A Unified Approach to the Design of Interpolated and Frequency-Response-Masking FIR Filters

Wu-Sheng Lu
University of Victoria
Victoria, Canada

Takao Hinamoto
Hiroshima University
Higashi-Hiroshima, Japan

May 2016
Outline

• Early Work
• Filter Structures
• Convex-Concave Procedure (CCP)
• Design of Interpolated FIR (IFIR) Filters
• Design of Frequency-Response Masking (FRM) FIR Filters
• Design Examples
1. Early Work

Interpolated FIR (IFIR) Filters

Frequency-Response Masking (FRM) FIR Filters
• Y. C. Lim, 1986.
• Many variants since 1990s.
2. Filter Structures

Interpolated FIR (IFIR) Filters

\[ H(z) = F(z^L)M(z) \]
Frequency-Response Masking (FRM) FIR Filters

\[
H(z) = F(z^L)M_a(z) + \left[ z^{-L(N-1)/2} - F(z^L) \right] M_c(z)
\]
3. Convex-Concave Procedure (CCP)

CCP refers to a heuristic method to solve a general class of nonconvex problems of the form

\[
\text{minimize } f(x) - g(x) \\
\text{subject to: } f_i(x) \leq g_i(x) \quad \text{for } i = 1, 2, \ldots, m
\]

where \( f(x), \ g(x), \ f_i(x), \) and \( g_i(x) \) for \( i = 1, 2, \ldots, m \) are convex. The basic CCP algorithm is an iterative procedure including two steps:

(i) Convexify the objective function and constraints by replacing \( g(x) \) and \( g_i(x) \), respectively, with their affine approximations

\[
\begin{align*}
\hat{g}(x, x_k) &= g(x_k) + \nabla g(x_k)^T (x - x_k) \\
\hat{g}_i(x, x_k) &= g_i(x_k) + \nabla g_i(x_k)^T (x - x_k) \quad \text{for } i = 1, 2, \ldots, m
\end{align*}
\]
(ii) Solve the convex problem

\[
\begin{align*}
\text{minimize} & \quad f(x) - \hat{g}(x, x_k) \\
\text{subject to:} & \quad f_i(x) \leq \hat{g}_i(x, x_k) \quad \text{for} \quad i = 1, 2, \ldots, m
\end{align*}
\]

- **Property 1**
  - ◊ If \( x_0 \) is feasible for the original problem, \( x_0 \) is also a feasible point for the convexified problem.
  - ◊ If \( x_{k+1} \) is produced by solving the convexified problem, then \( x_{k+1} \) is also feasible for the original problem.

- **Property 2**
  CCP is a descent algorithm, namely, \( \{f(x_k), k = 0,1,\ldots\} \) decreases monotonically.

- **Property 3**
  Iterates \( \{x_k, k = 0,1,\ldots\} \) converge to a critical point of the original problem.
4. Design of Interpolated FIR (IFIR) Filters

Frequency response of an IFIR filter:
\[ H(e^{j\omega}) = F(e^{jL\omega})M(e^{j\omega}) \]

Its zero-phase frequency response:
\[ H_0(x, \omega) = \begin{bmatrix} a_f^T t_f(L\omega) \\ a_m^T t_m(\omega) \end{bmatrix} \]

where \( a_f \) and \( a_m \) are coefficient vectors determined by the impulse responses of \( F(z) \) and \( M(z) \) respectively, and \( t_f(\omega) \) and \( t_m(\omega) \) are vectors with trigonometric components determined by the filter lengths and types.

Let \( H_d(\omega) \) be the desired zero-phase response of the IFIR filter, the frequency-weighted minimax design of an IFIR filter amounts to finding \( a_f \) and \( a_m \) that solve the nonconvex minimax problem

\[
\text{minimize} \quad \max_{\omega \in \Omega} w(\omega) \left| \begin{bmatrix} a_f^T t_f(L\omega) \\ a_m^T t_m(\omega) \end{bmatrix} - H_d(\omega) \right|
\]

where \( w(\omega) > 0 \) is a frequency-selective weight over \( \omega \in \Omega \).
Converting the problem to:

\[
\begin{align*}
\text{minimize} & \quad \delta \\
\text{subject to:} & \quad \begin{bmatrix} a_f^T t_f (L\omega) \\ a_m^T t_m (\omega) \end{bmatrix} \leq \delta_w + H_d (\omega) \\
& \quad -\begin{bmatrix} a_f^T t_f (L\omega) \\ a_m^T t_m (\omega) \end{bmatrix} \leq \delta_w - H_d (\omega)
\end{align*}
\]

Convexifying the problem by adding \( \frac{1}{2} p(x, \omega) \) with

\[
p(x, \omega) = \left[ a_f^T t_f (L\omega) \right]^2 + \left[ a_m^T t_m (\omega) \right]^2
\]

hence the constraints become

\[
\begin{align*}
\left[ a_f^T t_f (L\omega) + a_m^T t_m (\omega) \right]^2 & \leq p(x, \omega) + 2\delta_w + 2H_d (\omega) \\
\left[ a_f^T t_f (L\omega) - a_m^T t_m (\omega) \right]^2 & \leq p(x, \omega) + 2\delta_w - 2H_d (\omega)
\end{align*}
\]

which fit nicely into a CCP, hence the convexification is done by linearizing \( p(x, \omega) \) on the right-hand sides of the constraints.
Summarizing, the $k$-th iteration in the CCP solves the convex problem

\[
\begin{align*}
\text{minimize} & \quad \delta \\
\text{subject to:} & \quad \begin{bmatrix}
\eta_i(x, x_k, \omega) & t_i^T(\omega)x \\
t_i^T(\omega)x & 1
\end{bmatrix} \succeq 0 \quad \text{for } \omega \in \Omega_d, i = 0, 1
\end{align*}
\]

where

\[
\eta_i = p(x_k, \omega) + \nabla p(x_k, \omega)^T(x - x_k) + 2\delta_w + (-1)^i 2H_d(\omega),
\]

\[
t_i(\omega) = \begin{bmatrix}
t_f(L\omega) \\
(-1)^i t_m(\omega)
\end{bmatrix}
\]

\[
\Omega_d = \{\omega_j, j = 1, 2, \ldots, K\} \subseteq \Omega
\]

In words, the $k$-th iteration of the design algorithm solves an SDP problem involving a total of $2K$ 2-by-2 matrices that are required to be positive semidefinite.
5. Design of Frequency-Response Masking (FRM) FIR Filters

Frequency response of an FRM filter:

\[ H(e^{j\omega}) = F(e^{jL\omega})M_a(e^{j\omega}) + \left[ e^{-j(L(N-1)\omega/2} - F(e^{jL\omega}) \right]M_c(e^{j\omega}) \]

Its zero-phase frequency response:

\[ H(x, \omega) = \left[ a_f^T t_f(L\omega) \right] \left[ a_a^T t_a(\omega) - a_c^T t_c(\omega) \right] + a_c^T t_c(\omega) \]

where \( a_f, a_a, \) and \( a_c \) are coefficient vectors determined by the impulse responses of \( F(z), M_a(z) \) and \( M_c(z) \), respectively.

Let \( H_d(\omega) \) be the desired zero-phase response of the IFIR filter, the frequency-weighted minimax design of an IFIR filter amounts to finding \( a_f, a_a, \) and \( a_c \) that solve the nonconvex minimax problem

\[
\text{minimize} \quad \max_{\omega \in \Omega} \ w(\omega) \left| H(x, \omega) - H_d(\omega) \right|
\]

where \( w(\omega) > 0 \) is a frequency-selective weight over \( \omega \in \Omega \).
Converting the problem to:

\[
\begin{align*}
\text{minimize} & \quad \delta \\
\text{subject to:} & \quad H(x, \omega) - \delta_w - H_d(\omega) \leq 0, \quad \omega \in \Omega \\
& \quad -H(x, \omega) - \delta_w + H_d(\omega) \leq 0, \quad \omega \in \Omega
\end{align*}
\]

Convexifying the problem by adding the term

\[
p(x, \omega) = \left[ a^T f (L \omega) \right]^2 + \frac{1}{2} \left[ a^T a (\omega) \right]^2 + \frac{1}{2} \left[ a^T c (\omega) \right]^2
\]

hence the constraints become

\[
\begin{align*}
\nu(x, \omega) & \leq p(x, \omega) \\
\nu(x, \omega) & \leq p(x, \omega)
\end{align*}
\]

where

\[
\begin{align*}
u(x, \omega) & = p(x, \omega) + H(x, \omega) - \delta_w - H_d(\omega) \\
\nu(x, \omega) & = p(x, \omega) - H(x, \omega) - \delta_w + H_d(\omega)
\end{align*}
\]

are convex because
\[ p(x, \omega) + H(x, \omega) = \frac{1}{2} [a_f^T a_a^T] M_0 \begin{bmatrix} a_f \\ a_a \end{bmatrix} + \frac{1}{2} [a_f^T a_c^T] N_1 \begin{bmatrix} a_f \\ a_c \end{bmatrix} + a_c^T t_c \]

\[ p(x, \omega) - H(x, \omega) = \frac{1}{2} [a_f^T a_a^T] M_1 \begin{bmatrix} a_f \\ a_a \end{bmatrix} + \frac{1}{2} [a_f^T a_c^T] N_0 \begin{bmatrix} a_f \\ a_c \end{bmatrix} - a_c^T t_c \]

with positive semidefinite \( M_i = m_i m_i^T \) and \( N_i = n_i n_i^T \) where

\[
m_i = \begin{bmatrix} t_f(L\omega) \\ (-1)^i t_a(\omega) \end{bmatrix} \quad \text{and} \quad n_i = \begin{bmatrix} t_f(L\omega) \\ (-1)^i t_c(\omega) \end{bmatrix} \quad \text{for } i = 0, 1
\]

The above form of constraints fits nicely into a CCP, hence convexification can be done by linearizing \( p(x, \omega) \) on the right-hand sides of the constraints, i.e.,

\[ u(x, \omega) \leq \tilde{p}(x, x_k, \omega) \quad \text{for } \omega \in \Omega \]

\[ v(x, \omega) \leq \tilde{p}(x, x_k, \omega) \quad \text{for } \omega \in \Omega \]

where \( \tilde{p}(x, x_k, \omega) = p(x_k, \omega) + \nabla p(x_k, \omega)^T (x - x_k) \) with

13
\[ \nabla p(x_k, \omega) = \begin{bmatrix} 2[a_f^T t_f(L\omega)] t_f(L\omega) \\ [a_a^T t_a(\omega)] t_a(\omega) \\ [a_c^T t_c(\omega)] t_c(\omega) \end{bmatrix} \]

\[ \diamond \text{Summarizing, the } k\text{-th iteration in the CCP solves the convex problem} \]

\[ \text{minimize } \delta \]

\[ \text{subject to: } u(x, \omega_j) \leq \tilde{p}(x, x_k, \omega_j) \text{ for } j = 1, \ldots, K \]

\[ v(x, \omega_j) \leq \tilde{p}(x, x_k, \omega_j) \text{ for } j = 1, \ldots, K \]

where \( \{\omega_j, j = 1, 2, \ldots, K\} \subset \Omega. \)

In words, the \( k\)-th iteration of the design algorithm solves an SOCP problem minimizing a linear objective subject to a total of \( 2K \) quadratic constraints.
6. Design Examples

Example 1 The first algorithm was applied to design a lowpass IFIR filter with normalized passband edge $\omega_p = 0.15\pi$, stopband edge $\omega_a = 0.2\pi$. The sparsity factor was set to $L = 4$, and orders of $F(z)$ and $M(z)$ are 31 and 17, respectively. The frequency weight $w(\omega)$ was set to $w(\omega) \equiv 1$ for $\omega$ in the passband and $w(\omega) \equiv 2$ for $\omega$ in the stopband. An initial was generated by the standard technique proposed in [3]. A total of $K = 1400$ frequency grids were uniformly placed in $[0, \omega_p] \cup [\omega_a, \pi]$ to form the discrete set $\Omega_d$ for problem. It took the algorithm 91 iterations to converge to an IFIR filter with $A_p = 0.03171$ dB and $A_a = 60.84$ dB.

The same design problem was addressed as Example 10.29 in [4] using the method described in [Saramäki, 1993]. The method was implemented as function ifir in Signal Processing Toolbox of MATLAB. With

$$[F,M] = \text{ifir}(4,\text{'low'}, [0.15, 0.2], [0.002, 0.001], \text{'advanced'})$$

the function returns with optimized impulses of filter $F(z)$ of order 31 and $M(z)$ of order 17 (in Example 10.29 of [4], the order of $M(z)$ was said to be 16, however the order of $M(z)$ produced by the above MATLAB code was actually 17), with $A_p = 0.0340$ dB and $A_a = 60.18$ dB.
Passband Ripple (0.03171 dB vs 0.0340 dB)

(a) Normalized frequency

Proposed Method of [27], [4]
Stopband Attenuation (60.84 dB vs 60.18 dB)

(b) Normalized frequency

Proposed
Method of [27], [4]
Example 2  The second algorithm was applied to design a lowpass FRM filter with the same design specifications as in the first example in [3] and [10]. The normalized passband and stopband edges were $\omega_p = 0.6\pi$ and $\omega_a = 0.61\pi$. The sparsity factor was set to $L = 9$, and orders of $F(z)$, $M_a(z)$, and $M_c(z)$ were 44, 40, and 32, respectively. A trivial weight $w(\omega) \equiv 1$ was utilized. With $K = 1100$, it took the algorithm 60 iterations to converge to an FRM filter with $A_p = 0.1321$ dB and $A_a = 42.44$ dB, which are favorably compared with those achieved in [3] ($A_p = 0.1792$ dB and $A_a = 40.96$ dB), which has been a benchmark for FRM filters, and those reported in [10] ($A_p = 0.1348$ dB and $A_a = 42.25$ dB).
Amplitude Response in dB

The graph shows a plot of amplitude response in dB. The x-axis represents a range from 0 to 1, while the y-axis ranges from -70 to 0 dB. The plot features a sharp drop in amplitude around the 0.6 mark, followed by a series of oscillations with decreasing amplitude towards the right.
Thank you.

Q & A