1 Short help on Parks-McClellan design of FIR Low Pass Filters using Matlab

The design of an FIR filter using Parks-McClellan algorithm is a two-step process. First, you need to use the *remezord* command to estimate the order of the optimal Parks-McClellan FIR filter to meet your design specifications. The syntax of the command is as follows:

```
[n,fo,mo,w]=remezord(f,m,dev)
```

`f` is the vector of band frequencies. For a low pass filter, `f=[wp ws]` where `wp` is the upper edge of the passband and `ws` is the lower edge of the stopband. The vector `m` contains the desired magnitude response values at the passbands and the stopbands of the filter. Since a lowpass filter consists of a passband followed by a stopband, `m` has two entires. Namely, `m=[1 0]` because you would like the magnitude response to be equal to 1 in the passband and equal to 0 in the stopband. The vector `dev` has the maximum allowable deviations of the magnitude response of the filter from the desired magnitude response. It has the same number of entries as there are `m`. Thus, for a low-pass filter it has two entries and is equal to `[passband ripple, stopband ripple]`. After you specify the vectors `f`, `m` and `dev`, you can run the *remezord* command in the syntax given above to compute the order of the filter `n`. The *remezord* command also outputs the resulting `fo` and `mo` vectors. In actually designing your filter using *remez* , you should use these two vectors instead of `f` and `m`. The second stage is the actual design of the filter, using the *remez* command. After running *remezord* and finding the `n`, `fo` and `mo`, type

```
b=remez(n,fo,mo)
```

to find the impulse response `b` of the Parks-McClellan FIR filter you need to design. The vector `b` contains the coefficients for the $Z-$transform of your filter $H(z)$. That is,

$$H(z) = b(1) + b(2)z^{-1} + b(3)z^{-2} + \cdots + b(n+1)z^{-n}$$

This conludes the design of your filter. The *remezord* and *remez* can be used to design also highpass, bandpassand multiband filters. See the Matlab help for details.

2 Matlab Help on Remezord

```
 REMEZORD  FIR order estimator (lowpass, highpass, bandpass, multiband)
    [N,Fo,Ao,W] = REMEZORD(F,A,DEV,Fs) finds the approximate order N,
    normalized frequency band edges Fo, frequency band magnitudes Ao and
    weights W to be used by the REMEZ function as follows:
        B = REMEZ(N,Fo,Ao,W)
    The resulting filter will approximately meet the specifications given
```

Questions or comments concerning this laboratory should be directed to Prof. Charles A. Bouman, School of Electrical and Computer Engineering, Purdue University, West Lafayette IN 47907; (765) 494-0340; bouman@ecn.purdue.edu

```
by the input parameters F, A, and DEV.  F is a vector of cutoff
frequencies in Hz, in ascending order between 0 and half the sampling
frequency Fs. If you do not specify Fs, it defaults to 2.  A is a vector
specifying the desired function's amplitude on the bands defined by F.
The length of F is twice the length of A, minus 2 (it must therefore
be even).  The first frequency band always starts at zero, and the last
always ends at Fs/2.  DEV is a vector of maximum deviations or ripples
allowable for each band.

C = REMEZORD(F,A,DEV,FS,'cell') is a cell-array whose elements are the
parameters to REMEZ.

EXAMPLE
   Design a lowpass filter with a passband cutoff of 1500, a
   stopband cutoff of 2000Hz, passband ripple of 0.01, stopband ripple
   of 0.1, and a sampling frequency of 8000Hz:

      [n,fo,mo,w] = remezord( [1500 2000], [1 0], [0.01 0.1], 8000 );
      b = remez(n,fo,mo,w);

   This is equivalent to
      c = remezord( [1500 2000], [1 0], [0.01 0.1], 8000, 'cell');
      b = remez(c{:});

CAUTION 1: The order N is often underestimated. If the filter does not
meet the original specifications, a higher order such as N+1 or N+2 will.
CAUTION 2: Results are inaccurate if cutoff frequencies are near zero
frequency or the Nyquist frequency.
```

3 Matlab Help on Remez

```
REMEZ Parks-McClellan optimal equiripple FIR filter design.
   B=REMEZ(N,F,A) returns a length N+1 linear phase (real, symmetric
   coefficients) FIR filter which has the best approximation to the
   desired frequency response described by F and A in the minimax
   sense. F is a vector of frequency band edges in pairs, in ascending
   order between 0 and 1. 1 corresponds to the Nyquist frequency or half
   the sampling frequency. A is a real vector the same size as F
   which specifies the desired amplitude of the frequency response of the
   resultant filter B. The desired response is the line connecting the
   points (F(k),A(k)) and (F(k+1),A(k+1)) for odd k; REMEZ treats the
   bands between F(k+1) and F(k+2) for odd k as "transition bands" or
   "don't care" regions. Thus the desired amplitude is piecewise linear
```

```
with transition bands. The maximum error is minimized.

B=REMEZ(N,F,A,W) uses the weights in W to weight the error. W has one
entry per band (so it is half the length of F and A) which tells
REMEZ how much emphasis to put on minimizing the error in each band
relative to the other bands.

B=REMEZ(N,F,A,'Hilbert') and B=REMEZ(N,F,A,W,'Hilbert') design filters
that have odd symmetry, that is, B(k) = -B(N+2-k) for k = 1, ..., N+1.
A special case is a Hilbert transformer which has an approx. amplitude
of 1 across the entire band, e.g. B=REMEZ(30,[.1 .9],[1 1],'Hilbert').

B=REMEZ(N,F,A,'differentiator') and B=REMEZ(N,F,A,W,'differentiator')
also design filters with odd symmetry, but with a special weighting
scheme for non-zero amplitude bands. The weight is assumed to be equal
to the inverse of frequency times the weight W. Thus the filter has a
much better fit at low frequency than at high frequency. This designs
FIR differentiators.

B=REMEZ(...,{LGRID}), where {LGRID} is a one-by-one cell array containing
an integer, controls the density of the frequency grid. The frequency grid
size is roughly LGRID*N/2*BW, where BW is the fraction of the total band
interval [0,1] covered by F. LGRID should be no less than its default of 16.
Increasing LGRID often results in filters which are more exactly equi-
ripple, at the expense of taking longer to compute.

[B,ERR]=REMEZ(...) returns the maximum ripple height ERR.

[B,ERR,RES]=REMEZ(...) returns a structure RES of optional results
computed by REMEZ, and contains the following fields:

   RES.fgrid: vector containing the frequency grid used in
              the filter design optimization
     RES.des: desired response on fgrid
      RES.wt: weights on fgrid
       RES.H: actual frequency response on the grid
   RES.error: error at each point on the frequency grid (desired - actual)
   RES.iextr: vector of indices into fgrid of extremal frequencies
   RES.fextr: vector of extremal frequencies

See also CREMEZ, FIRLS, FIR1, FIR2, BUTTER, CHEBY1, CHEBY2, ELLIP,
FREQZ, FILTER.
```