DCSP-13: Simple Filter

Jianfeng Feng

Department of Computer Science Warwick Univ., UK

Jianfeng.feng@warwick.ac.uk

http://www.dcs.warwick.ac.uk/~feng/dcsp.html
A bit fun today
Applications of filter
Applications of filter

Where is Thumbkin?

Traditional

1. Do Re Mi Do Do Re Mi Do
2. Mi Fa So Mi Fa So
3. So Fa Mi Do So La So Fa Mi Do

[Image: HumanEar.jpg]
simple filter design

• In a number of cases, we can design a linear filter almost by inspection.

• moving poles and zeros like pieces on a chessboard.
simple filter design

• This is not just a simple exercise designed for an introductory course

• For in many cases the use of more sophisticated techniques might not yield a significant improvement in performance.
simple filter design

- In this example consider the audio signal $s(n)$, digitized with a sampling frequency $F_s$ (12kHz).
simple filter design

- In this example consider the audio signal \( s(n) \), digitized with a sampling frequency \( F_s \) (12kHz).

- The signal is affected by a narrowband (i.e., very close to sinusoidal) disturbance \( w(n) \).
simple filter design

- Consider the audio signal $s(n)$ with $F_s=12\text{kHz}$.

- Affected by a narrowband (i.e., very close to sinusoidal) disturbance $w(n)$.

- Fig. below show the frequency spectrum of the overall signal plus noise,

  $$x(n)=s(n)+w(n),$$

  which can be easily determined.
simple filter design

• Notice two facts:

  • The signal has a frequency spectrum within the interval 0 to \(F_s/2\) kHz.

  • The disturbance is at frequency \(F_0\) (1.5) kHz.

• Design and implement a filter that rejects *the disturbance without affecting the signal excessively*.

• We will follow three steps:
Step 1: Frequency domain specifications

- Reject the signal at the frequency of the disturbance.

- Ideally, we would like to have the following frequency response:

\[
H(\omega) = \begin{cases} 
0 & \text{if } \omega = \omega_0 \\
1 & \text{Otherwise}
\end{cases}
\]

where \( \omega_0 = 2\pi \left( \frac{F_0}{F_s} \right) = \pi/4 \) radians, the digital frequency of the disturbance.
Step 2: Determine poles and zeros

We need to place two zeros on the unit circle

\[ z_1 = \exp(j\omega_0) = \exp(j\pi/4) \]

and

\[ z_2 = \exp(-j\omega_0) = \exp(-j\pi/4) \]

\[ z^2 H(z) = k(z - z_1)(z - z_2) \]

\[ = k[z^2 - (z_1 + z_2)z + z_1z_2] \]

\[ = k[z^2 - (1.414)z + 1] \]
Step 2: Determine poles and zeros

- We need to place two zeros on the unit circle
  
  \[ z_1 = \exp(j\omega_0) = \exp(j\pi/4) \]
  
  and
  
  \[ z_2 = \exp(-j\omega_0) = \exp(-j\pi/4) \]

- If we choose, say \( K=1 \), the frequency response is shown in the figure.

\[ z^2 H(z) = k(z - z_1)(z - z_2) \]

\[ = k[z^2 - (z_1 + z_2)z + z_1z_2] \]

\[ = k[z^2 - (1.414)z + 1] \]
• As we can see, it rejects the desired frequency

• As expected, but greatly distorts the signal
Result

Nonrecursive filter

Amplitude

Frequency (kHz)

Disturbed Signal
Filtered Signal

Original Signal
Filtered Signal
A better choice would be to select the poles close to the zeros, \textit{within the unit circle, for stability}.

For example, let the poles be \( p_1 = \rho \exp(j\omega_0) \) and \( p_2 = \rho \exp(-j\omega_0) \).

With \( \rho = 0.95 \), for example we obtain the transfer function

\[
H(z) = k \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)}
\]
Recursive Filter

\[ H(z) = k \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} = k \frac{(z - p_1 + \varepsilon)(z - z_2)}{(z - p_1)(z - p_2)} = k \left( 1 + \frac{(z - z_1)}{(z - p_1)(z - p_2)} \right) \]

\[ \varepsilon = z_1 - p_1 = (1-\rho) z_1 \sim 0. \] So we have \( H(\omega_0) = 0, \) but close to 1 everywhere else.
Recursive Filter

- We choose the overall gain $k$ so that $H(z)|_{z=1}=1$.
- This yields the frequency response function as shown in the figures below.

$$H(z) = k \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} = 0.9543 \frac{z^2}{z^2} - \frac{1.414z + 1}{1.343z + 0.9025}$$
Step 3: Determine the difference equation in the time domain

• From the transfer function, the difference equation is determined by inspection:

\[ y(n) = 0.954 \, x(n) - 1.3495x(n-1) + 0.9543 \, x(n-2) + 1.343 \, y(n-1) - 0.9025 \, y(n-2) \]
Step 3: difference equation

- Easily implemented as a recursion in a high-level language.

- The final signal $y(n)$ with the frequency spectrum shown in the following Fig.,

- We notice the absence of the disturbance
Step 3: Outcomes
\Program Files\MATLAB71\work

simple_filter_design

soundsc(s,Fs)
soundsc(x,Fs)
soundsc(y,Fs)
soundsc(yrc,Fs)
Filter Types

There are four main classes of filter in widespread use:

- Lowpass
- highpass
- bandpass
- bandstop

The name are self-explanatory.
Filter Types

(i)

\[ H(\omega) = \begin{cases} 1 & \omega = 0, \omega_c, 2\pi-\omega_c, 2\pi \\ 0 & \text{otherwise} \end{cases} \]

(ii)

\[ H(\omega) = \begin{cases} 1 & \omega = 0, \omega_c, 2\pi-\omega_c, 2\pi \\ 0 & \text{otherwise} \end{cases} \]

(iii)

\[ H(\omega) = \begin{cases} 1 & \omega = \omega_0, \omega_1, 2\pi-\omega_0, 2\pi-\omega_1 \\ 0 & \text{otherwise} \end{cases} \]

(iv)

\[ H(\omega) = \begin{cases} 1 & \omega = \omega_0, \omega_1, 2\pi-\omega_0, 2\pi-\omega_1 \\ 0 & \text{otherwise} \end{cases} \]
In a general content: General Filter Design

- Any function can be approximated by a polynomials.
- Optimal filter design aims to minimize these ripples.
- We are not discussing them here in details, but the principle is the same.
In a general content: dealing with data

Computer and Information Science » Numerical Analysis and Scientific Computing

Smoothing, Filtering and Prediction - Estimating The Past, Present and Future


DOI: 10.5772/2706
Monograph
OPEN ACCESS

This book describes the classical smoothing, filtering and prediction techniques together with some more
This book describes the classical smoothing, filtering and prediction techniques together with some more
This book describes the classical smoothing, filtering and prediction techniques together with some more
In a general content: machine learning

- \( y[n] = a_0 x[n] + a_1 x[n-1] + \ldots + a_N x[n-N] \)
In a general content: machine learning

- \( y[n] = a_0 x[n] + a_1 x[n-1] + \ldots + a_N x[n-N] \)

Convolution !!!
In a general content: machine learning
A recent example

Dermatologist-level Classification of Skin Cancer with Deep Neural Networks
A recent example
• Some further funs

  Matched filter  (radar for example)

  Wiener filter  (get rid of white noise)
DCSP-14: Matched Filter

Jianfeng Feng

Department of Computer Science Warwick Univ., UK

Jianfeng.feng@warwick.ac.uk

http://www.dcs.warwick.ac.uk/~feng/dcsp.html
Filters

- Stop or allow to pass for certain signals as we have talked before

- Detect certain signals such as radar etc (pattern recognitions)
Ex: Detect it when it comes close
Matched filters: Question

Assume that we can detect

\[ S = (s_1, s_2, s_3, s_4, s_5, s_6) \]

To find \( a_i \)

\[ y(0) = a_0 x(0) + a_1 x(-1) + a_2 x(-2) + a_3 x(-3) + a_4 x(-4) + a_5 x(-5) \]

so that we can find \( S \), more precisely

\[ Y(0) = a_0 s_6 + a_1 s_5 + a_2 s_4 + a_3 s_3 + a_4 s_2 + a_5 s_1 \]
At time $-5$ we have $x(-5) = S_1=1$ (length)

$Y(-5) = a_0 \ 1 \ + \ a_1 \ x(-6) \ + \ a_2 \ x(-7) \ + \ a_3 \ x(-8) \ + \ a_4 \ x(-9) \ + \ a_5 \ x(-10)$
At time -4 we have \( x(-4) = S_2 = 2 \) (length)

\[
Y(-5) = a_0 \ 1 \ + \ a_1 \ x(-6) \ + \ a_2 \ x(-7) \ + \ a_3 \ x(-8) \ + \ a_4 \ x(-9) \ + \ a_5 \ x(-10)
\]

\[
Y(-4) = a_0 \ 2 \ + \ a_1 \ 1 \ + \ a_2 \ x(-6) \ + \ a_3 \ x(-7) \ + \ a_4 \ x(-8) \ + \ a_5 \ x(-9)
\]
At time -3 we have \( x(-3) = S_3 = 3 \) (length)

\[
Y(-5) = a_0 \ 1 + a_1 \ x(-6) + a_2 \ x(-7) + a_3 \ x(-8) + a_4 \ x(-9) + a_5 \ x(-10)
\]

\[
Y(-4) = a_0 \ 2 + a_1 \ 1 + a_2 \ x(-6) + a_3 \ x(-7) + a_4 \ x(-8) + a_5 \ x(-9)
\]

\[
Y(-3) = a_0 \ 3 + a_1 \ 2 + a_2 \ 1 + a_3 \ x(-4) + a_4 \ x(-5) + a_5 \ x(-6)
\]
At time -2 we have $x(-2) = S_4 = 4$ (length)
At time -1 we have $x(-1) = S_5 = 8$ (length)
At time 0 we have $x(0) = S_6 = 6$ (length)
The input signal is

\[ S = (s_1, s_2, s_3, s_4, s_5, s_6) \]

\[ = (1, 2, 3, 4, 8, 6) \]
Matched filters: Question

To find $a_i$

$y(0) = a_0 6 + a_1 8 + a_2 4$

$+ a_3 3 + a_4 2 + a_5 1$

to detect that $S = (s_1, s_2, s_3, s_4, s_5, s_6)$ is here
Matched filters: visualization

\[ a_0 \ x(0) + a_1 \ x(-1) + a_2 \ x(-2) + a_3 \ x(-3) + a_4 \ x(-4) + a_5 \ x(-5) = y(0) \]

Matched signals is matched by the coefficients
Data requirement

Without loss of generality, we can assume that

\[ S_i^2 = 1 \]  \hspace{1cm} \text{(normalized signals)}

and we can assume that

\[ a_i^2 = 1 \]  \hspace{1cm} \text{(normalized coefficients)}
Ideas

Remember that

\[ \sum_{i=0}^{N-1} (a_i - s_{N-i})^2 = 0 \]

If and only if \( a_i = s_{N-i} \) for all \( i \) which essentially says that the coefficients \( a \) and the incoming signal \( s \) are identical (matched, \( a_i = s_{N-i} \)).

Or equivalently

\[ a_0 S_N + a_1 S_{N-1} + \ldots + a_{N-1} S_1 = 1 \]

if and only if \( a_i = s_{N-i} \) for all \( i \)

\[ a_0 S_N + a_1 S_{N-1} + \ldots + a_{N-1} S_1 < 1 \]

otherwise
Matched filter

- Define $a_i = s_{N-i}$ (reversing the order)

$$y(n) = a_0 x(n) + a_1 x(n-1) + \ldots + a_N x(n-N)$$

So when the signal arrives we have

$$y(n) = S_N x(n) + S_{N-1} x(n-1) + \ldots + S_1 x(n-N)$$

$$= S_N S_N + S_{N-1} S_{N-1} + \ldots + S_1 S_1$$
At time -5 we have $y(-5) = 0.0877 \times 0.5262 = 0.0461$

$Y(-5) = a_0 x(-5) + a_1 x(-6) + \ldots$

$S = (0.0877 \ 0.1754 \ 0.2631 \ 0.3508 \ 0.7016 \ 0.5262)$

$a = (0.5262 \ 0.7016 \ 0.3508 \ 0.2631 \ 0.1754 \ 0.0877)$

$x = \begin{pmatrix} s_1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

At time -5 we have $y(-5) = 0.0877 \times 0.5262 = 0.0461$
\[ Y(-4) = a_0 x(-4) + a_1 x(-5) + a_2 x(-6) + \ldots = 0.1538 \]

\[
S = (0.0877, 0.1754, 0.2631, 0.3508, 0.7016, 0.5262)
\]
\[
a = (0.5262, 0.7016, 0.3508, 0.2631, 0.1754, 0.0877)
\]
\[
s_2 \quad s_1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0
\]
At time $-3$ we have

$$Y(-3) = a_0 x(-3) + a_1 x(-4) + a_2 x(-5) + a_3 x(-6) \ldots = .2923$$

S = (0.0877 0.1754 0.2631 0.3508 0.7016 0.5262)

a = (0.5262 0.7016 0.3508 0.2631 0.1754 0.0877)

\[\begin{array}{cccccc}
 s_3 & s_2 & s_1 & 0 & 0 & 0 \\
\end{array}\]
At time -2 we have

\[ Y(-2) = a_0 x(-2) + a_1 x(-3) + a_2 x(-4) + a_3 x(-5) + a_4 x(-6) \ldots = \]

\[
\begin{array}{cccccc}
0.5262 & 0.3508 & 0.7016 & 0.2631 & 0.3508 & 0.1754 \\
0.2631 & 0.3508 & 0.7016 & 0.2631 & 0.3508 & 0.1754
\end{array}
\]

\[ S = (0.0877 \ 0.1754 \ 0.2631 \ 0.3508 \ 0.7016 \ 0.5262) \]

\[ a = (0.5262 \ 0.7016 \ 0.3508 \ 0.2631 \ 0.1754 \ 0.0877) \]
At time \(-1\) we have
\[ y(-1) = a_0 x(-1) + a_1 x(-2) + a_2 x(-3) + a_3 x(-4) + a_4 x(-5) \ldots \]

\[ S = (0.0877 \ 0.1754 \ 0.2631 \ 0.3508 \ 0.7016 \ 0.5262) \]

\[ a = (0.5262 \ 0.7016 \ 0.3508 \ 0.2631 \ 0.1754 \ 0.0877) \]

\[ \begin{array}{cccccc}
 s_5 & s_4 & s_3 & s_2 & s_1 & 0 \\
\end{array} \]
\[ S = (0.0877 \ 0.1754 \ 0.2631 \ 0.3508 \ 0.7016 \ 0.5262) \]

\[ a = (0.5262 \ 0.7016 \ 0.3508 \ 0.2631 \ 0.1754 \ 0.0877) \]

At time 0 we have

\[ Y(0) = a_0 x(0) + a_1 x(-1) + a_2 x(-2) + a_3 x(-3) + a_4 x(-4) + a_5 x(-5) + a_6 x(-6) = 1 \]
At time 1 we have
\[ y(1) = a_0 x(1) + a_1 x(0) + a_2 x(-1) + a_3 x(-2) + a_4 x(-3) + a_5 x(-4) + a_6 x(-5) = 0.7961 \]
Outcome
Matlab Demo

• Matched-filter-demo

• It is a simple idea, but useful

• A bit theory first
Auto-correlation Function

\[ y(n) = a_m x(m) \]
\[ = s(-m)s(n - m) \]
\[ = s(m)s(n + m) \]
\[ = r_{ss}(n) \]
Detection theory

• A means to quantify the ability to discern between information-bearing patterns and noise

• Patterns: stimulus in human, signal in machines

• Noise: random patterns that distract from the information
Can it be useful?

Any signal here?
Filter output
Filter output

```matlab
s = rand(1,100);
s = s/sqrt(s'*s);
a = s;
k = 100;
N = 100*k;
sigma = 0.2;
x = randn(1,N)*sigma;
signal = zeros(1,N);
for i = 3*N/k+1:N/4/k
    x(i) = a(i-3*N/k)+x(i);
signal(i) = a(i-3*N/k);
end
for i = 1:N*(k-1)/k
    c(i) = x([i:i+N/k-1])*s';
end
figure(1)
plot(x);
hold on
plot(signal,'r');
figure(2)
plot(c)
```

Matched filter

- The result is amazing
- It depends on SNR
- We will not go into details, but you might be able to investigate it using Matlab
Next week

RGB = imread('bush.png');
I = rgb2gray(RGB);
J = imnoise(I,'gaussian',0,0.005);
K = wiener2(J,[5 5]);
figure, imshow(J), figure, imshow(K)