A MULTIPLE EXCHANGE ALGORITHM FOR CONSTRAINED DESIGN OF FIR FILTERS IN THE COMPLEX DOMAIN

Mathias C. Lang

Institut für Nachrichtentechnik und Hochfrequenztechnik
Vienna University of Technology
Gusshausstrasse 25/389, A-1040 Vienna, AUSTRIA
mathias.lang@tuwien.ac.at

ABSTRACT

We present a fast multiple exchange algorithm that designs FIR filters with magnitude and phase specifications subject to constraints on the error function. We use a constrained least squares criterion which minimizes error energy and imposes bounds on the magnitude of the error. We can trade error energy versus peak error, and complex least squares and complex Chebyshev filters result as special cases. We provide a Matlab program implementing the proposed algorithm. This program has proved to be efficient and reliable.

1. INTRODUCTION

Conventionally, the frequency domain design of filters is accomplished by minimizing an appropriate measure of the approximation error. Error measures that are frequently used are error energy and maximum errors. Filters minimizing these error measures are optimum in the least squares and in the Chebyshev sense, respectively. However, it has turned out that unconstrained minimization of some error measure is often not the most appropriate formulation of a filter design problem. It is often desired to impose bounds on certain error functions, or to mix different error criteria. A standard design specification is a tolerance scheme for some frequency domain function. A tolerance scheme is the actual motivation for all algorithms based on Chebyshev approximation. However, unconstrained Chebyshev approximation does not solve this problem, because the maximum approximation error is a result of the design process and cannot be prescribed. Prescribing a tolerance scheme is only possible if some type of constrained optimization is used. Constraints are not only useful for specifying tolerance schemes, but they also allow the mix of different error criteria. The constrained least squared error criterion mixes Chebyshev and least squares criteria. It has turned out that filters mixing both criteria in an appropriate way can have more desirable properties than filters that are optimum with respect to only one of these criteria.

This research has been supported by the Austrian Science Fund (FWF) under grant P11133-MAT.

An important special case of constrained least squares designs are filters that have Chebyshev passbands and least squares stopbands. These filters simultaneously minimize the distortion of desired signals in the passbands and the power of broadband noise in the stopbands. An early paper on the design of digital FIR filters of this type is [1]. More research on the constrained least squares design of FIR filters has been published in [2, 3, 4] where the design of linear phase FIR filters is considered. The more complicated case of arbitrary magnitude and phase or group delay responses has been considered in [5, 6, 7, 8]. However, for the complex approximation case, no reliable programs have become available so far.

2. PROBLEM

Define the complex error function

$$E(\omega, \boldsymbol{h}) = H(e^{j\omega}, \boldsymbol{h}) - D(e^{j\omega}), \qquad (1)$$

where $H(e^{j\omega}, \boldsymbol{h})$ is the actual frequency response

$$H(e^{j\omega}, \boldsymbol{h}) = \sum_{n=0}^{N-1} h(n) e^{-jn\omega}$$

of a length N FIR filter with coefficients $\boldsymbol{h} = [h(0), h(1), \ldots, h(N-1)]^T$. The complex function $D(e^{j\omega})$ is the desired frequency response containing the desired magnitude and phase responses. We define a complex weighted least squares filter design problem by

$$\underset{\boldsymbol{h}}{\text{minimize}} \int_{\Omega} W(\omega) |E(\omega, \boldsymbol{h})|^2 d\omega, \qquad (2)$$

with a non-negative weighting function $W(\omega)$. The set Ω contains all frequency intervals on which an approximation is desired. Adding constraints to problem (2) results in a complex constrained least squares filter design problem. In [9], we motivate that it is most useful to impose constraints on magnitude and phase errors. We believe that the group delay error is not an important quantity and can be ignored if we consider the phase error instead. The magnitude and phase errors can be constrained independently of each other (see [7, 9]), or they can be constrained by constraining the magnitude of the complex error function (1). If

$$|E(\omega, \boldsymbol{h})| \leq \delta(\omega), \quad \omega \in \Omega_B, \tag{3}$$

is satisfied for some strictly positive constraint function $\delta(\omega)$, then the magnitude and phase error functions $E_m(\omega, \boldsymbol{h}) = \left|H(e^{j\omega}, \boldsymbol{h})\right| - \left|D(e^{j\omega})\right|$ and $E_\phi(\omega, \boldsymbol{h}) = \arg\left\{H(e^{j\omega}, \boldsymbol{h})\right\} - \arg\left\{D(e^{j\omega})\right\}$ satisfy

$$\begin{aligned} E_m(\omega, \boldsymbol{h}) &\leq \delta(\omega), \quad \omega \in \Omega_B, \\ E_\phi(\omega, \boldsymbol{h}) &\leq \arcsin\left(\frac{\delta(\omega)}{|D(e^{j\omega})|}\right), \quad \omega \in \Omega_B. \end{aligned}$$

Hence, bounding the magnitude of the complex error function also bounds magnitude and phase errors. The set Ω_B is the union of all frequency intervals where error bounds are desired.

The problem we consider here is the weighted least squares problem (2) with the additional constraint (3). This problem is a convex quadratic programming problem. If a solution exists, it is unique.

3. ALGORITHM

The optimum solution to the constrained least squares problem (2,3) will satisfy the inequality constraint (3) with equality only at a finite number of frequency points. It is sufficient to impose the constraint (3) only at these frequency points in order to compute the optimum solution to the constrained least squares problem. However, this set of frequency points is not known in advance. The proposed design algorithm identifies a set of constraints that contains the relevant frequency points by solving a sequence of quadratic programming problems. Each of these problems contains only a relatively small number of constraints. In each iteration step, a new set of constraints is formulated by exchanging several old constraints for several new constraints. The total number of constraints generally changes from one iteration step to the next. Such algorithm are called *implicit multiple exchange* algorithms. Implicit multiple exchange algorithm for solving the complex Chebyshev approximation problem have been published in [10, 11]. The algorithm given in [11] can also be used to solve constrained least squares problems if they are convex. However, its computational effort is very high compared to the algorithm presented in this paper. The proposed multiple exchange algorithm can be stated as follows:

0. $k := 0$. Solve the unconstrained quadratic minimization problem (2) for $\boldsymbol{h}^{(0)}$.
1. Determine the local maxima of $|E(\omega, \boldsymbol{h}^{(k)})| - \delta(\omega)$, $\omega \in \Omega_B$. If $|E(\omega, \boldsymbol{h}^{(k)})| \leq \delta(\omega)$, $\omega \in \Omega_B$, is satisfied up to some specified tolerance, stop. Otherwise, go to 2.
2. Determine the set Ω_{viol} of local maximizers of $|E(\omega, \boldsymbol{h}^{(k)})| - \delta(\omega)$ that satisfy $|E(\omega, \boldsymbol{h}^{(k)})| > \delta(\omega)$.
3. Formulate a new set of constraints by using the current active constraints and the new constraints
$$\operatorname{Re}\left[E(\omega, \boldsymbol{h})e^{-j\arg\left\{E(\omega, \boldsymbol{h}^{(k)})\right\}}\right] \leq \delta(\omega), \; \omega \in \Omega_{viol}.$$
4. $k := k+1$. Compute $\boldsymbol{h}^{(k)}$ by solving the quadratic programming problem given by (2) and the constraints determined in step 3. Go to 1.

The vector $\boldsymbol{h}^{(k)}$ denotes the solution of the subproblem in iteration step k. In step 3, the *current active constraints* denote the constraints of the current subproblem that are satisfied with equality. In the first iteration step ($k = 0$), there are of course no current active constraints, and the set of new constraints formulated in step 3 only consists of the constraints determined by the set Ω_{viol}. For solving the quadratic programming subproblems in step 4, it is advantageous to use the dual formulation (see e. g. [12]). A more detailed discussion of the algorithm is given in [9].

4. EXAMPLE

We design a chirp-lowpass filter according to the following specification:

$$D(e^{j\omega}) = \begin{cases} e^{j\phi_d(\omega)}, & 0 \leq \omega \leq \omega_p \\ 0, & \omega_s \leq \omega \leq \pi \end{cases}, \tag{4}$$

with the desired phase response $\phi_d(\omega)$ given by

$$\phi_d(\omega) = -\frac{N-1}{2}\omega - 8\pi\left(\frac{\omega}{\omega_p} - \frac{1}{2}\right)^2, \; 0 \leq \omega \leq \omega_p, \tag{5}$$

where N is the filter length, and ω_p and ω_s are the passband and stopband edges, respectively. The desired phase response $\phi_d(\omega)$ corresponds to a linearly ascending desired group delay response. We choose $\omega_p = 0.2\pi$, $\omega_s = 0.225\pi$, and $N = 201$. We choose the constraint function $\delta(\omega)$ according to

$$\delta(\omega) = \begin{cases} 7 \cdot 10^{-3}, & 0 \leq \omega \leq \omega_p \\ 10^{-45/20}, & \omega_s \leq \omega \leq \pi \end{cases},$$

which corresponds to a maximum passband error of 0.007 and a minimum stopband attenuation of 45 dB. The weighting function $W(\omega)$ in (2) is chosen to be 1 in

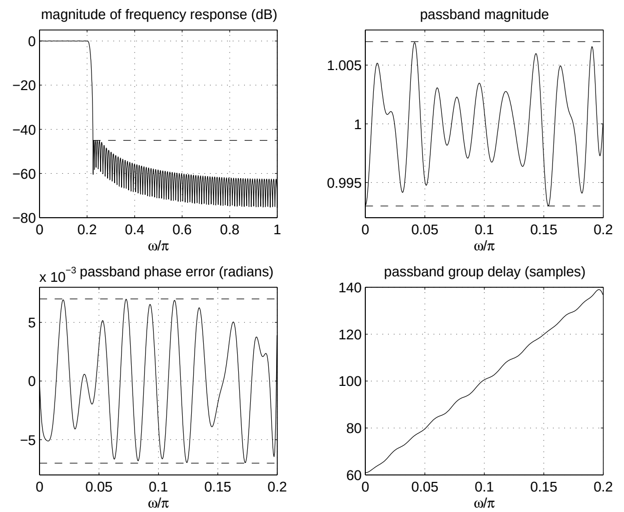

Figure 1: Constrained least squares design of chirp-lowpass filter (length $N = 201$). Dashed lines: constraints.

the passband and 500 in the stopband. Consequently, error energy is minimized mainly in the stopband. For designing the optimum filter, we use the Matlab program `exch_ce` given in section 5. The filter specification is generated by the following commands:

```
N=201; omp=.2*pi;
om=pi*[linspace(0,.2,800),linspace(.225,1,2800)];
Pd=-(N-1)/2*om-8*pi*(om/omp-.5).^2;
D=[exp(j*Pd(1:800)),zeros(1,2800)];
W=[ones(1,800),500*ones(1,2800)];
dc=[.007*ones(1,800),10^(-2.25)*ones(1,2800)];
```

The filter is designed by the command

```
h = exch_ce(N,om,D,W,dc);
```

Fig. 1 shows the magnitude of the designed frequency response, the details of the passband magnitude behavior, the deviation of the passband phase response from the desired phase response (5), and the passband group delay response. Dashed lines indicate the constraints imposed by the function $\delta(\omega)$.

5. MATLAB PROGRAM

The following Matlab program implements the proposed multiple exchange algorithm. It works with a grid of frequencies such that the integral in (2) is replaced by a sum. The constraint (3) is also formulated on this frequency grid. Choosing $10 - 20N$ grid points results in a good approximation to the continuous design problem. Note that the algorithm given in section 3 also works for continuous frequency intervals. In this case, the determination of local maxima in step 1 must be done on a continuum. This can be achieved by using Newton's method for refining local maxima determined on a grid [3]. The program has been written for Matlab 5.1.

```
function h = exch_ce(N,om,D,W,dc)
% EXCH_CE: constrained least squares FIR
% filter design in the complex domain
% h = exch_ce(N,om,D,W,dc)
%
% h      (real-valued) filter impulse response
% N      filter length
% om     frequency grid (0 <= om <= pi)
% D      complex desired frequency response on the
%        grid om
% W      positive weighting function on the
%        grid om
% dc     positive constraint function on the
%        grid om;
%        for intervals without constraints specify
%        negative dc
%
% EXAMPLE: low-delay bandpass filter with passband
% and stopband constraints
% om=pi*[linspace(0,.34,750),...
%        linspace(.4,.6,500),linspace(.66,1,750)];
% D=[zeros(1,750),exp(-j*30*om(751:1250)),...
%    zeros(1,750)];
% W=[500*ones(1,750),ones(1,500),500*ones(1,750)];
% dc=[.001*ones(1,750),.01*ones(1,500),...
%     .001*ones(1,750)];
% h = exch_ce(100,om,D,W,dc);
%
% Author: Mathias C. Lang
% Vienna University of Technology, Aug. 98
om=om(:);D=D(:);W=W(:);dc=dc(:);Lom=length(om);
tol = 1e-3; se = sqrt(eps);
% ---- set up objective function -----------------
t=zeros(N,1); c=t;
for i = 1:N,
   t(i) = W.'*cos((i-1)*om);
   c(i) = W.'*(real(D).*cos((i-1)*om) - ...
               imag(D).*sin((i-1)*om));
end
t=t/Lom; c=c/Lom;
% ---- solve unconstrained L2 problem -----------
[L,d] = invtoep(t);
for i=1:N, L(i:N,i)=L(i:N,i)*sqrt(d(i));end
%L = inv(chol(toeplitz(t)));
h = L*(L'*c); h_uc = h;
% ---- prepare for iteration ---------------------
Aact=[]; bact=[]; allconstr = 1;
IB = find(dc>=0); if ~length(IB), return; end
if length(IB) < length(dc), allconstr = 0; end
Dm = abs(D(IB)); Dp = angle(D(IB));
dc = dc(IB); om = om(IB); evec = exp(-j*om);
% ---- iterate ----------------------------------
while 1,
   % ---- compute error maxima -------------------
   E = polyval(h(N:-1:1),evec) - D(IB);
   Em = abs(E); Ep = angle(E);
   Em_dc = Em-dc; Imax = locmax(Em_dc);
   % ---- find violating maxima ------------------
   Iviol=find(Em_dc(Imax)>0); Iviol=Imax(Iviol);
   if length(Iviol)==length(Imax)&allconstr,
```

```
    disp(' There is no feasible solution.');
    disp(' Relax the constraints.'); break;
  end
  % ---- check stopping criterion ---------------
  if all(Em_dc(Iviol)<=max(dc(Iviol)*tol,se)),
     break;
  end
  % ---- formulate new constraints --------------
  omviol = om(Iviol); Epviol = Ep(Iviol);
  Dmviol = Dm(Iviol); Dpviol = Dp(Iviol);
  Anew = cos(omviol*(0:N-1)+Epviol(:,ones(N,1)));
  bnew = Dmviol.*cos(Dpviol-Epviol) + dc(Iviol);
  % ---- make a plot ----------------------------
  plot(om/pi,Em_dc,omviol/pi,Em_dc(Iviol),'rx');
  title('constraint violation');
  xlabel('\omega/\pi'); grid, drawnow
  % ---- solve subproblem (dual form) -----------
  A=[Aact;Anew]; b=[bact;bnew]; nb=length(b);
  AL = A*L; H = AL*AL'; f = b-A*h_uc;
  for i=1:nb, H(i,i) = H(i,i)+se; end
  l = nnqmin(H,f);
  %l = qp(H,f,[],[],zeros(nb,1));
  h = h_uc-L*(AL'*l);
  % ---- find active constraints ----------------
  act = find(l>se);Aact = A(act,:);bact = b(act);
end
```

The program `exch_ce` calls the program `locmax` that computes the indices of the local maxima of a data vector:

```
function I = locmax(x)
% LOCMAX: I = locmax(x)
% finds indices of local maxima of data x
x = x(:); n = length(x);
if n, I = find(x > [x(1)-1;x(1:n-1)] & ...
               x > [x(2:n);x(n)-1]);
else, I = []; end
```

The program `invtoep` called by `exch_ce` computes the Cholesky factorization of the inverse of a Hermitian Toeplitz matrix in an efficient way. This program can be obtained from the author. It can, however, be replaced by a less efficient standard Matlab command as indicated in the program `exch_ce`. The duals of the quadratic programming subproblems are solved by the program `nnqmin`. This program can also be obtained from the author. Alternatively, the subproblems can be solved by the program `qp` contained in the Matlab Optimization Toolbox. However, the computation times become considerably larger. Using `nnqmin`, the design presented in section 4 takes 20 seconds on a 166 MHz Pentium PC. If `qp` is used instead, the design takes 150 seconds.

6. CONCLUSION

We have considered the constrained least squares filter design problem for the case that there are specifications on magnitude and phase responses. We have presented a fast implicit multiple exchange algorithm that solves this problem. An arbitrary constraint function bounding the magnitude of the approximation error can be specified. The filters that can be designed by the proposed algorithm are mixes between complex least squares and complex Chebyshev filters. Arbitrary trade-offs between these two standard criteria are possible. We have provided an efficient Matlab implementation of the proposed algorithm.

7. REFERENCES

[1] J. T. Lewis, R. Murphy, and D. W. Tufts, "Design of Minimum Noise Digital Filters Subject to Inequality Constraints Using Quadratic Programming," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, pp. 434-436, Oct. 1976.

[2] 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.

[3] I. W. Selesnick, M. Lang, and C. S. Burrus, "Constrained Least Square Design of FIR Filters without Specified Transition Bands," *IEEE Transactions on Signal Processing*, vol. SP-44, pp. 1879-1892, August 1996.

[4] J. W. Adams and J. L. Sullivan, "Peak-Constrained Least-Squares Optimization," *IEEE Transactions on Signal Processing*, vol. SP-46, pp. 306-321, Feb. 1998.

[5] M. Lang and J. Bamberger, "Nonlinear Phase FIR Filter Design According to the L_2 Norm with Constraints for the Complex Error," *Signal Processing*, vol. 36, pp. 31-40, March 1994.

[6] J. L. Sullivan and J. W. Adams, "A Nonlinear Optimization Algorithm for Asymmetric FIR Digital Filters," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, pp. 869-872, April-May 1995.

[7] M. C. Lang, "Constrained Least Square Design of FIR Filters with Arbitrary Magnitude and Phase Responses," in *Proc. IEEE Int. Symp. Circuits and Systems*, Hong Kong, June 1997, vol. 4, pp. 2264-2267.

[8] M. C. Lang, "An Iterative Reweighted Least Squares Algorithm for Constrained Design of Nonlinear Phase FIR Filters," in *Proc. IEEE Int. Symp. Circuits and Systems*, Monterey (CA), May/June 1998.

[9] M. C. Lang, *Constrained Design of Digital Filters with Arbitrary Magnitude and Phase Responses*, Ph.D. Thesis, Vienna University of Technology, Austria, in preparation.

[10] D. Burnside and T. W. Parks, "Optimal Design of FIR Filters with the Complex Chebyshev Error Criteria," *IEEE Transactions on Signal Processing*, vol. SP-43, pp. 605-616, March 1995.

[11] A. W. Potchinkov and R. M. Reemtsen, "The Design of FIR Filters in the Complex Plane by Convex Optimization," *Signal Processing*, vol. 46, pp. 127-146, 1995.

[12] R. Fletcher, *Practical Methods of Optimization*, Wiley, Chichester, 2nd Edition, 1987.