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	实验三	实验次序	03
实验地点	笃行南楼 A301	实验日期	06.03

实验内容：

5-3

实验目的

本实验的目的是通过实际仿真与理论推导对比，验证某因果线性时不变系统（LTI）的单位阶跃响应特性。具体而言，通过差分方程描述该系统，利用卷积和递归方法求解系统的单位阶跃响应，并将仿真数据与理论推导数据进行比较，以确认仿真结果的准确性和一致性。

实验原理

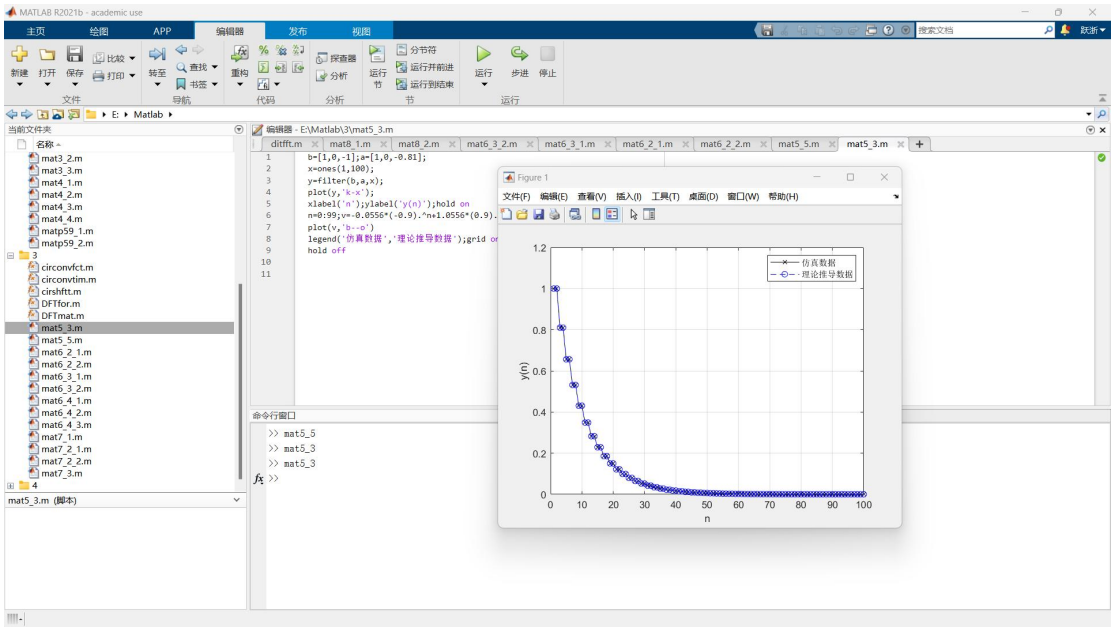
因果线性时不变系统 (LTI)可以用差分方程来描述。对于给定的差分方程 $y(n) = 0.81y(n-2) + x(n) - x(n-2)$,其系统函数$H(z)$可以通过 Z 变换求得为 $H(z) = \frac{1-z^{-2}}{1-0.81z^{-2}}$。单位阶跃响应$y(n)$的定义是当输入$x(n)$为单位阶跃函数$u(n)$时，系统的输出响应。单位阶跃函数的 Z 变换为 $U(z) = \frac{1}{1-z^{-1}}$。因此，单位阶跃响应的 Z 变换为$Y(z) = H(z) \cdot U(z)$,经过部分分式展开和逆 Z 变换，可以得到单位阶跃响应的时间域表达式。仿真部分使用 MATLAB 中的 filter 函数计算系统的实际单位阶跃响应，通过绘制仿真和理论的响应曲线，验证仿真数据与理论数据的符合程度，从而确认仿真模型的准确性。

实验代码

```
b=[1,0,-1];a=[1,0,-0.81];
x=ones(1,100);
y=filter(b,a,x);
plot(y,'k-x');
```

```
xlabel('n');ylabel('y(n)');hold on
n=0:99;v=-0.0556*(-0.9).^n+1.0556*(0.9).^n;
plot(v,'b--o')
legend('仿真数据','理论推导数据');grid on;
hold off
```

实验结果



5-5

实验目的

本实验的目的是通过 MATLAB 仿真验证二阶系统对随机噪声的响应特性。通过给定系统的传递函数，利用随机噪声作为输入信号，分析系统的输出响应曲线，以理解系统在噪声输入下的动态行为。

实验原理

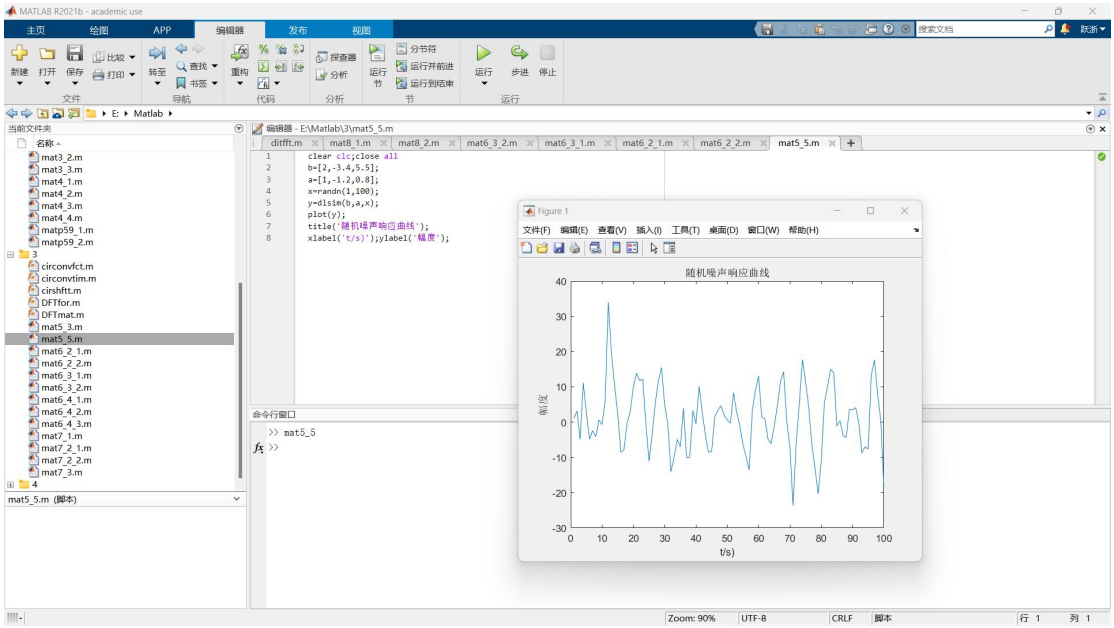
二阶系统的传递函数 $H(z)=\frac{2z^2-3.4z+5.5}{z^2-1.2z+0.8}$ 可以描述系统的频域特性。为了研究该系统在随机噪声输入下的响应，我们使用 MATLAB 进行仿真。具体过程包括定义系统的传递函数系数 a 和 b，生成随机噪声信号 x，然后使用 MATLAB 中的 dlsim

函数求解系统的输出响应 y。通过绘制输出响应曲线， 可以直观地观察系统对随机噪声的动态响应特性。通过这种方法，可以分析系统的稳定性和滤波特性，从而理解系统在实际应用中的表现。

实验代码

```
clear clc;close all
b=[2,-3.4,5.5];
a=[1,-1.2,0.8];
x=randn(1,100);
y=dlsim(b,a,x);
plot(y);
title('随机噪声响应曲线');
xlabel('t/s)');ylabel('幅度');
```

实验结果



6-2

实验目的

本实验的目的是计算序列$x(n) = (-0.9)^n$的离散时间傅里叶变换 (DTFT)和 11 点离散傅里叶变换(DFT),并绘制它们的幅度和相位响应曲线，通过比较两者，理

解 DTFT 和 DFT 在频域分析中的应用及差异。

实验原理

离散时间傅里叶变换（DTFT）用于分析离散时间信号的频域特性，它将时间序列转换为频域函数。对于序列$x(n) = (-0.9)^n$，其 DTFT 可以通过公式$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$计算。通过在 MATLAB 中使用指数函数和矩阵运算，可以数值计算并绘制出 DTFT 的幅度和相位响应。离散傅里叶变换（DFT）是傅里叶变换在离散域和有限长序列上的实现，通常用于数字信号处理。对于长度为 11 的序列$x(n)$，其 DFT 可以通过公式$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$计算，N 为序列长度。在 MATLAB 中，可以使用自定义的 DFT 矩阵或内置函数来计算 DFT，并绘制其频域特性。通过计算 DTFT 和 DFT，并绘制它们的幅度和相位响应曲线，可以比较两者在频域分析中的表现。DTFT 提供了连续的频域表示，而 DFT 则提供了离散的频域表示，通过对比可以理解两者在频域分析中的优缺点和应用场景。实验中使用了两个 MATLAB 程序，第一个程序用于计算并绘制 DTFT，第二个程序用于计算并绘制 DFT 及其与 DTFT 的比较。通过这些仿真，直观地展示了 DTFT 和 DFT 的频域特性。

实验代码

代码 1：

```
n=-5:5;x=(-0.9).^n;
k=-200:200;w=(pi/100)*k;
X=x*(exp(-j*pi/100)).^(n'*k);
magX=abs(X);angX=angle(X);
subplot(2,1,1);plot(w/pi,magX);grid on;
axis([-2,2,0,15]);xlabel('\omega(x\pi)');ylabel('幅度|H(e^j^\omega)|');
subplot(2,1,2);plot(w/pi,angX/pi);grid on;
axis([-2,2,-1,1]); xlabel('\omega(x\pi)');ylabel('相位（弧度/\pi)');
```

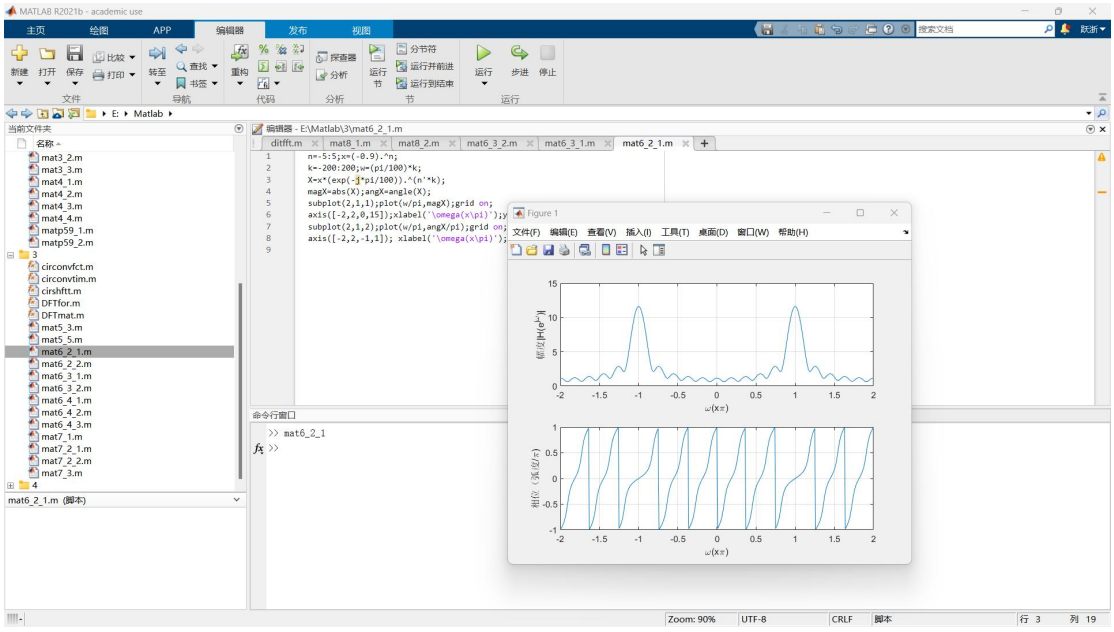
函数 DFTmat()

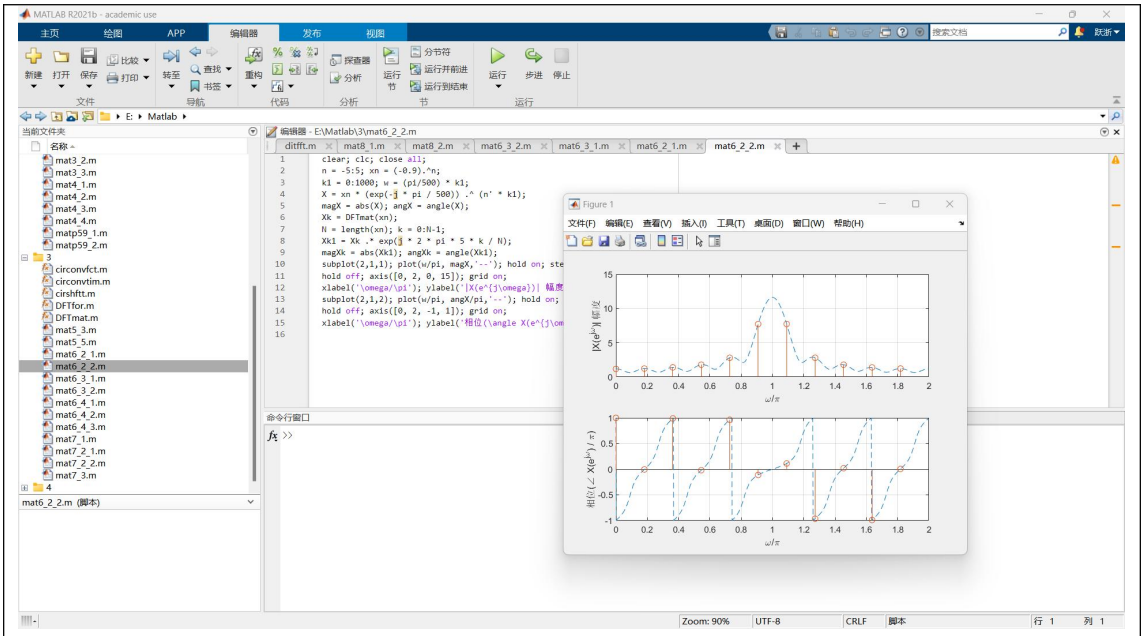
```
function Xk = DFTmat(xn)
N = length(xn);
n = 0:N-1; k = n';
WN = exp(-j * 2 * pi / N);
Wnk = WN .^ (k * n);
Xk = xn * Wnk;
```

代码 2:

```
clear; clc; close all;
n = -5:5; xn = (-0.9).^n;
k1 = 0:1000; w = (pi/500) * k1;
X = xn * (exp(-j * pi / 500)) .^ (n' * k1);
magX = abs(X); angX = angle(X);
Xk = DFTmat(xn);
N = length(xn); k = 0:N-1;
Xk1 = Xk .* exp(j * 2 * pi * 5 * k / N);
magXk = abs(Xk1); angXk = angle(Xk1);
subplot(2,1,1); plot(w/pi, magX,'--'); hold on; stem(2 * k / N, magXk);
hold off; axis([0, 2, 0, 15]); grid on;
xlabel('\omega/\pi'); ylabel('|X(e^{j\omega})|  幅度');
subplot(2,1,2); plot(w/pi, angX/pi,'--'); hold on; stem(2 * k / N, angXk/pi);
hold off; axis([0, 2, -1, 1]); grid on;
xlabel('\omega/\pi'); ylabel('相位(\angle X(e^{j\omega}) / \pi)');
```

实验结果





6-3

实验目的

本实验的目的是通过计算序列 x(n)的离散时间傅里叶变换（DTFT）和 4 点离散傅里叶变换（DFT），并绘制它们的幅度和相位响应曲线，理解 DTFT 和 DFT 在频域分析中的应用及差异。

实验原理

离散时间傅里叶变换（DTFT）用于分析离散时间信号的频域特性，它将时间序列转换为频域函数。对于给定的序列$x(n) = \{1,1,1,1\}$，其 DTFT 可以通过公式$X(e^{j\omega}) = \sum_{n=0}^{3} x(n)e^{-j\omega n}$计算。通过在 MATLAB 中使用指数函数和矩阵运算，可以数值计算并绘制出 DTFT 的幅度和相位响应。 离散傅里叶变换（DFT）是傅里叶变换在离散域和有限长序列上的实现，通常用于数字信号处理。对于长度为 4 的序列$x(n)$，其 DFT 可以通过公式$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$计算，N 为序列长度。在 MATLAB 中，可以使用自定义的 DFT 矩阵或内置函数来计算 DFT，并绘制其频域特性。 通过计算 DTFT 和 DFT，并绘制它们的幅度和相位响应曲线，可以比较两者在频域分析中的表现。 DTFT 提供了连续的频域表示，而 DFT 则提供了离散的频域表示，通过对比可以理解两者在频域分析中 的优缺点和应用场景。

实验中使用了两个 MATLAB 程序，第一个程序用于计算并绘制 DTFT，第二个程序用于计算并绘制 DFT 及其与 DTFT 的比较。通过这些仿真，直观地展示了 DTFT 和 DFT 的频域特性。

实验代码

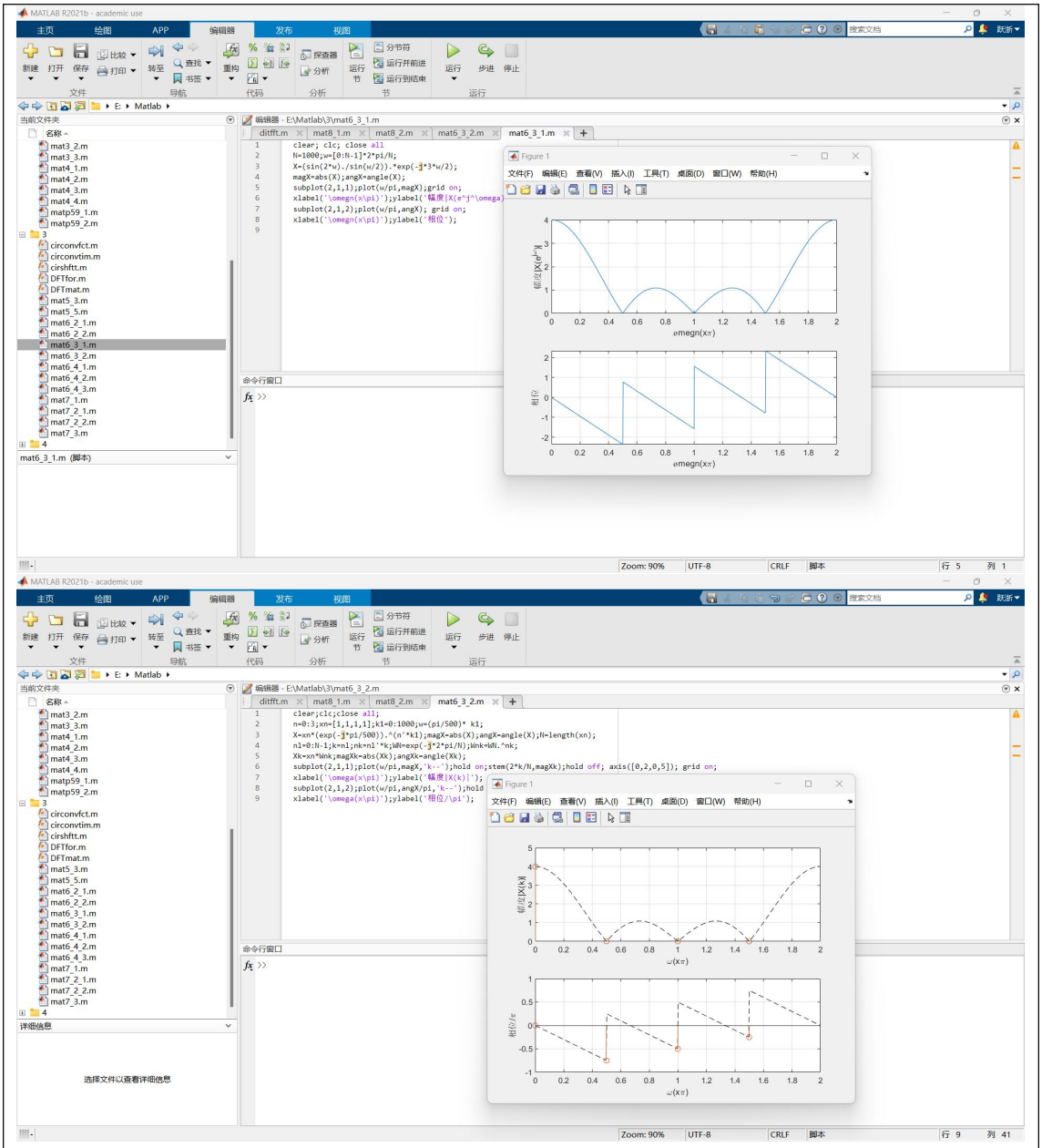
代码 1：

```
clear; clc; close all
N=1000;w=[0:N-1]*2*pi/N;
X=(sin(2*w)./sin(w/2)).*exp(-j*3*w/2);
magX=abs(X);angX=angle(X);
subplot(2,1,1);plot(w/pi,magX);grid on;
xlabel('\omegn(x\pi)');ylabel('幅度|X(e^j^\omega)|');
subplot(2,1,2);plot(w/pi,angX); grid on;
xlabel('\omegn(x\pi)');ylabel('相位');
```

代码 2：

```
clear;clc;close all;
n=0:3;xn=[1,1,1,1];k1=0:1000;w=(pi/500)* k1;
X=xn*(exp(-j*pi/500)).^(n'*k1);magX=abs(X);angX=angle(X);N=length(xn);
nl=0:N-1;k=nl;nk=nl'*k;WN=exp(-j*2*pi/N);Wnk=WN.^nk;
Xk=xn*Wnk;magXk=abs(Xk);angXk=angle(Xk);
subplot(2,1,1);plot(w/pi,magX,'k--');hold on;stem(2*k/N,magXk);hold off; axis([0,2,0,5]); grid on;
xlabel('\omega(x\pi)');ylabel('幅度|X(k)|');
subplot(2,1,2);plot(w/pi,angX/pi,'k--');hold on;stem(2*k/N,angXk/pi);hold off;axis([0,2,-1,1]); grid on;
xlabel('\omega(x\pi)');ylabel('相位/\pi');
```

实验结果



6-4

实验目的

本实验的目的是分析序列 x(n)=cos(0.48πn)+cos(0.52πn)在不同情况下的频谱特性。通过计算并比较其截取部分、补零延长以及原始序列的离散傅里叶变换（DFT），理解 DFT 在处理有限长序列时的频谱泄露和分辨率特性。

实验原理

离散傅里叶变换（DFT）是离散信号频域分析的重要工具。对于一个长度为 N

的序列$x(n)$,其中 DFT 由以下公式定义：

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

实验中，考虑三个不同的序列：

1. 序列 $x_1(n) = x(n)$的前 10 个点。
2. 序列 $x_2(n)$,将 $x(n)$的前 10 个点补零延长到 100 个点。
3. 完整序列 $x_3(n) = x(n)$。

通过计算这三个序列的 DFT，并绘制其幅度谱，可以分析截断和零填充对频谱的影响。特别地，通 过 DFT 可以观察到频谱泄露和频谱分辨率的变化。MATLAB 代码分为三部分，每部分对应一个不同的序列处理情况。每段代码计算序列的 DFT，并绘制序列和其频谱的幅度图。通过比较这些图，可以直观地看到不同处理方法对频谱特性的影响，从而理解 DFT 在有限长序列频域分析中的局限性和优点。

实验代码

函数 DFTfor（）

```
function X=DFTfor(xn)
N = length(xn);
X = zeros(1, N);
for k = 0 : N - 1
for n = 0 : N - 1
X(k+1) = X(k+1) + xn(n+1) * exp(-j * 2 * pi * n * k / N);
end
end
```

代码 1：

```
clear; clc; close all;
n = 0:99; x = cos(0.48 * pi * n) + cos(0.52 * pi * n);
n1 = 0:9; x1 = x(1:10); N = length(x1);
X1 = DFTfor(x1);
k = n1; w = 2 * pi * k / N;
magX1 = abs(X1);
subplot(2,1,1); stem(n1, x1); ylabel('x(n)'); xlabel('n'); grid on;
```

```
subplot(2,1,2); stem(w / pi, magX1); ylabel('|X_1_0(k)|'); xlabel('omega(x\pi)'); grid on;
```

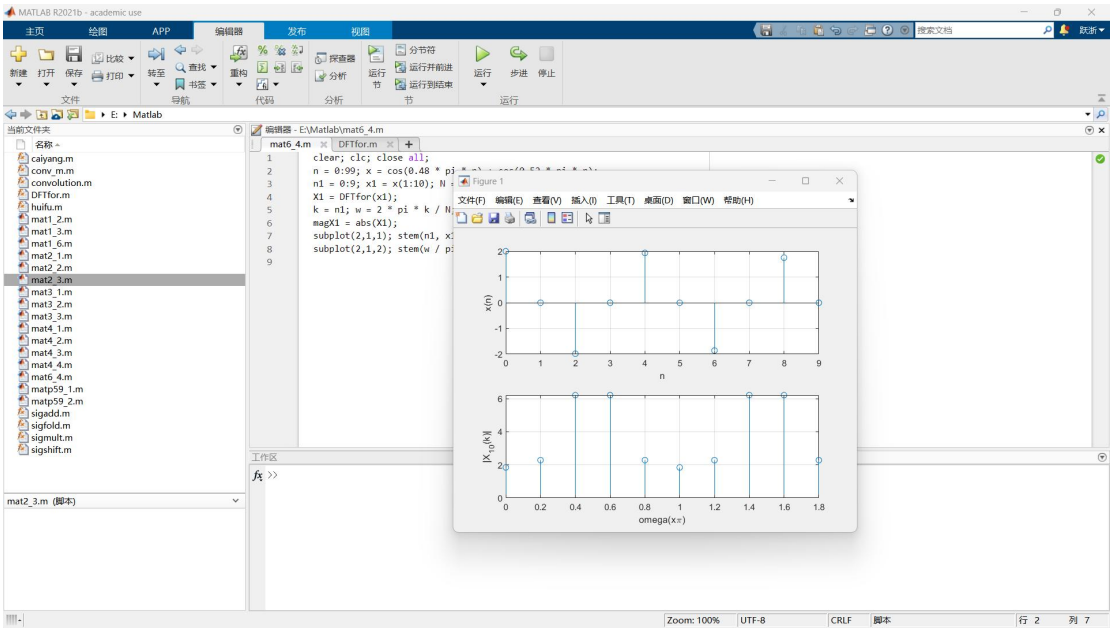
代码 2：

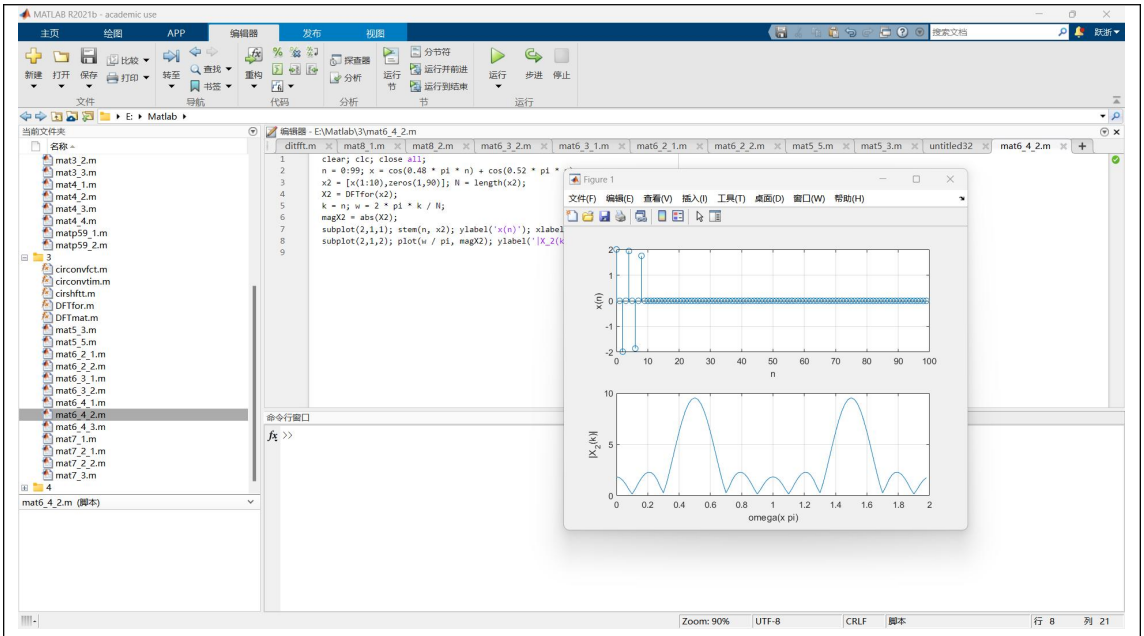
```
clear; clc; close all;
n = 0:99; x = cos(0.48 * pi * n) + cos(0.52 * pi * n);
x2 = [x(1:10),zeros(1,90)]; N = length(x2);
X2 = DFTfor(x2);
k = n; w = 2 * pi * k / N;
magX2 = abs(X2);
subplot(2,1,1); stem(n, x2); ylabel('x(n)'); xlabel('n'); grid on;
subplot(2,1,2); plot(w / pi, magX2); ylabel('|X_2(k)|'); xlabel('omega(x pi)'); grid on;
```

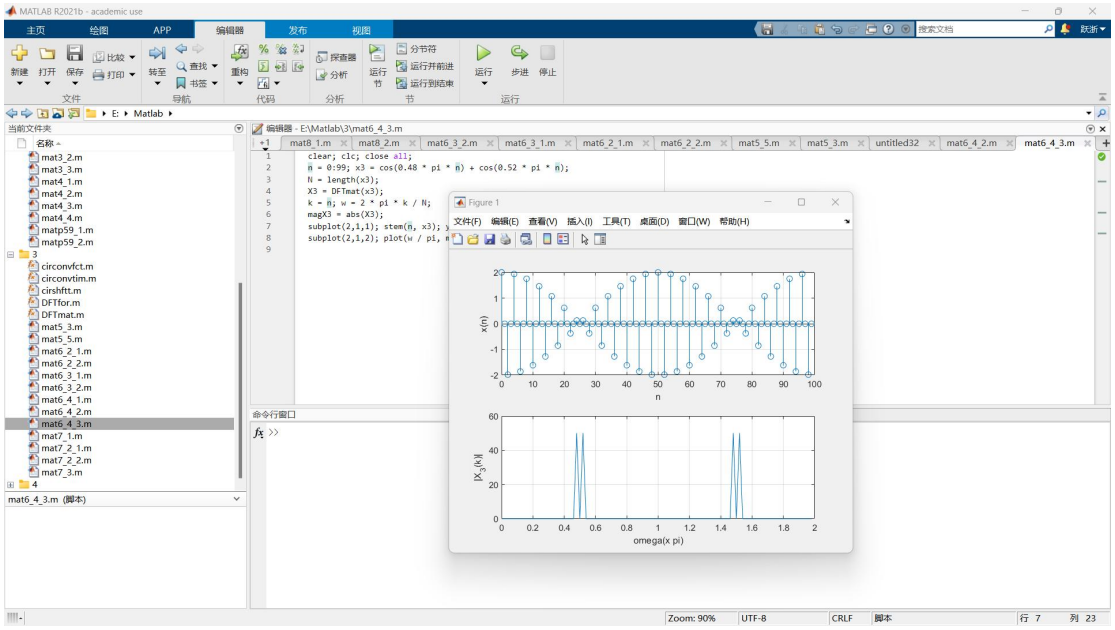
代码 3：

```
clear; clc; close all;
n = 0:99; x3 = cos(0.48 * pi * n) + cos(0.52 * pi * n);
N = length(x3);
X3 = DFTmat(x3);
k = n; w = 2 * pi * k / N;
magX3 = abs(X3);
subplot(2,1,1); stem(n, x3); ylabel('x(n)'); xlabel('n'); grid on;
subplot(2,1,2); plot(w / pi, magX3); ylabel('|X_3(k)|'); xlabel('omega(x pi)'); grid on;
```

实验结果







7-1

实验目的

本实验的目的是通过 MATLAB 编程实现对一个给定 11 点序列 x(n) 进行多种循环移位操作，并绘制相应的样本，以理解和验证循环移位在信号处理中的应用。

实验原理

循环移位是信号处理中一个重要的操作，通过对信号进行循环移位，可以方便地实现信号的时移和对齐操作。给定一个序列$x(n) = 10 \cdot 0.8^n$,其中$0 \le n \le 10$，在进行循环移位时，超出序列长度的部分会回到序列的开头，实现周期性延拓。具体操作包括： 1.循环移位模运算：通过 MATLAB 中的‘mod’函数，实现序列的循环移位和周期性延拓。 2. 补零操作：为了处理循环移位，需要将序列长度扩展到 15 点，在序列后补零。 3. 循环移位函数：使用自定义的 cirshftt 函数对序列进行循环移位，得到相应的移位序列。

实验代码

函数 cirshftt()

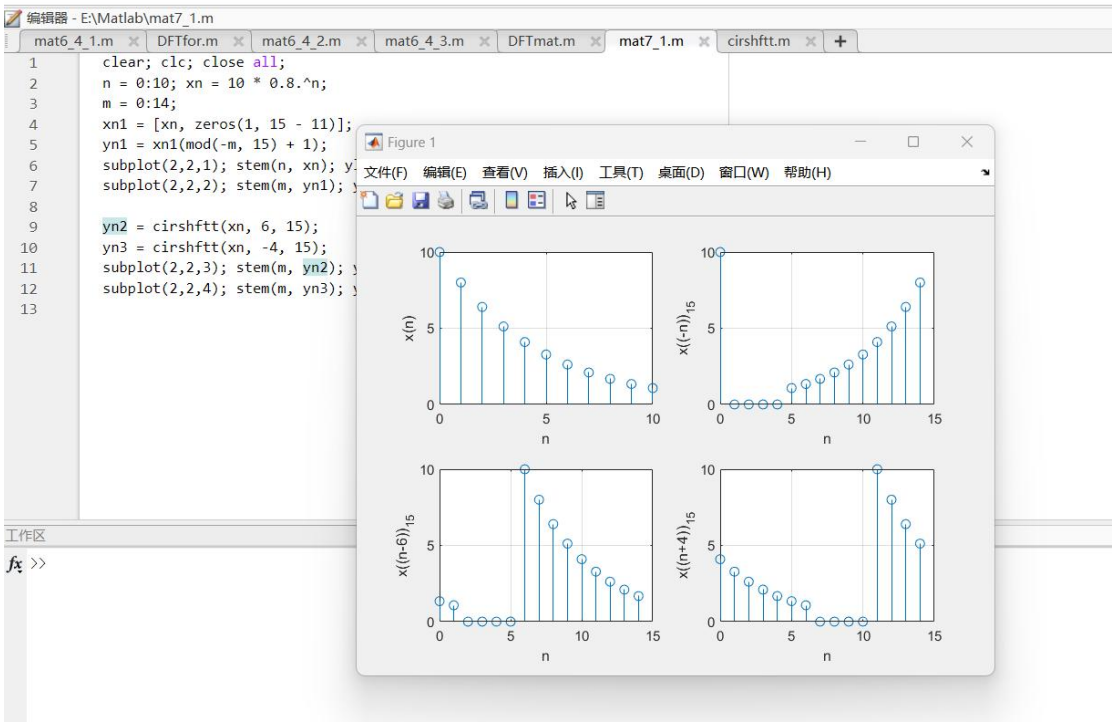
```
function y=cirshftt(x,m,N)
if length(x)>N
error
('N must be >=the length of x')
end
x=[x,zeros(1,N-length(x))];
n=0:N-1;
n=mod(n-m,N);
y=x(n+1);
```

代码

```
clear; clc; close all;
n = 0:10; xn = 10 * 0.8.^n;
m = 0:14;
xn1 = [xn, zeros(1, 15 - 11)];
yn1 = xn1(mod(-m, 15) + 1);
subplot(2,2,1); stem(n, xn); ylabel('x(n)'); xlabel('n'); grid on;
subplot(2,2,2); stem(m, yn1); ylabel('x((-n))_{15}'); xlabel('n'); grid on;

yn2 = cirshftt(xn, 6, 15);
yn3 = cirshftt(xn, -4, 15);
subplot(2,2,3); stem(m, yn2); ylabel('x((n-6))_{15}'); xlabel('n'); grid on;
subplot(2,2,4); stem(m, yn3); ylabel('x((n+4))_{15}'); xlabel('n'); grid on;
```

实验结果



7-2

实验目的

本实验的目的是通过时间域和频域卷积计算两个已知序列的循环卷积。通过 MATLAB 编程实现并验证序列 x1(n)={1,1,1}和 x2(n)={1,2,3,0,0,0,4}的循环卷积，并对比时间域和频域方法的结果。

实验原理

循环卷积是信号处理中用于分析周期性序列的重要操作。时间域循环卷积通过直接计算每个点的卷积和来实现，而频域循环卷积则利用离散傅里叶变换（DFT）和逆离散傅里叶变换（IDFT）来实现。频域方法通过将序列变换到频域进行点乘运算，然后再变换回时域，从而得到卷积结果。两种方法在理论上应得出相同的结果。

实验代码

函数 circonvit（）

```
function y = circonvtim(x1, x2, N)
y = zeros(1, N);
x1 = [x1, zeros(1, N - length(x1))];
x2 = [x2, zeros(1, N - length(x2))];
n = 0:N-1;
x3 = x2(mod(-n, N) + 1);
for m = 0:N-1
x4 = cirshftt(x3, m, N);
x5 = x1 .* x4;
y(m + 1) = sum(x5);
end
end
```

函数 circonvfre（）

```
function yn = circonvfre(x1, x2, N)
x1 = [x1, zeros(1, N - length(x1))];
x2 = [x2, zeros(1, N - length(x2))];
Xk1 = DFTmat(x1);
Xk2 = DFTmat(x2);
Yk = Xk1 .* Xk2;
n=0:N-1;k=n;nk=n'*k;
WN=exp(j*2*pi/N);Wnk=WN.^nk;
yn=Yk*Wnk/N;
end
```

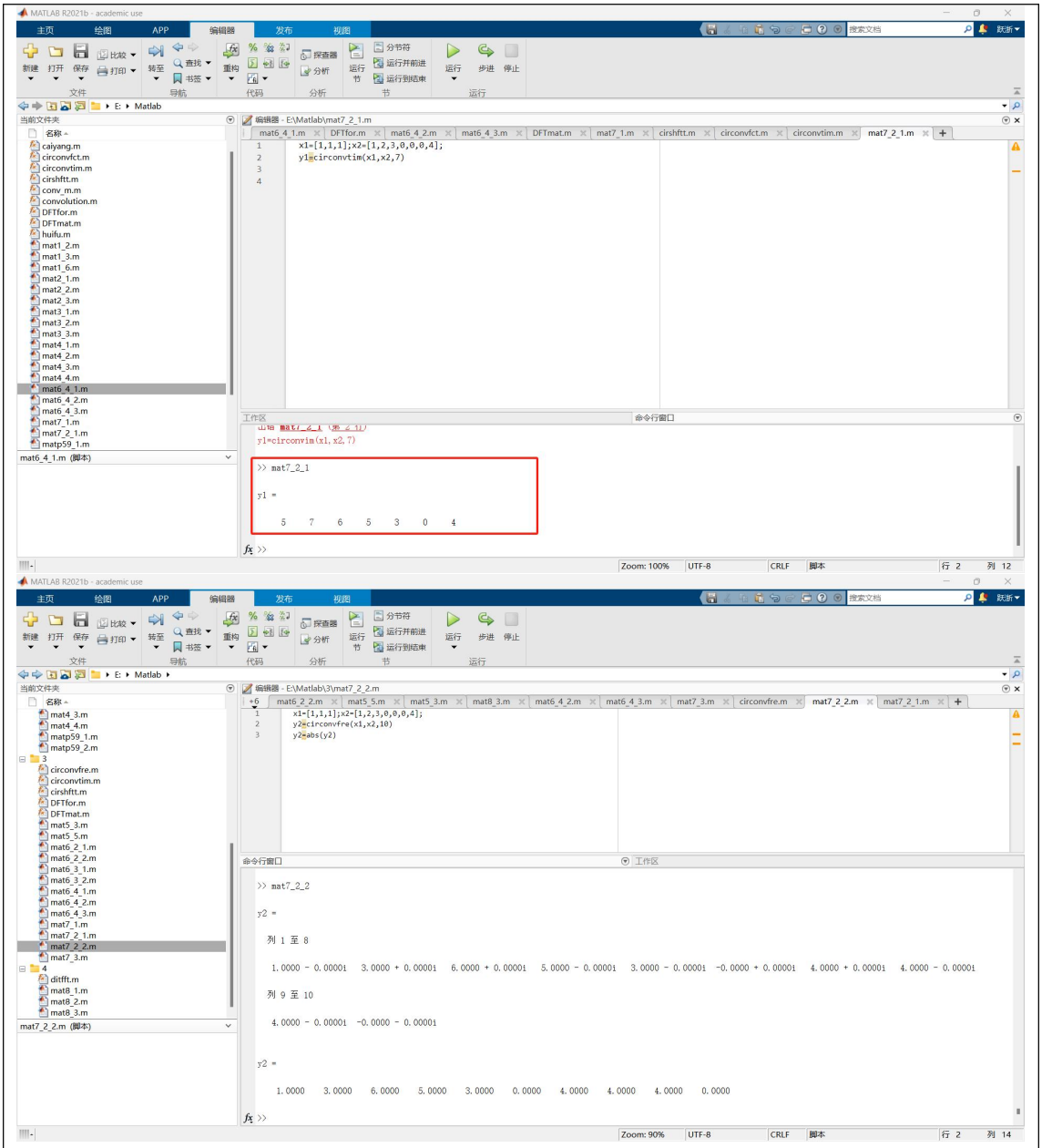
代码 1：

```
x1=[1,1,1];x2=[1,2,3,0,0,0,4];
y1=circonvtim(x1,x2,7)
```

代码 2：

```
x1=[1,1,1];x2=[1,2,3,0,0,0,4];
y2=circonvfre(x1,x2,10)
y2=abs(y2)
```

实验结果



7-3

实验目的

本实验的目的是验证线性卷积和循环卷积之间的关系。通过计算序列 x1(n)={1,1,1}和 x2(n)={1,2,3,4,5} 的线性卷积和不同长度下的循环卷积，分析并比较其结果。

实验原理

卷积是信号处理中的基本操作，用于分析信号的相互作用。线性卷积用于非周期信号的卷积，而循环卷积用于周期信号的卷积。线性卷积的结果长度为两个序列长度之和减一，而循环卷积的结果长度与序列长度相同。

线性卷积和循环卷积之间的关系可以通过以下公式描述：

$$y(n) = x_1(n) * x_2(n)$$
$$y_{\text{circ}}(n) = x_1(n) \circledast x_2(n)$$

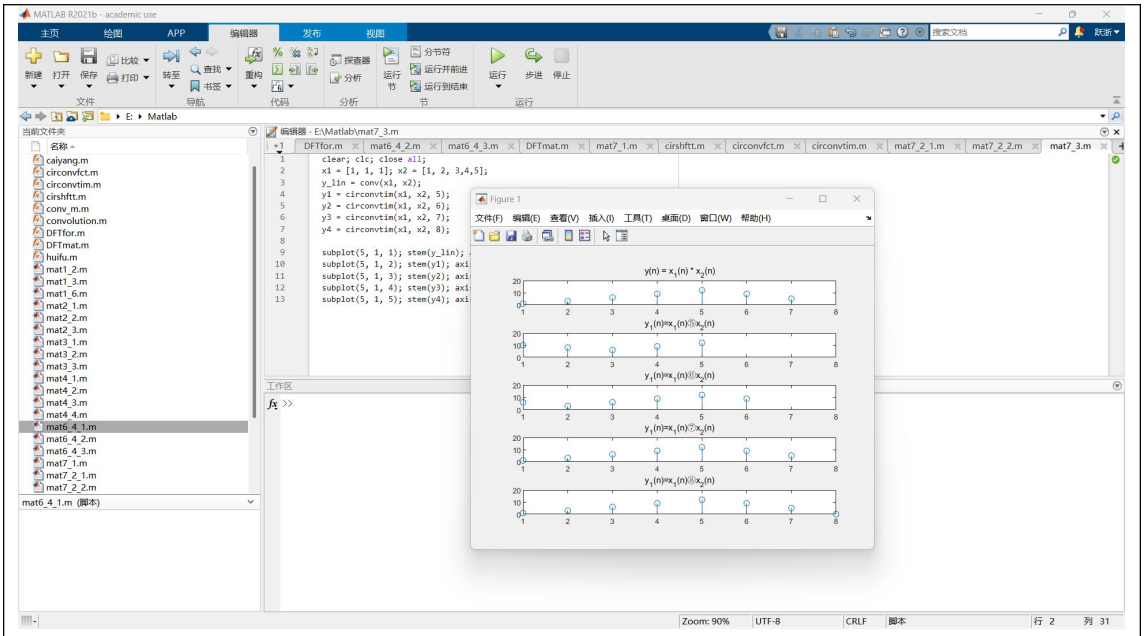
通过对不同长度的循环卷积结果进行比较，可以验证线性卷积和循环卷积之间的关系。特别地，当 循环卷积的长度大于或等于线性卷积的长度时，循环卷积的结果与线性卷积的结果相同。

实验代码

```
clear; clc; close all;
x1 = [1, 1, 1]; x2 = [1, 2, 3,4,5];
y_lin = conv(x1, x2);
y1 = circonvtim(x1, x2, 5);
y2 = circonvtim(x1, x2, 6);
y3 = circonvtim(x1, x2, 7);
y4 = circonvtim(x1, x2, 8);

subplot(5, 1, 1); stem(y_lin); axis([1, 8, 0, 20]); title('y(n) = x_1(n) * x_2(n)');
subplot(5, 1, 2); stem(y1); axis([1, 8, 0, 20]); title('y_1(n)=x_1(n)⑤x_2(n)');
subplot(5, 1, 3); stem(y2); axis([1, 8, 0, 20]); title('y_1(n)=x_1(n)⑥x_2(n)');
subplot(5, 1, 4); stem(y3); axis([1, 8, 0, 20]); title('y_1(n)=x_1(n)⑦x_2(n)');
subplot(5, 1, 5); stem(y4); axis([1, 8, 0, 20]); title('y_1(n)=x_1(n)⑧x_2(n)');
```

实验结果



实验总结:

在整个实验过程中，我通过一系列 MATLAB 仿真，验证并理解了不同信号处理操作的理论和实际效果。首先，通过对因果线性时不变系统的单位阶跃响应的仿真与理论推导的比较，我验证了使用卷积和递归方法求解系统响应的准确性和一致性。接着，通过二阶系统对随机噪声的响应仿真，我观察并分析了系统在噪声输入下的动态行为，验证了系统的稳定性和滤波特性。

在傅里叶变换的实验中，我计算并绘制了序列的离散时间傅里叶变换（DTFT）和离散傅里叶变换（DFT）的幅度和相位响应曲线。通过对比两者，我理解了 DTFT 和 DFT 在频域分析中的应用及差异，特别是频谱泄露和分辨率的影响。

在循环移位和卷积的实验中，我通过实际仿真验证了序列的多种循环移位操作，直观地理解了循环移位在信号处理中的应用。随后，通过时间域和频域卷积计算两个已知序列的循环卷积，并对比时间域和频域方法的结果，验证了它们在理论上的一致性。

最后，通过验证线性卷积和循环卷积之间的关系，我进一步理解了卷积在信号处理中的基本作用，尤其是在非周期信号和周期信号处理中的不同应用场景。整个实验过程，使我全面掌握了 MATLAB 在信号处理中的应用方法，并加深了对相关理论的理解。通过对实验结果的分析和比较，我能够验证仿真结果的准确

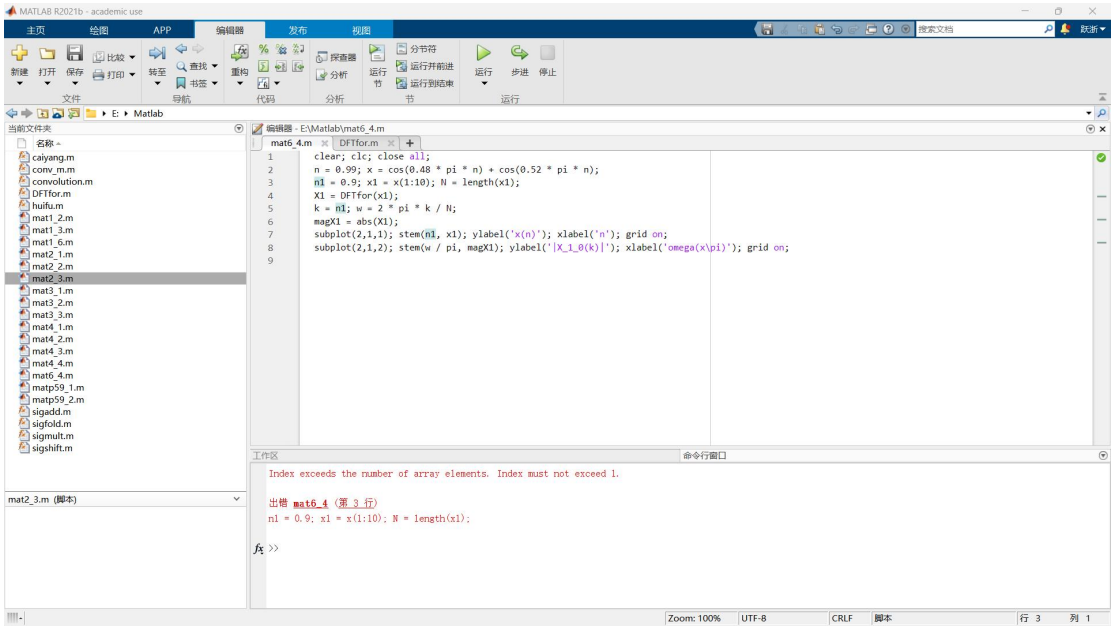
性，并有效地将理论知识应用于实际问题的解决。

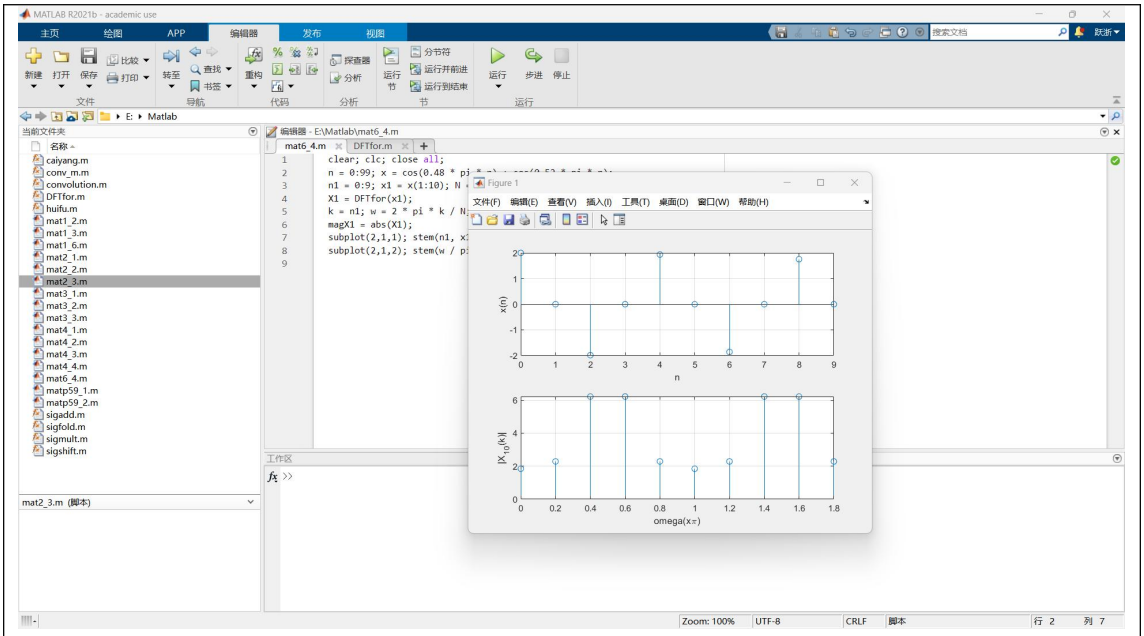
代码调试过程：

遇到的问题：编译错误

解决方法：在 MATLAB 中，n = 0:99 表示创建一个从 0 到 99 的行向量。这个行向量包含了从 0 到 99 之间的所有整数，这里的冒号运算符（:）用于生成等差序列，语法为 start:step:end，在这个例子中，start 是 0，step 默认为 1，end 是 99。因此，n = 0:99 生成了从 0 到 99 的连续整数，步长为 1。

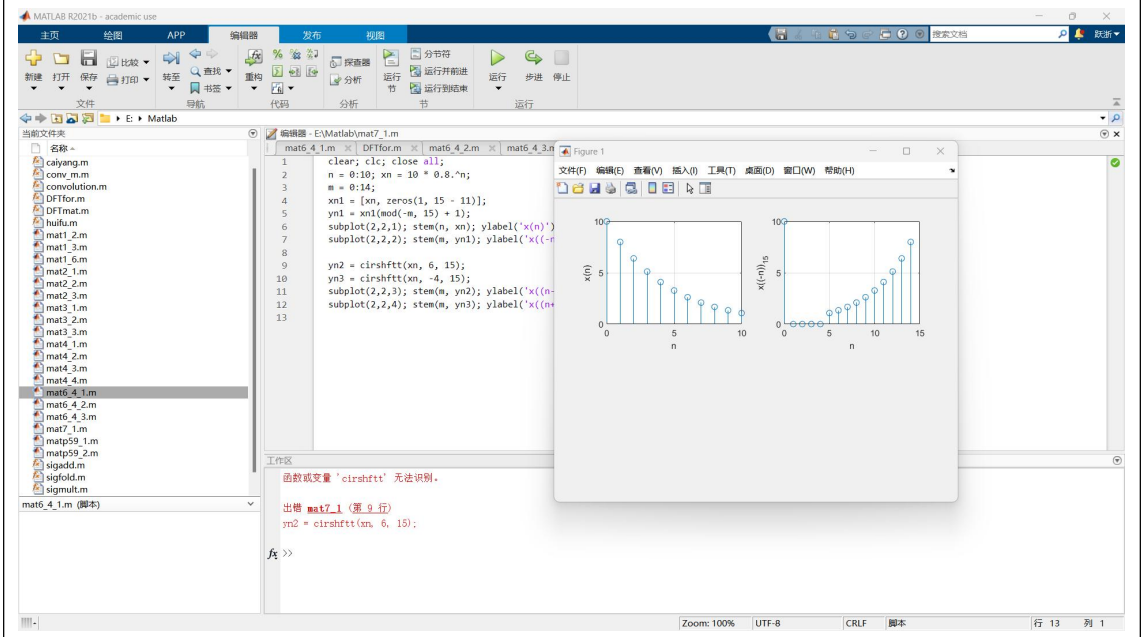
这种方式生成的向量常用于循环迭代、索引数组以及创建时间序列等应用场景。在信号处理实验中，n 常常用于表示离散时间索引。

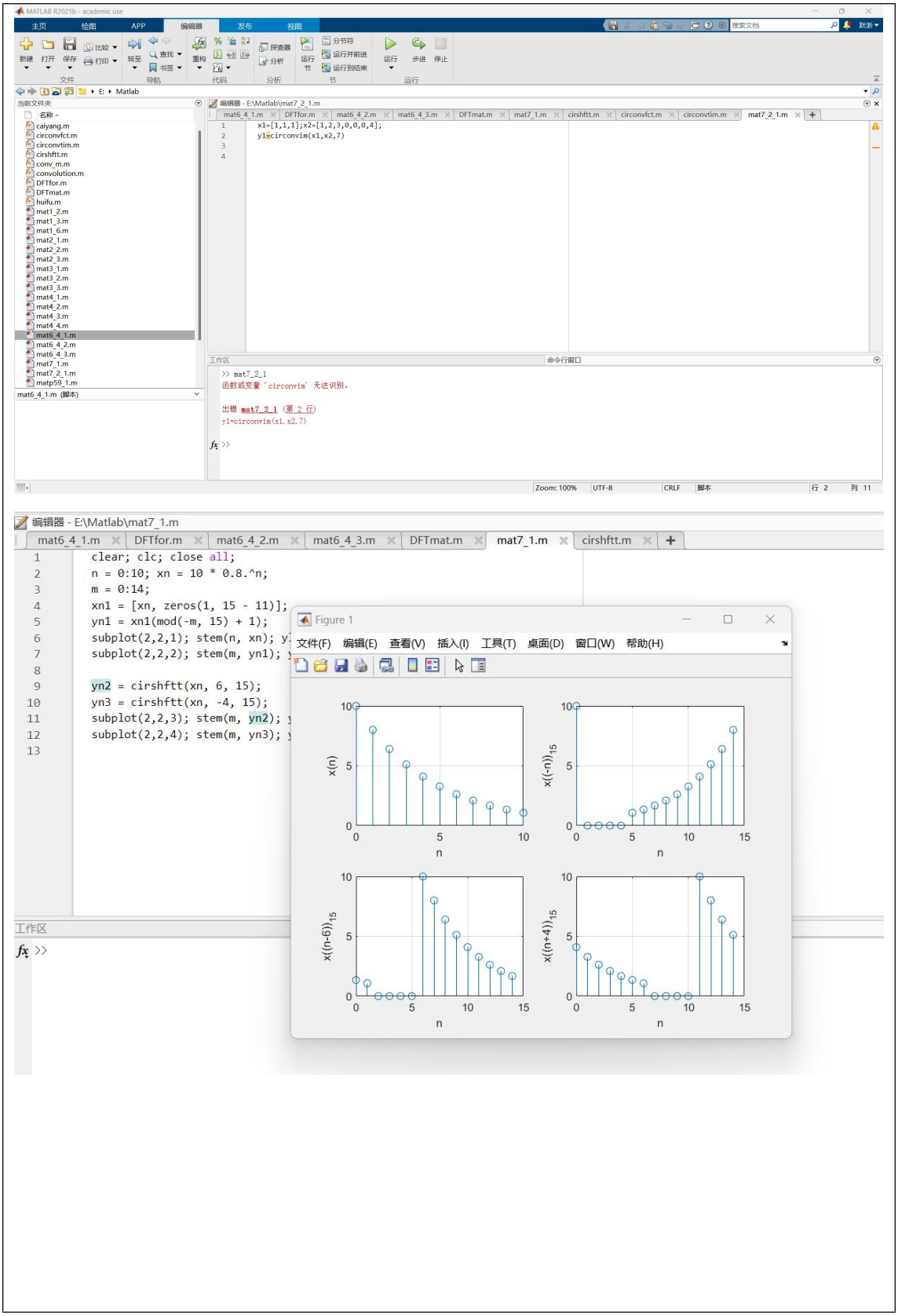


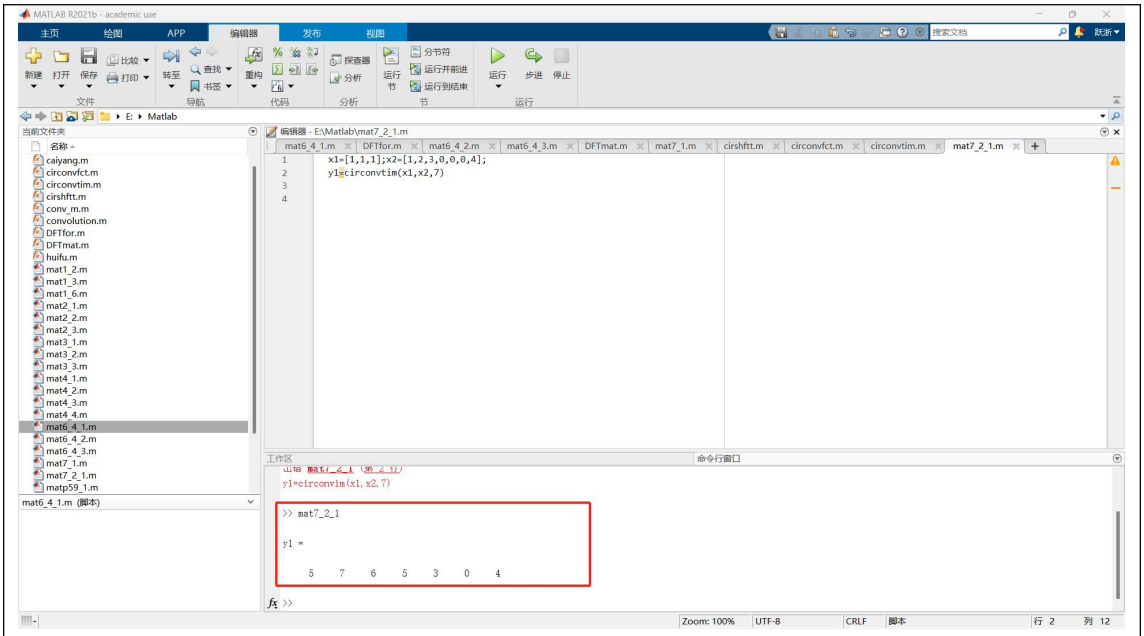


遇到的问题：函数名错误

解决方法：修正函数名




MATLAB R2021b - academic use
编辑器 - E:\Matlab\mat7_1.m
mat6_4_1.m
DFTfor.m
mat6_4_2.m
mat6_4_3.m
DFTmat.m
mat7_1.m
cirshftt.m
clear; clc; close all;
n = 0:10; xn = 10 * 0.8.^n;
m = 0:14;
xn1 = [xn, zeros(1, 15 - 11)];
yn1 = xn1(mod(-m, 15) + 1);
subplot(2,2,1); stem(n, xn); y
subplot(2,2,2); stem(m, yn1);
yn2 = cirshftt(xn, 6, 15);
yn3 = cirshftt(xn, -4, 15);
subplot(2,2,3); stem(m, yn2);
subplot(2,2,4); stem(m, yn3);
工作区
Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
x(n)
x((-n))15
x((n-6))15
x((n+4))15
n


MATLAB R2021b - academic use
主页
绘图
APP
编辑器
发布
视图
新建
打开
保存
比较
打印
转至
查找
书签
重构
探查器
分析
运行节
分节符
运行并前进
运行到结束
运行
步进
停止
文件
导航
代码
分析
节
运行
E: ▸ Matlab
当前文件夹
名称
caiyang.m
circonvfct.m
circonvtim.m
cirshftt.m
conv_m.m
convolution.m
DFTfor.m
DFTmat.m
huifu.m
mat1_2.m
mat1_3.m
mat1_6.m
mat2_1.m
mat2_2.m
mat2_3.m
mat3_1.m
mat3_2.m
mat3_3.m
mat4_1.m
mat4_2.m
mat4_3.m
mat4_4.m
mat6_4_1.m
mat6_4_2.m
mat6_4_3.m
mat7_1.m
mat7_2_1.m
matp59_1.m
mat6_4_1.m (脚本)
编辑器 - E:\Matlab\mat7_2_1.m
mat6_4_1.m
DFTfor.m
mat6_4_2.m
mat6_4_3.m
DFTmat.m
mat7_1.m
cirshftt.m
circonvfct.m
circonvtim.m
mat7_2_1.m
1 x1=[1,1,1];x2=[1,2,3,0,0,0,4];
2 y1=circonvtim(x1,x2,7)
3
4
工作区
命令行窗口
y1=circonvtim(x1,x2,7)
>> mat7_2_1
y1 =
5 7 6 5 3 0 4
fx >>
Zoom: 100%
UTF-8
CRLF
脚本
行 2
列 12