

# 信号与系统实验报告

名 称： 数字信号卷积和的实现及应用

学 院： 计算机科学与工程学院

专 业： 计算机科学与技术专业

学 号： 09021235

姓 名： 江紫弦

日期： 2023 年 3 月 21 日

评分：

## 一、实验目的

1. 掌握 Matlab 中相关函数的使用，程序代码编制与调试的流程。
2. 熟悉卷积和的运算规则及其意义，加深对离散时间信号分析的理解。

## 二、实验任务

1. 完成实验内容全部题目，分析解决调试代码过程中出现的问题。
2. 认真完成本次实验小结，思考卷积和的应用。

## 三、主要设备、软件平台

1. 硬件：计算机
2. 软件：Matlab

## 四、实验内容

1. 输出杨辉三角。
  - 1) 函数编写
  - 2) 控制台输出
  - 3) 循环语句、条件语句
  - 4) 程序运行、调试
2. 编程实现信号  $x(n)$ ,  $h(n)$  间的卷积和运算函数  $my\_cov(x,h)$ , 并绘制出下列信号卷积和波形。

1)  $x(n)=[1,2,3,...,10]$ ,  $h(n)=[1,1]$

2)  $x(n)=[1,2,3,9,4,5,6,0,7,8]$ ,  $h(n)=[-1,2,-1]$

3)  $x(n)=[1,2,3,...,10]$ ,  $h(n)=[1,2,3,...,10]$

3. 设计游戏时，若对小怪使用一次技能的效果是“小怪会在接下来 5 秒内持续掉血，每秒掉血量分别为[5 4 3 2 1]”；如果间隔 1 秒连续发动 3 次技能，请绘制出每次攻击后小怪的累计掉血量情况。

## 五、探究拓展

1. 给定一个如下所示的二维矩阵，实现其自身的卷积运算。

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

## 六、实验结果

### 1、杨辉三角

#### (1) 思路

杨辉三角的特点：第  $n+1$  行的第  $i$  个元素等于第  $n$  行的第  $i-1$  个数和第  $i$  个数之和，即  $\text{triangle}(i,j) = \text{triangle}(i-1,j-1) + \text{triangle}(i-1,j)$ 。我们知道杨辉三角的每一行的最左边和最右边元素均为 1，则杨辉三角的第一行和第二行均为 1，所以我们可以用循环逐行求出指定行数的杨辉三角。

#### (2) 代码

```
% 定义杨辉三角的行数
```

```
n = 10;
```

```
% 初始化杨辉三角矩阵
```

```
triangle = zeros(n);
```

```
% 计算杨辉三角
```

```
for i = 1:n
```

```
    triangle(i,1:i) = 1;%先将三角范围全初始化为 1（也可以只初始化左边和后边两列为 1）
```

```

        for j = 2:i-1
            triangle(i,j) = triangle(i-1,j-1) + triangle(i-1,j);
        end
    end

% 输出杨辉三角
disp('杨辉三角: ');
for i = 1:n
    for j = 1:i
        fprintf('%d ',triangle(i,j));
    end
    fprintf('\n');
end

```

### (3)输出（打印一个十行的杨辉三角）

```

命令窗口
>> Yanghui_Triangle
杨辉三角:
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
fx >>

```

## 2. 卷积和运算

### （1）思路

#### ①公式法

卷积公式为 $y(n) = x(n) * h(n) = \sum_{k=0}^n x(k)h(n-k)$ ; 由于题目中要求对有限长

离散信号进行卷积运算，在 matlab 中可对 n, k 进行双重循环，对数组进行访问，按公式累积求和，即可得到卷积结果。

## ②图解法

图解法原理如下图；在 matlab 运用数组进行数据存储，易对 h\_n 数组进行反褶、平移，再将 x\_n 数组右移 h\_len-1 个，即可完成 x(k) 与 h(-k) 元素的下标对齐。之后利用循环，进行相乘叠加，每次循环后对 h\_n 数组右移一位，直至两个数组的下标完全错开，这之后对应 n 处的 y\_n 值恒等于 0。由于在运算过程中反复用到右移操作，单独设计一个右移函数 MoveRight。

**卷积和计算：** 
$$y(n) = \sum_{k=0}^n x(k)h(n-k)$$

1) 图解法： 反褶——>平移——>相乘——>叠加（求和）

例2: x(n)={2,1,5}, h(n)={1,2,3}

x(k):	2 1 5	
h(k):	1 2 3	
n=0, h(-k):	3 2 1	—>y(0)=2
n=1, h(1-k):	3 2 1	—>y(1)=5
n=2, h(2-k):	3 2 1	—>y(2)=13
n=3, h(3-k):	3 2 1	—>y(3)=13
n=4, h(4-k):	3 2 1	—>y(4)=15
n=5, h(5-k):	3 2 1	—>y(5)=0
n=6, h(6-k):	3 2 1	—>y(6)=0

.....

所以, y(n)={2,5,13,13,15}

(2) 代码

1) 右移函数

```
function [x_MoveRight,Newleft] = MoveRight(x,left,m)
```

```
%将数组 x 右移 m 位，前面补 0(有效位数的下标+m)
```

```
%left 是 x 最左边一个有效数字的下标
```

```
%Newleft 是 x_MoveRight 最左边一个有效数字的下标
```

```
x_len=length(x);
```

```
x_MoveRight(1,1+m:x_len+m)=x(1,1:x_len);
```

```
for i=1:m
```

```
    x_MoveRight(i)=0;
```

```

end
Newleft=left+m;
end

```

## 2) 公式法实现卷积运算

```

function [y_n] = convolution1(x_n,h_n)
%用公式法实现卷积
x_len=length(x_n);
h_len=length(h_n);

for n=1:x_len+h_len
    y_n(n)=0;
    for k=max(1,n-h_len):min(x_len,n-1)
        y_n(n)=y_n(n)+x_n(k)*h_n(n-k);
    end
end
y_n(1)=[];
end

```

## 3) 图解法实现卷积运算

```

function [y_n] = convolution(x_n,h_n)
%编程实现信号 h(n),x(n)间的卷积和运算函数 y(n) 图解法
%取长度
x_len=length(x_n);
h_len=length(h_n);

%偏移量
offset=h_len-1;

```

```

% 翻转 h(n)
h_n=flip1r(h_n);

%将 x_n 右移 offset 个
[x_n,1]=MoveRight(x_n,1,offset);

%每次循环 h_n 右移一位,初始化 h_n 有效数字的最左一位下标
left=1;

%用循环相乘后叠加
for i=1:x_len+h_len
    if x_len+offset<left
        break;%之后的 y_n(i)==0
    else
        y_n(i)=0;%初始化 y_n(i)的值
        for j=1:min(h_len+left-1,x_len)%j 为当前列数关于 x_n 数
            组有效数字最左边一个值的下标的偏移量
            y_n(i)=y_n(i)+x_n(j+offset)*h_n(j+offset);%相乘
            后叠加
        end
        [h_n,left]=MoveRight(h_n,left,1);%每次循环后 h_n 右移一
        位
    end
end
end

```

4) 分别调用公式法和图解法公式运算卷积函数完成题中要求

```
clc
```

```
clear
```

```
%% 编程实现信号 h(n),x(n)间的卷积和运算函数 y(n)
```

```

% 用图解法实现卷积
x_n=[1 2 3 4 5 6 7 8 9 10];
h_n=[1 1];
y_n1=convolution(x_n,h_n)

x_n=[1 2 3 9 4 5 6 0 7 8];
h_n=[-1 2 -1];
y_n2=convolution(x_n,h_n)

x_n=[1 2 3 4 5 6 7 8 9 10];
h_n=[1 2 3 4 5 6 7 8 9 10];
y_n3=convolution(x_n,h_n)

%% 用公式法实现卷积
x_n=[1 2 3 4 5 6 7 8 9 10];
h_n=[1 1];
y_n1=convolution1(x_n,h_n)

x_n=[1 2 3 9 4 5 6 0 7 8];
h_n=[-1 2 -1];
y_n2=convolution1(x_n,h_n)

x_n=[1 2 3 4 5 6 7 8 9 10];
h_n=[1 2 3 4 5 6 7 8 9 10];
y_n3=convolution1(x_n,h_n)

%% 绘制
n=1:length(y_n1)+1;
figure('name','第一小题运算结果');
stem(y_n1,'filled');

```



```

n=1:length(y_n2)+1;
figure('name','第二小题运算结果');
stem(y_n2,'filled');

n=1:length(y_n3)+1;
figure('name','第三小题运算结果');
stem(y_n3,'filled');

```

### (3) 输出

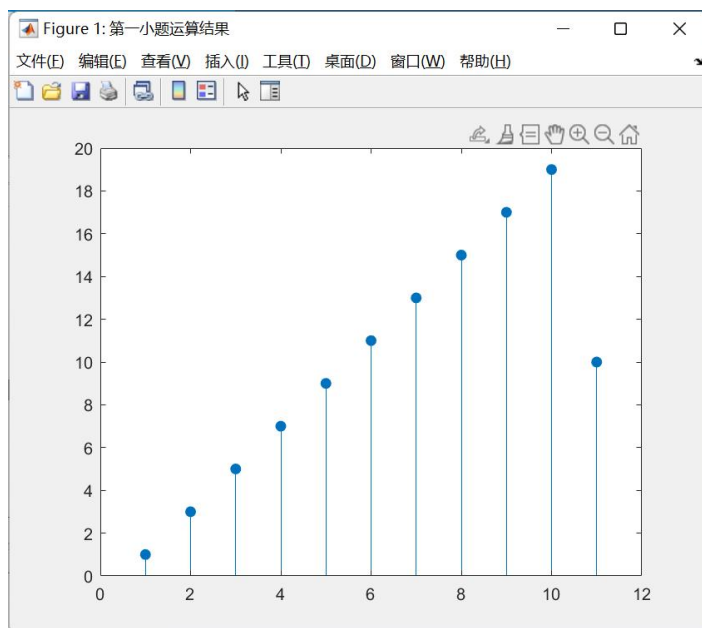
#### ①第一小题

```

y_n1 =

    1     3     5     7     9    11    13    15    17    19    10

```



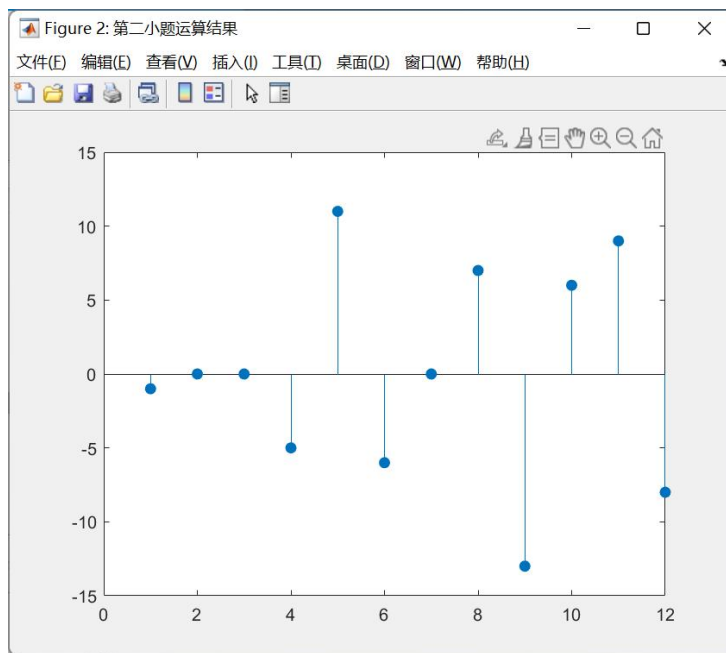
#### ②第二小题

```

y_n2 =

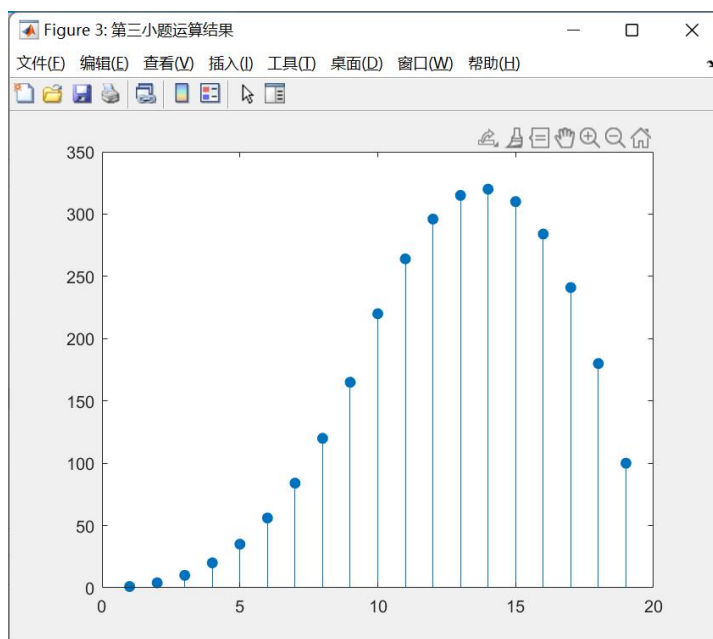
   -1     0     0    -5    11    -6     0     7   -13     6     9    -8

```



### ③第三小题

```
y_n3 =
    1     4    10    20    35    56    84   120   165   220   264   296   315   320   310   284   241   180   100
```



### 3、打怪游戏

#### (1) 思路

将问题转化为  $x(n)=[1 \ 1 \ 1]$ ,  $h(n)=[5 \ 4 \ 3 \ 2 \ 1]$  两个信号的卷积运算

#### (2) 代码

```
clc
```

```
clear
```

```
%将问题转化为[1 1 1]和[5 4 3 2 1]两个信号的卷积
```

```
x_n=[1 1 1];
```

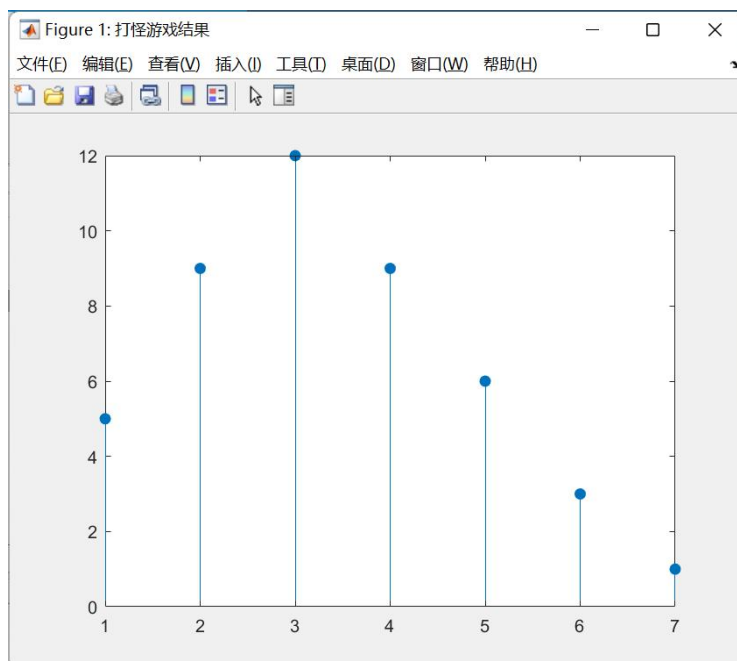
```
h_n=[5 4 3 2 1];
```

```
y_n=convolution(x_n,h_n)
```

(3)输出

```
y_n =
```

5      9      12      9      6      3      1



每次攻击后的掉血情况为[5 9 12 9 6 3 1]。

#### 4、矩阵卷积

##### (1) 思路

设待运算的两矩阵为 A, B, 其大小分别为  $m \times m, n \times n$  (假设  $n$  为奇数)。矩阵卷积运算步骤如下:

- 让卷积核 B 翻转 180 度
- 对矩阵 A 补零, 即对矩阵上下左右都补  $(n-1)/2$  行(列) 0, 得到一个  $(m+n-1) \times (m+n-1)$  大小的矩阵 Y;
- 滑动旋转后的 B, 使 B 的中心位于矩阵 Y 的每一个元素, 并求卷积和 (即将旋

转后的 B 在 Y 上进行华东, 然后对应位置相乘后相加的结果就是当前矩阵 Y 元素对应位置上的卷积结果)

d. 在计算好的矩阵 Y 中心取  $m \times m$  大小的矩阵, 即为矩阵 A, B 卷积后的结果

## (2) 代码

### 1) 实现矩阵卷积的函数

```
function [Y] = MatrixConvolution(A,B)
```

%矩阵 A 和 B 卷积 前提: 卷积核 B 的行列数为奇数

%先将 B 翻转 180 度

```
B=flip(B);
```

```
B=flip(B,2);
```

```
m=size(A);
```

```
m=m(2);
```

```
n=size(B);
```

```
n=n(2);
```

```
Y=zeros(m+n-1,m+n-1);
```

```
Y1=zeros(m+n-1,m+n-1);
```

```
Y((n-1)/2+1:(n-1)/2+m,(n-1)/2+1:(n-1)/2+m)=A;
```

```
for i=1:m
```

```
    for j=1:m
```

```
        re=0;
```

```
        for k1=1:n
```

```
            for k2=1:n
```

```
                re=re+B(k1,k2)*Y(i+k1-1,j+k2-1);
```

```
            end
```

```
        end
```

```
        Y1((n-1)/2+i,(n-1)/2+j)=re;
```

```

        end
    end
Y=Y1((n-1)/2+1:(n-1)/2+m,(n-1)/2+1:(n-1)/2+m);
end

```

2)调用矩阵卷积函数完成题目要求运算

```

clc
clear
%% 实现矩阵二维卷积
A=[0 1 0;1 1 1;0 1 0];
Y=MatrixConvolution(A,A)

```

(3) 输出

```

命令行窗口

Y =

     2     2     2
     2     5     2
     2     2     2

```

## 七、实验心得

在写卷积运算代码时，我的第一反应是实现更为复杂的图解法。图解法需对数组的下标严谨计算，同时需要防止数组越界问题，让我对课堂上介绍的图解法有了更深的印象和理解。其次，我还用公式法实现了卷积运算，公式法的实现较为简单，但同样需要对下标进行严谨计算。两种思路下实现对离散信号的卷积运算，让我巩固了上课所学知识。同时拓宽了视野，让我学习了二维矩阵卷积的计算方法并成功实现。