**集美大学计算机工程学院实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称：** 数字信号与图像处理 | **班级： 计算2114** | **实验成绩：** |
| **实验编号：001** | **姓名：庄佳强** | **实验日期：11.5** |
| **实验名称：实验一：**信号处理基础 | **学号：202121331104** | **实验地点：** |

1. **目的**（本次实验所涉及并要求掌握的知识点）
2. 掌握信号处理的基本思想。
3. 理解采样信号的频谱特性,加强信号采样与重建的有关基本概念的理解。
4. 深入理解线性时不变系统输出与输入的关系。
5. 了解数字信号采样率转换前后信号频谱的特征。
6. **实验内容与设计思想**（设计思路、主要代码结构、主要代码段的文字分析、输出结果截图）
7. 给定序列，绘制其图像并分析其混叠

①

**思路：**

首先，原函数是一个混合了余弦和正弦函数的信号，频率分别为0.125π和0.25π，相位分别为0.2π和0.1π。

**实现：**

1.构造函数

f=3\*cos(0.125\*pi\*n+0.2\*pi)+2\*sin(0.25\*pi\*n+0.1\*pi);

2，绘图

plot(f);

title('原函数');

xlabel('

时间序列n ');

ylabel('x(n)');

②将①中的扩展为以16为周期的函数，绘制出四个周期

**思路：**

周期为16时，取n的范围为0到70，然后对n进行取模运算，得到的结果就是周期为16的序列。然后，将这个序列代入原函数中，得到周期为16的信号。

**实现：**

%%周期为16

n2=0:70;

通过mod函数控制x(n)的周期

n=mod(n2,15);

构造函数

f=3\*cos(0.125\*pi\*n+0.2\*pi)+2\*sin(0.25\*pi\*n+0.1\*pi);

再绘图

plot(f);

title('周期为16');

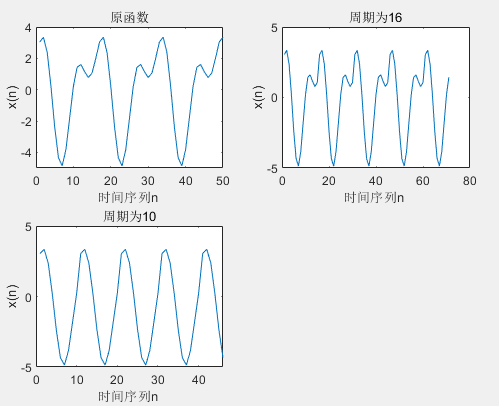
xlabel('时间序列n ');

ylabel('x(n)');

③将①中的扩展为以10为周期的函数，绘制出四个周期

周期为10时，同样的方法，取n的范围为0到45，然后对n进行取模运算，得到的结果就是周期为10的序列。然后，将这个序列代入原函数中，得到周期为10的信号。

与上方同理



1. 设计序列的移位和累加函数

**实现：**

先用一个函数计算x(n-k)部分，

用于计算（n-k）部分

outputArg2 = m+k;

用于计算x部分

outputArg1 = x;

再用一个函数实现y(n)=x1\*(n)+x2\*(n),

最后依次进行计算，绘图。

计算(x(n+2))

[x11, n11] = sigshift(x, n, 2);

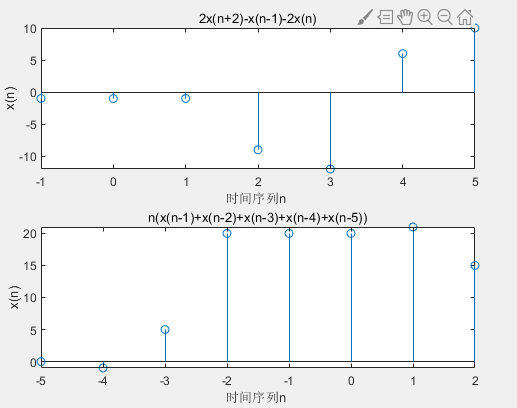
计算2\*x(n+2)-x(n-1)

[x1, n] = sigadd(2\*x11, n11, (-1)\*x12, n12);

绘图

stem(n, x1);

之后同理



1. 绘制因果系统的幅频响应和相频响应

**思路：**

首先，给定了一个因果系统的传递函数系数b和a，其中b表示系统的分子多项式系数，a表示系统的分母多项式系数。  
接下来，使用freqz函数计算系统的频率响应。freqz函数返回两个输出参数，第一个参数H是频率响应的复数值，第二个参数x是对应的角频率。

然后，分别计算频率响应的幅度值和相位值，使用abs函数取幅度值的实部，使用angle函数取相位值对应的相位角。  
接下来，使用plotyy函数将幅度和相位分别画在两个y轴上，同时将幅度的单位转换为分贝单位。  
最后，设置图的标题和y轴标签，并设置幅度和相位的线型为虚线和点线。

**实现：**

b = [2 0]; a = [1 -0.67 0.9];

[H,x] = freqz(b,a);

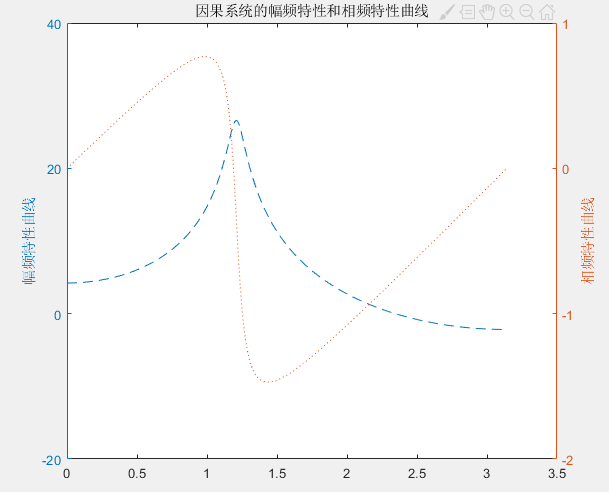
之后去幅度值实部和对应相位角

Hf=abs(H); % 取幅度值实部

Hx=angle(H); % 取相位值对应相位角

然后把幅值转化成分贝再绘图

[AX, H1, H2] = plotyy(x, 20\*log10(Hf), x, Hx); % 幅值变换为分贝单位



1. 验证DTFT的折叠特性

**思路：**

首先，给定了一个离散时间序列x，其中包含了实部和虚部。  
接下来，将x进行傅里叶变换，得到频域上的复数序列X。这里使用了指数函数exp来实现频域上的折叠操作。  
然后，分别计算了X的幅度、相位、实部和虚部，并将它们画在不同的子图中。

接下来，将原始离散时间序列x进行折叠操作，得到折叠后的离散时间序列。同样地，将折叠后的序列进行傅里叶变换，得到频域上的复数序列X。

**实现：**

先绘未折叠的离散数学序列，x = M + i\*K;

由题，首先生成− 10 ≤ n ≤ 20共包含31个随机复数的序列：

使用函数X = x \* (exp(1j \* pi/100)).^(n' \* k);

折叠前的离散时间序列

n = -10:20; M = rand([1,31]); K = rand([1,31]); x = M + 1i\*K;

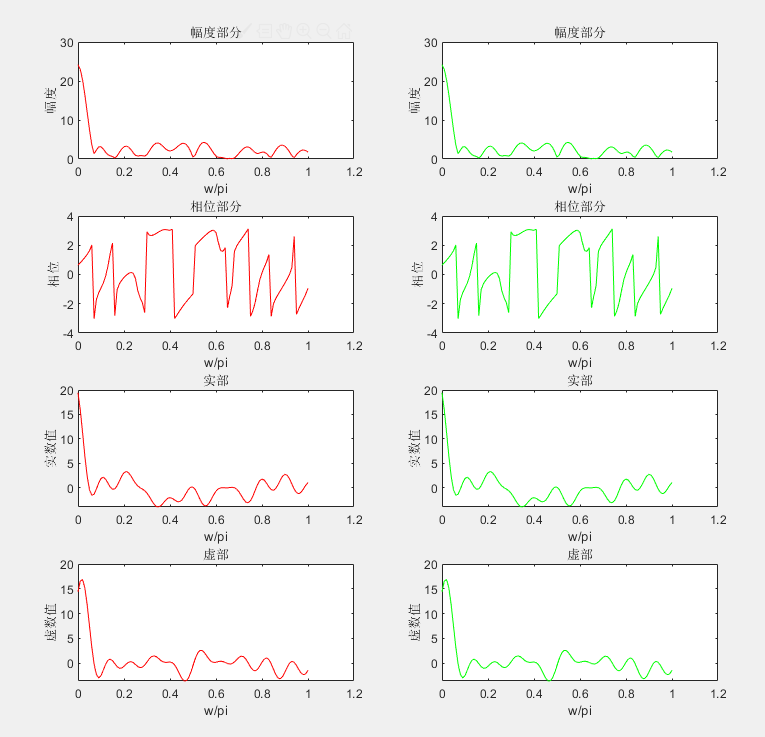
k = 0:100; w = (pi/100)\*k;

折叠后的离散时间序列

使用fliplr

n = -fliplr(n); x = fliplr(x);

k = 0:100; w = (pi/100)\*k;



1. 绘制给定系统的稳态响应

**思路：**

首先，给定了一个差分方程y(n) = 0.8y(n-1) + x(n),其中x(n) = cos(0. 05rn)u(n)是输入信号。接下来，使用zplane函数绘制了分子系数Y. \_vect的零点和分母系数X\_ .vect的极点的图形。这个图形可以帮助分析系统的稳定性和频率响应特性。

接下来，生成了输入信号Input,它是一个cos函数乘以单位阶跃函数u(n)。使用filter函数计算了输入信号的响应0utput。然后，绘制了原始输入信号x(n)的图形，可以观察到它是一个频率为0. 05π的周期信号。最后，绘制了输入信号x(n)的稳态响应图，可以观察到系统对输入信号的响应情况。通过观察稳态响应图，可以分析系统的滤波效果和输出信号的特性。

先绘制列向量Y的零点和列向量X的极点，再绘制时域脉冲响应图，最后再画出下x（n）的图。

**实现：**

先写入分子和分母的系数

Y\_vect = [1 -0.8]; X\_vect = [1];

绘制列向量Y的零点和列向量X的极点

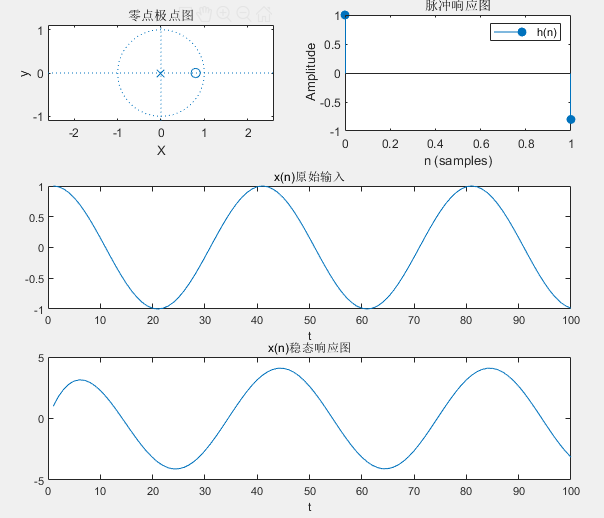
zplane(Y\_vect, X\_vect);

再 时域脉冲响应图

impz(Y\_vect, X\_vect);

Input = cos(0.05\*pi\*t); % 输入信号，频率w=0.05pi，相位为0

Output = filter(X\_vect, Y\_vect, Input); % 输入信号的响应



**三、实验使用环境**（本次实验所使用的平台和相关软件）

MatLab

1. **实验小结（**实验中遇到的问题及解决过程、实验中产生的错误及原因分析、实验体会和收获）

**问题：**

一开始根本不会使用matlab,不知道如何下手。

**解决方法：**

在网上先找了一个课程进行学习，然后再开始实验部分。

**问题：**

对应matlab中很多的函数都不会使用，如freqz函数这样的，但社区中的对应的软件版本的使用方法不一样。

**解决方法：**

通过查询matlab的官方文档，得知他在对应版本的使用方法。

**问题：**数组操作对不上。

**解决方法:**

仔细检查涉及到的矩阵和数组的维度，确保它们在相互操作时是兼容的。可以使用 size 函数来检查数组的维度。

**实验体会和收获：**

通过查找社区的资料，学习使用如何再matlab上完成图形的绘制。

在做实验的过程中，学会了使用 MATLAB工具。

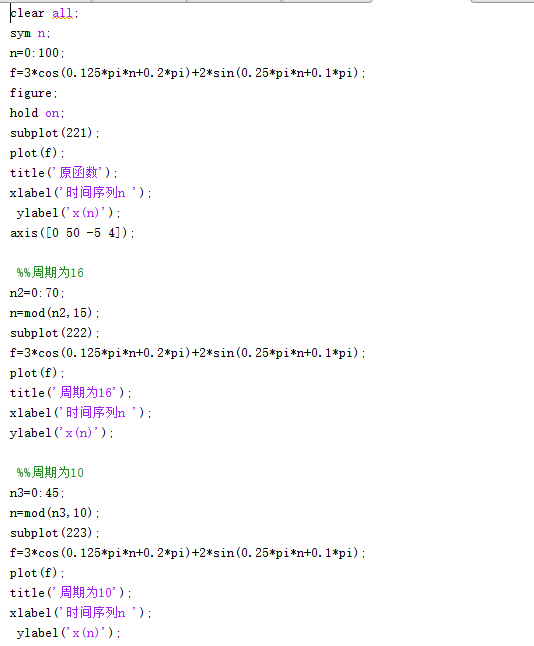
实验中涉及到信号处理和系统分析，通过验证这些，我更好的掌握了书中的知识。

1. **参考文献**（书籍、网址链接等）

无

**六、源代码附件**（完整的程序源代码，**注意排版紧凑，如果代码较长，字体、字间距选择小一点的**）

**Lab1\_1**



**Sigshift.m**

function [outputArg1,outputArg2] = sigshift(x,m,k)

outputArg2 = m+k;

outputArg1 = x;

End

**Sigadd.m**

function [y,n] = sigadd(x1,n1,x2,n2)

% 实现序列相加

% 实现y(n)=x1(n)+x2(n)

% [y,n ]=sigadd(x1 ,n1,x2,n2)

n = min(min(n1),min(n2)):max(max(n1),max(n2)); % 求出序列的长度

y1 = zeros(1,length(n));

y2=y1;

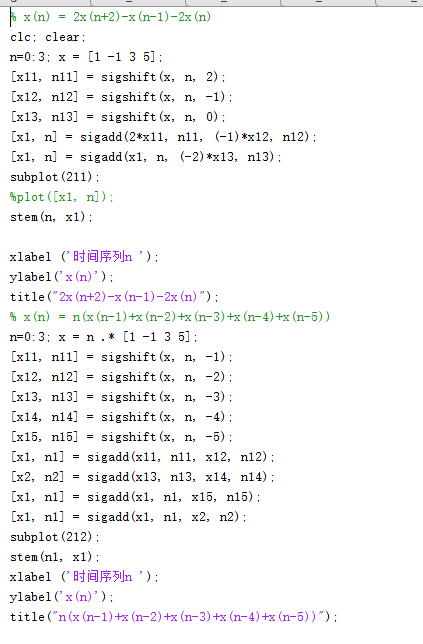
y1 ( (n>=min (n1) ) & (n<=max (n1) )==1 )=x1 ; % 将子序列的值复制到和序列的对应位置

y2 ( (n>=min (n2) ) & (n<=max (n2) )==1 )=x2 ;

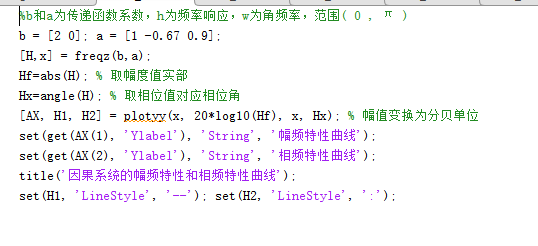
y=y1+y2;

end

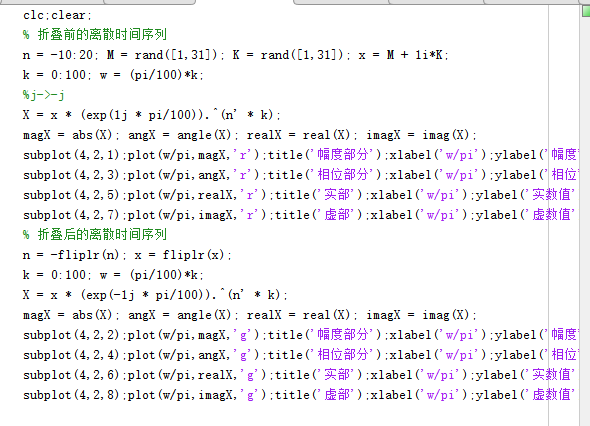
**Lab1\_2**



**Lab1\_3**



**Lab1\_4**



**Lab1\_5**

