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 entries. 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 $[\text{passband ripple, stopband ripple}]$. 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 concludes 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.
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:

\[
\begin{align*}
[n,fo,mo,w] &= \text{remezord}([1500 2000], [1 0], [0.01 0.1], 8000); \\
b &= \text{remez}(n,fo,mo,w);
\end{align*}
\]

This is equivalent to

\[
\begin{align*}
c &= \text{remezord}([1500 2000], [1 0], [0.01 0.1], 8000, 'cell'); \\
b &= \text{remez}(c{:});
\end{align*}
\]

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.

\( B = \text{REMEZ}(N,F,A,W) \) 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 \( \text{REMEZ} \) how much emphasis to put on minimizing the error in each band relative to the other bands.

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

\( B = \text{REMEZ}(N,F,A,'\text{differentiator}') \) and \( B = \text{REMEZ}(N,F,A,W,'\text{differentiator}') \) 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.

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

\([B, \text{ERR}] = \text{REMEZ}(...)\) returns the maximum ripple height \( \text{ERR} \).

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

\[
\begin{align*}
\text{RES.fgrid}: & \text{ vector containing the frequency grid used in} \\
& \text{the filter design optimization} \\
\text{RES.des}: & \text{desired response on fgrid} \\
\text{RES.wt}: & \text{weights on fgrid} \\
\text{RES.H}: & \text{actual frequency response on the grid} \\
\text{RES.error}: & \text{error at each point on the frequency grid (desired - actual)} \\
\text{RES.iextr}: & \text{vector of indices into fgrid of extremal frequencies} \\
\text{RES.fextr}: & \text{vector of extremal frequencies}
\end{align*}
\]

See also \( \text{CREMEZ}, \text{FIRLS}, \text{FIR1}, \text{FIR2}, \text{BUTTER}, \text{CHEBY1}, \text{CHEBY2}, \text{ELLIP}, \text{FREQZ}, \text{FILTER} \).