|  |
| --- |
| **实验摘要：**  实验目的：   1. 熟悉使用matlab求解信号傅立叶变化、傅立叶反变换的方法 2. 熟悉使用matlab求解信号拉普拉斯变换、拉普拉斯反变换的方法 3. 熟悉使用拉普拉斯变换求解零状态响应、冲击响应、阶跃响应的方法   实验要求：  使用matlab正确求解傅立叶变换、反变换，正确求解拉普拉斯变换、拉普拉斯反变换，零状态响应、冲击响应、阶跃响应 |
| **实验题目** |

|  |
| --- |
| **实验内容**  **1.**  根据信号的时域频域关系可知，非周期信号f(t)可看成是周期T -> ∞时的周期信号，为了描述非周期信号的频谱特性，引入频谱密度的概念。令，  再使用傅立叶级数针对变量进行替换，可以得到    而针对本题目的两个函数，可以分别使用上述公式进行结果计算，可以得到结果：  而使用matlab进行傅立叶变换和反变换求解时，可以使用fourier和ifourier函数，使用syms定义符号常量t和w后直接调用函数即可。  运行结果如下：    针对结果进行绘图，如下图所示： |
| 2.  背景和matlab中的具体操作和同题目1相同，此处直接给出结果图片：    由图可以看出，信号进行时移变化后，其傅立叶变换的信号幅度并为发生变化，这符合理论知识；而相位谱发生了显著变化，这是因为对于因果时移的信号，其傅立叶变换会乘，因此当对信号取幅度时，其幅度不变而相位发生变化。  3.  对于拉普拉斯反变换，部分分式展开法适用于具有复根的多项式形式的频域函数。它的基本思想是将频域函数拆分为多个简单的分式形式，每个分式对应一个复根。然后，使用拉普拉斯反变换表格中的已知拉普拉斯对应关系将每个简单分式反变换回时域函数，最后将所有分式对应的时域函数相加，得到完整的时域函数 |
| 在matlab中，可以使用residue函数实现分式分解，之后可以使用ilaplace函数将各个部分转换为拉普拉斯时域函数。  求解结果如下：    针对求解结果画图：    4.  要求解拉普拉斯系统H(s)的极点和零点，并判断系统的稳定性，可以按照以下步骤进行：   1. 将系统的传递函数H(s)表示为分子项和分母项的比值形式：H(s) = N(s)/D(s)，其中N(s)是分子多项式，D(s)是分母多项式。 2. 零点是使得分子项N(s)等于零的s值。通过解方程N(s) = 0，找到所有零点。 3. 极点是使得分母项D(s)等于零的s值。通过解方程D(s) = 0，找到所有极点。 4. 判断系统的稳定性： |
| * 如果所有极点都位于左半平面（实部为负），则系统是稳定的。 * 如果存在任何一个极点位于右半平面（实部为正），则系统是不稳定的。 * 如果所有极点的实部都为零，或者有一些极点在虚轴上，则需要进一步进行分析。   在matlab中，可以使用tf函数创建传递函数模型，之后可以对模型直接使用pzmap函数求出信号系统对应的极点和零点坐标。绘制分布图如下：    由图可知，对系统H1，其有极点实部大于0，即在右半平面，故系统不稳定；对于系统H2，其所有极点实部均小于0，即全在左半平面，故系统稳定。  5.  针对本题，使用拉普拉斯反变换求冲击响应和阶跃响应时，首先根据分子分母创建传递函数模型，得到系统对象，之后便可以使用 ilaplace(Hs)求出对应的冲击响应；使用 ilaplace(Hs/s)求出对应的阶跃响应。  针对f(t)的零状态响应，我选择对系统使用lsim函数进行零状态响应数值解的计算。  结果如下： |
| 冲击  阶跃  对于整个题目的图片： |
| **实验总结**   1. 熟悉使用matlab求解信号傅立叶变化、傅立叶反变换、拉普拉斯变换、拉普拉斯反变换，并使用拉普拉斯变换求的系统响应 2. 首次针对题目求解析解而非数值解 3. 再次练习了使用画图 |
| **参考文献**   1. 信号与线性系统分析第5版 2. Matlab Documentation |
| **问题一**  % 求解 f(t) = e^(-2|t|) 的傅立叶变换  syms t w  f\_t = exp(-2\*abs(t));  F\_w = fourier(f\_t, t, w);  % 绘制 f(t) 和 F(w) 的图像  figure;  subplot(2, 1, 1);  fplot(f\_t, [-5, 5], 'LineWidth', 2);  title('f(t) = e^{-2|t|}');  xlabel('t');  ylabel('f(t)');  subplot(2, 1, 2);  fplot(F\_w, [-5, 5], 'LineWidth', 2);  title('F(w) = Fourier Transform of f(t)');  xlabel('w');  ylabel('F(w)');  % 求解 F(w) = 1/(1+w^2) 的傅立叶反变换  F\_w = 1/(1+w^2);  f\_t = ifourier(F\_w, w, t);  % 绘制 F(w) 和 f(t) 的图像  figure;  subplot(2, 1, 1);  fplot(F\_w, [-5, 5], 'LineWidth', 2);  title('F(w) = 1/(1+w^2)');  xlabel('w');  ylabel('F(w)');  subplot(2, 1, 2);  fplot(f\_t, [-5, 5], 'LineWidth', 2);  title('f(t) = Inverse Fourier Transform of F(w)');  xlabel('t');  ylabel('f(t)'); |
| **问题二**  % 清空环境变量  clear;  clc;  % 加载符号工具箱  syms t;  % 定义函数  f1 = 1/2 \* exp(-2\*t) \* heaviside(t);  f2 = 1/2 \* exp(-2\*(t-1)) \* heaviside(t-1);  % 计算傅里叶变换  F1 = fourier(f1);  F2 = fourier(f2);  % 频率范围  f = -10:0.01:10;  % 将符号表达式转换为函数并进行数值计算  F1\_abs\_func = matlabFunction(abs(F1));  F2\_abs\_func = matlabFunction(abs(F2));  F1\_phase\_func = matlabFunction(angle(F1));  F2\_phase\_func = matlabFunction(angle(F2));  % 计算幅度谱  F1\_abs = F1\_abs\_func(f);  F2\_abs = F2\_abs\_func(f);  % 绘制幅度谱  figure;  subplot(2,1,1);  plot(f, F1\_abs);  title('Amplitude Spectrum of f1(t)');  xlabel('Frequency');  ylabel('Amplitude');  subplot(2,1,2);  plot(f, F2\_abs);  title('Amplitude Spectrum of f2(t)');  xlabel('Frequency');  ylabel('Amplitude');  beauty\_plot  % 计算相位谱  F1\_phase = F1\_phase\_func(f);  F2\_phase = F2\_phase\_func(f);  % 绘制相位谱  figure;  subplot(2,1,1);  plot(f, F1\_phase);  title('Phase Spectrum of f1(t)');  xlabel('Frequency');  ylabel('Phase'); |
| subplot(2,1,2);  plot(f, F2\_phase);  title('Phase Spectrum of f2(t)');  xlabel('Frequency');  ylabel('Phase');  beauty\_plot |
| **问题三**  syms s t  % 定义输入函数 H(s)  H = (s+1)\*(s+4) / (s\*(s+2)\*(s+3));  % 求部分分式分解  [A, B, C] = residue([1, 5, 4], [1, 5, 6, 0]);  % 计算每个部分分式的拉普拉斯反变换  H1 = A(1)/s;  H2 = B(1)/(s+2);  H3 = 0; % 初始化 H3  for i = 1:numel(C)  H3 = H3 + C(i)/(s+3);  end  % 求和得到 H(t)  Ht = ilaplace(H1, s, t) + ilaplace(H2, s, t) + ilaplace(H3, s, t);  % 绘制 H(s) 的图像  subplot(2, 1, 1);  fplot(H, [-10, 10], 'LineWidth', 2); % 可根据需要调整范围  xlabel('s');  ylabel('H(s)');  title('H(s)图像');  grid on;  % 绘制 H(t) 的图像  subplot(2, 1, 2);  fplot(Ht, [0, 10], 'LineWidth', 2); % 可根据需要调整时间范围  xlabel('t');  ylabel('H(t)');  title('H(t)图像');  grid on;  beauty\_plot |
| **问题四**  % 定义系统函数 H1(s) 和 H2(s)  num1 = [1 2]; % H1(s) 的分子多项式系数  den1 = [1 1 2 6]; % H1(s) 的分母多项式系数  num2 = [1 0 1]; % H2(s) 的分子多项式系数  den2 = [3 5 4 6]; % H2(s) 的分母多项式系数 |
| % 创建系统模型  sys1 = tf(num1, den1);  sys2 = tf(num2, den2);  % 绘制零点和极点分布  figure;  % 绘制 H1(s) 的零点和极点分布  subplot(2, 1, 1);  pzmap(sys1);  hold on;  % 获取 H1(s) 的零点和极点的坐标  [z1, p1] = pzmap(sys1);  % 绘制 H1(s) 的零点和极点  scatter(real(z1), imag(z1), 'ro', 'LineWidth', 2);  scatter(real(p1), imag(p1), 'bx', 'LineWidth', 2);  % 添加标题和图例  title('H1(s) 零点和极点分布');  legend('H1(s) 零点', 'H1(s) 极点');  % 绘制 H2(s) 的零点和极点分布  subplot(2, 1, 2);  pzmap(sys2);  hold on;  % 获取 H2(s) 的零点和极点的坐标  [z2, p2] = pzmap(sys2);  % 绘制 H2(s) 的零点和极点  scatter(real(z2), imag(z2), 'ro', 'LineWidth', 2);  scatter(real(p2), imag(p2), 'bx', 'LineWidth', 2);  % 添加标题和图例  title('H2(s) 零点和极点分布');  legend('H2(s) 零点', 'H2(s) 极点');  % 判断系统的稳定性  if all(real(z1) < 0)  disp('H1(s) 是稳定系统');  else  disp('H1(s) 是不稳定系统');  end  if all(real(z2) < 0)  disp('H2(s) 是稳定系统');  else  disp('H2(s) 是不稳定系统');  end |
| **问题五**  % 定义系统传递函数  num = [1 0]; % 分子多项式的系数  den = [1 3 2]; % 分母多项式的系数  sys = tf(num, den); % 创建传递函数模型  % 转换为符号表达式  syms s;  Hs = poly2sym(num, s) / poly2sym(den, s);  % 计算冲击响应  impulse\_response = ilaplace(Hs);  % 转换为函数句柄  h\_impulse = matlabFunction(impulse\_response);  % 计算阶跃响应  step\_response = ilaplace(Hs/s);  % 转换为函数句柄  h\_step = matlabFunction(step\_response);  % 定义输入信号  t = 0:0.01:5; % 时间范围  f = cos(20\*t).\*(t>=0); % 输入信号 f(t) = cos(20t)ε(t)  % 计算零状态响应  zero\_state\_response = lsim(sys, f, t);  % 绘制冲击响应  subplot(3, 1, 1);  plot(t, h\_impulse(t));  xlabel('时间');  ylabel('冲击响应');  title('系统的冲击响应');  % 绘制阶跃响应  subplot(3, 1, 2);  plot(t, h\_step(t));  xlabel('时间');  ylabel('阶跃响应');  title('系统的阶跃响应');  % 绘制零状态响应  subplot(3, 1, 3);  plot(t, zero\_state\_response);  xlabel('时间');  ylabel('零状态响应');  title('零状态响应');  beauty\_plot  % 显示图形  sgtitle('系统的响应'); |
| **beauty\_plot函数部分**  function beautiplot(model)  % --input : model  % --model : 'default'  % : 'small'  % --usage :  % beautiplot  % beautiplot('small')  %%  if nargin<1  model = 'default';  end  fig=gcf;  % fontname = 'Times New Roman';  fontname = 'STFangSong';  %% set Figure  set(fig,'Units','pixels');  %%  figchildren = fig.Children;  for ax\_i = 1 : length(figchildren)  figChildren = figchildren(ax\_i);  if strcmp(figChildren.Type,'axes') == 1  ax = figChildren;  class1=ax.Children;  %% set Axes  % 字体  set(ax,'FontName',fontname,'FontSize',10,'FontWeight','bold');  % 网格  set(ax,'XGrid','on','YGrid','on','GridLineStyle','--','GridAlpha',0.15);  % 边框  set(ax,'Box','off','LineWidth',1.5);  % 刻度  set(ax,'XMinorTick','on','YMinorTick','on');  % 标题    switch model  case 'default'  set(ax.XLabel,'FontSize',12,'FontWeight', 'bold');%normal  set(ax.YLabel,'FontSize',12,'FontWeight', 'bold');  set(ax.ZLabel,'FontSize',12,'FontWeight', 'bold');  set(ax.Title,'FontSize',13,'FontWeight', 'bold');  case 'small'  set(ax.XLabel,'FontSize',10,'FontWeight', 'bold');%normal  set(ax.YLabel,'FontSize',10,'FontWeight', 'bold');  set(ax.ZLabel,'FontSize',10,'FontWeight', 'bold');  set(ax.Title,'FontSize',11,'FontWeight', 'bold');  end  %% set Legend  % try  %  set(ax.Legend,'Location','best','FontName',fontname,...%'Orientation','horizontal'  % 'FontWeight', 'bold');  % catch |
| try  set(ax.Legend,'FontName',fontname,...%'Orientation','horizontal'  'FontWeight', 'bold');  catch    end  %% set class  for i=1:length(class1)  % line(线图)  if strcmp(class1(i).Type,'line')==1  set(class1(i),'LineWidth',2,'MarkerSize',6);  % scatter(散点图)  elseif strcmp(class1(i).Type,'scatter')==1  set(class1(i),'MarkerFaceColor',[0.15,0.15,0.15],...  'LineWidth',2,...  'MarkerFaceAlpha',0.8);  % bar(柱状图)  % elseif strcmp(class1(i).Type,'bar')==1  % set(class1(i),'FaceColor',[0.47,0.67,0.19],'EdgeColor',[0 0 0],...  % 'LineWidth',1.5);  elseif strcmp(class1(i).Type,'bar')==1  set(class1(i),...  'LineWidth',1.5);  % text(文字)  elseif strcmp(class1(i).Type,'text')==1  set(class1(i),'FontSize',12,'FontName',fontname,'FontWeight', 'bold');  end  end  end  end    end |