Lab 4 The Continuous-Time Fourier Transform

|  |  |
| --- | --- |
| Author | Name： 赵奕帆 崔俞崧 Student ID:11812020 11811305 |
| Introduction  In the lab, we learned the Continuous Time Fourier Transform. And we have three problems in the lab.  In 4.2, we learned how to compute the numerical approximation of CTFT.  In 4.5, we will learn how to find analytic expressions for the impulse responses of stable LTI systems whose inputs and outputs satisfy linear constant-coefficient differential equations.  In 4.6, we will explore amplitude modulation of Morse code messages.  Lab results & Analysis：  4.2        Here, , and  So,    tau=0.01;  T=10;  N=T/tau;  t=(0:tau:T-tau);  y=exp(-2 \* abs(t-5));    Y = fftshift(tau \* fft(y));  Here, we can use the fftshift to get the value of Y(jw).    w = -(pi/tau) + (0:N-1)\*(2\*pi / (N\*tau));  (The output is too large to put the screenshot here.)    X = exp(li\*5\*w).\*Y;  We can get that .    We can find that the difference of approximate X(jw) is small to theoretical X(jw) in low frequency, and is bigger in high frequency.      We can find that the abs magnitude of X(jw) and Y(jw) are same, but the angles are the different. The reason is that y(t) is time shifting of x(t).  4.5      We would find that the so    We find that then we get    From the problem (a) and problem (b), we know that the so it’s integrable.    then we find      We find that then we get    From the problem (d) and problem (e), we know that the so it’s integrable.    then we find  We find that then we get  From the problem (d) and problem (e), we know that the because the so it’s not causal.  4.6        Z = [dash dash dot dot];              In this case, we can see that most of the energy in the Fourier transformation will move beyond the filter's passband.      We may find that so the letter is D.  We may find that so the letter is S.  We may find that so the letter is P.  So it’s DSP in altogether.  代码：  代码：  4.2  % 4.2(a)  % 4.2(b)  tau=0.01;  T=10;  N=T/tau;  t=(0:tau:T-tau);  y=exp(-2 \* abs(t-5));  %4.2(c)  Y = fftshift(tau\*fft(y));  %4.2(d  w = -(pi/tau)+(0:N-1)\*(2\*pi/(N\*tau));  %4.2(e)  X = exp(1i\*5\*w).\*Y;  %4.2(f)  magnitudeX\_a = abs(X);  phaseX\_a = angle(X);  X2 = 1 ./ (2 + 1j \* w) + 1 ./ (2 - 1j \* w);  magnitudeX = abs(X2);  phaseX = angle(X2);  figure;  subplot(2, 1, 1);  semilogy(w, magnitudeX\_a , w, magnitudeX);  legend('approximation |X|', '|X| ', 'Location', 'northeast');  title('X magnitude and X approximation');  subplot(2, 1, 2);  hold on;  semilogy(w, phaseX\_a, w, phaseX);  legend('approximation |X|', '|X|', 'Location', 'northeast');  title('X phase and X approximation');  saveas(gcf, "P4\_2\_out1.png")  close;  %4.2(g)  magnitudeY = abs(Y);  phaseY = angle(Y);  figure;  subplot(2, 1, 1);  semilogy(w, magnitudeY , 'r--h',w, magnitudeX\_app, 'b');  legend('Y', 'X', 'Location', 'northeast');  title('magnitude of X and Y');  subplot(2, 1, 2);  hold on;  semilogy(w, phaseY, 'b');  semilogy(w, phaseX, 'g');  legend('Y', 'X', 'Location', 'northeast');  title('phase of X and Y');  saveas(gcf, "P4\_2\_out2.png")  4.5  % 4.5\_a  a1 = [1 1.5 0.5];  b1 = [1 -2];  % 4.5\_b  [r1, p1] = residue(b1, a1);  % 4.5\_d  a2 = [1 7 16 12];  b2 = [3 10 5];  [r2, p2] = residue(b2, a2);  % 4.5\_g  a3 = [1 0 -4];  b3 = -4;  % 4.5(h)  [r3, p3] = residue(b3, a3);  4.6  load ctftmod.mat  % 4.6\_a  z = [dash dash dot dot];  % 4.6\_b  figure;  freqs(bf, af)  title('Frequency response of bf and af');  saveas(gcf, "./P4\_6\_b.png")  close  % 4.6\_c  ydash = lsim(bf, af, dash, t(1:length(dash)));  ydot = lsim(bf, af, dot, t(1:length(dot)));  figure;  subplot(2,1,1)  hold on;  plot(t(1:length(dash)), dash);  plot(t(1:length(dash)), ydash);  legend('dash without lowpass filter', 'dash with lowpass filter', 'Location', 'northeast');  title('Dash');  xlabel('t');  subplot(2,1,2)  hold on;  plot(t(1:length(dot)), dot);  plot(t(1:length(dot)), ydot);  legend('dot without lowpass filter', 'dot with lowpass filter', 'Location', 'northeast');  title('Dot');  xlabel('t');  saveas(gcf, './P4\_6\_c.png')  close  % 4.6\_d  y = dash .\* cos(2\*pi\*f1\*t(1:length(dash)));  yo = lsim(bf, af, y, t(1:length(y)));  figure;  subplot(2,1,1)  hold on;  plot(t(1:length(dash)), y);  title('Without filter');  subplot(2,1,2)  plot(t(1:length(y)), yo);  title('With filter');  saveas(gcf, './P4\_6\_d.png')  close  % 4.6\_g  x1 = x .\* cos(2\*pi\*f1\*t(1:length(x)));  m1 = lsim(bf, af, x1, t(1:length(x1)));  x2 = x .\* sin(2\*pi\*f2\*t(1:length(x)));  m2 = lsim(bf, af, x2, t(1:length(x2)));  x3 = x .\* sin(2\*pi\*f1\*t(1:length(x)));  m3 = lsim(bf, af, x3, t(1:length(x3)));  figure;  subplot(3,1,1)  plot(t, m1);  title('m1');  xlabel('t');  subplot(3,1,2)  plot(t, m2);  title('m2');  xlabel('t');  subplot(3,1,3)  plot(t, m3);  title('m3');  xlabel('t');  saveas(gcf, "./P4\_6\_g.png")  close  Note: Please indicate meaning of the symbols in all expressions. Please indicate the coordinate and unit in all figures. | |
| Experience  You can write your experience with this project. Any comment and suggestion on this course are also very welcome. | |
| Score |  |