Complex FIR design

High level picture

- → Almost always, we design FIR filters to be linear phase, or generalized linear phase, because we want to avoid dispersion due to non-constant group delay.
- → However, we can design FIR filters with any type of phase response
- → matlab's cfirpm does this
- → theory is below
  - an example of how we can use the least-squares design framework!

2

See ccrma.stanford.edu/~jos/filters

Complex FIR Filter Design

In linear-phase filter design, we assumed symmetry of our filter coefficients [$h(n) = h(-n)$] $\Rightarrow$

- The filter frequency response became a *sum of cosines* (``zero phase")
- The matrix A was real
- The desired magnitude response b was real
- The final zero-phase filter $\hat{x}$ could be right-shifted $L/2$ samples to get a corresponding causal linear-phase FIR filter

Now we would like to specify a *complex* frequency response. This means that:

- b is complex
- A is complex
- We still want x (our filter coefficients) to be real

see page 4 notes

If we try to use ' $\backslash$ ' or `pinv` in Matlab, we will generally get a complex result for $\hat{x}$

Summarizing our problem:

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

where $A \in \mathbf{C}^{N \times M}$, $b \in \mathbf{C}^{N \times 1}$, and $x \in \mathbf{R}^{M \times 1}$

Hence we have,

$$\min_x \| [\mathcal{R}(A) + j\mathcal{I}(A)]\, x - [\mathcal{R}(b) + j\mathcal{I}(b)] \|_2^2$$

Which can be written as:

$$\min_x \| \mathcal{R}(A)x - \mathcal{R}(b) + j\, [\mathcal{I}(A)x + \mathcal{I}(b)] \|_2^2$$

or

3

← see page 5 notes

$$\min_x \left\| \begin{bmatrix} \mathcal{R}(A) \\ \mathcal{I}(A) \end{bmatrix} x - \begin{bmatrix} \mathcal{R}(b) \\ \mathcal{I}(b) \end{bmatrix} \right\|_2^2$$

which is written in terms of only *real* variables.

Hence, we can use the standard least squares solvers in Matlab and end up with a *real* solution.

Optimal FIR Filters: Arbitrary Magnitude and Phase Specificiations

`cfirpm` (Matlab Signal Processing Toolbox) performs *complex* $L - infinity$ FIR filter design:

- Documented online at The Mathworks
  (search for `cfirpm`)
- Convergence theoretically guaranteed for *arbitrary* magnitude and phase specifications versus frequency.
- Reduces to Parks-McClellan algorithm (Remez second algorithm) as a special case.
- Written by Karam and McClellan. See ``Design of Optimal Digital FIR Filters with Arbitrary Magnitude and Phase Responses," by Lina J. Karam and James H. McClellan, ISCAS-96. The paper may be downloaded at

  `http://www.eas.asu.edu/~karam/papers/iscas96_km.html`

`firgr` (formerly `gremez`) in the Matlab Filter Design Toolbox performs ``generalized" $L - infinity$ FIR filter design, adding support for minimum-phase FIR filter design, among other features [236].

(from elsewhere on the website)

How do we get the filter design A matrix for this case?

<u>class notes</u>: we said $\vec{H}(\omega) = A h$ was vector of filter responses at different frequencies.

last pages call this 'A x' but same thing.

definition of DTFT: $H(\omega) = \sum_{n=0}^{L-1} h(n) e^{-j\omega n}$

for filter, length L, starting at n=0

at a frequency ω_k:

$$H(\omega_k) = h(0) + h(1)e^{-j\omega_k} + h(2)e^{-j2\omega_k} + \dots h(L-1)e^{-j(L-1)\omega}$$

stack up into a vector for all values of ω_k where we specify the filter response:

$$\begin{bmatrix} H(\omega_0) \\ H(\omega_1) \\ H(\omega_2) \\ \vdots \\ H(\omega_{N-1}) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & e^{-j\omega_0} & e^{-j2\omega_0} & \dots & e^{-j(L-1)\omega_0} \\ 1 & e^{-j\omega_1} & e^{-j2\omega_1} & \dots & e^{-j(L-1)\omega_1} \\ 1 & e^{-j\omega_2} & e^{-j2\omega_2} & \dots & e^{-j(L-1)\omega_2} \\ & & \vdots & & \\ 1 & e^{-j1\omega_{N-1}} & e^{-j2\omega_{N-1}} & \dots & e^{-j(L-1)(N-1)} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ \vdots \\ h(L-1) \end{bmatrix}}_{h}$$

N freq.
we design filter based on response at these points

similar to Type1 filter we did before, but
1) A is real
2) h has all coefficients
filter not ½ due to ...

Why can we stack real, imaginary parts as shown on page 3 ?

---

first, think about real-valued case.

we want to minimize $\|A\vec{x}-\vec{b}\|_2$ (CCRMA notation)

$\vec{b}$ = vector of desired filter response, $A\vec{x}$ = actual filter response

define $\vec{\varepsilon} = A\vec{x}-\vec{b}$ = vector of error at all freq

$$\vec{\varepsilon} = \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_{N-1} \end{bmatrix}$$

by definition of norm $\| \ \|_2$,

$$\|\vec{\varepsilon}\|_2 = \sqrt{\sum_{k=0}^{N-1} |\varepsilon_k|^2} = \sqrt{\sum_{k=0}^{N-1} \varepsilon_k^2}$$

for real-valued ε_k

but we can get the same thing through vector multiply:

$$\vec{\varepsilon}^T\varepsilon = [\varepsilon_0 \ \varepsilon_1 \ \varepsilon_2 \cdots \varepsilon_{N-1}] \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_{N-1} \end{bmatrix} = \sum_{k=0}^{N-1} \varepsilon_k^2$$

then can take square root: so $\|\vec{\varepsilon}\|_2 = \sqrt{\vec{\varepsilon}^T\vec{\varepsilon}}$

or, $\|\vec{\varepsilon}\|_2^2 = \vec{\varepsilon}^T\varepsilon$

easier: square to get rid of $\sqrt{\ }$

6

now, think about a complex error vector $\vec{\varepsilon}$

$$\underset{\text{complex}}{\vec{\varepsilon}} = \underset{\text{real part}}{\vec{\varepsilon}_R} + j \underbrace{\vec{\varepsilon}_I}_{\text{imag part}}$$

minus

now, $\|\vec{\varepsilon}\|_2^2 = \sum_{k=0}^{N-1} |\vec{\varepsilon}_k|^2 = \sum_{k=0}^{N-1} (\varepsilon_{R,k} + j\varepsilon_{I,k})(\varepsilon_{R,k} - j\varepsilon_{I,k})$

work with squared so dont have to keep writing $\sqrt{}$!

$$= \sum_{k=0}^{N-1} \left(\varepsilon_{R,k}^2 + \varepsilon_{I,k}^2\right)$$

$$= \sum_{k=0}^{N-1} \varepsilon_{R,k}^2 + \sum_{k=0}^{N-1} \varepsilon_{I,k}^2$$ answer 1

Ok... if we stack ε_R and ε_R, do we get same thing?

$$\vec{\varepsilon}_{stack} = \begin{bmatrix} \vec{\varepsilon}_R \\ \vec{\varepsilon}_I \end{bmatrix} = \begin{bmatrix} \varepsilon_{R,1} \\ \varepsilon_{R,2} \\ \varepsilon_{R,3} \\ \varepsilon_{I,1} \\ \varepsilon_{I,2} \\ \varepsilon_{I,3} \end{bmatrix}$$ ← these are all real-valued

example if $N=3$

then, $\|\vec{\varepsilon}_{stack}\|_2^2 = \sum_{k=0}^{2N-1} (\varepsilon_{stack,k})^2 = \varepsilon_{R_1}^2 + \varepsilon_{R_2}^2 + \varepsilon_{R_3}^2$
$+ \varepsilon_{I,1}^2 + \varepsilon_{I,2}^2 + \varepsilon_{I,3}^2$

$$= \sum_{k=0}^{N-1} \varepsilon_{R,k}^2 + \sum_{k=0}^{N-1} \varepsilon_{I,k}^2$$

answer 2

matches answer 1!

also could say:
$\|\vec{\varepsilon}_{stack}\|_2^2 = \vec{\varepsilon}_{stack}^{\,T} \vec{\varepsilon}_{stack}$

same answer