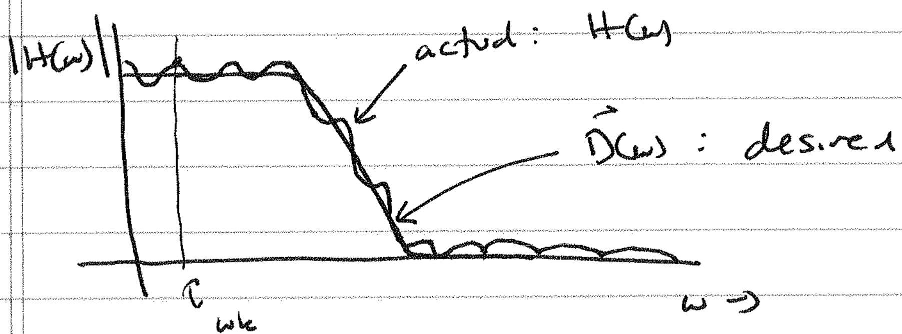
Optimal filter design: Background

Let's say we have a *desired* filter response $\vec{D}(\omega)$, and an *actual* response, $\vec{H}(\omega)$

We can measure error between these at a ~~to~~ set of points in frequency



at ω_k, error is $|D(\omega_k) - H(\omega_k)|$.

Can define a vector of errors:

$$\underline{\varepsilon} = \begin{bmatrix} \varepsilon(\omega_0) \\ \varepsilon(\omega_1) \\ \varepsilon(\omega_2) \\ \vdots \\ \varepsilon(\omega_{N-1}) \end{bmatrix} = \begin{bmatrix} H(\omega_0) - D(\omega_0) \\ H(\omega_1) - D(\omega_1) \\ \vdots \\ H(\omega_{N-1}) - D(\omega_{N-1}) \end{bmatrix}$$

note: in general, no reason # points where we measure shall = filter length

3 questions

1. how to measure overall error?
2. how to compute $\vec{H}(\omega)$ from $h(n)$?
3. algorithms for minimizing error?

Optimal filter design

notation: L_p norms.

If $\vec{x}$ is a vector, it's p-norm is

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

L0 norm → basically just counts the # nonzero elements

so if $\vec{x} = [1, 4, 0]$, $\|\vec{x}\|_0 = 2$

L1 norm → "city block distance"

$$\|\vec{x}\|_1 = \sum_{i=1}^{N-1} |x_i|$$

for example above,

$$\|\vec{x}\|_1 = |1| + |4| + |0| = 5$$

L2 - norm "Euclidian distance"

basically, pythagorean theorem

$$\|\vec{x}\|_2 = \sqrt{\sum_{i=1}^{N-1} x_i^2}$$

for above,

$$\|\vec{x}\|_2 = \sqrt{1^2 + 4^2} = \sqrt{17} = 4.12$$

L-∞ norm "max"

note above that as p increased, $\|x\|_p$ got closer to 4, the max value

in the limit,

$$\|\vec{x}\|_p \xrightarrow[p \to \infty]{} \max(\vec{x})$$

so $\|\vec{x}\|_\infty = 4$

example:

$$\|\vec{x}\|_{\frac{1000}{20}} = \left(1^{\frac{20}{1000}} + 4^{\frac{20}{1000}}\right)^{\frac{1}{1000}20}$$

$= 4.000...018$

13 zeros

② notation:

$\min_x f(x)$ means "pick the x that minimizes $f(x)$"

Optimal filter design

FIR design problem becomes

$$\min_h \| W(\omega_k) | H(\omega_k) - D(\omega_k) | \|_p$$

in words: pick h to minimize $\|(\cdot)\|_p$

optional $W(\omega_k)$: weights (we may care more about errors at some frequencies)

ω_k: a set of frequencies

$H(\omega_k)$: response of filter ~~we~~ with weights h

$D(\omega_k)$: desired response at <u>one</u> frequency

Least-squares filter design (p=2)

here, we want to minimize the <u>sum of the squared</u> <u>error</u> between H, D.

If we use $W(\omega_k)$, it's <u>weighted</u> least-squares

First we need to write $H(\omega_k)$

If we assume $h(n)$ is centered around origin, & $L+1$ long, then

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

if symmetric

← note h_0 is special as its not doubled

③

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

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

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

If we collect the desired response into a vector

$$\underline{d} = \begin{bmatrix} D(\omega_0) \\ D(\omega_1) \\ \vdots \\ D(\omega_{N-1}) \end{bmatrix}$$

so our problem is $\min_h \|Ah-d\|_2^2$

"pick h that minimizes sum-squared error"

→ if $N = L/2+1$, can solve w/ matrix inverse.

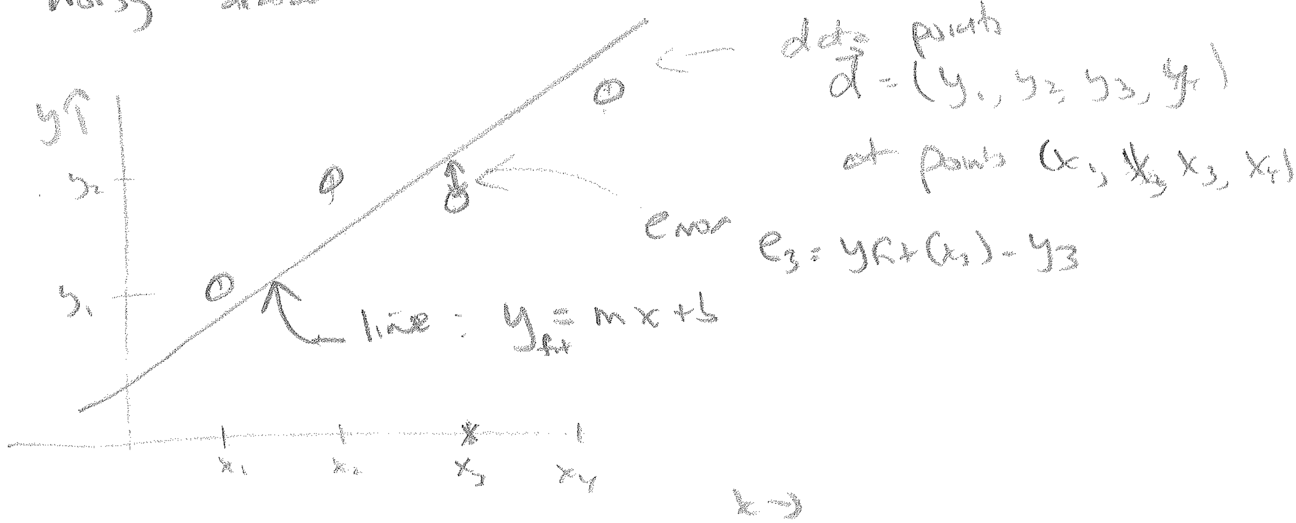
$$Ah - d = 0$$
$$Ah = d$$
$$h = A^{-1}d$$

→ in general, use <u>least-squares</u>

$$\hat{h} = \underbrace{\left[(A^TA)^{-1}A^T\right]}_{\text{pseudo-inverse – matlab 'pinv'}} d$$

whole thing: matlab firls

Sideline: least squares is most often used for REGRESSION, for example fit a line to noisy data



at point x_1, our line formula gives
$y_{fit} = m x_1 + b$, or

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{bmatrix}}_{\text{"A"}} \underbrace{\begin{bmatrix} m \\ b \end{bmatrix}}_{\text{"z"}} = \begin{bmatrix} y_{fit} \end{bmatrix} = Az$$

Want to pick m, b to minimize

$$\min_z \left\| Az - \vec{d} \right\|_2^2$$

$$= \sum_{k=1}^{4} \left| A_k \vec{z} - d(k) \right|^2$$

Soln is $\quad \vec{z} = (A^T A)^{-1} A^T \vec{d}$

or $A \backslash d$ in matlab

Weighted Least Squares

say we want to minimize error^2. Define error vector

$$\underline{\varepsilon} = \begin{bmatrix} \varepsilon(\omega_0) \\ \varepsilon(\omega_1) \\ \vdots \\ \varepsilon(\omega_{N-1}) \end{bmatrix} = \begin{bmatrix} H(\omega_0) - D(\omega_0) \\ H(\omega_1) - D(\omega_1) \\ \vdots \\ H(\omega_{N-1}) - D(\omega_{N-1}) \end{bmatrix}$$

then we want to minimize

$$\sum_i \varepsilon^2(i) = \underbrace{\begin{bmatrix} \varepsilon(0) & \varepsilon(1) & \cdots & \varepsilon(N-1) \end{bmatrix}}_{1\times N} \underbrace{\begin{bmatrix} \varepsilon(0) \\ \varepsilon(1) \\ \vdots \\ \varepsilon(N-1) \end{bmatrix}}_{N\times 1} = \|\underline{\varepsilon}\|_2^2$$

notation

---

now, say we want to <u>weight</u> different errors differently

minimize $\sum_i w_i \, \varepsilon^2(i)$ (idea is: $w_i \in [0,1]$)

in vector/matrix form

$$\underbrace{\begin{bmatrix} \varepsilon(0) & \varepsilon(1) & \cdots & \varepsilon(N-1) \end{bmatrix}}_{1\times N} \underbrace{\begin{bmatrix} w_0 & & & 0 \\ & w_1 & & \\ & & \ddots & \\ 0 & & & w_{N-1} \end{bmatrix}}_{N\times N} \underbrace{\begin{bmatrix} \varepsilon(0) \\ \varepsilon(1) \\ \vdots \\ \varepsilon(N-1) \end{bmatrix}}_{N\times 1}$$

$$= \|\underline{\varepsilon}\|_W^2$$ shorthand notation

Simpler solution approach

$$\sum_i w_i \varepsilon^2(i) = \sum_i \left(\sqrt{w_i}\, \varepsilon(i)\right)^2$$

$$= \sum_i \tilde{\varepsilon}(i)^2 = \|\tilde{\underline{\varepsilon}}\|_2^2$$

define $\tilde{\underline{\varepsilon}} = \sqrt{w_i}\, \underline{\varepsilon}$

if $\underline{\varepsilon} = \underline{A}\,\underline{h} - \underline{d}$

then $\tilde{\underline{\varepsilon}} = \underline{W}\,\underline{A}\,\underline{h} - \underline{W}\,\underline{d}$

Chebyshev optimal linear phase filter

Goal is different: minimize the worst-case error

$$\min_h \|Ah-d\|_\infty = \underbrace{\min_h \underbrace{\max_k |a_k^T x - b_k|}_{\text{find the frequency } k \text{ with the biggest error}}}_{\text{find } h \text{ that minimizes this}}$$

If we added weightings, problem would be

$$\min_h \max_k w_k |a_k^T x - b_k|$$

We can write this as:

problem statement

$$\begin{aligned} &\text{minimize } t \\ &\text{such that } \quad |a_k^T h - d_k| < t \qquad \text{(no weightings)} \\ &\text{or} \qquad w_k |a_k^T h - d_k| < t \qquad \text{(weighted)} \end{aligned}$$

We can also add other constraints: response at a certain frequency should be exactly 1, etc.

How to solve?

- classical: Remez algorithm (in book)
  Matlab 'firpm'
- more general: linear programming

Least-squares solution

Comes up in filter design, also curve fits, statistics etc.

problem is:

$$\min_x \|Ax-b\|_2^2$$

i.e. given a matrix A, and vector we'd like to fit b, find the vector x that minimizes the summed-squared error.

note: if $\vec{x}$ is a vector $\|x\|_2^2 = \sum_i x(i)^2 = [x(1)\ x(2) \cdots] \begin{bmatrix} x(1) \\ x(2) \\ \vdots \end{bmatrix}$

This is like minimizing

$$\|Ax-b\|_2^2 = (Ax-b)^T(Ax-b)$$

$$= (b^T - x^TA^T)(Ax-b) \quad \Leftarrow \text{using facts 1,2 below}$$

$$= b^TAx - b^Tb - x^TA^TAx + x^TA^Tb$$

to minimize, we take derivative & set it to zero

$$0 = \frac{d}{dx}\|Ax-b\|_2^2 = b^TA - 0 + b^TA - x^T(A^TA + (A^TA)^T)$$

$$0 = 2b^TA - x^T(A^TA + A^TA)$$

$$0 = 2b^TA - 2x^TA^TA$$

transpose everything, get

$$2A^Tb = 2(A^TA)x$$

then,

$$x = (A^TA)^{-1}A^Tb \quad //$$

A matrix, x, y vectors

linear algebra facts

1. $(A+B)^T = A^T + B^T$
2. $(AB)^T = B^TA^T$
3. $\frac{d}{d\vec{x}}(\vec{y}^TA\vec{x}) = \vec{y}^TA$
4. $\frac{d}{d\vec{x}}(\vec{x}^TA\vec{y}) = \vec{y}^TA$
5. $\frac{d}{d\vec{x}}(\vec{x}^TA\vec{x}) = x^T(A + A^T)$