

回声系统设计

设计详细说明：

1. 社回声系统的单位冲击响应函数 $h[n]$ 为：

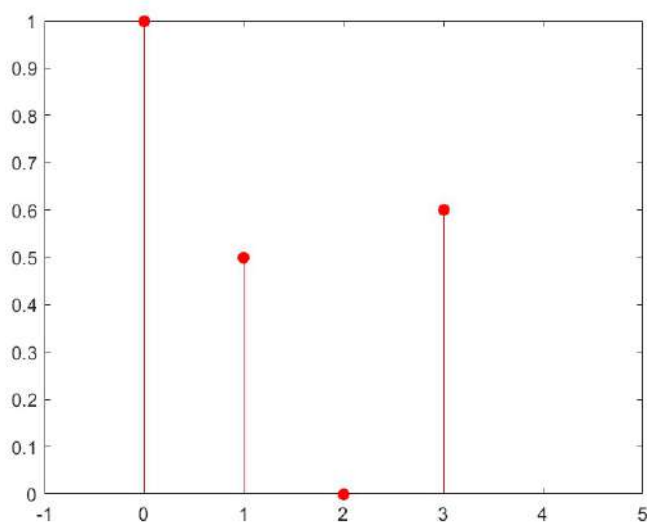
$$h[n] = \begin{cases} 1, & n = 0 \\ 0.5, & n = 1 \\ 0.6, & n = 3 \\ 0, & \text{其他} \end{cases}$$

2. 假设简单输入信号 $x[n]$ 为：

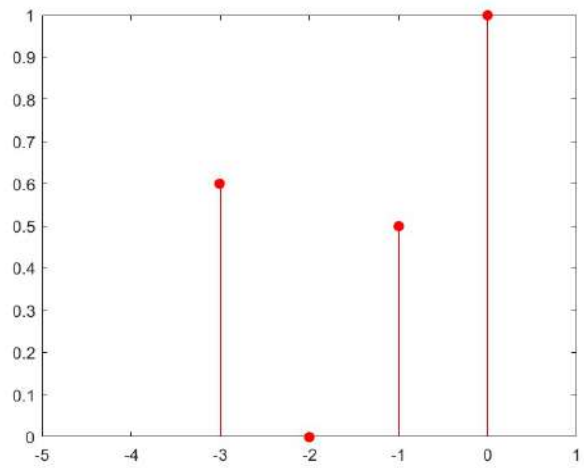
$$x[n] = \begin{cases} 1, & n = 0 \\ 2, & n = 1, \\ -1, & n = 2 \\ 0, & \text{其他} \end{cases}$$

则根据卷积和公式，输出 $y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$

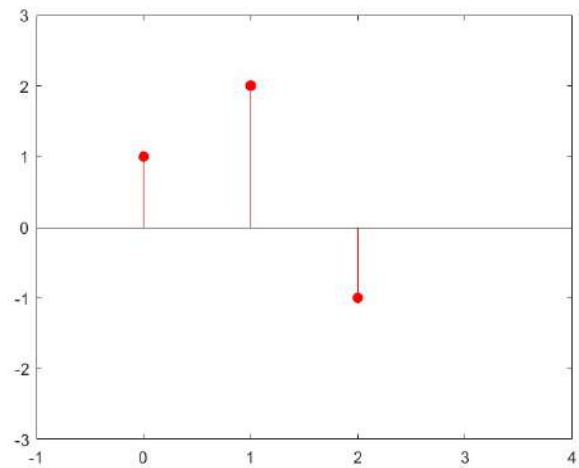
- 1) $h[k]$ 的图像为：



- 2) $h[-k]$ 的图像为：



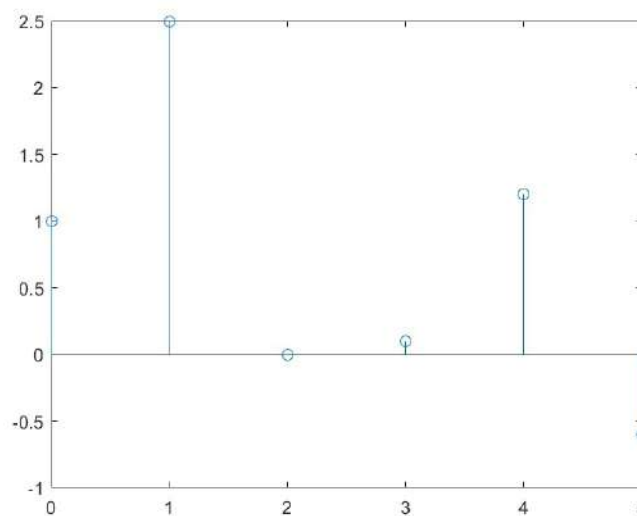
3) $x[k]$ 的图像为:



通过公式计算输出信号表达式:

- $n < 0$ 时或 $n > 5$ 时, $y[n]$ 恒为 0;
- $n = 0$ 时, $y[0]$ 恰为 1
- $n = 1$ 时 $y[1] = 1 * 0.5 + 2 * 1 = 2.5$
- $n = 2$ 时, $y[2] = 1 * 0 + 2 * 0.5 + (-1) * 1 = 0$
- $n = 3$ 时, $y[3] = 1 * 0.6 + 2 * 0 + (-1) * 0.5 = 0.1$
- $n = 4$ 时, $y[4] = 1 * 0 + 2 * 0.6 + (-1) * 0 = 1.2$
- $n = 5$ 时, $y[5] = 1 * 0 + 2 * 0 + (-1) * 0.6 = -0.6$

得出 $y[n]$ 的表达式: $y(n) = \begin{cases} 0, & n < 0 \\ 1, & n = 0 \\ 2.5, & n = 1 \\ 0, & n = 2 \\ 0.1, & n = 3 \\ 1.2, & n = 4 \\ -0.6, & n = 5 \\ 0, & n > 5 \end{cases}$

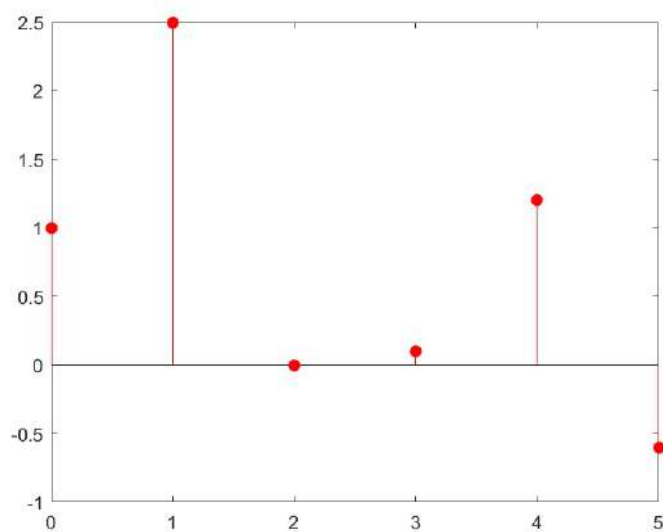


3. 在 MATLAB 中建立回声系统模型

具体代码如下: (**conv.m** 文件) (文件名不能写成 conv 不然会 bug)

```
h=[1 0.5 0 0.6]; %单位冲激响应
x=[1 2 -1]; %输入信号
y=conv(h,x); %利用conv() 卷积函数
n=0:1:length(y)-1; %n从0开始
stem(n,y,'filled','red')
```

具体结果如图:



与计算得出的结果图形相同

再进行计算编程：

- 1) 卷积实际上是一个累乘累加的过程；
- 2) 对 $h[k]$ 来说， $h[n-k]$ 是对 $h[k]$ 进行了倒序后进行右移与 $x[k]$ 非零相重合部分相乘；
- 3) 要想利用 matlab 帮助我们进行累加计算，需要将 $x[k]$ 和 $h[n-k]$ 对应的矩阵填补成为同维矩阵，且填补后的矩阵长度正是输出 $y[n]$ 的矩阵长度；
- 4) 建立 $y[n]$ 的 0 矩阵用于接收累乘累加后的结果
- 5) 绘制输出的 $y[n]$ 图像

编程代码：(my_conv.m 文件)

```
x = [1 2 -1]; %输入信号

h = [1 0.5 0 0.6]; %冲激响应函数

x0 = [ x,zeros(1,length(h)-1)];

h0 = [zeros(1,length(x)-1), flip(h)]; %给两个矩阵补0
```

到同维度才可以利用矩阵的计算方法，响应函数矩阵将其顺序倒置

```
n = length(x0) ; %计算得到的输出信号长度

n0 = 0:1:n-1; %绘图时n从0开始

y = zeros(1,n); %建立存储结果的矩阵

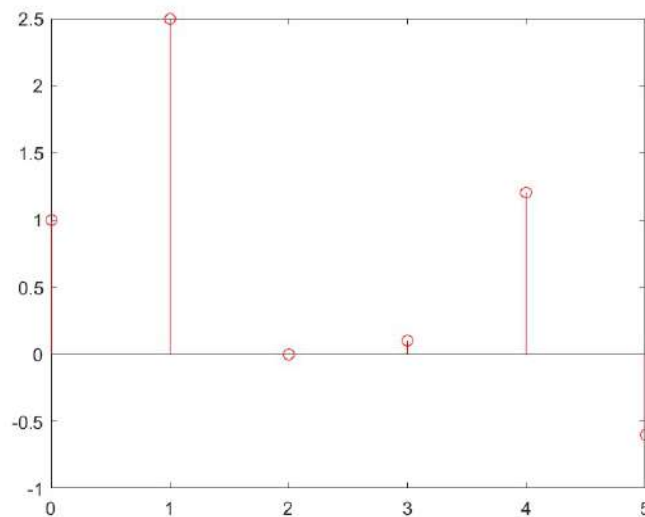
for i = 1:n %迭代计算

    y(i) = sum( x0(1:i) .* h0(end-i+1:end));

    %让x[k]和h[n-k]有非零重叠部分累乘累加存入y[ni]

end

stem(n0,y,'red')
```



4. 利用 matlab 调用声卡进行回音程序的设计

在设计前提出以下问题

- 1) 思考录音时间、录音的分辨率、回音次数应该多少合适
- 2) 该怎样将获得的录音信息转化成为可以用于计算的数字信息
- 3) 需要怎么定义冲激响应函数
- 4) 利用 conv 卷积函数得出的 y 数字信息怎么转化为可以播放的音

频信息

这是在设计过程中几个比较重要的问题

- 第一次编程：（选择了简单的单声道，结果就是电流音大，音质失真了）

编程代码：（huiyinandluyin.m 文件）

```
fs = 10000;

%设置录音的分辨率（这个是多次试验得出的）

duration = 3;

%设置录音时间，都用的duration，便于修改，录音时间为duration-1s

fprintf(' Press any key to start % g seconds of re-
cording. . . ',duration-1) ;

%录音标识

pause; %停顿便于及时反应录音

fprintf( 'Recording... ');

%进行录音

myRecording = audiorecorder;
recordblocking(myRecording, duration);

f0=getaudiodata(myRecording);%这里得到的f0是列向量

%存储录音信息

filename0='chushiyin1.wav';
audiowrite(filename0,f0,10000);

%结束标识

fprintf ( 'Finished Recording. . . ' ) ;

%设置冲激函数h (t) 系数

h0 = 2 ;
```

```

h1 =1 ;
h2 =0.5;

%用于设置分辨率矩阵

T =1/fs;
t =0: T: duration;

%冲激响应函数（自定义delta函数在函数底下）

h = h0*delta(t) +h1*delta(t-0.5) + h2*delta(t-0.8);

f0=f0';%f0变为行向量

%卷积

y =conv(f0,h);

%y再次转置用来播放声音信息

y=y';

%播放回声信息

k=audioplayer(y,fs);

play(k)

%存储回音文件wav

filename = 'huiyin1.wav';

audiowrite(filename,y,10000);

%绘制两次声音的波形

subplot(211);

plot (f0) ;

title('原始音');

subplot(212);

plot (y);

title('回音');

```

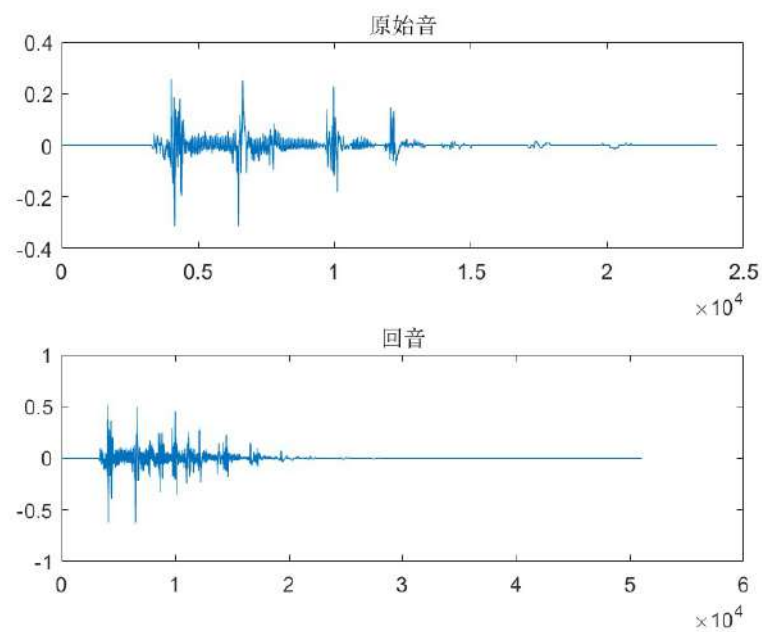
delta函数：（delta.m文件）

```
function y = delta (t)

for i =1 : length(t)
    if t(i) == 0
        %t=0时为1其余为0
        y(i) = 1;
    else
        y(i) =0;
    end
end

end
```

输出结果波形如下图所示：



huiyin1.wav

输出回音文件：

- 第二次编程：（改进了分频时间间隔，音质得到较好的优化）

编程代码：（huiyinandluyin2.m 文件）


```

clear,clc,close all;

% 1, 记录数据

fs=8000; % 声音频率

duration=3; % 记录时间

%记录输入声音
myrecording = audiorecorder;
fprintf('Speaking for %d seconds. \n',duration);
disp('Start speaking.')
recordblocking(myrecording, duration);
disp('End of Recording.');
```

$x = \text{getaudiodata}(\text{myrecording});$

```

%存储录音信息

filename0='chushiyin2.wav';
audiowrite(filename0,x,fs);

% n的长度

n=length(x);
duration=n/fs;

% 定义冲激响应函数h(t)

t=[0:1/fs:duration]';
h = 1*delta(t-0.2) + 0.8*delta(t-0.4) + 0.6*delta(t-
0.6) + ...
    0.4*delta(t-0.8) + 0.2*delta(t-1.0);

% 卷积计算
```

```

y=conv(x,h);
k=audioplayer(y,fs);

play(k); % 播放回音

%存储回音

filename = 'huiyin2.wav';
audiowrite(filename,y,fs);

%绘图

subplot(211);

plot (x) ;%绘制原始音

title('原始音');

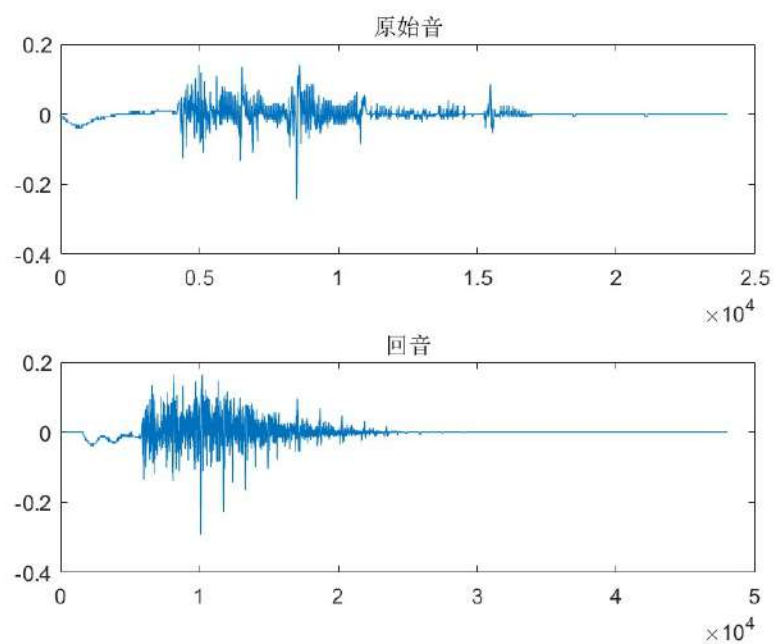
subplot(212);

plot (y);%绘制回音

title('回音');

```

输出结果波形如下图所示：





huiyin2.wav

输出回音：

5. 参考文献：

[1]刘正春,赵锦成,王勇,王文婷.基于 Matlab 的卷积教学仿真设计——回音系统[J].实验科学与技术,2015,13(05):62-64.

[2] <https://www.zhihu.com/question/387811709>

[3]https://blog.csdn.net/qg_45499204/article/details/103539290

6. 个人感想：

这个题目看起来挺复杂的，实际做下来慢慢分析卷积的原理其实还是可以挺好做的，题目的可操作空间很大，至于输出回音的结果也可以多样化，应该可以在这个代码的基础上实现变声等对声音的变化操作；分频的处理操作还是比较重要的，处理不好失真严重；