

安徽大学人工智能学院

《数字信号处理》

实验案例设计报告

课程名称: 数字信号处理实验

专 业: 人工智能

班 级: 人工智能二班

学 号: WA2214014

姓 名: 杨跃浙

任课老师: 谭春雨

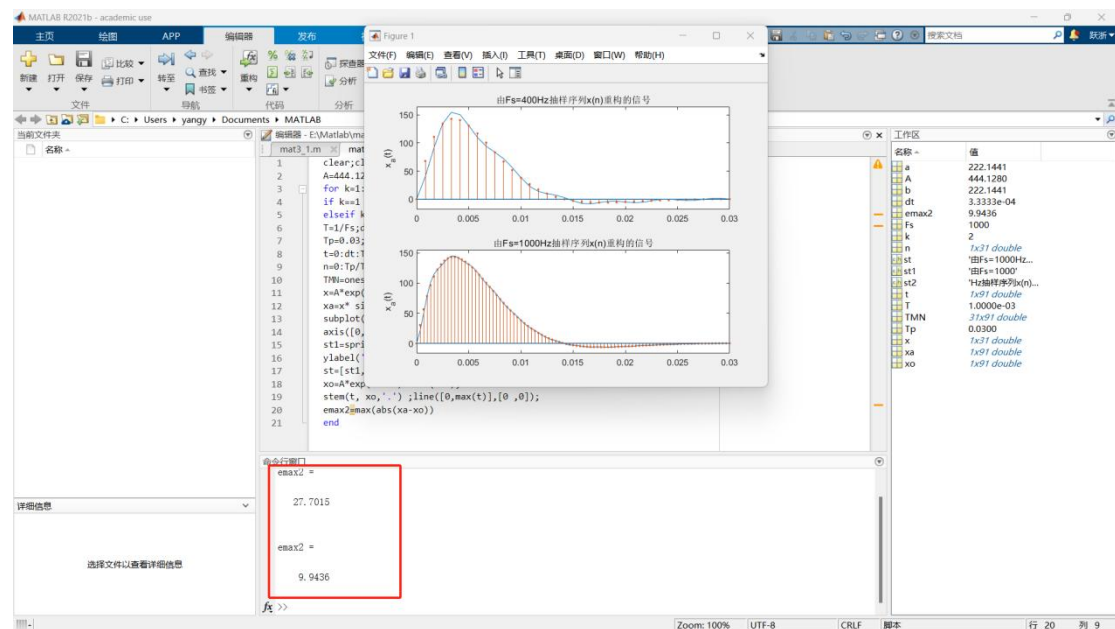
实验名称	实验二	实验次序	02
实验地点	笃行南楼 A301	实验日期	05.27
<p>实验内容:</p> <p>3-2</p> <p>实验目的</p> <p>本实验的主要目的是探究不同抽样频率对信号重构质量的影响。通过在 400 Hz 和 1000 Hz 两种抽样频率下，对信号进行抽样和重构，我们可以观察和比较恢复波形的差异。实验还旨在计算和分析在不同抽样频率下，重构信号与原始信号之间的最大恢复误差。这有助于理解抽样定理在实际应用中的效果，以及高于或等于奈奎斯特率抽样的必要性。</p> <p>实验原理</p> <p>实验基于信号处理中的抽样定理，该定理指出，为了无失真地重构一个带限信号，抽样频率必须至少是信号最高频率的两倍。在本实验中，信号 $x(t)$ 由指数衰减的正弦波构成，表达式为 $x(t) = Ae^{-at}\sin(bt)$，其中 A, a 和 b 是已知常数。首先，我们使用两种不同的抽样频率 F_s 对信号进行抽样，产生抽样点集 $x(n) = Ae^{-anT}\sin(bnT)$，其中 $T = \frac{1}{F_s}$ 是抽样间隔。随后，利用 sinc 函数作为插值函数，通过理想低通滤波器的方式，对离散抽样点进行重构，以获得重构信号 $x_a(t)$。计算重构信号 $x_a(t)$ 与原始信号 $x(t)$ 的最大差异 e_{\max}，即最大恢复误差，以评估重构的准确性。实验通过 MATLAB 编程实现上述过程，通过图形界面直观地显示原始信号和重构信号，以及不同抽样频率下的性能差异。</p> <p>实验代码</p> <pre>clear;close all;clc; A=444.128;a=50*sqrt(2)* pi;b=a; for k=1:2 if k==1 Fs=400; elseif k==2 Fs=1000;end T=1/Fs;dt=T/3; Tp=0.03; t=0:dt:Tp; n=0:Tp/T;</pre>			

```

TMN=ones(length(n),1)*t-n'*T*ones(1,length(t));
x=A*exp(-a.*n*T).*sin(b*n*T);
xa=x* sinc(Fs* TMN);
subplot(2,1,k);plot(t, xa);hold on ;
axis([0,max(t),min(xa)-10,max(xa)+10]);
st1=sprintf('由 Fs=%d',Fs);st2='Hz 抽样序列 x(n)重构的信号';
ylabel('x_a(t)');
st=[st1,st2];title(st)
xo=A*exp(-a.*t).*sin(b*t);
stem(t, xo,'r');line([0,max(t)],[0,0]);
emax2=max(abs(xa-xo))
end

```

实验结果



3-3

实验目的

本实验旨在通过对信号 $f(t) = \sin(120\pi t) + \cos(50\pi t) + \cos(60\pi t)$ 的时域波形和频谱特性的分析，了解不同抽样频率对信号重构的影响。实验包括绘制原始信号的时域波形及其幅频特性曲线，对信号进行不同抽样频率下的抽样，绘

制抽样序列的波形, 并对抽样序列进行频谱分析以比较不同抽样频率下的频谱特性, 最终通过抽样序列恢复原始信号, 并比较恢复信号与原始信号在时域上的差异。通过这一系列步骤, 深入理解采样定理的实际应用及其对信号处理的影响。

实验原理

实验首先定义了信号 $f(t) = \sin(120\pi t) + \cos(50\pi t) + \cos(60\pi t)$, 该信号包含三个不同频率的正弦和余弦成分。通过 MATLAB 绘制该信号的时域波形, 可以直观地观察信号随时间的变化。然后, 通过对信号进行傅里叶变换, 得到其幅频特性曲线, 分析信号在频域上的分布情况。接下来, 分别采用 80Hz、120Hz 和 150Hz 三种抽样频率对信号进行抽样, 得到不同抽样频率下的抽样序列。利用抽样序列绘制其波形, 并进行频谱分析, 绘制不同抽样频率下的幅频特性曲线, 通过对比分析不同抽样频率下频谱的变化, 观察频谱折叠现象。最后, 通过理想低通滤波器对抽样序列进行重构, 绘制重构后的时域波形, 并与原始信号进行对比, 分析不同抽样频率下重构信号与原始信号的差异, 从而评估不同抽样频率对信号重构质量的影响。

实验代码

```
f1='sin(2*pi*60*t)+cos(2*pi*25*t)+cos(2*pi*30*t)';  
fs0=caiyang(f1,80);  
fr0=huifu(fs0,80);  
fs1=caiyang(f1,120);  
fr1=huifu(fs1,120);  
fs2=caiyang(f1,150);  
fr2=huifu(fs2,150);
```

函数 caiyang()

```
function fz=caiyang(fy,fs)  
fs0=10000;tp=0.1;  
t=[-tp:1/fs0:tp];  
k1=0:999;k2=-999:-1;  
m1=length(k1);m2=length(k2);  
f=[fs0*k2/m2,fs0*k1/m1];  
w=[-2*pi*k2/m2,2*pi*k1/m1];  
fx1=eval(fy);  
FX1=fx1 * exp(-j*[1:length(fx1)]*w);  
figure;  
subplot(2,1,1);plot(t,fx1,'r');  
title('原信号');xlabel('时间 t/s');  
axis([min(t),max(t),min(fx1),max(fx1)]);  
subplot(2,1,2);plot(f,abs(FX1),'r')
```

```

title('原信号幅度频谱');xlabel('频率 f/Hz');
axis([-100,100,0,max(abs(FX1))+5]);
Ts=1/fs;
t1=-tp:Ts:tp;
f1=[fs * k2/m2,fs * k1/m1];
t=t1;
fz=eval(fy);
FZ=fz * exp(-j* [1:length(fz)]' * w);
figure;
subplot(2,1,1);stem(t,fz,'r');
title('抽样信号');xlabel('时间 t/s');
line([min(t),max(t)], [0,0]);
subplot(2,1,2);plot(f1,abs(FZ),'m');
title('抽样信号幅度频谱');xlabel('频率 f/Hz');

```

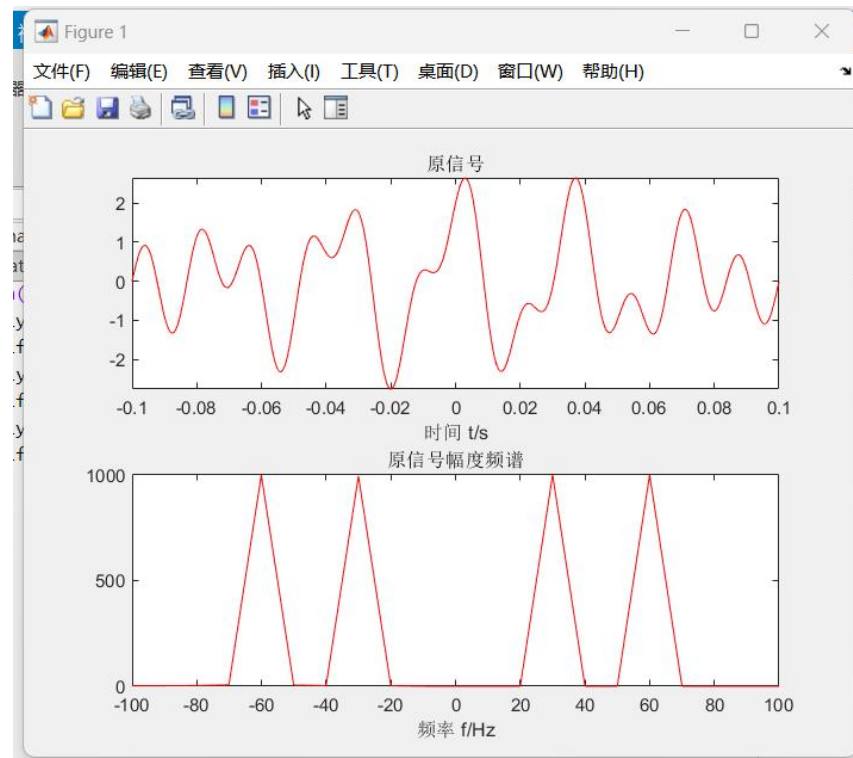
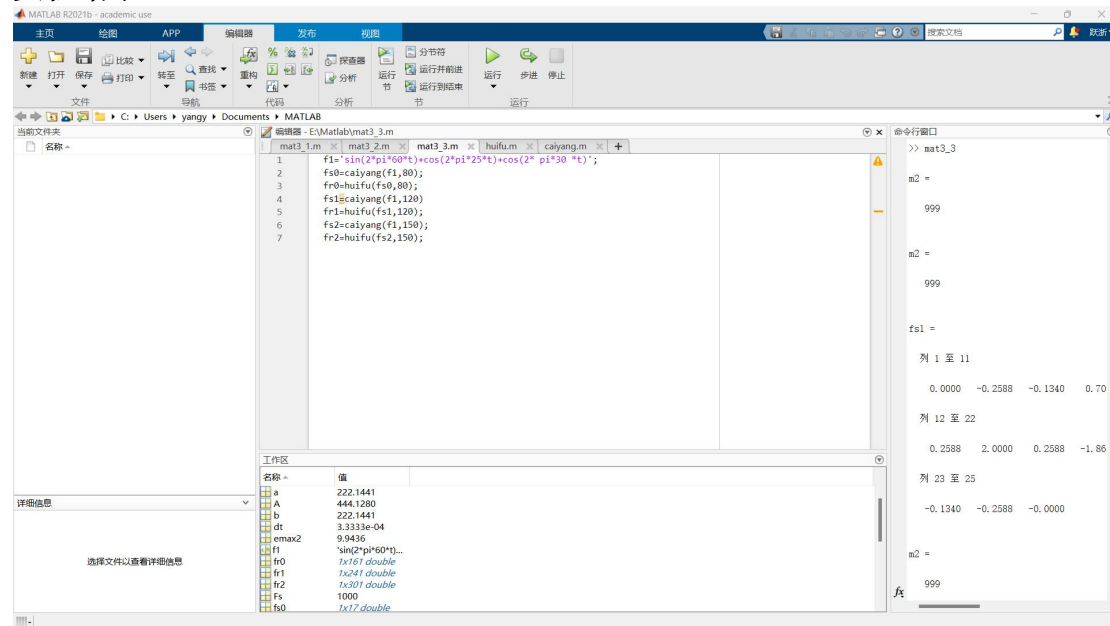
函数 huifu ()

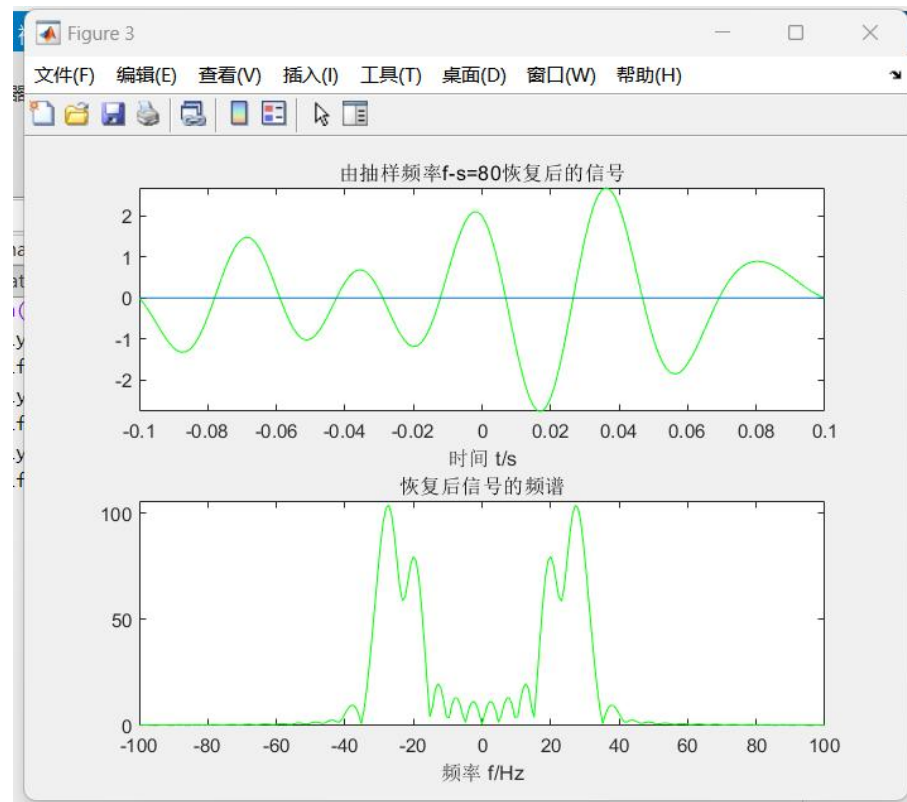
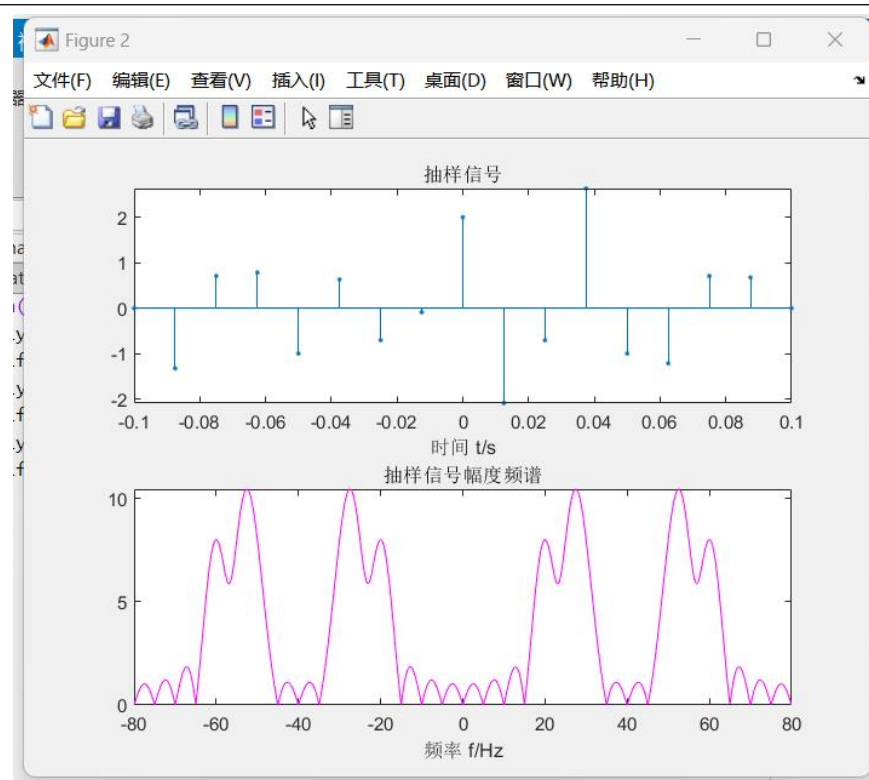
```

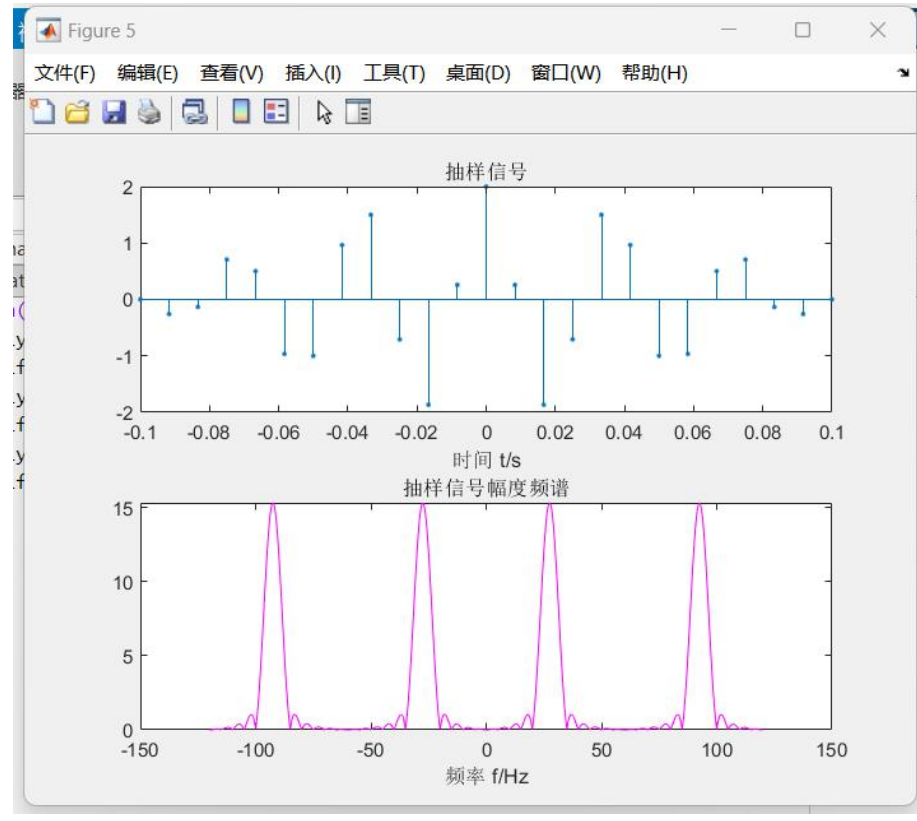
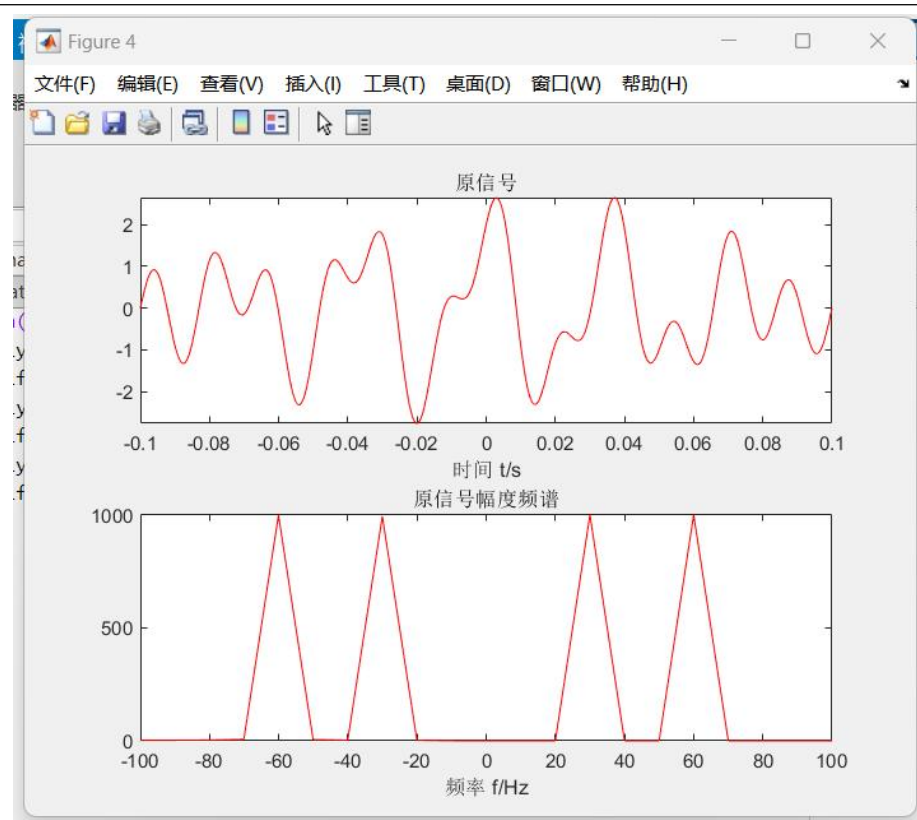
function fh=huifu(fz,fs)
T=1/fs;dt=T/10;
tp=0.1;t=-tp:dt:tp;
n=-tp/T:tp/T;
TMN=ones(length(n),1)* t-n'* T* ones(1,length(t));
fh=fz * sinc(fs * TMN);
k1=0:999;k2=-999:-1;
m1=length(k1);m2=length(k2);
w=[-2*pi*k2/m2,2*pi* k1/m1];
FH=fh* exp(-j*[1 :length(fh)]' * w);
figure;
subplot(2,1,1);plot(t,fh,'g');
st1=sprintf('由抽样频率 f-s=%d',fs);
st2='恢复后的信号';
st=[st1,st2];
title(st);xlabel('时间 t/s');
axis([min(t),max(t),min(fh),max(fh)]);
line([min(t),max(t)], [0,0]);
f=[10 * fs* k2/m2,10 * fs * k1/m1];
subplot(2,1,2);plot(f,abs(FH),'g');
title('恢复后信号的频谱');xlabel('频率 f/Hz');
axis([-100,100,0,max(abs(FH))+2]);

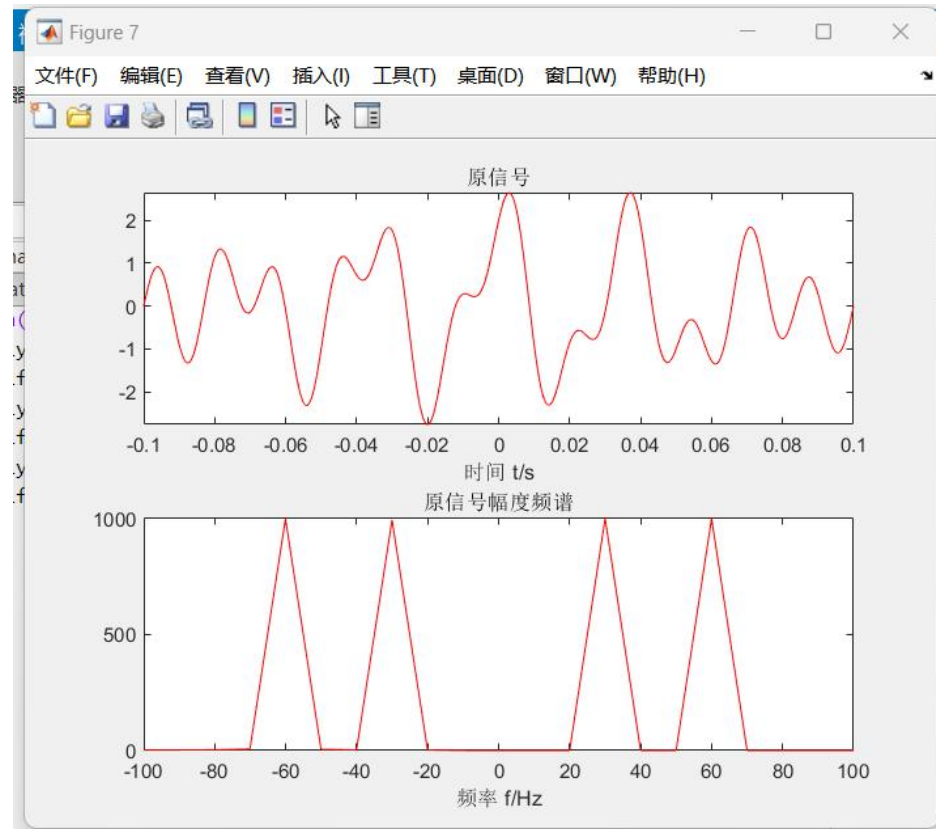
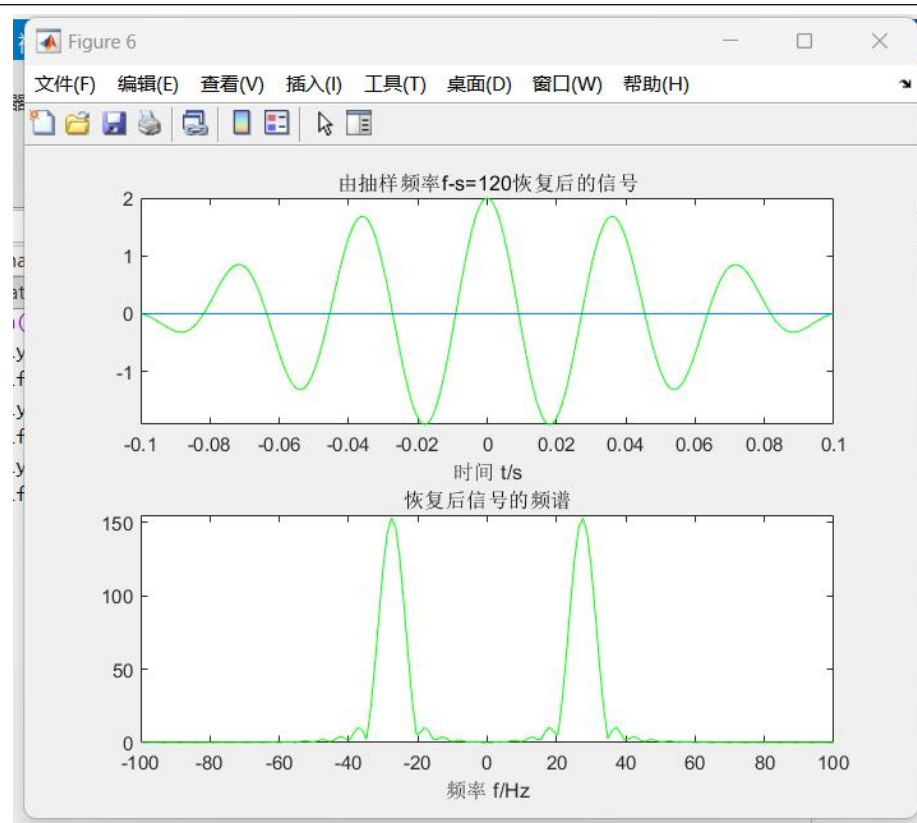
```

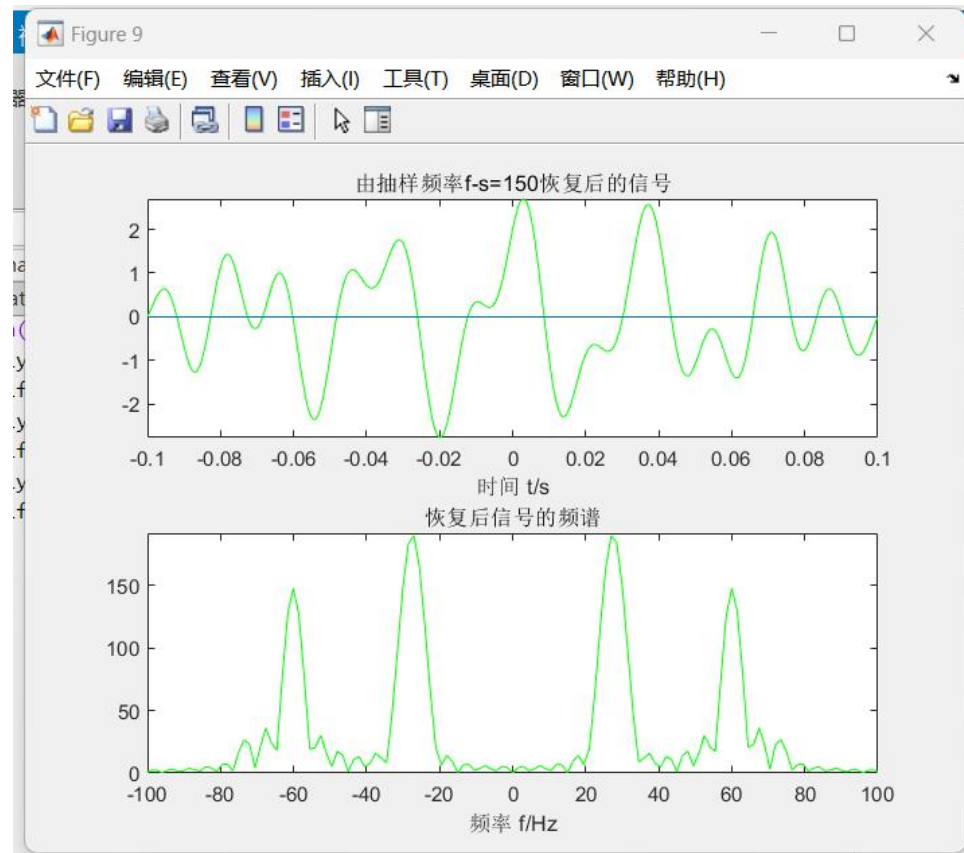
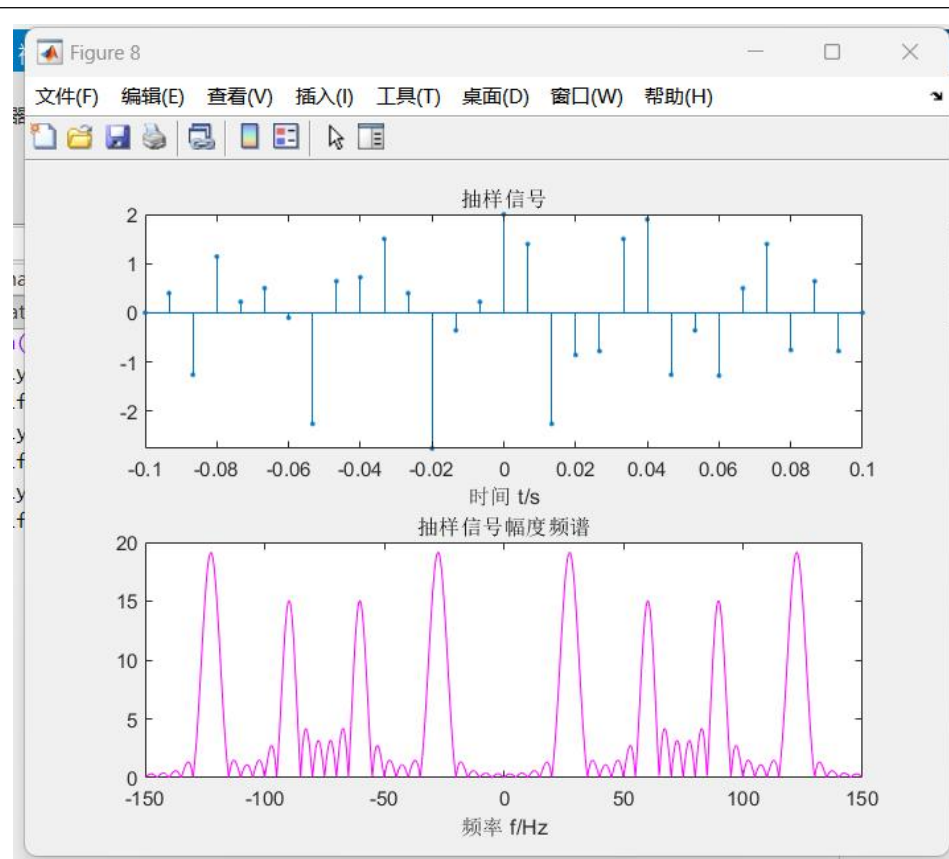
实验结果











P59, 1

实验目的

本实验旨在通过求解不同数学函数序列的 Z 变换, 深入理解 Z 变换的概念和其在信号处理中的应用。通过具体计算几种典型函数序列的 Z 变换, 如指数序列、多项式序列和三角函数序列, 本实验帮助学生掌握 Z 变换的理论基础和计算方法, 同时通过实际操作加深对离散时间系统分析的理解。

实验原理

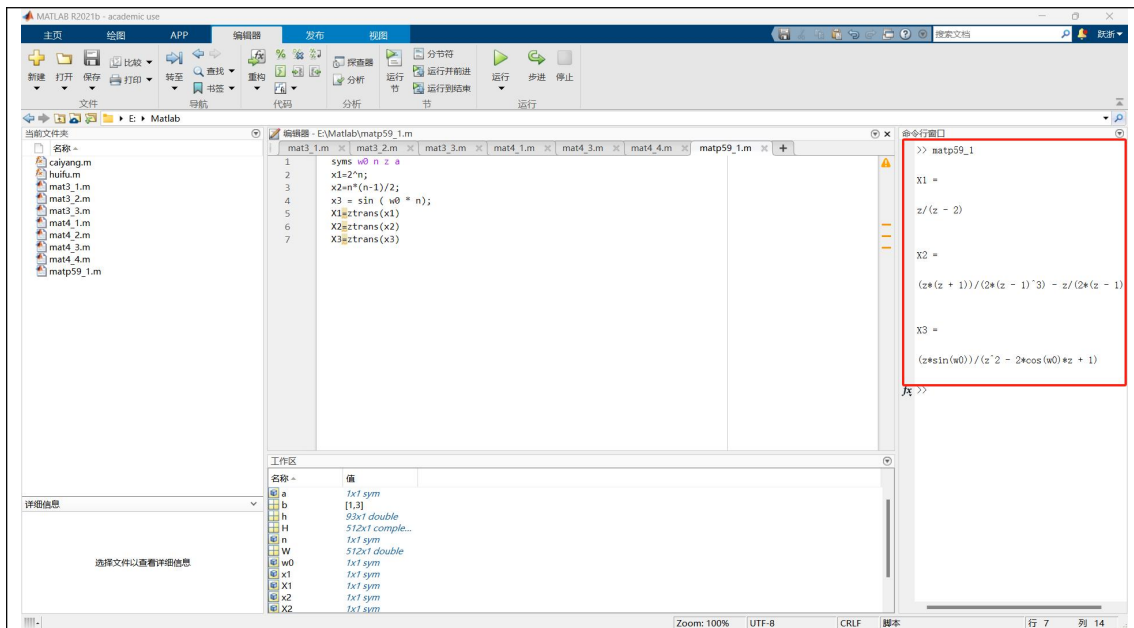
Z 变换是离散时间信号分析中的一个基本工具, 用于将离散时间信号从时域转换到复频域。实验中将对三种不同类型的函数序列进行 Z 变换: 指数函数

2^n , 多项式函数 $\frac{n(n-1)}{2}$, 以及三角函数 $\sin(w_0 n)$ 。指数函数的 Z 变换能展示如何处理具有指数增长特性的序列, 多项式函数的 Z 变换则涉及到序列的时间增长率和累积效应, 而三角函数的 Z 变换则是分析周期性信号的重要方法。通过这些序列的 Z 变换, 可以探究不同类型函数在 Z 域中的表示及其物理意义, 例如极点、零点以及它们对系统行为的影响。这些理论和技术是理解和分析各种数字信号处理问题的基础。

实验代码

```
syms w0 n z a
x1=2^n;
x2=n*(n-1)/2;
x3 = sin ( w0 * n);
X1=ztrans(x1)
X2=ztrans(x2)
X3=ztrans(x3)
```

实验结果



P59, 2

实验目的

本实验旨在通过计算 Z 逆变换，从复频域信号恢复到离散时间域信号，进一步理解 Z 逆变换在信号处理中的应用。通过对具体的 Z 域表达式进行逆变换，掌握 Z 逆变换的理论基础和计算方法，同时通过实际操作加深对离散时间系统分析的理解。

实验原理

Z 逆变换是将复频域中的信号转换回离散时间域的一种方法。给定一个 Z 域表达式，通过 Z 逆变换，可以得到对应的时域序列。在本实验中，考虑一个特定的 Z 域函数 $X(z) = \frac{z^{-1}-a^{-1}}{1-a^{-1}z^{-1}}$ ，我们使用 MATLAB 中的符号工具箱计算其 Z 逆变换。首先，将表达式转换为适合进行 Z 逆变换的形式，然后应用 `iztrans` 函数得到时域序列。通过这个过程，可以理解如何从 Z 域表示恢复原始序列，以及 Z 逆变换在分析和设计离散时间系统中的作用。

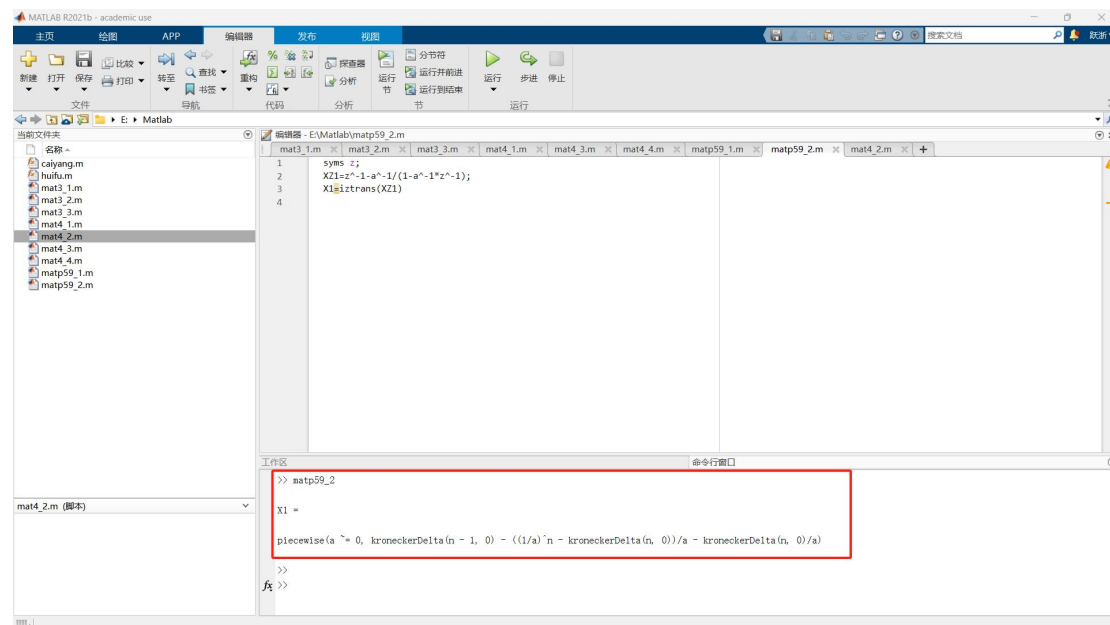
实验代码

```

syms z;
XZ1=z^-1-a^-1/(1-a^-1*z^-1);
X1=iztrans(XZ1)

```

实验结果



4-3

实验目的

本实验旨在分析由差分方程 $y(n) = 0.8y(n-2) + x(n) - x(n-2)$ 描述的因果线性时不变系统(LTI)的特性。通过求解系统函数 $H(z)$ 、绘制零极点分布图、计算并绘制单位冲激响应 $h(n)$ 、以及分析系统的频率响应 $H(e^{j\omega})$ 并绘制其幅度和相位响应曲线, 深入理解 LTI 系统在 Z 域和频域中的行为及其在信号处理中的应用。

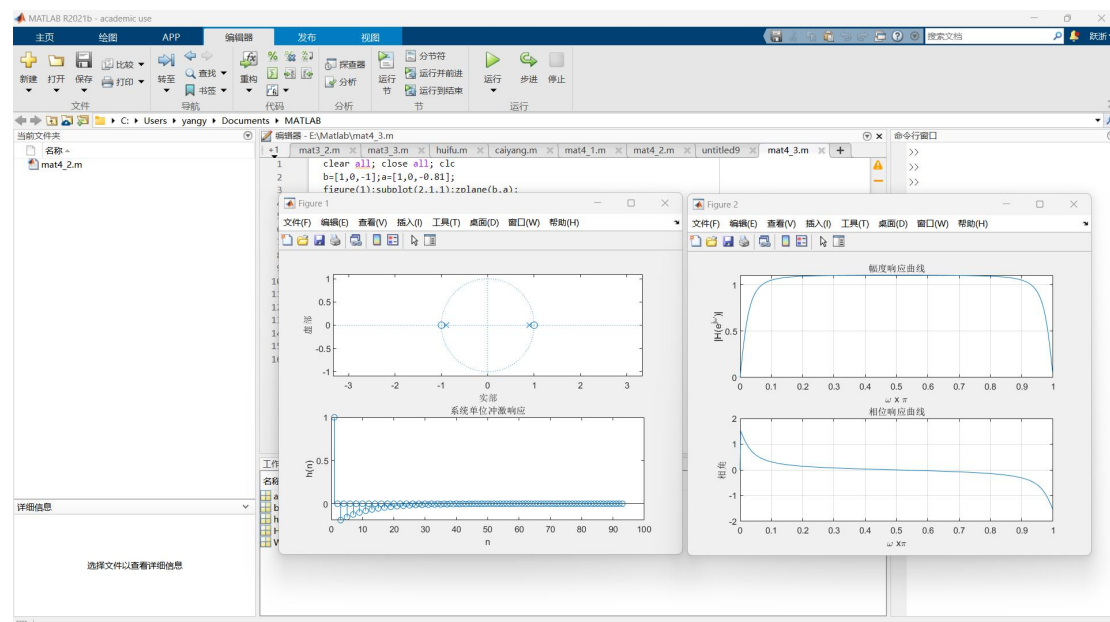
实验原理

LTI 系统由给定的差分方程描述, 其 Z 变换可以用来求解系统函数 $H(z)$ 。系统函数 $H(z)$ 是输入信号 $X(z)$ 和输出信号 $Y(z)$ 在 Z 域中的比值, 通过求解 $H(z)$, 可以得到系统的零极点分布图, 帮助分析系统的稳定性和频率特性。系统的单位冲激响应 $h(n)$ 是系统对单位冲激输入的响应, 通过计算其逆 Z 变换可以获得。频率响应 $H(e^{j\omega})$ 描述了系统对不同频率输入的响应, 可以通过对 $H(z)$ 进行变换得到, 并在频域上绘制其幅度和相位响应曲线, 帮助理解系统对不同频率分量的放大或衰减作用。通过 MATLAB 实现上述步骤, 可以直观地观察和分析 LTI 系统的各项特性, 从而深入理解其在信号处理中的应用。

实验代码

```
clear all; close all; clc;
b=[1,0,-1];a=[1,0,-0.81];
figure(1);subplot(2,1,1);zplane(b,a);
h=impz(b,a);
subplot(2,1,2);stem(h);
title('系统单位冲激响应');
xlabel('n'); ylabel('h(n)');
[H, W]=freqz(b,a);
figure(2);subplot(2,1,1);
plot(W/pi,abs(H));
title('幅度响应曲线');grid on;
xlabel('\omega x \pi'); ylabel('|H(e^{j\omega})|');
subplot(2,1,2);
plot(W/pi,angle(H));
title('相位响应曲线');
xlabel('\omega x \pi');ylabel('相角');grid on;
```

实验结果



4-4

实验目的

本实验旨在通过对离散时间信号 $x(n)$ 的 Z 变换进行分析, 绘制系统的零极点图, 并判断系统函数的收敛域及其稳定性。通过分析 Z 域中的零极点分布, 可以更深入地理解因果序列在复平面中的行为和系统稳定性条件。

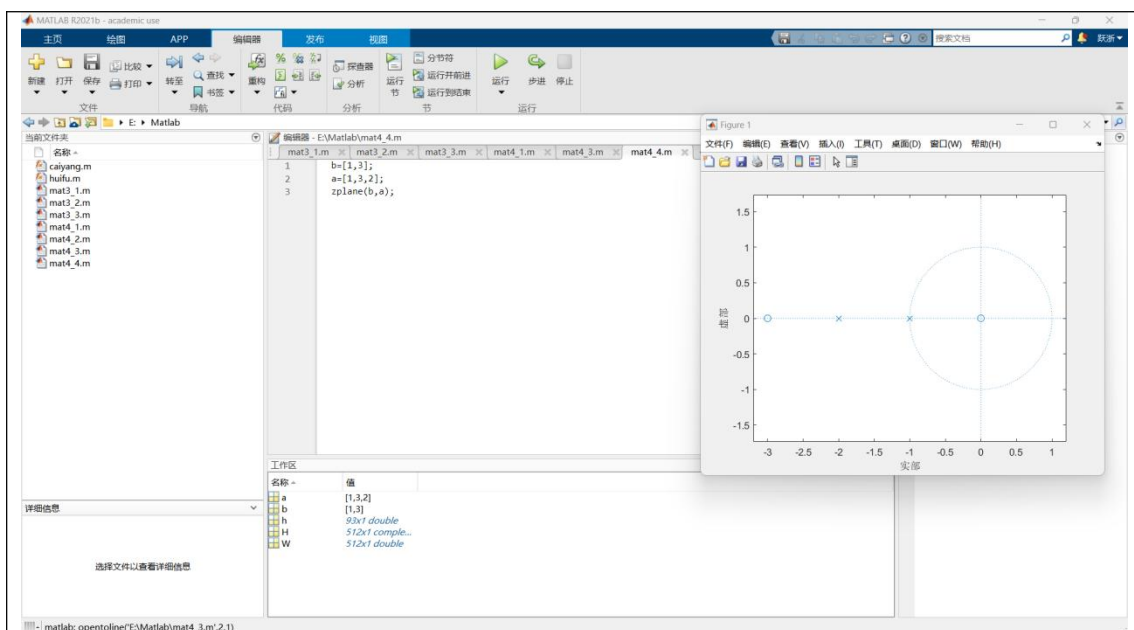
实验原理

Z 变换是将离散时间信号从时域转换到 Z 域的重要工具。给定信号 $x(n)$ 的 Z 变换表达式 $X(z) = \frac{1+3z^{-1}}{1+3z^{-1}+2z^{-2}}$, 首先, 我们可以将其转换为多项式形式, 分别表示分子和分母的系数向量。通过 MATLAB 的 `zplane` 函数, 可以绘制该系统的零极点图, 以直观地展示零点和极点在复平面上的位置。根据因果序列的定义和系统函数的零极点分布, 可以判断系统的收敛域。对于因果系统, 收敛域是所有极点之外的区域。系统的稳定性由其极点的位置决定, 当且仅当所有极点都在单位圆内时, 系统是稳定的。通过这些分析步骤, 可以全面理解该系统的特性及其在信号处理中的应用。

实验代码

```
b=[1,3];  
a=[1,3,2];  
zplane(b,a);
```

实验结果



根据极点分布情况,它的收敛域应该分为 3 个部分: $|z|<1$ 、 $1<|z|<2$ 和 $|z|>2$ 。当 $x(n)$ 为因果序列时,它的收敛域需要包含 $z=\infty$,所以只有 $|z|>2$ 才是它的收敛域,因此收敛域不包含单位圆 $|z|=1$,故该系统不稳定。

实验总结

在本次实验中,我探讨了不同抽样频率对信号重构质量的影响,通过对时域波形和频谱特性的分析,进一步理解了抽样定理在实际应用中的效果以及不同抽样频率对信号处理的影响。实验分为几个部分,包括信号的抽样和重构、时域波形和频谱特性的绘制和分析,以及通过 MATLAB 编程实现各个环节。

首先,我通过不同的抽样频率对信号进行了抽样,并使用理想低通滤波器对抽样点进行了重构。通过计算和比较不同抽样频率下重构信号与原始信号之间的最大恢复误差,我观察到,随着抽样频率的增加,重构信号与原始信号的最大误差逐渐减小。这验证了抽样定理中关于抽样频率必须至少为信号最高频率两倍的要求,表明在高于奈奎斯特率的条件下,信号能够被无失真地重构。

接着,我绘制了信号的时域波形和幅频特性曲线,通过傅里叶变换得到了信号在频域上的分布情况。在对信号进行不同抽样频率下的抽样后,我绘制了抽样序列的波形和频谱特性曲线。通过比较不同抽样频率下的频谱变化,观察到了频谱折叠现象,并通过理想低通滤波器对抽样序列进行了重构,分析了不同抽样频率下重构信号与原始信号在时域上的差异。这些分析帮助我理解了抽样频率对信号重构质量的影响,进一步验证了抽样定理在实际信号处理中的应用。

在实验的最后部分,我通过求解不同数学函数序列的 Z 变换和 Z 逆变换,加

深了对离散时间信号分析的理解。通过具体计算几种典型函数序列的 Z 变换和 Z 逆变换，掌握了 Z 变换和逆变换的理论基础和计算方法，并通过实际操作深化了对离散时间系统分析的理解。

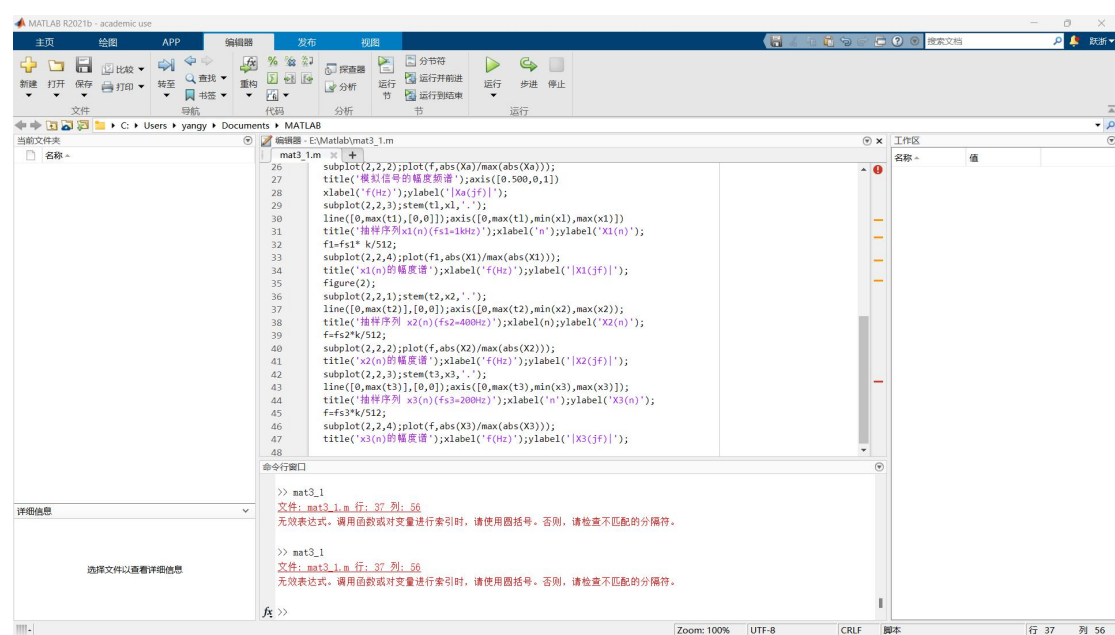
整体而言，本次实验通过对信号抽样、重构及频谱分析等多个环节的研究，系统地探讨了抽样频率对信号重构质量的影响。实验结果表明，正确的抽样频率对信号处理的准确性至关重要，验证了抽样定理的实际应用效果。

通过本次实验，我不仅掌握了抽样和重构的基本方法，还加深了对 Z 变换和逆变换在离散时间信号分析中的应用理解，为进一步研究和应用信号处理技术奠定了坚实的基础。

代码调试过程

遇到的问题：在代码编写过程中，我遇到了很多有关括号匹配，变量匹配的问题，导致代码无法正确运行。

解决方法：逐个检查报错，仔细修改，最终代码成功运行。



The screenshot shows the MATLAB R2021b - academic use interface. The code editor displays a script named 'mat3_1.m' with the following code:

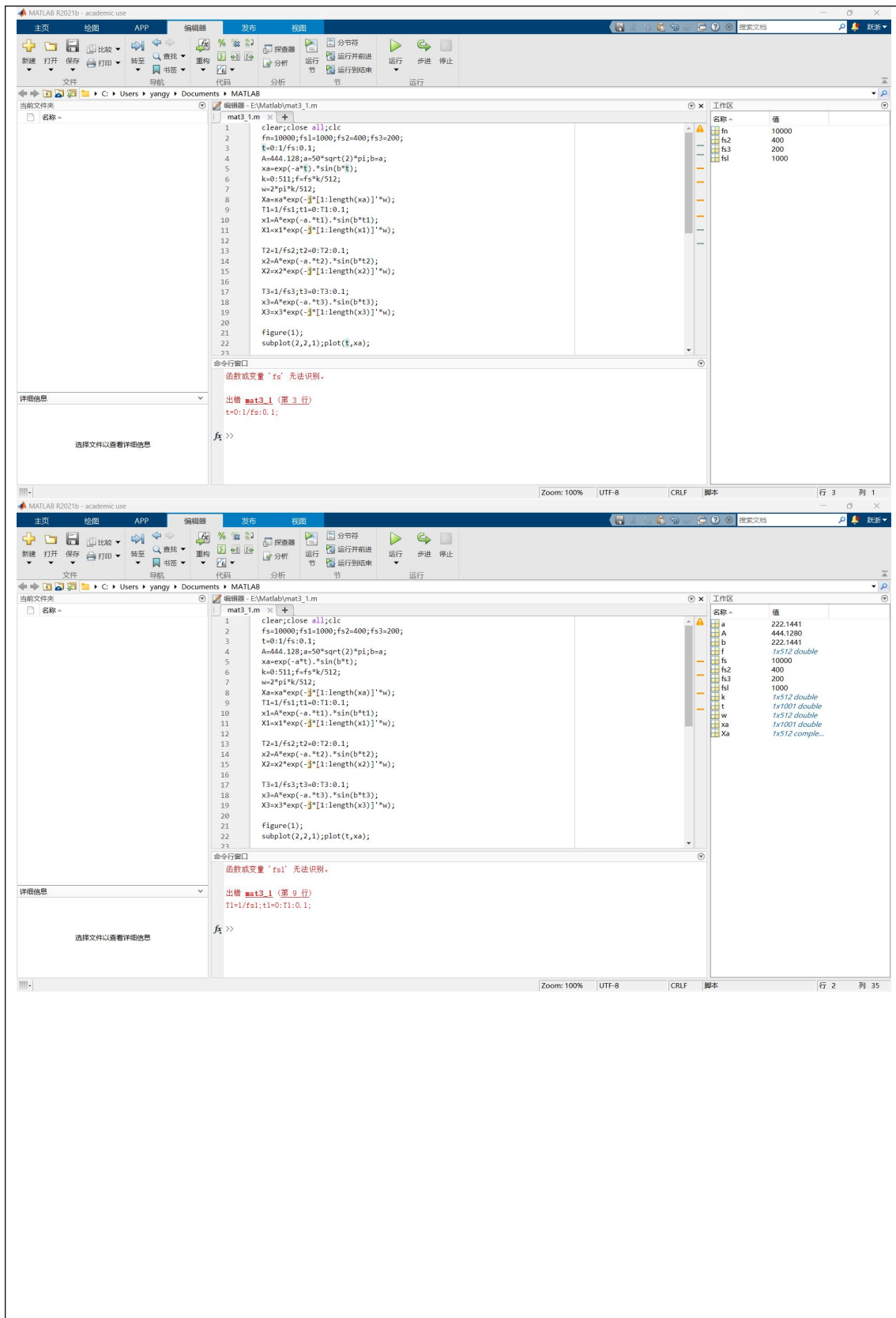
```
26 subplot(2,2,2);plot(f,abs(Xa)/max(abs(Xa)));
27 title('模拟信号的幅度频谱');axis([0.500,0,1])
28 xlabel('f(Hz)');ylabel('|Xa(f)|');
29 subplot(2,2,3);stem(t1,x1,'-');
30 line([0,max(t1),[0,0]]);axis([0,max(t1),min(x1),max(x1)]);
31 title('抽样序列 x1(n)(fs1=180Hz)');xlabel('n');ylabel('X1(n)');
32 f1=fs1*k/512;
33 subplot(2,2,4);plot(f1,abs(X1)/max(abs(X1)));
34 title('x1(n)的幅度谱');xlabel('f(Hz)');ylabel('|X1(f)|');
35 figure(2);
36 subplot(2,2,1);stem(t2,x2,'-');
37 line([0,max(t2)], [0,0]);axis([0,max(t2),min(x2),max(x2)]);
38 title('抽样序列 x2(n)(fs2=400Hz)');xlabel('n');ylabel('X2(n)');
39 f2=fs2*k/512;
40 subplot(2,2,2);plot(f,abs(X2)/max(abs(X2)));
41 title('x2(n)的幅度谱');xlabel('f(Hz)');ylabel('|X2(f)|');
42 subplot(2,2,3);stem(t3,x3,'-');
43 line([0,max(t3)], [0,0]);axis([0,max(t3),min(x3),max(x3)]);
44 title('抽样序列 x3(n)(fs3=200Hz)');xlabel('n');ylabel('X3(n)');
45 f3=fs3*k/512;
46 subplot(2,2,4);plot(f,abs(X3)/max(abs(X3)));
47 title('x3(n)的幅度谱');xlabel('f(Hz)');ylabel('|X3(f)|');
```

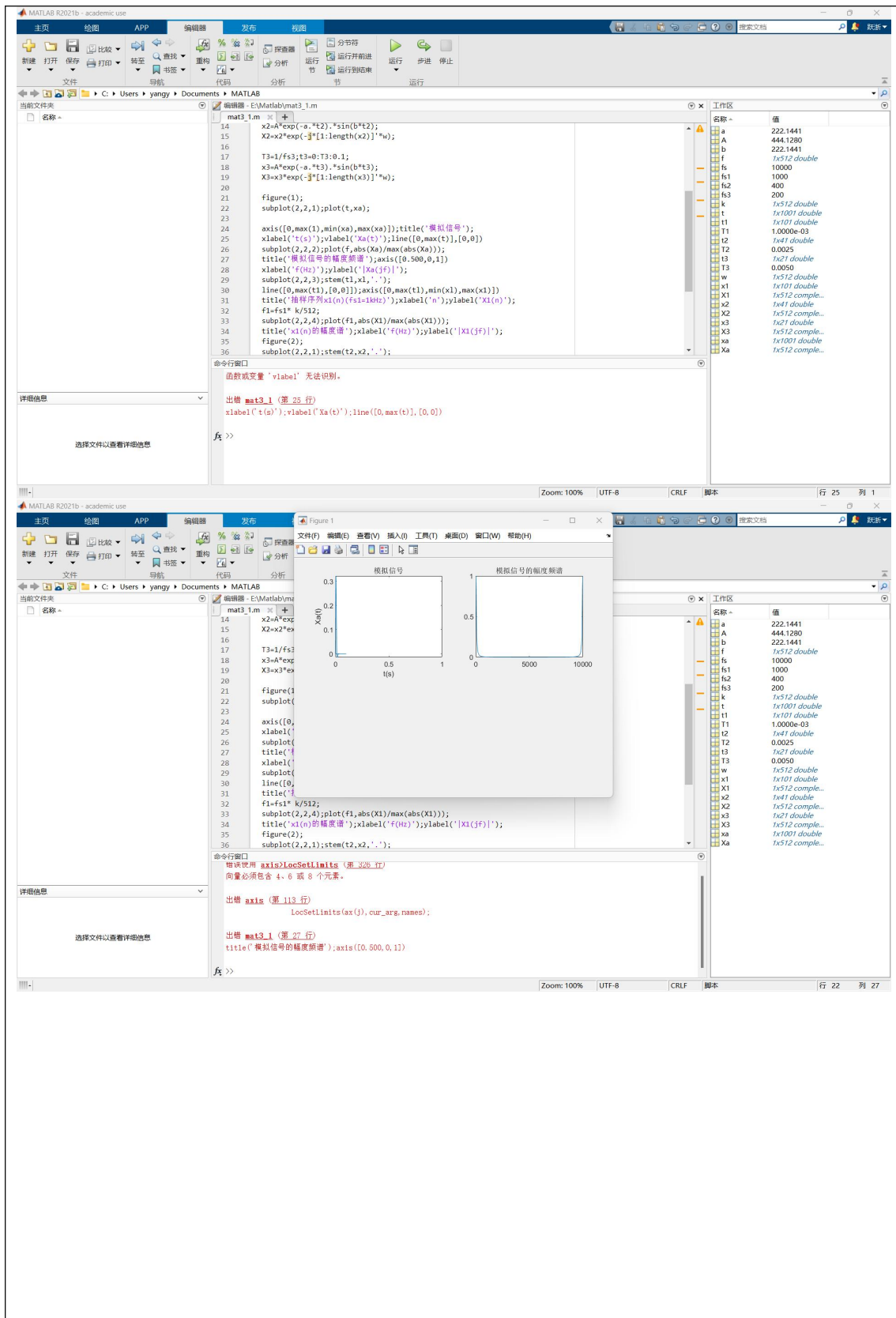
The command window shows the following error messages:

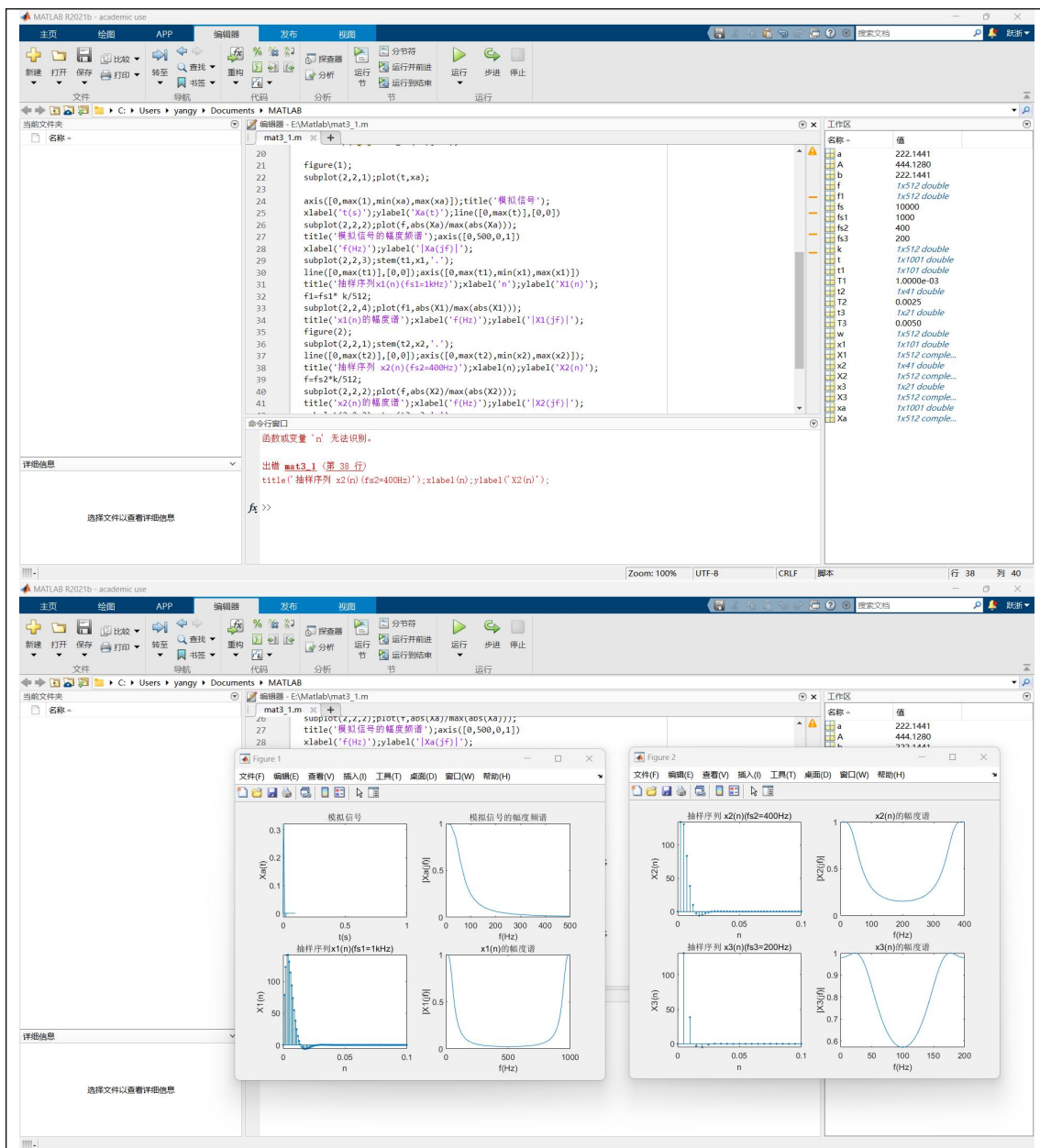
```
>> mat3_1
文件: mat3_1.m 行: 37 列: 56
无效表达式。调用函数或变量进行索引时，请使用圆括号。否则，请检查不匹配的分隔符。

>> mat3_1
文件: mat3_1.m 行: 37 列: 56
无效表达式。调用函数或变量进行索引时，请使用圆括号。否则，请检查不匹配的分隔符。

在 >>
```







遇到的问题：在代码编写过程中，我发现有些语句之后有分号，而有些没有，多分号和少分号，都会导致代码运行结果不符合期望。

解决方法：修改代码，查询资料了解分号的作用。

在 MATLAB 中，语句末尾是否加分号 (;) 有很大的区别。如果在语句末尾加上分号，MATLAB 会执行该语句但不会在命令窗口中显示输出，这样可以提高代码运行速度，尤其是在处理大数据时。相反，如果在语句末尾不加分号，MATLAB 会执行该语句并在命令窗口中显示输出，这对于调试代码或需要检查中间结果时很有帮助。因此，加分号用于提高效率，而不加分号则便于调试。

