**Lab 4：The Continuous-Time Fourier Transform**

|  |  |
| --- | --- |
| **Author** | Name：刘叶充&陈子蔚 Student ID:11912716 & 11910926 |
| **Introduction:**  The assignment of the lab 4 focus on the properties and applications of CTFT. The major goals includes to understand how to approximate CTFT by MATLAB, to understand the relation between frequency response and impluse response, and to learn easy application of CTFT.  **Lab results & Analysis**：  **4.2(a):** The expression of CTFT of is  It can be calculated as following steps  let  Since,  And  Therefore,  **4.2(b):** Create a vector containing the samples of , for and T=10  The figure of is shown below in *4.2\_Fig.1*.    **4.2(c):** Calculating by command , the resulting figure is shown in *4.2\_Fig.2*.    **4.2(d):** Construct the vector as the question wanted.  %b,c,d  tau = 0.01;  T = 10;  t = 0:tau:T - tau;  N = T / tau;  x = exp(-2 \* abs(t));  y = exp(-2 \* abs(t - 5));  Y = fftshift(tau \* fft(y));  w = -(pi / tau) + (0:N - 1) \* (2 \* pi / (N \* tau));  figure  plot(0:N - 1, real(y));  xlabel('t');  ylabel('y(t)');  title('4.2\\_Fig.1')  figure  subplot(2, 1, 1)  plot(w, real(Y));  xlabel('w');  ylabel('Re\{Y(jw)\}');  title('4.2\\_Fig.2')  hold on  subplot(2, 1, 2)  plot(w, imag(Y));  xlabel('w');  ylabel('Im\{Y(jw)\}');  **4.2(e):**  As ,  Thus,  Calculating and its figure is shown in *4.2\_Fig.3*    %e  X = exp(1j \* (5) \* w) .\* Y;  figure  subplot(2, 1, 1)  plot(w, real(X))  xlabel('w');  ylabel('Re\{X(jw)\}');  title('4.2\\_Fig.3')  hold on  subplot(2, 1, 2)  plot(w, imag(X))  hold on  xlabel('w');  ylabel('Im\{X(jw)\}');  **4.2(f):** Use two ways to calculate and print it in same figure *4.2\_Fig.4*  We can find the magnitude and angle is almost the same. (though image part same to be different, but the difference is actually lower than .  However, if use command semilogy to plot the magnitude of calculating by the two ways. We can find that the approximation at the high frequency is not as good as which in low frequency (shown in *4.2\_Fig.5*).  Changing the value of to a larger value plot the magnitude of by command semilogy again (shown in *4.2\_Fig.6*), we found the approximation is much better.      %f  X1 = zeros(1, length(w));  for i = 1:length(w)      X1(i) = 1 / (2 + 1j \* w(i)) + 1 / (2 - 1j \* w(i));  end  figure  subplot(2, 1, 1)  plot(w, abs(X), '--', 'LineWidth', 1.5);  hold on  plot(w, abs(X1))  hold on  xlabel('w');  ylabel('magnituide of X(jw)');  title('4.2\\_Fig.4')  subplot(2, 1, 2)  plot(w, angle(X))  hold on  plot(w, angle(X1))  hold on  xlabel('w');  ylabel('phrase angle of X(jw)');  figure  semilogy(w, abs(X));  xlabel('w');  ylabel('log|X(jw)|');  title('4.2\\_Fig.5')  hold on  semilogy(w, abs(X1));  xlabel('w');  ylabel('log|X2(jw)|');  tau = 0.001;  t = 0:tau:T - tau;  N = T / tau;  w = -(pi / tau) + (0:N - 1) \* (2 \* pi / (N \* tau));  X1 = zeros(1, length(w));  y = exp(-2 \* abs(t - 5));  Y = fftshift(tau \* fft(y));  X = exp(1j \* (-5) \* w) .\* Y;  for i = 1:length(w)      X1(i) = 1 / (2 + 1j \* w(i)) + 1 / (2 - 1j \* w(i));  end  figure  semilogy(w, abs(X));  hold on  semilogy(w, abs(X1));  xlabel('w');  ylabel('log|X(jw)|');  title('4.2\\_Fig.6')  **4.2(g):** Plot the magnitude and phrase angle of and on the same figure *4.2\_Fig.7*, we find that the magnitude of and is the same but the angle is different.    %g  figure  subplot(2, 1, 1)  plot(w, abs(X), '--', 'LineWidth', 1.5);  hold on  plot(w, abs(Y))  hold on  xlabel('w');  ylabel('magnituide of X(jw) and Y(jw)');  title('4.2\\_Fig.7')  subplot(2, 1, 2)  plot(w, unwrap(angle(X)));  hold on  plot(w, unwrap(angle(Y)))  hold on  xlabel('w');  ylabel('phrase angle of X(jw) and Y(jw)');  **4.5(a):** The differential equation is  Thus, the frequency response is  So, the and  %a  b1 = [1 -2];  a1 = [1 1.5 0.5];  **4.5(b):** Use command residue to calculate the value of r1 and p1, which is shown in figure below  Thus the ,  As , this function returns correct answer  The    %b  [r1, p1] = residue(b1, a1);  figure  subplot(2, 1, 1)  stem(r1, '\*', 'LineWidth', 2)  ylabel('r1')  title('4.5\\_Fig.1')  hold on  subplot(2, 1, 2)  stem(p1, '\*', 'LineWidth', 2)  hold on  ylabel('p1')  **4.5(c):** As  Thus,  So,  as  is absolutely integrable.  **4.5(d):**  The differential equation is  Thus, the frequency response is  So, the and  %d  b2 = [3 10 5];  a2 = [1 7 16 12];  **4.5(e):**  Use command residue to calculate the value of r1 and p1, which is shown in figure below  Thus the ,  As , this function returns correct answer  The    %e  [r2, p2] = residue(b2, a2);  figure  subplot(2, 1, 1)  stem(r2, '\*', 'LineWidth', 2);  hold on  title('4.5\\_Fig.2')  subplot(2, 1, 2)  stem(r1, '\*', 'LineWidth', 2)  ylabel('r2')  stem(p2, '\*', 'LineWidth', 2);  hold on  ylabel('p2')  **4.5(f):** As  We can get  as  is absolutely integrable.  **4.5(g):** The differential equation is  Thus, the frequency response is  So, the and  %g  b3 = -4;  a3 = [1 0 -4];  **4.5(h):**  Use command residue to calculate the value of r1 and p1, which is shown in figure below  Thus the ,  As , this function returns correct answer  The    %h  [r3, p3] = residue(b3, a3);  figure  subplot(2, 1, 1)  stem(r3, '\*', 'LineWidth', 2);  hold on  title('4.5\\_Fig.3')  subplot(2, 1, 2)  stem(r1, '\*', 'LineWidth', 2)  ylabel('r3')  stem(p3, '\*', 'LineWidth', 2);  hold on  ylabel('p3')  **4.5(i):** As  We can get  as  is absolutely integrable.  As when , thus it is not causal.  **4.6(a):** *4.6 Fig.1* shows the original signal of ‘Z’ through Morse Code.    %lab3\_6.m  load ctftmod.mat;  %a  z = [dash dash dot dot];  figure()  plot(t, z);  xlabel('t');  ylabel('z(t)');  title('4.6 Fig.1');  grid;  **4.6(b):** *4.6 Fig.2* shows the magnitude and the phase angle of the frequency response with **unwrap** used.    %b  omega = linspace(0, 50 \* pi, 200);  H = freqs(bf, af, omega);  figure()  subplot(211);  plot(omega, abs(H));  xlabel('\omega');  ylabel('|H(j\omega)|');  title('4.6 Fig.2');  grid;  subplot(212);  plot(omega, unwrap(angle(H)));  xlabel('\omega');  ylabel('angle(H)');  grid;  **4.6(c):** *4.6 Fig.3* shows the comparison between the original signals and the output signals of **dash** and **dot**. According to the figure in (b), the filter is obviously a lowpass filter. Since **dash** and **dot** are each composed of low frequency components, the output signals should be quite similar to the original signals, which matches the figure.    %c  ydash = lsim(bf, af, dash, t(1:length(dash)));  ydot = lsim(bf, af, dot, t(1:length(dot)));  figure()  subplot(211);  plot(t(1:length(dash)), dash);  xlabel('t');  title('4.6 Fig.3');  hold on;  plot(t(1:length(dash)), ydash);  legend('x\_{dash}', 'y\_{dash}');  grid;  subplot(212);  plot(t(1:length(dot)), dot);  xlabel('t');  hold on;  plot(t(1:length(ydot)), ydot);  legend('x\_{dot}', 'y\_{dot}');  grid;  **4.6(d):** *4.6 Fig.4* shows the signals of and the output signal of it through the filter. When the signal dash is modulated by , the frequency components of gaining signal is converge at or , then the energy of the output signal will be low after passing a lowpass filter. We can easily find that the magnitude of the signal is much smaller after the signal, which means the energy has moved outside.    %d  y = dash .\* cos(2 \* pi \* f1 \* t(1:length(dash)));  yo = lsim(bf, af, y, t(1:length(dash)));  figure()  plot(t(1:length(dash)), y);  xlabel('t');  hold on;  plot(t(1:length(dash)), yo);  legend('y(t)', 'yo(t)');  title('4.6 Fig.4');  grid;  **4.6(e):**  As for :  When , we have  As for :  When , we have  As for :  When , we have  **4.6(f):**  According to (e),  Therefore, we can use the given lowpass filter to get the approximate solution of with the method discussed.  As shown in *4.6 Fig.5*, is consist of **dash**es and **dot**s shown in *4.6 Fig.3* corresponding to different time. is ‘dash dot dot’, which can be found ‘D’ in the letter table of Mors Code.    **4.6(g):**  Use the same method in (f), and are accessible shown in *4.6 Fig.5*.  is ‘dot dot dot’, which means the letter ‘S’. is ‘dot dash dash dot’, which means the letter ‘P’. Ultimately, the future of technology life lies in DSP.  %f,g  figure()  subplot(311)  x1 = x .\* cos(2 \* pi \* f1 \* t(1:length(x)));  m1 = lsim(bf, af, x1, t(1:length(x)));  plot(t(1:length(m1)), 2 \* m1);  xlabel('t');  ylabel('m\_{1}(t)');  title('4.6 Fig.5');  grid;  subplot(312)  x2 = x .\* sin(2 \* pi \* f2 \* t(1:length(x)));  m2 = lsim(bf, af, x2, t(1:length(x)));  plot(t(1:length(m2)), 2 \* m2);  xlabel('t');  ylabel('m\_{2}(t)');  grid;  subplot(313)  x3 = x .\* sin(2 \* pi \* f1 \* t(1:length(x)));  m3 = lsim(bf, af, x3, t(1:length(x)));  plot(t(1:length(m3)), 2 \* m3);  xlabel('t');  ylabel('m\_{2}(t)');  grid; | |
|  | |
| **Score** |  |