to a-1 are stored from a3(17) to a3(31). For example, a-2=a3(30), a-5=a3(27), a-10=a3(22). We get all the negative values of k.    As Fig 3.5f shows x3\_2[n], x3\_8[n], x3\_12[n], x3\_all[n].  **Question 3.5(g)**    **Analysis 3.5(g)**  x3[n] is real, so ak=a\*-k.As we had proved in Part(a), is real. Therefore, is real. , a16 is real, too. Thus  must be a real signal.  **Question 3.5(h)**    3.5d  Fig 3.5h  **Analysis 3.5(h)**  This time we use six different numbers of DTFS to generate x[n]. Fig 3.5h shows us that as we use more numbers of DTFS we use, x3[n] is more like its original signal. That will not display Gibb’s phenomenon. With discrete signals this is not a problem because the overshoot is eliminated when the last frequency is added.  **Advance Problem**    **Analysis**  function a = dtfs(x,n\_init)  a=[];  w0=2\*pi/length(x); %fundamental frequency  for k=n\_init:n\_init+length(x)-1 %period from 0+n0 to N-1+n0  aInPeriod=0;  for n=1:length(x)  aInPeriod=aInPeriod+x(n)\*exp(-j\*k\*w0\*(n+n\_init-1));  end  a=[a aInPeriod/length(x)];  end    %for 3.5, to get the same result as the examples, need to adjust order  if n\_init>0  for i=1:n\_init  a=[a(length(a)+1-i) a];  end  a=a(1:length(x));  end  if n\_init<0  np=-n\_init;  for i=1:np  a=[a a(i)];  end  a=a(np+1:length(a));  end  end  The code of the function is shown.      The results are totally the same as what is given in the example. Our function is working correctly by computing the DTFS coefficients for the signals given and the output of our function matches the outputs below.      **Question 3.8(a)**    **Analysis 3.8(a)**  In the format specified by filter and freqz,  a1=[1 -0.8];  b1=1;  a2=[1 0.8];  b2=1;  **Question 3.8(b)**    **3.8b**  Fig 3.8b  **Analysis 3.8(b)**  As Fig 3.8b shows, system 1 is a lowpass filter and system 2 is a highpass filter.  **Question 3.8(c)**  **D:\SUSTECH2020春\信号和系统\lab\HW3\3.8c.png3.8c**  Fig 3.8c  **Analysis 3.8(c)**  Fig 3.8c shows us the DTFS of x[n] together with frequency responses of system 1 and system 2.When x[n] is the input to the system, for system 1, when ω=0.3142 or 5.969,they will be amplified,while others will be attenuated. For system 2,when ω=2.827 or 3.456,they will be amplified,while others will be attenuated. In conclusion, for system 1, we state that low frequency components will be amplified, and others will be attenuated. For system 2, we state that high frequency components will be amplified, and others will be attenuated.  **Question 3.8(d)**    **3.8d**  Fig 3.8d  **Analysis 3.8(d)**  The signal x[n] is shown in Fig 3.8d.  **Question 3.8(e)**    3.8e  Fig 3.8e  **Analysis 3.8(e)**  Fig 3.8e shows us the outputs y1 and y2. We find that y2 has more high frequency energy and y1 has more low frequency energy. The plots confirm our answers in Part(c) “In conclusion, for system 1, we state that low frequency components will be amplified, and others will be attenuated. For system 2, we state that high frequency components will be amplified, and others will be attenuated.”  **Question 3.8(f)**    3.8f  Fig 3.8f  **Analysis 3.8(f)**  Fig 3.8f shows us the DTFS coefficients for y1 and y2. We find that for y1, low aks are amplified. For y2, high aks are amplified. Others are attenuated. These plots agree with our answers in Part(e).      **Question 3.9(a)**    **D:\SUSTECH2020春\信号和系统\lab\HW3\3.9a.png3.9a**  Fig 3.9a  **Analysis 3.9(a)**  When the input x(t)=cos(t),    Transfer function H(s)=1/(1+s). The output y(t) has    There is also another way to get y(t).    , for y(t), , so  **Question 3.9(b)**    3.9b  Fig 3.9b  **Analysis 3.9(b)**  Fig 3.9b shows the result y2(t).  **Question 3.9(c)**    3.9c  Fig 3.9c  **Analysis 3.9(c)**  **Code for ssum:**  apos\_k=[2\*sin(pi/2)/pi 2\*sin(3\*pi/2)/(3\*pi) 2\*sin(5\*pi/2)/(5\*pi) 2\*sin(7\*pi/2)/(7\*pi) 2\*sin(9\*pi/2)/(9\*pi)];  aneg\_k=apos\_k;  s1=apos\_k(1)\*exp(j\*t)+aneg\_k(1)\*exp(-j\*t);  s2=apos\_k(2)\*exp(3j\*t)+aneg\_k(2)\*exp(-3j\*t);  s3=apos\_k(3)\*exp(5j\*t)+aneg\_k(3)\*exp(-5j\*t);  s4=apos\_k(4)\*exp(7i\*t)+aneg\_k(4)\*exp(-7j\*t);  s5=apos\_k(5)\*exp(9j\*t)+aneg\_k(5)\*exp(-9j\*t);  ssum=s1+s2+s3+s4+s5;  Fig 3.9c shows us the sum of s1 to s5 and x2. We find that they are very similar.  **Question 3.9(d)**      **3.9d2**  Fig 3.9d-1  **D:\SUSTECH2020春\信号和系统\lab\HW3\3.9d.png3.9d**  Fig 3.9d-2  **Analysis 3.9(d)**  Fig 3.9d-1 shows us y1 to y5 separately. In Fig 3.9d-2, ysum is the sum of y1 to y5 and y\_ssum is the response to the signal ssum. We find that they are totally the same. Therefore, we verify the linear property of the system and the methods can both achieve the same results.  **Question 3.9(e)**    **3.9e**  Fig 3.9e  **Analysis 3.9(e)**  In Fig 3.9e, ys(t) is the response of the system to the sum of the first 5 harmonics to the response of the system. And y2(t) is the response to the original x2. We find that two plots are almost the same. According to Parseval relation,  we substitute the first 5 harmonics to the right, and get . The energy of the system will not change much if we apply enough aks, so here the energy of the sum of first 5 harmonics is nearly the energy of y(t).  **Question 3.9(f)**    **D:\SUSTECH2020春\信号和系统\lab\HW3\3.9f.png3.9f**  Fig 3.9f  **Analysis 3.9(f)**  Fig 3.9f shows us both the analytically determined and simulated signal of y1 to y5. We find that they are almost the same. Therefore, our signals y1 to y5 are correct.    **Question 3.10(a)**    **Analysis 3.10(a)**  It’s clear to see that for (n+1) numbers(n terms of x[n]\* and 1/N) the multiplication operations required to compute the DTFS is N+1，and the required addition operations for N numbers(additions of n terms of x[n]) is N-1, then the total number of required operations is 2N.  **Question 3.10(b)(c)**      Fig 3.10b  **Analysis 3.10(b)(c)**  Actually, due to the fact that function flops is outdated, we consider using function timeit to record the time required for the conv function to get the results. Besides, for each N, we calculate 10 times and get the average as the results to raise the accuracy. Also, we use loglog function to linearize the graph to make the results more clear and reduce the error following the instruction of the question. Then, from Fig3.10b we can draw two conclusions. Firstly, the required time for fft function to do the calculation is less than the dtfs function, which is written according to the definition of the Fourier series. Meanwhile, we can also learn that as the period N is made following the sequence 8,32,64,128,256, the larger N is, the difference between dtfs and fft function is more obvious, showing fft function is more relatively effective.  **Question 3.10(d)**  **Analysis 3.10(d)**  . Obviously, the period of y[n] is N, too. Besides, the multiplication operations required is N for each r, and r ranges from 0 to N-1, thus the number of multiplication operations required is N^2, meanwhile, the number of addition operations is N-1 for each r, while r ranges from 0 to N-1, thus the number of addition operations is N(\*N-1), plus the two results we get results, which is .  **Question 3.10(e)(f)**        Fig 3.10e    Fig 3.10f  **Analysis 3.10(e)(f)**  After extracting the results over the interval [0 N-1], we can successfully draw the graph of the y[n], the convolution of h[n] and x[n],which is clearly shown in the Fig3.10e and Fig 3.10f for N=40 and N=80 respectively.  **Question 3.10(g)(h)**      Fig 3.10g    Fig 3.10h  **Analysis 3.10(g)(h)**  After comparing the y[n] from difs function and fft function, we can easily seen from the Fig3.10d and Fig3.10e for N=40 and N=80 respectively, the two convolution results are the same utilizing the two methods.  **Question 3.10(i)**       |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | average | | f40c | 2.9177 | 2.8442 | 2.8535 | 2.9726 | 2.822 | 2.795 | 2.782 | 2.968 | 2.859 | 2.850 | 2.8664 | | f40f | 7.8483 | 7.9797 | 8.227 | 7.942 | 8.069 | 8.135 | 8.239 | 8.123 | 8.268 | 8.289 | 8.1120 | | f80c | 4.3498 | 4.3444 | 4.4908 | 4.507 | 4.368 | 4.355 | 4.821 | 4.705 | 5.327 | 5.273 | 4.6541 | | f80f | 1.1034 | 1.0166 | 1.0028 | 1.0475 | 1.0012 | 1.0410 | 1.0501 | 1.1732 | 1.0365 | 1.0431 | 1.0515 |   Table 3.10g  **Analysis 3.10(i)**  After comparing the y[n] from dtfs function and fft function, we can easily seen from the Fig3.10d and Fig3.10e for N=40 and N=80 respectively, the two convolution results are the same utilizing the two methods. We use ratio1 to represent the f40c/f40f, ratio to represent f80c/f80f, then we get ratio1=3.5335, ratio2=4.4262, Table3.10a is the data we get from ten times of operation, gaining the average to reduce the error.  **Note**: Please indicate meaning of the symbols in all expressions. Please indicate the coordinate and unit in all figures. | |
| **Experience**  It is a wonderful experience for us.  We completed more tasks in this exercise. The first is to learn how to use new functions such as freqz(). Then I learned about the Fast Fourier Transform and its functions fft() and ifft(), and wrote a simple algorithm to calculate DTFS, and later compared the efficiency with advanced algorithms.  Tan Yonghao: I found that what I learned in class was just not enough to complete this exercise. So I went through the textbooks, theoretical textbooks and even the previous notes on circuit basics to try to complete this exercise. Many times I can use MATLAB to write code that meets the requirements after reading the topic, but I don't understand it very well, so I have calculated and estimated by myself, and successfully mastered more skills. When doing 3.9, I tried to use more aks to try to achieve the original signal, and was likely to encounter Gibb’s phenomenon. At the same time, we encountered some problems when writing the advance problem of 3.5, which was different from the answer given in the title after calculating the result. After trying several times, we decided to change the output order, and finally it perfectly met the requirements of the problem.  Liu Jiawei: We are faced with more challenges in this lab on the utilization of new functions. Firstly, the flops function in the question 3.10 is out of fashion, thus we are requested to use functions like tic toc or timeit to record and compare the time for operations, instead. After attempting, we find that the time result of the same input is not exactly the same, confronted with such situation, my partner come up with the idea that we need to record 10 times and calculate the average to enhance the accuracy of the results, which works on the accuracy of the results a lot. However, I think that the N in the question needs to be larger for the illustration of the difference between fft functions and dtfs function made according to the definition.  The lab is a bit difficult for us. But after practicing, we manage to finished most of the tasks and gain more understandings. Thanks Dr. Wu for teaching us MATLAB skills and understanding of Fourier series, also thanks TA for marking our work and giving feedback. | |
| **Score** |  |

字体：英文Times new Roman；中文宋体，正文五号

Codes:

3.5

clc;

clear;

clf;

%3.5b

N1=5;

n1=0:1:4;

% a2=[1 0 exp(1i\*pi/4) 0 2\*exp(1i\*pi/3)];

% a1=[1 0 exp(-1i\*pi/4) 0 2\*exp(-1i\*pi/3)];

% x\_0=a(1);

% x\_1=a(2)\*exp(1i\*2\*pi\*n1/N1)+a1(2)\*exp(-1i\*2\*pi\*n1/N1);

% x\_2=a(3)\*exp(2i\*2\*pi\*n1/N1)+a1(3)\*exp(-2i\*2\*pi\*n1/N1);

% x\_3=a(4)\*exp(3i\*2\*pi\*n1/N1)+a1(4)\*exp(-3i\*2\*pi\*n1/N1);

% x\_4=a(5)\*exp(4i\*2\*pi\*n1/N1)+a1(5)\*exp(-4i\*2\*pi\*n1/N1);

% x\_sum=x\_0+x\_1+x\_2+x\_3+x\_4;

a=[1 2\*exp(-1i\*pi/3) exp(1i\*pi/4) exp(-1i\*pi/4) 2\*exp(1i\*pi/3)];

for i=2:5

x(i,:)=a(i)\*exp((i-1)\*1i\*2\*pi\*n1/5)+conj(a(i))\*exp(-(i-1)\*1i\*2\*pi\*n1/5);

end

x(1,:)=a(1);

x\_sum=sum(x);

xr=N1\*ifft(a);

figure(1)

stem(n1,xr,'m'),grid on,title('one period of x[n]'),xlabel('n'),ylabel('x[n]');

figure(7)

stem(n1,x\_sum,'-o'),grid on,title('one period of x[n]'),xlabel('n'),ylabel('x[n]');

%3.5d

n=0:1:63;

nx1=0:1:7;

nx2=0:1:15;

nx3=0:1:31;

x1=1.\*(nx1>=0);

x1n=[x1 x1 x1 x1 x1 x1 x1 x1];

x2=1.\*(nx2<=7);

x2n=[x2 x2 x2 x2];

x3=1.\*(nx3<=7);

x3n=[x3 x3];

figure(2)

subplot(311),stem(n,x1n,'b'),grid on,title('x\_1[n] over 0≤n≤63'),xlabel('n'),ylabel('x\_1[n]');

subplot(312),stem(n,x2n,'r'),grid on,title('x\_2[n] over 0≤n≤63'),xlabel('n'),ylabel('x\_2[n]');

subplot(313),stem(n,x3n,'m'),grid on,title('x\_3[n] over 0≤n≤63'),xlabel('n'),ylabel('x\_3[n]');

%3.5e

a1=fft(x1)/length(x1);

a2=fft(x2)/length(x2);

a3=fft(x3)/length(x3);

figure(3)

subplot(311),stem(nx1,abs(a1),'b'),grid on,title('DTFS of x\_1[n]'),xlabel('k'),ylabel('abs(a1)'),legend('abs(a1)');

subplot(312),stem(nx2,abs(a2),'r'),grid on,title('DTFS of x\_2[n]'),xlabel('k'),ylabel('abs(a2)'),legend('abs(a2)');

subplot(313),stem(nx3,abs(a3),'m'),grid on,title('DTFS of x\_3[n]'),xlabel('k'),ylabel('abs(a3)'),legend('abs(a3)');

%3.5fgh

for i=2:3

x3\_2(i,:)=a3(i)\*exp((i-1)\*1i\*2\*pi/32\*n)+conj(a3(i))\*exp(-(i-1)\*1i\*2\*pi/32\*n);

end

x3\_2(1,:)=a3(1);

x3\_2=sum(x3\_2);

for i=2:5

x3\_4(i,:)=a3(i)\*exp((i-1)\*1i\*2\*pi/32\*n)+conj(a3(i))\*exp(-(i-1)\*1i\*2\*pi/32\*n);

end

x3\_4(1,:)=a3(1);

x3\_4=sum(x3\_4);

for i=2:9

x3\_8(i,:)=a3(i)\*exp((i-1)\*1i\*2\*pi/32\*n)+conj(a3(i))\*exp(-(i-1)\*1i\*2\*pi/32\*n);

end

x3\_8(1,:)=a3(1);

x3\_8=sum(x3\_8);

for i=2:13

x3\_12(i,:)=a3(i)\*exp((i-1)\*1i\*2\*pi/32\*n)+conj(a3(i))\*exp(-(i-1)\*1i\*2\*pi/32\*n);

end

x3\_12(1,:)=a3(1);

x3\_12=sum(x3\_12);

for i=2:15

x3\_14(i,:)=a3(i)\*exp((i-1)\*1i\*2\*pi/32\*n)+conj(a3(i))\*exp(-(i-1)\*1i\*2\*pi/32\*n);

end

x3\_14(1,:)=a3(1);

x3\_14=sum(x3\_14);

for i=2:17

x3\_all(i,:)=a3(i)\*exp((i-1)\*1i\*2\*pi/32\*n)+conj(a3(i))\*exp(-(i-1)\*1i\*2\*pi/32\*n);

end

x3\_all(1,:)=a3(1);

x3\_all=sum(x3\_all);

figure(4)

subplot(411),stem((0:31),x3\_2(1:32),'b'),grid on,title('x\_{3\\_2}[n]'),xlabel('n'),ylabel('x\_{3\\_2}[n]');

subplot(412),stem((0:31),x3\_8(1:32),'r'),grid on,title('x\_{3\\_8}[n]'),xlabel('n'),ylabel('x\_{3\\_8}[n]');

subplot(413),stem((0:31),x3\_12(1:32),'g'),grid on,title('x\_{3\\_12}[n]'),xlabel('n'),ylabel('x\_{3\\_12}[n]');

subplot(414),stem((0:31),x3\_all(1:32),'m'),grid on,title('x\_{3\\_all}[n]'),xlabel('n'),ylabel('x\_{3\\_all}[n]');

figure(5)

subplot(611),stem((0:31),x3\_2(1:32),'black'),grid on,title('use 5 series'),xlabel('n'),ylabel('x\_{3\\_2}[n]'),legend('x\_{3\\_2}[n]');

subplot(612),stem((0:31),x3\_4(1:32),'b'),grid on,title('use 9 series'),xlabel('n'),ylabel('x\_{3\\_4}[n]'),legend('x\_{3\\_4}[n]');

subplot(613),stem((0:31),x3\_8(1:32),'r'),grid on,title('use 17 series'),xlabel('n'),ylabel('x\_{3\\_8}[n]'),legend('x\_{3\\_8}[n]');

subplot(614),stem((0:31),x3\_12(1:32),'g'),grid on,title('use 25 series'),xlabel('n'),ylabel('x\_{3\\_12}[n]'),legend('x\_{3\\_12}[n]');

subplot(615),stem((0:31),x3\_14(1:32),'c'),grid on,title('use 29 series'),xlabel('n'),ylabel('x\_{3\\_14}[n]'),legend('x\_{3\\_14}[n]');

subplot(616),stem((0:31),x3\_all(1:32),'m'),grid on,title('use all series'),xlabel('n'),ylabel('x\_{3\\_all}[n]'),legend('x\_{3\\_all}[n]');

function a = dtfs(x,n\_init)

a=[];

w0=2\*pi/length(x); %fundamental frequency

for k=n\_init:n\_init+length(x)-1 %period from 0+n0 to N-1+n0

aInPeriod=0;

for n=1:length(x)

aInPeriod=aInPeriod+x(n)\*exp(-j\*k\*w0\*(n+n\_init-1));

end

a=[a aInPeriod/length(x)];

end

%for 3.5, to get the same result as the examples, need to adjust order

if n\_init>0

for i=1:n\_init

a=[a(length(a)+1-i) a];

end

a=a(1:length(x));

end

if n\_init<0

np=-n\_init;

for i=1:np

a=[a a(i)];

end

a=a(np+1:length(a));

end

end

3.8

clc;

clear;

clf;

%3.8a

a1=[1 -0.8];

b1=1;

a2=[1 0.8];

b2=1;

%3.8b

[H1 Omega]=freqz(b1,a1,1024,'whole');

[H2 Omega]=freqz(b2,a2,1024,'whole');

figure(1),

%System 1 is a lowpass filter

subplot(211),plot(Omega,abs(H1),'b-'),grid on,xlabel('\omega'),ylabel('H\_1(e^{j\omega})'),title('frequency response of system 1');

%System 2 is a highpass filter

subplot(212),plot(Omega,abs(H2),'m-'),grid on,xlabel('\omega'),ylabel('H\_2(e^{j\omega})'),title('frequency response of system 2');

%3.8c

k=0:1:19;

ak=((3/4).\*((k==1)+(k==19)))+((-1/2).\*((k==9 )+(k==11)));

wk=(2\*pi/20).\*k;

figure(2),

subplot(311),stem(wk,ak),grid on,title('DTFS coefficients of x[n]'),xlabel('w\_k=(2pi/20)k'),ylabel('a\_k'),xlim([0 7]);

%System 1 is a lowpass filter

subplot(312),plot(Omega,abs(H1),'b-'),grid on,xlabel('\omega'),ylabel('H\_1(e^{j\omega})'),title('frequency response of system 1');

%System 2 is a highpass filter

subplot(313),plot(Omega,abs(H2),'m-'),grid on,xlabel('\omega'),ylabel('H\_2(e^{j\omega})'),title('frequency response of system 2');

n=0:1:19;

%3.8d&e

N=20;

x\_20=ifft(ak,N);

xn=N\*x\_20;

figure(3),

m=-20:1:99;

x=[xn xn xn xn xn xn];

stem(m,x),grid on,title('signal x[n]'),xlabel('n'),ylabel('x(n)');

figure(4),

y1=filter(b1,a1,x);

subplot(211),stem(m(21:120),y1(21:120)),grid on,title('y\_1[n] -filtered by system 1'),xlabel('n'),ylabel('y\_1[n]');

y2=filter(b2,a2,x);

subplot(212),stem(m(21:120),y2(21:120),'m'),grid on,title('y\_2[n] -filtered by system 2'),xlabel('n'),ylabel('y\_2[n]');

%3.8f

figure(5),

a\_y1=fft(y1(1:20))/20;

subplot(211),stem(n,abs(a\_y1),'r'),grid on,title('DTFS coefficients for y1'),xlabel('k'),ylabel('a\\_y1');

% a\_y2=fft(y2(1:20),20);

a\_y2=fft(y2(1:20))/20;

subplot(212),stem(n,abs(a\_y2),'b'),grid on,title('DTFS coefficients for y2'),xlabel('k'),ylabel('a\\_y2');

clc;

clear;

clf;

3.9

%3.9a&b

t=linspace(0,20,1000);

t1=linspace(10,20,500);

% Time range [0 to 20] with 1000 samples.

x1=cos(t);

RC=1; % time constant=1

sys=tf(1,[1 1]);

figure(1)

y1=lsim(sys,x1,t);

plot(t,x1),grid on,title('x(t) & y(t)'),xlabel('t'),xlim([10 20]);hold on;

plot(t,y1,'m'),legend('x(t)','y(t)'),ylabel('x(t) & y(t)');

x2=x1;

x2(x2>0)=ones(size(x2(x2>0)));

x2(x2<0)=-ones(size(x2(x2<0)));

figure(2)

y2=lsim(sys,x2,t);

plot(t,y2);grid on,axis([10,20,-1,1]),title('y\_2(t)'),xlabel('t'),ylabel('y\_2(t)');

%3.9c

% k=1:1:5;

apos\_k=[2\*sin(pi/2)/pi 2\*sin(3\*pi/2)/(3\*pi) 2\*sin(5\*pi/2)/(5\*pi) 2\*sin(7\*pi/2)/(7\*pi) 2\*sin(9\*pi/2)/(9\*pi)];

aneg\_k=apos\_k;

s1=apos\_k(1)\*exp(j\*t)+aneg\_k(1)\*exp(-j\*t);

s2=apos\_k(2)\*exp(3j\*t)+aneg\_k(2)\*exp(-3j\*t);

s3=apos\_k(3)\*exp(5j\*t)+aneg\_k(3)\*exp(-5j\*t);

s4=apos\_k(4)\*exp(7i\*t)+aneg\_k(4)\*exp(-7j\*t);

s5=apos\_k(5)\*exp(9j\*t)+aneg\_k(5)\*exp(-9j\*t);

ssum=s1+s2+s3+s4+s5;

figure(3)

plot(t,ssum);grid on,hold on;

plot(t,x2),xlabel('t'),ylabel('x\_2(t)&ssum(t)'),title('sum from s1 to s5 compared with x2'),legend('ssum(t)','x\_2(t)');

ksum=sum(apos\_k.^2+aneg\_k.^2);

%3.9d

y\_1=lsim(sys,s1,t);

y\_2=lsim(sys,s2,t);

y\_3=lsim(sys,s3,t);

y\_4=lsim(sys,s4,t);

y\_5=lsim(sys,s5,t);

y\_sum=y\_1+y\_2+y\_3+y\_4+y\_5;

ysum=lsim(sys,ssum,t);

figure(4)

plot(t,ysum,'r-');grid on;hold on,xlabel('t'),title('response to ssum and sum of y1 to y5'),ylabel('y\\_ssum & ysum');

plot(t,y\_sum,'b--'),legend('ysum','y\\_ssum');

%3.9e

figure(5)

plot(t,ysum,'b-'),grid on,title('y2(t) and ys(t)'),xlabel('t'),ylabel('y2(t) & ys(t)');hold on;

plot(t,y2,'m--'),legend('ys(t)','y2(t)');

Y\_1=1/(1+1i)\*apos\_k(1)\*exp(1i\*t)+1/(1-1i)\*aneg\_k(1)\*exp(-1i\*t);

Y\_2=1/(1+3\*1i)\*apos\_k(2)\*exp(3\*1i\*t)+1/(1-3\*1i)\*aneg\_k(2)\*exp(-3\*1i\*t);

Y\_3=1/(1+5\*1i)\*apos\_k(3)\*exp(5\*1i\*t)+1/(1-5\*1i)\*aneg\_k(3)\*exp(-5\*1i\*t);

Y\_4=1/(1+7\*1i)\*apos\_k(4)\*exp(7\*1i\*t)+1/(1-7\*1i)\*aneg\_k(4)\*exp(-7\*1i\*t);

Y\_5=1/(1+9\*1i)\*apos\_k(5)\*exp(9\*1i\*t)+1/(1-9\*1i)\*aneg\_k(5)\*exp(-9\*1i\*t);

figure(6)

subplot(5,2,1);plot(t1,y\_1(501:1000)),title('simulated signal y1'),xlabel('t');ylabel('y\_{1}(t)');grid on;

subplot(5,2,2);plot(t1,Y\_1(501:1000));title('analytically determined signal y1'),xlabel('t');ylabel('y\_{1}(t)');grid on;

subplot(5,2,3);plot(t1,y\_2(501:1000));xlabel('t');title('simulated signal y2'),ylabel('y\_{2}(t)');grid on;

subplot(5,2,4);plot(t1,Y\_2(501:1000));title('analytically determined signal y2'),xlabel('t');ylabel('y\_{2}(t)');grid on;

subplot(5,2,5);plot(t1,y\_3(501:1000));title('simulated signal y3'),plot(t1,y\_3(501:1000));xlabel('t');ylabel('y\_{3}(t)');grid on;

subplot(5,2,6);plot(t1,Y\_3(501:1000));title('analytically determined signal y3'),xlabel('t');ylabel('y\_{3}(t)');grid on;

subplot(5,2,7);plot(t1,y\_4(501:1000));title('simulated signal y4'),xlabel('t');ylabel('y\_{4}(t)');grid on;

subplot(5,2,8);plot(t1,Y\_4(501:1000));title('analytically determined signal y4'),xlabel('t');ylabel('y\_{4}(t)');grid on;

subplot(5,2,9);plot(t1,y\_5(501:1000));title('simulated signal y5'),xlabel('t');ylabel('y\_{5}(t)');grid on;

subplot(5,2,10);plot(t1,Y\_5(501:1000));title('analytically determined signal y5'),xlabel('t');ylabel('y\_{5}(t)');grid on;

3.103.10（b）(c)

clc;

clear;

N=[8 32 64 128 256 ];

x1= (0.9).^[0:N(1)-1];

x2= (0.9).^[0:N(2)-1];

x3= (0.9).^[0:N(3)-1];

x4= (0.9).^[0:N(4)-1];

x5= (0.9).^[0:N(5)-1];

dtfscompso=[zeros(1,5)];

fftcompso=[zeros(1,5)];

for i=1:10;

f1 = @()dtfs(x1,0);

f2 = @()dtfs(x2,0);

f3 = @()dtfs(x3,0);

f4 = @()dtfs(x4,0);

f5 = @()dtfs(x5,0);

dtfscompso = dtfscompso+[timeit(f1) timeit(f2) timeit(f3) timeit(f4) timeit(f5)]

h1 = @()fft(x1,N(1))

h2 = @()fft(x2,N(2));

h3 = @()fft(x3,N(3));

h4 = @()fft(x4,N(4));

h5 = @()fft(x5,N(5));

fftcompso = fftcompso+[timeit(h1) timeit(h2) timeit(h3) timeit(h4) timeit(h5)];

end;

dtfscomps=dtfscompso./10

fftcomps=fftcompso./10

loglog(N,dtfscomps,'-s',N,fftcomps,'-s');

legend('dtfscomps','fftcomps ','Location','northwest')

grid on;

xlabel('N');

ylabel('time');

title('time comparasion between dtsf and fft function');

3.10(e)(f)

N=40;

x=(0.9).^[0:N-1];

h=(0.5).^[0:N-1];

f400c=0;

f400f=0;

for i=1:10

f40=@()conv([x x],h);% Store the DTFS of x [n] in X

ax=fft(x,N);

ah=fft(h,N);

ay=ax.\*ah;

y2=conv([x x],h);

z2=ifft(ay,N);

h40=@()ifft(ay,N);

f400c=f400c+timeit(f40)

f400f=f400f+timeit(h40)

end

for i=1:N

y22(i)=y2(i);

end

f40c=f400c./10;

f40f=f400f./10;

stem([0:N-1],y22,'o-','color','r');

hold on

stem([0:N-1],z2,'s-','color','b');

legend('conv','ifft ','Location','northeastoutside')

grid on;

xlabel('N');

ylabel('y[n]');

title('N=80,comparasion between the results from conv and ifft');

3.10(g)(h)

N=80;

x=(0.9).^[0:N-1];

h=(0.5).^[0:N-1];

f800c=0;

f800f=0;

for i=1:10

f80=@()conv([x x],h);% Store the DTFS of x [n] in X

ax=fft(x,N);

ah=fft(h,N);

ay=ax.\*ah;

y2=conv([x x],h);

z2=ifft(ay,N);

for i=1:N

y22(i)=y2(i);

end

h80=@()ifft(ay,N);

f800c=f800c+timeit(f80)

f800f=f800f+timeit(h80)

end

f80c=f800c./10;

f80f=f800f./10;

stem([0:N-1],y22,'o-','color','r');

hold on

stem([0:N-1],z2,'s-','color','b');

legend('conv','ifft ','Location','northeastoutside')

grid on;

xlabel('N');

ylabel('y[n]');

title('N=80,comparasion between the results from conv and ifft');

3.10(i)

N1=40;

N2=80;

x1=(0.9).^[0:N1-1];

h1=(0.5).^[0:N1-1];

x2=(0.9).^[0:N2-1];

h2=(0.5).^[0:N2-1];

f400c=0;

f400f=0;

f800c=0;

f800f=0;

for i=1:10;

f40=@()conv([x1 x1],h1);

f80=@()conv([x2 x2],h2);% Store the DTFS of x [n] in X

ax1=fft(x1,N1);

ah1=fft(h1,N1);

ay1=ax1.\*ah1;

y1=conv([x1 x1],h1);

z1=ifft(ay1,N1);

ax2=fft(x2,N2);

ah2=fft(h2,N2);

ay2=ax2.\*ah2;

y2=conv([x2 x2],h2);

z2=ifft(ay1,N2);

h40=@()ifft(ay1,N1);

h80=@()ifft(ay2,N2);

f400c=f400c+timeit(f40)

f800c=f800c+timeit(f80)

f400f=f400f+timeit(h40)

f800f=f800f+timeit(h80)

end

f40c=f400c./10;

f40f=f400f./10;

f80c=f800c./10;

f80f=f800f./10;

ratio1=f40c./f40f

ratio2=f80c./f80f

%stem([0:N-1],y11,'o-','color','g');

%hold on

%stem([0:N-1],z1,'s-','color','r');

%legend('conv','ifft ','Location','northeastoutside')

%grid on;

%xlabel('N');

%ylabel('y[n]');

%title('N=40,comparasion between results from conv and ifft');