Lab 4 – filter design

MATLAB resources for filter design

Note: in the parameters of MATLAB filter design functions, the frequency value is a multiplier to π (e.g. 1.0 means $f = f_s/2$ or equivalently $\theta = 1.0 \cdot \pi$).

filter implements a digital filter
`Y = filter(B, A, X);` $y(n) = \sum_{k=1}^{length(B)} B_k \cdot x(n-(k-1)) - \sum_{l=2}^{length(A)} A_l \cdot y(n-(l-1))$

freqz computes digital filter frequency response
`[H,W] = freqz(B,A,N);` $H(e^{jW})$, B,A - coefficients, N - num. points
Put A=[1] for FIR filters. In Matlab $\geq$ 4.0 freqz with nargout==0 gives nice plot of magnitude and phase of H.

grpdelay calculate (and plot) group delay `grpdelay(B,A);`

zplane `zplane(B,A)` nicely plots zeros and poles on Z-plane when B,A are **row vectors** of filter coefficients

abs calculates the magnitude of a complex number

firpm (in Matlab < 6.3 it was **remez**) Parks-McClellan optimal equirriple FIR filter design.
`B = firpm(N,F,M);` N - order, F - frequencies, M - magnitudes
example:an LP filter with passband of 0.1π, stopband from 0.2π to $1 \cdot \pi$ is specified as:
`F=[0 0.1 0.2 1]; M=[1 1 0 0];plot(F,M);` (last command plots the specified frequency response)

fir2 FIR filter design using the window method
`B = fir2(N,F,M,win);` – `N, F, M` as in remez; `win` – chosen window (e.g `bartlett(N+1)`)
Implemented windows: `blackman boxcar butter chebwin hamming hanning kaiser`

poly *poly(r)* gives the vector of coefficients of a polynomial whose roots are specified in vector r.

roots *roots(p)* gives roots of a polynomial whose coefficients are in vector p

IIR design by bilinear transformation:

Approximation		Butterworth	Chebyshev type 1	Chebyshev type 2	Cauer (elliptical)
Min. order	[n, Wc] =f(Wp, Ws, Rp, Rs)	buttord()	cheb1ord()	cheb2ord()	[n,Wp] = ellipord()
Filter coefficients	[B,A] =	butter(n, Wc)	cheby1(n, Rp, Wc)	cheby2(n, Rs, Wc)	ellip(n, Rp, Rs, Wp)

Experiments

1. Prepare a 2-nd order IIR filter with no significant zeros and with poles at $0.9e^{\pm j0.2\pi}$.(hint: do it straight from the $H(z)$ description above – you need to find coefficients of a polynomial so use `poly`). Plot on the screen its:

   - impulse response (hint: prepare a delta as `dlt=zeros(1,64); dlt(1)=1;`),
   - frequency response (quick: use `freqz` without output arguments),
   - group delay (use `grpdelay(...);`,
   - responses for sine waves of different frequencies (at least 3 freq. values)

   In the report, note:

- period of oscillations of $h(n)$ and its decay rate (after how much time the envelope decays to 1/2 of the initial value)
- min and max value of group delay; location of the group delay peak
- frequencies of the sine waves and magnitudes of the responses

Save coefficients for future use (e.g. on paper, just in case of reboot).

2. Put poles at $0.9e^{\pm j0.8\pi}$. Note the oscillation period and decay rate. Comment on the differences in the h(n) period and rate, and in shape of $H(\theta)$ w.r.t. the previous experiment.

3. Use `firpm` to design a lowpass FIR filter. Specify easy parameters – wide transition band, e.g passband to 0.2π, stopband from 0.6π. Use order of 10, increase if necessary. Plot the impulse response using `stem(B)`, plot the frequency response to see the passband gain and stopband gain.
   Then put narrower transition constraint with the same order – check the change in frequency characteristics. Allow greater order.
   Hint: stopband gain is better viewed in log (decibel) scale. But decibels may run to $-\infty$ for zero input, so use `max(-80,yourvalue-in-decibels)` for nice plot.
   In the report note:

   - How well the requirements were fulfilled?
   - What shape do the sidelobes have (in the stopband)?

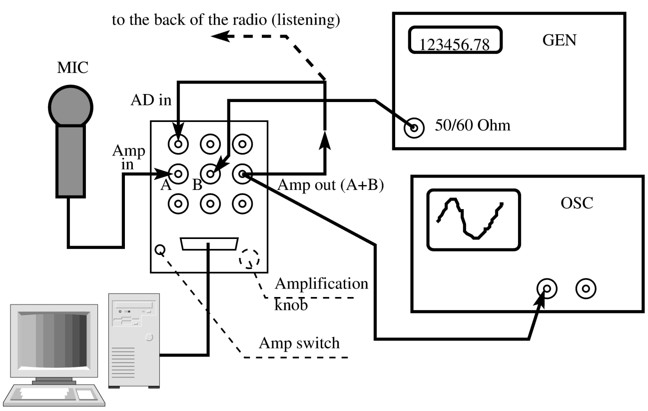
4. Design FIR filters for the same specifications, but by window method – use `fir2`.

   (a) with boxcar window
   (b) with hamming window

   Identify the window effects. Compare results against the Parks-McClellan filter.

5. Design an IIR LP filter with passband till 0.2π, stopband from 0.3π, passband ripple 0.5 dB, stopband 80 dB down (not all the specifications are used with different filter types – see help). Compare two of the types shown in the table (see "resources" section).
   Plot zeros and poles (sketch zeros/poles layout in the report), and frequency responses for each filter. Note the order necessary to fulfill the requirements with different approximations.

6. Mini-project: remove interference from speech signal.
   Connect a circuit to disturb speech signal by adding a sinusoidal tone from generator. Then, record the signal (5-10 seconds) and design a bandstop filter to remove the tone. Play the result.



(connect to the loudspeaker in the "NDN" universal ....)

For recording use `y=getdata(Nsamples_in_row, 1, Tsampling)`.

File: lab4˙2016z LaTeXed on December 5, 2016