EE-125:
Digital Signal Processing

Discuss Exam 3

More Optimal FIR Design- or, Fun with Least-squares

Tufts

Outline

- Quick review of optimal filtering
- New methods, mainly using some version of least-squares

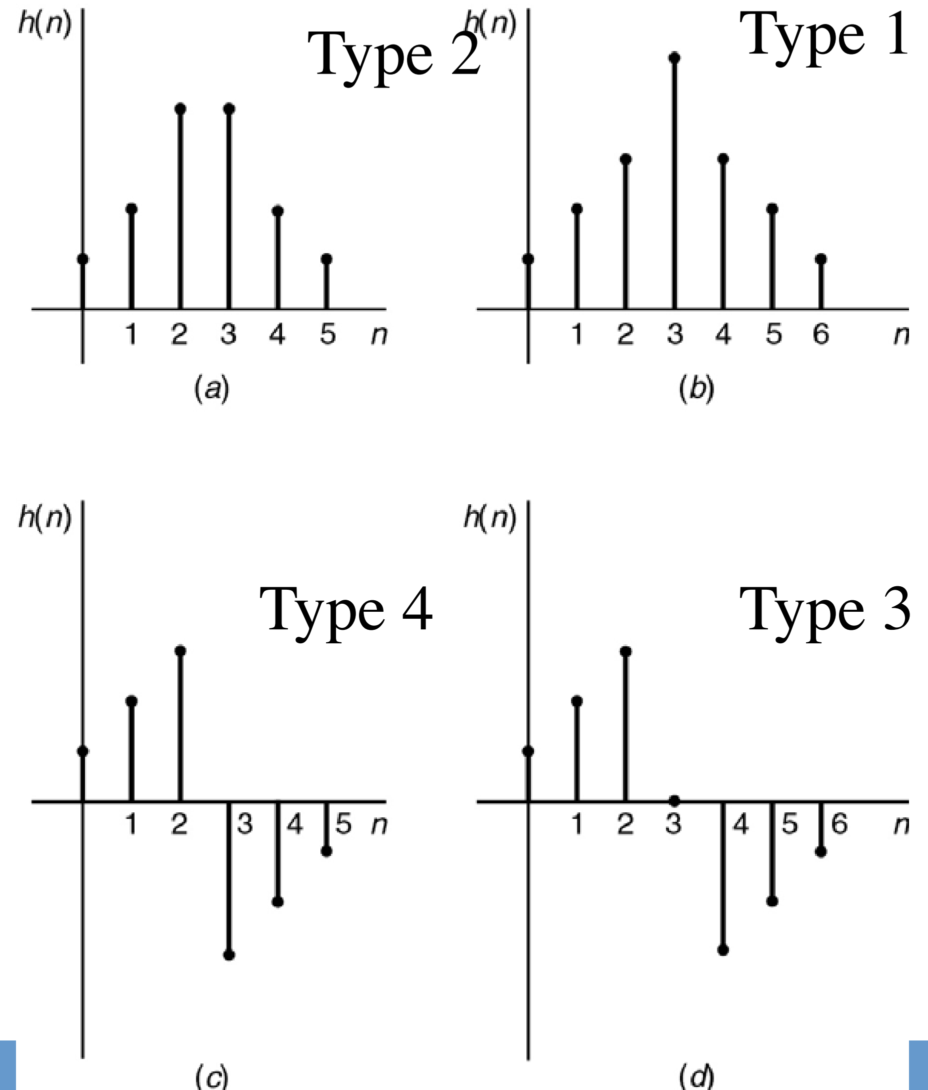
Special forms (in gory detail) – see also P&M 10.2 and 10.2.3

There are four possibilities for FIR filters that will give (generalized) linear phase

- Type 1: Symmetric, odd length
- Type 2: Symmetric, even
- Type 3: Antisymmetric, odd
- Type 4: Antisymetric, even

These let write down 4 different H(w) expressions – P&M Eq. 10.2.7-9 and 10.2.11-13

One consequence: for odd, antisymmetric filter, $H(\omega) = 0$ at $\omega=0$ and ω=pi; so can only be bandpass



Notation: Lp norms

- If x is a vector, it's L-p norm is defined as

$$\|\vec{x}\|_p = \left(\sum_{i=0}^{N-1} |x_i|^p \right)^{1/p}$$

- Important cases:
  - p=0; number of non-zero elements in x
  - p=1; "city block distance", sum of absolute values
  - p=2; Euclidian distance (Pythagorus)
  - p = infinity; max value of x
- Example: if x = [1,4,0], then
  - ||x||_0 = 2
  - ||x||_1 = 1+4 +0= 5
  - ||x||_2 = sqrt(1^2 + 4^2+ 0^2) = 4.12
  - ||x||_20 = 4.00000000000001
  - ||x||_infinity = 4

Filter design matrix

- Make some assumptions:
  - Filter is of length L+1, where L is an even number
  - Filter is centered around n=0 (we can shift it later to make it causal)
  - We want a linear phase filter, so make it <u>symmetric</u>
- Then, we can find $H(\omega_k)$ by taking the DFT of h(n):

$$
\begin{aligned}
H(\omega_k) &= \sum_{n=-L/2}^{L/2} h(n) e^{-j\omega_k n} \\
&= h_0 + 2\sum_{n=1}^{L/2} h(n) cos(\omega_k n)
\end{aligned}
$$

- We can collect these into a big matrix, one row for each k
- Typically, # frequencies k > filter length L+1

For a type 1 (symmetric, odd-length) filter

3

then, we can collect the filter response into a big matrix:

$$\underbrace{\begin{bmatrix} H(\omega_0) \\ H(\omega_1) \\ \vdots \\ H(\omega_{N-1}) \end{bmatrix}}_{H:\ N\times 1} = \underbrace{\begin{bmatrix} 1 & 2\cos\omega_0 & \cdots & 2\cos\omega_0 \frac{L}{2} \\ 1 & 2\cos\omega_1 & & 2\cos\omega_1 \frac{L}{2} \\ \vdots & & & \\ 1 & 2\cos\omega_{N-1} & & 2\cos\omega_{N-1}\frac{L}{2} \end{bmatrix}}_{\substack{A \\ N\times(L/2+1)}} \underbrace{\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_{L/2} \end{bmatrix}}_{\substack{h \\ (L/2+1)}}$$

typically # frequencies >> # taps ($N \gg L$)

Least-squared filter design

- We can collect the matrix into A, the filter coefficients into a vector *h*, and the desired response into vector *d* (*)

- Now, our problem in matrix form is:

$$\min_{h} \|A\vec{h} - \vec{d}\|_2$$

- The solution is found by expanding out the matrix-vector terms, taking 1st derivative, and setting it to zero (*)

$$\hat{\vec{h}} = [(A^T A)^{-1}] A^T \vec{d}$$

* (details in handwritten notes)

Chebyshev optimal linear phase filter

- Goal is different: L-infinity norm means minimize the worst-case error

$$\min_h \|A\vec{h} - \vec{d}\|_\infty \quad \text{Same as} \quad \underbrace{\min_h \underbrace{\max_k |a_k^T \vec{h} - d_k|}_{\text{a) Find the k with worst error…}}}_{\text{b) Pick the h that minimizes it}}$$

- If we add weighting,

$$\min_h \max_k w_k |a_k^T \vec{h} - d_k|$$

New stuff

- Iteratively Reweighted Least Squares
  - –Gives one way to approximate the Parks-McClellan filter
  - –Note Remez algorithm is in book; but lots of optimization algorithms are available these days

- Savitsky-Golay filtering
  - – Low-pass filter by polynomial fitting
  - – Avoids edge effects seen in other FIR (or IIR) filters

- Non-linear-phase FIR filters (arbitrary magnitude and phase)

Complex FIR filters

- Almost always, we design FIR filters to have "nice" phase responses – distortionless (linear phase or generalized linear phase)
- The ability to easily make linear phase filters is a key advantage of FIR digital filters!!
- But, we can also specify arbitrary phase and amplitude characteristics
  - Filters will no longer have symmetry or antisymmetry – don't fit into the "type 1" through "type 4" filters described above
  - 'A' matrix will be complex
  - desired response will be complex (because we specify arbitrary magnitude and phase
  - but generally, we still want filter coefficients h(n) to be real-valued (why?)
- Matlab function **cfirpm**