ELEC 3218 — Signal and Image Processing — Digital Filter Design

Professor Eric Rogers

School of Electronics and Computer Science
University of Southampton
etar@ecs.soton.ac.uk
Office: Building 1, Room 2041
Road map

- To introduce the structure and basic properties of Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters.
- To consider the design of these filters.
Basics

The $z$-transform of the impulse response $\{h_n\}$ of the causal LTI system defined by

$$
\sum_{l=0}^{k} a_l \cdot y_{n-l} = \sum_{l=0}^{m} b_l \cdot x_{n-l}
$$

with $\{y_n\} = \{h_n\} \ast \{x_n\}$ is a rational function
$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_k z^{-k}}$

$(b_m \neq 0, \ k \neq 0)$ which can also be written as:

$$H(z) = \frac{z^k \sum_{l=0}^{m} b_l z^{m-l}}{z^m \sum_{l=0}^{k} a_l z^{k-l}}$$

$H(z)$ has $m$ zeros and $k$ poles at non-zero locations in the $z$ plane, plus $k - m$ zeros (if $k > m$) of $m - k$ zeros (if $m > k$) at $z = 0$. 
Basics

This function can be converted into the form

\[ H(z) = \frac{b_0}{a_0} \cdot \frac{\prod_{l=1}^{m}(1 - c_l \cdot z^{-1})}{\prod_{l=1}^{k}(1 - d_l \cdot z^{-1})} = \frac{b_0}{a_0} \cdot z^{k-m} \cdot \frac{\prod_{l=1}^{m}(z - c_l)}{\prod_{l=1}^{k}(z - d_l)} \]

where the \( c_l \) are the non-zero positions of zeros \( (H(c_l) = 0) \) and the \( d_l \) are the non-zero positions of the poles (i.e. \( z \rightarrow d_l \Rightarrow |H(z)| \rightarrow \infty \)) of \( H(z) \). Except for a constant factor, \( H(z) \) is entirely characterized by the position of these zeros and poles.

As with the Fourier transform, convolution in the time domain corresponds to complex multiplication in the \( z \)-domain:

\[ u_n \cdot \circ \cdot \circ \ U(z), v_n \cdot \circ \cdot \circ \ V(z) \Rightarrow u_n \ast v_n \cdot \circ \cdot \circ \ U(z) \cdot V(z) \]

Delaying a sequence by one corresponds in the \( z \)-domain to multiplication with \( z^{-1} \):

\[ u_n - \Delta n \cdot \circ \cdot \circ \ U(z) \cdot z^{-\Delta n} \]
This example is an amplitude plot of

\[ H(z) = \frac{0.8}{1 - 0.2 \cdot z^{-1}} = \frac{0.8z}{z - 0.2} \]

which has a zero at 0 and a pole at 0.2.
\[ H(z) = \frac{z}{z-0.7} = \frac{1}{1-0.7z^{-1}} \]

\[ H(z) = \frac{z}{z-0.9} = \frac{1}{1-0.9z^{-1}} \]
$$H(z) = \frac{z}{z-1} = \frac{1}{1-z^{-1}}$$

$$H(z) = \frac{z}{z-1.1} = \frac{1}{1-1.1z^{-1}}$$

[Diagrams of z-planes and impulse responses]
$$H(z) = \frac{z^2}{(z-0.9\cdot e^{j\pi/6})(z-0.9\cdot e^{-j\pi/6})} = \frac{1}{1-1.8\cos(\pi/6)z^{-1}+0.92\cdot z^{-2}}$$
\[ H(z) = \frac{z^2}{(z - 0.9 e^{j\pi/2})(z - 0.9 e^{-j\pi/2})} = \frac{1}{1 - 1.8 \cos(\pi/2)z^{-1} + 0.9^2 z^{-2}} \]

\[ H(z) = \frac{z}{z+1} = \frac{1}{1+z^{-1}} \]
Digital FIR Filtering

- Finite impulse response (or moving average, MA) filter:

- Tap delay line (TDL) of length $N$;

- Described by difference equation, $y[n] = \sum_{\nu=0}^{N-1} b_n x[n - \nu]$;

- No feedback: inherently stable.
Impulse Response

- For an FIR filter, the impulse response $h[n]$ (earlier stated as zero-state response, i.e. the TDL is initialised to zero) is given by its filter coefficients:

- The impulse response has finite support with $h[n] = 0$ for all $n \geq N$;
- The impulse response can give direct clues about multipath propagation, or echos and reverberation.
Frequency Response

- Frequency response $H(e^{j\Omega})$: measures the gain of the system in the steady state.
- Impulse and frequency response are a Fourier pair: $h[n] \circ - \bullet H(e^{j\Omega})$. 

![Graphs showing impulse and frequency response](image)
Magnitude and Phase Response

- The magnitude decides whether the system has e.g. low-pass, high-pass, or band-pass characteristic;

- The phase contains a number of filter characteristics which are critical in some applications:
Group Delay

- From $H(e^{j\Omega}) = |H(e^{j\Omega})|e^{j\Phi(\Omega)}$, note for the group delay $g(\Omega)$

$$g(e^{j\Omega}) = -\frac{d}{d\Omega} \Phi(\Omega) \quad (1)$$

- The group delay tells us by how long certain frequency components will be delayed when propagating through the filter;
- Note the difference between linear phase and non-linear phase systems.
Reconsider the FIR filter:

- The \( z \)-transform of the defining difference equation:

\[
Y(z) = b_0X(z) + b_1z^{-1}X(z) + b_2z^{-2}X(z) + \cdots + b_{N-1}z^{-N+1}X(z) \tag{2}
\]

\[
H(z) = \frac{Y(z)}{X(z)} = \sum_{i=0}^{N-1} b_i z^{-i} \tag{3}
\]

- The \( z \)-transform with \( z = e^{j\Omega + \alpha} \) can capture the transient behaviour of a system;

- Note that \( H(e^{j\Omega}) = H(z) \big|_{z=e^{j\Omega}} \)
z-Transform — More Useful Properties

- Assume a z-transform pair \( h[n] \circ - \bullet H(z) = \sum_{i=-\infty}^{\infty} h_i z^{-i} \)

- Time shift:
  \[
  h[n - \Delta] \circ - \bullet H(z) z^{-\Delta}
  \]  
  (4)

- Complex conjugation:
  \[
  h^*[n] \circ - \bullet H^*(e^{j\Omega}) = \sum_{i=-\infty}^{\infty} h_i^* z^{-i}
  \]  
  (5)

- Time reverse:
  \[
  h[-n] \circ - \bullet H(z^{-1}) = \sum_{i=-\infty}^{\infty} h_i z^i
  \]  
  (6)
We can factorise the transfer-function $H(z)$:

$$H(z) = b_0 + b_1 z^{-1} + \cdots + b_{N-1} z^{-N+1}$$

$$= b_0 (1 - \beta_0 z^{-1})(1 - \beta_1 z^{-1}) \cdots (1 - \beta_{N-1} z^{-N+1})$$

The roots $\beta_i$ of $H(z)$ are called the zeros of the transfer-function;

Often a pole-zero plot is used for visualisation.
Real Valued Systems

- If the impulse only contains real-valued coefficients, the zeros $\beta_i$ must be either real valued, or occur as complex conjugate pairs.
- We could factorise the system into a sequence of first order sections:

\[
\begin{align*}
X(z) & \xrightarrow{b_0} 1 - \beta_0 z^{-1} \rightarrow \cdots \rightarrow 1 - \beta_{N-1} z^{-1} \rightarrow Y(z)
\end{align*}
\]

- A complex multiplication requires 4 real valued multiplications; therefore, for implementation purposes, we would always prefer (7) over (8).
Minimum and Maximum Phase

- For a minimum phase system, all zeros lie inside the unit circle;
- A maximum phase system has all zeros outside the unit circle;
- Interesting: if $h[n]$ is minimum phase, then $h[-n]$ is maximum phase;
- Example for $H_1(z) = 1 + \frac{1}{2}z^{-1}$ and $H_2(z) = \frac{1}{2} + z^{-1}$

$$H_1(z) = 1 + \frac{1}{2}z^{-1}$$
$$H_2(z) = \frac{1}{2} + z^{-1}$$
$$H_3(z) = H_1(z)H_2(z)$$
Linear Phase Filter

- Recall: linear phase means constant group delay;
- A linear phase system must have a z-transform of the form (including a possible delay for causality)

\[ H(z) = H_{\text{min. phase}}(z)H_{\text{min. phase}}(z^{-1})\prod_m (1 - e^{j\phi_m}z^{-1}) \]  \hspace{1cm} (9)

- Any zeros of a linear phase system must have a reciprocally matching partner (unless on the unit circle);
- Linear phase if, and only if, the impulse response is symmetric.
Note that certain properties of a system are best assessed in a specific domain: either impulse response, frequency domain, or transfer function;

Phase information is often vital, e.g. if we need to invert the system (for equalisation or control purposes);

Most FIR filter designs yield a symmetric impulse response and hence are linear phase;

We will consider such filter designs next ...
Filter Design Characteristics

- Design characteristics: pass-band width, transition band width, stop-band edge, stop-band attenuation, and pass band ripple:

- Generally: high quality requires high number of coefficients.
Different Frequency Response

- Four different basic filter types as defined by their magnitude response: (a) low-pass, (b) high-pass, (c) band-pass, and (d) band-stop filter:
FIR Filter Design: IDFT and Windowing

- We often have an idea what the magnitude response $|H(e^{j\Omega})|$ of the desired filter should look like; additionally, we need to “invert” a phase response (e.g. linear);

- The frequency response is inversely Fourier transformed;

- The resulting time domain response is an approximation of the desired impulse response (holding the filter coefficients);

- Some windowing may be required in order to enforce finite support in the time domain.

- A variety of design approaches based on minimax or least squares methods exists.
FIR Design Examples

- Comparison of two filter design with 16 and 48 coefficients, respectively:

- Both design have identical pass-band edge at $\Omega = 0.1\pi$. 
IIR Filtering

- There are various ways to represent the difference equation as a flow graph:

\[ y[n] = \sum_{l=0}^{L-1} a_l x[n - l] + \sum_{j=1}^{J-1} b_j y[n - j] \]  

(10)
IIR Stability

- Transfer-function:

\[
H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1z^{-1} + \cdots + b_{L-1}z^{-L+1}}{1 - a_1z^{-1} - a_2z^{-2} - \cdots - a_{J-1}z^{-J+1}} = b_0 \frac{(1 - \beta_0 z^{-1})(1 - \beta_1 z^{-1}) \cdots (1 - \beta_{L-2} z^{-1})}{(1 - \alpha_0 z^{-1})(1 - \alpha_1 z^{-1}) \cdots (1 - \alpha_{J-2} z^{-1})}
\]

- Note that \(H(z)\) can be split into a cascade of FIR and recursive first-order sections;

- Each recursive section \(\frac{1}{1-\alpha z^{-1}}\) must be stable; note geometric series

\[
\frac{1}{1 - \alpha z^{-1}} = \sum_{i=0}^{\infty} a^i z^{-i} \text{ for } |a| < 1
\]

- The poles of the filter transfer-function must be inside the unit circle for stability.
IIR Filter design techniques I

The design of a filter starts with specifying the desired parameters:

- The **pass-band** is the frequency range where we want to approximate a gain of one.
- The **stop-band** is the frequency range where we want to approximate a gain of zero.
- The **order** of a filter is the number of poles it uses in the $z$-domain, and equivalently the number of delay elements necessary to implement it.
- Both the pass-band and stop-bands will in practice not have gains of exactly one and zero, respectively, but may show several deviations from these ideal values. Also these **ripples** may have a specified maximum quotient between the highest and lowest gain.
- There will in practice not be an abrupt change of gain between pass-band and stop-band, but a **transition band** where the frequency response will gradually change from its pass-band to its stop-band value.
IIR Filter design techniques II

The designer can then trade off conflicting goals such as a small transition band, a low order, a low ripple amplitude, or even an absence of ripples.

Design techniques for making these tradeoffs for analog filters (involving capacitors, resistors, coils) can also be used to design digital IIR filters:

**Butterworth filters**
Have no ripples, gain falls monotonically across the pass and transition band. Within the pass-band, the gain drops slowly down to \(1 - \sqrt{1/2}\) (-3dB). Outside the pass-band, it drops asymptotically by a factor \(2^N\) per octave \((N \cdot 20\text{dB/decade})\).

**Chebyshev type I filters**
Distribute the gain error uniformly throughout the pass-band (equi-ripples) and drop off monotonically outside.

**Chebyshev type II filters**
Distribute the gain error uniformly throughout the stop-band (equi-ripples) and drop off monotonically in the passband.
Elliptic filters (Cauer filters)
Distribute the gain error as equi-ripples both in the pass-band and stop-band. This type of filter is optimal in terms of the combination of the pass-band gain tolerance, stop-band gain tolerance, and transition-band width that can be achieved at a given filter order.

All these filter design techniques are implemented in the MATLAB Signal Processing Toolbox in the functions butter, cheby1, cheby2, and ellip, which output the coefficients $a_n$ and $b_n$ of the difference equation that describes the filter. These can be applied with filter to sequence, or can be visualized with zplane as poles/zeros in the z-domain, with impz as an impulse response, and with freqz as an amplitude and phase spectrum. The commands sptool and fdatool provide interactive GUIs to design digital filters.
Butterworth filter design example

order: 1, cut-off frequency (-3 dB): $0.25 \times f_s/2$
Butterworth filter design example

order: 5, cut-off frequency (-3 dB): $0.25 \times f_s / 2$
Chebyshev type I filter design example

order: 5, cut-off frequency: $0.5 \times f_s / 2$, pass-band ripple: -3 dB
Chebyshev type II filter design example

order: 5, cut-off frequency: $0.5 \times f_s/2$, stop-band ripple: -20 dB
Elliptic filter design example

- Order: 5, cut-off frequency: $0.5 \times f_s/2$, pass-band ripple: -3dB, stop-band ripple: -20dB
Digital Notch Filters

- In some cases it is required to remove the effects of a single frequency from a signal — acoustics. If the frequency is known (strong assumption) then a notch filter can be used.

- The filter is required to produce real signals in response to real input signals. Hence include a complex conjugate pair of zeros and two poles.

- Hence design an FIR filter. Let $f_s$ denote the sampling frequency and $f$ the frequency component to be removed.

- Filter transfer-function

\[ G(z) = \frac{(z - e^{j\frac{2\pi f}{f_s}})(z - e^{-j\frac{2\pi f}{f_s}})}{z^2} \]
Digital Notch Filters

For implementation

\[ G(z) = 1 - 2 \cos \left( \frac{2\pi f}{f_s} \right) z^{-1} + z^{-2} \]

2 tap FIR filter.

If \( f \) varies then an adaptive notch filter is applicable.
FIR/IIR Comparison

- Comparison between FIR and IIR filters (both 24 coefficients):
- Note that quantisation can lead to instability of an IIR filter!
Recall that for the general IIR filter:

\[
y[n] = \sum_{l=0}^{L-1} a_l x[n - l] + \sum_{j=1}^{J-1} b_j y[n - j] = a^T x_n + b^T y_{n-1} \tag{14}
\]

Vector notation:

\[
x_n = \begin{bmatrix} x[n] & x[n-1] & \cdots & x[n-L+1] \end{bmatrix}^T \tag{15}
\]

\[
y_{n-1} = \begin{bmatrix} y[n-1] & y[n-2] & \cdots & y[n-J+1] \end{bmatrix}^T \tag{16}
\]

\[
a = \begin{bmatrix} a[0] & a[1] & \cdots & a[L-1] \end{bmatrix}^T \tag{17}
\]

\[
b = \begin{bmatrix} b[1] & b[2] & \cdots & b[J-1] \end{bmatrix}^T \tag{18}
\]
The required steps per sampling period are:
- Update TDL (memory moves) – Calculate $L + J$
multiply-accumulates (MAC)

**Example 1.** Sampling rate $f_s = 8\text{kHz}$; we want to model the
impulse response of an acoustic system with a $T = .5\text{s}$ duration,
and therefore need $T \cdot f_s = 4000$ coefficients. Total complexity:
$C = 4000f_s = 32\text{MMAC/s}$;

**Example 2.** We double the sampling rate to wide-band audio with
$f_s = 16\text{kHz}$; we require 8000 coefficients and yield a total
complexity of $C = 128\text{MMAC/s}$;

Note: doubling the sampling rate means quadrupling the complexity!

State-of-the-art digital signal processors perform approximately 1
GMAC/s.
Summary of FIR/IIR Filtering

- Characterisation by impulse response, frequency response, or transfer-function;
- Be aware of properties such as causality and minimum/maximum/non-minimum phase;
- **Stability.** FIR is strictly stable, IIR is not;
- **Quality/Complexity.** IIR filters need less coefficients than FIR.
- **Linear Phase.** For FIR filters a symmetric impulse response is necessary and sufficient for linear phase; IIR are never linear phase (although some yield an approximation in the pass-band);
- **Implementation.** Both FIR and IIR are sensitive to fixed point implementation, but IIR has larger dynamics in its filter coefficients and may even become unstable when quantised.