Digital Filters IIR
(& Their Corresponding Analog Filters)

ELEC 3004: Systems: Signals & Controls
Dr. Surya Singh

Lecture 10

elec3004@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~elec3004/

Lecture Schedule:

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Lecture Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28-Feb</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>7-Mar</td>
<td>Systems Overview</td>
</tr>
<tr>
<td>3</td>
<td>9-Mar</td>
<td>Systems: Linear Differential Systems</td>
</tr>
<tr>
<td>4</td>
<td>14-Mar</td>
<td>Sampling Theory &amp; Data Acquisition</td>
</tr>
<tr>
<td>5</td>
<td>16-Mar</td>
<td>Aliasing &amp; Anti-aliasing</td>
</tr>
<tr>
<td>6</td>
<td>21-Mar</td>
<td>Discrete Time Analysis &amp; Z-Transform</td>
</tr>
<tr>
<td>7</td>
<td>23-Mar</td>
<td>Second Order LTID (Filtering) &amp; Convolution Review</td>
</tr>
<tr>
<td>8</td>
<td>28-Mar</td>
<td>Frequency Response</td>
</tr>
<tr>
<td>9</td>
<td>30-Mar</td>
<td>Filter Analysis</td>
</tr>
<tr>
<td>10</td>
<td>4-Apr</td>
<td>Digital Filters (IIR) &amp; Filter Analysis</td>
</tr>
<tr>
<td>11</td>
<td>6-Apr</td>
<td>Digital Windows</td>
</tr>
<tr>
<td>12</td>
<td>11-Apr</td>
<td>Digital Filter (FIR)</td>
</tr>
<tr>
<td>13</td>
<td>13-Apr</td>
<td>FFT</td>
</tr>
<tr>
<td>14</td>
<td>15-Apr</td>
<td>FFT</td>
</tr>
<tr>
<td>15</td>
<td>18-Apr</td>
<td>Digital Filter (FIR)</td>
</tr>
<tr>
<td>16</td>
<td>20-Apr</td>
<td>Holiday</td>
</tr>
<tr>
<td>17</td>
<td>25-Apr</td>
<td>Holiday</td>
</tr>
<tr>
<td>18</td>
<td>27-Apr</td>
<td>Active Filters &amp; Estimation</td>
</tr>
<tr>
<td>19</td>
<td>2-Aug</td>
<td>Introduction to Feedback Control</td>
</tr>
<tr>
<td>20</td>
<td>4-Aug</td>
<td>Servo-regulation/PID</td>
</tr>
<tr>
<td>21</td>
<td>9-Aug</td>
<td>Introduction to (Digital) Control</td>
</tr>
<tr>
<td>22</td>
<td>11-Aug</td>
<td>Digital Control</td>
</tr>
<tr>
<td>23</td>
<td>16-Aug</td>
<td>Digital Control Design</td>
</tr>
<tr>
<td>24</td>
<td>18-Aug</td>
<td>Stability</td>
</tr>
<tr>
<td>26</td>
<td>25-Aug</td>
<td>Applications in Industry</td>
</tr>
<tr>
<td>27</td>
<td>30-Aug</td>
<td>System Identification &amp; Information Theory</td>
</tr>
<tr>
<td>28</td>
<td>1-Sep</td>
<td>Summary and Course Review</td>
</tr>
</tbody>
</table>
Follow Along Reading:

B. P. Lathi
Signal processing and linear systems
1998
TK5102.9.L38 1998

Today

• Chapter 10
  (Discrete-Time System Analysis Using the $z$-Transform)
  – § 10.3 Properties of DTFT
  – § 10.5 Discrete-Time Linear System analysis by DTFT
  – § 10.7 Generalization of DTFT to the $z$-Transform

• Chapter 12
  (Frequency Response and Digital Filters)
  – § 12.1 Frequency Response of Discrete-Time Systems
  – § 12.3 Digital Filters
  – § 12.4 Filter Design Criteria
  – § 12.7 Nonrecursive Filters

Next Time

Announcements: Cyclone Debbie

• Lecture 10: Cancelled (Sorry!)
  – We will makeup some of the material today! 😊

Let’s Start With: (analog) Filters!

Filters

- **Frequency-shaping filters**: LTI systems that change the shape of the spectrum
- **Frequency-selective filters**: Systems that pass some frequencies undistorted and attenuate others
Filters

Specified Values:
- $G_p =$ minimum passband gain

Typically:
$$ G_p = \frac{1}{\sqrt{2}} = -3dB $$

- $G_s =$ maximum stopband gain
  - Low, not zero (sorry!)
  - For realizable filters, the gain cannot be zero over a finite band (Paley-Wiener condition)

- Transition Band:
  transition from the passband to the stopband $\omega_p \neq \omega_s$

Filter Design & z-Transform

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Mapping</th>
<th>Design Parameters</th>
</tr>
</thead>
</table>
| Low-pass    | $z^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$ | $\alpha = \sin(\omega_c - \omega_i/2)/\sin(\omega_c + \omega_i/2)$  
$\omega_c = \text{desired cutoff frequency}$  
$\omega_i = \text{desired cutoff frequency}$ |
| High-pass   | $z^{-1} = \frac{-z^{-1} + \alpha}{1 + \alpha z^{-1}}$ | $\alpha = -\cos(\omega_c + \omega_i/2)/\cos(\omega_c - \omega_i/2)$  
$\omega_c = \text{desired cutoff frequency}$  
$\omega_i = \text{desired cutoff frequency}$ |
| Bandpass    | $z^{-1} = \frac{z^{-1} - [2\alpha(1 + \beta)]z^{-1} + [(\beta - 1)(1 + \beta) - 2\alpha(1 + \beta)]z^{-2} - [2\alpha(\beta + 1)]z^{-3} + 1}{(1 - \beta)/(\beta + 1)z^{-2} - [2\alpha(\beta + 1)]z^{-3} + 1}$ | $\alpha = \cos(\omega_{cl} + \omega_{cu}/2)/\cos(\omega_{cl} - \omega_{cu}/2)$  
$\beta = \tan((\omega_{cl} - \omega_{cu}/2) \tan(\omega_c/2)$  
$\omega_{cl} = \text{desired lower cutoff frequency}$  
$\omega_{cu} = \text{desired upper cutoff frequency}$ |
| Bandstop    | $z^{-1} = \frac{z^{-1} - [2\alpha(1 + \beta)]z^{-1} + [(1 - \beta)/(1 + \beta) - 2\alpha(1 + \beta)]z^{-2} - [2\alpha(\beta + 1)]z^{-3} + 1}{(1 - \beta)/(1 + \beta)z^{-2} - [2\alpha(\beta + 1)]z^{-3} + 1}$ | $\alpha = \cos(\omega_{cl} + \omega_{cu}/2)/\cos(\omega_{cl} - \omega_{cu}/2)$  
$\beta = \tan((\omega_{cl} - \omega_{cu}/2) \tan(\omega_c/2)$  
$\omega_{cl} = \text{desired lower cutoff frequency}$  
$\omega_{cu} = \text{desired upper cutoff frequency}$ |
Butterworth Filters

- Butterworth: Smooth in the pass-band
- The amplitude response $|H(j\omega)|$ of an $n^{th}$ order Butterworth low pass filter is given by:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}}$$

- The normalized case ($\omega_c=1$)

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \omega^{2n}}} \quad \Rightarrow \quad H(j\omega)H(-j\omega) = |H(j\omega)|^2 = \frac{1}{1 + \omega^{2n}}$$

Recall that: $|H(j\omega)|^2 = H(j\omega)H(-j\omega)$
Butterworth Filters of Increasing Order:
Seeing this Using a Pole-Zero Diagram

- Increasing the order, increases the number of poles:

  ➔ Odd orders (n=1,3,5…):
  - Have a pole on the Real Axis

  ➔ Even orders (n=2,4,6…):
  - Have a pole on the off axis

Since H(s) is stable and causal, its poles must lie in the LHP
- Poles of -H(s) are those in the RHP
- Poles lie on the unit circle (for a normalized filter)

\[ H(s) = \frac{1}{(s - s_1)(s - s_2)\ldots(s - s_n)} \]

Where:

\[ s_k = e^{\frac{j\pi}{2n}(2k + n - 1)} \]

\[ = \cos \frac{\pi}{2n}(2k + n - 1) + j \sin \frac{\pi}{2n}(2k + n - 1) \quad k = 1, 2, 3, \ldots, n \]

n is the order of the filter
Butterworth Filters: 4th Order Filter Example

- Plugging in for n=4, k=1,…4:

\[ H(s) = \frac{1}{(s + 0.3827 - j0.9239)(s + 0.3827 + j0.9239)(s + 0.9239 - j0.3827)(s + 0.9239 + j0.3827)} \]

\[ = \frac{1}{(s^2 + 0.7654s + 1)(s^2 + 1.8478s + 1)} \]

\[ = \frac{1}{s^4 + 2.6131s^3 + 3.4142s^2 + 2.6131s + 1} \]

- We can generalize → Butterworth Table

<table>
<thead>
<tr>
<th>n</th>
<th>a₁</th>
<th>a₂</th>
<th>a₃</th>
<th>a₄</th>
<th>a₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>1.41421356</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>2.00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.61312503</td>
<td>3.41421356</td>
<td>2.61312503</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.23606798</td>
<td>5.23606798</td>
<td>5.23606798</td>
<td>2.32666798</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.86370331</td>
<td>7.48410162</td>
<td>9.14352017</td>
<td>7.48410162</td>
<td>3.86370331</td>
</tr>
</tbody>
</table>

This is for 3dB bandwidth at \( \omega_c = 1 \)

Butterworth Filters: Scaling Back (from Normalized)

- Start with Normalized equation & Table
- Replace \( \omega \) with \( \frac{\omega_c}{\omega_c} \) in the filter equation

- For example:
  for \( f_c = 100Hz \) → \( \omega_c = 200\pi \) rad/sec

  From the Butterworth table: for n=2, \( a_1 = \sqrt{2} \)

  Thus:

  \[ H(s) = \frac{1}{\left( \frac{s}{200\pi} \right)^2 + \sqrt{2} \left( \frac{s}{200\pi} \right) + 1} \]

  \[ = \frac{1}{s^2 + 200\pi \sqrt{2} + 40,000\pi^2} \]
Butterworth: Determination of Filter Order

- Define $G_x$ as the gain of a lowpass Butterworth filter at $\omega = \omega_x$
- Then:

$$\hat{G}_d = 20 \log_{10} |H(j\omega_x)| = -10 \log \left[ 1 + \left( \frac{\omega_x}{\omega_c} \right)^{2n} \right]$$

And thus:

$$\hat{G}_p = -10 \log \left[ 1 + \left( \frac{\omega_p}{\omega_c} \right)^{2n} \right]$$

$$\hat{G}_s = -10 \log \left[ 1 + \left( \frac{\omega_s}{\omega_c} \right)^{2n} \right]$$

Or alternatively:

$$\omega_c = \frac{\omega_p}{\left[ 10^{-\hat{G}_p/10} - 1 \right]^{1/2n}} \quad \& \quad \omega_c = \frac{\omega_s}{\left[ 10^{-\hat{G}_s/10} - 1 \right]^{1/2n}}$$

Solving for $n$ gives:

$$n = \frac{\log \left( \left( 10^{-\hat{G}_s/10} - 1 \right) / \left( 10^{-\hat{G}_p/10} - 1 \right) \right)}{2 \log(\omega_s/\omega_p)}$$

PS. See Lathi 4.10 (p. 453) for an example in MATLAB

---

Chebyshev Filters

- **equal-ripple:**
  - Because all the ripples in the passband are of equal height
  - If we reduce the ripple, the passband behaviour improves, but it does so at the cost of stopband behaviour
Chebyshev Filters

- Chebyshev Filters: Provide tighter transition bands (sharper cutoff) than the same-order Butterworth filter, but this is achieved at the expense of inferior passband behavior (rippling).

⇒ For the lowpass (LP) case: at higher frequencies (in the stopband), the Chebyshev filter gain is smaller than the comparable Butterworth filter gain by about $6(n - 1)$ dB.

- The amplitude response of a normalized Chebyshev lowpass filter is:

$$|\mathcal{H}(j\omega)| = \frac{1}{\sqrt{1 + \epsilon^2 C_n^2(\omega)}}$$

Where $C_n(\omega)$, the nth-order Chebyshev polynomial, is given by:

$$C_n(\omega) = \cos(n \cos^{-1}(\omega))$$

and where $C_n$ is given by:

$$C_n = \begin{cases} 
0 & n = 1 \\
1 & n = \omega \\
2\omega^2 - 1 & n = 2 \\
4\omega^4 - 3\omega^2 + 1 & n = 3 \\
16\omega^6 - 20\omega^4 + 5\omega^2 & n = 4 \\
32\omega^8 - 48\omega^6 + 18\omega^2 - 1 & n = 5 \end{cases}$$

Normalized Chebyshev Properties

- It’s normalized: The passband is $0 < \omega < 1$.
- **Amplitude response**: has ripples in the passband and is smooth (monotonic) in the stopband.
- **Number of ripples**: there is a total of $n$ maxima and minima over the passband $0 < \omega < 1$.

$$C_n^2(0) = \begin{cases} 
0, & n: odd \\
1, & n: even 
\end{cases} \quad |H(0)| = \begin{cases} 
1, & n: odd \\
\frac{1}{\sqrt{1 + \epsilon^2}}, & n: even 
\end{cases}$$

- $\epsilon$: ripple height $\rightarrow r = \sqrt{1 + \epsilon^2}$

- The Amplitude at $\omega = 1$: $\frac{1}{r} = \frac{1}{\sqrt{1 + \epsilon^2}}$

- For Chebyshev filters, the ripple $r$ dB takes the place of $G_p$. 


Determination of Filter Order

- The gain is given by: \[ \hat{G} = -10 \log [1 + \epsilon^2 C_n^2(\omega)] \]
  Thus, the gain at \( \omega_s \) is: \[ \epsilon^2 C_n^2(\omega_s) = 10^{-\hat{G_s}/10} - 1 \]

- Solving:
  \[
  n = \frac{1}{\cosh^{-1}(\omega_s)} \cosh^{-1} \left[ \frac{10^{-\hat{G_s}/10} - 1}{10^{\hat{G_s}/10} - 1} \right]^{1/2}
  \]

- General Case:
  \[
  n = \frac{1}{\cosh^{-1}(\omega_s/\omega_p)} \cosh^{-1} \left[ \frac{10^{-\hat{G_s}/10} - 1}{10^{\hat{G_s}/10} - 1} \right]^{1/2}
  \]

Chebyshev Pole Zero Diagram

- Whereas Butterworth poles lie on a semi-circle, the poles of an \( n \)th-order normalized Chebyshev filter lie on a semiellipse of the major and minor semiaxes:
  \[ a = \sinh \left( \frac{1}{n} \sinh^{-1} \left( \frac{1}{\epsilon} \right) \right) \quad \& \quad b = \cosh \left( \frac{1}{n} \sinh^{-1} \left( \frac{1}{\epsilon} \right) \right) \]

And the poles are at the locations:

\[
H(s) = \frac{1}{(s - s_1)(s - s_2) \ldots (s - s_n)}
\]

\[
s_k = -\sin \left( \frac{(2k-1)\pi}{2n} \right) \sinh x + j \cos \left( \frac{(2k-1)\pi}{2n} \right) \cosh x, \quad k = 1, \ldots, n
\]
Ex: Chebyshev Pole Zero Diagram for $n=3$

Procedure:
1. Draw two semicircles of radii $a$ and $b$ (from the previous slide).
2. Draw radial lines along the corresponding Butterworth angles ($\pi/n$) and locate the $n^{th}$-order Butterworth poles (shown by crosses) on the two circles.
3. The location of the $k^{th}$ Chebyshev pole is the intersection of the horizontal projection and the vertical projection from the corresponding $k^{th}$ Butterworth poles on the outer and the inner circle, respectively.

Chebyshev Values / Table

\[
\mathcal{H}(s) = \frac{K_n}{C'(s)} = \frac{K_n}{s^n + a_{n-1}s^{n-1} + \ldots + a_1s + a_0}
\]

\[
K_n = \begin{cases} 
  a_0 & \text{n odd} \\
  \frac{a_0}{\sqrt{1+\varepsilon^2}} = \frac{a_0}{10^{\varepsilon/20}} & \text{n even}
\end{cases}
\]

\[
\begin{array}{cccccc}
 n & a_0 & a_1 & a_2 & a_3 \\
 1 & 1.9652267 & & & & \\
 2 & 1.1025103 & 1.0977343 & & & \\
 3 & 0.4913067 & 1.2384092 & 0.9883412 & & \\
 4 & 0.2756276 & 0.7426194 & 1.4539248 & 0.9528114 &
\end{array}
\]

1 db ripple ($\varepsilon = 1$)
**Other Filter Types:**

**Chebyshev Type II = Inverse Chebyshev Filters**

- Chebyshev filters passband has ripples and the stopband is smooth.
- **Instead:** this has passband have smooth response and ripples in the stopband.

- Exhibits maximally flat passband response and equi-ripple stopband
- **Cheby2** in MATLAB

\[
|\eta(\omega)|^2 = 1 - \left|\eta_C(1/\omega)\right|^2 = \frac{e^2C_n^2(1/\omega)}{1 + e^2C_n^2(1/\omega)}
\]

Where: \(H_i\) is the Chebyshev filter system from before

- Passband behavior, especially for small \(\omega\), is **better** than Chebyshev
- **Smallest transition band** of the 3 filters (Butter, Cheby, Cheby2)
- Less time-delay (or phase loss) than that of the Chebyshev
- Both needs the **same order** \(n\) to meet a set of specifications.
- $$$ (or number of elements):
  - Cheby < Inverse Chebyshev < Butterworth (of the same performance [not order])

---

**Other Filter Types:**

**Elliptic Filters (or Cauer) Filters**

- Allow ripple in both the passband and the stopband,
  - we can achieve **tighter** transition band

\[
|\eta(j\omega)| = \frac{1}{\sqrt{1 + R_n^2(\omega)}}
\]

Where: \(R_n\) is the \(n^{th}\)-order Chebyshev rational function determined from a given ripple spec.
- \(\epsilon\) controls the ripple
- \(G_p = \frac{1}{\sqrt{1 + \epsilon^2}}\)

- Most efficient (\(\eta\))
  - the **largest ratio** of the passband gain to stopband gain
  - or for a given ratio of passband to stopband gain, it requires the **smallest transition band**

- in MATLAB: \texttt{ellipord} followed by \texttt{ellip}
### In Summary

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Passband Ripple</th>
<th>Stopband Ripple</th>
<th>Transition Band</th>
<th>MATLAB Design Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Butterworth</td>
<td>No</td>
<td>No</td>
<td>Loose</td>
<td>butter</td>
</tr>
<tr>
<td>Chebyshev</td>
<td>Yes</td>
<td>No</td>
<td>Tight</td>
<td>cheby</td>
</tr>
<tr>
<td>Chebyshev Type II (Inverse Chebyshev)</td>
<td>No</td>
<td>Yes</td>
<td>Tight</td>
<td>cheby2</td>
</tr>
<tr>
<td>Elliptic</td>
<td>Yes</td>
<td>Yes</td>
<td>Tightest</td>
<td>ellip</td>
</tr>
</tbody>
</table>

**Almost there: (digital) Signal Types!**
Impulse Response of Both Types

\[ y[n] = \frac{1}{2} u[n - 1] + \frac{1}{2} u[n] \]

\[ h[n] \]

‘Finite impulse response’ (FIR)

\[ y[n] = \frac{1}{2} y[n - 1] + u[n] \]

\[ h[n] \]

‘Infinite impulse response’ (IIR)

Digital Filters Types

\textbf{FIR}

\textbf{From H(z):}

\[ H(\omega) = h_0 + h_1 e^{-i\omega} + \cdots + h_{n-1} e^{-i(n-1)\omega} \]

\[ = \sum_{t=0}^{n-1} h_t \cos t\omega - i \sum_{t=0}^{n-1} h_t \sin t\omega \]

\textbf{Filter becomes a “multiply, accumulate, and delay” system:}

\[ y(t) = \sum_{t=0}^{n-1} h_t u(t - \tau) \]

\[ y[n] = b_0 x[n] + b_1 x[n - 1] + \cdots + b_N x[n - N] \]

\textbf{IIR}

\textbf{Impulse response} function that is non-zero over an infinite length of time.
**FIR Properties**

- Require no feedback.
- Are inherently stable.
- They can easily be designed to be **linear phase** by making the coefficient sequence symmetric
- Flexibility in shaping their magnitude response
- Very Fast Implementation (based around FFTs)

- The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or **selectivity**, especially when low frequency (relative to the sample rate) cutoffs are needed.

---

**FIR as a class of LTI Filters**

- Transfer function of the filter is
  \[
  H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}
  \]

- Finite Impulse Response (FIR) Filters: (N = 0, no feedback)

  \[
  H(\omega) = h_0 + h_1 e^{-i\omega} + \ldots + h_{n-1} e^{-i(n-1)\omega}
  \]

  \[
  = \sum_{t=0}^{n-1} h_t \cos t\omega - i \sum_{t=0}^{n-1} h_t \sin t\omega
  \]

  \[\because H(\omega) \text{ is periodic and conjugate} \]
  \[\because \text{Consider } \omega \in [0, \pi] \]
FIR Filters

- Let us consider an FIR filter of length $M$
- Order $N=M-1$ (watch out!)
- Order $\rightarrow$ number of delays

\[ y(n) = \sum_{k=0}^{M-1} b_k x(n-k) = \sum_{k=0}^{M-1} h(k) x(n-k) \]

FIR Filters

- Obtained immediately with $x(n) = \delta(n)$:

\[ h(n) = y(n) = \sum_{k=0}^{M-1} b_k \delta(n-k) = b_n \]

- The impulse response is of finite length $M$ (good!)

- FIR filters have only zeros (no poles) (as they must, N=0 !!)
  - Hence known also as all-zero filters

- FIR filters also known as feedforward or non-recursive, or transversal filters
FIR & Linear Phase

- The phase response of the filter is a linear function of frequency.

- Linear phase has constant group delay, all frequency components have equal delay times. No distortion due to different time delays of different frequencies.

- FIR Filters with:

\[
\sum_{n=-\infty}^{\infty} h[n] \cdot \sin(\omega \cdot (n - \alpha) + \beta) = 0
\]

FIR & Linear Phase → Four Types

<table>
<thead>
<tr>
<th>Impulse response</th>
<th># coeffs</th>
<th>( H(\omega) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h(n) = h(M - 1 - n) )</td>
<td>Odd</td>
<td>[ e^{-j(M-1)/2} \left( h \left( \frac{M-1}{2} \right) + 2 \sum_{k=1}^{M-3/2} h \left( \frac{M-1}{2} - k \right) \cos(\omega k) \right) ]</td>
</tr>
<tr>
<td>( h(n) = h(M - 1 - n) )</td>
<td>Even</td>
<td>[ e^{-j(M-1)/2} \sum_{k=1}^{M-3/2} h \left( \frac{M-1}{2} - k \right) \cos(\omega k - \frac{\pi}{4}) ]</td>
</tr>
<tr>
<td>( h(n) = -h(M - 1 - n) )</td>
<td>Odd</td>
<td>[ e^{-j(M-1)/2} \left( 2 \sum_{k=1}^{M-3/2} h \left( \frac{M-1}{2} - k \right) \sin(\omega k) \right) ]</td>
</tr>
<tr>
<td>( h(n) = -h(M - 1 - n) )</td>
<td>Even</td>
<td>[ e^{-j(M-1)/2} \sum_{k=1}^{M-3/2} h \left( \frac{M-1}{2} - k \right) \sin(\omega k - \frac{\pi}{4}) ]</td>
</tr>
</tbody>
</table>

- Type 1: most versatile
- Type 2: frequency response is always 0 at \( \omega = \pi \) (not suitable as a high-pass)
- Type 3 and 4: introduce a \( \pi/2 \) phase shift, 0 at \( \omega = 0 \) (not suitable as a high-pass)
Digital Filters $\Rightarrow$ DTFT

- First Thought:
  - How to get DTFT? FFT?
  - Slightly Naïve $\because$
    - $H(\omega)$ cannot be exactly zero over any band of frequencies
      (Paley-Wiener Theorem)

Lathi, p. 621
The frequency response is limited to $2\pi$

- DTFT is a convolution responses in time domain...
  \[
  \mathcal{F}\{x \ast h\} = \mathcal{F}\{x\} \cdot \mathcal{F}\{h\}
  \]
  \[
  y[n] = x[n] \ast h[n] = \mathcal{F}^{-1}\{X(\omega) \cdot H(\omega)\},
  \]

DTFT $\Rightarrow$ z-Transform

The above results motivate the definitions of the $z$ transform, the discrete-time Fourier transform (DTFT), and the discrete Fourier series (DFS) to be presented in this chapter and the next. In particular, if the basis functions for the input can be enumerated as

\[
\phi_k[n] = e^{j2\pi kn}
\]

that is, if $x(t)$ can be expressed in the form of Eq. (6.1.1) as

\[
x[n] = \sum_k x_k \phi_k[n]
\]

then the corresponding output is simply, from Eqs. (6.1.2) and (6.1.3),

\[
y[n] = \sum_k x_k H[n] \phi_k[n]
\]
The Discrete-Time Fourier Transform

• **Synthesis:**

The function \( X(e^{j\Omega}) \) defined by

\[
X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}
\] (7.1.1)

(if it converges) is called the *discrete-time Fourier transform (DTFT)* of the signal \( x[n] \). In particular, if the region of convergence for the \( z \)-transform

\[
X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}
\]

includes the unit circle, then the DTFT equals \( X(z) \) evaluated on the unit circle, that is,

\[
x[n] = X(z)|_{z=e^{j\Omega}}
\] (7.1.2)

• **Analysis/Inverse:**

\[
x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\Omega})e^{j\Omega n} \, d\Omega
\]

• \( x[n] \) is the (limiting) sum of sinusoidal components of the form \( \left[ \frac{1}{2\pi} X(e^{j\Omega}) \right] e^{j\Omega n} \)

• Together: Forms the DTFT Pair
The Discrete-Time Fourier Transform

- Ex:
  \[ x[n] = a^n u[n] \]
  has the z transform
  \[ X(z) = \frac{1}{1 - az^{-1}}, \quad |z| > |a|, \]
  and thus \( X(e^{j\omega}) \) exists for \( |a| < 1 \) because the ROC then contains the unit circle. Specifically,
  \[ X(e^{j\omega}) = \frac{1}{1 - ae^{-j\omega}}, \quad |a| < 1. \quad (7.1.8) \]
  The corresponding magnitude spectrum \( |X(e^{j\omega})| \) and phase spectrum \( \angle X(e^{j\omega}) \) are shown in Fig. 6.8. Clearly, from the defining sum in Eq. (7.1.1), the DTFT of \( x[n] \) does not converge for \( |a| > 1 \), and we defer until later the case of \( |a| = 1 \).
  On the other hand, the anticausal exponential
  \[ u[n] = -a^n u[-n - 1] \]
  has the z transform
  \[ W(z) = \frac{1}{1 - az^{-1}}, \quad |z| < |a|, \]
  and thus \( W(e^{j\omega}) \) exists for \( |a| > 1 \), but not for \( |a| < 1 \). That is,
  \[ W(e^{j\omega}) = \frac{1}{1 - ae^{-j\omega}}, \quad |a| > 1. \quad (7.1.9) \]
  Again the case of \( |a| = 1 \) is deferred until later.

The Discrete-Time Fourier Transform

- Observe:
  “Kinship Of Difference Equations To Differential Equations”

\[ y[n] = x[n] + x[n-1] + \dots + x[n-N+1] \]

Consider uniform samples of \( x(t) \) at intervals of \( T \) seconds. As usual, we use the notation \( x[n] \) to denote \( x(nT) \), the \( n \)th sample of \( x(t) \). Similarly, \( y[n] \) denotes \( y(nT) \), the \( n \)th sample of \( y(t) \). From the basic definition of a derivative, we can express Eq. (3.15a) at \( t = nT \) as

\[ y[n] = y[n] + y[n-1] + \dots + y[n-N+1] \]

Where
\[ \frac{d}{dt} \] and
\[ \frac{d}{dt} \] operator form as

\[ x[n+1] = x[n] \]
The Discrete-Time Fourier Transform

- Ex(2): The DTFT of the real sinusoid

\[ x[n] = \text{sin} (\Omega_0 n) = \frac{1}{2j} (e^{j\Omega_0 n} - e^{-j\Omega_0 n}) \]

is simply

\[ X(e^{j\Omega}) = 2\pi \left[ \frac{1}{2j} \delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0) \right] \]

\[ = \pi \delta(\Omega - \Omega_0) - \pi \delta(\Omega + \Omega_0) \]

for \(|\Omega_0| \leq \pi\), while that of the cosine signal,

\[ x[n] = \cos (\Omega_0 n) = \frac{1}{2} (e^{j\Omega_0 n} + e^{-j\Omega_0 n}) \]

is likewise

\[ X(e^{j\Omega}) = 2\pi \left[ \frac{1}{2} \delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0) \right] \]

\[ = \pi \delta(\Omega - \Omega_0) + \pi \delta(\Omega + \Omega_0) \]

In addition, the DTFT pair for the dc signal \( x[n] = 1 \) is simply

\[ 1 \Leftrightarrow 2\pi \delta(\Omega), \quad |\Omega| \leq \pi \]

as opposed to the dual relationship

\[ \delta[n] \Leftrightarrow 1, \quad \forall \Omega \]
Now: (digital) Filters!

Flashback: Fourier Series & Rectangular Functions

\[ \tilde{f}^{-1}\left\{ \text{rect}\left(\frac{\omega}{2}\right) \right\} = \frac{sinc(t)}{\pi} \]

\[ \tilde{f}\left\{ \text{rect}(t) \right\} = \text{sinc}\left(\frac{\omega}{2}\right) \]

See:

- Table 7.1 (p. 702) Entry 17
- Table 9.1 (p. 852) Entry 7
**Flashback**: Fourier Series & Rectangular Functions

- The sinc function might look familiar
  - This is the frequency content of a square wave (box)

![Image of sinc function]

- This also applies to **signal reconstruction!**

  ➜ **Whittaker–Shannon interpolation formula**
  
  - This says that the “better way” to go from Discrete to Continuous (i.e. D to A) is not ZOH, but rather via the sinc!

\[
x(t) = \sum_{n=-\infty}^{\infty} x[n] \cdot \text{sinc} \left( \frac{t-nT}{T} \right)
\]

---

**Filter Design**

- Previously we have analysed
  - difference equations (y[n])
  - transfer functions (H(z))

- To obtain time/frequency domain response
  - Impulse (h[n]) or frequency (H(w)) response

- Now we have a specification
  - frequency response (filters)
  - time response (control)

- Goal to design a filter that meets specification
  - i.e., determine transfer function
  - and therefore difference equation (implementation)
Filter Specification in the Frequency Domain

Specified
- $\delta_1$ = passband ripple (dB)
- $\delta_2$ = stopband attenuation (dB)
- $w_p$ = passband edge (Hz)
- $w_{st}$ = stopband edge (Hz)

Calculated
- $w_c$ = cutoff frequency (@ 3dB)
- filter type/order to meet specification

| $|H(w)|$ |
|-------|
| $\delta_1$ |
| $\delta_2$ |

Passband Transition Stopband

Transfer Function $\rightarrow$ Difference Equation

- Example, consider

$$H(z) = \frac{z^2 - 0.2z - 0.08}{z^2 + 0.5}$$

Make $H(z)$ causal $\times$ by $\frac{z^{-2}}{z^{-2}}$

- Normalise to negative powers of $z$ (causal)
  - re-arrange and take inverse $z$ transform

$$H(z) = \frac{1 - 0.2z^{-1} - 0.08z^{-2}}{1 + 0.5z^{-2}} = \frac{Y(z)}{X(z)}$$

$$Y(z)(1 + 0.5z^{-2}) = X(z)(1 - 0.2z^{-1} - 0.08z^{-2})$$

$y[n] + 0.5y[n-2] = x[n] - 0.2x[n-1] - 0.08x[n-2]$

$y[n] = x[n] - 0.2x[n-1] - 0.08x[n-2] - 0.5y[n-2]$
Two LTI filters in cascade:
1. feedforward (a_i)
   • forms x'[n]
2. feedback (b_i)
   • forms y[n]

Filters are linear so can swap order. Redundant time delays (z^{-1}) as A=A', B=B' and C=C'

Note: y'[n] ≠ x'[n] of previous slide BUT y[n] = y[n] so, same filter.
Direct form II:
Canonical form of realisation (minimum memory)

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

Re-arranging in terms of output

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

Which as a difference equation is

\[ y(n) = \sum_{i=0}^{N} a_i y'(n-i) \]

Direct II

\[ y'(n) = x(n) + \sum_{i=1}^{M} b_i y'(n-i) \]

Remember

Direct I

\[ y(n) = \sum_{i=0}^{N} a_i x(n-i) + \sum_{i=1}^{M} b_i y(n-i) \]

Canonical terms

\[ A', B', C' \]
### Canonical Realisation

- **Direct Form I**
  - Conceptually simplest realisation
  - Often less susceptible to noise
- **Canonical/Direct Form II**
  - Minimum memory (storage)
- **Filter design**
  - Determine value of filter coefficients (all ai & bi)
  - Poles controlled by bi coefficients
    - if any bi ≠ 0 then filter IIR (recursive)
    - if all bi = 0 then filter FIR (non-recursive)
  - Zeros controlled by ai coefficients

### Cascade Form

- Transfer function factorised to
  - Product of second order terms \( H_n(z) \)
  - \( C \) is a constant (gain)

\[
H(z) = C \prod_{n=1}^{N} H_n(z)
\]

- Most common realisation
  - Often assumed by many filter design packages
  - many 2\textsuperscript{nd} order sections have integer coefficients
Parallel Form

- Transfer function expressed as
  - partial fraction expansion of second order terms

\[ H(z) = C + \sum_{n=1}^{N} H_n(z) \]

Bi-quadratic Digital Filter

- Canonic form of Second order system
- 2nd order, system ‘building block’

Difference equation:

\[ y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + b_1 y[n-1] + b_2 y[n-2] \]
## IIR Filter Design Methods

- Normally based on analogue prototypes
  - Butterworth, Chebyshev, Elliptic etc
- Then transform $H(s) \rightarrow H(z)$
- Three popular methods:
  - Impulse invariant
    - produces $H(z)$ whose impulse response is a sampled version of $h(t)$ (also step invariant)
  - Matched z – transform
    - poles/zeros $H(s)$ directly mapped to poles/zeros $H(z)$
  - Bilinear z – transform
    - left hand $s$ – plane mapped to unit circle in $z$ - plane

## Impulse Invariant

- Simplest approach, proceeds as follows,
- Select prototype analogue filter
- Determine $H(s)$ for desired $wc$ and $ws$
- Inverse Laplace,
  - i.e., calculate impulse response, $h(t)$
- Sample impulse response $h(t)|t=n\Delta t$
  - $h[n] = \Delta t h(n\Delta t)$
- Take $z$ - transform of $h[n] \Rightarrow H(z)$
  - poles, p1 map to $\exp(p1\Delta t)$ (maintains stability)
  - zeros have no simple mapping
**Impulse Invariant**

- Useful approach when
  - Impulse (or step) invariance is required, or
    - e.g., control applications
  - Designing Lowpass or Bandpass filters
- Has problems when
  - $H(w)$ does not $\to 0$ as $w \to \infty$
  - i.e., if $H(w)$ is not bandlimited, aliasing occurs
  - e.g., highpass or bandstop filters

**Matched z - transform**

- Maps poles/zeros in $s$ – plane directly
  - to poles/zeros in $z$ – plane
- No great virtues/problems
- Fairly old method
  - not commonly used
  - so we won’t consider it further
Bilinear z - transform

- Maps complete imaginary s -plane (±∞)
  - to unit circle in z -plane
- i.e., maps analogue frequency wa to
  - discrete frequency wd
- uses continuous transform,

\[ w_a = \frac{2}{\Delta t} \tan \left( \frac{w_d \Delta t}{2} \right) \]

This compresses (warps) \( w_a \) to have finite extent \( \pm w_d/2 \)
i.e., this removes possibility of any aliasing 😊
### Bilinear Transform

The bilinear transform

\[ \omega_n = \frac{2}{\Delta t} \tan\left( \frac{\omega_d \Delta t}{2} \right) \]

Transforming to s-domain

Remember: \( s = j\omega_a \)

and \( \tan \theta = \sin \theta / \cos \theta \)

Where \( \theta = \omega_d \Delta t / 2 \)

Using Euler’s relation

This becomes...

(\( \text{note: } j \text{ terms cancel} \))

\[ s = \frac{2}{\Delta t} \frac{\sqrt{2}(\exp(-j\omega_a \Delta t) - \exp(-j\omega_d \Delta t))}{\sqrt{2}(\exp(-j\omega_a \Delta t) + \exp(-j\omega_d \Delta t))} \]

\[ s = \frac{2}{\Delta t} \frac{(1 - \exp(-j\omega_d \Delta t))}{(1 + \exp(-j\omega_d \Delta t))} \]

\[ s = \frac{2(1 - z^{-1})}{\Delta t(1 + z^{-1})} \]

As \( z = \exp(s_o \Delta t) = \exp(j\omega_a \Delta t) \)

---

### Bilinear Transform

- Convert \( H(s) \Rightarrow H(z) \) by substituting,

\[ s = \frac{2(1 - z^{-1})}{\Delta t(1 + z^{-1})} \]

- However, this transformation compresses the frequency response, which means
  - digital cut off frequency will be lower than the analogue prototype
- Therefore, analogue filter must be “pre-warped” prior to transforming \( H(s) \Rightarrow H(z) \)

Note: this comes directly from tan transform
Bilinear Pre-warping

\[
\omega_u = \frac{2}{\Delta t} \tan \left( \frac{\omega_d \Delta t}{2} \right)
\]

Bilinear Transform: Example

- Design digital Butterworth lowpass filter
  - order, \( n = 2 \), cut off frequency \( \omega_d = 628 \text{ rad/s} \)
  - sampling frequency \( \omega_s = 5024 \text{ rad/s (800Hz)} \)
- pre-warp to find \( \omega_a \) that gives desired \( \omega_d \)
  \[
  \omega_a = \left( \frac{2}{1/800} \right) \tan \left( \frac{628}{2 \times 800} \right) = 663 \text{ rad/s}
  \]
  Note: \( \omega_d < \omega_a \) due to compression

\[
H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}
\]
Bilinear Transform: Example

- De-normalised analogue prototype \( s' = s/\omega_c \)
  - \( \omega_c = 663 \text{ rad/s} \) (required \( \omega_a \) to give desired)

\[
H(s_a) = \frac{1}{\left( \frac{s}{663} \right)^2 + \frac{\sqrt{2}s}{663} + 1}
\]

- Convert \( H(s) \Rightarrow H(z) \) by substituting

\[
s = \frac{2(1 - z^{-1})}{\Delta t(1 + z^{-1})}
\]

\[
H(z) = \frac{1}{\left( \frac{2 \times 800(1 - z^{-1})}{663(1 + z^{-1})} \right)^2 + \sqrt{2} \left( \frac{2 \times 800(1 - z^{-1})}{663(1 + z^{-1})} \right) + 1}
\]

\[
H(z) = \frac{0.098z^2 + 0.195z + 0.098}{z^2 - 0.942z + 0.333}
\]

Note: \( H(z) \) has both poles and zeros

\( H(s) \) was all-pole

Finally, apply inverse z-transform to yield the difference equation:

\[
y[n] = 0.098x[n] + 0.195x[n - 1] + 0.098x[n - 2]
\]

\[
+ 0.942y[n - 1] - 0.333y[n - 2]
\]
Bilinear Transform: Example

1. same cut off frequency,
2. increased roll off and attenuation in stopband
3. \( \infty \) attenuation at \( \omega_c/2 \)
Bilinear Transform: Example

Phase response

Bilinear transform has effectively increased digital filter order (by adding zeros)

Increased phase delay

Bilinear Transform: Example

Canonical Implementation of the difference equation

\[
y[n] = 0.098y'[n] + 0.195y'[n-1] + 0.098y'[n-2] \\
y'[n] = x[n] + 0.942y'[n-1] - 0.333y'[n-2]
\]

of the difference equation

\[
y[n] = 0.098x[n] + 0.195x[n-1] + 0.098x[n-2] \\
+ 0.942y[n-1] - 0.333y[n-2]
\]
Bilinear Transform: Example

Bilinear Design Summary

- Calculate pre-warping analogue cutoff frequency
- De-normalise filter transfer function using pre-warping cut-off
- Apply bilinear transform and simplify
- Use inverse z-transform to obtain difference equation
Direct Synthesis

- Not based on analogue prototype
  - But direct placement of poles/zeros
- Useful for
  - First order lowpass or highpass
    - simple smoothers
  - Resonators and equalisers
    - single frequency amplification/removal
  - Comb and notch filters
    - Multiple frequency amplification/removal

First Order Filter: Example

- General first order transfer function
  - Gain, $G$, zero at $-b$, pole at $a$ ($a, b$ both < 1)

\[
H(z) = \frac{G(1+bz^{-1})}{1-az^{-1}}
\]

with $a$ +ve & $b$ –ve

Remember: $H(\omega) = H(z)|_{z = \exp(j\omega \Delta)}$

\[
H(0) = \frac{G(1+b)}{(1-a)}
\]

\[
H(\pi) = \frac{G(1-b)}{(1+a)}
\]

$\omega/2$

$z = \exp(j\omega)$

$1 = \exp(j0)$
First Order Filter: Example

- Possible design criteria
  - cut-off frequency, wc
    - $3\text{dB} = 20 \log(|H(w_c)|)$
    - e.g., at $w_c = \pi/2$, $(1+b)/(1+a) = \sqrt{2}$
  - stopband attenuation
    - assume $w_{\text{stop}} = \pi$ (Nyquist frequency)
    - e.g., $\delta_2 = H(\pi)/H(0) = 1/21$ i.e.,

$$\frac{H(\pi)}{H(0)} = \frac{(1-b)(1-a)}{(1+b)(1+a)} = \frac{1}{21}$$

- Two unknowns $(a, b)$
- Two (simultaneous) design equations.

Digital Resonator

- Second order ‘resonator’
  - single narrow peak frequency response
  - i.e., peak at resonant frequency, $w_0$
Quality factor (Q-factor)

- Dimensionless parameter that compares
  - Time constant for oscillator decay/bandwidth ($\Delta \omega$) to
  - Oscillation (resonant) period/frequency ($\omega_0$)
    - High Q = less energy dissipated per cycle
    
    $$Q = \frac{\omega_0}{\Delta \omega} = \frac{f_0}{\Delta f}$$

- Alternative to damping factor ($\zeta$) as

  $$Q = \frac{1}{2 \zeta}$$

  $$H(s) = \frac{s + \omega_0^2}{s^2 + 2 \zeta \omega_0 s + \omega_0^2} = \frac{s + \omega_0^2}{s^2 + \frac{\omega_0^2}{Q}}$$

- Note: $Q < \frac{1}{2}$ overdamped (not an oscillator)

Digital Resonator Design

- To make a peak at $\omega_0$ place pole
  - Inside unit circle (for stability)
  - At angle $\omega_0$ distance $R$ from origin
    - i.e., at location $p = R \exp(j\omega_0)$
      - $R$ controls $\Delta \omega$
        » Closer to unit circle $\rightarrow$ sharper peak
    - plus complex conj pole at $p^* = R \exp(-j\omega_0)$

  $$H(z) = \frac{1}{1 - R \cdot \exp(j\omega_0) z^{-1}} \frac{1}{1 - R \cdot \exp(-j\omega_0) z^{-1}}$$

  $$= \frac{1 - R(\exp(j\omega_0) + \exp(-j\omega_0))z^{-1} + R^2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

  Where (via Euler's relation)

  $$a_1 = -2R \cos(\omega_0) \quad \text{and} \quad a_2 = R^2$$
Discrete Filter Transformations

- By convention, design Lowpass filters
  - transform to HP, BP, BS, etc
- Simplest transformation
  - Lowpass $H(z') \rightarrow$ highpass $H(z)$
  - $H_{HP}(z) = H_{LP}(z)|z' \rightarrow -z$
    - reflection about imaginary axis $(\omega s/4)$
    - changing signs of poles and zeros
- LP cutoff frequency, $w_{CLP}$ becomes
  - HP cut-in frequency, $w_{CHP} = \frac{1}{2} - w_{CLP}$

---

Lowpass $\rightarrow$ highpass ($z' = -z$)

**z - plane**

- Lowpass prototype
  - $p_L = \frac{1}{4}$, $z_L = -1$
  - Poles/zeros reflected in imaginary axis: $w_{CHP} = \frac{1}{2} - w_{CLP}$
  - Same gain @ $w_s/4$ (i.e., $\pi/4$)
  - $|H(w_{HP})| = |H(\pi/2 - w_{LP})|$

- Highpass transform
  - $p_H = -\frac{1}{4}$, $z_H = 1$
### Discrete Filter Transformations

- **Lowpass** $H(z') \rightarrow \text{highpass } H(z)$  
  - Cut-off (3dB) frequency $= wc$ (remains same)

  \[
  z' = \frac{\cos(w_c \Delta t) - z}{1 - \cos(w_c \Delta t)z} 
  \]

- **Lowpass** $H(z') \rightarrow \text{Bandpass } H(z)$  
  - Centre frequency $= w_0$ & 3dB bandwidth $= wc$

  \[
  z' = \frac{\alpha z - z^2}{-\alpha z + 1} \quad \alpha = \frac{\cos(w_0 \Delta t)}{\cos(w_c \Delta t)} 
  \]

*Note: these are not the only possible BP and BS transformations!*

---

- **Lowpass** $H(z') \rightarrow \text{Bandstop } H(z)$  
  - Centre frequency $= w_0$ 3dB bandwidth $= wc$

  \[
  z' = \frac{z^2 - (2\alpha/(k+1))z + (1-k)/(1+k)}{1 + (2\alpha/(k+1))z + ((1-k)/(1+k))z^2} 
  \]

  \[
  \alpha = \frac{\cos(w_0 \Delta t)}{\cos(w_c \Delta t)} \quad k = \tan^2(w_c \Delta t) 
  \]

*Note: order doubles for bandpass/bandstop transformations*
Summary

- Digital Filter Structures
  - Direct form (simplest)
  - Canonical form (minimum memory)
- IIR filters
  - Feedback and/or feedforward sections
- FIR filters
  - Feedforward only
- Filter design
  - Bilinear transform (LP, HP, BP, BS filters)
  - Direct form (resonators and notch filters)
  - Filter transformations (LP $\rightarrow$ HP, BP, or BS)
- Stability & Precision improved
  - Using cascade of 1st/2nd order sections
• Digital Filters

• Review:
  – Chapter 10 of Lathi

• A signal has many signals 😊
  [Unless it’s bandlimited. Then there is the one ω]