



第一章 离散时间信号与系统

Discrete-time signals and systems

1.1 离散时间信号 —— 序列

MATLAB下序列运算的实现方法

华东理工大学信息科学与工程学院 万永菁





MATLAB下序列运算的实现方法

- MATLAB编程实现**单位抽样序列** $\delta(n)$
- 序列**求和、求积**的编程方法
- 序列**移位、反褶**的编程方法

例程：用MATLAB产生并画出单位抽样序列 $\delta(n)$ 。

% 确定位置向量(横坐标)显示范围

`n = -10:20;`

`zeros(m,n)` 产生
一个 $m \times n$ 的全零矩阵

% 确定与位置向量(横坐标)匹配的 $\delta(n)$

`delta = [zeros(1,10) 1 zeros(1,20)];`

% 画出 $\delta(n)$

% 图中显示网格

`stem(n, delta); grid on;`

% 位置向量(横坐标)标识

% 序列(纵坐标)标识

`xlabel('Time index n'); ylabel('Amplitude');`

% 图名

`title('Unit Sample Sequence');`

% 调整坐标显示范围

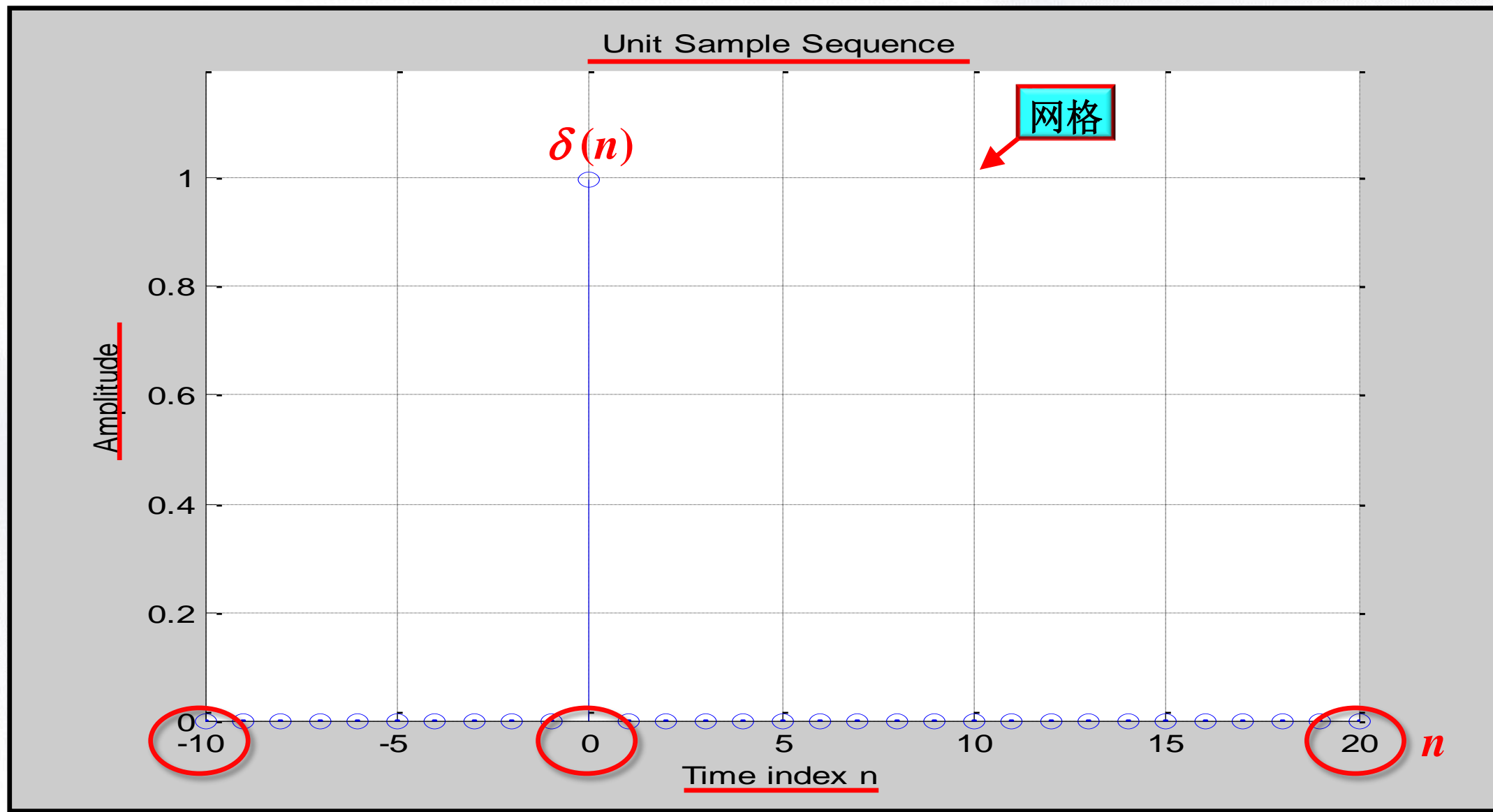
`axis([-10 20 0 1.2]);` → `axis([minX maxX minY maxY])`



例程结果： 用MATLAB产生并画出单位抽样序列 $\delta(n)$ 。



华东理工大学



例程：定义函数生成单位抽样序列 $\delta(n-np)$ 。

% 定义函数impseq, 函数参数为np, ns和nf, 返回值为n和x

function [x,n] = **impseq**(np,ns,nf)

$$x(n) = \delta(n - np), \quad n \in [ns, nf]$$

% 检查输入参数正确性

if ((np < ns) | (np > nf) | (ns > nf))

error('参数必须满足 ns <= np <= nf')

end

% 生成序列横坐标n

n = [ns:nf];

% 生成序列纵坐标x

x = [(n-np) == 0];

该函数存为文件impseq.m, 假设在dsp文件夹中

该文件所在文件目录('...\...\dsp')需用MATLAB中的set path(设置路径)菜单命令加载进来并保存好

% 在另一个脚本文件(.m文件)调用impseq函数

[x,n] = impseq(0,-10,20);

stem(n,x);

$$x(n) = \delta(n), \quad n \in [-10, 20]$$

[x,n] = impseq(2,0,10);

$$x(n) = \delta(n - 2), \quad n \in [0, 10]$$

例程(函数): 定义函数实现序列相加 $y(n)=x_1(n_1)+x_2(n_2)$ 。



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

```
% 确定n的范围
```

```
n = min(min(n1),min(n2)) : max(max(n1),max(n2)) ;
```

```
% y1和y2都具有最终的相同的长度n
```

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

```
y2 = y1 ;
```

```
% 将x1的值放入y1中, 位置保持好
```

```
y1(find((n>=min(n1)) & (n<=max(n1))))=x1 ;
```

```
% 将x2的值放入y2中, 位置保持好
```

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

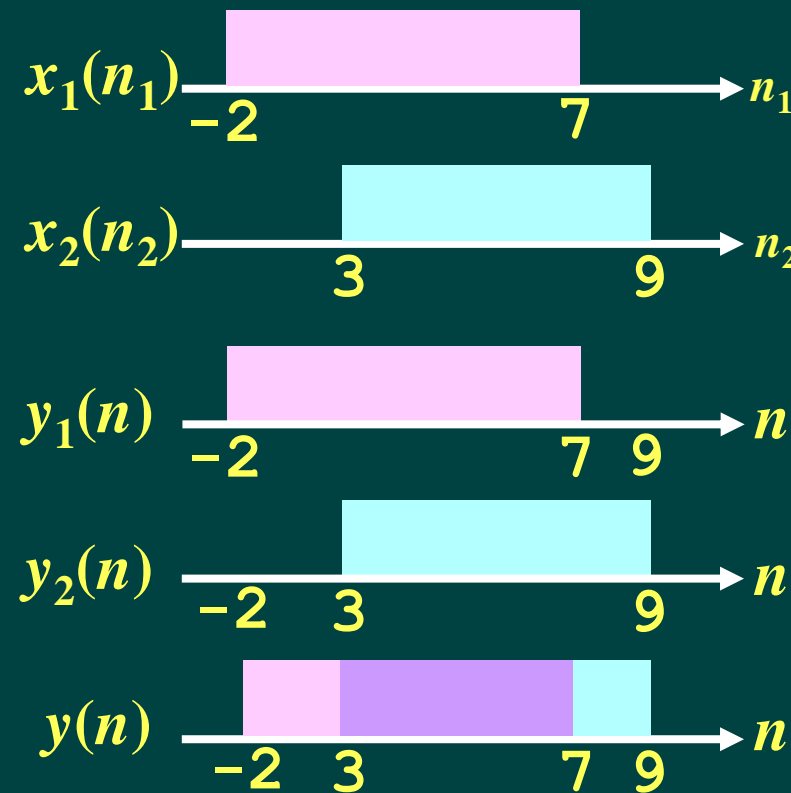
```
% 等长的序列y1和y2相加
```

```
y = y1+y2 ;
```

$$n_1 \in [-2, 7]$$

$$n_2 \in [3, 9]$$

$$n \in [-2, 9]$$



例程(函数): 定义函数实现序列相乘 $y(n)=x_1(n)\cdot x_2(n)$ 。



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

```
% 确定n的范围
```

```
n = min(min(n1),min(n2)) : max(max(n1),max(n2)) ;
```

```
% y1和y2都具有最终的相同的长度n
```

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

```
y2 = y1 ;
```

```
% 将x1的值放入y1中, 位置保持好
```

```
y1(find((n>=min(n1)) & (n<=max(n1))))=x1 ;
```

```
% 将x2的值放入y2中, 位置保持好
```

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

```
% 等长的序列y1和y2相乘
```

```
y = y1.*y2 ;
```

$$y1 = [1 \quad 2 \quad 3]$$

$$y2 = [1 \quad 2 \quad 3]$$

$$y1.*y2 = [1 \quad 4 \quad 9]$$

$$y1 * y2' = [1 \quad 2 \quad 3] \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 14$$

转置

例程：调用seqadd函数实现两序列相加 $y(n)=x_1(n)+x_2(n)$ 。



```
clear all; % 清除所有变量  
clc; % 清空命令窗口的显示记录
```

```
n1=[-2:3]; x1=[1 2 3 4 5 6];
```

$x_1(n)=\{1,2,3,4,5,6\}$

```
n2=[1:5]; x2=[5 4 3 2 1];
```

$x_2(n)=\{0,5,4,3,2,1\}$

```
[y,n]=seqadd(x1,n1,x2,n2);
```

序列求和

```
subplot(311); stem(n1,x1,'linewidth',2);
```

```
axis([min(n) max(n) 0 max(y)]); grid on;
```

```
subplot(312); stem(n2,x2,'linewidth',2);
```

```
axis([min(n) max(n) 0 max(y)]); grid on;
```

```
subplot(313); stem(n,y,'linewidth',2);
```

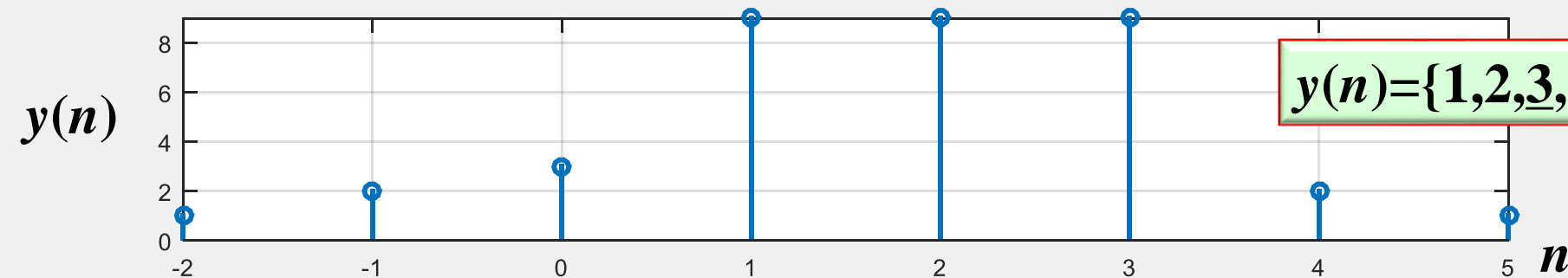
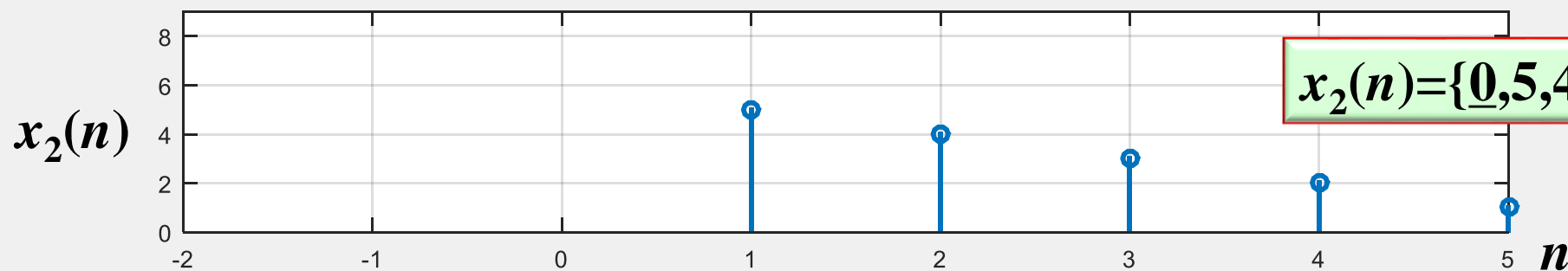
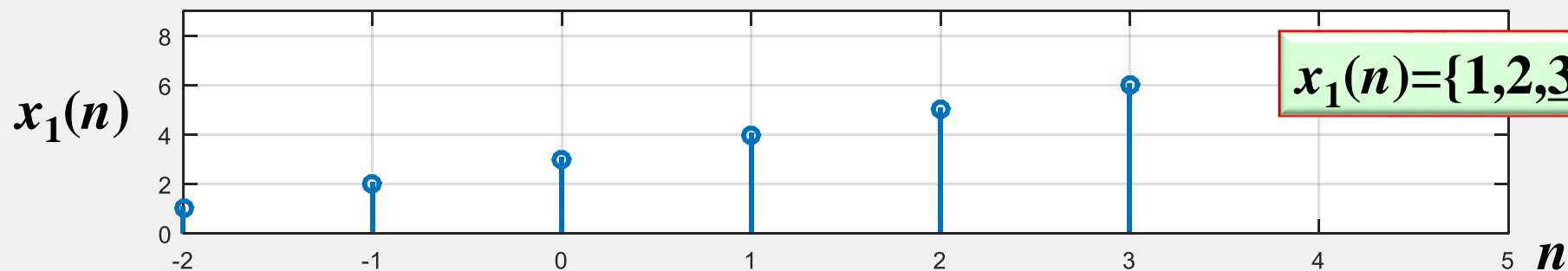
```
axis([min(n) max(n) 0 max(y)]); grid on;
```

调整画线宽度

统一显示范围



例程：调用seqadd函数实现两序列相加 $y(n)=x_1(n)+x_2(n)$ 。



例程(函数)： 定义函数实现序列移位 $y(n)=x(n-m)$ 。

```
function [y,ny] = seqshift(x,nx,m)
ny = nx+m; ny = 1:3
y = x;
y = [1,2,3]
```

$x = [1,2,3]$ $nx = 0:2$

$m = 1$

$x(n) = \{1,2,3\}$

$x(n-1) = \{0,1,2,3\}$

例程(函数)： 定义函数实现序列反褶 $y(n)=x(-n)$ 。

$x = [1, 2, 3]$ $nx = 0:2$

```
function [y,n] = seqfold(x,n)
```

```
y = fliplr(x);  $y = [3, 2, 1]$   $n = -2:0$ 
```

```
n = -fliplr(n);
```

$x(n) = \{1, 2, 3\}$

$x(-n) = \{3, 2, 1\}$

备注：MATLAB中的反褶函数： `fliplr`和`flipud`

行向量反褶函数： `fliplr(x)`

列向量反褶函数： `flipud(x)`

音频读取和写入文件

MATLAB自带的音频读取函数: *wavread* / *audioread*

```
[x,fs] = wavread(filename); [x,fs] = audioread(filename);
```

解释: **filename**是一个字符串, 给出 (**.wav**) 音频文件名, 也可以带指定路径。

返回值**x**是音频值, 如果是**单通道**音频, 则**x**是一个 **$n \times 1$** 的向量。

如果是**双通道**音频, 则**x**是一个 **$n \times 2$** 的向量。

返回值**fs**是音频**采样频率**。

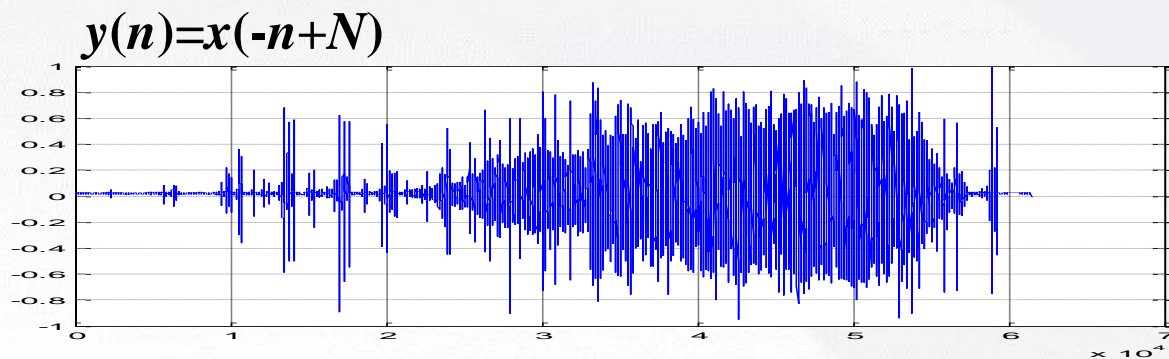
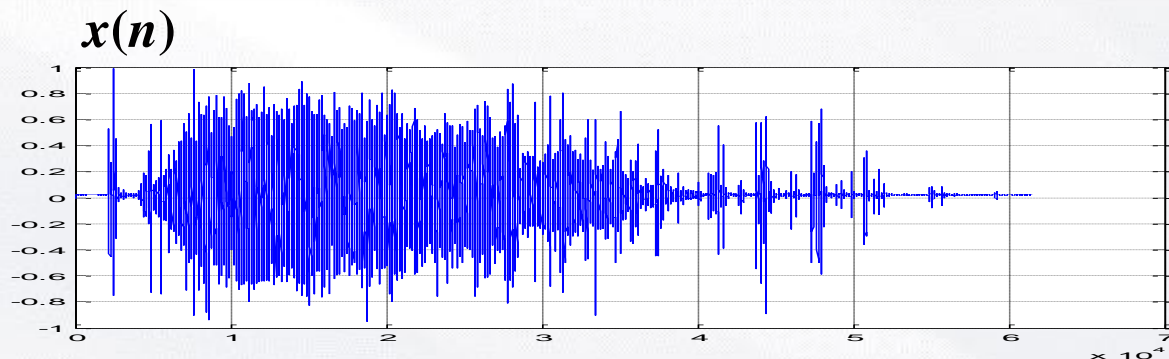
MATLAB自带的音频写入函数: *wavwrite* / *audiowrite*

```
wavwrite(y,fs,filename); audiowrite(filename,y,fs);
```

解释: **filename**是一个字符串, 给出 (**.wav**) 音频文件名, 也可以带指定路径。

y是要存入的音频值, **fs**是音频**采样频率**。

```
[x,fs] = wavread('w2.wav'); %读入单声道音频文件  
y = flipud(x); %反褶  
figure(1); plot(x); grid on; %画图显示结果  
figure(2); plot(y); grid on;  
wavwrite(y,fs,'w4.wav'); %结果保存为声音文件
```



原声



反褶效果



“我爱你”反褶后…?

人生感悟——

爱要勇于付出，
更要懂得珍惜！



MATLAB下序列运算的实现方法

- MATLAB编程实现**单位抽样序列** $\delta(n)$
- 序列**求和、求积**的编程方法
- 序列**移位、反褶**的编程方法