OCTOBER 3, 1st

1. Midterm II November 14 th
2. Assign project on Nov 7 th

Today: a few words on multirate

1. Using compressors and expanders can yield
   savings in computation

4.7.1: interchange filtering with compressor/expander

Claim:

\[ x[n] \xrightarrow{\text{JM}} H(z) \xrightarrow{} y[n] \equiv x[n] \rightarrow X(e^{j\omega}) \rightarrow H(e^{j\omega}) \rightarrow Y(e^{j\omega}) \]

Proof:

\[ Y(e^{j\omega}) = H(e^{j\omega})(\frac{1}{M} \sum_{i=0}^{M-1} x(i\omega - \frac{2\pi}{M})) \]

\[ = \frac{1}{M} \sum_{i=0}^{M-1} H(e^{j(i\omega - \frac{2\pi}{M})}) X(e^{j(i\omega - \frac{2\pi}{M})}) \]

\[ = \frac{1}{M} \sum_{i=0}^{M-1} V(e^{j(i\omega - \frac{2\pi}{M})}) = \tilde{V}(e^{j\omega}) \]

Result of compressor after JM
Summary

\[ x[n] \xrightarrow{\downarrow M} \{N\} \rightarrow y[n] \]
\[ x[n] \xrightarrow{H(z)} \downarrow M \rightarrow y[n] \]
\[ x[n] \xrightarrow{H(z)} \uparrow L \rightarrow y[n] \]
\[ x[n] \xrightarrow{\uparrow L} H(z) \rightarrow y[n] \]

Why useful?

Example:

\[ x[n] \rightarrow h(n) \rightarrow \bigoplus \rightarrow y[n] \rightarrow \]

\[ h(n) = h_0[n] + h_1[n] \]

\[ e_0[n] \rightarrow \uparrow L + e_1[n] \]

Or:

\[ x[n] \rightarrow \bigoplus \rightarrow E_0(z) \rightarrow \downarrow L \rightarrow \bigoplus \rightarrow \downarrow L \rightarrow \downarrow L \rightarrow E_1(z) \rightarrow + \rightarrow y[n] \]
complexity

1) straightforward 4 mult/clock

2) polyphase \( \Rightarrow \text{half clock rate} \times \text{half mult}/\text{filter} \times \)
\( \times \text{two filters} = \frac{4}{2} \times \frac{1}{2} = \frac{4}{4} = 2 \text{ mult}/\text{clock} \)

multi-rate appears in filter-banks \( \Rightarrow \) wavelets
C H 7: Filter design Techniques

Types of D.T Filters:

Infinite impulse response (IIR)

- Impulse response has infinite duration
- Implemented with recursion (feedback)

Pros

- fewer delays and multiplies for given performance

Cons

- more susceptible to errors and instability due to finite word length.
- Generally not linear phase (different delays for each frequency)

Finite Impulse Response (FIR)

- Response has finite duration
- No recursion

Pros

- Always stable! (poles only at \( z = 0 \))
- Less susceptible to quantization
- Can design linear phase
- Easier to design arbitrary filters.

Cons

- More delays and multiplies \( \rightarrow \) not a problem anymore
- Design can be more difficult \( \rightarrow \) not a problem
Filter specifications!

- \( \omega_p \) - pass band edge
- \( \omega_s \) - stop band edge
- \( \delta_p \) - peak band-pass ripple
- \( \delta_s \) - peak stop-band ripple

Ideal: \( \delta_p = \delta_s = 0 \), \( \omega_p = \omega_s \) \( \Rightarrow \) not realizable.
IIR Design

Historically → Continuosse IIR design was highly advanced use results from C.T to D.T

- C.T IIR designs have closed form → easy to use.
- With these designs, easy to control amp. response but not phase response.

Common types:

1) Butterworth → monotonic, no ripple
2) Chebyshev → type 1 pass-band ripple, type 2 stop-band ripple
3) Elliptic → Ripples in both s.s and p.s Has Zeros.
Design of DT IIR Filters from Analog

Discretize by using one of many techniques.

\[ H_c(s) \rightarrow H(z) \]

must satisfy:

1) Imaginary axis \( s \rightarrow \) mapped to unit circle

\[ s = ja \Rightarrow z = e^{j\omega} \]

2) Stability of \( H_c(s) \) should result in stable \( H(z) \)

Two methods:

i) Impulse invariance \( \rightarrow \) match impulse

\[ h[n] = T h_c(nT) \]

\[ H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} H_c(ja+jak) \big|_{a = \frac{\omega}{\pi}} \]

If \( H_c(ja) \) bandlimited \( |a| < \frac{\omega_s}{2} \) then:

\[ H(e^{j\omega}) = H_c(ja) \big|_{a = \frac{\omega}{\pi}} \]

Otherwise \( \rightarrow \) aliasing.
(c) Bilinear transform → basic idea.

(c) Distorts $H_e(j\omega)$ such that it is bandlimited.

$$s = \frac{1}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

$$z = \frac{1 + \frac{T}{2} s}{1 - \frac{T}{2} s}$$

for $s = 0 \Rightarrow z = 1$

$s = \infty \Rightarrow z = T$

Matlab: butter, cheby1, cheby2, ellip

Next time for ...