Lecture 8

Today:	FIR filter design IIR filter design Filter roundoff and overflow sensitivity
Announcements:	Team proposals are due tomorrow at 6PM Homework 4 is due next thur. Proposal presentations are next mon in 1311EECS.
References:	See last slide.

Please keep the lab clean and organized.

Last one out should close the lab door!!!!

We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil. — D. Knuth

Proposal presentations: Mon Sept 29

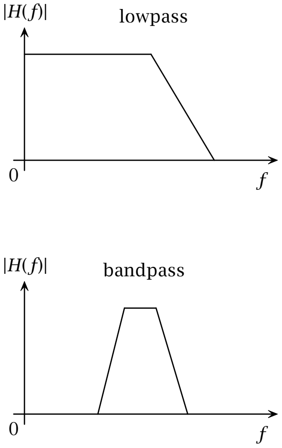
Schedule

- Presentations will occur from 6PM to 10:00PM in EECS 1311.
- Your team spokesperson must sign the team up for a 30 minute slot (20 min presentation).
- All team members must take part in their team's presentation.
- You may stay for any or all other portions of the presentation meeting.
- Team should arrive at least 20 minutes before their time slot.
- Team must use powerpoint or other projectable media for your presentations.
- The presentation must cover each section of the proposal.
- You should put your presentation on a thumb drive and/or email copy to *hero* before the meeting.

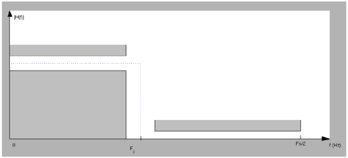
Digital filters: theory and implementation

- We have seen the need for several types of analog filters in A/D and D/A
  - Anti-aliasing filter
  - Reconstruction (anti-image) filter
  - Equalization filter
- Anti-aliasing and reconstruction require cts time filters
- Discrete time filters are used for spectral shaping post-digitization.
- There will be round-off error effects due to finite precision.

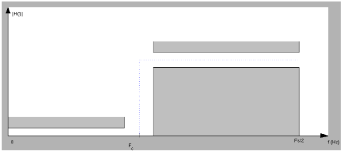
Different types of filter transfer functions



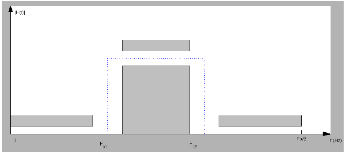
Matlab's fdatool for digital filter design



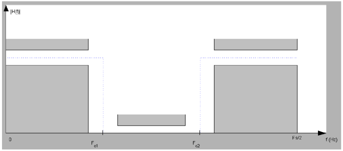
(a) LPF



(b) HPF



(c) BFP



(d) BSF

Figure: Lowpass, highpass, bandpass, bandstop (notch) in Matlab's `fdatool`

FIR vs IIR Digital filters

Output depends on current and previous M input samples.

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_M x[n-M]\,.$$

This is a FIR moving sum filter.

Output depends on current input and previous N filter outputs.

$$y[n] = x[n] - a_1 y[n-1] - a_2 y[n-2] - \cdots - a_N y[n-N]\,.$$

This is an IIR all-pole or autoregressive filter.

Output depends on current and previous M input samples and the previous N filter outputs.

$$\begin{aligned} y[n] &= b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + \cdots + b_M x[n-M] - \\ &\quad a_1 y[n-1] - a_2 y[n-2] - \cdots - a_N y[n-N]\,. \end{aligned}$$

This is the general pole-zero IIR digital filter equation.

Filter design procedure

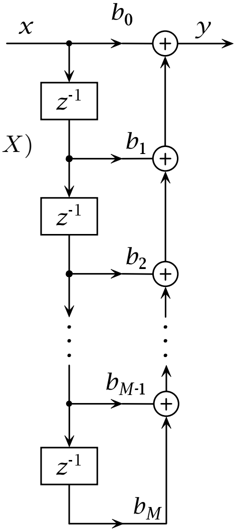
- Specification of filter requirements.
- Selection of FIR or IIR response.
- Calculation and optimization of filter coefficients.
- Realization of the filter by suitable structure.
- Analysis of finite word length effects on performance.
- Implementation.
- Testing/validation.

The above steps are generally not independent of each other. Filter design is usually an iterative process. The FIR–IIR response selection step is a major design choice.

FIR block diagram (again)

$$
\begin{aligned}
Y &= (b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M})X \\
&= b_0 X + b_1(z^{-1}X) + b_2(z^{-2}X) + \cdots + b_M(z^{-M}X)
\end{aligned}
$$

$$
\frac{Y}{X} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}
$$



This is sometimes referred to as the direct form (DF).

This implements well in a DSP with one or two MAC units. Can do all the MACs accumulating into a bit-rich accumulator. Once all the sums are formed truncate/round then saturate and finally use/store the result.

Well suited to a pipelined implementation

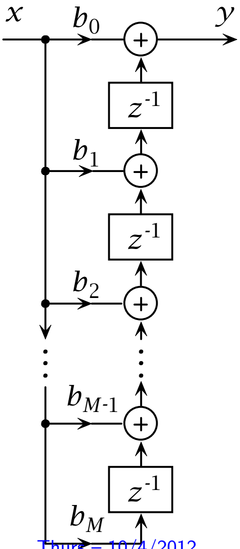
Transposed FIR block diagram

$$
\begin{aligned}
Y &= (b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M})X \\
&= b_0 X + (b_1 X) z^{-1} + (b_2 X) z^{-2} + \cdots + (b_M X) z^{-M}
\end{aligned}
$$

$$
\frac{Y}{X} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}
$$

This is sometimes referred to as the transposed direct form (TDF) or the broadcast form.

Well suited for cascade implementation.



FIR Direct form hardware implementation

Xilinx Application Note XAPP219 (v1.2) October 25, 2001

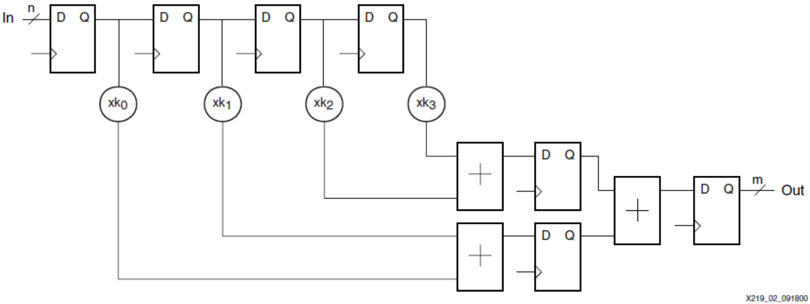


Figure 2: **FIR Filter Structure Employing Tree of Pipelined Adders**

FIR Transpose form hardware implementation

Xilinx Application Note XAPP219 (v1.2) October 25, 2001

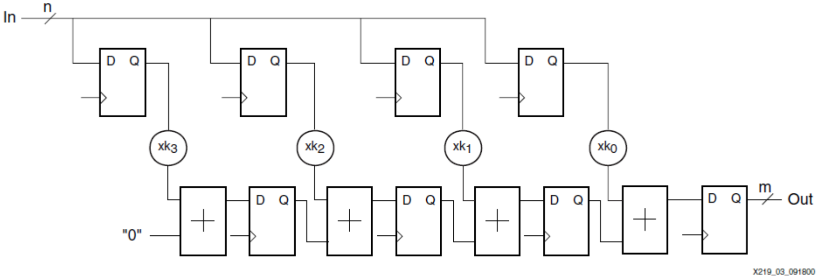


Figure 3: **Transposed Form FIR Filters Employing Cascaded Pipelined Adders**

Run time complexity?

Q. How many MULT and ADD operations are needed to calculate

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_{N-1} x[n-N]?$$

A. Could be as high as N ADDs and $N+1$ MULTs. However simplifications can occur

- May be able to group certain operations to reduce computations.
- Some coefficients may be equal, e.g., $b_0 = b_1 = \ldots = b_N$

  $$y[n] = b_0(x[n] + x[n-1] + \ldots + x[n-N])$$

  Only a single MULT required.
- Values of coefficients or data may be integer powers of two, e.g. $b_n = 2^{q_n}$. In this case MULTs can be performed by register shifts.

The running average filter

Running average filter ($b_0 = b_1 = b_2 = \cdots = b_N = 1/(N+1)$) has transfer function

$$H(z) = \frac{1 + z^{-1} + \cdots + z^{-N}}{N+1}.$$

This is the sum of a geometric series so has closed form

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

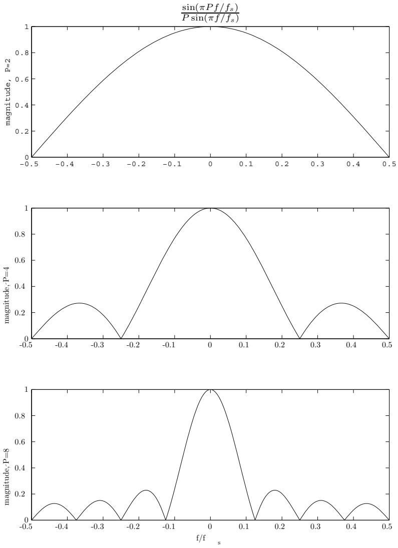
Expressing this in (digital) frequency domain ($z = e^{j2\pi f}$) gives

$$H(f) = \frac{1 - e^{-j2\pi(N+1)f}}{1 - e^{-j2\pi f}} \frac{1}{N+1} = e^{-j\pi N f} \frac{\sin[\pi(N+1)f]}{\sin(\pi f)} \frac{1}{N+1}.$$

Because of the periodicity of $e^{j2\pi f}$ we need only focus on range $-1/2 \leq f < 1/2$.

Note that $H(f)$ has **linear phase**

Running average filter magnitude



Number of FIR filter coefficients:

$$P = N + 1.$$

Distance to first zero: $1/P$.
Nominal bandwidth: $1/P$.
First side peak at: $3/(2P)$.
First lobe level:

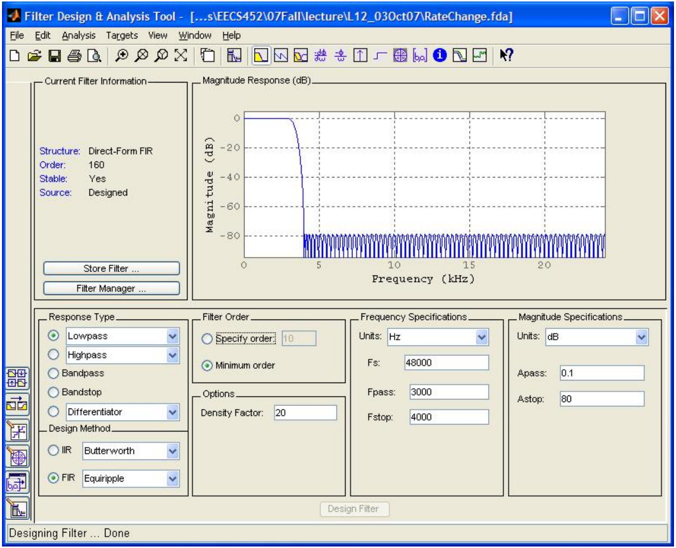
P	dB
4	-11.4
8	-13.0
16	-13.3
∞	-13.5

More general FIR filter design

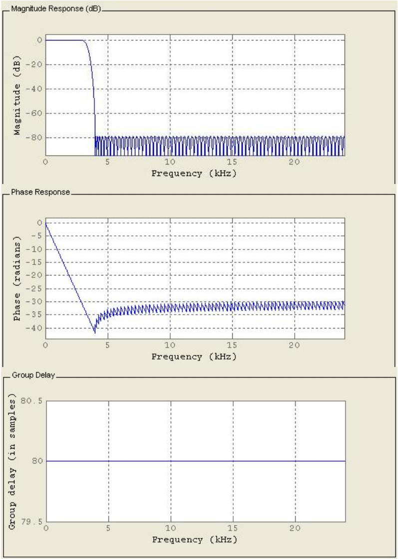
Recall our equiripple design example (Lecture 2):

- Low pass filter.
- f_s=48000 Hz.
- Bandpass ripple: ± 0.1 dB.
- Transition region 3000 Hz to 4000 Hz.
- Minimum stop band attenuation: 80 dB.

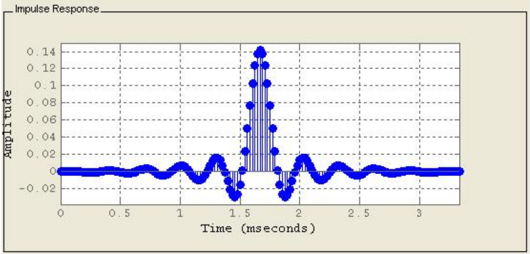
fdatool's solution



fdatool's magnitude, phase and group delay



Impulse response (coefficient values)



The filter impulse response has a delayed "peak"

Delay of peak is approximately 1.7 msecs

Delay corresponds to 80 integer units (1/2 of total length of filter). Note that the impulse response is symmetric about the peak

FIR filters can be designed with linear phase

Objective: design FIR filter whose magnitude response $|H(f)|$ meets constraints.
Can design filter to have linear phase over passband.

There are four FIR linear-phase *types* depending upon

- whether the number of coefficients is even or odd,
- whether the coefficients are even or odd symmetric.

Linear phase and FIR symmetry

Given M-th order FIR filter $h[n]$. Assume that $h[n]$ has even or odd symmetry about an integer m:

> **Even symmetry condition**: *There exists an integer m such that $h[m-n] = h[n]$.*
> **Odd symmetry condition**: *There exists an integer m such that $h[m-n] = -h[n]$.*

Then $h[n]$ is a linear phase FIR filter with transfer function.

$$H(f) = |H_m(f)|e^{-j2\pi fm + j\phi}$$

where $H_m(f)$ is the transfer function associated with $h_m[n] = h[n+m]$ and $\phi = 0$ if even symmetric while $\phi = \pi/2$ if odd symmetric.

Why? Because, $H_m(f)$ is the DTFT of a sequence $\{h_m[n]\}_n$ that is symmetric about $n = 0$.

Note: **Symmetry condition** cannot hold for (causal) IIR filters.

IIR filters

$$
\begin{aligned}
H(z) &= \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \\
&= (b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}) \times \frac{1}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \\
&= \frac{1}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \times (b_0 + b_1 z^{-1} + \cdots + b_M z^{-M})
\end{aligned}
$$

Without loss of (much) generality we will set $M = N$.

Comments on IIR

Most authors use b_i's as the numerator coefficients and a_i's as the denominator coefficients.

Writing the transfer function numerator first suggests implementing the zeros (the FIR part) first followed by the poles. Such a implementation is called *direct form 1*.

Writing the transfer function denominator first suggests implementing the poles (the IIR or feedback part) first followed by zeros. Such an implementation is called *direct form 2*.

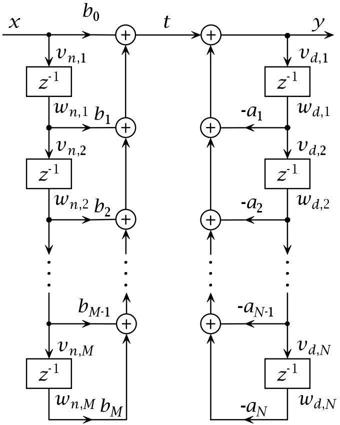
Direct forms 1 and 2

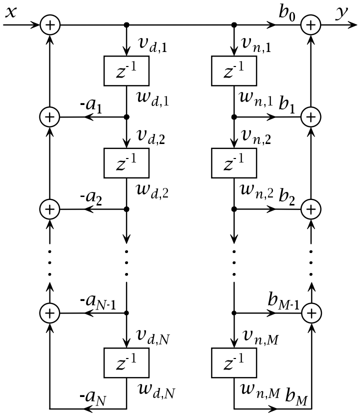
Direct Form 1 (DF1)

$$H(z) = B(z) \times \frac{1}{A(z)}$$

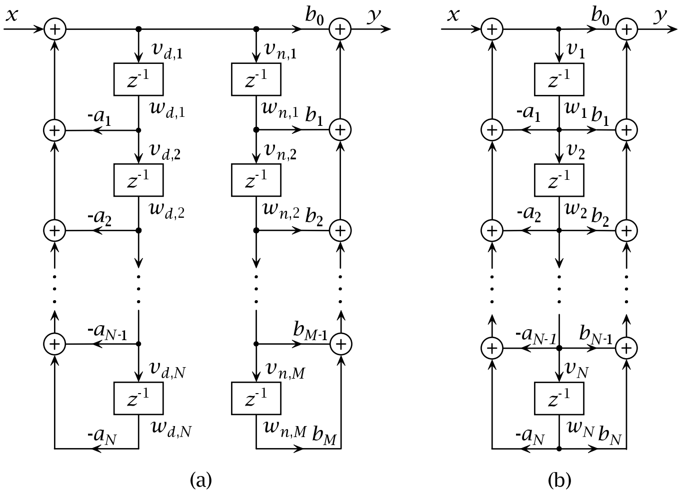
Direct Form 2 (DF2)

$$H(z) = \frac{1}{A(z)} \times B(z)$$





Canonical direct form 2



a) Non-canonical Direct Form 2.

b) DF2 in canonical form.

Comments on canonical form

Have assumed $N = M$. If $M > N$ then append a FIR filter of the necessary size. If $M < N$ then set the appropriate b values equal to zero.

The canonical form is canonical in the sense that it uses the minimum number of delay stages.

We will often simply assume that direct form 2 filters are in canonical form.

Stability and minimum phase

- The transfer function (TF) is stable if the zeros (the transfer function poles) of

$$1 + a_1 z^{-1} + \cdots + a_N z^{-N}$$

lie within the unit circle in the z-plane.

- The locations of the zeros of

$$b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}$$

do not affect the stability of the TF. The zeros can lie anywhere on the z-plane.

- A TF that has all of its numerator zeros inside of the unit circle is said to have minimum phase.
- Minimum phase TFs are useful when designing inverse filters, e.g. FM pre-emphasis and de-emphasis.

IIR in Z-domain and time domain

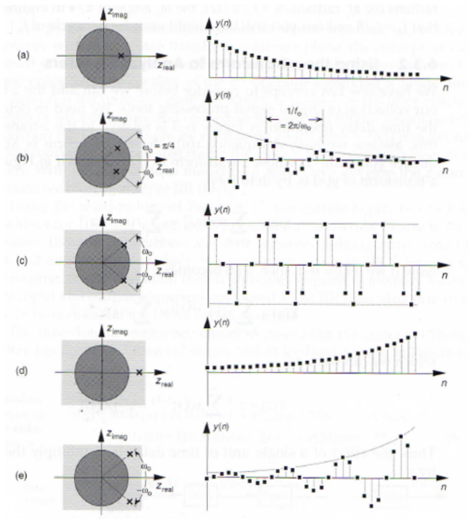


Fig. 6.14 from Lyons, "Understanding DSP"

IIR vs FIR. Which is better?

All pole IIR lowpass filter (requires 5 multiply-adds):

$$y[n] = 1.194y[n-1] - 0.436y[n-2] + 0.0605x[n] + 0.121x[n-1] + 0.0605x[n-2]$$

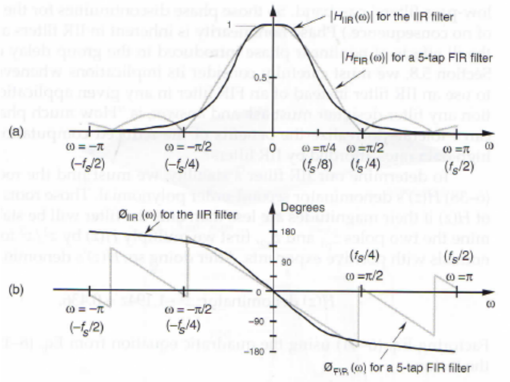
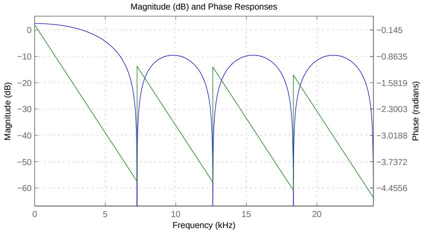


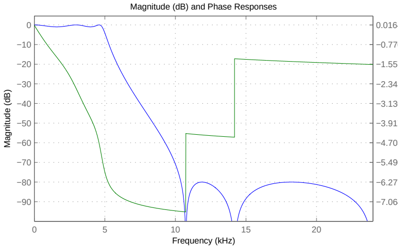
Fig. 6.14 from Lyons, "Understanding DSP"

IIR vs FIR. Which is better?(ctd)

Use fdatool:
5th order IIR lowpass filter (requires 10 multiply-adds):
10 tap FIR lowpass filter (requires 10 multiply-adds)



,



Left: FIR equiripple 10 tap. Right: IIR elliptical 5th order.

Comments

- Both filters have passband cutoff freq $f_s/10 = 4800$ and unity average magnitude response over passband.
- Both filters have the same number of multiply-adds.
- IIR has flatter passband, steeper rolloff, and lower sidelobes.
- Q. So why not always use IIR designs?
- A. IIR have disadvantages
  - (causal) IIR filters have non-linear phase response.
  - IIR filters can be very sensitive to **coefficient quantization**.
  - IIR filters can suffer from severe arithmetic **overflow** at internal nodes.

Summary of what we covered today

- FIR filter forms (Direct Form and Transposed Direct Form) and linear phase
- IIR filters forms (Direct Form 1, Direct Form 2 and Canonical forms)
- IIR vs FIR filter designs

References

"Transposed Form FIR Filters," Vikram Pasham, Andy Miller, and Ken Chapman, Xilinx Application Note XAPP219 (v1.2), Oct 25, 2001.

"Understanding digital signal processing," R. Lyons, 2006.

"Digital signal processing," Proakis and Manolakis, 3rd Edition.