Chapter 15
DESIGN OF NONRECURSIVE FILTERS USING OPTIMIZATION

15.1 Introduction 15.2 Problem Formulation 15.3 Remez Exchange Algorithm 15.4 Improved Search Methods 15.5 Efficient Remez Exchange Algorithm 15.7 Prescribed Specifications

---

Victoria, BC, Canada
Email: aantoniou@ieee.org

July 14, 2018

Introduction

- The weighted-Chebyshev method for the design of nonrecursive filters is an iterative multivariable optimization method based on the *Remez Exchange Algorithm*.

Introduction

- The weighted-Chebyshev method for the design of nonrecursive filters is an iterative multivariable optimization method based on the *Remez Exchange Algorithm*.
- It can be used to design optimal nonrecursive filters with arbitrary amplitude responses.

- Herrmann published a short paper in Electronics Letters in May 1970 on the design of nonrecursive filters.

Introduction – Historical Evolution

- Herrmann published a short paper in Electronics Letters in May 1970 on the design of nonrecursive filters.
- This paper was followed by a series of papers by Parks, McClellan, Rabiner, and Herrmann during the early seventies.

Introduction – Historical Evolution

- Herrmann published a short paper in Electronics Letters in May 1970 on the design of nonrecursive filters.
- This paper was followed by a series of papers by Parks, McClellan, Rabiner, and Herrmann during the early seventies.
- These developments led in 1975 to the well-known McClellan-Parks-Rabiner computer program for the design of nonrecursive filters, which has found widespread applications.

Introduction – Historical Evolution

- Herrmann published a short paper in Electronics Letters in May 1970 on the design of nonrecursive filters.
- This paper was followed by a series of papers by Parks, McClellan, Rabiner, and Herrmann during the early seventies.
- These developments led in 1975 to the well-known McClellan-Parks-Rabiner computer program for the design of nonrecursive filters, which has found widespread applications.
- Enhancements to the weighted-Chebyshev method were proposed by Antoniou during the early eighties.

Problem Formulation

Consider a nonrecursive filter characterized by the transfer function

$$H(z) = \sum_{n=0}^{N-1} h(nT)z^{-n}$$

and assume that

- the filter length N is odd (the filter order $N-1$ is even),
- the impulse response is symmetrical, and
- the sampling frequency is $\omega_s = 2\pi$ rad/s (the Nyquist frequency is π rad/s) and the sampling period is $T = 1$ s.

- The frequency response of the filter can be expressed as

$$H(e^{j\omega}) = e^{-jc\omega} P_c(\omega)$$

where

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega \qquad \text{(A)}$$

is the frequency response of a *noncausal* version of the required filter and

$$\begin{aligned} a_0 &= h(c) \\ a_k &= 2h(c-k) \quad \text{for} \quad k = 1, 2, \ldots, c \\ c &= (N-1)/2 \end{aligned}$$

Error Function

- An error function $E(\omega)$ can be constructed as

$$E(\omega) = W(\omega)[D(\omega) - P_c(\omega)]$$

where $e^{-jc\omega}D(\omega)$ is the idealized frequency response of the desired filter, $W(\omega)$ is a weighting function, and

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega$$

Error Function

- An error function $E(\omega)$ can be constructed as

$$E(\omega) = W(\omega)[D(\omega) - P_c(\omega)]$$

where $e^{-jc\omega}D(\omega)$ is the idealized frequency response of the desired filter, $W(\omega)$ is a weighting function, and

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega$$

- If $|E(\omega)|$ is minimized such that

$$|E(\omega)| = |W(\omega)[D(\omega) - P_c(\omega)]| \leq \delta_p \quad \text{for} \quad \omega \in \Omega \qquad \text{(B)}$$

with respect to a set of frequencies in the interval $[0, \pi]$, say Ω, a filter can be obtained in which

$$|E_0(\omega)| = |D(\omega) - P_c(\omega)| \leq \frac{\delta_p}{|W(\omega)|} \quad \text{for} \quad \omega \in \Omega \qquad \text{(C)}$$

- In the case of a lowpass filter, the minimization of $|E(\omega)|$ will force the inequality

$$|E_0(\omega)| = |D(\omega) - P_c(\omega)| \leq \frac{\delta_p}{|W(\omega)|} \quad \text{for} \quad \omega \in \Omega \qquad \text{(C)}$$

where

$$D(\omega) = \begin{cases} 1 & \text{for } 0 \leq \omega \leq \omega_p \\ 0 & \text{for } \omega_a \leq \omega \leq \pi \end{cases}$$

- In the case of a lowpass filter, the minimization of $|E(\omega)|$ will force the inequality

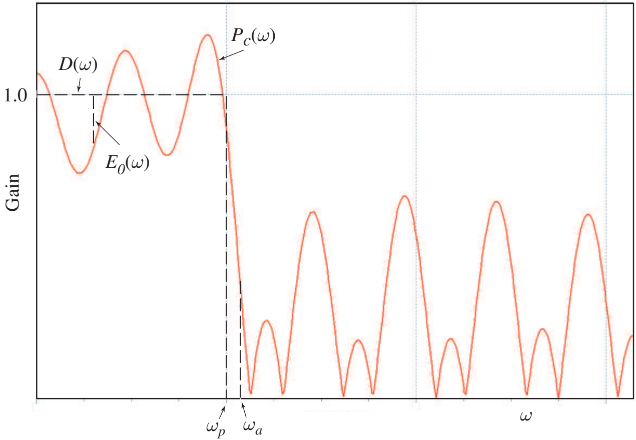
$$|E_0(\omega)| = |D(\omega) - P_c(\omega)| \leq \frac{\delta_p}{|W(\omega)|} \quad \text{for} \quad \omega \in \Omega \qquad \text{(C)}$$

where

$$D(\omega) = \begin{cases} 1 & \text{for } 0 \leq \omega \leq \omega_p \\ 0 & \text{for } \omega_a \leq \omega \leq \pi \end{cases}$$

- In effect, a minimization algorithm will force the actual gain function $P_c(\omega)$ to approach the ideal gain function $D(\omega)$.

Lowpass Filters *Cont'd*



- If we choose the weighting function

$$W(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_p \\ \frac{\delta_p}{\delta_a} & \text{for } \omega_a \le \omega \le \pi \end{cases}$$

then from Eq. (C), i.e.,

$$|E_0(\omega)| = |D(\omega) - P_c(\omega)| \le \frac{\delta_p}{|W(\omega)|} \quad \text{for } \; \omega \in \Omega \qquad \text{(C)}$$

we get

$$|E_0(\omega)| \le \begin{cases} \delta_p & \text{for } 0 \le \omega \le \omega_p \\ \delta_a & \text{for } \omega_a \le \omega \le \pi \end{cases}$$

Minimax Problem

- The most appropriate approach for the solution of the optimization problem just described is to solve the *minimax* problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad \{\max_{\omega} |E(\omega)|\}$$

where

$$\mathbf{x} = [a_0 \quad a_1 \quad \cdots \quad a_c]^T$$

Minimax Problem

- The most appropriate approach for the solution of the optimization problem just described is to solve the *minimax* problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad \{\max_{\omega} |E(\omega)|\}$$

where

$$\mathbf{x} = [a_0 \quad a_1 \quad \cdots \quad a_c]^T$$

- By virtue of the so-called *alternation theorem*, there is a *unique equiripple* solution of the above minimax problem.

Minimax Problem

- The most appropriate approach for the solution of the optimization problem just described is to solve the *minimax* problem

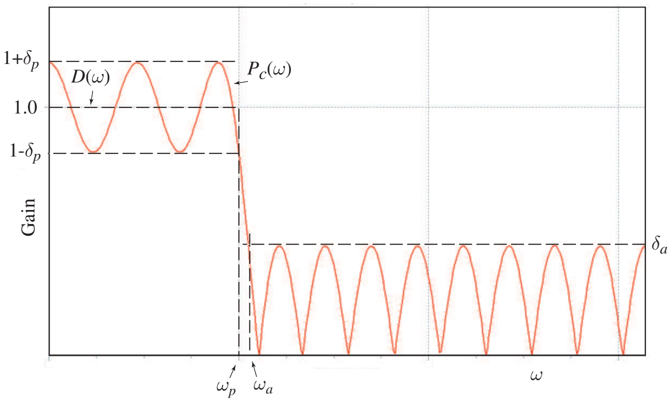
$$\underset{\mathbf{x}}{\text{minimize}} \quad \{\max_{\omega} |E(\omega)|\}$$

where

$$\mathbf{x} = [a_0 \quad a_1 \quad \cdots \quad a_c]^T$$

- By virtue of the so-called *alternation theorem*, there is a *unique equiripple* solution of the above minimax problem.

- *Note* that weighted-Chebyshev filters are so called because they have an *equiripple* amplitude response just like Chebyshev filters but are not related to Chebyshev filters in any other way.



1+δ_p
1.0
1-δ_p
Gain
$D(\omega)$
$P_c(\omega)$
δ_a
ω_p
ω_a
ω

Alternation Theorem

- If $P_c(\omega)$ is a linear combination of $r = c + 1$ cosine functions of the form

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega$$

then a necessary and sufficient condition that $P_c(\omega)$ be the unique, best, weighted-Chebyshev approximation to a continuous function $D(\omega)$ on Ω, where Ω is a dense and compact subset of the frequency interval $[0,\ \pi]$, is that the weighted error function $E(\omega)$ exhibit at least $r + 1$ *extremal frequencies* $\hat{\omega}_i$ in Ω such that

$$\hat{\omega}_0 < \hat{\omega}_1 < \cdots < \hat{\omega}_r$$

$$E(\hat{\omega}_{i+1}) = -E(\hat{\omega}_i) \quad \text{for} \quad i = 0,\ 1,\ \ldots,\ r-1$$

and

$$|E(\hat{\omega}_i)| = \max_{\omega \in \Omega} |E(\omega)| \quad \text{for} \quad i = 0,\ 1,\ \ldots,\ r$$

Notes:

- A subset Ω is *dense* if it has a sufficiently large number of members for the application at hand.
- A subset Ω is *compact* if it is closed and bounded.
- A subset is *closed* if all its limits are members of the set.
- A subset is *bounded* if all its members are bounded.

- From the alternation theorem and Eq. (B), i.e.,

$$E(\omega) = W(\omega)[D(\omega) - P_c(\omega)] \qquad \text{(B)}$$

we can write

$$E(\hat{\omega}_i) = W(\hat{\omega}_i)[D(\hat{\omega}_i) - P_c(\hat{\omega}_i)] = (-1)^i\delta$$

for $i = 0, 1, \ldots, r$, where δ is a constant.

- From the alternation theorem and Eq. (B), i.e.,

$$E(\omega) = W(\omega)[D(\omega) - P_c(\omega)] \qquad \text{(B)}$$

we can write

$$E(\hat{\omega}_i) = W(\hat{\omega}_i)[D(\hat{\omega}_i) - P_c(\hat{\omega}_i)] = (-1)^i\delta$$

for $i = 0, 1, \ldots, r$, where δ is a constant.

- The above system of equations can be put in matrix form as

$$\begin{bmatrix} 1 & \cos\hat{\omega}_0 & \cos 2\hat{\omega}_0 & \cdots & \cos c\hat{\omega}_0 & \frac{1}{W(\hat{\omega}_0)} \\ 1 & \cos\hat{\omega}_1 & \cos 2\hat{\omega}_1 & \cdots & \cos c\hat{\omega}_1 & \frac{-1}{W(\hat{\omega}_1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \cos\hat{\omega}_r & \cos 2\hat{\omega}_r & \cdots & \cos c\hat{\omega}_r & \frac{(-1)^r}{W(\hat{\omega}_r)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_c \\ \delta \end{bmatrix} = \begin{bmatrix} D(\hat{\omega}_0) \\ D(\hat{\omega}_1) \\ \vdots \\ D(\hat{\omega}_{r-1}) \\ D(\hat{\omega}_r) \end{bmatrix}$$

- If the extremal frequencies (or extremals for short) were known, coefficients a_k and, in turn, the frequency response of the filter could be computed using Eq. (A), i.e.,

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega \qquad \text{(A)}$$

- If the extremal frequencies (or extremals for short) were known, coefficients a_k and, in turn, the frequency response of the filter could be computed using Eq. (A), i.e.,

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega \qquad \text{(A)}$$

- The solution of this system exists since the above $(r+1) \times (r+1)$ matrix is known to be *nonsingular*.

Basic Remez Exchange Algorithm

- The Remez exchange algorithm is an *iterative multivariable algorithm* that is naturally suited for the solution of the minimax problem just described.

  It is based on the *second optimization method of Remez*.

Basic Remez Exchange Algorithm *Cont'd*

1. Initialize extremal frequencies $\hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_r$ and ensure that an extremal is assigned at each band edge.

1. Initialize extremal frequencies $\hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_r$ and ensure that an extremal is assigned at each band edge.
2. Locate the frequencies $\widehat{\omega}_0, \widehat{\omega}_1, \ldots, \widehat{\omega}_\rho$ at which the magnitude of the error

$$|E(\omega)| = |W(\omega)[D(\omega) - P_c(\omega)]|$$

   is maximum and $|E(\widehat{\omega}_i)| \geq \delta$ (these frequencies are *potential extremals* for the next iteration).

1. Initialize extremal frequencies $\hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_r$ and ensure that an extremal is assigned at each band edge.
2. Locate the frequencies $\widehat{\omega}_0, \widehat{\omega}_1, \ldots, \widehat{\omega}_\rho$ at which the magnitude of the error

$$|E(\omega)| = |W(\omega)[D(\omega) - P_c(\omega)]|$$

   is maximum and $|E(\widehat{\omega}_i)| \geq \delta$ (these frequencies are *potential extremals* for the next iteration).
3. Compute the convergence parameter

$$Q = \frac{\max |E(\widehat{\omega}_i)| - \min |E(\widehat{\omega}_i)|}{\max |E(\widehat{\omega}_i)|}$$

   where $i = 0, 1, \ldots, \rho$.

4. Reject $\rho - r$ *superfluous potential extremals* $\widehat{\omega}_i$ according to an appropriate rejection criterion and renumber the remaining $\widehat{\omega}_i$ by setting $\hat{\omega}_i = \widehat{\omega}_i$ for $i = 0, 1, \ldots, r$.

4. Reject $\rho - r$ *superfluous potential extremals* $\overset{\frown}{\omega}_i$ according to an appropriate rejection criterion and renumber the remaining $\overset{\frown}{\omega}_i$ by setting $\hat{\omega}_i = \overset{\frown}{\omega}_i$ for $i = 0, 1, \ldots, r$.

5. If $Q > \varepsilon$, where ε is a convergence tolerance (say $\varepsilon = 0.01$), repeat from step 2; otherwise continue to step 6.

4. Reject $\rho - r$ *superfluous potential extremals* $\widehat{\omega}_i$ according to an appropriate rejection criterion and renumber the remaining $\widehat{\omega}_i$ by setting $\hat{\omega}_i = \widehat{\omega}_i$ for $i = 0, 1, \ldots, r$.
5. If $Q > \varepsilon$, where ε is a convergence tolerance (say $\varepsilon = 0.01$), repeat from step 2; otherwise continue to step 6.
6. Compute $P_c(\omega)$ using the last set of extremal frequencies; then deduce $h(n)$, the impulse response of the required filter, and stop.

Initialization of Extremal Frequencies

The implementation of the basic Remez algorithm can be accomplished as follows:

Step 1:

- A simple initialization scheme is to distribute the extremals uniformly in each passband and stopband such that

The implementation of the basic Remez algorithm can be accomplished as follows:

Step 1:

- A simple initialization scheme is to distribute the extremals uniformly in each passband and stopband such that
  - the total number of extremals is exactly equal to $r + 1 = (N + 3)/2$,

Initialization of Extremal Frequencies

The implementation of the basic Remez algorithm can be accomplished as follows:

Step 1:

- A simple initialization scheme is to distribute the extremals uniformly in each passband and stopband such that
  - the total number of extremals is exactly equal to $r + 1 = (N + 3)/2$,
  - the number of extremals in each passband or stopband is proportional to the bandwidth of the passband or stopband,

Initialization of Extremal Frequencies

The implementation of the basic Remez algorithm can be accomplished as follows:

Step 1:

- A simple initialization scheme is to distribute the extremals uniformly in each passband and stopband such that
  - the total number of extremals is exactly equal to $r + 1 = (N + 3)/2$,
  - the number of extremals in each passband or stopband is proportional to the bandwidth of the passband or stopband,
  - there is an extremal at each band edge.

The implementation of the basic Remez algorithm can be accomplished as follows:

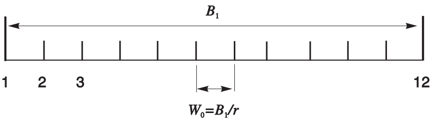
Step 1:

- A simple initialization scheme is to distribute the extremals uniformly in each passband and stopband such that
  - the total number of extremals is exactly equal to $r + 1 = (N + 3)/2$,
  - the number of extremals in each passband or stopband is proportional to the bandwidth of the passband or stopband,
  - there is an extremal at each band edge.

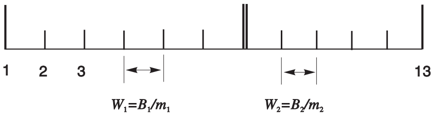
Such a scheme is illustrated in the next slide.

Initialization of Extremal Frequencies *Cont'd*

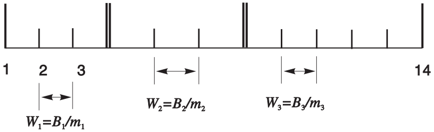
Bands: 1
Extremals: r+1 (12)
Intervals: r (11)



Bands: 2
Extremals: r+1 (13)
Intervals: r-1 (11)



Bands: 3
Extremals: r+1 (14)
Intervals: r-2 (11)



- For a filter with J bands with bandwidths B_1, B_2, ..., B_J, the number of extremals and intervals between extremals for each band can be calculated by using the formulas

$$W_0 = \frac{1}{r+1-J}\sum_{j=1}^{J} B_j$$

$$m_j = \text{int}\left(\frac{B_j}{W_0}+0.5\right) \quad \text{for} \;\; j = 1, 2, \ldots, J-1$$

$$\text{and} \quad m_J = r - \sum_{j=1}^{J-1}(m_j+1)$$

$$W_j = \frac{B_j}{m_j} \quad \text{for} \;\; j = 1, 2, \ldots, J$$

where $r = (N+1)/2$ and N is the filter length.

Step 2:

- In order to locate the frequencies $\widehat{\omega}_0, \widehat{\omega}_1, \ldots, \widehat{\omega}_\rho$ at which $|E(\omega)|$ is maximum such that $|E(\widehat{\omega}_i)| \geq \delta$, we calculate coefficients $a_0, a_1, \ldots, a_c$ and parameter δ by solving the system

$$\begin{bmatrix} 1 & \cos\hat{\omega}_0 & \cos 2\hat{\omega}_0 & \cdots & \cos c\hat{\omega}_0 & \frac{1}{W(\hat{\omega}_0)} \\ 1 & \cos\hat{\omega}_1 & \cos 2\hat{\omega}_1 & \cdots & \cos c\hat{\omega}_1 & \frac{-1}{W(\hat{\omega}_1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \cos\hat{\omega}_r & \cos 2\hat{\omega}_r & \cdots & \cos c\hat{\omega}_r & \frac{(-1)^r}{W(\hat{\omega}_r)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_c \\ \delta \end{bmatrix} = \begin{bmatrix} D(\hat{\omega}_0) \\ D(\hat{\omega}_1) \\ \vdots \\ D(\hat{\omega}_{r-1}) \\ D(\hat{\omega}_r) \end{bmatrix}$$

- With coefficients $a_0, a_1, \ldots, a_c$ known, polynomial

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega$$

can be calculated.

Updating of Extremals *Cont'd*

- With coefficients $a_0, a_1, \ldots, a_c$ known, polynomial

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega$$

can be calculated.

- With $P_c(\omega)$ known, the error function

$$|E(\omega)| = |W(\omega)[D(\omega) - P_c(\omega)]|$$

can be calculated.

Updating of Extremals *Cont'd*

- The maxima of the error function can be obtained by evaluating $|E(\omega)|$ over a dense set of frequencies in the passband(s) and stopband(s) of the required filter.

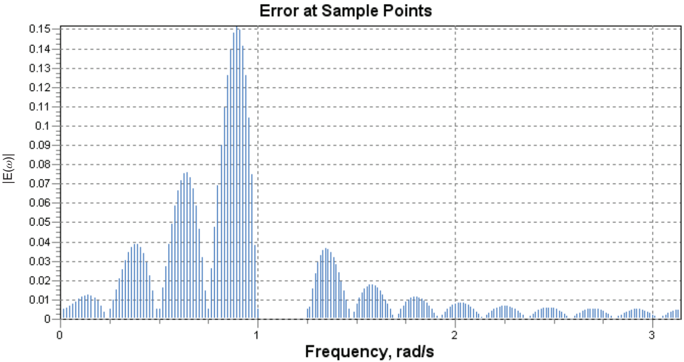
- The maxima of the error function can be obtained by evaluating $|E(\omega)|$ over a dense set of frequencies in the passband(s) and stopband(s) of the required filter.

- A sufficient number of frequency points for most applications is around 16 sample points per ripple in $|E(\omega)|$, i.e., $8(N+1)$.

- The maxima of the error function can be obtained by evaluating $|E(\omega)|$ over a dense set of frequencies in the passband(s) and stopband(s) of the required filter.
- A sufficient number of frequency points for most applications is around 16 sample points per ripple in $|E(\omega)|$, i.e., $8(N+1)$.
- An actual plot of $|E(\omega)|$ versus ω is shown in the next slide.

Filter length: 27
Iteration no: 1

Function Evals: 0



Updating of Extremals *Cont'd*

- The approach just described is easy to apply.

  However, it is *inefficient* and may be subject to numerical ill-conditioning in particular if δ is small and N is large.

  Note that a 50×50 matrix is quite typical and a 100×100 matrix is not unusual.

- An alternative and *more efficient* approach is to deduce δ analytically (by using Cramer's rule) and then interpolate $P_c(\omega)$ on the r frequency points using the *barycentric* form of the Lagrange interpolation formula, as follows:

- An alternative and *more efficient* approach is to deduce δ analytically (by using Cramer's rule) and then interpolate $P_c(\omega)$ on the r frequency points using the *barycentric* form of the Lagrange interpolation formula, as follows:

- Calculate parameter δ as

$$\delta = \sum_{k=0}^{r} \frac{\alpha_k D(\hat{\omega}_k)}{\frac{\sum_{k=0}^{r} (-1)^k \alpha_k}{W(\hat{\omega})}}$$

- With δ and

$$P_c(\hat{\omega}_k) = C_k = D(\hat{\omega}_k) - (-1)^k \frac{\delta}{W(\hat{\omega}_k)}$$

known, the following interpolation formula can be constructed:

$$P_c(\omega) = \begin{cases} C_k & \text{for } \omega = \hat{\omega}_0,\, \hat{\omega}_1,\, \ldots,\, \hat{\omega}_{r-1} \\ \dfrac{\displaystyle\sum_{k=0}^{r-1} \frac{\beta_k C_k}{x - x_k}}{\displaystyle\sum_{k=0}^{r-1} \frac{\beta_k}{x - x_k}} & \text{otherwise} \end{cases}$$

where $\quad \alpha_k = \prod_{i=0,\ i \neq k}^{r} \frac{1}{x_k - x_i}, \quad \beta_k = \prod_{i=0,\ i \neq k}^{r-1} \frac{1}{x_k - x_i}$

and $\quad x = \cos\omega \quad$ and $\quad x_i = \cos\hat{\omega}_i \quad$ for $\quad i = 0,\, 1,\, \ldots,\, r$

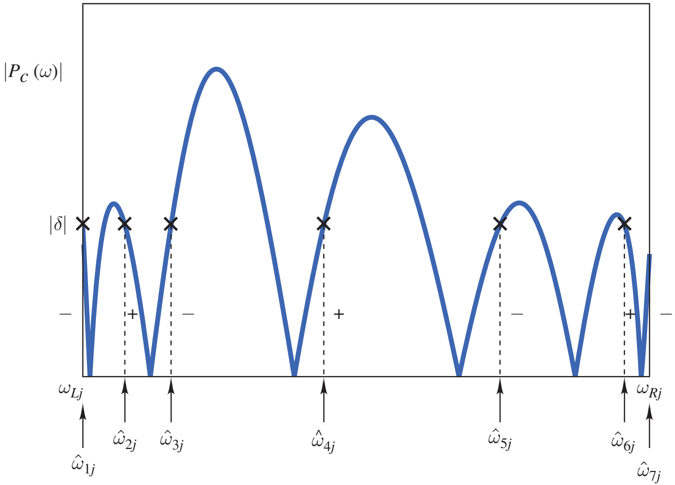
- Using the interpolation formula, the value of $P_c(\omega)$ for any frequency ω can be computed.

Updating of Extremals *Cont'd*

- Using the interpolation formula, the value of $P_c(\omega)$ for any frequency ω can be computed.
- Since $W(\omega)$ and $D(\omega)$ are known, the error function

$$|E(\omega)| = |W(\omega)[D(\omega) - P_c(\omega)]|$$

and, in turn, the frequencies $\widehat{\omega}_0, \widehat{\omega}_1, \ldots, \widehat{\omega}_\rho$ at which $|E(\omega)|$ is maximum can be deduced.


$|P_c(\omega)|$
$|\delta|$
−
+
−
+
−
+
−
ω_{Lj}
ω_{Rj}
$\hat{\omega}_{1j}$
$\hat{\omega}_{2j}$
$\hat{\omega}_{3j}$
$\hat{\omega}_{4j}$
$\hat{\omega}_{5j}$
$\hat{\omega}_{6j}$
$\hat{\omega}_{7j}$

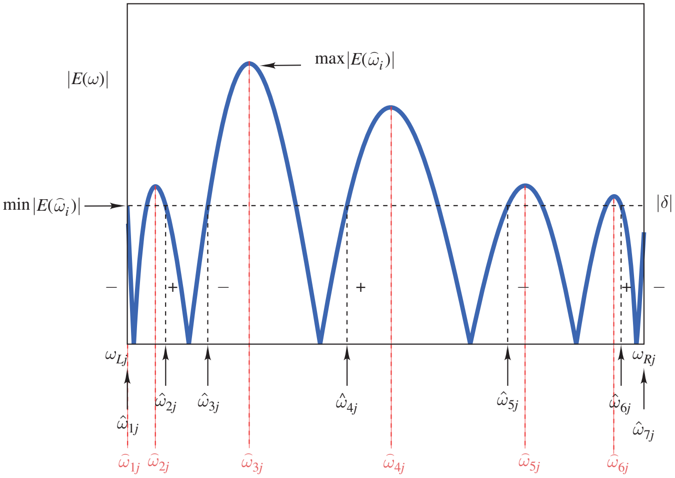
Convergence Parameter

Step 3:

- Compute the convergence parameter

$$Q = \frac{\max |E(\widehat{\omega}_i)| - \min |E(\widehat{\omega}_i)|}{\max |E(\widehat{\omega}_i)|}$$

Convergence Parameter *Cont'd*



Step 4:

- The problem formulation is such that *there must be exactly $r+1$ extremals* in each iteration.

Step 4:

- The problem formulation is such that *there must be exactly* $r+1$ *extremals* in each iteration.
- Analysis will show that $|E(\omega)|$ can have as many as $r+2J-1$ maxima where J is the number of bands:

Rejection of Superfluous Potential Extremals

Step 4:

- The problem formulation is such that *there must be exactly $r+1$ extremals* in each iteration.
- Analysis will show that $|E(\omega)|$ can have as many as $r+2J-1$ maxima where J is the number of bands:
  - For a 1-band filter (differentiators): r+1 (no extra maxima)

Step 4:

- The problem formulation is such that *there must be exactly $r+1$ extremals* in each iteration.
- Analysis will show that $|E(\omega)|$ can have as many as $r+2J-1$ maxima where J is the number of bands:
  - For a 1-band filter (differentiators): r+1 (no extra maxima)
  - For a 2-band filter (lowpass or highpass filter): r+3 (2 extra maxima)

Rejection of Superfluous Potential Extremals

Step 4:

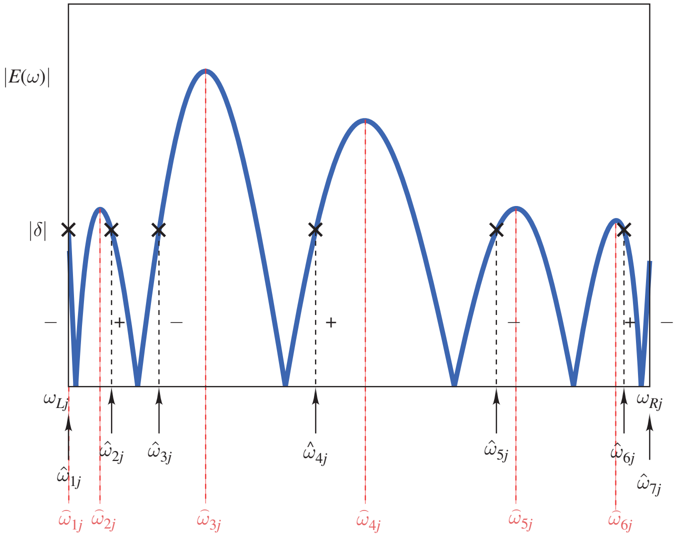
- The problem formulation is such that *there must be exactly $r+1$ extremals* in each iteration.
- Analysis will show that $|E(\omega)|$ can have as many as $r+2J-1$ maxima where J is the number of bands:
  - For a 1-band filter (differentiators): r+1 (no extra maxima)
  - For a 2-band filter (lowpass or highpass filter): r+3 (2 extra maxima)
  - For a 3-band filter (bandpass or bandstop filter): r+5 (4 extra maxima)

Rejection of Superfluous Potential Extremals

Step 4:

- The problem formulation is such that *there must be exactly $r + 1$ extremals* in each iteration.
- Analysis will show that $|E(\omega)|$ can have as many as $r + 2J - 1$ maxima where J is the number of bands:
  - For a 1-band filter (differentiators): r+1 (no extra maxima)
  - For a 2-band filter (lowpass or highpass filter): r+3 (2 extra maxima)
  - For a 3-band filter (bandpass or bandstop filter): r+5 (4 extra maxima)
- If in any iteration the number of maxima exceeds $r + 1$, then the iteration is said to have generated *superfluous potential extremals*.

- In the standard McClellan, Rabiner, and Parks algorithm, this difficulty is circumvented by rejecting the $\rho - r$ potential extremals $\widehat{\omega}_i$ that yield the lowest error $|E(\omega)|$.


$|E(\omega)|$
$|\delta|$
$-$
$+$
$-$
$+$
$-$
$+$
$-$
ω_{Lj}
ω_{Rj}
$\hat{\omega}_{1j}$
$\hat{\omega}_{2j}$
$\hat{\omega}_{3j}$
$\hat{\omega}_{4j}$
$\hat{\omega}_{5j}$
$\hat{\omega}_{6j}$
$\hat{\omega}_{7j}$
$\widehat{\omega}_{1j}$
$\widehat{\omega}_{2j}$
$\widehat{\omega}_{3j}$
$\widehat{\omega}_{4j}$
$\widehat{\omega}_{5j}$
$\widehat{\omega}_{6j}$

Check for Convergence

Step 5:

- If the convergence parameter is not small enough, i.e., if the ripples have not equalized sufficiently, repeat from Step 2.

Step 6:

- The impulse response can be determined by recalling that function $P_c(\omega)$ is the frequency response of a noncausal version of the required filter.

Computation of Impulse Response

Step 6:

- The impulse response can be determined by recalling that function $P_c(\omega)$ is the frequency response of a noncausal version of the required filter.

- The impulse response of the noncausal filter, denoted as $h_0(n)$ for $-c \leq n \leq c$, can be determined by computing $P_c(k\Omega)$ for $k = 0, 1, \ldots, c$ where $\Omega = 2\pi/N$, and then using the inverse discrete Fourier transform.

- It can be shown that

$$h_0(n) = h_0(-n) = \frac{1}{N}\left\{P_c(0) + \sum_{k=1}^{c} 2P_c(k\Omega)\cos\left(\frac{2\pi kn}{N}\right)\right\}$$

for $n = 0, 1, \ldots, c$.

Computation of Impulse Response *Cont'd*

- It can be shown that

$$h_0(n) = h_0(-n) = \frac{1}{N}\left\{P_c(0) + \sum_{k=1}^{c} 2P_c(k\Omega)\cos\left(\frac{2\pi kn}{N}\right)\right\}$$

  for $n = 0, 1, \ldots, c$.

- The impulse response of the required causal filter is given by

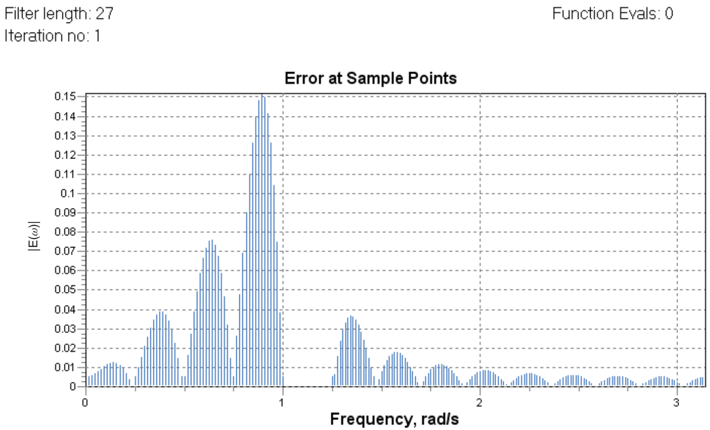
$$h(n) = h_0(n - c)$$

  for $n = 0, 1, \ldots, c$.

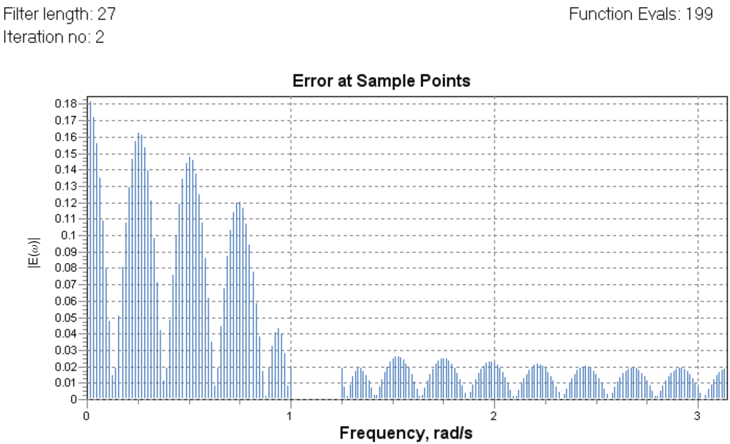
Example

Band	$D(\omega)$	$W(\omega)$	Left band edge	Right band edge
1	1	1	0	1.0
2	0	0.4	1.25	π
Sampling frequency: 2π				

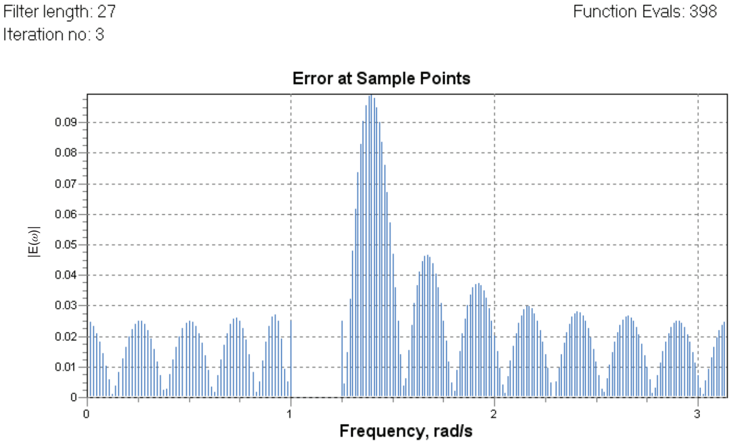
Example *Cont'd*



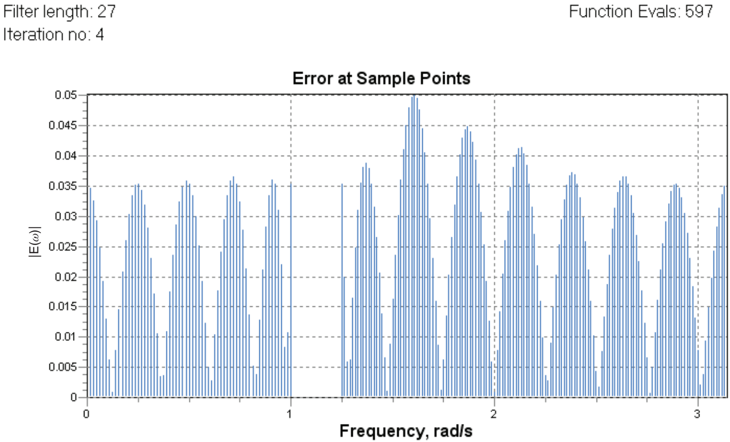
Example *Cont'd*



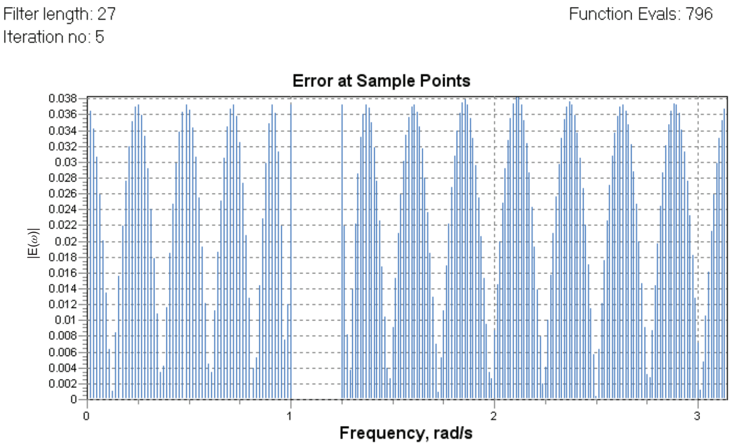
Example *Cont'd*



Example *Cont'd*



Example *Cont'd*

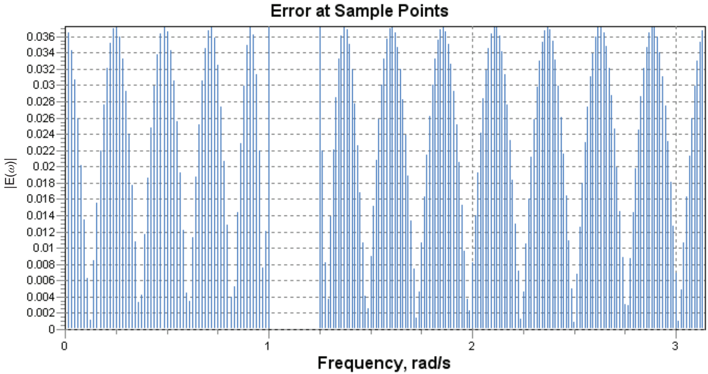


Example *Cont'd*

Filter length: 27

Iteration no: 6

Function Evals: 995



Computational Complexity

- The Remez exchange described is using an *exhaustive search* to identify the maxima of $|E(\omega)|$.

Computational Complexity

- The Remez exchange described is using an *exhaustive search* to identify the maxima of $|E(\omega)|$.
- Consider a filter of length N and assume that $|E(\omega)|$ is evaluated at S sample points per ripple.

Computational Complexity

- The Remez exchange described is using an *exhaustive search* to identify the maxima of $|E(\omega)|$.
- Consider a filter of length N and assume that $|E(\omega)|$ is evaluated at S sample points per ripple.
- The algorithm presented would require $S \times (N+1)/2$ function evaluations.

Computational Complexity

- The Remez exchange described is using an *exhaustive search* to identify the maxima of $|E(\omega)|$.
- Consider a filter of length N and assume that $|E(\omega)|$ is evaluated at S sample points per ripple.
- The algorithm presented would require $S \times (N+1)/2$ function evaluations.
- One function evaluation requires:

Computational Complexity

- The Remez exchange described is using an *exhaustive search* to identify the maxima of $|E(\omega)|$.
- Consider a filter of length N and assume that $|E(\omega)|$ is evaluated at S sample points per ripple.
- The algorithm presented would require $S \times (N + 1)/2$ function evaluations.
- One function evaluation requires:
  - $N - 1$ additions

Computational Complexity

- The Remez exchange described is using an *exhaustive search* to identify the maxima of $|E(\omega)|$.
- Consider a filter of length N and assume that $|E(\omega)|$ is evaluated at S sample points per ripple.
- The algorithm presented would require $S \times (N+1)/2$ function evaluations.
- One function evaluation requires:
  - $N - 1$ additions
  - $(N+1)/2$ multiplications

- The Remez exchange described is using an *exhaustive search* to identify the maxima of $|E(\omega)|$.
- Consider a filter of length N and assume that $|E(\omega)|$ is evaluated at S sample points per ripple.
- The algorithm presented would require $S \times (N+1)/2$ function evaluations.
- One function evaluation requires:
  - $N-1$ additions
  - $(N+1)/2$ multiplications
  - $(N+1)/2$ divisions

Computational Complexity *Cont'd*

- A Remez optimization usually requires
  - 4 to 8 iterations for lowpass or highpass filters,
  - 6 to 10 iterations for bandpass filters, and
  - 8 to 12 iterations for bandstop filters.

- A Remez optimization usually requires
  - 4 to 8 iterations for lowpass or highpass filters,
  - 6 to 10 iterations for bandpass filters, and
  - 8 to 12 iterations for bandstop filters.
- If prescribed specifications are to be achieved and the appropriate value of N is unknown, typically two to four Remez optimizations have to be performed.

- For example, if
  - $N = 101$,
  - $S = 16$,
  - number of Remez optimizations $= 4$,
  - iterations per optimization $= 6$,

  the design would entail 24 iterations, 19,200 function evaluations, 1.92×10^6 additions, 0.979×10^6 multiplications, and 0.979×10^6 divisions.

- For example, if
  - $N = 101$,
  - $S = 16$,
  - number of Remez optimizations = 4,
  - iterations per optimization = 6,

  the design would entail 24 iterations, 19,200 function evaluations, 1.92×10^6 additions, 0.979×10^6 multiplications, and 0.979×10^6 divisions.

- This is in addition to the computation required for the evaluation of δ and coefficients α_k, C_k, and β_k once per iteration.

- For example, if
  - $N = 101$,
  - $S = 16$,
  - number of Remez optimizations = 4,
  - iterations per optimization = 6,

  the design would entail 24 iterations, 19,200 function evaluations, 1.92×10^6 additions, 0.979×10^6 multiplications, and 0.979×10^6 divisions.

- This is in addition to the computation required for the evaluation of δ and coefficients α_k, C_k, and β_k once per iteration.

- In effect, the amount of computation required to complete a design *is quite substantial*.

- When the system of equations

$$\begin{bmatrix} 1 & \cos\hat{\omega}_0 & \cos 2\hat{\omega}_0 & \cdots & \cos c\hat{\omega}_0 & \frac{1}{W(\hat{\omega}_0)} \\ 1 & \cos\hat{\omega}_1 & \cos 2\hat{\omega}_1 & \cdots & \cos c\hat{\omega}_1 & \frac{-1}{W(\hat{\omega}_1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \cos\hat{\omega}_r & \cos 2\hat{\omega}_r & \cdots & \cos c\hat{\omega}_r & \frac{(-1)^r}{W(\hat{\omega}_r)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_c \\ \delta \end{bmatrix} = \begin{bmatrix} D(\hat{\omega}_0) \\ D(\hat{\omega}_1) \\ \vdots \\ D(\hat{\omega}_{r-1}) \\ D(\hat{\omega}_r) \end{bmatrix}$$

is solved, the error function $|E(\omega)|$ is forced to satisfy the relation

$$|E(\hat{\omega}_i)| = |W(\hat{\omega}_i)[D(\hat{\omega}_i) - P_c(\hat{\omega}_i)]| = |\delta|$$

- When the system of equations

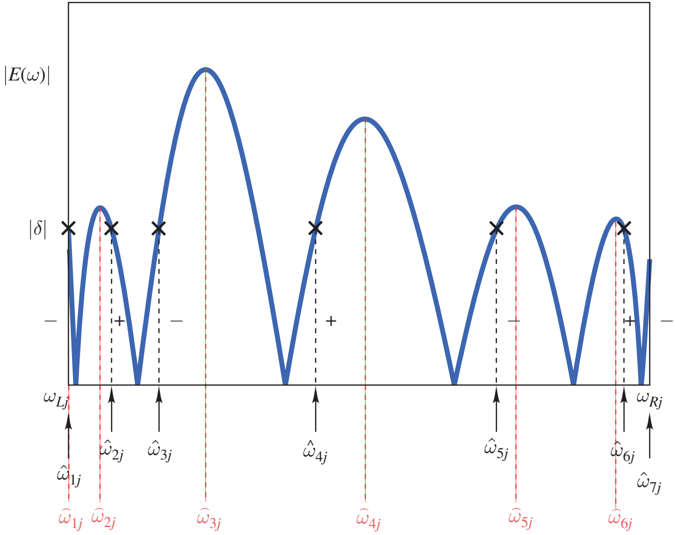
$$\begin{bmatrix} 1 & \cos\hat{\omega}_0 & \cos 2\hat{\omega}_0 & \cdots & \cos c\hat{\omega}_0 & \frac{1}{W(\hat{\omega}_0)} \\ 1 & \cos\hat{\omega}_1 & \cos 2\hat{\omega}_1 & \cdots & \cos c\hat{\omega}_1 & \frac{-1}{W(\hat{\omega}_1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \cos\hat{\omega}_r & \cos 2\hat{\omega}_r & \cdots & \cos c\hat{\omega}_r & \frac{(-1)^r}{W(\hat{\omega}_r)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_c \\ \delta \end{bmatrix} = \begin{bmatrix} D(\hat{\omega}_0) \\ D(\hat{\omega}_1) \\ \vdots \\ D(\hat{\omega}_{r-1}) \\ D(\hat{\omega}_r) \end{bmatrix}$$

  is solved, the error function $|E(\omega)|$ is forced to satisfy the relation

$$|E(\hat{\omega}_i)| = |W(\hat{\omega}_i)[D(\hat{\omega}_i) - P_c(\hat{\omega}_i)]| = |\delta|$$

- This relation can be satisfied in a number of ways but the most likely possibility for the jth band is illustrated in the next slide where ω_{Lj} and ω_{Rj} are the left-hand and right-hand edges, respectively.

Selective Step-by-Step Search *Cont'd*

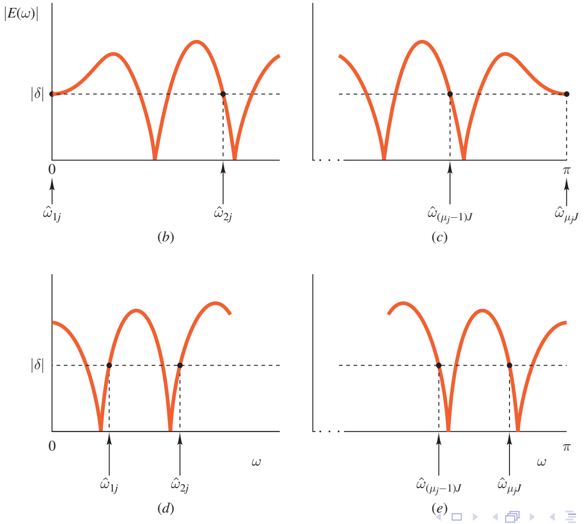


- Because of the special nature of the error function
  (a) the maxima of $|E(\omega)|$ can be easily found by searching in the vicinity of the extremals;
  (b) gradient information can be used to expedite the search for the maxima of $|E(\omega)|$; and
  (c) the closer we get to the solution, the closer are the maxima of the error function to the extremals.

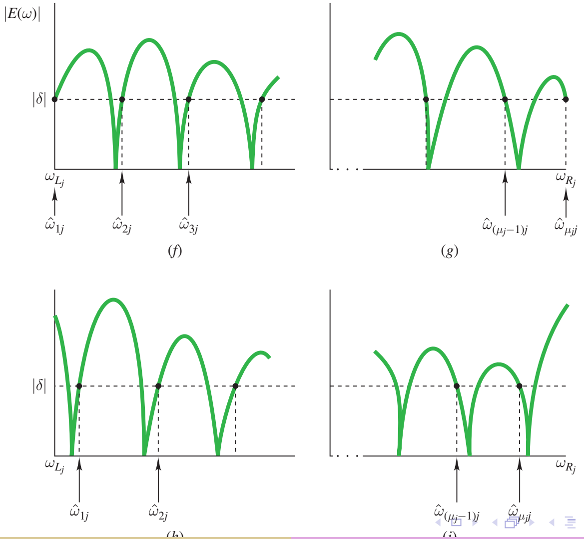
Selective Step-by-Step Search *Cont'd*

- Because of the special nature of the error function
  (a) the maxima of $|E(\omega)|$ can be easily found by searching in the vicinity of the extremals;
  (b) gradient information can be used to expedite the search for the maxima of $|E(\omega)|$; and
  (c) the closer we get to the solution, the closer are the maxima of the error function to the extremals.
- By using a *selective step-by-step search*, a large amount of computation can be eliminated.

- Extra ripples can arise in the first and last bands:



- Also in interior bands:



Cubic Interpolation Search

- Increased computational efficiency can be achieved by using a search based on *cubic interpolation*.

Cubic Interpolation Search

- Increased computational efficiency can be achieved by using a search based on *cubic interpolation*.
- Assuming that the magnitude of the error can be represented by the third-order polynomial

$$|E(\omega)| = M = a + b\omega + c\omega^2 + d\omega^3$$

where $a,\ b,\ c$, and d are constants then

$$\frac{dM}{d\omega} = G = b + 2c\omega + 3d\omega^2$$

Hence, the frequencies at which M has stationary points are given by

$$\bar{\omega} = \frac{1}{3d}\left[-c \pm \sqrt{(c^2 - 3bd)}\right]$$

Cubic Interpolation Search

- Increased computational efficiency can be achieved by using a search based on *cubic interpolation*.
- Assuming that the magnitude of the error can be represented by the third-order polynomial

$$|E(\omega)| = M = a + b\omega + c\omega^2 + d\omega^3$$

where a, b, c, and d are constants then

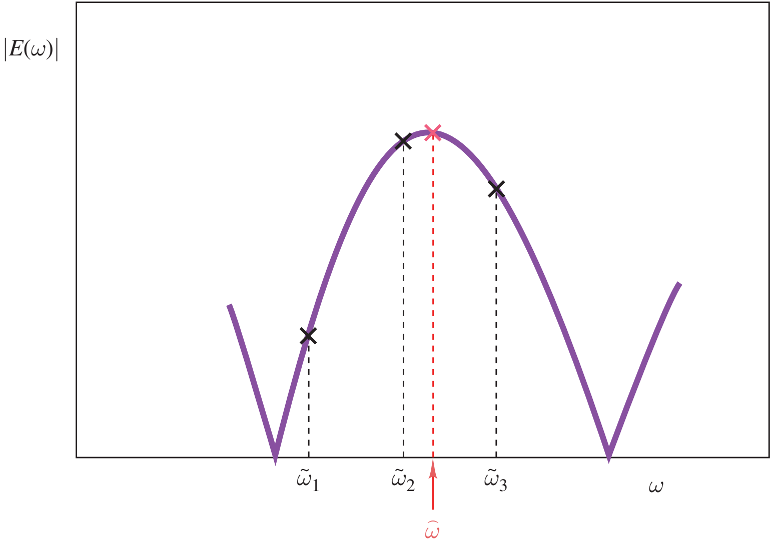
$$\frac{dM}{d\omega} = G = b + 2c\omega + 3d\omega^2$$

Hence, the frequencies at which M has stationary points are given by

$$\bar{\omega} = \frac{1}{3d}\left[-c \pm \sqrt{(c^2 - 3bd)}\right]$$

- Therefore, $|E(\omega)|$ has a maximum if

$$\frac{d^2M}{d\omega^2} = 2c + 6d\widehat{\omega} < 0 \quad \text{or} \quad \widehat{\omega} < -\frac{c}{3d}$$


$|E(\omega)|$
$\tilde{\omega}_1$
$\tilde{\omega}_2$
$\tilde{\omega}_3$
ω
$\hat{\omega}$

Cubic Interpolation Search *Cont'd*

- The cubic interpolation method requires four function evaluations per potential extremal consistently.

Cubic Interpolation Search *Cont'd*

- The cubic interpolation method requires four function evaluations per potential extremal consistently.
- The selective step-by-step search may require as many as eight function evaluations per potential extremal in the first two or three iterations but as the solution is approached only two or three function evaluations are required.

- The cubic interpolation method requires four function evaluations per potential extremal consistently.
- The selective step-by-step search may require as many as eight function evaluations per potential extremal in the first two or three iterations but as the solution is approached only two or three function evaluations are required.
- By using the cubic interpolation to start with and then switching over to the step-by-step search, a very efficient algorithm can be constructed.

Cubic Interpolation Search *Cont'd*

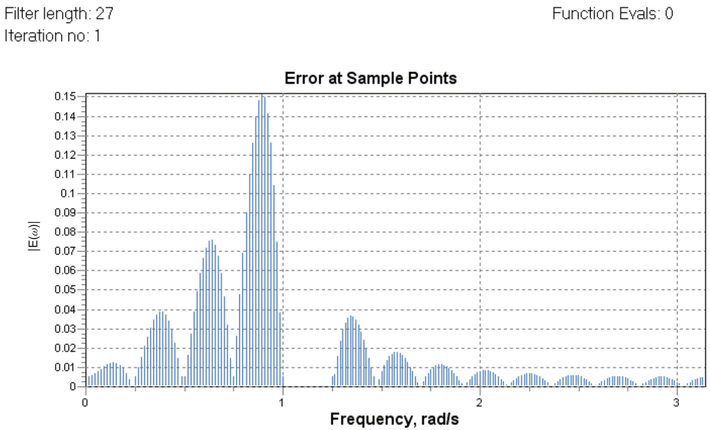
- The cubic interpolation method requires four function evaluations per potential extremal consistently.
- The selective step-by-step search may require as many as eight function evaluations per potential extremal in the first two or three iterations but as the solution is approached only two or three function evaluations are required.
- By using the cubic interpolation to start with and then switching over to the step-by-step search, a very efficient algorithm can be constructed.
- The decision to switch from cubic to selective can be based on the value of the convergence parameter Q (see Step 5).

  Switching from the cubic to the selective when Q is reduced below 0.65 works well.

Improved Rejection Scheme for Superfluous Potential Extremals

- If an extremal does not move from one iteration to the next, then the minimum value of $E(\widehat{\omega}_i)$ is simply δ, as can be easily shown, and this happens quite often even in the first or second iteration of the Remez algorithm.

Improved Rejection Scheme *Cont'd*



- As a consequence, rejecting potential extremals on the basis of the individual values of $E(\widehat{\omega}_i)$ tends to become random and this can slow the Remez algorithm quite significantly particularly for multiband filters.

Improved Rejection Scheme *Cont'd*

- As a consequence, rejecting potential extremals on the basis of the individual values of $E(\widehat{\omega}_i)$ tends to become random and this can slow the Remez algorithm quite significantly particularly for multiband filters.
- An improved scheme for the rejection of superfluous extremals based the rejection on the lowest average band error as well as the individual values of $E(\widehat{\omega}_i)$ is described in the next slide.

- Compute the average band errors

$$E_j = \frac{1}{\nu_j} \sum_{\widehat{\omega}_i \in \Omega_j} |E(\widehat{\omega}_i)| \quad \text{for} \;\; j = 1, 2, \ldots, J$$

where Ω_j is the set of extremals in band j given by

$$\Omega_j = \{\widehat{\omega}_i : \omega_{Lj} \le \widehat{\omega}_i \le \omega_{Rj}\}$$

ν_j is the number of potential extremals in band j, and J is the number of bands.

Improved Rejection Scheme *Cont'd*

- Compute the average band errors

$$E_j = \frac{1}{\nu_j} \sum_{\widehat{\omega}_i \in \Omega_j} |E(\widehat{\omega}_i)| \quad \text{for} \quad j = 1, 2, \ldots, J$$

where Ω_j is the set of extremals in band j given by

$$\Omega_j = \{\widehat{\omega}_i : \omega_{Lj} \leq \widehat{\omega}_i \leq \omega_{Rj}\}$$

ν_j is the number of potential extremals in band j, and J is the number of bands.

- Rank the J bands in the order of lowest average error and let $l_1, l_2, \ldots, l_J$ be the ranked list obtained, i.e., l_1 and l_J are the bands with the lowest and highest average error, respectively.

- Reject one $\widehat{\omega}_i$ in each of bands $I_1,\ I_2,\ \ldots,\ I_{J-1},\ I_1,\ I_2,\ \ldots$ until $\rho - r$ superfluous $\widehat{\omega}_i$ are rejected. In each case, reject the $\widehat{\omega}_i$, other than a band edge, that yields the lowest $|E(\widehat{\omega}_i)|$ in the band.

Example:

If $J = 3$, $\rho - r = 3$, and the average errors for bands 1, 2, and 3 are 0.05, 0.08, and 0.02, then $\widehat{\omega}_i$ are rejected in bands 3, 1, and 3.

Note: The potential extremals are not rejected in band 2 which is the band of highest average error.

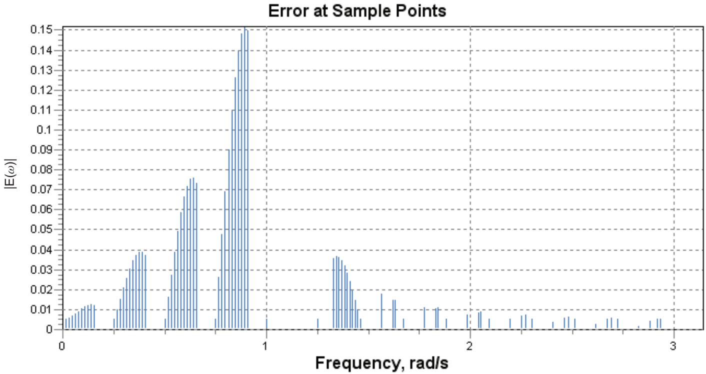
Example

Band	$D(\omega)$	$W(\omega)$	Left band edge	Right band edge
1	1	1	0	1.0
2	0	0.4	1.25	π
Sampling frequency: 2π				

Example *Cont'd*

Filter length: 27
Iteration no: 1

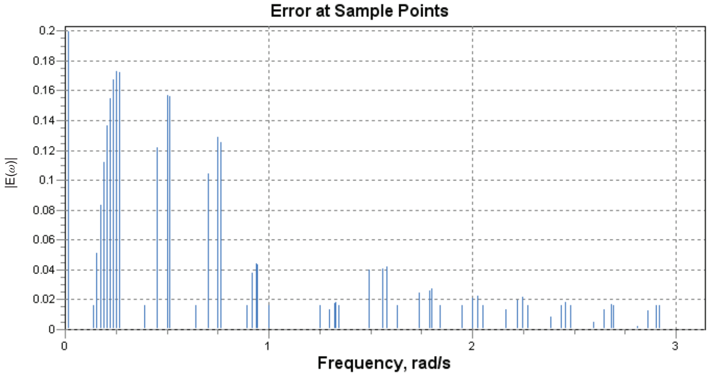
Function Evals: 0



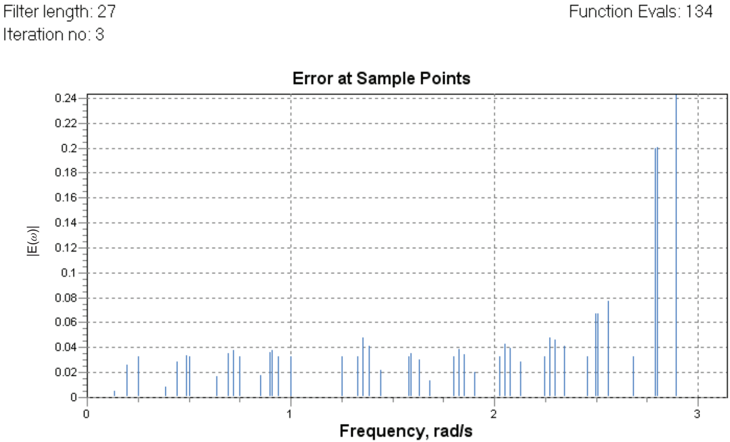
Example *Cont'd*

Filter length: 27 Function Evals: 87

Iteration no: 2



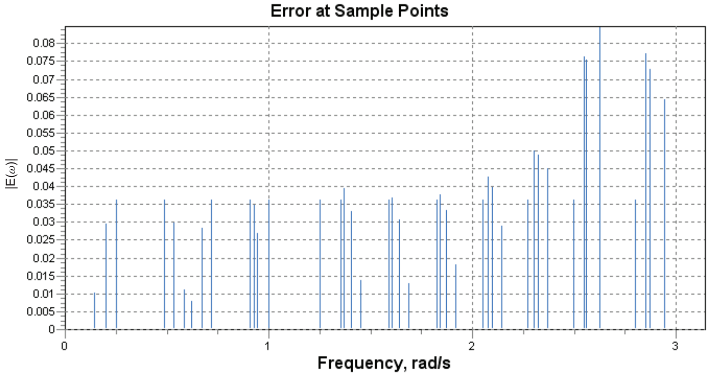
Example *Cont'd*



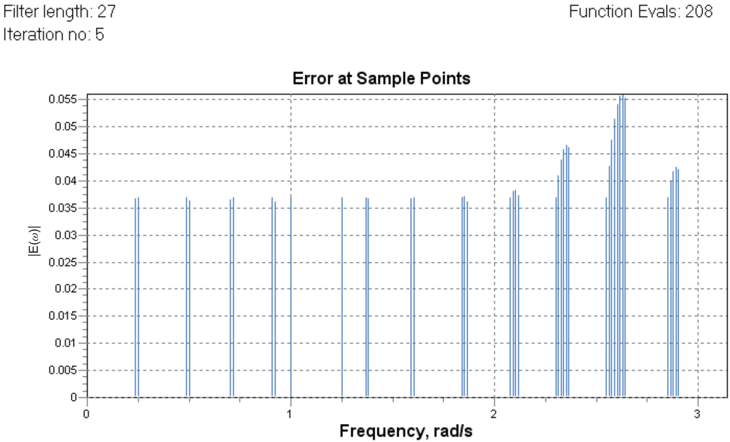
Example *Cont'd*

Filter length: 27
Iteration no: 4

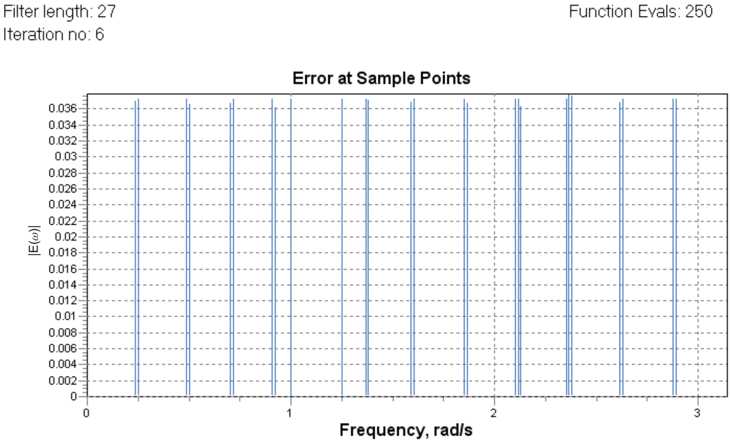
Function Evals: 171



Example *Cont'd*



Example *Cont'd*

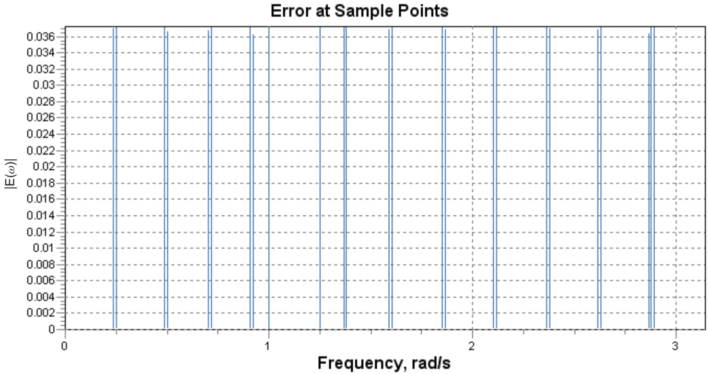


Example *Cont'd*

Filter length: 27

Iteration no: 7

Function Evals: 278



Comparisons — Amount of Computation

Type of Filter	No. of Examples	Range of N	Ave. Funct. Evals.			Saving, %	
			A	B	C	C v B	C v A
LP	45	9-101	2691	722	372	48.9	86.3
HP	42	9-101	2774	710	356	49.9	87.2
BP	44	21-89	2777	667	338	49.3	87.8
BS	35	21-91	2720	639	336	47.4	87.6

A: Exhaustive search

B: Selective search

C: Selective plus cubic search

Comparisons — Robustness

Type of Filter	No. of Examples	No. Failures		
		A	B	C
LP	46	1	0	0
HP	43	1	0	0
BP	50	3	2	5
BS	45	6	8	8

A: Exhaustive search

B: Selective search

C: Selective plus cubic search

Prescribed Specifications

- A nonrecursive filter of length N, passband and stopband weights of 1 and δ_p/δ_a, respectively, and specified passband and stopband edges can be readily designed.

Prescribed Specifications

- A nonrecursive filter of length N, passband and stopband weights of 1 and δ_p/δ_a, respectively, and specified passband and stopband edges can be readily designed.
- While the filter obtained will have passband and stopband edges at the correct locations and the ratio δ_p/δ_a will be exactly as required, the amplitudes of the passband and stopband ripples are highly unlikely to have the specified values.

Prescribed Specifications

- A nonrecursive filter of length N, passband and stopband weights of 1 and δ_p/δ_a, respectively, and specified passband and stopband edges can be readily designed.
- While the filter obtained will have passband and stopband edges at the correct locations and the ratio δ_p/δ_a will be exactly as required, the amplitudes of the passband and stopband ripples are highly unlikely to have the specified values.
- An acceptable design can be obtained by predicting the value of N on the basis of the required specifications and then designing filters for increasing or decreasing values of N until the lowest value of N that satisfies the specifications is found.

Filter Length Prediction

- A reasonably accurate empirical formula for the prediction of the required filter length, N, for the case of lowpass and highpass filters, due to Herrmann, Rabiner, and Chan, is

$$N = \text{int}\left[\frac{(D - FB^2)}{B} + 1.5\right]$$

where

$$\begin{aligned} B &= |\omega_a - \omega_p|/2\pi \\ D &= [0.005309(\log_{10}\delta_p)^2 + 0.07114\log_{10}\delta_p - 0.4761]\log_{10}\delta_a \\ &\quad -[0.00266(\log_{10}\delta_p)^2 + 0.5941\log_{10}\delta_p + 0.4278] \\ F &= 0.51244(\log_{10}\delta_p - \log_{10}\delta_a) + 11.012 \end{aligned}$$

Filter Length Prediction

- The formula of Herrmann et al. can also be used to predict the filter length in the design of bandpass, bandstop, and multiband filters in general.

Filter Length Prediction

- The formula of Herrmann et al. can also be used to predict the filter length in the design of bandpass, bandstop, and multiband filters in general.
- In these filters, a value of N is computed for each transition band between a passband and stopband or a stopband and passband and the largest value of N so obtained is taken to be the predicted filter length.

Algorithm

1. Compute N using the prediction formula of Herrmann et al.; if N is even, set $N = N + 1$.
2. Design a filter of length N using the Remez algorithm and determine the minimum value of δ, say $\breve{\delta}$.
   (A) If $\breve{\delta} > \delta_p$, then do:
      (a) Set $N = N + 2$, design a filter of length N using the Remez algorithm, and find $\breve{\delta}$;
      (b) If $\breve{\delta} \leq \delta_p$, then go to step 3; else, go to step $2(A)(a)$.
   (B) If $\breve{\delta} < \delta_p$, then do:
      (a) Set $N = N - 2$, design a filter of length N using the Remez algorithm, and find $\breve{\delta}$;
      (b) If $\breve{\delta} > \delta_p$, then go to step 4; else, go to step $2(B)(a)$.

Algorithm *Cont'd*

3. If part A of the algorithm was executed, use the last set of extremals and the corresponding value of N to obtain the impulse response of the required filter and stop.
4. If part B of the algorithm was executed, use the last but one set of extremals and the corresponding value of N to obtain the impulse response of the required filter and stop.

Example

In an application, a nonrecursive equiripple bandstop filter is required which should satisfy the following specifications:

- Odd filter length
- Passband ripple A_p: 0.5 dB
- Minimum stopband attenuation A_a: 50.0 dB
- Lower passband edge ω_{p1}: 0.8 rad/s
- Upper passband edge ω_{p2}: 2.2 rad/s
- Lower stopband edge ω_{a1}: 1.2 rad/s
- Upper stopband edge ω_{a2}: 1.8 rad/s
- Sampling frequency ω_s: 2π rad/s

Design the lowest-order filter that will satisfy the specifications.

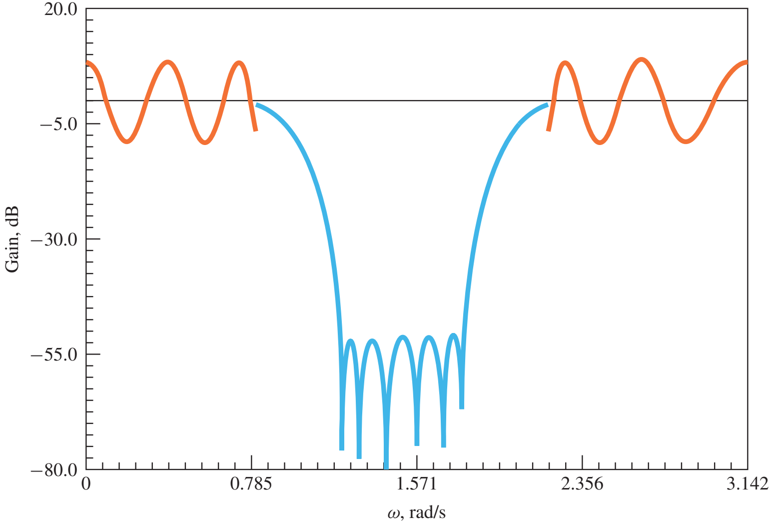
Example *Cont'd*

The design algorithm gave a filter with the following specifications:

- Passband ripple: 0.4342 dB
- Minimum stopband attenuation: 51.23 dB

Progress of Algorithm

N	Iters.	FE's	A_p, dB	A_a, dB
31	10	582	0.5055	49.91
33	7	376	0.5037	49.94
35	9	545	0.4342	51.23



Note: Passband errors multiplied by a factor of 40.

Advantages of Weighted-Chebyshev Method

- Designs are optimal, i.e., the required filter order for a set of prescribed specifications is the lowest that can be achieved.

Advantages of Weighted-Chebyshev Method

- Designs are optimal, i.e., the required filter order for a set of prescribed specifications is the lowest that can be achieved.
- The minimum filter order to satisfy certain prescribed specifications can be predicted by using certain empirical formulas.

Advantages of Weighted-Chebyshev Method

- Designs are optimal, i.e., the required filter order for a set of prescribed specifications is the lowest that can be achieved.
- The minimum filter order to satisfy certain prescribed specifications can be predicted by using certain empirical formulas.
- Minimum filter order implies a more efficient and faster filter implementation for real-time applications.

Advantages of Weighted-Chebyshev Method

- Designs are optimal, i.e., the required filter order for a set of prescribed specifications is the lowest that can be achieved.
- The minimum filter order to satisfy certain prescribed specifications can be predicted by using certain empirical formulas.
- Minimum filter order implies a more efficient and faster filter implementation for real-time applications.
- The method is very flexible in that it can be used to design filters, differentiators, Hilbert transformers, etc.

Advantages of Weighted-Chebyshev Method

- Designs are optimal, i.e., the required filter order for a set of prescribed specifications is the lowest that can be achieved.
- The minimum filter order to satisfy certain prescribed specifications can be predicted by using certain empirical formulas.
- Minimum filter order implies a more efficient and faster filter implementation for real-time applications.
- The method is very flexible in that it can be used to design filters, differentiators, Hilbert transformers, etc.
- The solutions achieved are equiripple.

Disadvantages of Weighted-Chebyshev Method

- The design requires a very large amount of computation.

Disadvantages of Weighted-Chebyshev Method

- The design requires a very large amount of computation.
- Not suitable for applications where the design has to be carried out in real- or quasi-real time, for example, in programmable or adaptable filters.

D-Filter

A DSP software package that incorporates the design techniques described in this presentation is *D-Filter*.

For more information about D-Filter or to download a *free* copy, click the following link:

http://ece.uvic.ca/∼dsp/Software-ne.html

Summary

- Three design techniques that bring about substantial improvements in the efficiency of the Remez algorithm have been described:
  - A step-by-step exhaustive search
  - A cubic interpolation search
  - An improved scheme for the rejection of superfluous potential extremals

Summary

- Three design techniques that bring about substantial improvements in the efficiency of the Remez algorithm have been described:
  - A step-by-step exhaustive search
  - A cubic interpolation search
  - An improved scheme for the rejection of superfluous potential extremals
- These techniques are implemented in a DSP software package known as D-Filter.

Summary

- Three design techniques that bring about substantial improvements in the efficiency of the Remez algorithm have been described:
  - A step-by-step exhaustive search
  - A cubic interpolation search
  - An improved scheme for the rejection of superfluous potential extremals
- These techniques are implemented in a DSP software package known as D-Filter.
- Extensive experimentation has shown that the selective and cubic interpolation searches reduce the amount of computation required by the Remez algorithm by almost 90% without degrading its robustness.

- The rejection scheme described increases the efficiency and robustness of the Remez algorithm further but the scheme has not been compared with the original method of McClellan, Rabiner, and Parks.

- The rejection scheme described increases the efficiency and robustness of the Remez algorithm further but the scheme has not been compared with the original method of McClellan, Rabiner, and Parks.
- By using a prediction technique for the required filter length proposed by Herrmann, Rabiner, and Chan, filters that satisfy prescribed specifications can be designed.

- The rejection scheme described increases the efficiency and robustness of the Remez algorithm further but the scheme has not been compared with the original method of McClellan, Rabiner, and Parks.
- By using a prediction technique for the required filter length proposed by Herrmann, Rabiner, and Chan, filters that satisfy prescribed specifications can be designed.
- For off-line applications, the Remez algorithm continues to be the method of choice for the design of linear-phase filters, multiband filters, differentiators, Hilbert transformers.

- Despite the improvements described, the Remez algorithm continues to require a large amount of computation.

  For applications that need the filter to be designed on-line in real or quasi-real time, *the window method is preferred* although the filters obtained are suboptimal.

This slide concludes the presentation.
Thank you for your attention.