DCSP-12: Filter II

Jianfeng Feng

Department of Computer Science Warwick Univ., UK

Jianfeng.feng@warwick.ac.uk

http://www.dcs.warwick.ac.uk/~feng/dcsp.html
We can formally represent the frequency response of the filter by substituting
\[ z = \exp(j \omega) \]
and obtain
\[ H(\omega) = K[(\exp(-j \omega) - \alpha_1) \ldots (\exp(-j \omega) - \alpha_N)] \]

Consider an example
What is a Filter?

• For a given power spectrum of a signal
What is a Filter?

- For a given power spectrum of a signal
What is a Filter?

- For a given power spectrum of a signal

![Diagram showing signal and filter](image)
Ideal response function as plotted below

- Find coefficients $a$ and $b$

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

$$+ b_1 y(n-1) + \ldots + b_N y(n-N)$$
Example

\[ H(\omega) = K[(\exp(-j\omega) - \alpha_1)(\exp(-j\omega) - \alpha_2)] \]

(sampling frequency \( F_s \), stop frequency \( F_0 \), put zero = \( (F_0/F_s/2)\pi = 2\pi F_0/F_s \))

with \( \alpha_1 = \exp(-j\pi/4) \), \( \alpha_2 = \exp(j\pi/4) \), for example, then

\[ Y(\omega) = H(\omega) X(\omega) \]

and \( Y(\omega) = 0 \) whenever \( \omega = \pi/4 \).

Any signal with a frequency of \( F_0 \) will be stopped
Transfer function

```matlab
h = 0.01;
for i = 1:314
    x(i) = i*h;
    f(i) = (exp(-j*x(i)) - exp(-j*pi/4)) * (exp(-j*x(i)) - exp(j*pi/4));
end
plot(x, abs(f))
```
Moving $\alpha_1$ along the circle, we are able to stop a signal with a given frequency.
The original difference expression can be recovered by

\[
H(z) = k [(z^{-1} - \alpha_1)(z^{-1} - \alpha_2)]
= k [z^{-2} - (\alpha_1 + \alpha_2) z^{-1} + (\alpha_1 \alpha_2)]
\]

\[
y(n) = k [x(n-2) - (\alpha_1 + \alpha_2) x(n-1) + (\alpha_1 \alpha_2) x(n)]
\]

\[
a_0 = k (\alpha_1 \alpha_2) = k, \quad a_1 = -k(\alpha_1 + \alpha_2), \quad a_2 = k
\]
Recursive Filters

Of the many filter transfer function, the most commonly use in DSP are the recursive filters, so called because their current output depends not only on the last $N$ inputs but also on the last $N$ outputs.

\[ y(n) = a_0 x(n) + a_1 x(n-1) + \ldots + a_N x(n-N) + b_1 y(n-1) + \ldots + b_N y(n-N) \]
Transfer function

\[ H(z) = \frac{A(z)}{B(z)} \]

where

\[ A(z) = a_0 + a_1 z^{-1} + \ldots + a_N z^{-N} \]
\[ B(z) = 1 - b_1 z^{-1} - \ldots - b_N z^{-N} \]
Poles and zeros

\[ H(z) = \frac{A(z)}{B(z)} = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{z - \alpha_i}{z - \beta_i} \right) \]

- We know that the roots \( \alpha_n \) are the zeros of the transfer function.
- The roots of the equation \( B(z) = 0 \) are called the poles of the transfer function.
- They have greater significance for the behaviour of \( H(z) \): it is singular at the points \( z = \beta_n \).
- Poles are drawn on the z-plane as crosses, as shown
Poles and zeros
Poles and zeros

ON this figure, the unit circle has been shown on the z-plane.

In order for a recursive filter to be *bounded input bounded output (BIBO) stable*, all of the poles of its transfer function must lie inside the unit circle.

A filter is BIBO stable if any bounded input sequence gives rise to a bounded output sequence.

Now if the pole of the transfer lie insider the unit circle,

A simple case (N=1)

\[ H(z) = k \frac{(z)}{(z - b)} = \frac{k}{(1 - bz^{-1})} \]

\[ y(n) = y(n - 1) + kx(n) \]
Poles and zeros

$y(n)$ is stable if and only if the pole is inside the unit circle

$|\beta| < 1$

In general, $y(n)$ is stable if and only if all poles are inside the unit circle

$|\beta_i| < 1, \ i=1, \ldots, N$
Poles and zeros

If $|\beta_i| = 1$, the filter is said to be conditionally stable: some input sequence will lead to bounded output sequence and some will not.

Since MA filters have no poles, they are always BIBO (bound input bound output) stable: zeros have no effect on stability.
Now suppose we have the transfer function of a filter

\[ H(z) = \frac{A(z)}{B(z)} = k \sum_{i=1}^{N} \left( z - \alpha_i \right) \prod_{i=1}^{N} \left( z - \beta_i \right) \]

We can formally represent the frequency response of the filter by substituting

\[ z = \exp(j \omega) \]
Obviously, $H(\omega)$ depends on the locations of the poles and zeros of the transfer function, a fact which can be made more explicit by factoring the numerator and denominator polynomials to write

$$H(\omega) = \frac{A(\omega)}{B(\omega)} = k \prod_{i=1}^{N} \left( e^{j\omega} - a_i \right) \prod_{i=1}^{N} \left( e^{j\omega} - b_i \right)$$
Each factor in the numerator or denominator is a complex function of frequency, which has a graphical interpretation in the terms of the location of the corresponding roots in relation to the unit circle.

Thus we can make a reasonable guess about the filter frequency response imply by looking at the pole-zero diagram.
Filter Types

There are four main classes of filter in widespread use:

• Lowpass
• highpass
• bandpass
• bandstop

filters.
The name are self-explanatory
This four types are shown in the next figure
Filter Types

(i)  
\[ H(\omega) \]

(ii)  
\[ H(\omega) \]

(iii)  
\[ H(\omega) \]

(iv)  
\[ H(\omega) \]
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
Google’s AI AlphaGo to take on world No 1 Lee Se-dol in live broadcast
DeepMind’s Go-playing software will play South Korean in five-match game live-streamed on YouTube, following victory over European champion
Applications of filter
Applications of filter

Where is Thumbkin?

Traditional

Do Re Mi Do Do Re Mi Do

Mi Fa So Mi Fa So

Fa Mi Do So Le So Fa Mi Do
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).
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 $w(n)$. 
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.
Signal
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.

\[
 z^2 H(z) = k(z - z_1)(z - z_2) \\
 = k[z^2 - (z_1 + z_2)z + z_1z_2] \\
 = k[z^2 - (1.414)z + 1]
\]
PSD and Phase

- 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

Amplitude

AMI
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

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

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

$$H(z) = k \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} = 0.9543 \frac{z^2 - 1.414z + 1}{z^2 + 1.343z + 0.9025}$$

Recursive Filter
Step 3: Determine the difference equation in the time domain

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

\[ y(n) = 0.954 \times x(n) - 1.3495 \times x(n-1) + 0.9543 \times x(n-2) \\
+ 1.343 \times y(n-1) - 0.9025 \times y(n-2) \]
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

Recursive filter

Amplitude vs. Frequency (kHz)

Disturbed Signal vs. Filtered Signal

Original Signal vs. Filtered Signal
simple_filter_design
soundsc(s,Fs)
soundsc(x,Fs)
soundsc(y,Fs)
soundsc(yrc,Fs)
General Filter Design

- Using Chebyshev or Butterworth polynomials to approximate a square wave function in the frequency domain.
- Any function can be approximated by a polynomial.
- Optimal filter design aims to minimize these ripples.
- We are not discussing them here in details, but the principle is the same.
Next week

• Some further funs

  Matched filter (radar for example)

  Wiener filter (get rid of white noise)
Stability revisited

In order for a recursive filter to be *bounded input bounded output (BIBO) stable*, all of the poles must lie inside the unit circle.

A simple case (N=1)

\[
H(z) = k \frac{z}{z - b} = \frac{k}{1 - bz^{-1}}
\]

\[
y(n) = y(n-1) + kx(n)
\]
Stability revisited

\( y(n) \) is stable if and only if the pole is inside the unit circle

\[ |\beta| < 1 \]

In general, \( y(n) \) is stable if and only if all poles are inside the unit circle (can you prove this!!)

\[ |\beta_i| < 1, \quad i = 1, \ldots, N \]