Application of Quadratic Programming to FIR Digital Filter Design Problems

John W. Adams, James L. Sullivan, David R. Gleeson, P.H. Chang, and R. Hashemi

Electrical and Computer Engineering Department, Mail Code 8346
California State University, Northridge, CA 91330-8346

Abstract

Quadratic programming problems have long been of interest in the business community. Hundreds of papers have been published that deal with applying quadratic programming algorithms to business problems. Quadratic programming is often used as the basis for "program trading" where stocks are automatically bought and sold by mutual funds to optimize profits. Quadratic programming algorithms can also be used to optimize digital filters, as discussed in this paper.

1. Introduction

In filter design applications quadratic programming problems arise when we wish to minimize the total weighted-squared error in the zero-phase response, ε,

$$\varepsilon = \int_0^{0.5} \Psi(e^{j2\pi f}) \left| H_0(e^{j2\pi f}) - H_d(e^{j2\pi f}) \right|^2 df$$

subject to inequality constraints on the zero-phase response,

$$\delta_L(e^{j2\pi f}) \leq H_0(e^{j2\pi f}) \leq \delta_U(e^{j2\pi f})$$

and inequality constraints on the derivative of the zero-phase response,

$$D_L(e^{j2\pi f}) \leq \frac{d H_0(e^{j2\pi f})}{df} \leq D_U(e^{j2\pi f}),$$

and inequality constraints on the curvature of the zero-phase response,

$$\beta_L(e^{j2\pi f}) \leq \frac{d^2 H_0(e^{j2\pi f})}{d^2 f} \leq \beta_U(e^{j2\pi f})$$

and equality constraints on the filter coefficients,

$$Cv = c$$

and inequality constraints on the filter coefficients,

$$Uv \geq u.$$

$H_0(e^{j2\pi f})$ denotes the actual zero-phase response; $H_d(e^{j2\pi f})$ denotes the desired zero-phase response; $\Psi(e^{j2\pi f})$ denotes the squared-error weighting; $\delta_U(e^{j2\pi f})$ denotes the upper limit on the zero phase response; $\delta_L(e^{j2\pi f})$ denotes the lower limit on the zero phase response; $D_L(e^{j2\pi f})$ denotes the lower limit on the zero-phase response derivative; and $D_U(e^{j2\pi f})$ denotes the upper limit on the zero-phase response derivative; $\beta_L(e^{j2\pi f})$ denotes the lower limit on the zero-phase response curvature; and $\beta_U(e^{j2\pi f})$ denotes the upper limit on the zero-phase response curvature.

The slope and curvature bounds can be used to prevent transition band anomalies, such as transition band ripples which occasionally occur in multiband filters. In particular, the slope constraints can be used to ensure that the gain is monotonic in the transition bands. The upper and lower limits on the zero phase response can be used to ensure that gains in the transition bands do not exceed the maximum passband gain.

We refer to the above design problem as a constrained least-squares (CLS) filter design problem. We can extend the CLS concept to do constrained total least-squares

This work was supported in part by the National Science Foundation under Grant MIPS-9200581.

1058-6393/95 $4.00 © 1995 IEEE

(CTLS) optimization. We plan to discuss CTLS optimization of digital filters in a future paper. In this paper we focus on solving CLS problems.

2. Generalized Multiple Exchange

The generalized multiple exchange (GME) algorithm is guaranteed to converge to the unique optimal solution for any feasible positive-definite quadratic programming problem. Of course, in practical filter design problems the objective function is always positive definite.

The generalized multiple exchange (GME) algorithm can do either multiple exchanges or single exchanges. It is a generalization of the algorithms discussed in [1] and [2]. Although it can execute either type of exchange, the GME algorithm emphasizes multiple exchanges because they converge faster than single exchanges, as discussed in [1]-[2]. The GME algorithm avoids single exchanges unless they are determined to be necessary. Fortunately, the GME algorithm is able to execute multiple exchanges for most of the iterations in practical filter design problems.

The GME algorithm has separate procedures for implementing single and multiple exchanges. The single exchange procedure (SEP) maintains dual feasibility in all iterations. In each major iteration it increases the primal objective function and it adds one new active constraint. We now define some terminology and acronyms that are used throughout this paper.

Smooth Inequality Constraints: The "smooth inequality constraints" correspond to the error bounds that vary smoothly inside of each frequency band. For example, constant error bounds that are not at band edge frequencies are smooth inquality constraints.

Non-Smooth Inequality Constraints: The "non-smooth constraints" correspond to constraints that are not smooth functions of frequency. In particular, we have non-smooth inequality constraints at band edge frequencies. Also, inequality constraints at isolated frequencies are non-smooth inequality constraints.

Multiple Exchange Procedure: The multiple exchange procedure (MEP) is a generalized version of the multiple exchange algorithm in [1]-[2]. It is generalized to handle equality and inequality constraints in both the frequency-domain and the sample-domain. At the start of the k-th iteration we have an active set of inequality constraints, $S_A(k)$. Violations of the smooth inequality constraints are searched for extremal frequencies and the corresponding constraints are denoted S_{VS}. (Violations are determined within a small numerical tolerance, Δ, to compensate for machine rounding errors.) The violated non-smooth constraints are denoted S_{VN}. A temporary set of active constraints, S_T, is determined according to $S_T \leftarrow S_{VS} \cup S_{VN}$. For computational purposes we also define $S_C = S_A(k) \cap S_T$, $S_{DA} = S_A(k) - S_C$ and $S_N = S_T - S_C$. The objective function is then minimized subject to the constraints in $S_T \cup S_E$, where S_E denotes the equality constraints. If any KT multiplier is negative or if any constraint in S_{DA} is violated then we branch out of the MEP. Otherwise, we update the major iteration index, $k \leftarrow k + 1$, update the error energy $\varepsilon(k) \leftarrow \varepsilon_T$, update the active constraint set, $S_A(k) \leftarrow S_T$, and test for optimality using the Kuhn-Tucker conditions. If the solution to the filter design problem is not optimal we do another iteration of the MEP.

Drop Procedure: The purpose of the drop procedure (DP) is to drop constraints with negative Kuhn-Tucker multipliers. NKT denotes the maximum number of constraints that the DP will drop. $NA(k)$ denotes the number of active constraints at the end of Iteration k. N_T denotes the number of active constraints in S_T. We have found that a good choice for NKT is given by $N_T - NA(k) + 1$ when $N_T - NA(k) + 1 > 0$.

Single Exchange Procedure: The single exchange procedure (SEP) performs a single exchange based on a modified version of the Goldfarb-Idnani (GI) algorithm in [3] to solve the designated subproblem. For improved numerical efficiency our implementation exploits the special structures of the vectors and matrices in filter and window design problems, instead of using the implementation in [3]. The SEP starts with the active set, $S_A(k)$, and primal and dual space solutions denoted $\nu(k)$ and $\mu(k)$ from the previous successful iteration where $\mu(k) \geq 0$. The most violated constraint is denoted S_{MV}. The SEP takes a full step to solve the subproblem defined by $S_T = S_A(k) \cup S_{MV}$ and obtains primal and dual space solutions denoted $\nu_f(k)$ and $\mu_f(k)$ with error energy ε_T. If $\mu_f(k) > 0$ we increment the major iteration index, $k \leftarrow k + 1$, update the active constraint set, $S_A(k) \leftarrow S_T$, update the error energy, $\varepsilon(k) \leftarrow \varepsilon_T$, and then test for optimality. If the solution is optimal we terminate. If any KT multiplier is negative the SEP finds $\nu_f(k) - \nu(k)$ and $\mu_f(k) - \mu(k)$ and moves the maximum distances in the corresponding directions while maintaining $\mu \geq 0$. The constraint with zero KT multiplier is dropped from S_T. We update the major iteration index, $k \leftarrow k + 1$, update the active constraint set, $S_A(k) \leftarrow S_T$, update the error energy $\varepsilon(k) \leftarrow \varepsilon_T$, and test for optimality. If the solution is optimal we terminate.

A. Simplified GME Algorithm

We start with the initial active set of constraints, $S_A(0) = \varnothing$. The initial guess minimizes the objective function subject to the equality constraints. If the initial solution satisfies the Kuhn-Tucker optimality conditions, we terminate. Otherwise, the MEP implements multiple exchanges. We continue to iterate with the MEP until optimality is detected provided that the objective function has increased. We also check that the Kuhn-Tucker (KT) multipliers are non-negative. Each succesful iteration of the MEP produces a new dual-feasible active set, $S_A(k+1)$. Otherwise, we branch to the SEP. After the SEP has completed we test for optimality. If the solution is not optimal we switch back to the MEP and continue to use it until optimality is detected, or until branching to the SEP for the previously discussed reasons.

We note that the initial solution is trivial if there are no equality constraints and if all of the squared-error weightings corresponding to nonzero $H_d(e^{j2\pi f})$ are zero. In this case we modify the initialization of the algorithm to activate one of the inequality constraints. This is usually done by selecting the band edge with the largest value of $H_d(e^{j2\pi f})$ and activating the inequality constraint specifying the lower limit on H_d.

The SGME algorithm always converges in a finite number of iterations. The number of iterations is finite because each iteration terminates with a different set of active constraints and because the total number of constraint combinations is finite. Each iteration terminates with a different set of active constraints because the objective function increases from one iteration to the next. Otherwise the problem is determined to be nonfeasible.

We note that total number of constraint combinations is finite when the inequality constraints are specified on a discrete frequency grid.

B. GME Algorithm

Step (0), Initialization: Solve the EQP subject to the set of equality constraints, S_E. Set the major iteration index, k, to zero. Compute error energy, $\varepsilon(k)$. Set the active constraint set, S_A, to $\{\varnothing\}$. (If there are no nonzero equality constraints, then select one passband edge and constrain the zero-phase response to the corresponding minimum passband gain specification. This allows us to get a non-trivial solution when the passband squared-error weightings are zero.) Test for optimality. Terminate if the solution is optimal. Otherwise, go to Step (1).

Step (1): MEP: Determine a temporary set of active constraints, denoted S_T. Include all non-smooth inequality constraints that are violated. Also include extremal frequencies that violate smooth inequality constraints. Solve the EQP subject to the equality constraints, S_E, and the temporary active constraints, S_T. Terminate if the solution is optimal. If it is not optimal, compute the temporary error energy, ε_T, and the KT multipliers. If any KT multiplier is negative, then go to Step (2). Otherwise, if $\varepsilon_T \le \varepsilon(k)$, then go to Step (3). If all KT multipliers are non-negative and if $\varepsilon_T > \varepsilon(k)$, then we increment the major iteration index, $k \leftarrow k + 1$, update the active constraint set, $S_A(k) \leftarrow S_T$, update the error energy, $\varepsilon(k) \leftarrow \varepsilon_T$, and go to Step (1).

Step (2), DP: If $NKT = 0$ the DP is not used and the GME algorithm reduces to the SGME algorithm. If $NKT \neq 0$ the DP drops the constraint with the most negative Kuhn-Tucker multiplier and sets $IKT = 1$. It then updates the KT multipliers and the temporary error energy, ε_T. If $\varepsilon_T \le \varepsilon(k)$, then we go to Step (3). If $\varepsilon_T > \varepsilon(k)$ and any KT multiplier is negative and $IKT < NKT$ then we set $IKT = IKT+1$ and drop the constraint with the most negative KT multiplier and cycle through the DP again if necessary. If $\varepsilon_T > \varepsilon(k)$ and any KT multiplier is negative and $IKT = NKT$ then we go to Step (3). Else if $\varepsilon_T > \varepsilon(k)$ and KT multipliers are non-negative we go to Step (1).

Step (3), SEP: The SEP is discussed above. At the end of the SEP we test for optimality. If the solution is optimal we terminate. Otherwise, we go to Step (1).

The GME algorithm must converge to the optimal solution within a finite number of iterations. The number of iterations is finite because each iteration terminates with a different set of active constraints, and because the number of different active constraint sets is finite. Each iteration terminates with a different set of active constraints because the objective function increases monotonically.

In order to illustrate the generality of the GME algorithm, we consider an example of an FIR linear-phase multiband filter with $L = 64$. The objective is to minimize the weighted-squared error, ε, in the stopband frequency interval, [0.35, 0.5], where

$$\varepsilon = \int_{0.35}^{0.5} (1000 - 100f)\left|H(e^{j2\pi f})\right|^2 df$$

subject to the maximum gain constraints in three stopbands,

$$\left|H(e^{j2\pi f})\right| \le \begin{cases} 0.125f, & \text{for } 0.08 \le f \le 0.16 \\ 0.01, & \text{for } 0.16 < f \le 0.22 \\ 0.03, & \text{for } 0.35 \le f \le 0.50 \end{cases}$$

and the inequality constraints in two passbands,

$$0.9 \le H_0(e^{j2\pi f}) \le 1.0;\ 0 \le f \le 0.05 \text{ and } 0.25 \le f \le 0.33$$

and a stopband null constraint: $H_0(e^{j2\pi f}) = 0.0$ at $f = 0.414$. The resulting optimal filter is shown in Fig. 1.

The GME algorithm can be used for designing multi-dimensional digital filters and windows with peak-constrained least-squared (PCLS) errors. Fig. 2 shows the frequency response of a two-dimensional PCLS window with a length of 21 samples in each dimension. The gain was constrained to be unity (0 dB) at DC and it was constrained to be less than or equal to –34.0 dB at frequencies greater than or equal to F_s in either dimension, with F_s = 0.075 cycles/sample. The energy at frequencies exceeding F_s in either dimension was minimized. The gain turned out to be exactly –34.0 dB at F_s in both dimensions and at the first several sidelobe peaks.

3. New Maximum Directivity Window

In [2] we introduced a new window with maximum directivity. We defined a window's directivity as the ratio of its peak mainlobe power (DC power) to the total energy in its Fourier transform. We denoted this ratio D, as follows:

$$D = \frac{\left|W(e^{j0})\right|^2}{\int_{-1/2}^{1/2} \left|W(e^{j2\pi f})\right|^2 df}$$

We maximized the directivity by specifying an equality constraint for a non-zero DC gain, and by minimizing the total energy in the frequency response on the interval [-0.5, 0.5] cycles/sample. We also specified a sidelobe boundary frequency, F_s, and a peak-sidelobe limit, δ_s, in [2].

In this paper we propose a new type of window with maximum directivity. The F_s specification is omitted but we still impose a peak-sidelobe limit denoted δ_s. In the new window with maximum directivity we only activate constraints at local error extrema. We call it the "ripple bounded maximum directivity window" because we bound the ripples and we do not specify the gain at F_s.

It is easy to modify the GME algorithm to design the new maximum directivity window. We ignore the stopband edge frequency, F_s, and we only permit the GME algorithm to activate constraints at extremal frequencies. An extremal frequency is defined as a frequency where there is a local maximum or minimum of the error function. On a continuous frequency scale we have a zero derivative at each extremal frequency. On a discrete frequency scale we have a value which is greater than (or less than) the values at the adjacent frequency points.

Fig. 3 shows an example of the ripple bounded maximum directivity (RBMD) window. It has a length of 101 samples and it was designed to have a maximum sidelobe amplitude 30.0 dB below the mainlobe peak.

4. Digital Filters with Minimized Energies Outside of their Passbands

We can extend the ripple bounded concept to filter stopbands by ignoring the errors at stopband edge frequencies. We call these filters "ripple bounded stopband (RBS) filters." The GME algorithm can be modified to design RBS filters by minimizing the weighted error energies outside of the passbands, and by imposing active constraints in stopbands at sidelobe extremal frequencies, but not at stopband edge frequencies.

We now consider an example of a lowpass RBS filter. The impulse response length is 95 and the passband edge frequency is 0.0625 cycles/sample. The maximum passband gain is specified to be 0 dB and the minimum passband gain is specified to be –1.0 dB. The maximum stopband sidelobe amplitude is specified to be –35.0 dB. The objective is to minimize the energy outside the passband subject to the previously stated specifications. Active constraints are only imposed in the stopband at sidelobe frequencies. In this example ε is defined by

$$\varepsilon = \int_{0.0625}^{0.5} \left|H(e^{j2\pi f})\right|^2 df$$

Fig. 4 shows the corresponding frequency response obtained by the modified GME algorithm. It has an effective stopband edge frequency of 0.0767 cycles/sample. This is the frequency where the gain is first equal to –35.0 dB. RBS filter designs are useful for problems where L, F_p, DB_p, and DB_s, are specified, and F_s is not specified.

5. Conclusion

Many publications deal with applying quadratic programming algorithms to business problems. However, quadratic programming algorithms can also be used to optimize digital filters, as discussed in this paper.

We presented the GME, SGME and modified GME algorithms for designing constrained least-squares (CLS) filters. CLS filters are generalizations of the popular minimax and least-squares filters. CLS filters are important not only because of their generality, but also because they are needed for many practical applications. We plan to present more details in a journal paper.

6. References

[1] J.W. Adams, "FIR digital filters with least-squares stopbands subject to peak-gain constraints," *IEEE Transactions on Circuits and Systems*, vol. CAS-39, pp. 376-388, April 1991.

[2] J.W. Adams, "A new optimal window," *IEEE Transactions on Signal Processing*, vol. 38, pp. 1753-1769, August 1991.

[3] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, pp. 1-33, 1983.

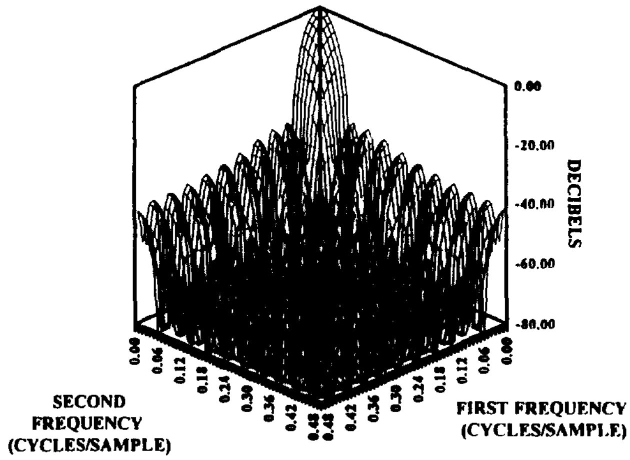

Fig. 2. Frequency response for the 2D example.

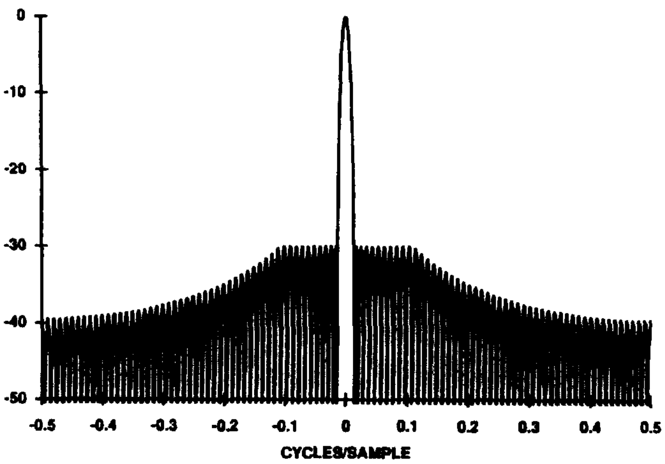

Fig. 3. Frequency response for the RBMD window.

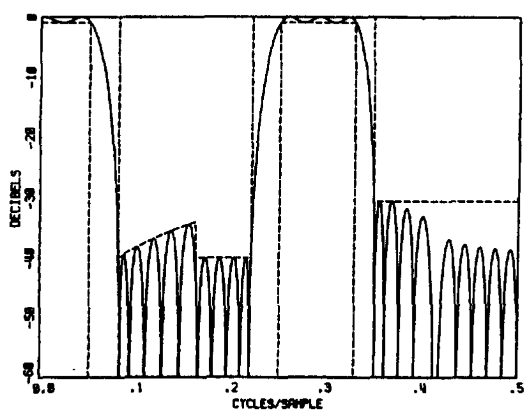

Fig. 1. Frequency response for the multiband example.

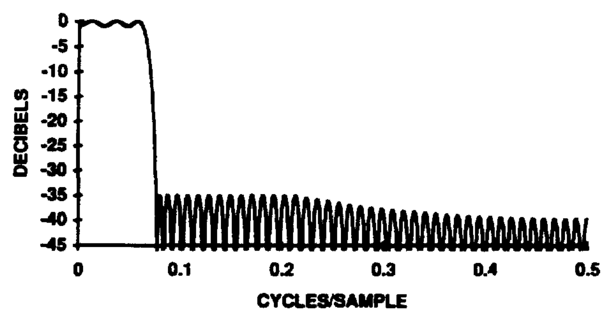

Fig. 4. Frequency response for the RBS filter.