Linear Programming Algorithms for Sparse Filter Design

Ludwig Mohr, Peter Reinprecht

SPSC Lab - Advanced Signal Processing 1, Seminar

January 16, 2012

Outline

Outline

Introduction

Efficient implementation of discrete-time filters

- Complexity reduction of filters
  - reduction of number of arithmetic operations
  - restriction of arithmetic operations to those with fast execution on dsp
  - our approach: sparse filters

Sparse filters

- filter with few nonzero impulse response coefficients
- pro: arithmetic operations for zero-valued coefficients can be omitted altogether

Sparse Filter Design

Direct design

- complex and difficult task
- time and ressource consuming

Complexity reduction of given filter

- start with filter with desired response
- zero-forcing of some coefficients
- optimize remaining nonzero coefficients to achieve minimum deviation of the resulting response and the desired response
- feasible result: sparse filter with a deviation below certain threshold

Sparse Filter Design

Problems and Challenges

- larger number of possible combinations of zero-valued and nonzero coefficients
- still computationally difficult task
- in theory guaranteed to yield a maximal sparse filter (alls sparsity problems are considered)

Solution: use approximations

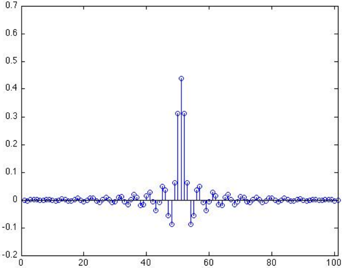
- goal: reasonably sparse filters
- resulting filter is not necessarily optimal
- idea: just consider a certain subset of possible combinations

Outline

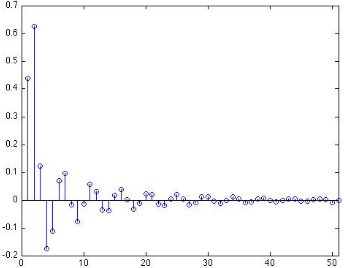
Problem Formulation

Given Filter: Type I linear-phase FIR

$$b_0 = h[m]$$
$$b_m = 2h[M-n] = 2h[M+n], \qquad n = 1, 2, ..., M$$



h



b

Problem Formulation

Frequency response

$$H(e^{j\omega}) = A(e^{j\omega})e^{-j\omega M}$$

$$A(e^{j\omega}) = \sum_{n=0}^{M} b_n \cos(n\omega)$$

$$W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \leq \delta_d \qquad \forall\omega$$

Problem Formulation

optimization problem

$$\min_{b_0,\dots,b_M} \quad \|\mathbf{b}\|_0$$
$$\text{s.t.} \quad W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \leq \delta_d \qquad \forall\omega$$

Successive Thinning Algorithm

$$\min_{\{b_n\},\delta} \quad \delta$$
$$\text{s.t.} \quad W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \leq \delta \qquad \forall\omega$$
$$b_n = 0 \qquad \forall n \in \mathcal{Z}^{(i)}$$

Successive Thinning Algorithms

discrete Frequencies

$$\begin{aligned} \min_{\{b_n\},\delta} \quad & \delta \\ \text{s.t.} \quad & W(\omega_k)\left|\sum_{n=0}^{M} b_n \cos(n\omega_k) - H_d(e^{j\omega_k})\right| \leq \delta, \quad k = 1, ..., K \\ & b_n = 0 \qquad \forall n \in \mathcal{Z}^{(i)} \end{aligned}$$

general linear program in standard form

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{aligned}$$

Successive Thinning Algorithms

$$\mathbf{A}_{kn} = W(\omega_k)\cos(n\omega_k), \quad k = 1, ..., K, n = 0, ..., M$$
$$\mathbf{h}_k = W(\omega_k)H_d(e^{j\omega_k}), \quad k = 1, ..., K$$
$$\mathbf{e}_k = 1, \quad k = 1, ..., K$$

resulting linear program

$$\begin{aligned} \min_{\delta,\mathbf{b}} \quad & \delta \\ \text{s.t.} \quad & \begin{bmatrix} -\mathbf{e} & \mathbf{A}_{kn} \end{bmatrix} \begin{bmatrix} \delta \\ \mathbf{b} \end{bmatrix} \leq \mathbf{h} \\ & \begin{bmatrix} -\mathbf{e} & -\mathbf{A}_{kn} \end{bmatrix} \begin{bmatrix} \delta \\ \mathbf{b} \end{bmatrix} \leq -\mathbf{h} \\ & b_n = 0 \qquad \forall n \in \mathcal{Z}^{(i)} \end{aligned}$$

Successive Thinning Algorithms

corresponding dual problem

$$
\begin{aligned}
\max_{\{p_k^+\},\{p_k^-\}} \quad & \sum_{k=1}^{K} W(\omega_k) H_d(e^{j\omega_k})(p_k^+ - p_k^-) \\
\text{s.t.} \quad & \sum_{k=1}^{K} W(\omega_k) \cos(n\omega_k)(p_k^+ - p_k^-) \qquad \forall n \in \mathcal{N}^{(i)} \\
& \sum_{k=1}^{K} (p_k^+ - p_k^-) = 1 \\
& p_k^+ \geq 0, p_k^- \geq 0, \qquad k = 1, ..., K
\end{aligned}
$$

Selection Rules for selecting next Zero coefficient

Two possible heuristics

- Minimum-Increase Rule

$$m^{(i)} = \arg \min_{p \in \mathcal{C}^{(i)}} \delta^{(i)}(p)$$

- Smallest-Coefficient Rule

$$m^{(i)} = \arg \min_{n \in \mathcal{N}^{(i)}} \left| b_n^{(i)} \right|$$

Efficiency Enhancements

$$\tilde{\mathcal{Z}} = \mathcal{Z} \cup \{m\}$$
$$\mathcal{N} = \tilde{\mathcal{N}} \cup \{m\}$$

exploid sparsity of **b**

$$\begin{aligned} \min_{\delta,\mathbf{b}} \quad & \delta \\ \text{s.t.} \quad & \begin{bmatrix} -\mathbf{e} & \mathbf{A}_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \delta \\ \mathbf{b}_{\mathcal{N}} \end{bmatrix} \leq \mathbf{h} \\ & \begin{bmatrix} -\mathbf{e} & -\mathbf{A}_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \delta \\ \mathbf{b}_{\mathcal{N}} \end{bmatrix} \leq -\mathbf{h} \end{aligned}$$

Outline

Minimum 1-Norm Design

Developement of an algorithm

- smallest coefficient method: in each loop iteration smallest nonzero coefficient set to zero, subsequent optimization
- similar strategy for filters with many small coefficients
- two stage implementation
- first stage: optimize filter for large number of small coefficients
- define a desired number of $\mathcal{J}$ coefficients allowed to be nonzero
- stage two: set $M + 1 - \mathcal{J}$ smallest coefficients to zero, re-optimize other coefficients

Stage 1

Optimization objective

- minimize 1-norm of impulse response vector
- thereby, promote designs with many small-valued coefficients
- idea: coefficients with small values in this step will be identical with those still yielding a feasible result when set to zero

Stage 1

definition of 1-norm

$$\|\mathbf{b}\|_1 = \sum_{n=0}^{M} |b_n|$$

resulting opimization problem

$$\begin{aligned} \min_{\{b_n\}} \quad & \|\mathbf{b}\|_1 \\ \text{s.t.} \quad & W(\omega)|A(e^{j\omega}) - H_d(e^{j\omega})| \geq \delta_d \quad \forall \omega \in \mathcal{S} \\ & b_n = 0 \quad \forall n \in \mathcal{Z} \end{aligned}$$

Stage 1

resulting linear program

$$\begin{aligned} \min_{\mathbf{m},\mathbf{b}} \quad & [1, 1, \cdots, 1]\mathbf{m} \\ \text{s.t.} \quad & \mathbf{A}_{\mathcal{N}}\mathbf{b}_{\mathcal{N}} \geq \mathbf{h} - \delta_d \mathbf{e} \\ & -\mathbf{A}_{\mathcal{N}}\mathbf{b}_{\mathcal{N}} \geq -\mathbf{h} - \delta_d \mathbf{e} \\ & \mathbf{m} + \mathbf{b}_{\mathcal{N}} \geq 0 \\ & \mathbf{m} - \mathbf{b}_{\mathcal{N}} \geq 0 \\ & \mathbf{m} \geq 0 \end{aligned}$$

m is strictly positive vector constrained to bound magnitude of coefficients b_n

decreasing $\sum m$ should decrease coefficients b_n

Stage 2

set a certain number of $M + 1 - \mathcal{J}$ coefficients to zero
only $\mathcal{J}$ coefficients of the impulse response have nonzero values

$$\begin{aligned} \min_{\{b_n\},\delta} \quad & \delta \\ \text{s.t.} \quad & W(\omega_k)|A(e^{j\omega_k}) - H_d(e^{j\omega_k})| \le \delta \\ & \qquad k = 1, \cdots, K \\ & b_n = 0 \quad \forall n \notin \mathcal{N}_{\mathcal{J}} \end{aligned}$$

$\mathcal{N}_{\mathcal{J}}$ denotes index set of $\mathcal{J}$ nonzero coefficients in b_n

Stage 2

Perform binary search to determine smallest $\mathcal{J}$ still yielding a feasible result

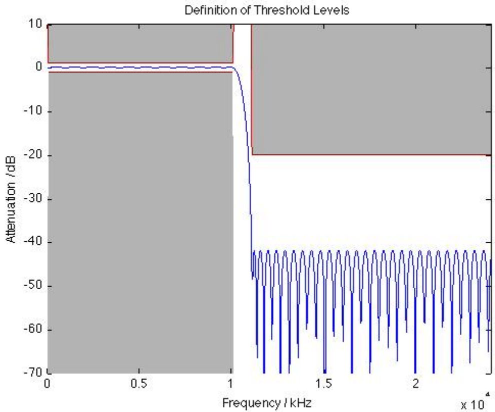
- $\mathcal{J}_0$ feasible: all $\mathcal{J} > \mathcal{J}_0$ are feasible
- $\mathcal{J}_0$ not feasible: all $\mathcal{J} < \mathcal{J}_0$ are not feasible

Outline

Definition of Thresholds

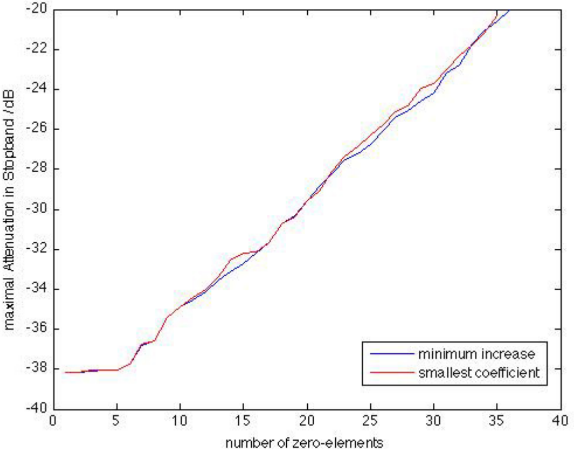
Ex: LP Filter

Passband:	$\pm$ 1 dB
Stopband attenutaion:	$<$ 20 dB



Error in Stopband over number of Zero-Elements

LP Filter, $N = 100$



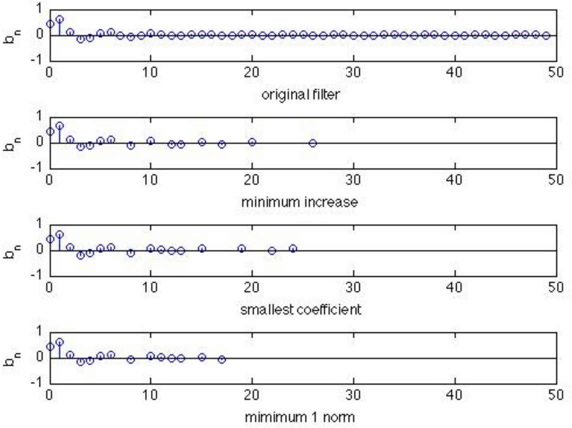
Impulse response

LP Filter, $N = 100$

stopband threshold level [dB]	algorithm	non-zero weights
-30	min. increase	32
	smalllest coeff.	32
	min. 1-norm	30
-25	min. increase	23
	smalllest coeff.	24
	min. 1-norm	24
-20	min. increase	15
	smalllest coeff.	16
	min. 1-norm	14

Nonzero elements of Impulse response

TP Filter, $N = 100$, threshold level = -20 dB



Outline

Conclusion

Linear Program Algorithms for sparse filter design results in

- decrease of computational complexity
- tradeoff between complexity and filter accuracy
- may not yield optimal solution

Thank you for your attention!

References

Thomas Baran, Dennis Wei and Alan V. Oppenheim,
"Linear Programming Algorithms for Sparse Filter Design,"
IEEE Volume 58 Issue 3, March 2010, pp. 1605–1617, Mar 2010

Juncheng Wei,
"Lecture notes on Linear Programming, 2011,"
http://www.math.cuhk.edu.hk/ wei/LP11.html

Stephen Boyd and Lieven Vandenberghe,
"Convex Optimization,"
Cambridge University Press, 2004