

# 重庆邮电大学

## 学生实验实习报告册

学年学期： 2020 -2021 学年 ☐春 ☒秋学期

课程名称： 信号处理实验

学生学院： 通信与信息工程学院

专业班级： 01011803

学生学号： 2018210129

学生姓名： 张海怡

联系电话：

重庆邮电大学教务处制

课程名称	信号处理实验	课程编号	
实验地点	YF315	实验时间	周二 12 节
校外指导教师		校内指导教师	邵凯
实验名称	系统响应及系统稳定性		
评阅人签字		成绩	

一、实验目的

学会运用 MATLAB 求解离散时间系统的零状态响应；

学会运用 MATLAB 求解离散时间系统的单位取样响应；

学会运用 MATLAB 求解离散时间系统的卷积和。

二、实验原理

MATLAB 中函数 `filter` 可对差分方程在指定时间范围内的输入序列所产生的响应进行求解。函数 `filter` 的语句格式为 `y=filter(b,a,x)` 其中，`x` 为输入的离散序列；`y` 为输出的离散序列；`y` 的长度与 `x` 的长度一样；`b` 与 `a` 分别为差分方程右端与左端的系数向量。

MATLAB 求解单位取样响应可利用函数 `filter`，并将激励设为单位抽样序列。

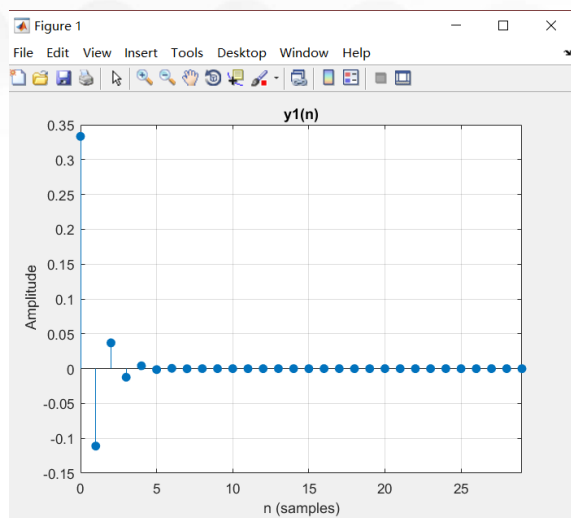
MATLAB 另一种求单位取样响应的方法是利用控制系统工具箱提供的函数 `impz` 来实现。`impz` 函数的常用语句格式为 `impz(b,a,N)` 其中，参数 `N` 通常为正整数，代表计算单位取样响应的样值个数。

MATLAB 求离散时间信号卷积和的命令为 `conv`，其语句格式为 `y=conv(x,h)` 其中，`x` 与 `h` 表示离散时间信号值的向量；`y` 为卷积结果。用 MATLAB 进行卷积和运算时，无法实现无限的累加，只能计算时限信号的卷积。

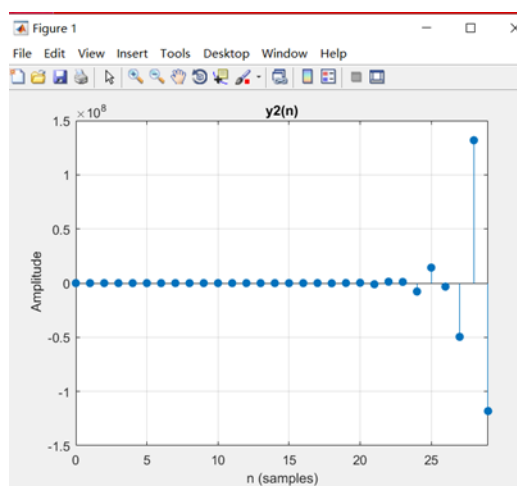
三、实验程序及结果分析

```
a1=[3,4,1];
```

```
b1=[1,1];  
impz(b1,a1,30);  
grid on;  
title('y1(n)');
```



```
a2=[2.5,6,10];  
b2=[1];  
impz(b2,a2,30);  
grid on;  
title('y2(n)');
```



```
nx=-2:6;  
nh=-4:12;
```

The figure displays three vertically stacked plots in a MATLAB environment, illustrating the convolution of two discrete-time signals. The x-axis for all plots is labeled  $n$  and ranges from -4 to 16. The y-axis for the top two plots ranges from 0 to 2, while the bottom plot ranges from 0 to 4.

- Top Plot:** Shows the signal  $x(n]$ . It is a discrete-time signal with non-zero values at  $n = -2, -1, 0, 1, 2, 3, 4, 5, 6$ . The values are approximately 0.5, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0, 0.5, and 0.5 respectively.
- Middle Plot:** Shows the signal  $h(n]$ . It is a discrete-time signal with non-zero values at  $n = -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$ . The values are approximately 0.5, 0.5, 0.5, 0.5, 1.0, 1.0, 1.0, 1.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, and 0.5 respectively.
- Bottom Plot:** Shows the resulting signal  $y(n) = x(n) * h(n]$ . It is a discrete-time signal with non-zero values at  $n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16$ . The values are approximately 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.0, 2.5, 2.0, 1.5, 1.0, 0.5, 0.5, 0.5, 0.5, 0.5, and 0.5 respectively.

**nh=-1:4;**

```
h=[2,3,0,-5,2,1];  
ny1=nx(1)+nh(1);  
ny2=nx(length(x))+nh(length(h));  
ny=ny1:ny2;  
y=conv(x,h);  
stem(ny,y);  
xlabel('n');ylabel('y');grid on;title('y(n)');
```

