# LINEAR-PHASE FIR DIGITAL FILTER DESIGN
# BY
# REMEZ EXCHANGE ALGORITHM

## Exercises 5.

**REMEZ Parks-McClellan optimal equiripple FIR filter design. Summary**

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 mini-max 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.

For filters with a gain other than zero at Fs/2, e.g., high-pass and band-stop filters, N must be even. Otherwise, N will be incremented by one.

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 equiripple, 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 and GREMEZ in the Filter Design toolbox.

## Examples

### Example 1. Low-pass filter design

$\omega_P = 0.25\pi,\ \omega_S = 0.3\pi$

**Solution:**
f =[ 0 0.2500 0.3000 1.0000] ;
a =[ 1 1 0 0];
N=10, 20, 30, 40, 50;
b=remez(N,f,a);

***Note:*** the designed filters will possess the same ripple in the pass-band and stop-band

---

### Example 2. High-pass filter design

$\omega_S = 0.25\pi,\ \omega_p = 0.3\pi$

f =[ 0 0.2500 0.3000 1.0000] ;
N=10, 20, 30, 40, **50**;
b=remez(N,f,a);

***Note:*** the designed filters will possess the same ripple in the pass-band and stop-band

---

### Example 3. Pass-band filter design

$\omega_{P1} = 0.25\pi,\ \omega_{P2} = 0.45\pi,\ \omega_{S1} = 0.2\pi,\ \omega_{S1} = 0.5\pi$

f=[0 .2 .25 .45 .5 1];
a=[0 0 1 1 0 0];
N=10, 20, 30, 40, **50**;
b=remez(N,f,a);

***Note:*** the designed filters will possess the same ripple in the pass-band and stop-band

---

### Example 4. Stop-band filter design

$\omega_{S1} = 0.25\pi,\ \omega_{S2} = 0.45\pi,\ \omega_{P1} = 0.2\pi,\ \omega_{P1} = 0.5\pi$

f=[0 .2 .25 .45 .5 1];
a=[1 1 0 0 1 1];
N=10, 20, 30, 40, **50**;
b=remez(N,f,a);

***Note:*** the designed filters will possess the same ripple in the pass-band and stop-band

**Example 5. Low-pass filter design**

$\omega_P = 0.25\pi, \ \omega_S = 0.3\pi$

**Solution:**

```
f =[ 0 0.2500 0.3000 1.0000] ;
a =[ 1 1 0 0];
ww=[1 .5];
N=10, 20, 30, 40, 50;
b=remez(N,f,a,ww);
```

***Note 1:*** The designed filters will possess the different ripple in the pass-band and stop-band. The ripple in the stop-band will be doubled of the ripple in the pass-band.

***Note 2:*** pass-band: $\delta_1, w_1$ ; stop-band: $\delta_2, w_2$; result: $\delta_2 = \delta_1 \frac{w_1}{w_2}$

---

**Example 6. High-pass filter design**

$\omega_S = 0.25\pi, \ \omega_p = 0.3\pi$

```
f =[ 0 0.2500 0.3000 1.0000] ;
a=[0 0 1 1];
ww=[.5 1];
N=10, 20, 30, 40, 50;
b=remez(N,f,a,ww);
```

***Note:*** The designed filters will possess the different ripple in the pass-band and stop-band. The ripple in the stop-band will be doubled of the ripple in the pass-band.

---

**Example 7. Pass-band filter design**

$\omega_{P1} = 0.25\pi, \ \omega_{P2} = 0.45\pi, \ \omega_{S1} = 0.2\pi, \ \omega_{S1} = 0.5\pi$

```
f=[0 .2 .25 .45 .5 1];
a=[0 0 1 1 0 0];
ww=[.5 1 .5]
N=10, 20, 30, 40, 50;
b=remez(N,f,a,ww);
```

---

**Example 8. Stop-band filter design**

$\omega_{S1} = 0.25\pi,\ \omega_{S2} = 0.45\pi,\ \omega_{P1} = 0.2\pi,\ \omega_{P1} = 0.5\pi$

```
f=[0 .2 .25 .45 .5 1];
a=[1 1 0 0 1 1];
ww=[1 .5 1];
N=10, 20, 30, 40, 50;
b=remez(N,f,a,ww);
```

---

**Example 9. Differentiator design**

```
f=0.01:.001:.99911;
a=0.01:.001:.99911;
b=remez(25,f,f,'differentiator');
```

---

**Example 10. Hilbert transformer design**

```
f= [.1 .9];
a=[1 1];
b=remez(30,f,a,'Hilbert');
```

---

**Example 11. Filtering**

Let $x_1(t) = 2\cos 2\pi f_1 t,\quad x_2(t) = 1.3\cos 2\pi f_2 t,\quad f_1 = 15\,kHz,\quad f_2 = 45\,kHz,\quad y(t) = x_1(t) + x_2(t)$ and $f_0 = 100\,kHz$ is sampling frequency. By using a suitable FIR filters, extract $x_1(t)$ and $x_2(t)$ from $y(t)$.

---

**Example 12. Filtering**

Let $x_1(t) = 2\cos 2\pi f_1 t,\quad x_2(t) = 2\cos 2\pi f_2 t,\quad f_1 = 15\,kHz,\quad f_2 = 45\,kHz,\quad y(t) = x_1(t)x_2(t)$ and $f_0 = 200\,kHz$ is sampling frequency. By using a suitable FIR filter extract $y(t)$ from signal $z(t) = y(t) + n(t)$, where $n(t)$ is zero-mean Gaussian noise with $\sigma_n^2 = 2$. For signal $z(t)$ as well as for the signal obtained by filtering $z(t)$ evaluate signal-to-noise ratio.

---