

LAMPIRAN

```
%-----  
clear;  
close all;  
clc;  
%-----  
% N=input ('Please enter the number of iterations = ');  
N=5000;  
H = zeros(5,N);  
sig = zeros(5,5*N);  
K = zeros(5,N);  
XX = zeros(5,N);  
y = zeros(1,N);  
v = 0.1*randn(1,N);  
  
figure;clf;%brings up blank figure.  
plot (v);  
title ('Plot of v(k)');  
xlabel('Number of Iterations');  
ylabel('v(k)');  
grid on; zoom on;  
%Start of estimated Filter  
Q = 0.0001*eye(5,5); R = 0.1;  
y = filter(1,[1,-0.8,0.2,-0.6,0.7,-0.4],v);  
sig(1:5,1:5) = 0.1*eye(5);  
  
% "For" Loop for estimate filter:  
for k=6:N  
H(1:5,k)=[y(k-1);y(k-2);y(k-3);y(k-4);y(k-5)];  
K(1:5,k)=sig(1:5,5*k-29:5*k-25)*H(1:5,k)*inv(H(1:5,k))*sig(1:5,5*k-29:5*k-...
```

```

25)*H(1:5,k)+R);% Kalman Gain
sig(1:5,5*k-24:5*k-20) = sig(1:5,5*k-29:5*k-25)-sig(1:5,5*k-29:5*k-...
25)*H(1:5,k)*inv(H(1:5,k)*sig(1:5,5*k-29:5*k-25)*H(1:5,k) +...
R)*(H(1:5,k)*sig(1:5,5*k-29:5*k-25))+Q; % error convariance matrix
XX(1:5,k) = (eye(5)-K(1:5,k)*H(1:5,k))*XX(1:5,k-1) + (K(1:5,k)*y(k)); %
posteriori value of estimate X(k)
end;

```

```

figure;clf;%brings up blank figure.
plot (XX(1,:), 'g');
title ('Estimated Filter');
xlabel('Number of Iterations');
ylabel('XX(1,k)');
grid on; zoom on;

```

```

figure;clf;%brings up blank figure.
plot (XX(2,:), 'r');
title ('Estimated Filter');
xlabel('Number of Iterations');
ylabel('XX(2,k)');
grid on; zoom on;

```

```

figure;clf;%brings up blank figure.
plot (XX(3,:), 'b');
title ('Estimated Filter');
xlabel('Number of Iterations');
ylabel('XX(3,k)');
grid on; zoom on;

```

```

figure;clf;%brings up blank figure.
plot (XX(4,:), 'k');
title ('Estimated Filter');

```

```
xlabel('Number of Iterations');  
ylabel('XX(4,k)');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.  
plot (XX(5,:), 'm');  
title ('Estimated Filter');  
xlabel('Number of Iterations');  
ylabel('XX(5,k)');  
grid on; zoom on;
```

```

%-----
% N=input('Please enter the number of iterations = ');
N=10000;
HH = zeros(5,N); % matrix of output signal yy.
sig = zeros(5,5*N); % priori or posteri covariance matrix.
K = zeros(5,N); % kalman gain.
X = zeros(5,N); % kalman coefficient for y.
XX = zeros(5,N); % kalman coefficient for yy.
I = eye(5);
y = zeros(1,N); % Input signal.
ZZ = zeros(1,N);
yy = zeros(1,N); % Output yy signal.
v = 0.1*randn(1,N); % white noise.

figure(1);clf;%brings up blank figure.
plot (v);
title ('Plot of v(k)');
xlabel('Number of Iterations');
ylabel('v(k)');
grid on; zoom on;

%Start of estimated Filter
Q = 0.0001*eye(5,5); % Process Noise Covariance.
R = 0.1; % Measurement Noise Covariance

A1=[1,-0.7,0.3,-0.6,0.7,-0.5];
A2=[1,-0.4,0.5,-0.2,0.5,-0.2];
A3=[1,-0.5,0.3,-0.5,0.4,-0.6];
A4=[1,-0.6,0.5,-0.4,0.2,-0.4];
A5=[1,-0.4,0.6,-0.1,0.3,-0.2];
y(1,1:2000) = filter(1,A1,v(1,1:2000));
y(1,2001:4000) = filter(1,A2,v(1,2001:4000));

```

```

y(1,4001:6000) = filter(1,A3,v(1,4001:6000));
y(1,6001:8000) = filter(1,A4,v(1,6001:8000));
y(1,8001:10000) = filter(1,A5,v(1,8001:10000));

sig(1:5,1:5) = 0.5*I;

% "For" Loop for estimate filter:
for k=6:N
H(1:5,k)=[y(k-1);y(k-2);y(k-3);y(k-4);y(k-5)];
HH(1:5,k)=[yy(k-1);yy(k-2);yy(k-3);yy(k-4);yy(k-5)];

K(1:5,k)=sig(1:5,5*k-29:5*k-25)*H(1:5,k)*inv(H(1:5,k)*sig(1:5,5*k-29:5*...
k-25)*H(1:5,k)+R);% Kalman Gain

sig(1:5,5*k-24:5*k-20)=sig(1:5,5*k-29:5*k-25)-sig(1:5,5*k-29:5*k-25)...
*H(1:5,k)*inv(H(1:5,k)*sig(1:5,5*k-29:5*k-25)*H(1:5,k) +...
R)*(H(1:5,k)*sig(1:5,5*k-29:5*k-25))+Q; % error covariance matrix

XX(1:5,k) = (I - K(1:5,k)*H(1:5,k)')*XX(1:5,k-1) + ...
(K(1:5,k)*y(k)); % posteriori value of estimate X(k)

vv(k) = y(k) - (H(1:5,k)'*XX(1:5,k));
yy(k) = (HH(1:5,k)*XX(1:5,k)) + vv(k); %
end;

figure;clf;%brings up blank figure.
plot (XX(1,:), 'g');
title ('Estimated Filter 1');
xlabel('Number of Iterations');
ylabel('XX(1,k)');
grid on; zoom on;

```

```
figure;clf;%brings up blank figure.  
plot (XX(2,:), 'r');  
title ('Estimated Filter 2');  
xlabel('Number of Iterations');  
ylabel('XX(2,k)');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.  
plot (XX(3,:), 'b');  
title ('Estimated Filter 3');  
xlabel('Number of Iterations');  
ylabel('XX(3,k)');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.  
plot (XX(4,:), 'k');  
title ('Estimated Filter 4');  
xlabel('Number of Iterations');  
ylabel('XX(4,k)');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.  
plot (XX(5,:), 'm');  
title ('Estimated Filter 5');  
xlabel('Number of Iterations');  
ylabel('XX(5,k)');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.  
plot (y(:));  
title ('y Signal');  
xlabel('Number of Iterations');
```

```
ylabel('y');  
grid on; zoom on;  
  
figure;clf;%brings up blank figure.  
plot (yy(:));  
title ('yy Signal');  
xlabel('Number of Iterations');  
ylabel('Output yy');  
grid on; zoom on;
```

```

%-----%
%Speech sample
%-----%
% N=input ('Please enter the number of iterations = ');
N=5000;
H = zeros(5,N); % matrix of output signal y.
HH = zeros(5,N); % matrix of output signal yy.
sig = zeros(5,5*N); % priori or posteri covariance matrix.
K = zeros(5,N); % kalman gain.
XX = zeros(5,N); % kalman coefficient for yy.
I = eye(5);
y = zeros(1,N); % Input signal.
vv = zeros(1,N); % estimated white noise.
yy = zeros(1,N); % reconstructed output signal.

%Start of estimated Filter
Q = 0.0001*eye(5,5); % Process Noise Covariance.
R = 0.1; % Measurement Noise Covariance
a=wavread('1.wav');%Sinyal Input
y=a(1:N); % y is the output signal produced.
sig(1:5,1:5) = 0.1*I;

% "For" Loop for estimate filter:
for k=6:N
H(1:5,k)=-[y(k-1);y(k-2);y(k-3);y(k-4);y(k-5)];
HH(1:5,k)=-[yy(k-1);yy(k-2);yy(k-3);yy(k-4);yy(k-5)];
K(1:5,k)=sig(1:5,5*k-29:5*k-25)*H(1:5,k)*inv(H(1:5,k)'*sig(1:5,5*k-29:5*...
k-25)*H(1:5,k)+R);%Kalman Gain

sig(1:5,5*k-24:5*k-20)=sig(1:5,5*k-29:5*k-25)-sig(1:5,5*k-29:5*k-...
25)*H(1:5,k)*inv(H(1:5,k)'*sig(1:5,5*k-29:5*k-25)*H(1:5,k) +...
R)*(H(1:5,k)'*sig(1:5,5*k-29:5*k-25))+Q; % error covariance matrix

```



```

XX(1:5,k) = (I - K(1:5,k)*H(1:5,k))*XX(1:5,k-1) +...
(K(1:5,k)*y(k)); % posteriori value of estimate X(k)

vv(k) = y(k) - (H(1:5,k)*XX(1:5,k)); % White Noise
yy(k) = (HH(1:5,k)*XX(1:5,k)) + vv(k); % Sinyal output
end;
% cross covariance
c = xcorr(y,yy,'coeff');

figure;clf;% brings up blank figure.
plot (vv(:));
title ('Sinyal Noise');
xlabel('Number of Iterations');
ylabel('VV');
grid on; zoom on;

figure;clf;% brings up blank figure.
plot (XX(1,:),'g');
title ('Estimated Filter 1');
xlabel('Number of Iterations');
ylabel('XX(1,k)');
grid on; zoom on;

figure;clf;% brings up blank figure.
plot (XX(2,:),'r');
title ('Estimated Filter 2');
xlabel('Number of Iterations');
ylabel('XX(2,k)');
grid on; zoom on;

figure;clf;% brings up blank figure.

```

```
plot (XX(3,:), 'b');  
title ('Estimated Filter 3');  
xlabel('Number of Iterations');  
ylabel('XX(3,k)');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.
```

```
plot (XX(4,:), 'k');  
title ('Estimated Filter 4');  
xlabel('Number of Iterations');  
ylabel('XX(4,k)');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.
```

```
plot (XX(5,:), 'm');  
title ('Estimated Filter 5');  
xlabel('Number of Iterations');  
ylabel('XX(5,k)');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.
```

```
plot (y(:));  
title ('Input Sinyal Suara');  
xlabel('Number of Iterations');  
ylabel('Input y');  
grid on; zoom on;
```

```
figure;clf;%brings up blank figure.
```

```
plot (yy(:));  
title ('Sinyal Output Suara');  
xlabel('Number of Iterations');  
ylabel('Output yy');
```

```

grid on; zoom on;

figure;clf;%brings up blank figure.
plot (c);
title ('Cross Correlation of signal y & yy');
xlabel('i');
ylabel('Correlation');
grid on; zoom on;

%-----
%SNR
%-----
length(a)
input=y';
output =yy;

%SNR secara langsung
daya_input1=sum(input(1:N).^2);
daya_error1=sum((input(1:N)-output(1:N)).^2);
SNR=10*log10(daya_input1-daya_error1)

```