



Department of
Electrical and Information Technology
Lunds Universitet – LTH

Optimal Signal Processing

Computer exercises 1 and 2

Leif Sörnmo
Martin Stridh
2012

1 Computer exercises 1: Signal modeling – inverse filtering

Purpose To increase the knowledge of designing filters using least squares methods.

Preparation Read Chapter 4, specially 4.4.5 about designing FIR inverse filters using least squares methods. This exercise use the m-file `spike.m` in the book by Hayes.

Background We use an example of inverse filtering. The observed signal is denoted $y(n)$. This signal consist of an impulse response $g(n)$.

$$g(n) = \begin{cases} \cos(0.2[n - 25]) \exp\{-0.01[n - 25]^2\} & 0 \leq n \leq 50 \\ 0 & \text{för övrigt} \end{cases}$$

convolved by a pulse train,

$$x(n) = \sum_{k=1}^{10} x(k)\delta(n - n_k)$$

The information included in $y(n)$ is amplitude and delay of spikes in the pulse train. We will here estimate this values. The correct values is given by

n_k	25	40	55	65	85	95	110	130	140	155
$x(k)$	1	0.8	0.7	0.5	0.7	0.2	0.9	0.5	0.6	0.3

Problem 1.1 Plot the observed signal $y(n) = g(n) * x(n)$ and estimate the amplitude and delay in $x(n)$ by looking at the peeks in $y(n)$.

```
% create the impulse response - g(n)
n = (0:50) - 25;
g = cos(0.2*n) .* exp(-0.01*n.^2);

% create the impulse sequense - x(n)
nk = [25 40 55 65 85 95 110 130 140 155]';
a = [1 0.8 0.7 0.5 0.7 0.2 0.9 0.5 0.6 0.3]';
x = zeros(200,1);
x(nk) = a .* ones(length(nk),1);

% determine the output - y(n)
y = conv(x,g);
plot(x), hold on, plot(y,'r')
axis([0 300 -0.5 1])
```

Problem 1.2 Design a least squares (LS) inverse filter of length $N = 50$ with the command `spike.m`.

```
% design a Least Squares inverse filter - choice of optimal filter delay
N = 50;
for n0 = 1:N
```

```

    [h(:,n0),err(n0)] = spike(g,n0,N);
end

% choose the filter corresponding to the minimum error
[tmp,j] = min(err);
h = h(:,j);

```

Problem 1.3 Plot the frequency function for $g(n)$ and $h(n)$, respective and compare the curves. Conclusions?

```

subplot(211), plot(abs(fft(g)))
subplot(212), plot(abs(fft(h)))

```

Problem 1.4 Compute the output $y(n)$ from the inverse optimal filter in 1.2 and plot the output from the filter, i.e. $\hat{x}(n) = h_N(n) * y(n)$. Estimate the amplitude and delay of the peaks in the signal.

```

% inverse filtering
xhat = conv(y,h);

% plot observed and filtered signal, respectively
figure
plot(j+1:j+length(x),x,':'),hold
plot(xhat,'g')
axis([0 300 -0.5 1])

```

Problem 1.5 The results in 1.4 assume the the observed signal is not disturbed by noise. Assume now that the observed signal is disturbed by noise

$$\tilde{y}(n) = g(n) * x(n) + v(n)$$

with $v(n)$ white gaussian noise with mean value equal to zero and variance σ_v^2 . Repeat problem 1.4 for $\tilde{y}(n)$ for various σ_v^2 , for example $\sigma_v^2 = 0.00001$ and $\sigma_v^2 = 0.001$. Comment the accuracy of the estimates (delay and amplitude).

```

% choose noise level
sigma_v2 = ...

% generate a noisy signal ("randn" will give Gaussian random variables N(0,1))
y = conv(x,g);
y = y + randn(size(y)) * sqrt(sigma_v2);

% determine the inverse filter for various delay
for n0 = 1:N
    [h(:,n0), err(n0)] = spike(g,n0,N);
end

```

```

[tmp,j] = min(err);
h = h(:,j);
xhat = conv(y,h);

% plot the observed and the signal after the inverse filter, respectively
figure
plot(j+1:j+length(x),x,':'),hold
plot(xhat,'g')
axis([0 300 -0.5 1])

```

Problem 1.6 Write a m-file `spike2.m` that implement the modified inverse filter in Problem 4.19 in the book by Hayes. (Modify `spike.m` and save as `spike2.m`).

Add the number of input to also include the design parameter α . Test the filter using various level of the noise (also without noise).

```

% choose the noise level ( 0 or >0)
sigma_v2 = ...

% choose the design parameter
alpha = 0.8;

% generate the signal
y = conv(x,g);
y = y + randn(size(y)) * sqrt(sigma_v2);

% determine the modified inverse filter using the function "spike2".

for n0 = 1:N
    [h(:,n0),err(n0)] = spike2(g,n0,N,alpha);
end
[tmp,j] = min(err);
h = h(:,j);
xhat = conv(y,h);

% plot the observed and signal after the inverse filter, respectively
figure
plot(j+1:j+length(x),x,':'),hold
plot(xhat,'g')
axis([0 300 -0.5 1])

```

2 Computer exercises 2: Levinson-Durbin recursion and Burgs algorithm

This exercises

This exercises we will estimate the parameters in an all-pole model of the signal.

An second order all-pole AR process $x(n)$ is given by

$$x(n) = -a_1x(n-1) - a_2x(n-2) + v(n) \quad (1)$$

with a_1 and a_2 the AR coefficients and with the innovation process $v(n)$ (white noise with zero mean and the variance σ_v^2).

The values of the parameters are

$$a_1 = -1 \quad (2)$$

$$a_2 = 0.8 \quad (3)$$

$$\sigma_v^2 = 2.56 \quad (4)$$

Prolem 2.1: Plot pover spectrum, $S_x(z)$ for the AR-process $x(n)$ in (1).

We know that the power spectrum can be written as

$$S_x(z) = \sigma_v^2 |H(z)|^2 \quad (5)$$

with the system function $H(z)$ and the innovation process $v(n)$ to the AR-process $x(n)$.

This can be determined fram (1) which gives

$$H(z) = \frac{1}{1 + a_1z^{-1} + a_2z^{-2}} = \frac{1}{1 - z^{-1} + 0.8z^{-2}} \quad (6)$$

In MATLAB-code:

```
a=[1 -1 0.8];           % Vector representing the denominator
b=1;                   % Vector representing nominator
[Hz,f]=freqz(1,a,1024); % H(z) determined by the command freqz
Sx=2.56*abs(Hz).^2;    % Determine the power spectrum
```

Plot the functions using:

```
figure                 % Create a new figure
subplot(211)          % Upper level
plot(f,10*log10(Sx),'b'); % Plot with logarithmic scale in blue
```

Problem 2.2: Generate an 128 samples gaussian sequence, $v(n)$, with zero mean and variance $\sigma_v^2 = 2.56$ and the autocorrelation $r_v(k) = 0$ for all $k \neq 0$.

```
v=sqrt(2.56)*randn(128,1); % Variace 2.56 means amplitude sqrt(2.56)
```

Uppgift 2.3: Generate now an AR-process $x(n)$ using *filter*. Remove the first 100 samples (the transient signal) and use only the remaining 28 samples.

```
x=filter(b,a,v); % Filter v(n) with H(z) (the polynoms a och b)
x=x(101:128); % New x are the 28 last values of x
```

Suppose that the signal $x(n)$ above is an unknown AR-process. Estimate the parameters in this process.

Problem 2.4: We use Burgs algorithm in the book. Estimate the power spectrum of the AR-process using the order of the model 5 and 14, respectively (of course we know that the correct order is 2).

The function *burgs* estimates the reflection parameters but also determine the energy of the error signal in step in the Lattice filter. Also the Final Prediction Error (FPE) after each step in the Lattice filter are given.

```
[gamma5,P5,FPE5]=burgs(x,5); % Burgs algorithm for order 5
[gamma14,P14,FPE14]=burgs(x,14); % Burgs algorithm for order 14
```

From the reflection coefficients we can determine the AR-parameters (Levinson-Durbin step-up recursion equation 5.14 in Hayes).

```
a5=[1];
for m=1:5
    a5=[a5;0]+gamma5(m)*[0;a5(m:-1:1)]; % Levinson-Durbin step-up recursion in matrix form
end

a14=[1];
for m=1:14
    a14=[a14;0]+gamma14(m)*[0;a14(m:-1:1)]; % Levinson-Durbin step-up recursion in matrix form
end
```

Compute a plot the estimated power spectra

```
[H5,f]=freqz(1,a5,1024);
S5=P5(5)*abs(H5).^2;

[H14,f]=freqz(1,a14,1024);
S14=P14(14)*abs(H14).^2;
```

Plot these curves in the same figure using

```
hold on %
plot(f,10*log10(S5),'r',f,10*log10(S14),'g') %
title('Sx - blå, Sx_{5} - red, Sx_{14} - green') %
```

Problem 2.5: Determine optimal model order using the FPE criterion (page 445-447).

The function *burgs* determine FPE according to equation 8.126.

```
subplot(212) % Lower
plot(FPE14)
title('Final Prediction Error')
Mo=2; % Minimum for 2?
```

2 COMPUTER EXERCISES 2: LEVINSON-DURBIN RECURSION AND BURGS ALGORITHM7

Problem 2.6: Plot the power spectrum of the AR-process with optimal model order.

```
ao=[1];
for m=1:Mo
    ao=[ao;0]+gamma14(m)*[0;ao(m:-1:1)]; % Levinson-Durbin step-up recursion i matrix form
end
ao
```

Determine the estimated power spectrum

```
[Ho,f]=freqz(1,ao,1024);
So=P14(Mo)*abs(Ho).^2;
```

and plot i (upper)

```
subplot(211) hold on plot(f,10*log10(So),'y') title('Sx - blue,
Sx_{5} - red, Sx_{14} - green, Sx_{opt} - yellow')
```

Problem 2.7: Compare the estimated power spectra with the theoretical power spectra. Comments?