Institutionen för
Elektro- och Informationsteknik
Lunds Universitet – Lunds Tekniska Högskola

Optimal Signalbehandling
Datorövning 1 och 2

Leif Sörnmo
Martin Stridh
2012

Department of Electrical and Information Technology, Lund University, Sweden

1 Datorövning 1: Signalmodellering – inversfiltrering

Syfte Att öka förstelsen för hur man designar filter m.h.a. minsta-kvadrat metoden och att belysa de problem som är förknippade med design av inversfilter.

Förberedelse Innehållet i Kapitel 4 är en förutsättning för denna datorövnings genomförande; speciell tonvikt ges åt Kap. 4.4.5 har som handlar om design av minsta-kvadrat inversfilter med FIR struktur. Datorövningen bygger på m-filen `spike.m` vars kod återfinns i Figur 4.15. Det är utomordentligt viktigt att ha en förståelse för vad denna m-fil gör även om den råkar finnas tillgänglig i färdigpaketerat format.

Studera noggrant lösningen på problem 4.19. Detta problem kommer gås igenom på räkneövningen som föregår denna datorövning.

Bakgrund På föreläsning 2 beskrevs kortfattat en speciell typ av signalbehandling som används flitigt inom reflektionsseismologi – s.k. inversfiltrering. Denna typ av signalbehandling är ofta baserad på en **modell** för hur den observerade signalen $y(n)$ är relaterad till signalkällans och reflektionssekvensens utseende. Denna modell kan naturligtvis också ta hänsyn till ev. förekomst av mätbrus. Signalkällan beskrivs m.h.a. impulssvaret $g(n)$

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

och reflektionssekvensen modelleras som ett icke–ekvidistant pulsståg,

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

och innehåller den information, d.v.s. amplitud och tidsläge, som skall skattas från den observerade signalen. Värdena för n_k och $x(k)$ ges av följande tabell,

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

Uppgift 1.1 Plotta den observerade signalen $y(n) = g(n) * x(n)$ och uppskatta visuellt hur pass noga man kan bestämma amplituden och tidsläget för impulserna i $x(n)$ genom att studera topparna i $y(n)$.

```
% skapa impulssvar - g(n)
n = (0:50) - 25;
g = cos(0.2*n) .* exp(-0.01*n.^2);

% skapa impulssekvens - 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);

% beräkna utsignalen - y(n)
```

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

Uppgift 1.2 Designa ett minsta-kvadrat (MK) inversfilter med längden $N = 50$ genom att använda m-filen `spike.m` (Hayes sid 172, finns i labdatorerna och kan hämtas direkt från OSB:s hemsida). Välj den filterfördröjning som resulterar i minimalt fel.

```
% design av MK inversfilter - val av optimal filterfördröjning
N = 50;
for n0 = 1:N
    [h(:,n0),err(n0)] = spike(g,n0,N);
end

% välj det filter som svarar mot minimalt fel
[tmp,j] = min(err);
h = h(:,j);
```

Uppgift 1.3 Plotta frekvensfunktionen för $g(n)$ respektive $h(n)$ och jämför dessa kurvors utseende. Vilken slutsats kan dras redan på detta stadie, dvs innan den observerade signalen har filtrerats?

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

Uppgift 1.4 Inversfiltrera $y(n)$ med det optimala filtret från uppgift 1.2 och plotta utsignalen från filtret, dvs $\hat{x}(n) = h_N(n) * y(n)$. Skatta pulsernas amplitud och tidsläge från utsignalen.

```
% inversfiltering
xhat = conv(y,h);

% plotta observerad respektive inversfilterad signal
figure
plot(j+1:j+length(x),x,':'),hold
plot(xhat,'g')
axis([0 300 -0.5 1])
```

Uppgift 1.5 Resultaten i uppgift 1.4 utgår från att den observerade signalen är brusfri. Antag nu att dessa är observationer är brusiga,

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

där $v(n)$ är vitt, normalfördelat brus med mv noll och varians σ_v^2. Upprepa uppgift 1.4 för $\tilde{y}(n)$ för några olika brusnivåer, t.ex. $\sigma_v^2 = 0.00001$ och $\sigma_v^2 = 0.001$. Kommentera noggrannheten i skattningen av pulsernas tidsläge.

```
% välj brusnivå
sigma_v2 = ...

% generera brusig signal ("randn" ger normalfördelade tal N(0,1))
y = conv(x,g);
y = y + randn(size(y)) * sqrt(sigma_v2);

% bestäm inversfilter för olika fördröjningar
for n0 = 1:N
      [h(:,n0), err(n0)] = spike(g,n0,N);
end
[tmp,j] = min(err);
h = h(:,j);
xhat = conv(y,h);

% plotta observerad respektive inversfilterad signal
figure
plot(j+1:j+length(x),x,':'),hold
plot(xhat,'g')
axis([0 300 -0.5 1])
```

Uppgift 1.6 Skriv en m-fil `spike2.m` som implementerar det modifierade inversfilter framräknat i Problem 4.19 (Duplicera `spike.m` och spara som `spike2.m`). Utöka inargumenten till denna funktion så att den inkluderar designparametern α. Studera detta filters prestanda på signaler med olika brusnivå (även utan brus!).

```
% välj brusnivå (antingen 0 eller >0)
sigma_v2 = ...

% välj designparameter
alpha = 0.8;

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

% bestäm modifierat inversfilter m.h.a. funktionen "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);

% plotta observerad respektive inversfilterad signal
figure
plot(j+1:j+length(x),x,':'),hold
```

```
plot(xhat,'g')
axis([0 300 -0.5 1])
```

2 Datorövning 2: Levinson-Durbin rekursion och Burgs algoritm

Denna datorövning går ut på att från en given signal skatta en polmodell med lämplig modellordning. I de tre första uppgifterna skapas en okänd signal. I de följande uppgifterna försöker vi från denna signal skatta den modell som användes för att skapa signalen.

Betrakta följande andra ordningen AR process $x(n)$

$$x(n) = -a_1 x(n-1) - a_2 x(n-2) + v(n) \tag{1}$$

där a_1 och a_2 är AR koefficienter och innovationsprocessen $v(n)$ är vitt brus med medelvärde noll och varians σ_v^2. Parametrarna har följande värden

$$a_1 = -1 \tag{2}$$

$$a_2 = 0.8 \tag{3}$$

$$\sigma_v^2 = 2.56 \tag{4}$$

Uppgift 2.1: Plotta spektraltätheten, $S_x(z)$ för AR-processen $x(n)$ i (1).

Vi vet att spektraltätheten kan skrivas

$$S_x(z) = \sigma_v^2 |H(z)|^2 \tag{5}$$

där $H(z)$ är överföringsfunktionen från innovationsprocessen $v(n)$ till AR-processen $x(n)$. Denna beräknas lätt ur (1) och blir

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{1}{1 - z^{-1} + 0.8 z^{-2}} \tag{6}$$

I MATLAB-kod:

```
a=[1 -1 0.8];             % Vektor som representerar nämnaren
b=1;                      % Vektor som representerar täljaren
[Hz,f]=freqz(1,a,1024); % H(z) kan sedan beräknas med hjälp av kommandot freqz
Sx=2.56*abs(Hz).^2;       % Beräkning av spektraltätheten
```

Plotta funktionen med hjälp av dessa rader:

```
figure                        % Skapar ny figur
subplot(211)                  % Väljer övre halvan
plot(f,10*log10(Sx),'b');   % Plottar med logaritmisk skala i blå färg
```

Uppgift 2.2: Producera en 128 sampel lång normalfördelad sekvens, $v(n)$, med medelvärde noll, varians $\sigma_v^2 = 2.56$ och autokorrelation $r_v(k) = 0$ för alla $k \neq 0$.

```
v=sqrt(2.56)*randn(128,1);    % Varians 2.56 betyder amplitud sqrt(2.56)
```

Uppgift 2.3: Skapa nu AR-processen $x(n)$ med hjälp av funktionen *filter*. Plocka sedan bort de 100 första samplen vilket eliminerar transienterna som fås vid filtreringen (IIR filter). Använd alltså fortsättningsvis endast de kvarvarnde 28 samplen.

```
x=filter(b,a,v);    % Filtrera v(n) med H(z) som representeras av polynomen a och b
x=x(101:128);       % Nya x är de 28 sista samplen av gamla x
```

Nu antar vi att signalen $x(n)$ ovan är en okänd AR-process. Syftet med följande uppgifter är att ta reda på hur modellen ser ut (dvs modellordning och AR-koefficienter).

Uppgift 2.4: Hämta hem Burgs algoritm. Skatta sedan med algoritmens hjälp AR-processens effektspektrum för de två gissade modellordningarna 5 och 14 (Vi vet egentligen att modellordningen är 2). Funktionen *burgs* är skriven så att den baserat på signalen $x(n)$ skattar ett givet antal reflexionskoefficienter. Dessutom ger den felenergin P i varje Lattice-steg. Baserat på felet i varje steg beräknas också felkriteriet Final Prediction Error (FPE) för varje Lattice-steg.

```
[gamma5,P5,FPE5]=burgs(x,5);      % Burgs algoritm för ordning 5
[gamma14,P14,FPE14]=burgs(x,14);  % Burgs algoritm för ordning 14
```

Från reflexionskoefficienterna vill vi nu beräkna AR-parametrarna. Detta kan som vi vet göras med hjälp av Levinson-Durbin step-up rekursion som ekv. 5.14 i Hayes.

```
a5=[1];
for m=1:5
  a5=[a5;0]+gamma5(m)*[0;a5(m:-1:1)]; % Levinson-Durbin step-up rekursion i matrisform
end

a14=[1];
for m=1:14
  a14=[a14;0]+gamma14(m)*[0;a14(m:-1:1)]; % Levinson-Durbin step-up rekursion i matrisform
end
```

Nu kan vi på samma sätt som tidigare beräkna och plotta tillhörande men nu skattade spektraltätheter

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

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

För att plotta dessa på kurvor i samma figur kan dessa rader användas

```
hold on                                          % Samma axlar
plot(f,10*log10(S5),'r',f,10*log10(S14),'g')     % Plotta dessa två kurvor i rött och grönt
title('Sx - blå, Sx_{5} - röd, Sx_{14} - grön')  % Skriv titel
```

Uppgift 2.5: Bestäm optimal modellordning med hjälp av FPE-kriteriet (läs noga sid. 445-447).

Funktionen *burgs* beräknar FPE enligt ekv. 8.126. Eftersom de första 5 länkarna är samma i Latticefiltret av ordning 14 som i det av ordning 5, så räcker det att titta på FPE för det längsta filtret

```
subplot(212)                    % Nedre halvan
plot(FPE14)
title('Final Prediction Error')
Mo=2;                           % Förhoppningsvis ser ni ett minima vid 2
```

Uppgift 2.6: Plotta AR-processens effektspektrum för den optimala modellordningen.

Börja med att köra step-up rekursion för modellordning 2. Burgs algoritm behöver inte köras om eftersom de två första reflexionskoefficinterna är samma för alla ordningar

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

Beräkna skattad spektraltäthet

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

och plotta i översta axlarna

```
subplot(211) hold on plot(f,10*log10(So),'y') title('Sx - blå,
Sx_{5} - röd, Sx_{14} - grön, Sx_{opt} - gul')
```

Uppgift 2.7: Jämför de tre olika skattade effektspektra med det teoretiskt beräknade och kommentera eventuella olikheter.