# 信号估计MATLAB编程

例2.1：在白噪声中的确定信号估计。数据模型和估计子为：

$$x[n] = As[n] + w[n]$$

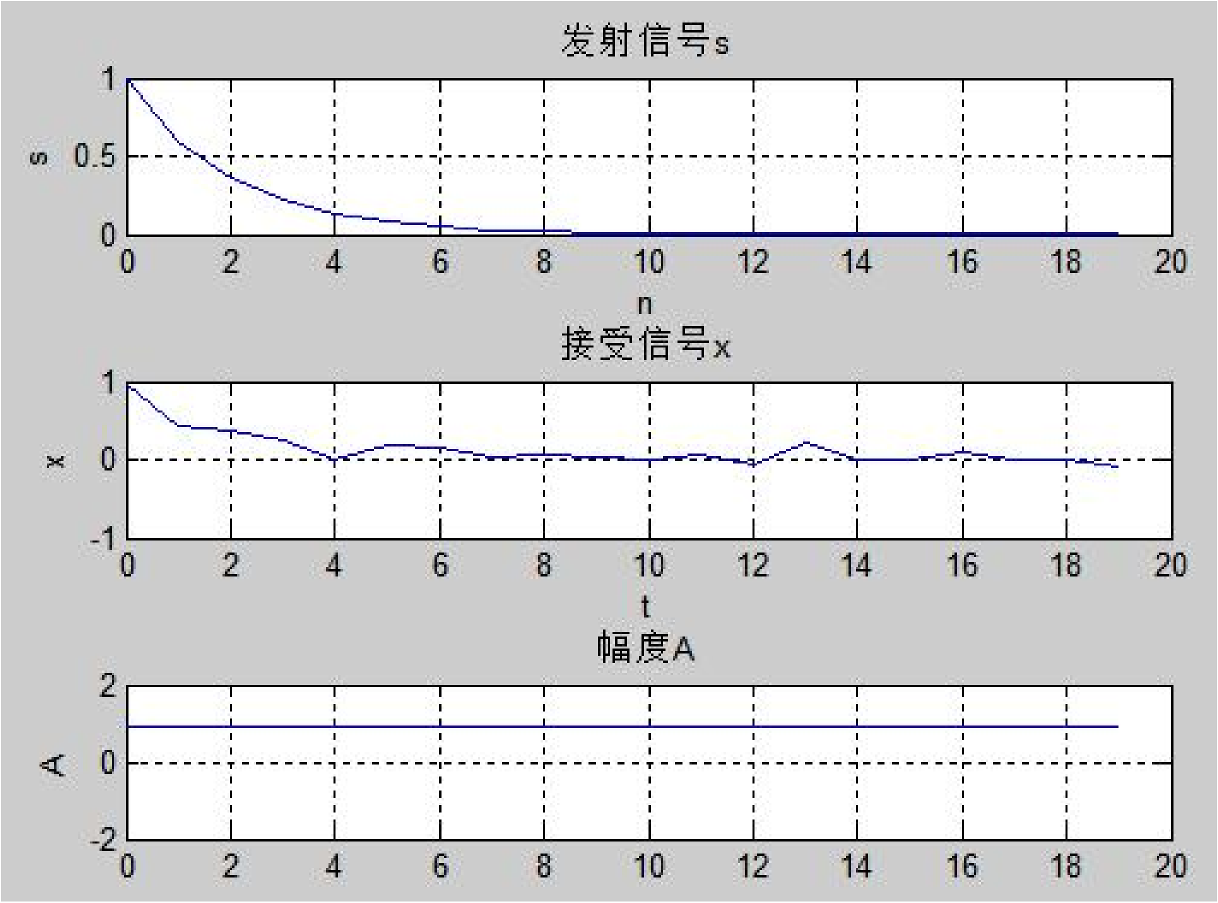
$$\hat{A} = \frac{\sum_{n=0}^{N-1} x[n]s[n]}{\sum_{n=0}^{N-1} s^2[n]}$$

$s[n] = r^n$，$A = 1$，$r = 0.9$，$N = 20$，$\sigma^2 = 0.04$

```
% This program estimates the amplitude of a damped exponential  signal
% embedded in WGN.
%
clear all
close all
randn('state',0) % initialize Gaussian random number generator
A=1; % set up signal parameters
r=0.6;
N=20;
n=[0:N-1]'; % set up time samples
s=A*r.^n; % compute signal samples
x=s+0.04*randn(N,1);  % add noise
Ahat=FSSP3alg9_1_sub(s,x)  % call estimation subprogram
A1=ones(1,N);
A2=A1.*Ahat;
subplot(311);
plot(n,s);
xlabel('n');ylabel('s');
title(‘发射信号s');grid;
subplot(312);
plot(n,x);xlabel('t');ylabel('x');
```

```
title('接收信号x');grid;
subplot(313);
plot(n,A2);ylabel('A');
title('幅度A');grid;

%  This function subprogram estimates the amplitude of a known
     signal
%  embedded in WGN.
%
%  Input parameters:
%    s  - real array of dimension Nx1 containing signal samples
%    x  - real array of dimension Lx1 containing data
%
%  Output parameters:
%    Ahat  - estimate of signal amplitude
%
function Ahat=FSSP3alg9_1_sub(s,x)
Ahat=(x'*s)/(s'*s);
```




**A=0.9663**

**例2.2：**高斯白噪声中正弦信号的幅度和相位估计。

数据模型和估计子为：

$$x[n] = A\cos[2\pi f_0 n + \phi] + w[n]$$

$$\hat{A} = \sqrt{\hat{\alpha}_1^2 + \hat{\alpha}_2^2}$$

$$\hat{\phi} = \arctan\left(\frac{-\hat{\alpha}_2}{\hat{\alpha}_1}\right)$$

$$\begin{bmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_2 \end{bmatrix} = \left(H^T H\right)^{-1} H^T x$$

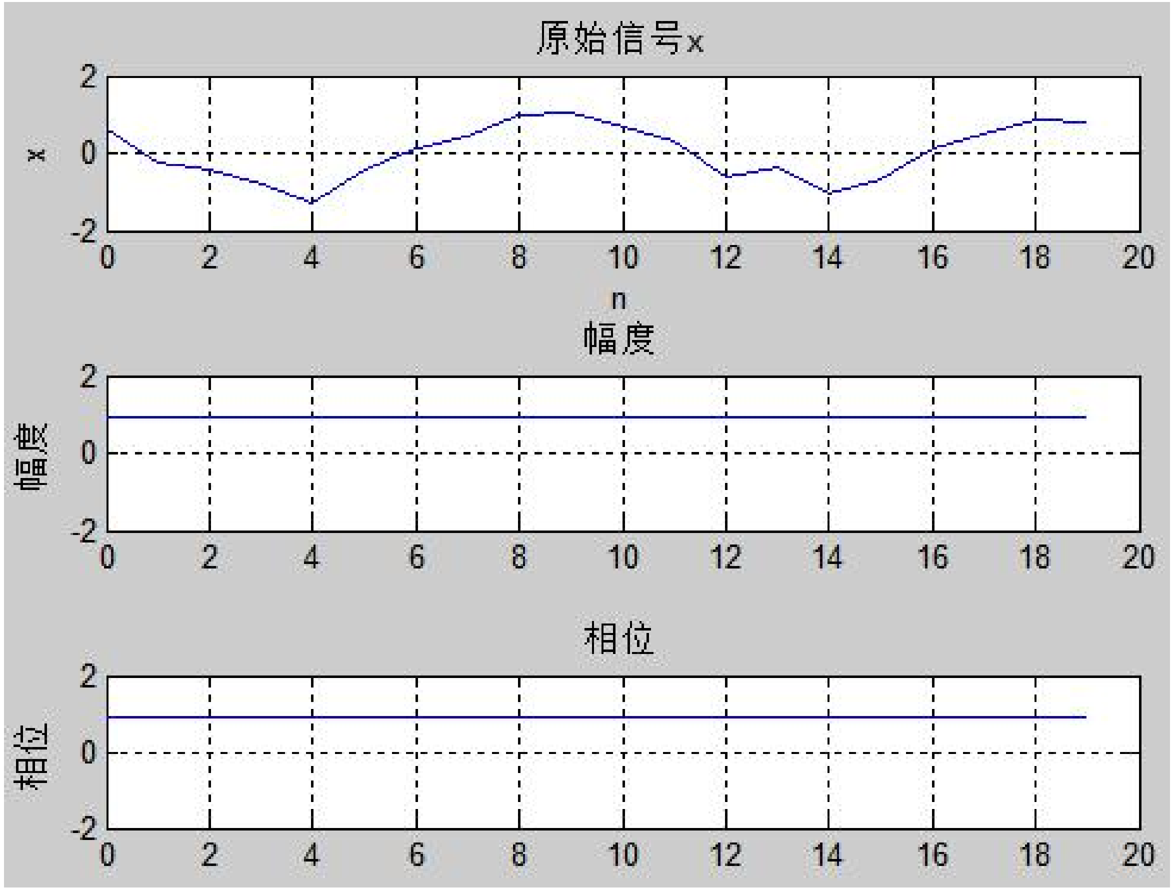
$$H = \begin{bmatrix} 1 & 0 \\ \cos 2\pi f_0 & \sin 2\pi f_0 \\ \vdots & \vdots \\ \cos[2\pi f_0 (N-1)] & \sin[2\pi f_0 (N-1)] \end{bmatrix}$$

$$A = 1, \; f_0 = 0.1, \; \phi = \pi/4, \; N = 20, \; \sigma^2 = 1/16$$

```
% This program estimates the amplitude and phase of a sinusoid of known
% frequency embedded in WGN.  Algorithm 9.2 is used.
%
clear all
close all
randn('state',0) % initialize Gaussian random number generator
A=1; % set up sinusoid parameters
f0=0.1;
phi=pi/4;
N=20;
n=[0:N-1]'; % set time samples
s=A*cos(2*pi*f0*n+phi); % compute signal
x=s+sqrt(1/16)*randn(N,1); % add noise with variance 10/16
[Ahat,phihat]=FSSP3alg9_2_sub(x,f0) % run subprogram
A1=ones(1,N);
A2=A1.*Ahat;
Phi=A1.*phihat;
```

```
subplot(311);
plot(n,x);
xlabel('n');ylabel('x');
title(‘原始信号x');grid;
subplot(312);
plot(n,A2);
ylabel(‘幅度');
title(‘幅度');grid;
subplot(313);
plot(n,Phi);
ylabel(‘相位');
title(‘相位');grid;
```

```
%  This function subprogram estimates the amplitude and phase of a
%  sinusoid with known frequency embedded in WGN, using the MLD.   It is
%  assumed that the sinusoid is of the form A*cos(2*pi*f0*n+phi)
%  for n=0,1,...,N-1.
%
%  Input parameters:
%    x  - real array of dimension Nx1 containing data
%    f0 - sinusoidal frequency
%
%  Output parameters:
%    Ahat  - estimate of amplitude
%    phihat - estimate of phase in radians
%
function [Ahat,phihat]=FSSP3alg9_2_sub(x,f0)
N=length(x);
n=[0:N-1]';
arg=2*pi*f0*n;
H=[cos(arg) sin(arg)]; % construct H matrix
alphahat=inv(H'*H)*H'*x;
Ahat=sqrt(alphahat(1)^2+alphahat(2)^2);
phihat=atan2(-alphahat(2),alphahat(1));
```




**A=0.9295，PHI=0.8887**

**例2.3：**从高斯白噪声中提取随机信号。

数据模型和估计子（维纳滤波器）为：

$$x[n] = s[n] + w[n]$$

$$\hat{s} = C_s\left(C_s + \sigma^2 I\right)^{-1} x$$

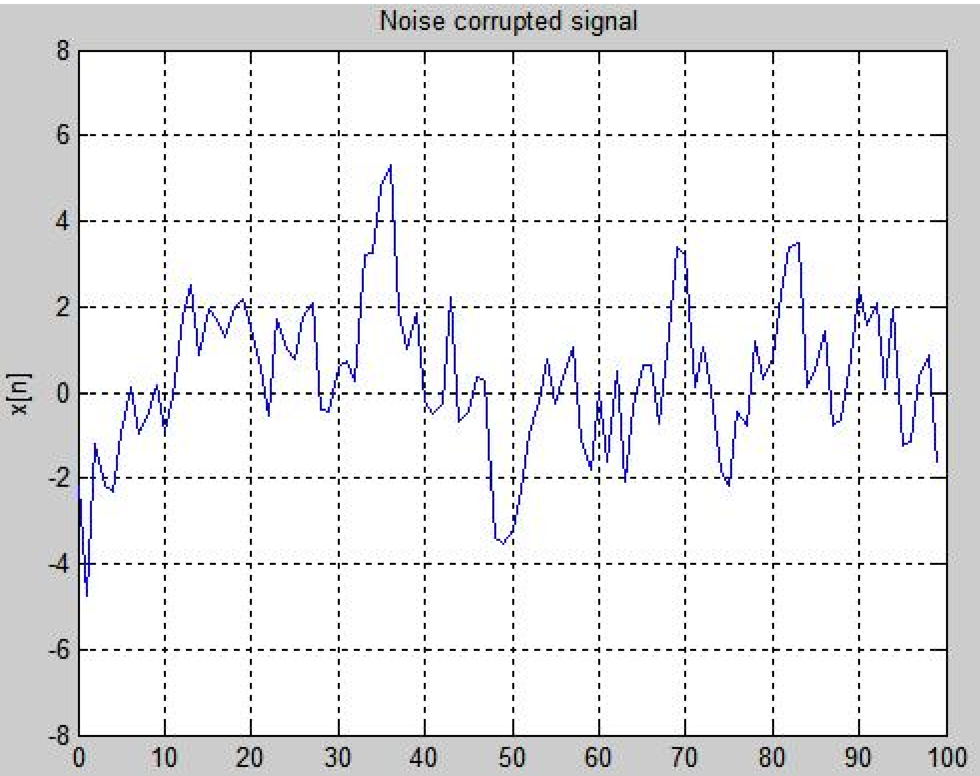
$$C_s = R_s$$

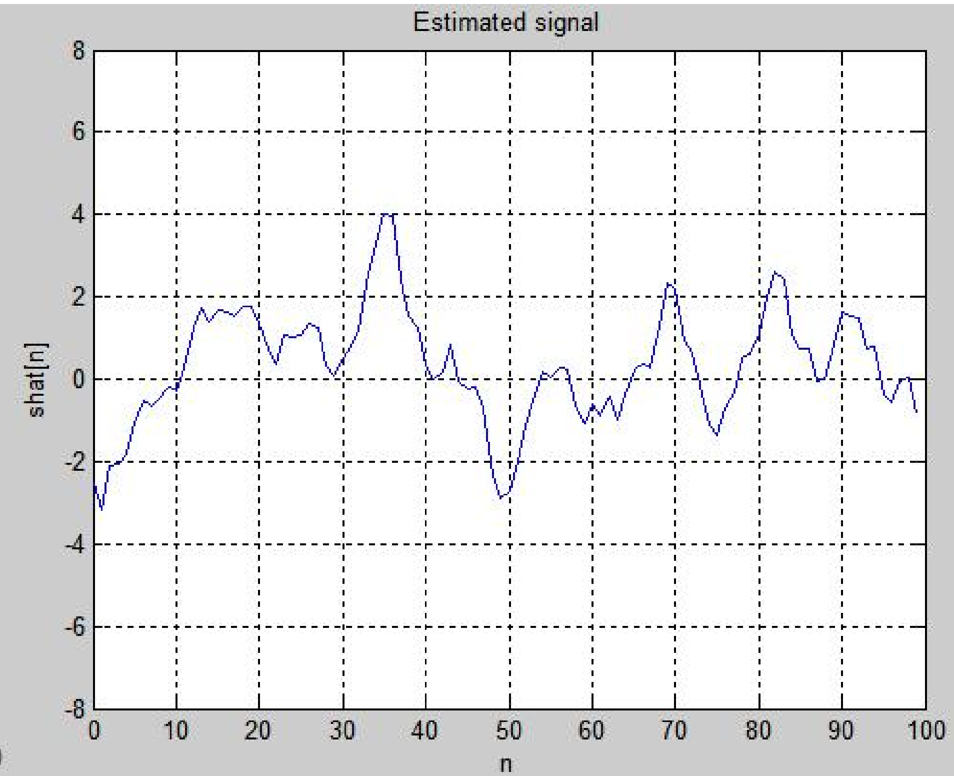
一阶高斯$AR$随机过程：$a[1] = -0.9$，$\sigma_u^2 = 1$,高斯白噪声$\sigma^2 = 1$

```
%  This program estimates the outcome of a random signal embedded in WGN
%  using a matrix Wiener smoother. See (9.21).  It assumes a Gaussian AR
%  random process which is stationary and so replaces the covariance
%  matrix by an autocorrelation matrix.
%
close all
clear all
randn('state',0);  % initialize Gaussian random number generator
a=-0.9;  % set up signal parameters
sig2u=1;
sig2=1;  % additive white noise variance
N=100;
n=[0:N-1]';  % set up time samples
s=ARgendata(a,sig2u,N);  % generate AR signal data
x=s+sqrt(sig2)*randn(N,1); %  add noise
figure
plot(n,s) % plot the true signal
xlabel('n')
ylabel('s[n]')
title('True signal')
grid
```

```
axis([0 N -8 8]);
figure
plot(n,x) % plot the noise corrupted signal
xlabel('n')
ylabel('x[n]')
title('Noise corrupted signal')
grid
axis([0 N -8 8]);
r0=sig2u/(1-a^2);  % compute r_s[0]
rs=r0*(-a).^[0:N-1]';  % compute r_s[0],r_s[1],...,r_s[N-1]
shat=FSSP3alg9_12_sub(x,rs,sig2);  % call subprogram to estimate the outcome
                                   % of the AR signal process
figure
plot(n,shat)  % plot estimated signal
xlabel('n')
ylabel('shat[n]')
title('Estimated signal')
grid
axis([0 N -8 8]);
```

```
%  This program estimates the outcome of a random signal embedded in white
%  noise using a Wiener smoother implemented in matrix form.
%
%  Input parameters:
%    x  - real array of dimension N x 1 containing the data samples
%    rs  - real array of dimension N x 1 containing the signal
%          autocorrelation lags arranged as [rs[0] rs[1] ... rs[N-1]]^T
%    sig2  - white noise variance
%
%  Output parameters:
%    shat - estimate of the signal
%
function shat=FSSP3alg9_12_sub(x,rs,sig2)
N=length(x);
for i=1:N
    for j=1:N
        Rs(i,j)=rs(abs(i-j)+1); % construct N x N signal autocorrelation
                                % matrix
    end
end
shat=Rs*inv(Rs+sig2*eye(N))*x; % compute MMSE estimate of s[n]
```


Noise corrupted signal
x[n]
8
6
4
2
0
-2
-4
-6
-8
0
10
20
30
40
50
60
70
80
90
100


Estimated signal
shat[n]
8
6
4
2
0
-2
-4
-6
-8
0
10
20
30
40
50
60
70
80
90
100
n

# 放映结束